Base de données *Bureau du Mercure*
Procédures d'importation des données externes

État : 15 octobre 2026

Introduction **3**
Catalogue général de la Bibliothèque nationale de France **4**
Auteurs 4
Objectif 4
Données concernées 4
Utilisation 4
Procédure détaillée 4
Oeuvres 19
Objectif 19
Données concernées 19
Utilisation 21
Procédure détaillée 21
GeoNames **134**
Données relatives aux lieux 134
Objectifs 134
Données concernées 134
Utilisation 135
Procédure détaillée 135
Copies numériques **212**
Fichiers et propriétés 212
Objectifs 212
Données concernées 212
Utilisation 212
Procédure détaillée 212
Les énigmes de Mercure. Base de données des énigmes, logogriphes et charades du *Mercure galant* et du *Mercure de France*, 1673-1799 ..... **222**
Données relatives aux textes 222
Objectifs 222
Données concernées 222
Utilisation 223
Procédure détaillée 223
Reconnaissance textuelle et balisage sur la plateforme *Transkribus* **389**
Données relatives aux textes 390
Objectifs 390
Données concernées 390
Utilisation 390
Procédure détaillée 391
Résumés produits par l'intelligence artificielle **418**
Données relatives aux textes 418
Objectifs 418
Données concernées 418
Utilisation 418
Procédure détaillée 419

Annexe : traitement des données et usage du modèle Mistral 422
Introduction 422
Notions générales 423
Données d’entrée 423
Pipeline de résumé 424
Formats de sortie et intégration 425
Limites et précautions 425

Introduction

La base de données *Bureau du Mercure* comprend des données produites par les collaborateurs du projet et des données externes que nous retouchons en partie avant de les importer. Ces données externes comptent :

- les notices d'autorité et les œuvres du *Catalogue général de la Bibliothèque nationale de France* ;
- les fiches de lieux de la base de données *GeoNames* ;
- les numérisations en format PDF des livraisons du *Mercure*, issues de plusieurs bibliothèques numériques dont *Gallica* et *Google Books* ;
- les données relatives aux énigmes, logogriphes et charades du *Mercure*, issues du site *Les énigmes de Mercure* ;
- la reconnaissance textuelle et d'autres données produites à l'aide de la plateforme *Transkribus* ;
- les résumés des textes du *Mercure*, produits par l'intelligence artificielle à partir de la reconnaissance textuelle.

Nous décrivons ici les processus de sélection, d'extraction, de modification et d'importation de ces différents lots de données, et l'usage que nous en faisons localement.

Les différentes procédures d'importation documentent avec précision le choix des données importées et la façon dont nous les modifions pour les adapter à nos besoins. Néanmoins, elles ne sont pas reproductibles telles quelles, parce qu'elles dépendent de l'écosystème spécifique de notre base de données. Elles reposent sur le système de gestion de contenu *Drupal* et sur plusieurs modules développés par sa communauté, documentés sur https://www.drupal.org (22 décembre 2025). Elles dépendent également de nos propres compétences en informatique, qui nous conduisent à privilégier le langage PHP et le format CSV.

Parmi les segments de code reproduits ci-après, plusieurs mériteraient d'être améliorés pour corriger de petites erreurs ou combler des lacunes, problèmes que nous avons identifiés après l'importation des données concernées. C'est en particulier le cas de la procédure d'importation des notices d'autorité du *Catalogue général de la Bibliothèque nationale de France*. Dans l'état actuel, chaque fiche de personne doit être modifiée manuellement après l'importation pour compléter et améliorer l'affichage des remarques de la Bibliothèque nationale de France, des dates de naissance et de mort (quand on ne connaît que le siècle ou la décennie) et des données relatives aux lieux d'activité des éditeurs. La procédure « Résumés produits par l'intelligence artificielle » a été conçue et rédigée en collaboration avec Ivo Durisch de la société Lineasoft (Riva San Vitale, Suisse).

Ce document dialogue avec deux autres fichiers qui comprennent les « Ontologies » et les « Référentiels » de la base de données *Bureau du Mercure*. Toute la documentation relative à la base de données est susceptible d'être complétée ou corrigée au fil de notre travail. Elle est déposée sur le site vitrine du projet de recherche « Le *Mercure de France* et l'institution littéraire » : https://bureaudumercure.org/public/project/database.html (22 décembre 2025).

Timothée Léchot, Université de Fribourg, 8 juin 2026.

Catalogue général de la Bibliothèque nationale de France

AUTEURS

Objectif

Le *Catalogue général de la Bibliothèque nationale de France* propose des notices d'autorité de personnes et de collectivités. Grâce à l'outil d'exportation du site, nous récupérons sous la forme de fichiers CSV l'ensemble des données pertinentes relatives à ces personnes et à ces collectivités pour les intégrer à notre base de données. Nous disposons ainsi de fiches de personnes partiellement conformes aux notices d'autorité de la Bibliothèque nationale de France.

Données concernées

L'importation concerne toutes les personnes et collectivités nées ou fondées avant 1821, soit 202 725 entités le 14 juin 2023.

À partir des notices d'autorité de la Bibliothèque nationale de France, nous récupérons les sous-zones (précédées d'un $) et les positions suivantes dans le format d'échange Intermarc :

- 008 : zone d'informations générales codées. Pos. 27 : ère (de naissance ou de création). Pos. 36 : fiabilité de la date (de naissance ou de création). Pos. 37 : ère (de mort ou de fin). Pos. 46 : fiabilité de la date (de mort ou de fin). Pos. 2831 : année (de naissance ou de création). Pos. 3841 : année (de mort ou de fin).
- 031 : ISNI. $a : ISNI.
- 100 : nom de personne phyique – forme retenue. $a : élément d'entrée. $d : dates biographiques. $e : qualificatif. $m : élément(s) du nom rejeté(s).
- 110 : nom de collectivité ou de congrès – forme retenue. $a : élément d'entrée. $b : sous-vedette. $c : lieu. $q : autre qualificatif.
- 600 : note d'information publique. $a : texte.
- 603 : note sur les lieux de naissance et de mort. $a : lieu de naissance. $b : lieu de mort.
- 604 : note sur l'adresse d'activité et l'enseigne des imprimeurs – libraires, marchands d'estampes ou relieurs. $a : ville d'activité. $d : dates d'activité dans la ville. $e : enseigne(s). $l : adresse(s) dans la ville.

À propos des zones et sous-zones Intermarc, voir la page « INTERMARC Autorités de diffusion » de la Bibiliothèque nationale de France : https://www.bnf.fr/fr/intermarc-autorites-de-diffusion (14 juin 2023).

Utilisation

Sur le site public, seules les données des personnes et de collectivités liées à un ou plusieurs autres éléments seront accessibles. Certaines données auront été modifiées, complétées ou supprimées par les collaborateurs, conformément aux besoins du site. Ainsi, une fois chargées dans *Drupal*, les données ne peuvent plus être réimportées et elles ne profitent plus des mises à jour auxquelles procède la Bibliothèque nationale de France. Que les données soient modifiées ou non, leur source est toujours indiquée et un lien hypertexte permet d'accéder à la fiche d'autorité concernée sur le site de la Bibliothèque nationale de France.

Procédure détaillée

1. Exportation des notices d'autorité.
    a. Accéder à https://catalogue.bnf.fr/recherche-autorite.do?pageRech=rat
    b. « Sélectionner un type de notice » : cocher « Personne » et « Collectivité »

c. « Par date » : choisir une « Date de naissance ou de création ». Nous téléchargons les données par petites tranches chronologiques (moins de 30 000 résultats par tranche).
d. « Lancer la recherche ».
e. Sélectionner « Tous les résultats ».
f. Cliquer sur le bouton « Exporter dans un tableau ».
g. Renseigner le champ avec la liste de sous-zones suivantes : 031$a;100$a;100$d;100$e;100$m;110$a;110$b;110$c;110$q;600$a;603$a;603$b;604$a;604$d;604$e;604$l;
h. Sélectionner « Intermarc », « 1 colonne par sous-zone », « Données avec étiquette ».
i. « Valider ».
j. Répéter les opérations pour toutes les autres tranches chronologiques.

2. Rassembler l'ensemble des données au sein d'un même fichier CSV.
3. Importer le fichier sur un serveur MariaDB 10.
    a. Dans une base de données « bnf », préconfigurer une table « person » contenant une unique colonne « col_2 » de type TEXT(1000000). Exemple de commande SQL :

```
CREATE TABLE person (
    col_2 TEXT(1000000)
);
```

    b. Importer l'ensemble des données. Exemple de commande SQL :

```
LOAD DATA INFILE '/path/to/filename.csv' INTO TABLE person;
```

    c. Devant la colonne « col_2 », ajouter une colonne « col_1 » de type INT.
    d. Ajouter une valeur autoincrémentielle à « col_1 » et définir cette colonne comme la clef primaire de la table.

```
ALTER TABLE person
MODIFY col_1 INT NOT NULL AUTO_INCREMENT PRIMARY KEY;
```

4. Répéter toutes les opérations antérieures pour la zone 008 en créant une nouvelle table « person_008 » et en conservant toutes les colonnes du fichier d'exportation de la Bibliothèque nationale de France.
    a. Après exportation et avant de fusionnier les fichiers CSV, ceux-ci doivent être retouchés, par exemple sur Microsoft Excel, de manière qu'ils possèdent tous le même nombre de colonnes et que ces colonnes se suivent dans le même ordre.
    b. Exemple de commande SQL pour la préconfiguration de la table « person_008 » :

```
CREATE TABLE person_008 (
    Identifiant_ARK TEXT(1000000),
    Notice_BnF TEXT(1000000),
    Type_de_notice TEXT(1000000),
    008_pos_0001 TEXT(1000000),
    008_pos_0203 TEXT(1000000),
    008_pos_0405 TEXT(1000000),
    008_pos_0607 TEXT(1000000),
    008_pos_0809 TEXT(1000000),
    008_pos_1011 TEXT(1000000),
```

```
    008_pos_1213 TEXT(1000000),
    008_pos_1416 TEXT(1000000),
    008_pos_17 TEXT(1000000),
    008_pos_21 TEXT(1000000),
    008_pos_2223 TEXT(1000000),
    008_pos_24 TEXT(1000000),
    008_pos_2526 TEXT(1000000),
    008_pos_27 TEXT(1000000),
    008_pos_2831 TEXT(1000000),
    008_pos_3233 TEXT(1000000),
    008_pos_3435 TEXT(1000000),
    008_pos_36 TEXT(1000000),
    008_pos_37 TEXT(1000000),
    008_pos_3841 TEXT(1000000),
    008_pos_4243 TEXT(1000000),
    008_pos_4445 TEXT(1000000),
    008_pos_46 TEXT(1000000),
    008_pos_4750 TEXT(1000000),
    008_pos_51 TEXT(1000000),
    008_pos_5255 TEXT(1000000),
    008_pos_56 TEXT(1000000),
    008_pos_59 TEXT(1000000),
    008_pos_60 TEXT(1000000),
    008_pos_61 TEXT(1000000),
    008_pos_62 TEXT(1000000),
    008_pos_63 TEXT(1000000),
    008_pos_64 TEXT(1000000)
);
```

c. Exemple de commande SQL pour l'importation :

```
LOAD DATA INFILE '/path/to/filename.csv' INTO TABLE person_008 FIELDS TERMINATED
BY ';' ;
```

d. Fusionner les deux tables person et person_008 dans une nouvelle table person_bnf, en associant les deux tables par le numéro de notice de la Bibliothèque nationale de France. Pour ce faire, il faut au préalable extraire ce numéro de notice de la table person, par exemple en PHP :

```
<?php

//Connect to the database.

$mysqli = new MySQLi('localhost','user','password','database');

if(mysqli_connect_errno())
    {
        printf('Échec de la connexion : base de données momentanément
inaccessible.');
        exit();
    }

//Query the database.

$result_set_2 = $mysqli->query('SELECT * FROM person');

if($result_set_2->num_rows > 0){
    while(null !==($rows = $result_set_2->fetch_assoc()))
```

```
    {

//      Retrieve col_2 from the database.

        $data = $rows['col_2'];

//      Split the lines.

        $split = explode(';', $data);

//      Echo the record numbers.

        echo $split[1].'<br/>';
    }
}

?>
```

e. Cette liste de numéros peut être ajoutée à une exportation de la table « person », par exemple grâce à Microsoft Excel, en ajoutant une colonne « bnf_number ».
f. Le fichier CSV Excel ainsi modifié peut être réimporté dans la base de données, de sorte que « col_2 » et la nouvelle colonne « bnf_number » forment une nouvelle table « person_temporary ». Les colonnes « bnf_number » et « Notice_BnF » devraient être de nature INT(11) et indexées.
g. Enfin, il est possible de fusionner la table « person_temporary » et la table « person_008 » dans une nouvelle table « person_bnf », grâce à la commande SQL suivante :

```
CREATE TABLE person_bnf
AS
SELECT *
FROM person_temporary
LEFT JOIN person_008
ON person_temporary.bnf_number LIKE person_008.notice_BnF;
```

5. À présent, nous sommes en mesure de créer un fichier CSV importable dans la base de données définitive. Il s'agit de retoucher les données brutes de la Bibliothèque nationale de France, de façon à permettre leur importation dans les champs correspondants de la base de données de notre site, grâce à la fonction d'importation du module Webform de *Drupal*. Pour ce faire, nous appelons les données brutes rassemblées dans la table « person_bnf » de la base de données « bnf », nous les modifions en langage PHP, puis nous affichons sur une page web l'ensemble des données modifiées, de manière à pouvoir les copier et les coller dans un nouveau fichier CSV.
    a. Modifier les données en langage PHP. Exemple de code (fichier « csv_bnf_person.php ») :

```
<?php
//Persons

//Determine the first and last row to be echoed. For $end_row, the value 0 will
echo all the rows.

$start_row = 0;
$end_row = 0;
```

```php
//Connect to the database.

$mysqli = new MySQLi('localhost','user','password','database');

if(mysqli_connect_errno())
    {
        printf('Échec de la connexion : base de données momentanément
inaccessible.');
        exit();
    }

//Total number of items.

$result_set_1 = $mysqli->query('SELECT COUNT(*) FROM person_bnf');
$total = $result_set_1->fetch_row();

if($end_row == 0){
    $end_row = $total[0];
}

//Query the database.

$result_set_2 = $mysqli->query('SELECT * FROM person_bnf');


//Retrieve and manage data.

$tmpCount = 0;
$total_displayed = 0;
$output = NULL;

if($result_set_2->num_rows > 0){
    while(null !==($rows = $result_set_2->fetch_assoc()))
    {

//      Assign NULL to the variables.

        $all_remark = $all_remark_birth = $all_remark_bnf = $all_remark_death =
$all_validity = $bnf_number = $colon = $data = $en_dash_1 = $en_dash_2 =
$left_bracket = $per_assumed_year_of_birth = $per_assumed_year_of_death =
$per_bnf_record = $per_bnf_record_number = $per_collectivity_name =
$per_data_origin_bnf = $per_first_name = $per_isni = $per_last_name =
$per_place_of_birth = $per_place_of_death = $per_qualifier = $per_type =
$per_year_of_birth = $per_year_of_birth_raw = $per_year_of_death =
$per_year_of_death_raw = $right_bracket = $semicolon_1 = $semicolon_2 = $split =
$split_031a = $split_100a = $split_100d = $split_100e = $split_100m = $split_110a
= NULL;

//      Count the lines and retrieve col_2 from the database.

        $tmpCount ++;
        $data = $rows['col_2'];

//      Retrieve columns corresponding to the 008 Intermarc zone

        $pos_27 = $rows['008_pos_27'];
        $pos_36 = $rows['008_pos_36'];
        $pos_2831 = $rows['008_pos_2831'];
        $pos_37 = $rows['008_pos_37'];
```

```
        $pos_46 = $rows['008_pos_46'];
        $pos_3841 = $rows['008_pos_3841'];

//      Add a fake Intermarc number at the end of the line.

        $data = $data.'000$a';

//      Manage semicolons and remove vertical bars.

        $data = str_replace(" ; ", "<span style='color:grey'>[space, semicolon,
space]</span>", $data);
        $data = str_replace("; ", "<span style='color:grey'>[space, semicolon,
space]</span>", $data);
        $data = str_replace(";", "<span style='color:blue'>[tab]</span>", $data);
        $data = str_replace("<span style='color:grey'>[space, semicolon,
space]</span>", " ; ", $data);
        $data = str_replace(" | ", "<span style='color:blue'>[tab]</span>",
$data);

//      Manage and escape double-quotes.

        $data = str_replace('"', '""', $data);
        $data = str_replace('« ', '""', $data);
        $data = str_replace('«', '""', $data);
        $data = str_replace(' »', '""', $data);
        $data = str_replace('»', '""', $data);

//      Split the lines.

        $split = explode("<span style='color:blue'>[tab]</span>", $data);

//      Create field per_type.

        if(stripos($split[2], 'Personne') !== false){
            $per_type = '255'; //Value matching "Personne" in taxonomy "Type de
notice".
        }
        elseif(stripos($split[2], 'Collectivité') !== false){
            $per_type = '256'; //Value matching "Collectivité" in taxonomy "Type
de notice".
        }

//      Create field per_first_name from 100$m.

        $split_100m = explode('100$m', $data);
        $split_100m = explode($split_100m[0].'100$m', $data);
        $split_100m = str_replace('100$m', '[space]', $split_100m[1]);
        $split_100m = preg_split('/(\d){3}\$./', $split_100m);

        $split_100m = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_100m[0]);

        $split_100m = explode('[space]', $split_100m);

        if($split_100m[0] == $split_100m[1]){
            $per_first_name = $split_100m[0];
        }else{
            $per_first_name = implode('[space]', $split_100m);
        }
```

```
        $per_first_name = str_replace('[space]', ' ', $per_first_name);

//      Create field per_last_name from 100$a.

        $split_100a = explode('100$a', $data);
        $split_100a = explode ($split_100a[0].'100$a', $data);
        $split_100a = str_replace('100$a', '[space]', $split_100a[1]);
        $split_100a = preg_split('/(\d){3}\$./', $split_100a);

        $split_100a = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_100a[0]);

        $split_100a = explode('[space]', $split_100a);

        if($split_100a[0] == $split_100a[1]){
            $per_last_name = $split_100a[0];
        }else{
            $per_last_name = implode('[space]', $split_100a);
        }

        $per_last_name = str_replace('[space]', ' ', $per_last_name);

//      Create field per_qualifier from 100$e.

        $split_100e = explode('100$e', $data);
        $split_100e = explode ($split_100e[0].'100$e', $data);
        $split_100e = str_replace('100$e', ', ', $split_100e[1]);
        $split_100e = preg_split('/(\d){3}\$./', $split_100e);
        $split_100e = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_100e[0]);

        $split_100e = explode(', ', $split_100e);

        if($split_100e[0] == $split_100e[1]){
            $per_qualifier = $split_100e[0];
        }else{
            $per_qualifier = implode(', ', $split_100e);
        }


//      Create field per_collectivity_name from 110$a, 110$b, 110$c, 110$q.

        $split_110a = explode('110$a', $data, 2);
        $split_110b = explode('110$b', $data, 2);
        $split_110c = explode('110$c', $data, 2);
        $split_110q = explode('110$q', $data, 2);

        if($split_110b[1] !== null || $split_110c[1] !== null || $split_110q[1]
!== null){
            $left_bracket = ' (';
            $right_bracket = ')';
        }

        if(($split_110b[1] !== null && $split_110c[1] !== null) || ($split_110b[1]
!== null && $split_110q[1] !== null)){
            $semicolon_1 = ' ; ';
        }
```

```
        if($split_110c[1] !== null && $split_110q[1] !== null){
            $semicolon_2 = ' ; ';
        }

        $split_110a = str_replace('110$a', ', ', $split_110a[1]);
        $split_110b = str_replace('110$b', ', ', $split_110b[1]);
        $split_110c = str_replace('110$c', ', ', $split_110c[1]);
        $split_110q = str_replace('110$q', ', ', $split_110q[1]);
        $split_110a = preg_split('/(\d){3}\$./', $split_110a);
        $split_110b = preg_split('/(\d){3}\$./', $split_110b);
        $split_110c = preg_split('/(\d){3}\$./', $split_110c);
        $split_110q = preg_split('/(\d){3}\$./', $split_110q);

        $per_collectivity_name =
$split_110a[0].$left_bracket.$split_110b[0].$semicolon_1.$split_110c[0].$semicolon
_2.$split_110q[0].$right_bracket;
        $per_collectivity_name = str_replace("<span
style='color:blue'>[tab]</span>", '', $per_collectivity_name);

//      Create field per_publisher_place_and_date from 604$a, 604$d, 604$e, 604$l.

        $split_604a = explode('604$a', $data, 2);
        $split_604d = explode('604$d', $data, 2);
        $split_604e = explode('604$e', $data, 2);
        $split_604l = explode('604$l', $data, 2);

        if($split_604d[1] !== null || $split_604l[1] !== null || $split_604e[1]
!== null){
            $colon = ' : ';
        }

        if(($split_604d[1] !== null && $split_604l[1] !== null) || ($split_604d[1]
!== null && $split_604e[1] !== null)){
            $en_dash_1 = ' – ';
        }

        if($split_604l[1] !== null && $split_604e[1] !== null){
            $en_dash_2 = ' – ';
        }

        $split_604a = str_replace('604$a', ', ', $split_604a[1]);
        $split_604d = str_replace('604$d', ', ', $split_604d[1]);
        $split_604e = str_replace('604$e', ', ', $split_604e[1]);
        $split_604l = str_replace('604$l', ', ', $split_604l[1]);
        $split_604a = preg_split('/(\d){3}\$./', $split_604a);
        $split_604d = preg_split('/(\d){3}\$./', $split_604d);
        $split_604e = preg_split('/(\d){3}\$./', $split_604e);
        $split_604l = preg_split('/(\d){3}\$./', $split_604l);

        $per_publisher_place_and_date =
$split_604a[0].$colon.$split_604d[0].$en_dash_1.$split_604l[0].$en_dash_2.$split_6
04e[0];
        $per_publisher_place_and_date = str_replace("<span
style='color:blue'>[tab]</span>", '', $per_publisher_place_and_date);

//      Create field per_year_of_birth from 100$d and 008 (pos. 27, 36 and 2831).
//      First, manage both year of birth and year of death.

        $split_100d = explode('100$d', $data);
```

```
        $split_100d = preg_split('/(\d){3}\$./', $split_100d[1]);

        $years = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_100d[0]);
        $years = preg_replace('/av(.*?)c/i', '[avJC]', $years);
        $years = preg_replace('/ap(.*?)c/i', '[apJC]', $years);
        $years = str_ireplace('[avJC].', '[avJC]', $years);
        $years = str_ireplace('[apJC].', '[apJC]', $years);
        $years = str_ireplace('ca', '?', $years);
        $years = str_ireplace('vers', '?', $years);
        $years = str_ireplace('v.', '?', $years);
        $years = str_ireplace('[ca].', '?', $years);
        $years = str_ireplace('??', '..', $years);

//      If the person is a collectivity, then use the 008 Intermarc zone for both
year of birth and year of death.

        if($per_type == '256'){

            $pos_27 = str_ireplace('-', '[avJC]', $pos_27);
            $pos_37 = str_ireplace('-', '[avJC]', $pos_37);

            $years = $pos_27.$pos_2831.$pos_36.'-'.$pos_37.$pos_3841.$pos_46;
            $years = str_ireplace('008$', '', $years);
        }

        if(stripos($years, '-') === false){
            $years = '';
        }

        if(str_starts_with($years, '-')){
            $years = str_replace('--', '-[avJC]', $years);
            $years = preg_replace('/^\-/', '[avJC]', $years);
        }elseif(stripos($years, '--') !== false){
            $years = '';
        }

//      Now determine the year of birth.

        $years = explode('-', $years);

        $per_year_of_birth_raw = $years[0];

        if(stripos($years[1], '[avJC]') !== false){
            $per_year_of_birth_raw = $per_year_of_birth_raw.' [avJC]';
        }

        if(stripos($per_year_of_birth_raw, '.') !== false){
            $per_year_of_birth = '';
        }else{
            $per_year_of_birth = $per_year_of_birth_raw;
        }

        $per_year_of_birth = preg_replace('/[^0-9]/', '', $per_year_of_birth);
        $per_year_of_birth = ltrim($per_year_of_birth, '0');

        if(stripos($per_year_of_birth_raw, '[avJC]') !== false &&
$per_year_of_birth != ''){
            $per_year_of_birth = '-'.$per_year_of_birth;
```

```
        }

//      Create field per_assumed_year_of_birth from 100$d and 008 (pos. 27, 36 and
2831).

        if(stripos($per_year_of_birth_raw, '?') !== false && $per_year_of_birth !=
''){
            $per_assumed_year_of_birth = '1';
        }

//      Create field per_year_of_death from 100$d and 008 (pos. 37, 46 and 3841).

        $per_year_of_death_raw = $years[1];

        if(stripos($per_year_of_death_raw, '.') !== false){
            $per_year_of_death = '';
        }else{
            $per_year_of_death = $per_year_of_death_raw;
        }

        $per_year_of_death = preg_replace('/[^0-9]/', '', $per_year_of_death);
        $per_year_of_death = ltrim($per_year_of_death, '0');

        if(stripos($per_year_of_death_raw, '[avJC]') !== false &&
$per_year_of_death != ''){
            $per_year_of_death = '-'.$per_year_of_death;
        }

//      Create field per_assumed_year_of_death from 100$d and 008 (pos. 37, 46 and
3841).

        if(stripos($per_year_of_death_raw, '?') !== false && $per_year_of_death !=
''){
            $per_assumed_year_of_death = '1';
        }

//      Create field per_place_of_birth from 603$a.

        $split_603a = explode('603$a', $data);
        $split_603a = explode ($split_603a[0].'603$a', $data);
        $split_603a = str_replace('603$a', ' ; ', $split_603a[1]);
        $split_603a = preg_split('/(\d){3}\$./', $split_603a);

        $per_place_of_birth = str_replace("<span style='color:blue'>[tab]</span>",
'', $split_603a[0]);

//      Create field per_place_of_death from 603$b.

        $split_603b = explode('603$b', $data);
        $split_603b = explode ($split_603b[0].'603$b', $data);
        $split_603b = str_replace('603$b', ' ; ', $split_603b[1]);
        $split_603b = preg_split('/(\d){3}\$./', $split_603b);

        $per_place_of_death = str_replace("<span style='color:blue'>[tab]</span>",
'', $split_603b[0]);

//      Create field per_bnf_record.

        $per_bnf_record = 'https://catalogue.bnf.fr/'.$split[0];
```

```
//      Create field per_bnf_record_number.

        $per_bnf_record_number = $split[1];

//      Create field per_isni from 031$a.

        $split_031a = explode('031$a', $data);
        $split_031a = explode ($split_031a[0].'031$a', $data);
        $split_031a = preg_split('/(\d){3}\$./', $split_031a[1]);

        $per_isni = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_031a[0]);

//      Create field per_data_origin_bnf.

        $per_data_origin_bnf = '1';

//      Create field all_remark_bnf from 600$a.

        $split_600a = explode('600$a', $data);
        $split_600a = explode ($split_600a[0].'600$a', $data);
        $split_600a = str_replace('600$a', ' - ', $split_600a[1]);
        $split_600a = preg_split('/(\d){3}\$./', $split_600a);

        $all_remark_bnf = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_600a[0]);

        if (str_ends_with($all_remark_bnf, '.') || str_ends_with($all_remark_bnf,
'?') || str_ends_with($all_remark_bnf, '!') || $all_remark_bnf == '') {
        }else{
            $all_remark_bnf = $all_remark_bnf.'.';
        }

//      Create field all_validity.

        $all_validity = '84'; //Value matching "Vérifié" in taxonomy "Validité".

//      Create field all_remark from 100$d. We use this field to specify the
century of birth and/or death of a person (and a collectivity), when the date of
birth/death of a person is given by the BnF in the form "16.." for seventeenth
century, "17.." for eighteenth century, and so on. We also correct the form "153."
to "les années 1530" (for example).

        if(preg_match('/^\d{3}\./', $per_year_of_birth_raw) ||
preg_match('/^\[avJC\]\d{3}\./', $per_year_of_birth_raw) ||
preg_match('/^\[apJC\]\d{3}\./', $per_year_of_birth_raw)){
            $all_remark_birth = substr($per_year_of_birth_raw, 0, 4);
            $all_remark_birth = ltrim($all_remark_birth, '0');
            $all_remark_birth = str_replace('.', '0', $all_remark_birth);

            if(stripos($per_year_of_birth_raw, '[avJC]') !== false){

                if($per_type == '256'){
                    $all_remark_birth = 'Fondé(e) dans les années
'.$all_remark_birth.' av. J.-C. ';
                }else{
                    $all_remark_birth = 'Né(e) dans les années
'.$all_remark_birth.' av. J.-C. ';
```

```
                }

            }else{

                if($per_type == '256'){
                    $all_remark_birth = 'Fondé(e) dans les années
'.$all_remark_birth;
                }else{
                    $all_remark_birth = 'Né(e) dans les années
'.$all_remark_birth;
                }
            }

        }elseif(preg_match('/^\d{2}\.\./', $per_year_of_birth_raw) ||
preg_match('/^\[avJC\]\d{2}\.\./', $per_year_of_birth_raw) ||
preg_match('/^\[apJC\]\d{2}\.\./', $per_year_of_birth_raw)){
            if(stripos($per_year_of_birth_raw, '[avJC]') !== false){
                $all_remark_birth = str_replace(array('26..', '25..', '24..',
'23..', '22..', '21..', '20..', '19..', '18..', '17..', '16..', '15..', '14..',
'13..', '12..', '11..', '10..', '09..', '08..', '07..', '06..', '05..', '04..',
'03..', '02..', '01..', '00..'), array('[cut]XXVII<sup>e</sup>[cut]',
'[cut]XXVI<sup>e</sup>[cut]', '[cut]XXV<sup>e</sup>[cut]',
'[cut]XXIV<sup>e</sup>[cut]', '[cut]XXIII<sup>e</sup>[cut]',
'[cut]XXII<sup>e</sup>[cut]', '[cut]XXI<sup>e</sup>[cut]',
'[cut]XX<sup>e</sup>[cut]', '[cut]XIX<sup>e</sup>[cut]',
'[cut]XVIII<sup>e</sup>[cut]', '[cut]XVII<sup>e</sup>[cut]',
'[cut]XVI<sup>e</sup>[cut]', '[cut]XV<sup>e</sup>[cut]',
'[cut]XIV<sup>e</sup>[cut]', '[cut]XIII<sup>e</sup>[cut]',
'[cut]XII<sup>e</sup>[cut]', '[cut]XI<sup>e</sup>[cut]',
'[cut]X<sup>e</sup>[cut]', '[cut]IX<sup>e</sup>[cut]',
'[cut]VIII<sup>e</sup>[cut]', '[cut]VII<sup>e</sup>[cut]',
'[cut]VI<sup>e</sup>[cut]', '[cut]V<sup>e</sup>[cut]', '[cut]IV<sup>e</sup>[cut]',
'[cut]III<sup>e</sup>[cut]', '[cut]II<sup>e</sup>[cut]',
'[cut]I<sup>er</sup>[cut]'), $per_year_of_birth_raw);
                $all_remark_birth = explode('[cut]', $all_remark_birth);
                $all_remark_birth = $all_remark_birth[1];

                if($per_type == '256'){
                    $all_remark_birth = '[cut]Fondé(e) au '.$all_remark_birth.'
siècle av. J.-C.[cut]';
                }else{
                    $all_remark_birth = '[cut]Né(e) au '.$all_remark_birth.'
siècle av. J.-C.[cut]';
                }

            }else{
                $all_remark_birth = str_replace(array('18..', '17..', '16..',
'15..', '14..', '13..', '12..', '11..', '10..', '09..', '08..', '07..', '06..',
'05..', '04..', '03..', '02..', '01..', '00..'),
array('[cut]XIX<sup>e</sup>[cut]', '[cut]XVIII<sup>e</sup>[cut]',
'[cut]XVII<sup>e</sup>[cut]', '[cut]XVI<sup>e</sup>[cut]',
'[cut]XV<sup>e</sup>[cut]', '[cut]XIV<sup>e</sup>[cut]',
'[cut]XIII<sup>e</sup>[cut]', '[cut]XII<sup>e</sup>[cut]',
'[cut]XI<sup>e</sup>[cut]', '[cut]X<sup>e</sup>[cut]', '[cut]IX<sup>e</sup>[cut]',
'[cut]VIII<sup>e</sup>[cut]', '[cut]VII<sup>e</sup>[cut]',
'[cut]VI<sup>e</sup>[cut]', '[cut]V<sup>e</sup>[cut]', '[cut]IV<sup>e</sup>[cut]',
'[cut]III<sup>e</sup>[cut]', '[cut]II<sup>e</sup>[cut]',
'[cut]I<sup>er</sup>[cut]'), $per_year_of_birth_raw);
                $all_remark_birth = explode('[cut]', $all_remark_birth);
```

```
                $all_remark_birth = $all_remark_birth[1];

                if($per_type == '256'){
                    $all_remark_birth = '[cut]Fondé(e) au '.$all_remark_birth.'
siècle[cut]';
                }else{
                    $all_remark_birth = '[cut]Né(e) au '.$all_remark_birth.'
siècle[cut]';
                }

            }
            $all_remark_birth = explode('[cut]', $all_remark_birth);
            $all_remark_birth = $all_remark_birth[1];
        }

        if($all_remark_birth != ''){

            if(stripos($per_year_of_birth_raw, '?') !== false){
                $all_remark_birth = $all_remark_birth.' ? ';
            }elseif(stripos($all_remark_birth, 'av. J.-C.') === false){
                $all_remark_birth = $all_remark_birth.'. ';
            }else{
                $all_remark_birth = $all_remark_birth.' ';
            }

        }

        if(preg_match('/^\d{3}\./', $per_year_of_death_raw) ||
preg_match('/^\[avJC\]\d{3}\./', $per_year_of_death_raw) ||
preg_match('/^\[apJC\]\d{3}\./', $per_year_of_death_raw)){
            $all_remark_death = substr($per_year_of_death_raw, 0, 4);
            $all_remark_death = ltrim($all_remark_death, '0');
            $all_remark_death = str_replace('.', '0', $all_remark_death);

            if(stripos($per_year_of_death_raw, '[avJC]') !== false){

                if($per_type == '256'){
                    $all_remark_death = 'Supprimé(e) dans les années
'.$all_remark_death.' av. J.-C.';
                }else{
                    $all_remark_death = 'Mort(e) dans les années
'.$all_remark_death.' av. J.-C.';
                }

            }else{

                if($per_type == '256'){
                    $all_remark_death = 'Supprimé(e) dans les années
'.$all_remark_death;
                }else{
                    $all_remark_death = 'Mort(e) dans les années
'.$all_remark_death;
                }

            }

        }elseif(preg_match('/^\d{2}\.\./', $per_year_of_death_raw) ||
preg_match('/^\[avJC\]\d{2}\.\./', $per_year_of_death_raw) ||
preg_match('/^\[apJC\]\d{2}\.\./', $per_year_of_death_raw)){
```

```
            if(stripos($per_year_of_death_raw, '[avJC]') !== false){
                $all_remark_death = str_replace(array('26..', '25..', '24..',
'23..', '22..', '21..', '20..', '19..', '18..', '17..', '16..', '15..', '14..',
'13..', '12..', '11..', '10..', '09..', '08..', '07..', '06..', '05..', '04..',
'03..', '02..', '01..', '00..'), array('[cut]XXVII<sup>e</sup>[cut]',
'[cut]XXVI<sup>e</sup>[cut]', '[cut]XXV<sup>e</sup>[cut]',
'[cut]XXIV<sup>e</sup>[cut]', '[cut]XXIII<sup>e</sup>[cut]',
'[cut]XXII<sup>e</sup>[cut]', '[cut]XXI<sup>e</sup>[cut]',
'[cut]XX<sup>e</sup>[cut]', '[cut]XIX<sup>e</sup>[cut]',
'[cut]XVIII<sup>e</sup>[cut]', '[cut]XVII<sup>e</sup>[cut]',
'[cut]XVI<sup>e</sup>[cut]', '[cut]XV<sup>e</sup>[cut]',
'[cut]XIV<sup>e</sup>[cut]', '[cut]XIII<sup>e</sup>[cut]',
'[cut]XII<sup>e</sup>[cut]', '[cut]XI<sup>e</sup>[cut]',
'[cut]X<sup>e</sup>[cut]', '[cut]IX<sup>e</sup>[cut]',
'[cut]VIII<sup>e</sup>[cut]', '[cut]VII<sup>e</sup>[cut]',
'[cut]VI<sup>e</sup>[cut]', '[cut]V<sup>e</sup>[cut]', '[cut]IV<sup>e</sup>[cut]',
'[cut]III<sup>e</sup>[cut]', '[cut]II<sup>e</sup>[cut]',
'[cut]I<sup>er</sup>[cut]'), $per_year_of_death_raw);
                $all_remark_death = explode('[cut]', $all_remark_death);
                $all_remark_death = $all_remark_death[1];

                if($per_type == '256'){
                    $all_remark_death = '[cut]Supprimé(e) au '.$all_remark_death.'
siècle av. J.-C.[cut]';
                }else{
                    $all_remark_death = '[cut]Mort(e) au '.$all_remark_death.'
siècle av. J.-C.[cut]';
                }

            }else{
                $all_remark_death = str_replace(array('19..', '18..', '17..',
'16..', '15..', '14..', '13..', '12..', '11..', '10..', '09..', '08..', '07..',
'06..', '05..', '04..', '03..', '02..', '01..', '00..'),
array('[cut]XX<sup>e</sup>[cut]', '[cut]XIX<sup>e</sup>[cut]',
'[cut]XVIII<sup>e</sup>[cut]', '[cut]XVII<sup>e</sup>[cut]',
'[cut]XVI<sup>e</sup>[cut]', '[cut]XV<sup>e</sup>[cut]',
'[cut]XIV<sup>e</sup>[cut]', '[cut]XIII<sup>e</sup>[cut]',
'[cut]XII<sup>e</sup>[cut]', '[cut]XI<sup>e</sup>[cut]',
'[cut]X<sup>e</sup>[cut]', '[cut]IX<sup>e</sup>[cut]',
'[cut]VIII<sup>e</sup>[cut]', '[cut]VII<sup>e</sup>[cut]',
'[cut]VI<sup>e</sup>[cut]', '[cut]V<sup>e</sup>[cut]', '[cut]IV<sup>e</sup>[cut]',
'[cut]III<sup>e</sup>[cut]', '[cut]II<sup>e</sup>[cut]',
'[cut]I<sup>er</sup>[cut]'), $per_year_of_death_raw);
                $all_remark_death = explode('[cut]', $all_remark_death);
                $all_remark_death = $all_remark_death[1];

                if($per_type == '256'){
                    $all_remark_death = '[cut]Supprimé(e) au '.$all_remark_death.'
siècle[cut]';
                }else{
                    $all_remark_death = '[cut]Mort(e) au '.$all_remark_death.'
siècle[cut]';
                }

            }
            $all_remark_death = explode('[cut]', $all_remark_death);
            $all_remark_death = $all_remark_death[1];
        }
```

```
        if($all_remark_death != ''){

            if(stripos($per_year_of_death_raw, '?') !== false){
                $all_remark_death = $all_remark_death.' ?';
            }elseif(stripos($all_remark_death, 'av. J.-C.') === false){
                $all_remark_death = $all_remark_death.'.';
            }

        }

        $all_remark = $all_remark_birth.$all_remark_death;

//      Manage fields with commas and/or quotes

        if(stripos($per_first_name, '"') !== false || stripos($per_first_name,
',') !== false){
            $per_first_name = '"'.$per_first_name.'"';
        }

        if(stripos($per_last_name, '"') !== false || stripos($per_last_name, ',')
!== false){
            $per_last_name = '"'.$per_last_name.'"';
        }

        if(stripos($per_qualifier, '"') !== false || stripos($per_qualifier, ',')
!== false){
            $per_qualifier = '"'.$per_qualifier.'"';
        }

        if(stripos($per_collectivity_name, '"') !== false ||
stripos($per_collectivity_name, ',') !== false){
            $per_collectivity_name = '"'.$per_collectivity_name.'"';
        }

        if(stripos($per_publisher_place_and_date, '"') !== false ||
stripos($per_publisher_place_and_date, ',') !== false){
            $per_publisher_place_and_date = '"'.$per_publisher_place_and_date.'"';
        }

        if(stripos($per_place_of_birth, '"') !== false ||
stripos($per_place_of_birth, ',') !== false){
            $per_place_of_birth = '"'.$per_place_of_birth.'"';
        }

        if(stripos($per_place_of_death, '"') !== false ||
stripos($per_place_of_death, ',') !== false){
            $per_place_of_death = '"'.$per_place_of_death.'"';
        }

        if(stripos($all_remark_bnf, '"') !== false || stripos($all_remark_bnf,
',') !== false){
            $all_remark_bnf = '"'.$all_remark_bnf.'"';
        }

        if(stripos($all_remark, '"') !== false || stripos($all_remark, ',') !==
false){
            $all_remark = '"'.$all_remark.'"';
        }
```

```
//      Combine the fields in CSV format.

        if(($tmpCount >= $start_row) && ($tmpCount <= $end_row)){
            $output .=
$per_type.','.$per_first_name.','.$per_last_name.','.$per_qualifier.','.$per_colle
ctivity_name.','.$per_publisher_place_and_date.','.$per_year_of_birth.','.$per_ass
umed_year_of_birth.','.$per_year_of_death.','.$per_assumed_year_of_death.','.$per_
place_of_birth.','.$per_place_of_death.','.$per_bnf_record.','.$per_bnf_record_num
ber.','.$per_isni.','.$per_data_origin_bnf.','.$all_remark_bnf.','.$all_remark.','
.$all_validity.'<br/>';
            $total_displayed ++;
        }
    }
}


//Echo the number of results.

echo 'Nombre de résultats : '.$total_displayed.'<br/>';
echo 'Affichage des résultats '.$start_row.' à '.$end_row.'<br/><br/>';

//Output in CSV format.

echo
'per_type,per_first_name,per_last_name,per_qualifier,per_collectivity_name,per_pub
lisher_place_and_date,per_year_of_birth,per_assumed_year_of_birth,per_year_of_deat
h,per_assumed_year_of_death,per_place_of_birth,per_place_of_death,per_bnf_record,p
er_bnf_record_number,per_isni,per_data_origin_bnf,all_remark_bnf,all_remark,all_va
lidity<br/>';
echo $output;
?>
```

b. Copier les données depuis un navigateur web et coller ces données dans de nouveaux fichiers CSV vides (20 000 lignes par fichier).
c. Importer les données dans *Drupal* par tranches de 20 000 lignes, grâce à l'outil d'importation du module Webform.

OEUVRES

Objectif

Le *Catalogue général de la Bibliothèque nationale de France* propose des notices bibliographiques détaillées de l'ensemble des œuvres (textes imprimés, œuvres musicales, documents iconographiques, documents cartographiques, documents d'archives et manuscrits, objets monétiformes, etc.). Grâce à l'outil d'exportation du site, nous récupérons sous la forme de fichiers CSV l'ensemble des données pertinentes relatives à ces œuvres pour les intégrer à notre base de données. Nous disposons ainsi de fiches d'œuvres partiellement conformes aux notices bibliographiques de la Bibliothèque nationale de France.

Données concernées

L'importation concerne toutes les œuvres publiées ou produites avant 1821, soit environ 1 400 000 entités le 11 août 2023 [1].

À partir des notices du catalogue, nous récupérons les sous-zones (précédées d'un $) dans le format d'échange Intermarc :

- 022 : ISSN et prix. $a : ISSN (saisi).
- 100 : vedette principale auteur personne physique. $3 : numéro de la notice d'autorité nom de personne liée.
- 110 : vedette principale auteur collectivité. $3 : numéro de la notice d'autorité collectivité liée.
- 243 : titre donné par le catalogueur. $a : titre. $b : autre titre du même auteur. $c : autre titre d'un auteur différent. $d : indication générale du type de document. $e : complément du titre. $f : première mention de responsabilité. $g : mention de responsabilité suivante. $h : numéro de partie – sous-zone de transcription. $i : titre dépendant.
- 245 : titre et mention de responsabilité. $a : titre. $b : autre titre du même auteur. $c : autre titre d'un auteur différent. $d : indication générale du type de document. $e : complément du titre. $f : première mention de responsabilité. $g : mention de responsabilité suivante. $h : numéro de partie – sous-zone de transcription. $i : titre dépendant. $r : reste de la zone. $v : numéro.
- 250 : édition, tirage, état. $a : mention d'édition. $s : mention d'état.
- 260 : adresse bibliographique : édition, diffusion, production ou copie. $a : lieu d'édition, de diffusion ou de copie. $c : nom de la maison d'édition ou de diffusion, ou du copiste. $d : date. $r : adresse entière.
- 261 : adresse bibliographique – documents d'archives. $a : lieu. $d : date.
- 280 : description matérielle du document. $a : indication spécifique du type de document et importance matérielle. $c : autres caractéristiques matérielles. $d : format.
- 300 : note générale. $a : texte.
- 326 : note sur la périodicité. $a : périodicité.
- 600 : vedette matière nom de personne. $3 : numéro de la notice d'autorité nom de personne liée (100 ou 160).
- 606 : vedette matière nom commun. $a : élément d'entrée.
- 607 : vedette matière nom géographique. $a : élément d'entrée.
- 608 : vedette de genre, forme et fonction. $a : élément d'entrée.
- 610 : vedette matière collectivité. $3 : numéro de la notice d'autorité collectivité liée (110 ou 161).
- 700 : vedette secondaire auteur personne physique. $3 : numéro de la notice d'autorité nom de personne liée.
- 701 : vedette secondaire interprète personne physique. $3 : numéro de la notice d'autorité nom de personne liée.
- 710 : vedette secondaire auteur collectivité. $3 : numéro de la notice d'autorité collectivité liée.
- 711 : vedette secondaire interprète collectivité. $3 : numéro de la notice d'autorité collectivité liée.
- 720 : éditeur commercial personne physique. $3 : numéro de la notice d'autorité nom de personne liée.
- 721 : distributeur personne physique. $3 : numéro de la notice d'autorité nom de personne liée.

[1] Une défaillance technique du site de la Bibliothèque nationale de France nous a toutefois empêché d'importer les documents iconographiques de l'année 1794.

- 725 : producteur personne physique. $3 : numéro de la notice d'autorité nom de personne liée.
- 727 : fabricant (ou prestataire) personne physique. $3 : numéro de la notice d'autorité nom de personne liée.
- 730 : éditeur commercial collectivité. $3 : numéro de la notice d'autorité collectivité liée.
- 731 : distributeur collectivité. $3 : numéro de la notice d'autorité collectivité liée.
- 735 : producteur collectivité. $3 : numéro de la notice d'autorité collectivité liée.
- 737 : fabricant (ou prestataire) collectivité. $3 : numéro de la notice d'autorité collectivité liée.
- 749 : titre d'un volume d'une monographie en plusieurs volumes. $a : titre du volume.
- 750 : variante du titre du document. $a : titre.
- 751 : variante du titre de l'œuvre. $a : titre.
- 753 : variante historique du titre. $a : titre.

À propos des zones et sous-zones Intermarc, voir la page « INTERMARC Bibliographique de diffusion » de la Bibiliothèque nationale de France : https://www.bnf.fr/fr/intermarc-bibliographique-de-diffusion (11 août 2023).

Utilisation

Sur le site public, seules les données des œuvres liées à un ou plusieurs autres éléments seront accessibles. Certaines données auront été modifiées, complétées ou supprimées par les collaborateurs, conformément aux besoins du site. Ainsi, une fois chargées dans *Drupal*, les données ne peuvent plus être réimportées et elles ne profitent plus des mises à jour auxquelles procède la Bibliothèque nationale de France. Que les données soient modifiées ou non, leur source est toujours indiquée et un lien hypertexte permet d'accéder à la fiche bibliographique concernée sur le site de la Bibliothèque nationale de France.

Procédure détaillée

1. Exportation des données relatives aux œuvres.
    a. Accéder à https://catalogue.bnf.fr/recherche-avancee.do?pageRech=rav
    b. « Par date de publication » : choisir un intervalle de dates. Nous téléchargons les données par petites tranches chronologiques (moins de 30 000 résultats par tranche).
    c. « Lancer la recherche ».
    d. Sélectionner « Tous les résultats ».
    e. Cliquer sur le bouton « Exporter dans un tableau ».
    f. Renseigner le champ avec la liste de sous-zones suivantes : 022$a;100$3;110$3;243$a;243$b;243$c;243$e;243$f;243$g;243$h;243$i;245$a;245$b;245$c;245$e;245$f;245$g;245$h;245$i;245$r;245$v;250$a;250$s;260$a;260$c;260$d;260$r;261$a;261$d;280$a;280$c;280$d;300$a;326$a;600$3;606$a;607$a;608$a;610$3;700$3;701$3;710$3;711$3;720$3;721$3;725$3;727$3;730$3;731$3;735$3;737$3;749$a;750$a;751$a;753$a;
    g. Sélectionner « Intermarc », « 1 colonne par sous-zone », « Données avec étiquette », « sans sous-notices », « sans données d'exemplaire ».
    h. « Valider ».
    i. Répéter les opérations pour toutes les autres tranches chronologiques.
2. Rassembler l'ensemble des données au sein d'un même fichier CSV.
3. Importer le fichier sur un serveur MariaDB 10.

a. Dans une base de données « bnf », préconfigurer une table « work » contenant une unique colonne « col_2 » de type TEXT(1000000). Exemple de commande SQL :

```
CREATE TABLE work_bnf (
    col_2 TEXT(1000000)
);
```

b. Importer l'ensemble des données. Exemple de commande SQL :

```
LOAD DATA INFILE '/path/to/filename.csv' INTO TABLE work_bnf;
```

c. Devant la colonne « col_2 », ajouter une colonne « col_1 » de type INT.
d. Ajouter une valeur autoincrémentielle à « col_1 » et définir cette colonne comme la clef primaire de la table. Exemple de commande SQL :

```
ALTER TABLE work_bnf
MODIFY col_1 INT NOT NULL AUTO_INCREMENT PRIMARY KEY;
```

4. À présent, nous sommes presque en mesure de créer un fichier CSV importable dans la base de données définitive. Il s'agit de retoucher les données brutes de la Bibliothèque nationale de France, de façon à permettre leur importation dans les champs correspondants de la base de données de notre site, grâce à la fonction d'importation du module Webform de *Drupal*. Pour ce faire, nous appelons les données brutes rassemblées dans table « work_bnf » de la base de données « bnf », nous les modifions en langage PHP, puis nous affichons sur une page web l'ensemble des données modifiées, de manière à pouvoir les copier et les coller dans un nouveau fichier CSV.
    a. En premier lieu, nous avons besoin de construire un tableau associatif relatif aux personnes et aux collectivités. Pour établir des relations entre les œuvres et les personnes (auteurs, éditeurs, etc.), nous devons en effet réunir le numéro de la notice d'autorité de la Bibliothèque nationale de France (per_bnf_record_number) à l'identifiant (NID) de la fiche de personne correspondante dans la base de données du site *Drupal*. Pour créer ce tableau associatif, nous envoyons une requête SQL à la table « node__per_bnf_record_number » de la base « mercure » et nous générons le tableau grâce au PHP. Exemple de code (fichier « csv_bnf_per_id.php ») :

```
<?php
//Array combining for a person their record number in the catalogue of the
Bibliothèque nationale de France and the node ID (NID) of this person.

//Determine the first and last row to be echoed. For $end_row, the value 0 will
echo all the rows.

$start_row = 0;
$end_row = 0;

//Connect to the database.

$mysqli = new MySQLi('localhost','user','password','database');

if(mysqli_connect_errno())
    {
```

```
        printf('Échec de la connexion : base de données momentanément
inaccessible.');
        exit();
    }

//Total number of items.

$result_set_1 = $mysqli->query('SELECT COUNT(*) FROM
node__per_bnf_record_number');
$total = $result_set_1->fetch_row();

if($end_row == 0){
    $end_row = $total[0];
}

//Query the database.

$result_set_2 = $mysqli->query('SELECT entity_id, per_bnf_record_number_value FROM
node__per_bnf_record_number');


//Retrieve and manage data.

$tmpCount = 0;
$total_displayed = 0;
$output = NULL;

if($result_set_2->num_rows > 0){
    while(null !==($rows = $result_set_2->fetch_assoc()))
    {

//      Count the lines and retrieve the two columns from the database.

        $tmpCount ++;
        $entity_id = $rows['entity_id'];
        $per_bnf_record_number_value = $rows['per_bnf_record_number_value'];

//      Combine the fields.

        if(($tmpCount >= $start_row) && ($tmpCount <= $end_row)){
            $output .= $per_bnf_record_number_value.' => '.$entity_id.', <br/>';
            $total_displayed ++;
        }
    }
}

//Echo the number of results.

echo 'Nombre de résultats : '.$total_displayed.'<br/>';
echo 'Affichage des résultats '.$start_row.' à '.$end_row.'<br/><br/>';

//Output.

echo '$per_bnf_record_number_array = array(<br/>'.$output.');';
?>
```

b. Le résultat obtenu dans un navigateur web peut ensuite être copié et collé dans un fichier vierge « per_bnf_record_number_array.php » qui doit être enregistré

dans le même répertoire que « csv_bnf_work.php » (voir plus loin) et qui sera utilisé plus tard. Exemple de code (extrait) :

```php
<?php
$per_bnf_record_number_array = array(
16224293 => 131844,
16257190 => 131845,
16509682 => 131846,
16184814 => 131847,
/* [...] */
17987929 => 333678,
17866995 => 333679,
);
?>
```

c. Autre étape préliminaire, nous devons préimporter dans une taxonomie *Drupal* « Mots clefs » l'ensemble de ce que la Bibliothèque nationale définit comme les « matières » dont traitent les œuvres. Il s'agit d'abord d'isoler ces données pour en faire un fichier CSV. Exemple de code (fichier « csv_bnf_work_subject.php ») :

```php
<?php
//Works: subjects

//Determine the first and last row to be echoed. For $end_row, the value 0 will
echo all the rows.

$start_row = 0;
$end_row = 0;

//Connect to the database.

$mysqli = new MySQLi('localhost','user','password','database');

if(mysqli_connect_errno())
    {
        printf('Échec de la connexion : base de données momentanément
inaccessible.');
        exit();
    }

//Total number of items.

$result_set_1 = $mysqli->query('SELECT COUNT(*) FROM work_bnf');
$total = $result_set_1->fetch_row();

if($end_row == 0){
    $end_row = $total[0];
}

echo $end_row;

//Query the database.

$result_set_2 = $mysqli->query('SELECT * FROM work_bnf WHERE col_1 >=
'.$start_row.' AND col_1 <= '.$end_row);

//Retrieve and manage data.
```

```
$tmpCount = 0;
$total_displayed = 0;
$output = NULL;

if($result_set_2->num_rows > 0){
    while(null !==($rows = $result_set_2->fetch_assoc()))
    {

//      Assign NULL to the variables.

        $wor_subject = $split_606a = $split_607a = NULL;

//      Count the lines and retrieve col_2 from the database.

        $tmpCount ++;
        $data = $rows['col_2'];

//      Add a fake Intermarc number at the end of the line.

        $data = $data.'000$a';

//      Manage semicolons and remove vertical bars.

        $data = str_replace(" ; ", "<span style='color:grey'>[space, semicolon,
space]</span>", $data);
        $data = str_replace("; ", "<span style='color:grey'>[space, semicolon,
space]</span>", $data);
        $data = str_replace(";", "<span style='color:blue'>[tab]</span>", $data);
        $data = str_replace("<span style='color:grey'>[space, semicolon,
space]</span>", " ; ", $data);
        $data = str_replace(" | ", "<span style='color:blue'>[tab]</span>",
$data);

//      Manage and escape double-quotes.

        $data = str_replace('"', '""', $data);
        $data = str_replace('« ', '""', $data);
        $data = str_replace('«', '""', $data);
        $data = str_replace(' »', '""', $data);
        $data = str_replace('»', '""', $data);

//      Split the lines.

        $split = explode("<span style='color:blue'>[tab]</span>", $data);

//      Create field wor_subject from 606$a and 607$a.

        $split_606a = explode('606$a', $data);
        $split_606a = explode ($split_606a[0].'606$a', $data);
        $split_606a = str_replace('606$a', '[comma]', $split_606a[1]);
        $split_606a = preg_split('/(\d){3}\$./', $split_606a);

        $split_606a = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_606a[0]);

        $split_606a = explode('[comma]', $split_606a);

        if($split_606a[0] == $split_606a[1]){
```

```
            $split_606a = $split_606a[0];
        }
        else{
            $split_606a = implode('[comma]', $split_606a);
        }

        $split_606a = str_replace(',', '%2C', $split_606a); //Escape original
commas in URI.
        $split_606a = str_replace('[comma]', ',', $split_606a); //Multiple values
are separated by commas.

        if(substr_count($split_606a, '"') === 1){
            $split_606a = str_replace('"', '', $split_606a);
        }

        if(stripos($split_606a, '[') !== false && stripos($split_606a, ']') ===
false){
            $split_606a = str_replace('[', '', $split_606a);
        }

        if(stripos($split_606a, ']') !== false && stripos($split_606a, '[') ===
false){
            $split_606a = str_replace(']', '', $split_606a);
        }

        if(stripos($split_606a, '(') !== false && stripos($split_606a, ')') ===
false){
            $split_606a = str_replace('(', '', $split_606a);
        }

        if(stripos($split_606a, ')') !== false && stripos($split_606a, '(') ===
false){
            $split_606a = str_replace(')', '', $split_606a);
        }

        $split_607a = explode('607$a', $data);
        $split_607a = explode ($split_607a[0].'607$a', $data);
        $split_607a = str_replace('607$a', '[comma]', $split_607a[1]);
        $split_607a = preg_split('/(\d){3}\$./', $split_607a);

        $split_607a = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_607a[0]);

        $split_607a = explode('[comma]', $split_607a);

        if($split_607a[0] == $split_607a[1]){
            $split_607a = $split_607a[0];
        }
        else{
            $split_607a = implode('[comma]', $split_607a);
        }

        $split_607a = str_replace(',', '%2C', $split_607a); //Escape original
commas in URI.
        $split_607a = str_replace('[comma]', ',', $split_607a); //Multiple values
are separated by commas.

        if(substr_count($split_607a, '"') === 1){
            $split_607a = str_replace('"', '', $split_607a);
```

```
        }

        if(stripos($split_607a, '[') !== false && stripos($split_607a, ']') ===
false){
            $split_607a = str_replace('[', '', $split_607a);
        }

        if(stripos($split_607a, ']') !== false && stripos($split_607a, '[') ===
false){
            $split_607a = str_replace(']', '', $split_607a);
        }

        if(stripos($split_607a, '(') !== false && stripos($split_607a, ')') ===
false){
            $split_607a = str_replace('(', '', $split_607a);
        }

        if(stripos($split_607a, ')') !== false && stripos($split_607a, '(') ===
false){
            $split_607a = str_replace(')', '', $split_607a);
        }

        if($split_606a != '' && $split_607a != ''){
            $wor_subject = $split_606a.','.$split_607a;
        }
        else{
            $wor_subject = $split_606a.$split_607a;
        }

        $wor_subject = str_replace(',', '<br/>', $wor_subject);
        $wor_subject = str_replace('%2C', ',', $wor_subject);

        if($wor_subject ==! NULL){
            $wor_subject = $wor_subject.'<br/>';
        }

//      Combine the fields in CSV format.

        $output .= $wor_subject;
        $total_displayed ++;

    }
}

//Remove duplicates and sort by alphabetical order.

$output = explode('<br/>', $output);
$output = array_unique($output);
asort($output);
$output = implode('<br/>', $output);

//Echo the number of results.

echo 'Nombre de résultats : '.$total_displayed.'<br/>';
echo 'Affichage des résultats '.$start_row.' à '.$end_row.'<br/><br/>';

//Output in CSV format.

echo 'wor_subject<br/>';
```

```
echo $output;

?>
```

d. Ces données peuvent ensuite être importées dans la taxonomie « Mots clefs » de *Drupal* au moyen d'un outil d'importation comme le module Taxonomy Import.
e. Il s'agit maintenant d'associer à chaque terme son identifiant (TID), tel qu'il apparaît dans la base de données *Drupal*. Nous avons choisi de le faire manuellement sur Microsoft Excel, de façon à procéder à de petites corrections dans la liste des termes. (Ces corrections sont utiles, en particulier, pour rétablir les doublons que forment des homonymes comme « Saint-Petersbourg » et « Saint-Pétersbourg », doublons que *Drupal* fusionne.)
f. Ensuite, les couples formés par le mot ou la phrase désignant la « matière » et l'identifiant (TID) doivent être présentés sous la forme d'un tableau associatif, dans un fichier PHP vierge « wor_subject_array.php ». Ce fichier doit être enregistré dans le même répertoire que « csv_bnf_work.php » (voir ci-après) et sera utilisé à l'étape suivante. Exemple de code (extrait) :

```
<?php
$wor_subject_array = array(
'Aalst' => 8234,
'Abattoirs' => 311,
'Abbayes' => 312,
'Abbés' => 313,
/* [...] */
'Zurich' => 8232,
'Zwolle' => 8233,
    );
?>
```

b. Nous pouvons maintenant modifier en langage PHP les données bibliographiques de la Bibliothèque nationale de France, de façon à les importer dans notre base de données *Drupal*. Exemple de code (fichier « csv_bnf_work.php ») :

```
<?php
//Works

//Determine the first and last row to be echoed. For $end_row, the value 0 will
echo all the rows.

$start_row = 0;
$end_row = 0;

//Connect to the database.

$mysqli = new MySQLi('localhost','user','password','database');

if(mysqli_connect_errno())
    {
        printf('Échec de la connexion : base de données momentanément
inaccessible.');
        exit();
    }
```

```
//Total number of items.

$result_set_1 = $mysqli->query('SELECT COUNT(*) FROM work_bnf');
$total = $result_set_1->fetch_row();

if($end_row == 0){
    $end_row = $total[0];
}

//Query the database.

$result_set_2 = $mysqli->query('SELECT * FROM work_bnf WHERE col_1 >=
'.$start_row.' AND col_1 <= '.$end_row);

//Retrieve and manage data.

$tmpCount = 0;
$total_displayed = 0;
$output = NULL;

if($result_set_2->num_rows > 0){
    while(null !==($rows = $result_set_2->fetch_assoc()))
    {

//      Assign NULL to the variables.

        $all_remark = $all_remark_verified = $all_validity = $matches =
$wor_address = $wor_assumed_end_date = $wor_assumed_date =
$wor_assumed_end_date_verified = $wor_bnf_record = $wor_bnf_record_number =
$wor_data_origin_bnf = $wor_date = $wor_date_check = $wor_date_raw =
$wor_description = $wor_document_type = $wor_edition = $wor_end_date =
$wor_end_date_raw = $wor_refers_to_per = $wor_has_author_per =
$wor_has_other_creator_per = $wor_has_publisher_per = $wor_issn = $wor_periodicity
= $wor_record_type = $all_remark_bnf = $wor_responsibility = $wor_subject =
$wor_title = $wor_title_243 = $wor_title_245 = $wor_variant_of_title =
$wor_volume_titles = $years = $split = $split_years = $split_022a = $split_100_3 =
$split_110_3 = $split_226a = $split_243a = $split_243b = $split_243c = $split_243e
= $split_243f = $split_243g = $split_243h = $split_243h = $split_243i =
$split_245a = $split_245b = $split_245c = $split_245e = $split_245f = $split_245g
= $split_245h = $split_245i = $split_245r = $split_245r = $split_245v =
$split_250a = $split_250s = $split_260c = $split_260d = $split_261d = $split_280a
= $split_280c = $split_280d = $split_300a = $split_600_3 = $split_606a =
$split_607a = $split_610_3 = $split_700_3 = $split_701_3 = $split_710_3 =
$split_711_3 = $split_720_3 = $split_721_3 = $split_725_3 = $split_727_3 =
$split_730_3 = $split_731_3 = $split_735_3 = $split_737_3 = $split_749a =
$split_750a = $split_751a = $split_753a = NULL;

//      Count the lines and retrieve col_2 from the database.

        $tmpCount ++;
        $data = $rows['col_2'];

//      Add a fake Intermarc number at the end of the line.

        $data = $data.'000$a';

//      Manage semicolons and remove vertical bars.
```

```
        $data = str_replace(" ; ", "<span style='color:grey'>[space, semicolon,
space]</span>", $data);
        $data = str_replace("; ", "<span style='color:grey'>[space, semicolon,
space]</span>", $data);
        $data = str_replace(";", "<span style='color:blue'>[tab]</span>", $data);
        $data = str_replace("<span style='color:grey'>[space, semicolon,
space]</span>", " ; ", $data);
        $data = str_replace(" | ", "<span style='color:blue'>[tab]</span>",
$data);

//      Manage and escape double-quotes.

        $data = str_replace('"', '""', $data);
        $data = str_replace('« ', '""', $data);
        $data = str_replace('«', '""', $data);
        $data = str_replace(' »', '""', $data);
        $data = str_replace('»', '""', $data);

//      Split the lines.

        $split = explode("<span style='color:blue'>[tab]</span>", $data);

//      Create field wor_record_type.

        if(stripos($split[2], 'Monographie') !== false){
            $wor_record_type = '263'; //Value matching "Monographie" in taxonomy
"Type de notice".
        }
        elseif(stripos($split[2], 'Périodique') !== false){
            $wor_record_type = '264'; //Value matching "Périodique" in taxonomy
"Type de notice".
        }
        elseif(stripos($split[2], 'Recueil') !== false){
            $wor_record_type = '265'; //Value matching "Recueil" in taxonomy "Type
de notice".
        }
        else{
            $wor_record_type = '266'; //Value matching "Autre ensemble ou
collection" in taxonomy "Type de notice".
        }

//      Create field wor_document_type

        if(stripos($split[3], 'Texte imprimé') !== false){
            $wor_document_type = '274'; //Value matching "Texte imprimé" in
taxonomy "Type de document".
        }
        elseif(stripos($split[3], 'Texte manuscrit') !== false){
            $wor_document_type = '275'; //Value matching "Texte manuscrit ou
document d'archive" in taxonomy "Type de document".
        }
        elseif(stripos($split[3], 'Document cartographique') !== false){
            $wor_document_type = '276'; //Value matching "Document cartographique"
in taxonomy "Type de document".
        }
        elseif(stripos($split[3], 'Document d\'archive') !== false){
            $wor_document_type = '275'; //Value matching "Texte manuscrit ou
document d'archive" in taxonomy "Type de document".
        }
```

```
        elseif(stripos($split[3], 'Image') !== false){
            $wor_document_type = '277'; //Value matching "Image" in taxonomy "Type
de document".
        }
        elseif(stripos($split[3], 'Musique') !== false){
            $wor_document_type = '278'; //Value matching "Musique" in taxonomy
"Type de document".
        }
        elseif(stripos($split[3], 'Objet') !== false){
            $wor_document_type = '279'; //Value matching "Objet" in taxonomy "Type
de document".
        }
        elseif(stripos($split[3], 'Spectacle') !== false){
            $wor_document_type = '280'; //Value matching "Spectacle" in taxonomy
"Type de document".
        }
        elseif(stripos($split[3], 'Manuscrit') !== false){
            $wor_document_type = '275'; //Value matching "Texte manuscrit ou
document d'archive" in taxonomy "Type de document".
        }

//      Create field wor_title from 243$a, 243$b, 243$c, 243$e, 243$h, 243$i,
245$a, 245$b, 245$c, 245$e, 245$h, 245$i, 245$r, 245$v.

//      Title given by the librarian (243).

        $split_243a = explode('243$a', $data);
        $split_243a = explode ($split_243a[0].'243$a', $data);
        $split_243a = str_replace('243$a', '[slash]', $split_243a[1]);
        $split_243a = preg_split('/(\d){3}\$./', $split_243a);

        $split_243a = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_243a[0]);

        $split_243a = explode('[slash]', $split_243a);

        if($split_243a[0] == $split_243a[1]){
            $split_243a = $split_243a[0];
        }
        else{
            $split_243a = implode('[slash]', $split_243a);
        }

        $split_243a = str_replace('[slash]', ' / ', $split_243a);

        $split_243b = explode('243$b', $data);
        $split_243b = explode ($split_243b[0].'243$b', $data);
        $split_243b = str_replace('243$b', '[slash]', $split_243b[1]);
        $split_243b = preg_split('/(\d){3}\$./', $split_243b);

        $split_243b = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_243b[0]);

        $split_243b = explode('[slash]', $split_243b);

        if($split_243b[0] == $split_243b[1]){
            $split_243b = $split_243b[0];
        }
        else{
```

```
            $split_243b = implode('[slash]', $split_243b);
        }

        $split_243b = str_replace('[slash]', ' / ', $split_243b);

        $split_243c = explode('243$c', $data);
        $split_243c = explode ($split_243c[0].'243$c', $data);
        $split_243c = str_replace('243$c', '[slash]', $split_243c[1]);
        $split_243c = preg_split('/(\d){3}\$./', $split_243c);

        $split_243c = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_243c[0]);

        $split_243c = explode('[slash]', $split_243c);

        if($split_243c[0] == $split_243c[1]){
            $split_243c = $split_243c[0];
        }
        else{
            $split_243c = implode('[slash]', $split_243c);
        }

        $split_243c = str_replace('[slash]', ' / ', $split_243c);

        if($split_243a != '' && $split_243b != '' && $split_243c != ''){
            $wor_title_243 = $split_243a.' / '.$split_243b.' / '.$split_243c;
        }
        elseif($split_243a != '' && $split_243b != ''){
            $wor_title_243 = $Split_243a.' / '.$split_243b;
        }
        elseif($split_243b != '' && $split_243c != ''){
            $wor_title_243 = $Split_243b.' / '.$split_243c;
        }
        elseif($split_243a != '' && $split_243c != ''){
            $wor_title_243 = $Split_243a.' / '.$split_243c;
        }
        elseif($split_243b != ''){
            $wor_title_243 = $split_243b;
        }
        elseif($split_243c != ''){
            $wor_title_243 = $Split_243c;
        }
        else{
            $wor_title_243 = $split_243a;
        }

        if(substr_count($wor_title_243, '"') === 1){
            $wor_title_243 = str_replace('"', '', $wor_title_243);
        }

        if(stripos($wor_title_243, '[') !== false && stripos($wor_title_243, ']')
=== false){
            $wor_title_243 = str_replace('[', '', $wor_title_243);
        }

        if(stripos($wor_title_243, ']') !== false && stripos($wor_title_243, '[')
=== false){
            $wor_title_243 = str_replace(']', '', $wor_title_243);
        }
```

```
        if(stripos($wor_title_243, '(') !== false && stripos($wor_title_243, ')')
=== false){
            $wor_title_243 = str_replace('(', '', $wor_title_243);
        }

        if(stripos($wor_title_243, ')') !== false && stripos($wor_title_243, '(')
=== false){
            $wor_title_243 = str_replace(')', '', $wor_title_243);
        }

        $split_243e = explode('243$e', $data);
        $split_243e = explode ($split_243e[0].'243$e', $data);
        $split_243e = str_replace('243$e', '[slash]', $split_243e[1]);
        $split_243e = preg_split('/(\d){3}\$./', $split_243e);

        $split_243e = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_243e[0]);

        $split_243e = explode('[slash]', $split_243e);

        if($split_243e[0] == $split_243e[1]){
            $split_243e = $split_243e[0];
        }
        else{
            $split_243e = implode('[slash]', $split_243e);
        }

        $split_243e = str_replace('[slash]', ' / ', $split_243e);

        if($wor_title_243 != '' && $split_243e != ''){
            $wor_title_243 = $wor_title_243.' : '.$split_243e;
        }
        elseif($split_243e != ''){
            $wor_title_243 = $split_243e;
        }

        if(substr_count($wor_title_243, '"') === 1){
            $wor_title_243 = str_replace('"', '', $wor_title_243);
        }

        if(stripos($wor_title_243, '[') !== false && stripos($wor_title_243, ']')
=== false){
            $wor_title_243 = str_replace('[', '', $wor_title_243);
        }

        if(stripos($wor_title_243, ']') !== false && stripos($wor_title_243, '[')
=== false){
            $wor_title_243 = str_replace(']', '', $wor_title_243);
        }

        if(stripos($wor_title_243, '(') !== false && stripos($wor_title_243, ')')
=== false){
            $wor_title_243 = str_replace('(', '', $wor_title_243);
        }

        if(stripos($wor_title_243, ')') !== false && stripos($wor_title_243, '(')
=== false){
            $wor_title_243 = str_replace(')', '', $wor_title_243);
```

```
        }

        $split_243h = explode('243$h', $data);
        $split_243h = explode ($split_243h[0].'243$h', $data);
        $split_243h = str_replace('243$h', '[comma]', $split_243h[1]);
        $split_243h = preg_split('/(\d){3}\$./', $split_243h);

        $split_243h = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_243h[0]);

        $split_243h = explode('[comma]', $split_243h);

        if($split_243h[0] == $split_243h[1]){
            $split_243h = $split_243h[0];
        }
        else{
            $split_243h = implode('[comma]', $split_243h);
        }

        $split_243h = str_replace('[comma]', ', ', $split_243h);

        $split_243i = explode('243$i', $data);
        $split_243i = explode ($split_243i[0].'243$i', $data);
        $split_243i = str_replace('243$i', '[comma]', $split_243i[1]);
        $split_243i = preg_split('/(\d){3}\$./', $split_243i);

        $split_243i = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_243i[0]);

        $split_243i = explode('[comma]', $split_243i);

        if($split_243i[0] == $split_243i[1]){
            $split_243i = $split_243i[0];
        }
        else{
            $split_243i = implode('[comma]', $split_243i);
        }

        $split_243i = str_replace('[comma]', ', ', $split_243i);

        if($split_243h != '' && $split_243i != ''){
            $wor_title_243 = $wor_title_243.'. '.$split_243h.', '.$split_243i;
        }
        elseif($split_243h != ''){
            $wor_title_243 = $wor_title_243.'. '.$split_243h;
        }
        elseif($split_243i != ''){
            $wor_title_243 = $wor_title_243.'. '.$split_243i;
        }

        if(substr_count($wor_title_243, '"') === 1){
            $wor_title_243 = str_replace('"', '', $wor_title_243);
        }

        if(stripos($wor_title_243, '[') !== false && stripos($wor_title_243, ']')
=== false){
            $wor_title_243 = str_replace('[', '', $wor_title_243);
        }
```

```
        if(stripos($wor_title_243, ']') !== false && stripos($wor_title_243, '[')
=== false){
            $wor_title_243 = str_replace(']', '', $wor_title_243);
        }

        if(stripos($wor_title_243, '(') !== false && stripos($wor_title_243, ')')
=== false){
            $wor_title_243 = str_replace('(', '', $wor_title_243);
        }

        if(stripos($wor_title_243, ')') !== false && stripos($wor_title_243, '(')
=== false){
            $wor_title_243 = str_replace(')', '', $wor_title_243);
        }

//      Printed title (245).

        $split_245v = explode('245$v', $data);
        $split_245v = explode ($split_245v[0].'245$v', $data);
        $split_245v = str_replace('245$v', '[comma]', $split_245v[1]);
        $split_245v = preg_split('/(\d){3}\$./', $split_245v);

        $split_245v = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_245v[0]);

        $split_245v = explode('[comma]', $split_245v);

        if($split_245v[0] == $split_245v[1]){
            $split_245v = $split_245v[0];
        }
        else{
            $split_245v = implode('[comma]', $split_245v);
        }

        $split_245v = str_replace('[comma]', ', ', $split_245v);

        if(substr_count($split_245v, '"') === 1){
            $split_245v = str_replace('"', '', $split_245v);
        }

        if(stripos($split_245v, '[') !== false && stripos($split_245v, ']') ===
false){
            $split_245v = str_replace('[', '', $split_245v);
        }

        if(stripos($split_245v, ']') !== false && stripos($split_245v, '[') ===
false){
            $split_245v = str_replace(']', '', $split_245v);
        }

        if(stripos($split_245v, '(') !== false && stripos($split_245v, ')') ===
false){
            $split_245v = str_replace('(', '', $split_245v);
        }

        if(stripos($split_245v, ')') !== false && stripos($split_245v, '(') ===
false){
            $split_245v = str_replace(')', '', $split_245v);
        }
```

```
        $split_245a = explode('245$a', $data);
        $split_245a = explode ($split_245a[0].'245$a', $data);
        $split_245a = str_replace('245$a', '[slash]', $split_245a[1]);
        $split_245a = preg_split('/(\d){3}\$./', $split_245a);

        $split_245a = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_245a[0]);

        $split_245a = explode('[slash]', $split_245a);

        if($split_245a[0] == $split_245a[1]){
            $split_245a = $split_245a[0];
        }
        else{
            $split_245a = implode('[slash]', $split_245a);
        }

        $split_245a = str_replace('[slash]', ' / ', $split_245a);

        if(substr_count($split_245a, '"') === 1){
            $split_245a = str_replace('"', '', $split_245a);
        }

        if(stripos($split_245a, '[') !== false && stripos($split_245a, ']') ===
false){
            $split_245a = str_replace('[', '', $split_245a);
        }

        if(stripos($split_245a, ']') !== false && stripos($split_245a, '[') ===
false){
            $split_245a = str_replace(']', '', $split_245a);
        }

        if(stripos($split_245a, '(') !== false && stripos($split_245a, ')') ===
false){
            $split_245a = str_replace('(', '', $split_245a);
        }

        if(stripos($split_245a, ')') !== false && stripos($split_245a, '(') ===
false){
            $split_245a = str_replace(')', '', $split_245a);
        }

        $split_245b = explode('245$b', $data);
        $split_245b = explode ($split_245b[0].'245$b', $data);
        $split_245b = str_replace('245$b', '[slash]', $split_245b[1]);
        $split_245b = preg_split('/(\d){3}\$./', $split_245b);

        $split_245b = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_245b[0]);

        $split_245b = explode('[slash]', $split_245b);

        if($split_245b[0] == $split_245b[1]){
            $split_245b = $split_245b[0];
        }
        else{
            $split_245b = implode('[slash]', $split_245b);
```

```
        }

        $split_245b = str_replace('[slash]', ' / ', $split_245b);

        if(substr_count($split_245b, '"') === 1){
            $split_245b = str_replace('"', '', $split_245b);
        }

        if(stripos($split_245b, '[') !== false && stripos($split_245b, ']') ===
false){
            $split_245b = str_replace('[', '', $split_245b);
        }

        if(stripos($split_245b, ']') !== false && stripos($split_245b, '[') ===
false){
            $split_245b = str_replace(']', '', $split_245b);
        }

        if(stripos($split_245b, '(') !== false && stripos($split_245b, ')') ===
false){
            $split_245b = str_replace('(', '', $split_245b);
        }

        if(stripos($split_245b, ')') !== false && stripos($split_245b, '(') ===
false){
            $split_245b = str_replace(')', '', $split_245b);
        }

        $split_245c = explode('245$c', $data);
        $split_245c = explode ($split_245c[0].'245$c', $data);
        $split_245c = str_replace('245$c', '[slash]', $split_245c[1]);
        $split_245c = preg_split('/(\d){3}\$./', $split_245c);

        $split_245c = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_245c[0]);

        $split_245c = explode('[slash]', $split_245c);

        if($split_245c[0] == $split_245c[1]){
            $split_245c = $split_245c[0];
        }
        else{
            $split_245c = implode('[slash]', $split_245c);
        }

        $split_245c = str_replace('[slash]', ' / ', $split_245c);

        if(substr_count($split_245c, '"') === 1){
            $split_245c = str_replace('"', '', $split_245c);
        }

        if(stripos($split_245c, '[') !== false && stripos($split_245c, ']') ===
false){
            $split_245c = str_replace('[', '', $split_245c);
        }

        if(stripos($split_245c, ']') !== false && stripos($split_245c, '[') ===
false){
            $split_245c = str_replace(']', '', $split_245c);
```

```
        }

        if(stripos($split_245c, '(') !== false && stripos($split_245c, ')') ===
false){
            $split_245c = str_replace('(', '', $split_245c);
        }

        if(stripos($split_245c, ')') !== false && stripos($split_245c, '(') ===
false){
            $split_245c = str_replace(')', '', $split_245c);
        }

        if($split_245a != '' && $split_245b != '' && $split_245c != ''){
            $wor_title_245 = $split_245a.' / '.$split_245b.' / '.$split_245c;
        }
        elseif($split_245a != '' && $split_245b != ''){
            $wor_title_245 = $Split_245a.' / '.$split_245b;
        }
        elseif($split_245b != '' && $split_245c != ''){
            $wor_title_245 = $Split_245b.' / '.$split_245c;
        }
        elseif($split_245a != '' && $split_245c != ''){
            $wor_title_245 = $Split_245a.' / '.$split_245c;
        }
        elseif($split_245b != ''){
            $wor_title_245 = $split_245b;
        }
        elseif($split_245c != ''){
            $wor_title_245 = $Split_245c;
        }
        else{
            $wor_title_245 = $split_245a;
        }

        if($split_245v != ''){
            $wor_title_245 = $split_245v.'. '.$wor_title_245;
        }

        $split_245r = explode('245$r', $data);
        $split_245r = explode ($split_245r[0].'245$r', $data);
        $split_245r = str_replace('245$r', '[dash]', $split_245r[1]);
        $split_245r = preg_split('/(\d){3}\$./', $split_245r);

        $split_245r = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_245r[0]);

        $split_245r = explode('[dash]', $split_245r);

        if($split_245r[0] == $split_245r[1]){
            $split_245r = $split_245r[0];
        }
        else{
            $split_245r = implode('[dash]', $split_245r);
        }

        $split_245r = str_replace('[dash]', ' - ', $split_245r);

        if(substr_count($split_245r, '"') === 1){
            $split_245r = str_replace('"', '', $split_245r);
```

```
        }

        if(stripos($split_245r, '[') !== false && stripos($split_245r, ']') ===
false){
            $split_245r = str_replace('[', '', $split_245r);
        }

        if(stripos($split_245r, ']') !== false && stripos($split_245r, '[') ===
false){
            $split_245r = str_replace(']', '', $split_245r);
        }

        if(stripos($split_245r, '(') !== false && stripos($split_245r, ')') ===
false){
            $split_245r = str_replace('(', '', $split_245r);
        }

        if(stripos($split_245r, ')') !== false && stripos($split_245r, '(') ===
false){
            $split_245r = str_replace(')', '', $split_245r);
        }

        if($split_245r != ''){
            $wor_title_245 = $wor_title_245.$split_245r;
        }

        $split_245e = explode('245$e', $data);
        $split_245e = explode ($split_245e[0].'245$e', $data);
        $split_245e = str_replace('245$e', '[slash]', $split_245e[1]);
        $split_245e = preg_split('/(\d){3}\$./', $split_245e);

        $split_245e = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_245e[0]);

        $split_245e = explode('[slash]', $split_245e);

        if($split_245e[0] == $split_245e[1]){
            $split_245e = $split_245e[0];
        }
        else{
            $split_245e = implode('[slash]', $split_245e);
        }

        $split_245e = str_replace('[slash]', ' / ', $split_245e);

        if(substr_count($split_245e, '"') === 1){
            $split_245e = str_replace('"', '', $split_245e);
        }

        if(stripos($split_245e, '[') !== false && stripos($split_245e, ']') ===
false){
            $split_245e = str_replace('[', '', $split_245e);
        }

        if(stripos($split_245e, ']') !== false && stripos($split_245e, '[') ===
false){
            $split_245e = str_replace(']', '', $split_245e);
        }
```

```
        if(stripos($split_245e, '(') !== false && stripos($split_245e, ')') ===
false){
            $split_245e = str_replace('(', '', $split_245e);
        }

        if(stripos($split_245e, ')') !== false && stripos($split_245e, '(') ===
false){
            $split_245e = str_replace(')', '', $split_245e);
        }

        if($wor_title_245 != '' && $split_245e != ''){
            $wor_title_245 = $wor_title_245.' : '.$split_245e;
        }
        elseif($split_245e != ''){
            $wor_title_245 = $split_245e;
        }

        $split_245h = explode('245$h', $data);
        $split_245h = explode ($split_245h[0].'245$h', $data);
        $split_245h = str_replace('245$h', '[comma]', $split_245h[1]);
        $split_245h = preg_split('/(\d){3}\$./', $split_245h);

        $split_245h = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_245h[0]);

        $split_245h = explode('[comma]', $split_245h);

        if($split_245h[0] == $split_245h[1]){
            $split_245h = $split_245h[0];
        }
        else{
            $split_245h = implode('[comma]', $split_245h);
        }

        $split_245h = str_replace('[comma]', ', ', $split_245h);

        if(substr_count($split_245h, '"') === 1){
            $split_245h = str_replace('"', '', $split_245h);
        }

        if(stripos($split_245h, '[') !== false && stripos($split_245h, ']') ===
false){
            $split_245h = str_replace('[', '', $split_245h);
        }

        if(stripos($split_245h, ']') !== false && stripos($split_245h, '[') ===
false){
            $split_245h = str_replace(']', '', $split_245h);
        }

        if(stripos($split_245h, '(') !== false && stripos($split_245h, ')') ===
false){
            $split_245h = str_replace('(', '', $split_245h);
        }

        if(stripos($split_245h, ')') !== false && stripos($split_245h, '(') ===
false){
            $split_245h = str_replace(')', '', $split_245h);
        }
```

```
        $split_245i = explode('245$i', $data);
        $split_245i = explode ($split_245i[0].'245$i', $data);
        $split_245i = str_replace('245$i', '[comma]', $split_245i[1]);
        $split_245i = preg_split('/(\d){3}\$./', $split_245i);

        $split_245i = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_245i[0]);

        $split_245i = explode('[comma]', $split_245i);

        if($split_245i[0] == $split_245i[1]){
            $split_245i = $split_245i[0];
        }
        else{
            $split_245i = implode('[comma]', $split_245i);
        }

        $split_245i = str_replace('[comma]', ', ', $split_245i);

        if(substr_count($split_245i, '"') === 1){
            $split_245i = str_replace('"', '', $split_245i);
        }

        if(stripos($split_245i, '[') !== false && stripos($split_245i, ']') ===
false){
            $split_245i = str_replace('[', '', $split_245i);
        }

        if(stripos($split_245i, ']') !== false && stripos($split_245i, '[') ===
false){
            $split_245i = str_replace(']', '', $split_245i);
        }

        if(stripos($split_245i, '(') !== false && stripos($split_245i, ')') ===
false){
            $split_245i = str_replace('(', '', $split_245i);
        }

        if(stripos($split_245i, ')') !== false && stripos($split_245i, '(') ===
false){
            $split_245i = str_replace(')', '', $split_245i);
        }

        if($split_245h != '' && $split_245i != ''){
            $wor_title_245 = $wor_title_245.'. '.$split_245h.', '.$split_245i;
        }
        elseif($split_245h != ''){
            $wor_title_245 = $wor_title_245.'. '.$split_245h;
        }
        elseif($split_245i != ''){
            $wor_title_245 = $wor_title_245.'. '.$split_245i;
        }

        if($wor_title_243 != '' && $wor_title_245 != ''){
        $wor_title = $wor_title_245.' '.$wor_title_243;
        }
        elseif($wor_title_243 != ''){
            $wor_title = $wor_title_243;
```

```
        }
        elseif($wor_title_245 != ''){
            $wor_title = $wor_title_245;
        }
        else{
            $wor_title = '[Sans titre]';
        }

        $wor_title = preg_replace('/^"{3,}/', '""', $wor_title);
        $wor_title = preg_replace('/"{3,}$/', '""', $wor_title);
        $wor_title = str_replace('""""', '""', $wor_title);
        $wor_title = str_replace('""""', '""', $wor_title);

        if(stripos($wor_title, '"') !== false || stripos($wor_title, ',') !==
false){
            $wor_title = '"'.$wor_title.'"';
        }

//      Create field wor_volume_titles from 749$a.

        $split_749a = explode('749$a', $data);
        $split_749a = explode ($split_749a[0].'749$a', $data);
        $split_749a = str_replace('749$a', '[comma]', $split_749a[1]);
        $split_749a = preg_split('/(\d){3}\$./', $split_749a);

        $split_749a = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_749a[0]);
        $split_749a = str_replace(',', '%2C', $split_749a); //Escape original
commas in URI.
        $split_749a = str_replace('[comma]', ',', $split_749a); //Multiple values
are separated by commas.

        if(substr_count($split_749a, '""') === 1){
            $split_749a = str_replace('""', '', $split_749a);
        }

        if(stripos($split_749a, '[') !== false && stripos($split_749a, ']') ===
false){
            $split_749a = str_replace('[', '', $split_749a);
        }

        if(stripos($split_749a, ']') !== false && stripos($split_749a, '[') ===
false){
            $split_749a = str_replace(']', '', $split_749a);
        }

        if(stripos($split_749a, '(') !== false && stripos($split_749a, ')') ===
false){
            $split_749a = str_replace('(', '', $split_749a);
        }

        if(stripos($split_749a, ')') !== false && stripos($split_749a, '(') ===
false){
            $split_749a = str_replace(')', '', $split_749a);
        }

        $split_749a = preg_replace('/^"{3,}/', '""', $split_749a);
        $split_749a = preg_replace('/"{3,}$/', '""', $split_749a);
        $split_749a = str_replace('""""', '""', $split_749a);
```

```
        $split_749a = str_replace('"""', '"', $split_749a);

        if(stripos($split_749a, ',') !== false){
            $wor_volume_titles = '"'.$split_749a.'"';
        }
        else{
            $wor_volume_titles = $split_749a;

            if(stripos($wor_volume_titles, '"') !== false) {
                $wor_volume_titles = '"'.$wor_volume_titles.'"';
            }

        }

//      Create field wor_variant_of_title from 750$a, 751$a, 753$a.

        $split_750a = explode('750$a', $data);
        $split_750a = explode ($split_750a[0].'750$a', $data);
        $split_750a = str_replace('750$a', '[comma]', $split_750a[1]);
        $split_750a = preg_split('/(\d){3}\$./', $split_750a);

        $split_750a = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_750a[0]);
        $split_750a = str_replace(',', '%2C', $split_750a); //Escape original
commas in URI.
        $split_750a = str_replace('[comma]', ',', $split_750a); //Multiple values
are separated by commas.

        if(substr_count($split_750a, '"') === 1){
            $split_750a = str_replace('"', '', $split_750a);
        }

        if(stripos($split_750a, '[') !== false && stripos($split_750a, ']') ===
false){
            $split_750a = str_replace('[', '', $split_750a);
        }

        if(stripos($split_750a, ']') !== false && stripos($split_750a, '[') ===
false){
            $split_750a = str_replace(']', '', $split_750a);
        }

        if(stripos($split_750a, '(') !== false && stripos($split_750a, ')') ===
false){
            $split_750a = str_replace('(', '', $split_750a);
        }

        if(stripos($split_750a, ')') !== false && stripos($split_750a, '(') ===
false){
            $split_750a = str_replace(')', '', $split_750a);
        }

        $split_751a = explode('751$a', $data);
        $split_751a = explode ($split_751a[0].'751$a', $data);
        $split_751a = str_replace('751$a', '[comma]', $split_751a[1]);
        $split_751a = preg_split('/(\d){3}\$./', $split_751a);

        $split_751a = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_751a[0]);
```

```
        $split_751a = str_replace(',', '%2C', $split_751a); //Escape original
commas in URI.
        $split_751a = str_replace('[comma]', ',', $split_751a); //Multiple values
are separated by commas.

        if(substr_count($split_751a, '"') === 1){
            $split_751a = str_replace('"', '', $split_751a);
        }

        if(stripos($split_751a, '[') !== false && stripos($split_751a, ']') ===
false){
            $split_751a = str_replace('[', '', $split_751a);
        }

        if(stripos($split_751a, ']') !== false && stripos($split_751a, '[') ===
false){
            $split_751a = str_replace(']', '', $split_751a);
        }

        if(stripos($split_751a, '(') !== false && stripos($split_751a, ')') ===
false){
            $split_751a = str_replace('(', '', $split_751a);
        }

        if(stripos($split_751a, ')') !== false && stripos($split_751a, '(') ===
false){
            $split_751a = str_replace(')', '', $split_751a);
        }

        $split_753a = explode('753$a', $data);
        $split_753a = explode ($split_753a[0].'753$a', $data);
        $split_753a = str_replace('753$a', '[comma]', $split_753a[1]);
        $split_753a = preg_split('/(\d){3}\$./', $split_753a);

        $split_753a = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_753a[0]);
        $split_753a = str_replace(',', '%2C', $split_753a); //Escape original
commas in URI.
        $split_753a = str_replace('[comma]', ',', $split_753a); //Multiple values
are separated by commas.

        if(substr_count($split_753a, '"') === 1){
            $split_753a = str_replace('"', '', $split_753a);
        }

        if(stripos($split_753a, '[') !== false && stripos($split_753a, ']') ===
false){
            $split_753a = str_replace('[', '', $split_753a);
        }

        if(stripos($split_753a, ']') !== false && stripos($split_753a, '[') ===
false){
            $split_753a = str_replace(']', '', $split_753a);
        }

        if(stripos($split_753a, '(') !== false && stripos($split_753a, ')') ===
false){
            $split_753a = str_replace('(', '', $split_753a);
        }
```

```
        if(stripos($split_753a, ')') !== false && stripos($split_753a, '(') ===
false){
            $split_753a = str_replace(')', '', $split_753a);
        }

        $split_750a = preg_replace('/^"{3,}/', '""', $split_750a);
        $split_750a = preg_replace('/"{3,}$/', '""', $split_750a);
        $split_750a = str_replace('"""""', '""', $split_750a);
        $split_750a = str_replace('"""""', '""', $split_750a);

        $split_751a = preg_replace('/^"{3,}/', '""', $split_751a);
        $split_751a = preg_replace('/"{3,}$/', '""', $split_751a);
        $split_751a = str_replace('"""""', '""', $split_751a);
        $split_751a = str_replace('"""""', '""', $split_751a);

        $split_753a = preg_replace('/^"{3,}/', '""', $split_753a);
        $split_753a = preg_replace('/"{3,}$/', '""', $split_753a);
        $split_753a = str_replace('"""""', '""', $split_753a);
        $split_753a = str_replace('"""""', '""', $split_753a);

        if($split_750a != '' && $split_751a != '' && $split_753a != ''){
            $wor_variant_of_title =
'"'.$split_750a.','.$split_751a.','.$split_753a.'"';
        }
        elseif($split_750a != '' && $split_751a != ''){
            $wor_variant_of_title = '"'.$split_750a.','.$split_751a.'"';
        }
        elseif($split_750a != '' && $split_753a != ''){
            $wor_variant_of_title = '"'.$split_750a.','.$split_753a.'"';
        }
        elseif($split_751a != '' && $split_753a != ''){
            $wor_variant_of_title = '"'.$split_751a.','.$split_753a.'"';
        }
        else{
            $wor_variant_of_title = $split_750a.$split_751a.$split_753a;

            if(stripos($wor_variant_of_title, '"') !== false ||
stripos($wor_variant_of_title, ',') !== false) {
                $wor_variant_of_title = '"'.$wor_variant_of_title.'"';
            }

        }

//      Create field wor_responsibility from 243$f, 243$g, 245$f, 245$g.

        $split_243f = explode('243$f', $data);
        $split_243f = explode ($split_243f[0].'243$f', $data);
        $split_243f = str_replace('243$f', '[slash]', $split_243f[1]);
        $split_243f = preg_split('/(\d){3}\$./', $split_243f);

        $split_243f = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_243f[0]);

        $split_243f = explode('[slash]', $split_243f);

        if($split_243f[0] == $split_243f[1]){
            $split_243f = $split_243f[0];
        }
```

```
        else{
            $split_243f = implode('[slash]', $split_243f);
        }

        $split_243f = str_replace('[slash]', ' / ', $split_243f);

        if(substr_count($split_243f, '"') === 1){
            $split_243f = str_replace('"', '', $split_243f);
        }

        if(stripos($split_243f, '[') !== false && stripos($split_243f, ']') ===
false){
            $split_243f = str_replace('[', '', $split_243f);
        }

        if(stripos($split_243f, ']') !== false && stripos($split_243f, '[') ===
false){
            $split_243f = str_replace(']', '', $split_243f);
        }

        if(stripos($split_243f, '(') !== false && stripos($split_243f, ')') ===
false){
            $split_243f = str_replace('(', '', $split_243f);
        }

        if(stripos($split_243f, ')') !== false && stripos($split_243f, '(') ===
false){
            $split_243f = str_replace(')', '', $split_243f);
        }

        $split_243g = explode('243$g', $data);
        $split_243g = explode ($split_243g[0].'243$g', $data);
        $split_243g = str_replace('243$g', '[slash]', $split_243g[1]);
        $split_243g = preg_split('/(\d){3}\$./', $split_243g);

        $split_243g = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_243g[0]);

        $split_243g = explode('[slash]', $split_243g);

        if($split_243g[0] == $split_243g[1]){
            $split_243g = $split_243g[0];
        }
        else{
            $split_243g = implode('[slash]', $split_243g);
        }

        $split_243g = str_replace('[slash]', ' / ', $split_243g);

        if(substr_count($split_243g, '"') === 1){
            $split_243g = str_replace('"', '', $split_243g);
        }

        if(stripos($split_243g, '[') !== false && stripos($split_243g, ']') ===
false){
            $split_243g = str_replace('[', '', $split_243g);
        }
```

```
        if(stripos($split_243g, ']') !== false && stripos($split_243g, '[') ===
false){
            $split_243g = str_replace(']', '', $split_243g);
        }

        if(stripos($split_243g, '(') !== false && stripos($split_243g, ')') ===
false){
            $split_243g = str_replace('(', '', $split_243g);
        }

        if(stripos($split_243g, ')') !== false && stripos($split_243g, '(') ===
false){
            $split_243g = str_replace(')', '', $split_243g);
        }

        $split_245f = explode('245$f', $data);
        $split_245f = explode ($split_245f[0].'245$f', $data);
        $split_245f = str_replace('245$f', '[slash]', $split_245f[1]);
        $split_245f = preg_split('/(\d){3}\$./', $split_245f);

        $split_245f = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_245f[0]);

        $split_245f = explode('[slash]', $split_245f);

        if($split_245f[0] == $split_245f[1]){
            $split_245f = $split_245f[0];
        }
        else{
            $split_245f = implode('[slash]', $split_245f);
        }

        $split_245f = str_replace('[slash]', ' / ', $split_245f);

        if(substr_count($split_245f, '"') === 1){
            $split_245f = str_replace('"', '', $split_245f);
        }

        if(stripos($split_245f, '[') !== false && stripos($split_245f, ']') ===
false){
            $split_245f = str_replace('[', '', $split_245f);
        }

        if(stripos($split_245f, ']') !== false && stripos($split_245f, '[') ===
false){
            $split_245f = str_replace(']', '', $split_245f);
        }

        if(stripos($split_245f, '(') !== false && stripos($split_245f, ')') ===
false){
            $split_245f = str_replace('(', '', $split_245f);
        }

        if(stripos($split_245f, ')') !== false && stripos($split_245f, '(') ===
false){
            $split_245f = str_replace(')', '', $split_245f);
        }

        $split_245g = explode('245$g', $data);
```

```
        $split_245g = explode ($split_245g[0].'245$g', $data);
        $split_245g = str_replace('245$g', '[slash]', $split_245g[1]);
        $split_245g = preg_split('/(\d){3}\$./', $split_245g);

        $split_245g = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_245g[0]);

        $split_245g = explode('[slash]', $split_245g);

        if($split_245g[0] == $split_245g[1]){
            $split_245g = $split_245g[0];
        }
        else{
            $split_245g = implode('[slash]', $split_245g);
        }

        $split_245g = str_replace('[slash]', ' / ', $split_245g);

        if(substr_count($split_245g, '"') === 1){
            $split_245g = str_replace('"', '', $split_245g);
        }

        if(stripos($split_245g, '[') !== false && stripos($split_245g, ']') ===
false){
            $split_245g = str_replace('[', '', $split_245g);
        }

        if(stripos($split_245g, ']') !== false && stripos($split_245g, '[') ===
false){
            $split_245g = str_replace(']', '', $split_245g);
        }

        if(stripos($split_245g, '(') !== false && stripos($split_245g, ')') ===
false){
            $split_245g = str_replace('(', '', $split_245g);
        }

        if(stripos($split_245g, ')') !== false && stripos($split_245g, '(') ===
false){
            $split_245g = str_replace(')', '', $split_245g);
        }

        if($split_243f != '' && $split_243g != '' && $split_245f != '' &&
$split_245g != ''){
            $wor_responsibility = $split_243f.' / '.$split_243g.' /
'.$split_245f.' / '.$split_245g;
        }
        elseif($split_243f != '' && $split_243g != '' && $split_245f != ''){
            $wor_responsibility = $split_243f.' / '.$split_243g.' / '.$split_245f;
        }
        elseif($split_243f != '' && $split_243g != '' && $split_245g != ''){
            $wor_responsibility = $split_243f.' / '.$split_243g.' / '.$split_245g;
        }
        elseif($split_243f != '' && $split_245f != '' && $split_245g != ''){
            $wor_responsibility = $split_243f.' / '.$split_245f.' / '.$split_245g;
        }
        elseif($split_243g != '' && $split_245f != '' && $split_245g != ''){
            $wor_responsibility = $split_243g.' / '.$split_245f.' / '.$split_245g;
        }
```

```
        elseif($split_243f != '' && $split_243g != ''){
            $wor_responsibility = $split_243f.' / '.$split_243g;
        }
        elseif($split_243f != '' && $split_245f != ''){
            $wor_responsibility = $split_243f.' / '.$split_245f;
        }
        elseif($split_243f != '' && $split_245g != ''){
            $wor_responsibility = $split_243f.' / '.$split_245g;
        }
        elseif($split_243g != '' && $split_245f != ''){
            $wor_responsibility = $split_243g.' / '.$split_245f;
        }
        elseif($split_243g != '' && $split_245g != ''){
            $wor_responsibility = $split_243g.' / '.$split_245g;
        }
        elseif($split_245f != '' && $split_245g != ''){
            $wor_responsibility = $split_245f.' / '.$split_245g;
        }
        else{
            $wor_responsibility = $split_243f.$split_243g.$split_245f.$split_245g;
        }

        $wor_responsibility = ucfirst($wor_responsibility);

        $wor_responsibility = preg_replace('/^"{3,}/', '"', $wor_responsibility);
        $wor_responsibility = preg_replace('/"{3,}$/', '"', $wor_responsibility);
        $wor_responsibility = str_replace('"""', '"', $wor_responsibility);
        $wor_responsibility = str_replace('"""', '"', $wor_responsibility);

        if(stripos($wor_responsibility, '"') !== false ||
stripos($wor_responsibility, ',') !== false){
            $wor_responsibility = '"'.$wor_responsibility.'"';
        }

//      Create field wor_edition from 250$a and 250$s.

        $split_250a = explode('250$a', $data);
        $split_250a = explode ($split_250a[0].'250$a', $data);
        $split_250a = str_replace('250$a', '[slash]', $split_250a[1]);
        $split_250a = preg_split('/(\d){3}\$./', $split_250a);

        $split_250a = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_250a[0]);

        $split_250a = explode('[slash]', $split_250a);

        if($split_250a[0] == $split_250a[1]){
            $split_250a = $split_250a[0];
        }
        else{
            $split_250a = implode('[slash]', $split_250a);
        }

        $split_250a = str_replace('[slash]', ' / ', $split_250a);

        if(substr_count($split_250a, '"') === 1){
            $split_250a = str_replace('"', '', $split_250a);
        }
```

```
        if(stripos($split_250a, '[') !== false && stripos($split_250a, ']') ===
false){
            $split_250a = str_replace('[', '', $split_250a);
        }

        if(stripos($split_250a, ']') !== false && stripos($split_250a, '[') ===
false){
            $split_250a = str_replace(']', '', $split_250a);
        }

        if(stripos($split_250a, '(') !== false && stripos($split_250a, ')') ===
false){
            $split_250a = str_replace('(', '', $split_250a);
        }

        if(stripos($split_250a, ')') !== false && stripos($split_250a, '(') ===
false){
            $split_250a = str_replace(')', '', $split_250a);
        }

        $split_250s = explode('250$s', $data);
        $split_250s = explode ($split_250s[0].'250$s', $data);
        $split_250s = str_replace('250$s', '[slash]', $split_250s[1]);
        $split_250s = preg_split('/(\d){3}\$./', $split_250s);

        $split_250s = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_250s[0]);

        $split_250s = explode('[slash]', $split_250s);

        if($split_250s[0] == $split_250s[1]){
            $split_250s = $split_250s[0];
        }
        else{
            $split_250s = implode('[slash]', $split_250s);
        }

        $split_250s = str_replace('[slash]', ' / ', $split_250s);

        if(substr_count($split_250s, '"') === 1){
            $split_250s = str_replace('"', '', $split_250s);
        }

        if(stripos($split_250s, '[') !== false && stripos($split_250s, ']') ===
false){
            $split_250s = str_replace('[', '', $split_250s);
        }

        if(stripos($split_250s, ']') !== false && stripos($split_250s, '[') ===
false){
            $split_250s = str_replace(']', '', $split_250s);
        }

        if(stripos($split_250s, '(') !== false && stripos($split_250s, ')') ===
false){
            $split_250s = str_replace('(', '', $split_250s);
        }
```

```
        if(stripos($split_250s, ')') !== false && stripos($split_250s, '(') ===
false){
            $split_250s = str_replace(')', '', $split_250s);
        }

        if($split_250a != '' && $split_250s != ''){
            $wor_edition = $split_250a.' / '.$split_250s;
        }
        else{
            $wor_edition = $split_250a.$split_250s;
        }

        $wor_edition = preg_replace('/^"{3,}/', '"', $wor_edition);
        $wor_edition = preg_replace('/"{3,}$/', '"', $wor_edition);
        $wor_edition = str_replace('"""', '"', $wor_edition);
        $wor_edition = str_replace('"""', '"', $wor_edition);

        if(stripos($wor_edition, '"') !== false || stripos($wor_edition, ',') !==
false){
            $wor_edition = '"'.$wor_edition.'"';
        }

//      Create wor_address from 260$a, 260$c, 260$r, 261$a.

        $split_260a = explode('260$a', $data);
        $split_260a = explode ($split_260a[0].'260$a', $data);
        $split_260a = str_replace('260$a', '[comma]', $split_260a[1]);
        $split_260a = preg_split('/(\d){3}\$./', $split_260a);

        $split_260a = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_260a[0]);

        $split_260a = explode('[comma]', $split_260a);

        if($split_260a[0] == $split_260a[1]){
            $split_260a = $split_260a[0];
        }
        else{
            $split_260a = implode('[comma]', $split_260a);
        }

        $split_260a = str_replace('[comma]', ', ', $split_260a);

        if(substr_count($split_260a, '"') === 1){
            $split_260a = str_replace('"', '', $split_260a);
        }

        if(stripos($split_260a, '[') !== false && stripos($split_260a, ']') ===
false){
            $split_260a = str_replace('[', '', $split_260a);
        }

        if(stripos($split_260a, ']') !== false && stripos($split_260a, '[') ===
false){
            $split_260a = str_replace(']', '', $split_260a);
        }

        if(stripos($split_260a, '(') !== false && stripos($split_260a, ')') ===
false){
```

```
            $split_260a = str_replace('(', '', $split_260a);
        }

        if(stripos($split_260a, ')') !== false && stripos($split_260a, '(') ===
false){
            $split_260a = str_replace(')', '', $split_260a);
        }

        $split_260c = explode('260$c', $data);
        $split_260c = explode ($split_260c[0].'260$c', $data);
        $split_260c = str_replace('260$c', '[comma]', $split_260c[1]);
        $split_260c = preg_split('/(\d){3}\$./', $split_260c);

        $split_260c = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_260c[0]);

        $split_260c = explode('[comma]', $split_260c);

        if($split_260c[0] == $split_260c[1]){
            $split_260c = $split_260c[0];
        }
        else{
            $split_260c = implode('[comma]', $split_260c);
        }

        $split_260c = str_replace('[comma]', ', ', $split_260c);

        if(substr_count($split_260c, '"') === 1){
            $split_260c = str_replace('"', '', $split_260c);
        }

        if(stripos($split_260c, '[') !== false && stripos($split_260c, ']') ===
false){
            $split_260c = str_replace('[', '', $split_260c);
        }

        if(stripos($split_260c, ']') !== false && stripos($split_260c, '[') ===
false){
            $split_260c = str_replace(']', '', $split_260c);
        }

        if(stripos($split_260c, '(') !== false && stripos($split_260c, ')') ===
false){
            $split_260c = str_replace('(', '', $split_260c);
        }

        if(stripos($split_260c, ')') !== false && stripos($split_260c, '(') ===
false){
            $split_260c = str_replace(')', '', $split_260c);
        }

        $split_260r = explode('260$r', $data);
        $split_260r = explode ($split_260r[0].'260$r', $data);
        $split_260r = str_replace('260$r', '[comma]', $split_260r[1]);
        $split_260r = preg_split('/(\d){3}\$./', $split_260r);

        $split_260r = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_260r[0]);
```

```
        $split_260r = explode('[comma]', $split_260r);

        if($split_260r[0] == $split_260r[1]){
            $split_260r = $split_260r[0];
        }
        else{
            $split_260r = implode('[comma]', $split_260r);
        }

        $split_260r = str_replace('[comma]', ', ', $split_260r);

        if(substr_count($split_260r, '"') === 1){
            $split_260r = str_replace('"', '', $split_260r);
        }

        if(stripos($split_260r, '[') !== false && stripos($split_260r, ']') ===
false){
            $split_260r = str_replace('[', '', $split_260r);
        }

        if(stripos($split_260r, ']') !== false && stripos($split_260r, '[') ===
false){
            $split_260r = str_replace(']', '', $split_260r);
        }

        if(stripos($split_260r, '(') !== false && stripos($split_260r, ')') ===
false){
            $split_260r = str_replace('(', '', $split_260r);
        }

        if(stripos($split_260r, ')') !== false && stripos($split_260r, '(') ===
false){
            $split_260r = str_replace(')', '', $split_260r);
        }

        if($split_260a != '' && $split_260c != '' && $split_260r != ''){
            $wor_address_260 = $split_260a.', '.$split_260c.' / '.$split_260r;
        }
        elseif($split_260a != '' && $split_260c != ''){
            $wor_address_260 = $split_260a.', '.$split_260c;
        }
        elseif($split_260a != '' && $split_260r != ''){
            $wor_address_260 = $split_260a.' / '.$split_260r;
        }
        elseif($split_260c != '' && $split_260r != ''){
            $wor_address_260 = $split_260c.' / '.$split_260r;
        }
        else{
            $wor_address_260 = $split_260a.$split_260c.$split_260r;
        }

        $split_261a = explode('261$a', $data);
        $split_261a = explode ($split_261a[0].'261$a', $data);
        $split_261a = str_replace('261$a', '[comma]', $split_261a[1]);
        $split_261a = preg_split('/(\d){3}\$./', $split_261a);

        $split_261a = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_261a[0]);
```

```
        $split_261a = explode('[comma]', $split_261a);

        if($split_261a[0] == $split_261a[1]){
            $split_261a = $split_261a[0];
        }
        else{
            $split_261a = implode('[comma]', $split_261a);
        }

        $wor_address_261 = str_replace('[comma]', ', ', $split_261a);

        if(substr_count($wor_address_261, '"') === 1){
            $wor_address_261 = str_replace('"', '', $wor_address_261);
        }

        if(stripos($wor_address_261, '[') !== false && stripos($wor_address_261,
']') === false){
            $wor_address_261 = str_replace('[', '', $wor_address_261);
        }

        if(stripos($wor_address_261, ']') !== false && stripos($wor_address_261,
'[') === false){
            $wor_address_261 = str_replace(']', '', $wor_address_261);
        }

        if(stripos($wor_address_261, '(') !== false && stripos($wor_address_261,
')') === false){
            $wor_address_261 = str_replace('(', '', $wor_address_261);
        }

        if(stripos($wor_address_261, ')') !== false && stripos($wor_address_261,
'(') === false){
            $wor_address_261 = str_replace(')', '', $wor_address_261);
        }

        if($wor_address_260 != '' && $wor_address_261 != ''){
            $wor_address = $wor_address_260.' / '.$wor_address_261;
        }
        else{
            $wor_address = $wor_address_260.$wor_address_261;
        }

        $wor_address = str_replace('.,)', '.)', $wor_address);
        $wor_address = str_replace('.,]', '.]', $wor_address);
        $wor_address = str_replace(',,', ',', $wor_address);
        $wor_address = ucfirst($wor_address);

        $wor_address = preg_replace('/^"{3,}/', '"', $wor_address);
        $wor_address = preg_replace('/"{3,}$/', '"', $wor_address);
        $wor_address = str_replace('"""', '"', $wor_address);
        $wor_address = str_replace('"""', '"', $wor_address);

        if(stripos($wor_address, '"') !== false || stripos($wor_address, ',') !==
false){
            $wor_address = '"'.$wor_address.'"';
        }

//      Create field wor_date, wor_assumed_date, wor_end_date and
wor_assumed_end_date from 260$d and 261$d.
```

```
//      First, manage both first date (publication date or publication start date)
and second date (publication end date).

        $split_260d = explode('260$d', $data);
        $split_260d = preg_split('/(\d){3}\$./', $split_260d[1]);

        $split_261d = explode('261$d', $data);
        $split_261d = preg_split('/(\d){3}\$./', $split_261d[1]);

        $split_260d = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_260d[0]);

        $split_261d = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_261d[0]);

        if($split_260d != ''){
            $years = $split_260d;
        }
        else{
            $years = $split_261d;
        }

        $all_remark_bnf = $years; //Prepare the field all_remark_bnf for later.

        //Special cases

        $years = str_ireplace('[avant 1753, ca 1750]', '1750?', $years);

        if(stripos($years, '(...)') !== false || stripos($years, 'In-') !==
false){
            $years = '';
        }

        if(preg_match('/\[\d{2,3}\.\.?\]/', $years)){
            $years = '';
        }

        if(preg_match('/\(\d{2,3}\.\.?\)/', $years)){
            $years = '';
        }

        if(preg_match('/\d{5,}/', $years)){
            $years = '';
        }

        if(preg_match('/\b175\[\?\]/', $years)){
            $years = '';
        }

        //Usual cases

        $years = str_ireplace('[', '', $years);
        $years = str_ireplace(']', '', $years);
        $years = str_ireplace('(', '', $years);
        $years = str_ireplace(')', '', $years);
        $years = str_ireplace('ave', '', $years);
        $years = str_ireplace('rav', '', $years);
        $years = preg_replace('/av(.*?)c/i', '[avJC]', $years);
        $years = preg_replace('/ap(.*?)c/i', '[apJC]', $years);
```

```
        $years = str_ireplace('[avJC].', '[avJC]', $years);
        $years = str_ireplace('[apJC].', '[apJC]', $years);
        $years = str_ireplace(' ca ', '?', $years);
        $years = str_ireplace(' vers', '?', $years);
        $years = str_ireplace(' v.', '?', $years);
        $years = str_ireplace(' c. ', '?', $years);
        $years = str_ireplace('??', '..', $years);
        $years = str_replace('–', '-', $years);

        if(str_starts_with($years, 'ca') !== false || str_starts_with($years,
'Ca') !== false){
            $years = str_ireplace('ca', '?', $years);
            $wor_assumed_end_date_verified = 'verified'; //Prepare
wor_assumed_end_date;
        }

        if(str_starts_with($years, 'Vers') !== false || str_starts_with($years,
'vers') !== false){
            $years = str_ireplace('vers', '?', $years);
            $wor_assumed_end_date_verified = 'verified'; //Prepare
wor_assumed_end_date;
        }

        if(str_starts_with($years, 'V.') !== false || str_starts_with($years,
'v.') !== false){
            $years = str_ireplace('v.', '?', $years);
            $wor_assumed_end_date_verified = 'verified'; //Prepare
wor_assumed_end_date;
        }

        if(str_starts_with($years, 'c. ') !== false || str_starts_with($years, 'C.
') !== false){
            $years = str_ireplace('c. ', '?', $years);
            $wor_assumed_end_date_verified = 'verified'; //Prepare
wor_assumed_end_date;
        }

        if(stripos($years, '.-c.') === false){
            $years = str_ireplace('-c.', '-?', $years);
        }

        if(stripos($years, 'entre ') !== false){
            $years = str_ireplace(' et ', '?-', $years);
            $wor_assumed_end_date_verified = 'verified'; //Prepare
wor_assumed_end_date;
        }

        if(str_starts_with($years, '-') || str_starts_with($years, '?-') ||
str_starts_with($years, '? -')){
            $years = str_replace('--', '-[avJC]', $years);
            $years = preg_replace('/^\-/', '[avJC]', $years);
        }

//      Now create wor_date.

        $split_years = explode('-', $years);

        if(preg_match('/\d+/', $split_years[0])){
            $wor_date_raw = $split_years[0];
```

```
        }
        elseif(preg_match('/\d+/', $split_years[1])){
            $wor_date_raw = $split_years[1];
        }
        elseif(preg_match('/\d+/', $split_years[2])){
            $wor_date_raw = $split_years[2];
        }
        else{
            $wor_date_raw = $years;
        }

        if(stripos($wor_date_raw, 'avant') !== false || stripos($wor_date_raw,
'après') !== false || stripos($wor_date_raw, 'A partir de ') !== false){
            $wor_date_raw = '';
        }

        if(preg_match('/\b\d{2}\.\.|\b\d{3}\.\b/', $wor_date_raw)){
            $wor_date_raw = '';
        }

        if(stripos($wor_date_raw, 'siècle') !== false || stripos($wor_date_raw, '
s.') !== false){
            if(stripos($wor_date_raw, 'fin du ') !== false ||
stripos($wor_date_raw, 'début du ') !== false || stripos($wor_date_raw, 'milieu du
') !== false || stripos($wor_date_raw, 'moitié du ') !== false ||
stripos($wor_date_raw, '1re') !== false || stripos($wor_date_raw, '1er') !== false
|| stripos($wor_date_raw, '2de') !== false || stripos($wor_date_raw, '2me') !==
false || stripos($wor_date_raw, '2nd') !== false || stripos($wor_date_raw, '2e')
!== false || stripos($wor_date_raw, '3e') !== false || stripos($wor_date_raw,
'4e') !== false || stripos($wor_date_raw, 'Ire') !== false ||
stripos($wor_date_raw, 'IIde') !== false || stripos($wor_date_raw, 'IIme') !==
false || stripos($wor_date_raw, 'IInd') !== false){
                $wor_date_raw = '';
            }
        }

        if(str_starts_with($wor_date_raw, 'av.') || str_starts_with($wor_date_raw,
'Av.') || str_starts_with($wor_date_raw, 'ap.') || str_starts_with($wor_date_raw,
'Ap.')){
            $wor_date_raw = '';
        }

        if(preg_match('/\d{4}/', $wor_date_raw)){
        }
        elseif(stripos($wor_date_raw, 's. d.') !== false){
            $wor_date_raw = '';
        }

        if(preg_match('/\d{1,2}
(janvier|février|mars|avril|mai|juin|juillet|août|septembre|octobre|novembre|décem
bre|aoust|7bre|8bre|9bre|10embre|VIIbre|VIIIbre|IXbre|Xbre|january|february|march|
april|may|june|july|august|september|october|november|december|januar|februar|märz
|april|juni|august|september|oktober|november|dezember|gennaio|febbraio|marzo|apri
le|maggio|giugno|luglio|agosto|settembre|ottobre|novembre|dicembre|nian)/i',
$wor_date_raw, $matches)){
            $wor_date = str_replace($matches[0], '', $wor_date_raw);
            $wor_date = str_replace($matches[1], '', $wor_date);
            $wor_date = str_replace($matches[2], '', $wor_date);
            $wor_date = str_replace($matches[3], '', $wor_date);
```

```
            $wor_date = str_replace($matches[4], '', $wor_date);
            $wor_date = str_replace($matches[5], '', $wor_date);

        }
        else{
            $wor_date = $wor_date_raw;
        }

        if(preg_match('/\d{4}\.\./', $wor_date)){
        }
        else{
            $wor_date = preg_replace('/\d{2}\.\./', '', $wor_date);
        }

        if(preg_match('/\d{4}/', $wor_date, $matches)){
            $wor_date = $matches[0];
        }
        elseif(preg_match('/\d{3}/', $wor_date, $matches)){
            $wor_date = $matches[0];
        }
        elseif(preg_match('/\d{2}/', $wor_date, $matches)){
            $wor_date = $matches[0];
        }

        $wor_date =
preg_replace('/\b1er\b|\b2e\b|\b3e\b|\b4e\b|\b5e\b|\b6e\b|\b7e\b|\b8e\b|\b9e\b|\b1
0e\b|\b11e\b|\b12e\b|\b13e\b|\b14e\b|\b15e\b|\b16e\b|\b17e\b|\b18e\b|\b19e\b|\b20e
\b|\b21e\b|\b22e\b|\b23e\b|\b24e\b|\b25e\b|\b26e\b|\b27e\b|\b28e\b|\b29e\b|\b30e\b
|\b431e\b/', '', $wor_date);

        if(preg_match('/\d{5}/', $wor_date)){
            $wor_date = '';
        }
        elseif(preg_match('/\d{4}\./', $wor_date)){
        }
        elseif(preg_match('/\d{3}\./', $wor_date)){
            $wor_date = '';
        }

        if(preg_match('/\d{2}\.\./', $wor_date)){
            $wor_date = '';
        }

        if(preg_match('/\d+/', $wor_date, $matches)){
            $wor_date = $matches[0];
        }
        else{
            $wor_date = '';
        }

        $wor_date = ltrim($wor_date, '0');

        if(stripos($years, '[avJC]') !== false && $wor_date != ''){
            $wor_date = '-'.$wor_date;
        }

        $wor_date = (int)$wor_date;

        if($wor_date < -2699 || $wor_date > 1820 || $wor_date == 0){
```

```
            $wor_date = '';
        }

        $wor_date = (string)$wor_date;

//      Now create wor_assumed_date

        if(stripos($wor_date_raw, '?') !== false && $wor_date != ''){
            $wor_assumed_date = '1';
        }

//      Now create wor_end_date (and add a few extra conditions for wor_date).

        if($wor_date == ''){
        }
        elseif(preg_match('/\d+/', $split_years[0])){
            if(preg_match('/\d+/', $split_years[1])){
                $wor_end_date_raw = $split_years[1];
            }
            elseif(preg_match('/\d+/', $split_years[2])){
                $wor_end_date_raw = $split_years[2];
            }
        }
        elseif(preg_match('/\d+/', $split_years[1])){
            if(preg_match('/\d+/', $split_years[2])){
                $wor_end_date_raw = $split_years[2];
            }
            elseif(preg_match('/\d+/', $split_years[3])){
                $wor_end_date_raw = $split_years[3];
            }
        }
        elseif(preg_match('/\d+/', $split_years[2])){
            if(preg_match('/\d+/', $split_years[3])){
                $wor_end_date_raw = $split_years[3];
            }
            elseif(preg_match('/\d+/', $split_years[4])){
                $wor_end_date_raw = $split_years[4];
            }
        }
        else{
            $wor_end_date_raw = '';
        }

        if(stripos($wor_end_date_raw, 'avant') !== false ||
stripos($wor_end_date_raw, 'après') !== false || stripos($wor_end_date_raw, 'A
partir de ') !== false){
            $wor_end_date_raw = '';
        }

        if(preg_match('/\b\d{2}\.\.|\b\d{3}\.\b/', $wor_end_date_raw)){
            $wor_end_date_raw = '';
        }

        if(stripos($wor_end_date_raw, 'siècle') !== false ||
stripos($wor_end_date_raw, ' s.') !== false){
            if(stripos($wor_end_date_raw, 'fin du ') !== false ||
stripos($wor_end_date_raw, 'début du ') !== false || stripos($wor_end_date_raw,
'milieu du ') !== false || stripos($wor_end_date_raw, 'moitié du ') !== false ||
stripos($wor_end_date_raw, '1re') !== false || stripos($wor_end_date_raw, '1er')
```

```
!== false || stripos($wor_end_date_raw, '2de') !== false ||
stripos($wor_end_date_raw, '2me') !== false || stripos($wor_end_date_raw, '2nd')
!== false || stripos($wor_end_date_raw, '2e') !== false ||
stripos($wor_end_date_raw, '3e') !== false || stripos($wor_end_date_raw, '4e') !==
false || stripos($wor_end_date_raw, 'Ire') !== false || stripos($wor_end_date_raw,
'IIde') !== false || stripos($wor_end_date_raw, 'IIme') !== false ||
stripos($wor_end_date_raw, 'IInd') !== false){
                $wor_end_date_raw = '';
            }
        }

        if(str_starts_with($wor_end_date_raw, 'av.') ||
str_starts_with($wor_end_date_raw, 'Av.') || str_starts_with($wor_end_date_raw,
'ap.') || str_starts_with($wor_end_date_raw, 'Ap.')){
            $wor_end_date_raw = '';
        }

        if(preg_match('/\d{4}/', $wor_end_date_raw)){
        }
        elseif(stripos($wor_end_date_raw, 's. d.') !== false){
            $wor_end_date_raw = '';
        }

        if(preg_match('/\d{1,2}
(janvier|février|mars|avril|mai|juin|juillet|août|septembre|octobre|novembre|décem
bre|aoust|7bre|8bre|9bre|10embre|VIIbre|VIIIbre|IXbre|Xbre|january|february|march|
april|may|june|july|august|september|october|november|december|januar|februar|märz
|april|juni|august|september|oktober|november|dezember|gennaio|febbraio|marzo|apri
le|maggio|giugno|luglio|agosto|settembre|ottobre|novembre|dicembre|nian)/i',
$wor_end_date_raw, $matches)){
            $wor_end_date = str_replace($matches[0], '', $wor_end_date_raw);
            $wor_end_date = str_replace($matches[1], '', $wor_end_date);
            $wor_end_date = str_replace($matches[2], '', $wor_end_date);
            $wor_end_date = str_replace($matches[3], '', $wor_end_date);
            $wor_end_date = str_replace($matches[4], '', $wor_end_date);
            $wor_end_date = str_replace($matches[5], '', $wor_end_date);

        }
        else{
            $wor_end_date = $wor_end_date_raw;
        }

        if(preg_match('/\d{4}\.\./', $wor_end_date)){
        }
        else{
            $wor_end_date = preg_replace('/\d{2}\.\./', '', $wor_end_date);
        }

        if(preg_match('/\d{4}/', $wor_end_date, $matches)){
            $wor_end_date = $matches[0];
        }
        elseif(preg_match('/\d{3}/', $wor_end_date, $matches)){
            $wor_end_date = $matches[0];
        }
        elseif(preg_match('/\d{2}/', $wor_end_date, $matches)){
            $wor_end_date = $matches[0];
        }
```

```
        $wor_end_date =
preg_replace('/\b1er\b|\b2e\b|\b3e\b|\b4e\b|\b5e\b|\b6e\b|\b7e\b|\b8e\b|\b9e\b|\b1
0e\b|\b11e\b|\b12e\b|\b13e\b|\b14e\b|\b15e\b|\b16e\b|\b17e\b|\b18e\b|\b19e\b|\b20e
\b|\b21e\b|\b22e\b|\b23e\b|\b24e\b|\b25e\b|\b26e\b|\b27e\b|\b28e\b|\b29e\b|\b30e\b
|\b431e\b/', '', $wor_end_date);

        if(preg_match('/\d{5}/', $wor_end_date)){
            $wor_end_date = '';
        }
        elseif(preg_match('/\d{4}\./', $wor_end_date)){
        }
        elseif(preg_match('/\d{3}\./', $wor_end_date)){
            $wor_end_date = '';
        }

        if(preg_match('/\d{2}\.\./', $wor_end_date)){
            $wor_end_date = '';
        }

        if(preg_match('/\d+/', $wor_end_date, $matches)){
            $wor_end_date = $matches[0];
        }
        else{
            $wor_end_date = '';
        }

        $wor_end_date = ltrim($wor_end_date, '0');

        if(stripos($years, '[avJC]') !== false && $wor_end_date != ''){
            $wor_end_date = '-'.$wor_end_date;
        }

        $wor_end_date = (int)$wor_end_date;
        $wor_date_check = (int)$wor_date;

        if($wor_end_date < -2699 || $wor_end_date > 2023 || $wor_end_date == 0 ||
$wor_date_check > $wor_end_date || $wor_date == ''){
            $wor_end_date = '';
        }

        if($wor_date != '' && $wor_end_date != '' && $wor_date_check >
$wor_end_date){
            $wor_date = '';
        }

        $wor_end_date = (string)$wor_end_date;

        if(preg_match_all('/\d{4}/', $years, $matches)){
            if(isset($matches[0][4])){
                $wor_date = '';
                $wor_end_date = '';
            }
        }

//      Now create wor_assumed_end_date

        if(stripos($wor_end_date_raw, '?') !== false && $wor_end_date != ''){
            $wor_assumed_end_date = '1';
        }
```

```
        if($wor_assumed_end_date_verified == 'verified' && $wor_end_date != ''){
            $wor_assumed_end_date = '1';
        }

//      Create field wor_description from 280$a, 280$c, 280$d.

        $split_280a = explode('280$a', $data);
        $split_280a = explode ($split_280a[0].'280$a', $data);
        $split_280a = str_replace('280$a', '[semicolon]', $split_280a[1]);
        $split_280a = preg_split('/(\d){3}\$./', $split_280a);

        $split_280a = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_280a[0]);

        $split_280a = explode('[semicolon]', $split_280a);

        if($split_280a[0] == $split_280a[1]){
            $split_280a = $split_280a[0];
        }
        else{
            $split_280a = implode('[semicolon]', $split_280a);
        }

        $split_280a = str_replace('[semicolon]', ' ; ', $split_280a);

        if(substr_count($split_280a, '"') === 1){
            $split_280a = str_replace('"', '', $split_280a);
        }

        if(stripos($split_280a, '[') !== false && stripos($split_280a, ']') ===
false){
            $split_280a = str_replace('[', '', $split_280a);
        }

        if(stripos($split_280a, ']') !== false && stripos($split_280a, '[') ===
false){
            $split_280a = str_replace(']', '', $split_280a);
        }

        if(stripos($split_280a, '(') !== false && stripos($split_280a, ')') ===
false){
            $split_280a = str_replace('(', '', $split_280a);
        }

        if(stripos($split_280a, ')') !== false && stripos($split_280a, '(') ===
false){
            $split_280a = str_replace(')', '', $split_280a);
        }

        $split_280c = explode('280$c', $data);
        $split_280c = explode ($split_280c[0].'280$c', $data);
        $split_280c = str_replace('280$c', '[semicolon]', $split_280c[1]);
        $split_280c = preg_split('/(\d){3}\$./', $split_280c);

        $split_280c = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_280c[0]);

        $split_280c = explode('[semicolon]', $split_280c);
```

```
        if($split_280c[0] == $split_280c[1]){
            $split_280c = $split_280c[0];
        }
        else{
            $split_280c = implode('[semicolon]', $split_280c);
        }

        $split_280c = str_replace('[semicolon]', ' ; ', $split_280c);

        if(substr_count($split_280c, '"') === 1){
            $split_280c = str_replace('"', '', $split_280c);
        }

        if(stripos($split_280c, '[') !== false && stripos($split_280c, ']') ===
false){
            $split_280c = str_replace('[', '', $split_280c);
        }

        if(stripos($split_280c, ']') !== false && stripos($split_280c, '[') ===
false){
            $split_280c = str_replace(']', '', $split_280c);
        }

        if(stripos($split_280c, '(') !== false && stripos($split_280c, ')') ===
false){
            $split_280c = str_replace('(', '', $split_280c);
        }

        if(stripos($split_280c, ')') !== false && stripos($split_280c, '(') ===
false){
            $split_280c = str_replace(')', '', $split_280c);
        }

        $split_280d = explode('280$d', $data);
        $split_280d = explode ($split_280d[0].'280$d', $data);
        $split_280d = str_replace('280$d', '[semicolon]', $split_280d[1]);
        $split_280d = preg_split('/(\d){3}\$./', $split_280d);

        $split_280d = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_280d[0]);

        $split_280d = explode('[semicolon]', $split_280d);

        if($split_280d[0] == $split_280d[1]){
            $split_280d = $split_280d[0];
        }
        else{
            $split_280d = implode('[semicolon]', $split_280d);
        }

        $split_280d = str_replace('[semicolon]', ' ; ', $split_280d);

        if(substr_count($split_280d, '"') === 1){
            $split_280d = str_replace('"', '', $split_280d);
        }

        if(stripos($split_280d, '[') !== false && stripos($split_280d, ']') ===
false){
```

```
            $split_280d = str_replace('[', '', $split_280d);
        }

        if(stripos($split_280d, ']') !== false && stripos($split_280d, '[') ===
false){
            $split_280d = str_replace(']', '', $split_280d);
        }

        if(stripos($split_280d, '(') !== false && stripos($split_280d, ')') ===
false){
            $split_280d = str_replace('(', '', $split_280d);
        }

        if(stripos($split_280d, ')') !== false && stripos($split_280d, '(') ===
false){
            $split_280d = str_replace(')', '', $split_280d);
        }

        if($split_280a != '' && $split_280c != '' && $split_280d != ''){
            $wor_description = $split_280a.' ; '.$split_280c.' ; '.$split_280d;
        }
        elseif($split_280a != '' && $split_280c != ''){
            $wor_description = $split_280a.' ; '.$split_280c;
        }
        elseif($split_280a != '' && $split_280d != ''){
            $wor_description = $split_280a.' ; '.$split_280d;
        }
        elseif($split_280c != '' && $split_280d != ''){
            $wor_description = $split_280c.' ; '.$split_280d;
        }
        else{
            $wor_description = $split_280a.$split_280c.$split_280d;
        }

        if(str_ends_with($wor_description, '.') || str_ends_with($wor_description,
'.]') || str_ends_with($wor_description, '.)' || str_ends_with($wor_description,
'."'') || str_ends_with($wor_description, '!') || str_ends_with($wor_description,
'?'))){
        }
        elseif($wor_description != ''){
            $wor_description = $wor_description.'.';
        }

        $wor_description = ucfirst($wor_description);

        $wor_description = preg_replace('/^"{3,}/', '"', $wor_description);
        $wor_description = preg_replace('/"{3,}$/', '"', $wor_description);
        $wor_description = str_replace('""""', '"', $wor_description);
        $wor_description = str_replace('""""', '"', $wor_description);

        if(stripos($wor_description, '"') !== false || stripos($wor_description,
',') !== false){
            $wor_description = '"'.$wor_description.'"';
        }

//      Create field wor_periodicity from 326$a, in accordance with taxonomy
"Périodicité".

        $split_326a = explode('326$a', $data);
```

```
        $split_326a = explode ($split_326a[0].'326$a', $data);
        $split_326a = str_replace('326$a', '[cut][cut]', $split_326a[1]);
        $split_326a = preg_split('/(\d){3}\$./', $split_326a);

        $split_326a = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_326a[0]);

        $split_326a = '[cut]'.$split_326a.'[cut]';

        $split_326a = str_ireplace('[cut][Hebdomadaire][cut]',
'[comma]Hebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]1 ou 2 fois par an[cut]', '[comma]Autre /
Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]1 vol. par an, publié par livraisons
décadaires[cut]', '[comma]Décadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]10 fois par an[cut]', '[comma]Autre /
Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]12 n° par vol.[cut]', '[comma]Autre /
Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]12 numéros formant 4 vol. par an[cut]',
'[comma]Autre / Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]12 numéros par mois (mention à partir du
n° 7)[cut]', '[comma]Autre / Irrégulière / Variable / Inconnue[comma]',
$split_326a);
        $split_326a = str_ireplace('[cut]12 tomes par an[cut]', '[comma]Autre /
Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]13 numéros par an[cut]', '[comma]Autre /
Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]15 cahiers par an[cut]', '[comma]Autre /
Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]15 fois par an (un cahier le 1er de
chaque mois, et un cahier supplémentaire le 15 des mois de janvier, mai et
septembre)[cut]', '[comma]Autre / Irrégulière / Variable / Inconnue[comma]',
$split_326a);
        $split_326a = str_ireplace('[cut]16 fois par mois[cut]', '[comma]Autre /
Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]16 tomes par an[cut]', '[comma]Autre /
Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]16 tomes par an (1 tome par mois ; 2
tomes à la fin de chaque trimestre)""[cut]', '[comma]Mensuelle[comma]',
$split_326a);
        $split_326a = str_ireplace('[cut]16 tomes par an (1 tome par mois ; 2
tomes dans les derniers mois de chaque trimestre)""[cut]',
'[comma]Mensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]16 vol. par an (1 vol. par mois ; 2 vol.
en janvier, avril, juillet, et octobre)""[cut]', '[comma]Mensuelle[comma]',
$split_326a);
        $split_326a = str_ireplace('[cut]18 n° formant 3 tomes par an[cut]',
'[comma]Autre / Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]2 à 3 fois par semaine, d\'après le
""Nouveau prospectus...""[cut]', '[comma]Autre / Irrégulière / Variable /
Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]2 à 3 n° par semaine[cut]', '[comma]Autre
/ Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]2 à 4 fois par semaine (prospectus, p.
3)[cut]', '[comma]Autre / Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]2 cahiers par mois[cut]',
'[comma]Bimensuelle[comma]', $split_326a);
```

```
        $split_326a = str_ireplace('[cut]2 fois par semaine[cut]',
'[comma]Bihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]2 fois par semaine (mercredi et
dimanche)[cut]', '[comma]Bihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]2 fois par semaine [1 n° et 1
suite][cut]', '[comma]Bihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]2 livraisons par mois[cut]',
'[comma]Bimensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]2 livraisons par mois + livraison suppl.
en fin de semestre[cut]', '[comma]Bimensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]2 numéros par semaine, réunis en tomes
bimestriels[cut]', '[comma]Bihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]2 parties par mois, réunies en tomes
mensuels[cut]', '[comma]Bimensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]2 tomes (4 parties) par an[cut]',
'[comma]Autre / Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]2 vol. par an[cut]',
'[comma]Semestrielle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]24 cahiers formant 4 volumes par
an[cut]', '[comma]Bimensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]24 n° par an[cut]',
'[comma]Bimensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]3 fois par mois[cut]',
'[comma]Trimensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]3 fois par semaine[cut]',
'[comma]Trihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]3 fois par semaine [1 n° et 2
suites][cut]', '[comma]Trihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]3 livraisons par mois + livraison suppl.
en fin de trimestre[cut]', '[comma]Trimensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]3 numéros par semaine[cut]',
'[comma]Trihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]3 ou 4 fois par semaine[cut]',
'[comma]Autre / Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]36 n° par an[cut]',
'[comma]Trimensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]4 fascicules formant 2 vol. par an[cut]',
'[comma]Autre / Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]4 fois par an (irrégulier)[cut]',
'[comma]Autre / Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]4 fois par décade[cut]', '[comma]Autre /
Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]4 fois par mois[cut]', '[comma]Autre /
Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]4 fois par semaine[cut]', '[comma]Autre /
Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]4 ou 5 fois par semaine[cut]',
'[comma]Autre / Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]4 parties par an[cut]', '[comma]Autre /
Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]4 tomes par an[cut]', '[comma]Autre /
Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]6 fois par an[cut]', '[comma]Autre /
Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]6 fois par semaine[cut]',
'[comma]Quotidienne[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]6 fois par semaine (irrégulier)[cut]',
'[comma]Quotidienne (irrégulière)[comma]', $split_326a);
```

```
        $split_326a = str_ireplace('[cut]6 livraisons par an[cut]', '[comma]Autre
/ Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]8 vol. par an[cut]', '[comma]Autre /
Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]À partir de 1798 tous les 2 jours[cut]',
'[comma]Tous les deux jours[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Annoncé comme bihebdomadaire[cut]',
'[comma]Bihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Annoncé comme bihebdomadaire (dimanche et
jeudi)[cut]', '[comma]Bihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Annoncé comme bimensuel[cut]',
'[comma]Bimensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Annoncé comme bimensuel ; en réalité,
huit numéros ont paru en 1780 et quatre en 1781""[cut]', '[comma]Bimensuelle
(irrégulière)[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Annoncé comme bimensuel ou
trimensuel[cut]', '[comma]Autre / Irrégulière / Variable / Inconnue[comma]',
$split_326a);
        $split_326a = str_ireplace('[cut]Annoncé comme devant paraître quatre fois
par décade, mais en réalité tridécadaire[cut]', '[comma]Autre / Irrégulière /
Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Annoncé comme hebdomadaire[cut]',
'[comma]Hebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Annoncé comme hebdomadaire ; en réalité
tous les cinq jours""[cut]', '[comma]Autre / Irrégulière / Variable /
Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Annoncé comme hebdomadaire, souvent
irrégulier[cut]', '[comma]Hebdomadaire (irrégulière)[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Annoncé comme irrégulier (selon le rythme
de parution des lois)[cut]', '[comma]Autre / Irrégulière / Variable /
Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]annoncé comme mensuel ; en réalité
bimensuel""[cut]', '[comma]Bimensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Annoncé comme paraissant deux fois par
décade ; en réalité irrégulier""[cut]', '[comma]Autre / Irrégulière / Variable /
Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Annoncé comme paraissant tous les jours
pairs ; en réalité semi hebdomadaire (?)""[cut]', '[comma]Bihebdomadaire[comma]',
$split_326a);
        $split_326a = str_ireplace('[cut]Annoncé comme quotidien[cut]',
'[comma]Quotidienne[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Annoncé comme trihebdomadaire ; en
réalité bihebdomadaire (irrégulier)""[cut]', '[comma]Bihebdomadaire
(irrégulière)[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Annoncé comme trihebdomadaire,
périodicité en fait plus espacée[cut]', '[comma]Autre / Irrégulière / Variable /
Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Annonce le 15 nov. 1790 qu\'il paraîtra
les 5, 15 et 25 de chaque mois à partir de 1791 ; annonce le 17 déc. 1790 qu\'il
paraîtra le vendredi de chaque semaine""[cut]', '[comma]Trimensuelle[comma]',
$split_326a);
        $split_326a = str_ireplace('[cut]Annoncé pour paraître tous les
dimanches[cut]', '[comma]Hebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Annonce qu\'il paraîtra quatre fois par
mois, à des époques indéterminées[cut]', '[comma]Autre / Irrégulière / Variable /
Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Annoncé quotidien[cut]',
'[comma]Quotidienne[comma]', $split_326a);
```

```
        $split_326a = str_ireplace('[cut]Annuel[cut]', '[comma]Annuelle[comma]',
$split_326a);
        $split_326a = str_ireplace('[cut]Annuel (1867-1959)[cut]',
'[comma]Annuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Annuel (2 séances en 1807)[cut]',
'[comma]Annuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Annuel (a paru très
irrégulièrement)[cut]', '[comma]Annuelle (irrégulière)[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Annuel (deux volumes par an)[cut]',
'[comma]Annuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Annuel (irrégulier jusqu\'en 1861)[cut]',
'[comma]Annuelle (irrégulière)[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Annuel (irrégulier)[cut]',
'[comma]Annuelle (irrégulière)[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Annuel (parfois 2 fois par an)[cut]',
'[comma]Annuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]annuel (parfois 2 ou 3 vol. par
an)[cut]', '[comma]Annuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]annuel ou biennal[cut]', '[comma]Autre /
Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]annuel, parfois biennal[cut]',
'[comma]Annuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Annuel, parfois irrégulier[cut]',
'[comma]Annuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Au moins 6 fois par mois[cut]',
'[comma]Autre / Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Au moins une fois par décade[cut]',
'[comma]Décadaire (irrégulière)[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Au moins une fois par semaine[cut]',
'[comma]Hebdomadaire (irrégulière)[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Bidécadaire[cut]',
'[comma]Bidécadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Bidécadaire [?][cut]',
'[comma]Bidécadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]biennal[cut]', '[comma]Biennale[comma]',
$split_326a);
        $split_326a = str_ireplace('[cut]Biennal, parfois annuel[cut]',
'[comma]Biennale (irrégulière)[comma]', $split_326a);

        $split_326a = str_ireplace('[cut]bihebdomadaire[cut]',
'[comma]Bihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Bi-hebdomadaire[cut]',
'[comma]Bihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Bihebdomadaire (irrégulier à partir de
1790)[cut]', '[comma]Bihebdomadaire (irrégulière)[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Bihebdomadaire (irrégulier)[cut]',
'[comma]Bihebdomadaire (irrégulière)[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Bihebdomadaire (mardi et vendredi)[cut]',
'[comma]Bihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Bihebdomadaire ou hebdomadaire[cut]',
'[comma]Autre / Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Bihebdomadaire ou trihebdomadaire[cut]',
'[comma]Autre / Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]bihebdomadaire ou trihebdomadaire ou six
fois par semaine[cut]', '[comma]Autre / Irrégulière / Variable / Inconnue[comma]',
$split_326a);
        $split_326a = str_ireplace('[cut]Bihebdomadaire puis 4 fois par
semaine[cut]', '[comma]Bihebdomadaire[comma]', $split_326a);
```

```
        $split_326a = str_ireplace('[cut]Bimensuel[cut]',
'[comma]Bimensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Bi-mensuel[cut]',
'[comma]Bimensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Bimensuel (24 numéros formant 3 tomes par
an)[cut]', '[comma]Bimensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Bimensuel (irrégulier)[cut]',
'[comma]Bimensuelle (irrégulière)[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Bi-mensuel (irrégulier)[cut]',
'[comma]Bimensuelle (irrégulière)[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Bimensuel (l\'année comprend 96 numéros,
parus en 24 livraisons de 4 numéros)[cut]', '[comma]Bimensuelle[comma]',
$split_326a);
        $split_326a = str_ireplace('[cut]Bi-mensuel (le 10 et le 25)[cut]',
'[comma]Bimensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Bimensuel [?][cut]',
'[comma]Bimensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Bimensuel ou hebdomadaire[cut]',
'[comma]Autre / Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Bimensuel ou mensuel[cut]',
'[comma]Bimensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Bi-mensuel, le 1 et le 15[cut]',
'[comma]Bimensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Bimensuel, le 15 et le 30 (très
irrégulier)[cut]', '[comma]Bimensuelle (irrégulière)[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Bi-mensuel, le 15 et le 30 de chaque
mois[cut]', '[comma]Bimensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Bi-mensuel, le 1er et le 15[cut]',
'[comma]Bimensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Bimensuel, le 1er et le 15 de chaque mois
(3 cahiers en décembre 1756 et 1757)[cut]', '[comma]Bimensuelle[comma]',
$split_326a);
        $split_326a = str_ireplace('[cut]bi-mensuel, les 8 et 23 du mois[cut]',
'[comma]Bimensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Bi-mensuel, parfois irrégulier (25
numéros par an)[cut]', '[comma]Bimensuelle (irrégulière)[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]bimestriel[cut]',
'[comma]Bimestrielle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]bimestriel (2 numéros par tome)[cut]',
'[comma]Bimestrielle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Cinq fois par décade[cut]', '[comma]Tous
les deux jours[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Cinq fois par mois[cut]', '[comma]Autre /
Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Cinq fois par semaine[cut]',
'[comma]Autre / Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Cinq ou six fois par semaine[cut]',
'[comma]Autre / Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]De 2 à 5 fois par semaine[cut]',
'[comma]Autre / Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]de quinzaine en quinzaine[cut]',
'[comma]Bimensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]De une à trois fois par semaine[cut]',
'[comma]Autre / Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]décadaire[cut]',
'[comma]Décadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Décennal[cut]',
'[comma]Décennale[comma]', $split_326a);
```

```
        $split_326a = str_ireplace('[cut]Deux ""demi-volumes"" ou parties par
an[cut]', '[comma]Semestrielle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Deux (parfois trois) fois par mois[cut]',
'[comma]Bimensuelle (irrégulière)[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]deux fois deux ou trois numéros par
mois[cut]', '[comma]Autre / Irrégulière / Variable / Inconnue[comma]',
$split_326a);
        $split_326a = str_ireplace('[cut]Deux fois par an[cut]',
'[comma]Semestrielle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]deux fois par décade[cut]',
'[comma]Bidécadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Deux fois par décade, les primidi et
sextidi[cut]', '[comma]Bidécadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Deux fois par décade, les quintidi et
décadi[cut]', '[comma]Bidécadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Deux fois par jour[cut]',
'[comma]Biquotidienne[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Deux fois par jour, juil-21 sept.[cut]',
'[comma]Biquotidienne[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Deux fois par jour, sept.-nov.
1792[cut]', '[comma]Biquotidienne[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]deux fois par mois[cut]',
'[comma]Bimensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Deux fois par semaine[cut]',
'[comma]Bihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Deux fois par semaine (?)[cut]',
'[comma]Bihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Deux fois par semaine (8 numéros par tome
mensuel)[cut]', '[comma]Bihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Deux fois par semaine (dimanche et
jeudi)[cut]', '[comma]Bihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Deux fois par semaine (dimanche et
mercredi)[cut]', '[comma]Bihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Deux fois par semaine (irrégulier)[cut]',
'[comma]Bihebdomadaire (irrégulière)[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Deux fois par semaine (jeudi et
dimanche)[cut]', '[comma]Bihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Deux fois par semaine (jeudi et
lundi)[cut]', '[comma]Bihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Deux fois par semaine (les mardi et
samedi)[cut]', '[comma]Bihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Deux fois par semaine (les mardi et
vendredi)[cut]', '[comma]Bihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Deux fois par semaine (les mardis et
samedis)[cut]', '[comma]Bihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Deux fois par semaine (les mercredi et
samedi)[cut]', '[comma]Bihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Deux fois par semaine (lundi et
jeudi)[cut]', '[comma]Bihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Deux fois par semaine (mention au n°
19)[cut]', '[comma]Bihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Deux fois par semaine (mention aux n° 2
et 4)[cut]', '[comma]Bihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Deux fois par semaine (mercredi et
samedi)[cut]', '[comma]Bihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Deux fois par semaine (samedi et
mardi)[cut]', '[comma]Bihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Deux fois par semaine (samedi et
mercredi)[cut]', '[comma]Bihebdomadaire[comma]', $split_326a);
```

```
        $split_326a = str_ireplace('[cut]Deux fois par semaine, le dimanche et le
jeudi[cut]', '[comma]Bihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Deux fois par semaine, le lundi et le
jeudi[cut]', '[comma]Bihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]deux fois par semaine, les jeudi et
dimanche[cut]', '[comma]Bihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Deux fois par semaine, les jeudi et
lundi[cut]', '[comma]Bihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Deux fois par semaine, les lundi et
jeudi[cut]', '[comma]Bihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Deux fois par semaine, les lundis et
jeudis[cut]', '[comma]Bihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Deux fois par semaine, les mardi et
samedi[cut]', '[comma]Bihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Deux fois par semaine, les mardi et
samedi (irrégulier à partir du n° VII, 3 août)[cut]', '[comma]Bihebdomadaire
(irrégulière)[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Deux fois par semaine, les mardis et
vendredis[cut]', '[comma]Bihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Deux fois par semaine, les mercredi et
dimanche[cut]', '[comma]Bihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Deux fois par semaine, les mercredi et
dimanche puis (1815) les lundi et jeudi puis (1816) les lundi et vendredi[cut]',
'[comma]Bihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Deux fois par semaine, les mercredi et
samedi[cut]', '[comma]Bihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Deux fois par semaine, les mercredis et
samedis[cut]', '[comma]Bihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Deux fois par semaine, samedi et vendredi
(n° 1) puis lundi et jeudi (n° 2-100)[cut]', '[comma]Bihebdomadaire[comma]',
$split_326a);
        $split_326a = str_ireplace('[cut]Deux numéros par an[cut]',
'[comma]Semestrielle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Deux numéros par semaine, les lundis et
jeudis, avec suppléments les mardis et vendredis[cut]',
'[comma]Bihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Deux ou trois fois par an[cut]',
'[comma]Autre / Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Deux ou trois fois par mois[cut]',
'[comma]Autre / Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Deux ou trois fois par semaine[cut]',
'[comma]Autre / Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]deux vol. par an[cut]',
'[comma]Semestrielle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Entre 1 et 3 fois par semaine[cut]',
'[comma]Autre / Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Hebd[cut]', '[comma]Hebdomadaire[comma]',
$split_326a);
        $split_326a = str_ireplace('[cut]Hebdo.[cut]',
'[comma]Hebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Hebdomadaire[cut]',
'[comma]Hebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Hebdomadaire (""au moins une [lettre]
chaque semaine"")[cut]', '[comma]Hebdomadaire (irrégulière)[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Hebdomadaire (?)[cut]',
'[comma]Hebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Hebdomadaire (2 tomes par an, comportant
52 numéros chiffrés en continu)[cut]', '[comma]Hebdomadaire[comma]', $split_326a);
```

```
        $split_326a = str_ireplace('[cut]Hebdomadaire (annoncé irrégulier pour
contourner l\'autorisation préalable)[cut]', '[comma]Hebdomadaire[comma]',
$split_326a);
        $split_326a = str_ireplace('[cut]Hebdomadaire (en fait très
irrégulier)[cut]', '[comma]Hebdomadaire (irrégulière)[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Hebdomadaire (irrégulier)[cut]',
'[comma]Hebdomadaire (irrégulière)[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Hebdomadaire (le lundi)[cut]',
'[comma]Hebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Hebdomadaire (le lundi, avec parfois un
extraordinaire le jeudi)[cut]', '[comma]Hebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Hebdomadaire (le mardi)[cut]',
'[comma]Hebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Hebdomadaire (le mercredi)[cut]',
'[comma]Hebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Hebdomadaire (le mercredi, puis le jeudi
à partir de 1791)[cut]', '[comma]Hebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Hebdomadaire (le samedi)[cut]',
'[comma]Hebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Hebdomadaire (le vendredi)[cut]',
'[comma]Hebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Hebdomadaire (légèrement
irrégulier)[cut]', '[comma]Hebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Hebdomadaire (parfois irrégulier)[cut]',
'[comma]Hebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Hebdomadaire (suspendu entre le 22 juin
et le 20 juillet)[cut]', '[comma]Hebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Hebdomadaire (tous les 15 jours jusqu\'au
n° 3)[cut]', '[comma]Hebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Hebdomadaire (tous les lundis)[cut]',
'[comma]Hebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Hebdomadaire (tous les mardis)[cut]',
'[comma]Hebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Hebdomadaire (très irrégulier)[cut]',
'[comma]Hebdomadaire (irrégulière)[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Hebdomadaire (varie)[cut]',
'[comma]Hebdomadaire (irrégulière)[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Hebdomadaire ?[cut]',
'[comma]Hebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Hebdomadaire [?][cut]',
'[comma]Hebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]hebdomadaire et bihebdomadaire en
alternance[cut]', '[comma]Autre / Irrégulière / Variable / Inconnue[comma]',
$split_326a);
        $split_326a = str_ireplace('[cut]Hebdomadaire ou bihebdomadaire[cut]',
'[comma]Autre / Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Hebdomadaire ou bi-mensuel[cut]',
'[comma]Autre / Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Hebdomadaire ou semi hebdomadaire[cut]',
'[comma]Autre / Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Hebdomadaire, avec quelques
extraordinaires[cut]', '[comma]Hebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Hebdomadaire, le dimanche[cut]',
'[comma]Hebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Hebdomadaire, le jeudi[cut]',
'[comma]Hebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Hebdomadaire, le lundi[cut]',
'[comma]Hebdomadaire[comma]', $split_326a);
```

```
        $split_326a = str_ireplace('[cut]Hebdomadaire, le mardi[cut]',
'[comma]Hebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Hebdomadaire, le mercredi[cut]',
'[comma]Hebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Hebdomadaire, le samedi[cut]',
'[comma]Hebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Hebdomadaire, le samedi puis le
mardi[cut]', '[comma]Hebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Hebdomadaire, le vendredi[cut]',
'[comma]Hebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Hebdomadaire, le vendredi (n° 1-6) puis
le samedi (n° 7)[cut]', '[comma]Hebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Hebdomadaire, parfois irrégulier[cut]',
'[comma]Hebdomadaire (irrégulière)[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Hebdomadaire, tous les jeudis[cut]',
'[comma]Hebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Hebdomadaire, tous les mardis[cut]',
'[comma]Hebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Hebomadaire[cut]',
'[comma]Hebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Huit cahiers par an[cut]', '[comma]Autre
/ Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Huit fois par an""[cut]', '[comma]Autre /
Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Huit fois par mois[cut]',
'[comma]Bihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Huit fois par trimestre[cut]',
'[comma]Autre / Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Huit livraisons formant deux volumes par
an[cut]', '[comma]Autre / Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Huit livraisons par mois (deux fois par
semaine)[cut]', '[comma]Bihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Inconnue[cut]', '[comma]Autre /
Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]irrégulier[cut]', '[comma]Autre /
Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Irrégulier (""""de tems à autres""""" ;
"""""jamais moins de deux fois par décade""""")""[cut]', '[comma]Autre / Irrégulière
/ Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Irrégulier (""un jour
entr\'autres"")[cut]', '[comma]Autre / Irrégulière / Variable / Inconnue[comma]',
$split_326a);
        $split_326a = str_ireplace('[cut]Irrégulier (""Une ou deux fois par
semaine"")[cut]', '[comma]Autre / Irrégulière / Variable / Inconnue[comma]',
$split_326a);
        $split_326a = str_ireplace('[cut]Irrégulier (1 à 3 fois par mois)[cut]',
'[comma]Autre / Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Irrégulier (1817-1866)[cut]',
'[comma]Autre / Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Irrégulier (2 à 4 fois par
semaine)[cut]', '[comma]Autre / Irrégulière / Variable / Inconnue[comma]',
$split_326a);
        $split_326a = str_ireplace('[cut]Irrégulier (2 ou 3 fois par
semaine)[cut]', '[comma]Autre / Irrégulière / Variable / Inconnue[comma]',
$split_326a);
        $split_326a = str_ireplace('[cut]Irrégulier (5 ou 6 fois par mois)[cut]',
'[comma]Autre / Irrégulière / Variable / Inconnue[comma]', $split_326a);
```

```
        $split_326a = str_ireplace('[cut]Irrégulier (annoncé bimestriel dans la
préface de la 1re partie et dans l\'avis de la 4e partie)[cut]', '[comma]Autre /
Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Irrégulier (annoncé trois fois par décade
au n° 1)[cut]', '[comma]Autre / Irrégulière / Variable / Inconnue[comma]',
$split_326a);
        $split_326a = str_ireplace('[cut]Irrégulier (avertissement au t. I)[cut]',
'[comma]Autre / Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Irrégulier (entre 1 et 4 numéros par
mois)[cut]', '[comma]Autre / Irrégulière / Variable / Inconnue[comma]',
$split_326a);
        $split_326a = str_ireplace('[cut]Irrégulier (pluriennal)[cut]',
'[comma]Autre / Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Irrégulier (prévu mensuel)[cut]',
'[comma]Autre / Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Irrégulier (t. 1 et 2 datés 1722, t. 3
daté 1726)[cut]', '[comma]Autre / Irrégulière / Variable / Inconnue[comma]',
$split_326a);
        $split_326a = str_ireplace('[cut]Irrégulier (tous les 4 à 10 jours ou
plus)[cut]', '[comma]Autre / Irrégulière / Variable / Inconnue[comma]',
$split_326a);
        $split_326a = str_ireplace('[cut]Irrégulier (tous les cinq, dix, quinze ou
vingt jours)[cut]', '[comma]Autre / Irrégulière / Variable / Inconnue[comma]',
$split_326a);
        $split_326a = str_ireplace('[cut]Irrégulier ou bi-mensuel[cut]',
'[comma]Autre / Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]irrégulier, paraissant par parties
complètes[cut]', '[comma]Autre / Irrégulière / Variable / Inconnue[comma]',
$split_326a);
        $split_326a = str_ireplace('[cut]Irrégulier, puis 1 vol. par
session[cut]', '[comma]Autre / Irrégulière / Variable / Inconnue[comma]',
$split_326a);
        $split_326a = str_ireplace('[cut]La périodicité varie[cut]', '[comma]Autre
/ Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Livraisons mensuelle réunies en tomes
annuels[cut]', '[comma]Mensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Livraisons mensuelles formant des tomes
semestriels[cut]', '[comma]Mensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Livraisons mensuelles regroupées en
volumes annuels[cut]', '[comma]Mensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Livraisons mensuelles regroupées en
volumes annuels (biennaux de 1764 à 1767)[cut]', '[comma]Mensuelle[comma]',
$split_326a);
        $split_326a = str_ireplace('[cut]Livraisons mensuelles réunies en tomes
annuels[cut]', '[comma]Mensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Livraisons mensuelles, formant 2 tomes
par an[cut]', '[comma]Mensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Livraisons mensuelles, formant 2 vol. par
an[cut]', '[comma]Mensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Livraisons mensuelles, formant 3 vol. par
an[cut]', '[comma]Mensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Livraisons mensuelles, formant un tome
par semestre[cut]', '[comma]Mensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Livraisons mensuelles, réunies en tomes
semestriels[cut]', '[comma]Mensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Livraisons mensuelles, réunies en vol.
par semestre[cut]', '[comma]Mensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]livraisons quadrimestrielles formant des
tomes annuels[cut]', '[comma]Mensuelle[comma]', $split_326a);
```

```
        $split_326a = str_ireplace('[cut]Mens[cut]', '[comma]Mensuelle[comma]',
$split_326a);
        $split_326a = str_ireplace('[cut]mensuel[cut]', '[comma]Mensuelle[comma]',
$split_326a);
        $split_326a = str_ireplace('[cut]Mensuel (12 livraisons formant 3 vol. par
an)[cut]', '[comma]Mensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Mensuel (12 livraisons formant 4 vol. par
an)[cut]', '[comma]Mensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Mensuel (18 numéros par an en 1676-
1677)[cut]', '[comma]Mensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Mensuel (2 numéros en janvier)[cut]',
'[comma]Mensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Mensuel (4 numéros par tome)[cut]',
'[comma]Mensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Mensuel (annonce à la fin ""la suite à la
fin de chaque mois"")[cut]', '[comma]Mensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Mensuel (avis à la fin : ""On donnera
dans la suite le Secretaire du Parnasse, le premier jour de chaque mois"")[cut]',
'[comma]Mensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Mensuel (d\'après le ""Mercure de
France"", avril 1756, 2e vol., p. 81)[cut]', '[comma]Mensuelle[comma]',
$split_326a);
        $split_326a = str_ireplace('[cut]Mensuel (irrégulier en 1686)[cut]',
'[comma]Mensuelle (irrégulière)[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Mensuel (irrégulier en 1817)[cut]',
'[comma]Mensuelle (irrégulière)[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Mensuel (irrégulier)[cut]',
'[comma]Mensuelle (irrégulière)[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]mensuel + livraison suppl. en fin de
quadrimestre[cut]', '[comma]Mensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Mensuel avec supplément semestriel inclus
dans la numérotation[cut]', '[comma]Mensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Mensuel ou bimestriel[cut]',
'[comma]Autre / Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Mensuel puis irrégulier[cut]',
'[comma]Autre / Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Mensuel, d\'après le catalogue de
l\'éditeur de fructidor an XI[cut]', '[comma]Mensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Mensuel, parfois bimestriel[cut]',
'[comma]Mensuelle (irrégulière)[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Mensuel, sauf septembre-octobre[cut]',
'[comma]Mensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Numéros bimensuels réunis en tomes
quadrimestriels[cut]', '[comma]Bimensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Ouvrage semi-périodique[cut]',
'[comma]Autre / Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Paraissant tous les jours, le lundi
excepté[cut]', '[comma]Quotidienne[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Paraît les 5 et 10 de chaque
décade[cut]', '[comma]Bidécadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Paraît les 5, 10, 15, 20, 25 et 30 de
chaque mois[cut]', '[comma]Bidécadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Paraît tous les dix jours[cut]',
'[comma]Décadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Paru par livraisons mensuelles[cut]',
'[comma]Mensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Paru par livraisons sans périodicité
régulière (entre 25 et 30 par volume)[cut]', '[comma]Autre / Irrégulière /
Variable / Inconnue[comma]', $split_326a);
```

```
        $split_326a = str_ireplace('[cut]Pas de périodicité indiquée[cut]',
'[comma]Autre / Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Pas de périodicité indiquée (tous les 2
ou 3 jours)[cut]', '[comma]Autre / Irrégulière / Variable / Inconnue[comma]',
$split_326a);
        $split_326a = str_ireplace('[cut]Pas moins de deux ni plus de trois fois
par semaine[cut]', '[comma]Autre / Irrégulière / Variable / Inconnue[comma]',
$split_326a);
        $split_326a = str_ireplace('[cut]périodicité inconnue[cut]', '[comma]Autre
/ Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Périodicité non définie, peut-être
hebdomadaire[cut]', '[comma]Autre / Irrégulière / Variable / Inconnue[comma]',
$split_326a);
        $split_326a = str_ireplace('[cut]Périodicité variable[cut]', '[comma]Autre
/ Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Périodicité variable (quotidien en
1792)[cut]', '[comma]Autre / Irrégulière / Variable / Inconnue[comma]',
$split_326a);
        $split_326a = str_ireplace('[cut]Pluriennal[cut]', '[comma]Autre /
Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Plusieurs fois par semaine[cut]',
'[comma]Autre / Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Prévu quotidien, d\'après le
prospectus[cut]', '[comma]Quotidienne[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]puis 2 fois par semaine[cut]',
'[comma]Bihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]puis 2 ou 3 vol. par an, publiés par
livraisons de périodicité variable[cut]', '[comma]Autre / Irrégulière / Variable /
Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]puis 5 fois par semaine[cut]',
'[comma]Autre / Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]puis annuel[cut]',
'[comma]Annuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]puis bi-mensuel[cut]',
'[comma]Bimensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]puis bimestriel[cut]',
'[comma]Bimestrielle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]puis deux fois par semaine[cut]',
'[comma]Bihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]puis hebdomadaire[cut]',
'[comma]Hebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]puis irrégulier[cut]', '[comma]Autre /
Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]puis mensuel[cut]',
'[comma]Mensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]puis périodicité variable[cut]',
'[comma]Autre / Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]puis pluriennal[cut]', '[comma]Autre /
Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]puis quotidien[cut]',
'[comma]Quotidienne[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]puis trihebdomadaire[cut]',
'[comma]Trihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]puis trimestriel[cut]',
'[comma]Trimestrielle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]puis trois fois par semaine[cut]',
'[comma]Trihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]quadriennal""[cut]',
'[comma]Quadriennale[comma]', $split_326a);
```

```
        $split_326a = str_ireplace('[cut]Quatre fois par an[cut]',
'[comma]Trimestrielle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]quatre fois par décade[cut]',
'[comma]Autre / Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Quatre fois par mois[cut]',
'[comma]Hebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Quatre fois par mois, à des époques
indéterminées[cut]', '[comma]Hebdomadaire (irrégulière)[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]quatre fois par semaine[cut]',
'[comma]Autre / Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Quatre fois par semaine
(irrégulier)[cut]', '[comma]Autre / Irrégulière / Variable / Inconnue[comma]',
$split_326a);
        $split_326a = str_ireplace('[cut]Quatre vol. par an (certaines années en 3
vol. ; 6 vol. en 1797 ; 5 vol. en 1798)""[cut]', '[comma]Autre / Irrégulière /
Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Quinquennal[cut]',
'[comma]Quinquennale[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Quotid[cut]',
'[comma]Quotidienne[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]quotiden[cut]',
'[comma]Quotidienne[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Quotidien[cut]',
'[comma]Quotidienne[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Quotidien (""la suite à demain"")[cut]',
'[comma]Quotidienne[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Quotidien (annonce ""la suite à
demain"")[cut]', '[comma]Quotidienne[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Quotidien (d\'après le supplément n ° 28,
du 9 sept. 1791, relié en début du t. 20 dans la collection de la BnF)[cut]',
'[comma]Quotidienne[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Quotidien (irrégulier)[cut]',
'[comma]Quotidienne (irrégulière)[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]quotidien (le dimanche excepté)[cut]',
'[comma]Quotidienne[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Quotidien (légèrement irrégulier)[cut]',
'[comma]Quotidienne (irrégulière)[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Quotidien (Ne paraît pas le
dimanche)[cut]', '[comma]Quotidienne[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Quotidien (paraît le lendemain de chaque
séance de l\'Assemblée nationale)[cut]', '[comma]Quotidienne[comma]',
$split_326a);
        $split_326a = str_ireplace('[cut]Quotidien (paraît le lendemain de chaque
séance nationale)[cut]', '[comma]Quotidienne[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Quotidien (sauf le dimanche)[cut]',
'[comma]Quotidienne[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Quotidien (tous les jours à sept heures
du soir)[cut]', '[comma]Quotidienne[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Quotidien (tous les matins)[cut]',
'[comma]Quotidienne[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Quotidien (tous les soirs)[cut]',
'[comma]Quotidienne[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Quotidien (très irrégulier)[cut]',
'[comma]Quotidienne (irrégulière)[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Quotidien [?][cut]',
'[comma]Quotidienne[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Quotidien ou irrégulier[cut]',
'[comma]Quotidienne (irrégulière)[comma]', $split_326a);
```

```
        $split_326a = str_ireplace('[cut]Quotidien ou six fois par semaine[cut]',
'[comma]Quotidienne (irrégulière)[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Quotidien puis tri-hebdomadaire puis
quotidien[cut]', '[comma]Autre / Irrégulière / Variable / Inconnue[comma]',
$split_326a);
        $split_326a = str_ireplace('[cut]Quotidien, 22 sept.- [nov.?][cut]',
'[comma]Quotidienne[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]quotidien, avec n° supplémentaires[cut]',
'[comma]Quotidienne[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Quotidien, déc. 1792-1793[cut]',
'[comma]Quotidienne[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Quotidien, excepté le dimanche[cut]',
'[comma]Quotidienne[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Quotidien, parfois bi-quotidien[cut]',
'[comma]Quotidienne (irrégulière)[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Quotidien, parfois irrégulier[cut]',
'[comma]Quotidienne (irrégulière)[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Quotidien, sauf dimanche[cut]',
'[comma]Quotidienne[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Quotidien, sauf le dimanche[cut]',
'[comma]Quotidienne[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Quotidien, sauf le lundi[cut]',
'[comma]Quotidienne[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Quotidien, sauf lundi[cut]',
'[comma]Quotidienne[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]semestriel[cut]',
'[comma]Semestrielle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Semestriel (irrégulier à partir de
1778)[cut]', '[comma]Semestrielle (irrégulière)[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Semestriel (irrégulier)[cut]',
'[comma]Semestrielle (irrégulière)[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Semestriel (Saint Jean d\'été et Saint
Jean d\'hiver)[cut]', '[comma]Semestrielle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]semestriel, parfois annuel[cut]',
'[comma]Semestrielle (irrégulière)[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Semestriel, parfois trimestriel[cut]',
'[comma]Semestrielle (irrégulière)[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Semi décadaire[cut]',
'[comma]Bidécadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Semi hebdomadaire[cut]',
'[comma]Bihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Semi hebdomadaire ou hebdomadaire[cut]',
'[comma]Autre / Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Semi-hebd[cut]',
'[comma]Bihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]semi-hebdomadaire[cut]',
'[comma]Bihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Semi-hebdomadaire (les mardi et
samedi)[cut]', '[comma]Bihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]semi-hebdomadaire (un numéro en deux
parties par semaine) à partir du n° IX [fin octobre 1789][cut]',
'[comma]Bihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Six fois la semaine[cut]',
'[comma]Quotidienne[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Six fois par mois[cut]', '[comma]Autre /
Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Six fois par semaine[cut]',
'[comma]Quotidienne[comma]', $split_326a);
```

```
        $split_326a = str_ireplace('[cut]six numéros par an[cut]',
'[comma]Bimestrielle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Tous les 10 jours[cut]',
'[comma]Décadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Tous les 10 jours (les quintidi)[cut]',
'[comma]Décadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]tous les 10 ou 20 jours[cut]',
'[comma]Autre / Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Tous les 15 jours[cut]',
'[comma]Bimensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Tous les 15 jours à peu près
(avertissement au n° I)[cut]', '[comma]Bimensuelle (irrégulière)[comma]',
$split_326a);
        $split_326a = str_ireplace('[cut]Tous les 2 jours[cut]', '[comma]Tous les
deux jours[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]tous les 2 jours [1 n° et 2 ou 3 suites
par semaine][cut]', '[comma]Tous les deux jours[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Tous les 2 ou 3 jours[cut]',
'[comma]Autre / Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Tous les 3 jours[cut]', '[comma]Tous les
trois jours[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Tous les 4 jours[cut]', '[comma]Tous les
quatre jours[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Tous les 5 jours[cut]',
'[comma]Bidécadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Tous les 5 jours (18 numéros constituant
3 tomes par trimestre)[cut]', '[comma]Bidécadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]tous les 5 jours [irrégulier][cut]',
'[comma]Bidécadaire (irrégulière)[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Tous les cinq jours[cut]',
'[comma]Bidécadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Tous les cinq jours (le 1er et le 6 de
chaque décade)[cut]', '[comma]Bidécadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]tous les cinq mois[cut]', '[comma]Autre /
Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]tous les deux jours[cut]', '[comma]Tous
les deux jours[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Tous les deux jours (irrégulier)[cut]',
'[comma]Tous les deux jours (irrégulière)[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Tous les deux jours (les jours pairs de
la décade)[cut]', '[comma]Tous les deux jours[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Tous les deux jours du calendrier
républicain[cut]', '[comma]Tous les deux jours[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Tous les deux jours jusqu\'au n° 9 (9
frimaire an II) [29 nov. 1793][cut]', '[comma]Tous les deux jours[comma]',
$split_326a);
        $split_326a = str_ireplace('[cut]Tous les deux jours ou
trihebdomadaire[cut]', '[comma]Trihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Tous les deux mois[cut]',
'[comma]Bimestrielle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]tous les deux ou quatre jours[cut]',
'[comma]Autre / Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Tous les deux ou trois jours[cut]',
'[comma]Autre / Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Tous les deux semaines, le
dimanche[cut]', '[comma]Bimensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Tous les dix jours[cut]',
'[comma]Décadaire[comma]', $split_326a);
```

```
        $split_326a = str_ireplace('[cut]Tous les dix jours (?)[cut]',
'[comma]Décadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Tous les dix jours ou irrégulier[cut]',
'[comma]Décadaire (irrégulière)[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Tous les jours impairs[cut]',
'[comma]Tous les deux jours[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Tous les jours impairs du calendrier
républicain[cut]', '[comma]Tous les deux jours[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Tous les jours ou tous les 2 jours[cut]',
'[comma]Autre / Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Tous les jours ou tous les deux
jours[cut]', '[comma]Autre / Irrégulière / Variable / Inconnue[comma]',
$split_326a);
        $split_326a = str_ireplace('[cut]Tous les jours pairs de la décade[cut]',
'[comma]Tous les deux jours[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Tous les jours sauf lundi[cut]',
'[comma]Quotidienne[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Tous les jours, excepté le
dimanche[cut]', '[comma]Quotidienne[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Tous les quatre jours[cut]', '[comma]Tous
les quatre jours[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Tous les quinze jours[cut]',
'[comma]Bimensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Tous les quinze jours (irrégulier)[cut]',
'[comma]Bimensuelle (irrégulière)[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Tous les quinze jours (livraisons réunies
en volumes bimestriels puis trimestriels)[cut]', '[comma]Bimensuelle[comma]',
$split_326a);
        $split_326a = str_ireplace('[cut]tous les six jours[cut]', '[comma]Autre /
Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Tous les trois jours[cut]', '[comma]Tous
les trois jours[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]toutes les 2 semaines[cut]',
'[comma]Bimensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Toutes les 3 semaines[cut]',
'[comma]Autre / Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Toutes les deux semaines[cut]',
'[comma]Bimensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Toutes les deux semaines
(irrégulier)[cut]', '[comma]Bimensuelle (irrégulière)[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Toutes les deux semaines, le
dimanche[cut]', '[comma]Bimensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Toutes les deux semaines, le jeudi[cut]',
'[comma]Bimensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Toutes les deux semaines, le lundi[cut]',
'[comma]Bimensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Toutes les deux semaines, le mardi[cut]',
'[comma]Bimensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Toutes les deux semaines, le
mercredi[cut]', '[comma]Bimensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Toutes les deux semaines, le
samedi[cut]', '[comma]Bimensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Toutes les deux semaines, le
vendredi[cut]', '[comma]Bimensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Toutes les six semaines
(irrégulier)[cut]', '[comma]Autre / Irrégulière / Variable / Inconnue[comma]',
$split_326a);
        $split_326a = str_ireplace('[cut]Tridécadaire[cut]',
'[comma]Tridécadaire[comma]', $split_326a);
```

```
        $split_326a = str_ireplace('[cut]triennal[cut]',
'[comma]Triennale[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]trihebdomadaire[cut]',
'[comma]Trihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Tri-hebdomadaire[cut]',
'[comma]Trihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Trihebdomadaire [?][cut]',
'[comma]Trihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]trimensuel[cut]',
'[comma]Trimensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Trimest[cut]',
'[comma]Trimestrielle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]trimestriel[cut]',
'[comma]Trimestrielle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Trimestriel (irrégulier à partir du tome
X)[cut]', '[comma]Trimestrielle (irrégulière)[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Trimestriel (irrégulier)[cut]',
'[comma]Trimestrielle (irrégulière)[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Trimestriel""[cut]',
'[comma]Trimestrielle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Trois fois par an[cut]',
'[comma]Quadrimestrielle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]trois fois par décade[cut]',
'[comma]Tridécadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Trois fois par décade, les 3, 6 et
9[cut]', '[comma]Tridécadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Trois fois par mois[cut]',
'[comma]Trimensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Trois fois par mois (?)[cut]',
'[comma]Trimensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Trois fois par mois (36 numéros formant 6
tomes par an)[cut]', '[comma]Trimensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Trois fois par mois (les 6, 16 et
26)[cut]', '[comma]Trimensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Trois fois par mois ?[cut]',
'[comma]Trimensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]trois fois par semaine[cut]',
'[comma]Trihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Trois fois par semaine (les lundi,
mercredi et vendredi)[cut]', '[comma]Trihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Trois fois par semaine (les mardi, jeudi
et samedi)[cut]', '[comma]Trihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Trois fois par semaine (lundi, mercredi
et vendredi)[cut]', '[comma]Trihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Trois fois par semaine (lundi, mercredi,
vendredi)[cut]', '[comma]Trihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Trois fois par semaine (mardi, jeudi et
samedi)[cut]', '[comma]Trihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]trois fois par semaine (mardi, jeudi et
samedi) à partir du 16 décembre 1783[cut]', '[comma]Trihebdomadaire[comma]',
$split_326a);
        $split_326a = str_ireplace('[cut]Trois fois par semaine, les lundi,
mercredi et vendredi[cut]', '[comma]Trihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]trois fois par semaine, les lundis,
mercredis et vendredis[cut]', '[comma]Trihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Trois fois par semaine, les mardi, jeudi
et samedi[cut]', '[comma]Trihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Trois fois par semaine, les mardi, jeudi
et samedi, à partir du n° X[cut]', '[comma]Trihebdomadaire[comma]', $split_326a);
```

```
        $split_326a = str_ireplace('[cut]Trois fois par semaine, les mardis,
jeudis et samedis[cut]', '[comma]Trihebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Trois livraisons par mois[cut]',
'[comma]Trimensuelle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]trois ou quatre fois par semaine[cut]',
'[comma]Autre / Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]un cahier tous les dix jours[cut]',
'[comma]Décadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]un cahier tous les huit jours[cut]',
'[comma]Autre / Irrégulière / Variable / Inconnue[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]un vol. tous les deux mois[cut]',
'[comma]Bimestrielle[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]une feuille par semaine (avertissement à
la fin du t. V)[cut]', '[comma]Hebdomadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]Une fois par décade[cut]',
'[comma]Décadaire[comma]', $split_326a);
        $split_326a = str_ireplace('[cut]une ou deux fois par semaine[cut]',
'[comma]Autre / Irrégulière / Variable / Inconnue[comma]', $split_326a);

        $split_326a = str_replace('[cut]', '', $split_326a);

        //Replace the periodicity phrase by the corresponding term ID in the
"Periodicity" taxonomy.

        $split_326a = str_replace('[comma]Décennale[comma]', ',94,', $split_326a);
        $split_326a = str_replace('[comma]Décennale (irrégulière)[comma]', ',95,',
$split_326a);
        $split_326a = str_replace('[comma]Quinquennale[comma]', ',281,',
$split_326a);
        $split_326a = str_replace('[comma]Quinquennale (irrégulière)[comma]',
',282,', $split_326a);
        $split_326a = str_replace('[comma]Quadriennale[comma]', ',283,',
$split_326a);
        $split_326a = str_replace('[comma]Quadriennale (irrégulière)[comma]',
',284,', $split_326a);
        $split_326a = str_replace('[comma]Triennale[comma]', ',285,',
$split_326a);
        $split_326a = str_replace('[comma]Triennale (irrégulière)[comma]',
',286,', $split_326a);
        $split_326a = str_replace('[comma]Biennale[comma]', ',96,', $split_326a);
        $split_326a = str_replace('[comma]Biennale (irrégulière)[comma]', ',97,',
$split_326a);
        $split_326a = str_replace('[comma]Annuelle[comma]', ',98,', $split_326a);
        $split_326a = str_replace('[comma]Annuelle (irrégulière)[comma]', ',99,',
$split_326a);
        $split_326a = str_replace('[comma]Semestrielle[comma]', ',100,',
$split_326a);
        $split_326a = str_replace('[comma]Semestrielle (irrégulière)[comma]',
',101,', $split_326a);
        $split_326a = str_replace('[comma]Quadrimestrielle[comma]', ',102,',
$split_326a);
        $split_326a = str_replace('[comma]Quadrimestrielle (irrégulière)[comma]',
',103,', $split_326a);
        $split_326a = str_replace('[comma]Trimestrielle[comma]', ',104,',
$split_326a);
        $split_326a = str_replace('[comma]Trimestrielle (irrégulière)[comma]',
',105,', $split_326a);
        $split_326a = str_replace('[comma]Bimestrielle[comma]', ',106,',
$split_326a);
```

```
        $split_326a = str_replace('[comma]Bimestrielle (irrégulière)[comma]',
',107,', $split_326a);
        $split_326a = str_replace('[comma]Mensuelle[comma]', ',108,',
$split_326a);
        $split_326a = str_replace('[comma]Mensuelle (irrégulière)[comma]',
',109,', $split_326a);
        $split_326a = str_replace('[comma]Bimensuelle[comma]', ',110,',
$split_326a);
        $split_326a = str_replace('[comma]Bimensuelle (irrégulière)[comma]',
',111,', $split_326a);
        $split_326a = str_replace('[comma]Trimensuelle[comma]', ',287,',
$split_326a);
        $split_326a = str_replace('[comma]Trimensuelle (irrégulière)[comma]',
',288,', $split_326a);
        $split_326a = str_replace('[comma]Décadaire[comma]', ',112,',
$split_326a);
        $split_326a = str_replace('[comma]Décadaire (irrégulière)[comma]',
',125,', $split_326a);
        $split_326a = str_replace('[comma]Hebdomadaire[comma]', ',114,',
$split_326a);
        $split_326a = str_replace('[comma]Hebdomadaire (irrégulière)[comma]',
',115,', $split_326a);
        $split_326a = str_replace('[comma]Bidécadaire[comma]', ',113,',
$split_326a);
        $split_326a = str_replace('[comma]Bidécadaire (irrégulière)[comma]',
',126,', $split_326a);
        $split_326a = str_replace('[comma]Tous les quatre jours[comma]', ',289,',
$split_326a);
        $split_326a = str_replace('[comma]Tous les quatre jours
(irrégulière)[comma]', ',290,', $split_326a);
        $split_326a = str_replace('[comma]Bihebdomadaire[comma]', ',116,',
$split_326a);
        $split_326a = str_replace('[comma]Bihebdomadaire (irrégulière)[comma]',
',117,', $split_326a);
        $split_326a = str_replace('[comma]Tridécadaire[comma]', ',291,',
$split_326a);
        $split_326a = str_replace('[comma]Tridécadaire (irrégulière)[comma]',
',292,', $split_326a);
        $split_326a = str_replace('[comma]Trihebdomadaire[comma]', ',118,',
$split_326a);
        $split_326a = str_replace('[comma]Trihebdomadaire (irrégulière)[comma]',
',119,', $split_326a);
        $split_326a = str_replace('[comma]Tous les trois jours[comma]', ',293,',
$split_326a);
        $split_326a = str_replace('[comma]Tous les trois jours
(irrégulière)[comma]', ',294,', $split_326a);
        $split_326a = str_replace('[comma]Tous les deux jours[comma]', ',295,',
$split_326a);
        $split_326a = str_replace('[comma]Tous les deux jours
(irrégulière)[comma]', ',296,', $split_326a);
        $split_326a = str_replace('[comma]Quotidienne[comma]', ',120,',
$split_326a);
        $split_326a = str_replace('[comma]Quotidienne (irrégulière)[comma]',
',121,', $split_326a);
        $split_326a = str_replace('[comma]Biquotidienne[comma]', ',122,',
$split_326a);
        $split_326a = str_replace('[comma]Biquotidienne (irrégulière)[comma]',
',123,', $split_326a);
```

```
        $split_326a = str_replace('[comma]Autre / Irrégulière / Variable /
Inconnue[comma]', ',124,', $split_326a);

        $wor_periodicity = str_replace(',,', ',', $split_326a);

        if(str_starts_with($wor_periodicity, ',')){
            $wor_periodicity = substr($wor_periodicity, 1);
        }
        if(str_ends_with($wor_periodicity, ',')){
            $wor_periodicity = substr($wor_periodicity, 0, -1);
        }

        $wor_periodicity = preg_replace('/[^0-9,]/', '', $wor_periodicity);

        if(stripos($wor_periodicity, ',') !== false){
            $wor_periodicity = '"'.$wor_periodicity.'"';
        }

//      Create field wor_subject from 606$a and 607$a.

        $split_606a = explode('606$a', $data);
        $split_606a = explode ($split_606a[0].'606$a', $data);
        $split_606a = str_replace('606$a', '[comma]', $split_606a[1]);
        $split_606a = preg_split('/(\d){3}\$./', $split_606a);

        $split_606a = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_606a[0]);

        $split_606a = explode('[comma]', $split_606a);

        if($split_606a[0] == $split_606a[1]){
            $split_606a = $split_606a[0];
        }
        else{
            $split_606a = implode('[comma]', $split_606a);
        }

        $split_606a = str_replace(',', '%2C', $split_606a); //Escape original
commas in URI.

        if(substr_count($split_606a, '"') === 1){
            $split_606a = str_replace('"', '', $split_606a);
        }

        if(stripos($split_606a, '[') !== false && stripos($split_606a, ']') ===
false){
            $split_606a = str_replace('[', '', $split_606a);
        }

        if(stripos($split_606a, ']') !== false && stripos($split_606a, '[') ===
false){
            $split_606a = str_replace(']', '', $split_606a);
        }

        if(stripos($split_606a, '(') !== false && stripos($split_606a, ')') ===
false){
            $split_606a = str_replace('(', '', $split_606a);
        }
```

```
        if(stripos($split_606a, ')') !== false && stripos($split_606a, '(') ===
false){
            $split_606a = str_replace(')', '', $split_606a);
        }

        $split_607a = explode('607$a', $data);
        $split_607a = explode ($split_607a[0].'607$a', $data);
        $split_607a = str_replace('607$a', '[comma]', $split_607a[1]);
        $split_607a = preg_split('/(\d){3}\$./', $split_607a);

        $split_607a = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_607a[0]);

        $split_607a = explode('[comma]', $split_607a);

        if($split_607a[0] == $split_607a[1]){
            $split_607a = $split_607a[0];
        }
        else{
            $split_607a = implode('[comma]', $split_607a);
        }

        $split_607a = str_replace(',', '%2C', $split_607a); //Escape original
commas in URI.

        if(substr_count($split_607a, '"') === 1){
            $split_607a = str_replace('"', '', $split_607a);
        }

        if(stripos($split_607a, '[') !== false && stripos($split_607a, ']') ===
false){
            $split_607a = str_replace('[', '', $split_607a);
        }

        if(stripos($split_607a, ']') !== false && stripos($split_607a, '[') ===
false){
            $split_607a = str_replace(']', '', $split_607a);
        }

        if(stripos($split_607a, '(') !== false && stripos($split_607a, ')') ===
false){
            $split_607a = str_replace('(', '', $split_607a);
        }

        if(stripos($split_607a, ')') !== false && stripos($split_607a, '(') ===
false){
            $split_607a = str_replace(')', '', $split_607a);
        }

        if($split_606a != '' && $split_607a != ''){
            $wor_subject = $split_606a.'[comma]'.$split_607a;
        }
        else{
            $wor_subject = $split_606a.$split_607a;
        }

        $wor_subject = str_replace('%2C', ',', $wor_subject);

        $wor_subject = explode('[comma]', $wor_subject);
```

```
//      At this point, we need to transform subject phrases into the IDs of the
corresponding taxonomy terms that have already been loaded into the Drupal
database. To accomplish this, we have extracted the wor_subject values along with
their corresponding IDs from the Drupal database. These values have been compiled
into an array on a separate PHP page (see documentation). With the help of this
array, we will be able to make the wor_subject field usable with the Drupal
Webform Import module.

        include 'wor_subject_array.php';

        if(array_key_exists($wor_subject[0], $wor_subject_array)){
            $wor_subject[0] = $wor_subject_array[$wor_subject[0]];
        }
        else{
            $wor_subject[0] = '';
        }

        if(array_key_exists($wor_subject[1], $wor_subject_array)){
            $wor_subject[1] = $wor_subject_array[$wor_subject[1]];
        }
        else{
            $wor_subject[1] = '';
        }

        if(array_key_exists($wor_subject[2], $wor_subject_array)){
            $wor_subject[2] = $wor_subject_array[$wor_subject[2]];
        }
        else{
            $wor_subject[2] = '';
        }

        if(array_key_exists($wor_subject[3], $wor_subject_array)){
            $wor_subject[3] = $wor_subject_array[$wor_subject[3]];
        }
        else{
            $wor_subject[3] = '';
        }

        if(array_key_exists($wor_subject[4], $wor_subject_array)){
            $wor_subject[4] = $wor_subject_array[$wor_subject[4]];
        }
        else{
            $wor_subject[4] = '';
        }

        if(array_key_exists($wor_subject[5], $wor_subject_array)){
            $wor_subject[5] = $wor_subject_array[$wor_subject[5]];
        }
        else{
            $wor_subject[5] = '';
        }

        if(array_key_exists($wor_subject[6], $wor_subject_array)){
            $wor_subject[6] = $wor_subject_array[$wor_subject[6]];
        }
        else{
            $wor_subject[6] = '';
        }
```

```
if(array_key_exists($wor_subject[7], $wor_subject_array)){
    $wor_subject[7] = $wor_subject_array[$wor_subject[7]];
}
else{
    $wor_subject[7] = '';
}

if(array_key_exists($wor_subject[8], $wor_subject_array)){
    $wor_subject[8] = $wor_subject_array[$wor_subject[8]];
}
else{
    $wor_subject[8] = '';
}

if(array_key_exists($wor_subject[9], $wor_subject_array)){
    $wor_subject[9] = $wor_subject_array[$wor_subject[9]];
}
else{
    $wor_subject[9] = '';
}

if(array_key_exists($wor_subject[10], $wor_subject_array)){
    $wor_subject[10] = $wor_subject_array[$wor_subject[10]];
}
else{
    $wor_subject[10] = '';
}

if(array_key_exists($wor_subject[11], $wor_subject_array)){
    $wor_subject[11] = $wor_subject_array[$wor_subject[11]];
}
else{
    $wor_subject[11] = '';
}

if(array_key_exists($wor_subject[12], $wor_subject_array)){
    $wor_subject[12] = $wor_subject_array[$wor_subject[12]];
}
else{
    $wor_subject[12] = '';
}

if(array_key_exists($wor_subject[13], $wor_subject_array)){
    $wor_subject[13] = $wor_subject_array[$wor_subject[13]];
}
else{
    $wor_subject[13] = '';
}

if(array_key_exists($wor_subject[14], $wor_subject_array)){
    $wor_subject[14] = $wor_subject_array[$wor_subject[14]];
}
else{
    $wor_subject[14] = '';
}

if(array_key_exists($wor_subject[15], $wor_subject_array)){
    $wor_subject[15] = $wor_subject_array[$wor_subject[15]];
```

```
        }
        else{
            $wor_subject[15] = '';
        }

        if(array_key_exists($wor_subject[16], $wor_subject_array)){
            $wor_subject[16] = $wor_subject_array[$wor_subject[16]];
        }
        else{
            $wor_subject[16] = '';
        }

        if(array_key_exists($wor_subject[17], $wor_subject_array)){
            $wor_subject[17] = $wor_subject_array[$wor_subject[17]];
        }
        else{
            $wor_subject[17] = '';
        }

        if(array_key_exists($wor_subject[18], $wor_subject_array)){
            $wor_subject[18] = $wor_subject_array[$wor_subject[18]];
        }
        else{
            $wor_subject[18] = '';
        }

        if(array_key_exists($wor_subject[19], $wor_subject_array)){
            $wor_subject[19] = $wor_subject_array[$wor_subject[19]];
        }
        else{
            $wor_subject[19] = '';
        }

        if(array_key_exists($wor_subject[20], $wor_subject_array)){
            $wor_subject[20] = $wor_subject_array[$wor_subject[20]];
        }
        else{
            $wor_subject[20] = '';
        }

        if(array_key_exists($wor_subject[21], $wor_subject_array)){
            $wor_subject[21] = $wor_subject_array[$wor_subject[21]];
        }
        else{
            $wor_subject[21] = '';
        }

        if(array_key_exists($wor_subject[22], $wor_subject_array)){
            $wor_subject[22] = $wor_subject_array[$wor_subject[22]];
        }
        else{
            $wor_subject[22] = '';
        }

        if(array_key_exists($wor_subject[23], $wor_subject_array)){
            $wor_subject[23] = $wor_subject_array[$wor_subject[23]];
        }
        else{
            $wor_subject[23] = '';
```

```
    }

    if(array_key_exists($wor_subject[24], $wor_subject_array)){
        $wor_subject[24] = $wor_subject_array[$wor_subject[24]];
    }
    else{
        $wor_subject[24] = '';
    }

    if(array_key_exists($wor_subject[25], $wor_subject_array)){
        $wor_subject[25] = $wor_subject_array[$wor_subject[25]];
    }
    else{
        $wor_subject[25] = '';
    }

    if(array_key_exists($wor_subject[26], $wor_subject_array)){
        $wor_subject[26] = $wor_subject_array[$wor_subject[26]];
    }
    else{
        $wor_subject[26] = '';
    }

    if(array_key_exists($wor_subject[27], $wor_subject_array)){
        $wor_subject[27] = $wor_subject_array[$wor_subject[27]];
    }
    else{
        $wor_subject[27] = '';
    }

    if(array_key_exists($wor_subject[28], $wor_subject_array)){
        $wor_subject[28] = $wor_subject_array[$wor_subject[28]];
    }
    else{
        $wor_subject[28] = '';
    }

    if(array_key_exists($wor_subject[29], $wor_subject_array)){
        $wor_subject[29] = $wor_subject_array[$wor_subject[29]];
    }
    else{
        $wor_subject[29] = '';
    }

    if(array_key_exists($wor_subject[30], $wor_subject_array)){
        $wor_subject[30] = $wor_subject_array[$wor_subject[30]];
    }
    else{
        $wor_subject[30] = '';
    }

    if(array_key_exists($wor_subject[31], $wor_subject_array)){
        $wor_subject[31] = $wor_subject_array[$wor_subject[31]];
    }
    else{
        $wor_subject[31] = '';
    }

    if(array_key_exists($wor_subject[32], $wor_subject_array)){
```

```
            $wor_subject[32] = $wor_subject_array[$wor_subject[32]];
        }
        else{
            $wor_subject[32] = '';
        }

        if(array_key_exists($wor_subject[33], $wor_subject_array)){
            $wor_subject[33] = $wor_subject_array[$wor_subject[33]];
        }
        else{
            $wor_subject[33] = '';
        }

        if(array_key_exists($wor_subject[34], $wor_subject_array)){
            $wor_subject[34] = $wor_subject_array[$wor_subject[34]];
        }
        else{
            $wor_subject[34] = '';
        }

        if(array_key_exists($wor_subject[35], $wor_subject_array)){
            $wor_subject[35] = $wor_subject_array[$wor_subject[35]];
        }
        else{
            $wor_subject[35] = '';
        }

        if(array_key_exists($wor_subject[36], $wor_subject_array)){
            $wor_subject[36] = $wor_subject_array[$wor_subject[36]];
        }
        else{
            $wor_subject[36] = '';
        }

        if(array_key_exists($wor_subject[37], $wor_subject_array)){
            $wor_subject[37] = $wor_subject_array[$wor_subject[37]];
        }
        else{
            $wor_subject[37] = '';
        }

        if(array_key_exists($wor_subject[38], $wor_subject_array)){
            $wor_subject[38] = $wor_subject_array[$wor_subject[38]];
        }
        else{
            $wor_subject[38] = '';
        }

        if(array_key_exists($wor_subject[39], $wor_subject_array)){
            $wor_subject[39] = $wor_subject_array[$wor_subject[39]];
        }
        else{
            $wor_subject[39] = '';
        }

        if(array_key_exists($wor_subject[40], $wor_subject_array)){
            $wor_subject[40] = $wor_subject_array[$wor_subject[40]];
        }
        else{
```

```
        $wor_subject[40] = '';
    }

    if(array_key_exists($wor_subject[41], $wor_subject_array)){
        $wor_subject[41] = $wor_subject_array[$wor_subject[41]];
    }
    else{
        $wor_subject[41] = '';
    }

    if(array_key_exists($wor_subject[42], $wor_subject_array)){
        $wor_subject[42] = $wor_subject_array[$wor_subject[42]];
    }
    else{
        $wor_subject[42] = '';
    }

    if(array_key_exists($wor_subject[43], $wor_subject_array)){
        $wor_subject[43] = $wor_subject_array[$wor_subject[43]];
    }
    else{
        $wor_subject[43] = '';
    }

    if(array_key_exists($wor_subject[44], $wor_subject_array)){
        $wor_subject[44] = $wor_subject_array[$wor_subject[44]];
    }
    else{
        $wor_subject[44] = '';
    }

    if(array_key_exists($wor_subject[45], $wor_subject_array)){
        $wor_subject[45] = $wor_subject_array[$wor_subject[45]];
    }
    else{
        $wor_subject[45] = '';
    }

    if(array_key_exists($wor_subject[46], $wor_subject_array)){
        $wor_subject[46] = $wor_subject_array[$wor_subject[46]];
    }
    else{
        $wor_subject[46] = '';
    }

    if(array_key_exists($wor_subject[47], $wor_subject_array)){
        $wor_subject[47] = $wor_subject_array[$wor_subject[47]];
    }
    else{
        $wor_subject[47] = '';
    }

    if(array_key_exists($wor_subject[48], $wor_subject_array)){
        $wor_subject[48] = $wor_subject_array[$wor_subject[48]];
    }
    else{
        $wor_subject[48] = '';
    }
```

```
        if(array_key_exists($wor_subject[49], $wor_subject_array)){
            $wor_subject[49] = $wor_subject_array[$wor_subject[49]];
        }
        else{
            $wor_subject[49] = '';
        }

        if(array_key_exists($wor_subject[50], $wor_subject_array)){
            $wor_subject[50] = $wor_subject_array[$wor_subject[50]];
        }
        else{
            $wor_subject[50] = '';
        }

        $wor_subject = implode(',', $wor_subject);

        $wor_subject = preg_replace('/,{2,}/', ',', $wor_subject);

        if(str_ends_with($wor_subject, ',')){
            $wor_subject = substr($wor_subject, 0, -1);
        }

        if(str_starts_with($wor_subject, ',')){
            $wor_subject = substr($wor_subject, 1);
        }

        if(stripos($wor_subject, ',')){
            $wor_subject = '"'.$wor_subject.'"';
        }


//      Create field wor_bnf_record.

        $wor_bnf_record = 'https://catalogue.bnf.fr/'.$split[0];

//      Create field wor_bnf_record_number.

        $wor_bnf_record_number = $split[1];

//      Create field wor_issn from 022$a.

        $split_022a = explode('022$a', $data);
        $split_022a = explode ($split_022a[0].'022$a', $data);
        $split_022a = str_replace('022$a', '[semicolon]', $split_022a[1]);
        $split_022a = preg_split('/(\d){3}\$./', $split_022a);

        $split_022a = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_022a[0]);

        $split_022a = explode('[semicolon]', $split_022a);

        if($split_022a[0] == $split_022a[1]){
            $split_022a = $split_022a[0];
        }
        else{
            $split_022a = implode('[semicolon]', $split_022a);
        }

        $wor_issn = str_replace('[semicolon]', ' ; ', $split_022a);
```

```
//      Create field wor_data_origin_bnf.

        $wor_data_origin_bnf = '1';

//      Create field all_remark_bnf from 260$d, 261$d (prepared above) and 300$a.

        if($all_remark_bnf != $wor_date.'-'.$wor_end_date && $all_remark_bnf !=
$wor_date){

            if(substr_count($all_remark_bnf, '"') === 1){
            $all_remark_bnf = str_replace('"', '', $all_remark_bnf);
            }

            if(stripos($all_remark_bnf, '[') !== false && stripos($all_remark_bnf,
']') === false){
                $all_remark_bnf = str_replace('[', '', $all_remark_bnf);
            }

            if(stripos($all_remark_bnf, ']') !== false && stripos($all_remark_bnf,
'[') === false){
                $all_remark_bnf = str_replace(']', '', $all_remark_bnf);
            }

            if(stripos($all_remark_bnf, '(') !== false && stripos($all_remark_bnf,
')') === false){
                $all_remark_bnf = str_replace('(', '', $all_remark_bnf);
            }

            if(stripos($all_remark_bnf, ')') !== false && stripos($all_remark_bnf,
'(') === false){
                $all_remark_bnf = str_replace(')', '', $all_remark_bnf);
            }

            if(str_ends_with($all_remark_bnf, '.') ||
str_ends_with($all_remark_bnf, '.]') || str_ends_with($all_remark_bnf, '.)' ||
str_ends_with($all_remark_bnf, '."') || str_ends_with($all_remark_bnf, '!') ||
str_ends_with($all_remark_bnf, '?'))){
            }
            else{
                $all_remark_bnf = $all_remark_bnf.'.';
            }
            $all_remark_bnf = 'Date : '.$all_remark_bnf;
        }
        else{
            $all_remark_bnf = '';
        }

        $split_300a = explode('300$a', $data);
        $split_300a = explode ($split_300a[0].'300$a', $data);
        $split_300a = str_replace('300$a', '[slash]', $split_300a[1]);
        $split_300a = preg_split('/(\d){3}\$./', $split_300a);

        $split_300a = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_300a[0]);

        $split_300a = explode('[slash]', $split_300a);

        if($split_300a[0] == $split_300a[1]){
```

```
            $split_300a = $split_300a[0];
        }
        else{
            $split_300a = implode('[slash]', $split_300a);
        }

        $split_300a = str_replace('[slash]', ' / ', $split_300a);

        $split_300a = ucfirst($split_300a);

        if(substr_count($split_300a, '"') === 1){
            $split_300a = str_replace('"', '', $split_300a);
        }

        if(stripos($split_300a, '[') !== false && stripos($split_300a, ']') ===
false){
            $split_300a = str_replace('[', '', $split_300a);
        }

        if(stripos($split_300a, ']') !== false && stripos($split_300a, '[') ===
false){
            $split_300a = str_replace(']', '', $split_300a);
        }

        if(stripos($split_300a, '(') !== false && stripos($split_300a, ')') ===
false){
            $split_300a = str_replace('(', '', $split_300a);
        }

        if(stripos($split_300a, ')') !== false && stripos($split_300a, '(') ===
false){
            $split_300a = str_replace(')', '', $split_300a);
        }

        if(str_ends_with($split_300a, '.') || str_ends_with($split_300a, '.]') ||
str_ends_with($split_300a, '.)') || str_ends_with($split_300a, '."') ||
str_ends_with($split_300a, '!') || str_ends_with($split_300a, '?')){
        }
        elseif($split_300a != ''){
            $split_300a = $split_300a.'.';
        }

        if($all_remark_bnf != '' && $split_300a != ''){
            $all_remark_bnf = $all_remark_bnf.' / '.$split_300a;
        }
        elseif($all_remark_bnf != ''){
        }
        else{
            $all_remark_bnf = $split_300a;
        }

        $all_remark_bnf = preg_replace('/^"{3,}/', '"', $all_remark_bnf);
        $all_remark_bnf = preg_replace('/"{3,}$/', '"', $all_remark_bnf);
        $all_remark_bnf = str_replace('""""', '"', $all_remark_bnf);
        $all_remark_bnf = str_replace('""""', '"', $all_remark_bnf);

        if(stripos($all_remark_bnf, '"') !== false || stripos($all_remark_bnf,
',') !== false){
            $all_remark_bnf = '"'.$all_remark_bnf.'"';
```

```
        }

//      To establish relationships between works and persons, it is necessary to
identify the right node ID (nid). For this purpose, we have extracted the
per_bnf_record_number values along with their corresponding NIDs from the Drupal
database. We have compiled these values into an array on a separate PHP page (see
documentation). With the help of this array, we will be able to create four
relationships: wor_has_author_per, wor_has_publisher_per,
wor_has_other_creator_per and wor_refers_to_per.

        include 'per_bnf_record_number_array.php';

//      Create field wor_has_author_per from 100$3, 110$3, 700$3, 710$3.

        $split_100_3 = explode('100$3', $data);
        $split_100_3 = explode ($split_100_3[0].'100$3', $data);
        $split_100_3 = str_replace('100$3', '[comma]', $split_100_3[1]);
        $split_100_3 = preg_split('/(\d){3}\$./', $split_100_3);

        $split_100_3 = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_100_3[0]);

        $split_100_3 = explode('[comma]', $split_100_3);

        if($split_100_3[0] == $split_100_3[1]){
            $split_100_3 = $split_100_3[0];
        }
        else{
            $split_100_3 = implode('[comma]', $split_100_3);
        }

        $split_100_3 = str_replace(',', '%2C', $split_100_3); //Escape original
commas in URI.
        $split_100_3 = str_replace('[comma]', ',', $split_100_3); //Multiple
values are separated by commas.

        $split_110_3 = explode('110$3', $data);
        $split_110_3 = explode ($split_110_3[0].'110$3', $data);
        $split_110_3 = str_replace('110$3', '[comma]', $split_110_3[1]);
        $split_110_3 = preg_split('/(\d){3}\$./', $split_110_3);

        $split_110_3 = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_110_3[0]);

        $split_110_3 = explode('[comma]', $split_110_3);

        if($split_110_3[0] == $split_110_3[1]){
            $split_110_3 = $split_110_3[0];
        }
        else{
            $split_110_3 = implode('[comma]', $split_110_3);
        }

        $split_110_3 = str_replace(',', '%2C', $split_110_3); //Escape original
commas in URI.
        $split_110_3 = str_replace('[comma]', ',', $split_110_3); //Multiple
values are separated by commas.

        $split_700_3 = explode('700$3', $data);
```

```
        $split_700_3 = explode ($split_700_3[0].'700$3', $data);
        $split_700_3 = str_replace('700$3', '[comma]', $split_700_3[1]);
        $split_700_3 = preg_split('/(\d){3}\$./', $split_700_3);

        $split_700_3 = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_700_3[0]);

        $split_700_3 = explode('[comma]', $split_700_3);

        if($split_700_3[0] == $split_700_3[1]){
            $split_700_3 = $split_700_3[0];
        }
        else{
            $split_700_3 = implode('[comma]', $split_700_3);
        }

        $split_700_3 = str_replace(',', '%2C', $split_700_3); //Escape original
commas in URI.
        $split_700_3 = str_replace('[comma]', ',', $split_700_3); //Multiple
values are separated by commas.

        $split_710_3 = explode('710$3', $data);
        $split_710_3 = explode ($split_710_3[0].'710$3', $data);
        $split_710_3 = str_replace('710$3', '[comma]', $split_710_3[1]);
        $split_710_3 = preg_split('/(\d){3}\$./', $split_710_3);

        $split_710_3 = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_710_3[0]);

        $split_710_3 = explode('[comma]', $split_710_3);

        if($split_710_3[0] == $split_710_3[1]){
            $split_710_3 = $split_710_3[0];
        }
        else{
            $split_710_3 = implode('[comma]', $split_710_3);
        }

        $split_710_3 = str_replace(',', '%2C', $split_710_3); //Escape original
commas in URI.
        $split_710_3 = str_replace('[comma]', ',', $split_710_3); //Multiple
values are separated by commas.

        $wor_has_author_per =
$split_100_3.','.$split_110_3.','.$split_700_3.','.$split_710_3;

        $wor_has_author_per = explode(',', $wor_has_author_per);

        if(array_key_exists($wor_has_author_per[0],
$per_bnf_record_number_array)){
            $wor_has_author_per[0] =
$per_bnf_record_number_array[$wor_has_author_per[0]];
        }
        else{
            $wor_has_author_per[0] = '';
        }

        if(array_key_exists($wor_has_author_per[1],
$per_bnf_record_number_array)){
```

```
            $wor_has_author_per[1] =
$per_bnf_record_number_array[$wor_has_author_per[1]];
        }
        else{
            $wor_has_author_per[1] = '';
        }

        if(array_key_exists($wor_has_author_per[2],
$per_bnf_record_number_array)){
            $wor_has_author_per[2] =
$per_bnf_record_number_array[$wor_has_author_per[2]];
        }
        else{
            $wor_has_author_per[2] = '';
        }

        if(array_key_exists($wor_has_author_per[3],
$per_bnf_record_number_array)){
            $wor_has_author_per[3] =
$per_bnf_record_number_array[$wor_has_author_per[3]];
        }
        else{
            $wor_has_author_per[3] = '';
        }

        if(array_key_exists($wor_has_author_per[4],
$per_bnf_record_number_array)){
            $wor_has_author_per[4] =
$per_bnf_record_number_array[$wor_has_author_per[4]];
        }
        else{
            $wor_has_author_per[4] = '';
        }

        if(array_key_exists($wor_has_author_per[5],
$per_bnf_record_number_array)){
            $wor_has_author_per[5] =
$per_bnf_record_number_array[$wor_has_author_per[5]];
        }
        else{
            $wor_has_author_per[5] = '';
        }

        if(array_key_exists($wor_has_author_per[6],
$per_bnf_record_number_array)){
            $wor_has_author_per[6] =
$per_bnf_record_number_array[$wor_has_author_per[6]];
        }
        else{
            $wor_has_author_per[6] = '';
        }

        if(array_key_exists($wor_has_author_per[7],
$per_bnf_record_number_array)){
            $wor_has_author_per[7] =
$per_bnf_record_number_array[$wor_has_author_per[7]];
        }
        else{
            $wor_has_author_per[7] = '';
```

```
        }

        if(array_key_exists($wor_has_author_per[8],
$per_bnf_record_number_array)){
            $wor_has_author_per[8] =
$per_bnf_record_number_array[$wor_has_author_per[8]];
        }
        else{
            $wor_has_author_per[8] = '';
        }

        if(array_key_exists($wor_has_author_per[9],
$per_bnf_record_number_array)){
            $wor_has_author_per[9] =
$per_bnf_record_number_array[$wor_has_author_per[9]];
        }
        else{
            $wor_has_author_per[9] = '';
        }

        if(array_key_exists($wor_has_author_per[10],
$per_bnf_record_number_array)){
            $wor_has_author_per[10] =
$per_bnf_record_number_array[$wor_has_author_per[10]];
        }
        else{
            $wor_has_author_per[10] = '';
        }

        if(array_key_exists($wor_has_author_per[11],
$per_bnf_record_number_array)){
            $wor_has_author_per[11] =
$per_bnf_record_number_array[$wor_has_author_per[11]];
        }
        else{
            $wor_has_author_per[11] = '';
        }

        if(array_key_exists($wor_has_author_per[12],
$per_bnf_record_number_array)){
            $wor_has_author_per[12] =
$per_bnf_record_number_array[$wor_has_author_per[12]];
        }
        else{
            $wor_has_author_per[12] = '';
        }

        if(array_key_exists($wor_has_author_per[13],
$per_bnf_record_number_array)){
            $wor_has_author_per[13] =
$per_bnf_record_number_array[$wor_has_author_per[13]];
        }
        else{
            $wor_has_author_per[13] = '';
        }

        if(array_key_exists($wor_has_author_per[14],
$per_bnf_record_number_array)){
```

```
            $wor_has_author_per[14] =
$per_bnf_record_number_array[$wor_has_author_per[14]];
        }
        else{
            $wor_has_author_per[14] = '';
        }

        if(array_key_exists($wor_has_author_per[15],
$per_bnf_record_number_array)){
            $wor_has_author_per[15] =
$per_bnf_record_number_array[$wor_has_author_per[15]];
        }
        else{
            $wor_has_author_per[15] = '';
        }

        if(array_key_exists($wor_has_author_per[16],
$per_bnf_record_number_array)){
            $wor_has_author_per[16] =
$per_bnf_record_number_array[$wor_has_author_per[16]];
        }
        else{
            $wor_has_author_per[16] = '';
        }

        if(array_key_exists($wor_has_author_per[17],
$per_bnf_record_number_array)){
            $wor_has_author_per[17] =
$per_bnf_record_number_array[$wor_has_author_per[17]];
        }
        else{
            $wor_has_author_per[17] = '';
        }

        if(array_key_exists($wor_has_author_per[18],
$per_bnf_record_number_array)){
            $wor_has_author_per[18] =
$per_bnf_record_number_array[$wor_has_author_per[18]];
        }
        else{
            $wor_has_author_per[18] = '';
        }

        if(array_key_exists($wor_has_author_per[19],
$per_bnf_record_number_array)){
            $wor_has_author_per[19] =
$per_bnf_record_number_array[$wor_has_author_per[19]];
        }
        else{
            $wor_has_author_per[19] = '';
        }

        if(array_key_exists($wor_has_author_per[20],
$per_bnf_record_number_array)){
            $wor_has_author_per[20] =
$per_bnf_record_number_array[$wor_has_author_per[20]];
        }
        else{
            $wor_has_author_per[20] = '';
```

```
        }

        if(array_key_exists($wor_has_author_per[21],
$per_bnf_record_number_array)){
            $wor_has_author_per[21] =
$per_bnf_record_number_array[$wor_has_author_per[21]];
        }
        else{
            $wor_has_author_per[21] = '';
        }

        if(array_key_exists($wor_has_author_per[22],
$per_bnf_record_number_array)){
            $wor_has_author_per[22] =
$per_bnf_record_number_array[$wor_has_author_per[22]];
        }
        else{
            $wor_has_author_per[22] = '';
        }

        if(array_key_exists($wor_has_author_per[23],
$per_bnf_record_number_array)){
            $wor_has_author_per[23] =
$per_bnf_record_number_array[$wor_has_author_per[23]];
        }
        else{
            $wor_has_author_per[23] = '';
        }

        if(array_key_exists($wor_has_author_per[24],
$per_bnf_record_number_array)){
            $wor_has_author_per[24] =
$per_bnf_record_number_array[$wor_has_author_per[24]];
        }
        else{
            $wor_has_author_per[24] = '';
        }

        if(array_key_exists($wor_has_author_per[25],
$per_bnf_record_number_array)){
            $wor_has_author_per[25] =
$per_bnf_record_number_array[$wor_has_author_per[25]];
        }
        else{
            $wor_has_author_per[25] = '';
        }

        if(array_key_exists($wor_has_author_per[26],
$per_bnf_record_number_array)){
            $wor_has_author_per[26] =
$per_bnf_record_number_array[$wor_has_author_per[26]];
        }
        else{
            $wor_has_author_per[26] = '';
        }

        if(array_key_exists($wor_has_author_per[27],
$per_bnf_record_number_array)){
```

```
            $wor_has_author_per[27] =
$per_bnf_record_number_array[$wor_has_author_per[27]];
        }
        else{
            $wor_has_author_per[27] = '';
        }

        if(array_key_exists($wor_has_author_per[28],
$per_bnf_record_number_array)){
            $wor_has_author_per[28] =
$per_bnf_record_number_array[$wor_has_author_per[28]];
        }
        else{
            $wor_has_author_per[28] = '';
        }

        if(array_key_exists($wor_has_author_per[29],
$per_bnf_record_number_array)){
            $wor_has_author_per[29] =
$per_bnf_record_number_array[$wor_has_author_per[29]];
        }
        else{
            $wor_has_author_per[29] = '';
        }

        if(array_key_exists($wor_has_author_per[30],
$per_bnf_record_number_array)){
            $wor_has_author_per[30] =
$per_bnf_record_number_array[$wor_has_author_per[30]];
        }
        else{
            $wor_has_author_per[30] = '';
        }

        if(array_key_exists($wor_has_author_per[31],
$per_bnf_record_number_array)){
            $wor_has_author_per[31] =
$per_bnf_record_number_array[$wor_has_author_per[31]];
        }
        else{
            $wor_has_author_per[31] = '';
        }

        if(array_key_exists($wor_has_author_per[32],
$per_bnf_record_number_array)){
            $wor_has_author_per[32] =
$per_bnf_record_number_array[$wor_has_author_per[32]];
        }
        else{
            $wor_has_author_per[32] = '';
        }

        if(array_key_exists($wor_has_author_per[33],
$per_bnf_record_number_array)){
            $wor_has_author_per[33] =
$per_bnf_record_number_array[$wor_has_author_per[33]];
        }
        else{
            $wor_has_author_per[33] = '';
```

```
        }

        if(array_key_exists($wor_has_author_per[34],
$per_bnf_record_number_array)){
            $wor_has_author_per[34] =
$per_bnf_record_number_array[$wor_has_author_per[34]];
        }
        else{
            $wor_has_author_per[34] = '';
        }

        if(array_key_exists($wor_has_author_per[35],
$per_bnf_record_number_array)){
            $wor_has_author_per[35] =
$per_bnf_record_number_array[$wor_has_author_per[35]];
        }
        else{
            $wor_has_author_per[35] = '';
        }

        if(array_key_exists($wor_has_author_per[36],
$per_bnf_record_number_array)){
            $wor_has_author_per[36] =
$per_bnf_record_number_array[$wor_has_author_per[36]];
        }
        else{
            $wor_has_author_per[36] = '';
        }

        if(array_key_exists($wor_has_author_per[37],
$per_bnf_record_number_array)){
            $wor_has_author_per[37] =
$per_bnf_record_number_array[$wor_has_author_per[37]];
        }
        else{
            $wor_has_author_per[37] = '';
        }

        if(array_key_exists($wor_has_author_per[38],
$per_bnf_record_number_array)){
            $wor_has_author_per[38] =
$per_bnf_record_number_array[$wor_has_author_per[38]];
        }
        else{
            $wor_has_author_per[38] = '';
        }

        if(array_key_exists($wor_has_author_per[39],
$per_bnf_record_number_array)){
            $wor_has_author_per[39] =
$per_bnf_record_number_array[$wor_has_author_per[39]];
        }
        else{
            $wor_has_author_per[39] = '';
        }

        if(array_key_exists($wor_has_author_per[40],
$per_bnf_record_number_array)){
```

```
            $wor_has_author_per[40] =
$per_bnf_record_number_array[$wor_has_author_per[40]];
        }
        else{
            $wor_has_author_per[40] = '';
        }

        if(array_key_exists($wor_has_author_per[41],
$per_bnf_record_number_array)){
            $wor_has_author_per[41] =
$per_bnf_record_number_array[$wor_has_author_per[41]];
        }
        else{
            $wor_has_author_per[41] = '';
        }

        if(array_key_exists($wor_has_author_per[42],
$per_bnf_record_number_array)){
            $wor_has_author_per[42] =
$per_bnf_record_number_array[$wor_has_author_per[42]];
        }
        else{
            $wor_has_author_per[42] = '';
        }

        if(array_key_exists($wor_has_author_per[43],
$per_bnf_record_number_array)){
            $wor_has_author_per[43] =
$per_bnf_record_number_array[$wor_has_author_per[43]];
        }
        else{
            $wor_has_author_per[43] = '';
        }

        if(array_key_exists($wor_has_author_per[44],
$per_bnf_record_number_array)){
            $wor_has_author_per[44] =
$per_bnf_record_number_array[$wor_has_author_per[44]];
        }
        else{
            $wor_has_author_per[44] = '';
        }

        if(array_key_exists($wor_has_author_per[45],
$per_bnf_record_number_array)){
            $wor_has_author_per[45] =
$per_bnf_record_number_array[$wor_has_author_per[45]];
        }
        else{
            $wor_has_author_per[45] = '';
        }

        if(array_key_exists($wor_has_author_per[46],
$per_bnf_record_number_array)){
            $wor_has_author_per[46] =
$per_bnf_record_number_array[$wor_has_author_per[46]];
        }
        else{
            $wor_has_author_per[46] = '';
```

```
        }

        if(array_key_exists($wor_has_author_per[47],
$per_bnf_record_number_array)){
            $wor_has_author_per[47] =
$per_bnf_record_number_array[$wor_has_author_per[47]];
        }
        else{
            $wor_has_author_per[47] = '';
        }

        if(array_key_exists($wor_has_author_per[48],
$per_bnf_record_number_array)){
            $wor_has_author_per[48] =
$per_bnf_record_number_array[$wor_has_author_per[48]];
        }
        else{
            $wor_has_author_per[48] = '';
        }

        if(array_key_exists($wor_has_author_per[49],
$per_bnf_record_number_array)){
            $wor_has_author_per[49] =
$per_bnf_record_number_array[$wor_has_author_per[49]];
        }
        else{
            $wor_has_author_per[49] = '';
        }

        if($wor_has_author_per[50] !=''){
            $all_remark_verified = 'verified';
        }

        $wor_has_author_per = implode(',', $wor_has_author_per);

        $wor_has_author_per = preg_replace('/,{2,}/', ',', $wor_has_author_per);

        if(str_ends_with($wor_has_author_per, ',')){
            $wor_has_author_per = substr($wor_has_author_per, 0, -1);
        }

        if(str_starts_with($wor_has_author_per, ',')){
            $wor_has_author_per = substr($wor_has_author_per, 1);
        }

        if(stripos($wor_has_author_per, ',')){
            $wor_has_author_per = '"'.$wor_has_author_per.'"';
        }

//      Create field wor_has_publisher_person from 720$3, 721$3, 727$3, 730$3,
731$3, 737$3.

        $split_720_3 = explode('720$3', $data);
        $split_720_3 = explode ($split_720_3[0].'720$3', $data);
        $split_720_3 = str_replace('720$3', '[comma]', $split_720_3[1]);
        $split_720_3 = preg_split('/(\d){3}\$./', $split_720_3);

        $split_720_3 = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_720_3[0]);
```

```
        $split_720_3 = explode('[comma]', $split_720_3);

        if($split_720_3[0] == $split_720_3[1]){
            $split_720_3 = $split_720_3[0];
        }
        else{
            $split_720_3 = implode('[comma]', $split_720_3);
        }

        $split_720_3 = str_replace(',', '%2C', $split_720_3); //Escape original
commas in URI.
        $split_720_3 = str_replace('[comma]', ',', $split_720_3); //Multiple
values are separated by commas.

        $split_721_3 = explode('721$3', $data);
        $split_721_3 = explode ($split_721_3[0].'721$3', $data);
        $split_721_3 = str_replace('721$3', '[comma]', $split_721_3[1]);
        $split_721_3 = preg_split('/(\d){3}\$./', $split_721_3);

        $split_721_3 = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_721_3[0]);

        $split_721_3 = explode('[comma]', $split_721_3);

        if($split_721_3[0] == $split_721_3[1]){
            $split_721_3 = $split_721_3[0];
        }
        else{
            $split_721_3 = implode('[comma]', $split_721_3);
        }

        $split_721_3 = str_replace(',', '%2C', $split_721_3); //Escape original
commas in URI.
        $split_721_3 = str_replace('[comma]', ',', $split_721_3); //Multiple
values are separated by commas.

        $split_727_3 = explode('727$3', $data);
        $split_727_3 = explode ($split_727_3[0].'727$3', $data);
        $split_727_3 = str_replace('727$3', '[comma]', $split_727_3[1]);
        $split_727_3 = preg_split('/(\d){3}\$./', $split_727_3);

        $split_727_3 = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_727_3[0]);

        $split_727_3 = explode('[comma]', $split_727_3);

        if($split_727_3[0] == $split_727_3[1]){
            $split_727_3 = $split_727_3[0];
        }
        else{
            $split_727_3 = implode('[comma]', $split_727_3);
        }

        $split_727_3 = str_replace(',', '%2C', $split_727_3); //Escape original
commas in URI.
        $split_727_3 = str_replace('[comma]', ',', $split_727_3); //Multiple
values are separated by commas.
```

```
        $split_730_3 = explode('730$3', $data);
        $split_730_3 = explode ($split_730_3[0].'730$3', $data);
        $split_730_3 = str_replace('730$3', '[comma]', $split_730_3[1]);
        $split_730_3 = preg_split('/(\d){3}\$./', $split_730_3);

        $split_730_3 = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_730_3[0]);

        $split_730_3 = explode('[comma]', $split_730_3);

        if($split_730_3[0] == $split_730_3[1]){
            $split_730_3 = $split_730_3[0];
        }
        else{
            $split_730_3 = implode('[comma]', $split_730_3);
        }

        $split_730_3 = str_replace(',', '%2C', $split_730_3); //Escape original
commas in URI.
        $split_730_3 = str_replace('[comma]', ',', $split_730_3); //Multiple
values are separated by commas.

        $split_731_3 = explode('731$3', $data);
        $split_731_3 = explode ($split_731_3[0].'731$3', $data);
        $split_731_3 = str_replace('731$3', '[comma]', $split_731_3[1]);
        $split_731_3 = preg_split('/(\d){3}\$./', $split_731_3);

        $split_731_3 = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_731_3[0]);

        $split_731_3 = explode('[comma]', $split_731_3);

        if($split_731_3[0] == $split_731_3[1]){
            $split_731_3 = $split_731_3[0];
        }
        else{
            $split_731_3 = implode('[comma]', $split_731_3);
        }

        $split_731_3 = str_replace(',', '%2C', $split_731_3); //Escape original
commas in URI.
        $split_731_3 = str_replace('[comma]', ',', $split_731_3); //Multiple
values are separated by commas.

        $split_737_3 = explode('737$3', $data);
        $split_737_3 = explode ($split_737_3[0].'737$3', $data);
        $split_737_3 = str_replace('737$3', '[comma]', $split_737_3[1]);
        $split_737_3 = preg_split('/(\d){3}\$./', $split_737_3);

        $split_737_3 = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_737_3[0]);

        $split_737_3 = explode('[comma]', $split_737_3);

        if($split_737_3[0] == $split_737_3[1]){
            $split_737_3 = $split_737_3[0];
        }
        else{
            $split_737_3 = implode('[comma]', $split_737_3);
```

```
        }

        $split_737_3 = str_replace(',', '%2C', $split_737_3); //Escape original
commas in URI.
        $split_737_3 = str_replace('[comma]', ',', $split_737_3); //Multiple
values are separated by commas.

        $wor_has_publisher_per =
$split_720_3.','.$split_730_3.','.$split_721_3.','.$split_731_3.','.$split_727_3.'
,'.$split_737_3;

        $wor_has_publisher_per = explode(',', $wor_has_publisher_per);

        if(array_key_exists($wor_has_publisher_per[0],
$per_bnf_record_number_array)){
            $wor_has_publisher_per[0] =
$per_bnf_record_number_array[$wor_has_publisher_per[0]];
        }
        else{
            $wor_has_publisher_per[0] = '';
        }

        if(array_key_exists($wor_has_publisher_per[1],
$per_bnf_record_number_array)){
            $wor_has_publisher_per[1] =
$per_bnf_record_number_array[$wor_has_publisher_per[1]];
        }
        else{
            $wor_has_publisher_per[1] = '';
        }

        if(array_key_exists($wor_has_publisher_per[2],
$per_bnf_record_number_array)){
            $wor_has_publisher_per[2] =
$per_bnf_record_number_array[$wor_has_publisher_per[2]];
        }
        else{
            $wor_has_publisher_per[2] = '';
        }

        if(array_key_exists($wor_has_publisher_per[3],
$per_bnf_record_number_array)){
            $wor_has_publisher_per[3] =
$per_bnf_record_number_array[$wor_has_publisher_per[3]];
        }
        else{
            $wor_has_publisher_per[3] = '';
        }

        if(array_key_exists($wor_has_publisher_per[4],
$per_bnf_record_number_array)){
            $wor_has_publisher_per[4] =
$per_bnf_record_number_array[$wor_has_publisher_per[4]];
        }
        else{
            $wor_has_publisher_per[4] = '';
        }
```

```
        if(array_key_exists($wor_has_publisher_per[5],
$per_bnf_record_number_array)){
            $wor_has_publisher_per[5] =
$per_bnf_record_number_array[$wor_has_publisher_per[5]];
        }
        else{
            $wor_has_publisher_per[5] = '';
        }

        if(array_key_exists($wor_has_publisher_per[6],
$per_bnf_record_number_array)){
            $wor_has_publisher_per[6] =
$per_bnf_record_number_array[$wor_has_publisher_per[6]];
        }
        else{
            $wor_has_publisher_per[6] = '';
        }

        if(array_key_exists($wor_has_publisher_per[7],
$per_bnf_record_number_array)){
            $wor_has_publisher_per[7] =
$per_bnf_record_number_array[$wor_has_publisher_per[7]];
        }
        else{
            $wor_has_publisher_per[7] = '';
        }

        if(array_key_exists($wor_has_publisher_per[8],
$per_bnf_record_number_array)){
            $wor_has_publisher_per[8] =
$per_bnf_record_number_array[$wor_has_publisher_per[8]];
        }
        else{
            $wor_has_publisher_per[8] = '';
        }

        if(array_key_exists($wor_has_publisher_per[9],
$per_bnf_record_number_array)){
            $wor_has_publisher_per[9] =
$per_bnf_record_number_array[$wor_has_publisher_per[9]];
        }
        else{
            $wor_has_publisher_per[9] = '';
        }

        if(array_key_exists($wor_has_publisher_per[10],
$per_bnf_record_number_array)){
            $wor_has_publisher_per[10] =
$per_bnf_record_number_array[$wor_has_publisher_per[10]];
        }
        else{
            $wor_has_publisher_per[10] = '';
        }

        if(array_key_exists($wor_has_publisher_per[11],
$per_bnf_record_number_array)){
            $wor_has_publisher_per[11] =
$per_bnf_record_number_array[$wor_has_publisher_per[11]];
        }
```

```
        else{
            $wor_has_publisher_per[11] = '';
        }

        if(array_key_exists($wor_has_publisher_per[12],
$per_bnf_record_number_array)){
            $wor_has_publisher_per[12] =
$per_bnf_record_number_array[$wor_has_publisher_per[12]];
        }
        else{
            $wor_has_publisher_per[12] = '';
        }

        if(array_key_exists($wor_has_publisher_per[13],
$per_bnf_record_number_array)){
            $wor_has_publisher_per[13] =
$per_bnf_record_number_array[$wor_has_publisher_per[13]];
        }
        else{
            $wor_has_publisher_per[13] = '';
        }

        if(array_key_exists($wor_has_publisher_per[14],
$per_bnf_record_number_array)){
            $wor_has_publisher_per[14] =
$per_bnf_record_number_array[$wor_has_publisher_per[14]];
        }
        else{
            $wor_has_publisher_per[14] = '';
        }

        if(array_key_exists($wor_has_publisher_per[15],
$per_bnf_record_number_array)){
            $wor_has_publisher_per[15] =
$per_bnf_record_number_array[$wor_has_publisher_per[15]];
        }
        else{
            $wor_has_publisher_per[15] = '';
        }

        if(array_key_exists($wor_has_publisher_per[16],
$per_bnf_record_number_array)){
            $wor_has_publisher_per[16] =
$per_bnf_record_number_array[$wor_has_publisher_per[16]];
        }
        else{
            $wor_has_publisher_per[16] = '';
        }

        if(array_key_exists($wor_has_publisher_per[17],
$per_bnf_record_number_array)){
            $wor_has_publisher_per[17] =
$per_bnf_record_number_array[$wor_has_publisher_per[17]];
        }
        else{
            $wor_has_publisher_per[17] = '';
        }
```

```
        if(array_key_exists($wor_has_publisher_per[18],
$per_bnf_record_number_array)){
            $wor_has_publisher_per[18] =
$per_bnf_record_number_array[$wor_has_publisher_per[18]];
        }
        else{
            $wor_has_publisher_per[18] = '';
        }

        if(array_key_exists($wor_has_publisher_per[19],
$per_bnf_record_number_array)){
            $wor_has_publisher_per[19] =
$per_bnf_record_number_array[$wor_has_publisher_per[19]];
        }
        else{
            $wor_has_publisher_per[19] = '';
        }

        if(array_key_exists($wor_has_publisher_per[20],
$per_bnf_record_number_array)){
            $wor_has_publisher_per[20] =
$per_bnf_record_number_array[$wor_has_publisher_per[20]];
        }
        else{
            $wor_has_publisher_per[20] = '';
        }

        if(array_key_exists($wor_has_publisher_per[21],
$per_bnf_record_number_array)){
            $wor_has_publisher_per[21] =
$per_bnf_record_number_array[$wor_has_publisher_per[21]];
        }
        else{
            $wor_has_publisher_per[21] = '';
        }

        if(array_key_exists($wor_has_publisher_per[22],
$per_bnf_record_number_array)){
            $wor_has_publisher_per[22] =
$per_bnf_record_number_array[$wor_has_publisher_per[22]];
        }
        else{
            $wor_has_publisher_per[22] = '';
        }

        if(array_key_exists($wor_has_publisher_per[23],
$per_bnf_record_number_array)){
            $wor_has_publisher_per[23] =
$per_bnf_record_number_array[$wor_has_publisher_per[23]];
        }
        else{
            $wor_has_publisher_per[23] = '';
        }

        if(array_key_exists($wor_has_publisher_per[24],
$per_bnf_record_number_array)){
            $wor_has_publisher_per[24] =
$per_bnf_record_number_array[$wor_has_publisher_per[24]];
        }
```

```
        else{
            $wor_has_publisher_per[24] = '';
        }

        if(array_key_exists($wor_has_publisher_per[25],
$per_bnf_record_number_array)){
            $wor_has_publisher_per[25] =
$per_bnf_record_number_array[$wor_has_publisher_per[25]];
        }
        else{
            $wor_has_publisher_per[25] = '';
        }

        if(array_key_exists($wor_has_publisher_per[26],
$per_bnf_record_number_array)){
            $wor_has_publisher_per[26] =
$per_bnf_record_number_array[$wor_has_publisher_per[26]];
        }
        else{
            $wor_has_publisher_per[26] = '';
        }

        if(array_key_exists($wor_has_publisher_per[27],
$per_bnf_record_number_array)){
            $wor_has_publisher_per[27] =
$per_bnf_record_number_array[$wor_has_publisher_per[27]];
        }
        else{
            $wor_has_publisher_per[27] = '';
        }

        if(array_key_exists($wor_has_publisher_per[28],
$per_bnf_record_number_array)){
            $wor_has_publisher_per[28] =
$per_bnf_record_number_array[$wor_has_publisher_per[28]];
        }
        else{
            $wor_has_publisher_per[28] = '';
        }

        if(array_key_exists($wor_has_publisher_per[29],
$per_bnf_record_number_array)){
            $wor_has_publisher_per[29] =
$per_bnf_record_number_array[$wor_has_publisher_per[29]];
        }
        else{
            $wor_has_publisher_per[29] = '';
        }

        if(array_key_exists($wor_has_publisher_per[30],
$per_bnf_record_number_array)){
            $wor_has_publisher_per[30] =
$per_bnf_record_number_array[$wor_has_publisher_per[30]];
        }
        else{
            $wor_has_publisher_per[30] = '';
        }
```

```
        if(array_key_exists($wor_has_publisher_per[31],
$per_bnf_record_number_array)){
            $wor_has_publisher_per[31] =
$per_bnf_record_number_array[$wor_has_publisher_per[31]];
        }
        else{
            $wor_has_publisher_per[31] = '';
        }

        if(array_key_exists($wor_has_publisher_per[32],
$per_bnf_record_number_array)){
            $wor_has_publisher_per[32] =
$per_bnf_record_number_array[$wor_has_publisher_per[32]];
        }
        else{
            $wor_has_publisher_per[32] = '';
        }

        if(array_key_exists($wor_has_publisher_per[33],
$per_bnf_record_number_array)){
            $wor_has_publisher_per[33] =
$per_bnf_record_number_array[$wor_has_publisher_per[33]];
        }
        else{
            $wor_has_publisher_per[33] = '';
        }

        if(array_key_exists($wor_has_publisher_per[34],
$per_bnf_record_number_array)){
            $wor_has_publisher_per[34] =
$per_bnf_record_number_array[$wor_has_publisher_per[34]];
        }
        else{
            $wor_has_publisher_per[34] = '';
        }

        if(array_key_exists($wor_has_publisher_per[35],
$per_bnf_record_number_array)){
            $wor_has_publisher_per[35] =
$per_bnf_record_number_array[$wor_has_publisher_per[35]];
        }
        else{
            $wor_has_publisher_per[35] = '';
        }

        if(array_key_exists($wor_has_publisher_per[36],
$per_bnf_record_number_array)){
            $wor_has_publisher_per[36] =
$per_bnf_record_number_array[$wor_has_publisher_per[36]];
        }
        else{
            $wor_has_publisher_per[36] = '';
        }

        if(array_key_exists($wor_has_publisher_per[37],
$per_bnf_record_number_array)){
            $wor_has_publisher_per[37] =
$per_bnf_record_number_array[$wor_has_publisher_per[37]];
        }
```

```
        else{
            $wor_has_publisher_per[37] = '';
        }

        if(array_key_exists($wor_has_publisher_per[38],
$per_bnf_record_number_array)){
            $wor_has_publisher_per[38] =
$per_bnf_record_number_array[$wor_has_publisher_per[38]];
        }
        else{
            $wor_has_publisher_per[38] = '';
        }

        if(array_key_exists($wor_has_publisher_per[39],
$per_bnf_record_number_array)){
            $wor_has_publisher_per[39] =
$per_bnf_record_number_array[$wor_has_publisher_per[39]];
        }
        else{
            $wor_has_publisher_per[39] = '';
        }

        if(array_key_exists($wor_has_publisher_per[40],
$per_bnf_record_number_array)){
            $wor_has_publisher_per[40] =
$per_bnf_record_number_array[$wor_has_publisher_per[40]];
        }
        else{
            $wor_has_publisher_per[40] = '';
        }

        if(array_key_exists($wor_has_publisher_per[41],
$per_bnf_record_number_array)){
            $wor_has_publisher_per[41] =
$per_bnf_record_number_array[$wor_has_publisher_per[41]];
        }
        else{
            $wor_has_publisher_per[41] = '';
        }

        if(array_key_exists($wor_has_publisher_per[42],
$per_bnf_record_number_array)){
            $wor_has_publisher_per[42] =
$per_bnf_record_number_array[$wor_has_publisher_per[42]];
        }
        else{
            $wor_has_publisher_per[42] = '';
        }

        if(array_key_exists($wor_has_publisher_per[43],
$per_bnf_record_number_array)){
            $wor_has_publisher_per[43] =
$per_bnf_record_number_array[$wor_has_publisher_per[43]];
        }
        else{
            $wor_has_publisher_per[43] = '';
        }
```

```
        if(array_key_exists($wor_has_publisher_per[44],
$per_bnf_record_number_array)){
            $wor_has_publisher_per[44] =
$per_bnf_record_number_array[$wor_has_publisher_per[44]];
        }
        else{
            $wor_has_publisher_per[44] = '';
        }

        if(array_key_exists($wor_has_publisher_per[45],
$per_bnf_record_number_array)){
            $wor_has_publisher_per[45] =
$per_bnf_record_number_array[$wor_has_publisher_per[45]];
        }
        else{
            $wor_has_publisher_per[45] = '';
        }

        if(array_key_exists($wor_has_publisher_per[46],
$per_bnf_record_number_array)){
            $wor_has_publisher_per[46] =
$per_bnf_record_number_array[$wor_has_publisher_per[46]];
        }
        else{
            $wor_has_publisher_per[46] = '';
        }

        if(array_key_exists($wor_has_publisher_per[47],
$per_bnf_record_number_array)){
            $wor_has_publisher_per[47] =
$per_bnf_record_number_array[$wor_has_publisher_per[47]];
        }
        else{
            $wor_has_publisher_per[47] = '';
        }

        if(array_key_exists($wor_has_publisher_per[48],
$per_bnf_record_number_array)){
            $wor_has_publisher_per[48] =
$per_bnf_record_number_array[$wor_has_publisher_per[48]];
        }
        else{
            $wor_has_publisher_per[48] = '';
        }

        if(array_key_exists($wor_has_publisher_per[49],
$per_bnf_record_number_array)){
            $wor_has_publisher_per[49] =
$per_bnf_record_number_array[$wor_has_publisher_per[49]];
        }
        else{
            $wor_has_publisher_per[49] = '';
        }

        if($wor_has_publisher_per[50] !=''){
            $all_remark_verified = 'verified';
        }

        $wor_has_publisher_per = implode(',', $wor_has_publisher_per);
```

```
        $wor_has_publisher_per = preg_replace('/,{2,}/', ',',
$wor_has_publisher_per);

        if(str_ends_with($wor_has_publisher_per, ',')){
            $wor_has_publisher_per = substr($wor_has_publisher_per, 0, -1);
        }

        if(str_starts_with($wor_has_publisher_per, ',')){
            $wor_has_publisher_per = substr($wor_has_publisher_per, 1);
        }

        if(stripos($wor_has_publisher_per, ',')){
            $wor_has_publisher_per = '"'.$wor_has_publisher_per.'"';
        }

//      Create field wor_has_other_creator_person from 701$3, 711$3, 725$3, 735$3.

        $split_701_3 = explode('701$3', $data);
        $split_701_3 = explode ($split_701_3[0].'701$3', $data);
        $split_701_3 = str_replace('701$3', '[comma]', $split_701_3[1]);
        $split_701_3 = preg_split('/(\d){3}\$./', $split_701_3);

        $split_701_3 = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_701_3[0]);

        $split_701_3 = explode('[comma]', $split_701_3);

        if($split_701_3[0] == $split_701_3[1]){
            $split_701_3 = $split_701_3[0];
        }
        else{
            $split_701_3 = implode('[comma]', $split_701_3);
        }

        $split_701_3 = str_replace(',', '%2C', $split_701_3); //Escape original
commas in URI.
        $split_701_3 = str_replace('[comma]', ',', $split_701_3); //Multiple
values are separated by commas.

        $split_711_3 = explode('711$3', $data);
        $split_711_3 = explode ($split_711_3[0].'711$3', $data);
        $split_711_3 = str_replace('711$3', '[comma]', $split_711_3[1]);
        $split_711_3 = preg_split('/(\d){3}\$./', $split_711_3);

        $split_711_3 = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_711_3[0]);

        $split_711_3 = explode('[comma]', $split_711_3);

        if($split_711_3[0] == $split_711_3[1]){
            $split_711_3 = $split_711_3[0];
        }
        else{
            $split_711_3 = implode('[comma]', $split_711_3);
        }

        $split_711_3 = str_replace(',', '%2C', $split_711_3); //Escape original
commas in URI.
```

```
        $split_711_3 = str_replace('[comma]', ',', $split_711_3); //Multiple
values are separated by commas.

        $split_725_3 = explode('725$3', $data);
        $split_725_3 = explode ($split_725_3[0].'725$3', $data);
        $split_725_3 = str_replace('725$3', '[comma]', $split_725_3[1]);
        $split_725_3 = preg_split('/(\d){3}\$./', $split_725_3);

        $split_725_3 = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_725_3[0]);

        $split_725_3 = explode('[comma]', $split_725_3);

        if($split_725_3[0] == $split_725_3[1]){
            $split_725_3 = $split_725_3[0];
        }
        else{
            $split_725_3 = implode('[comma]', $split_725_3);
        }

        $split_725_3 = str_replace(',', '%2C', $split_725_3); //Escape original
commas in URI.
        $split_725_3 = str_replace('[comma]', ',', $split_725_3); //Multiple
values are separated by commas.

        $split_735_3 = explode('735$3', $data);
        $split_735_3 = explode ($split_735_3[0].'735$3', $data);
        $split_735_3 = str_replace('735$3', '[comma]', $split_735_3[1]);
        $split_735_3 = preg_split('/(\d){3}\$./', $split_735_3);

        $split_735_3 = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_735_3[0]);

        $split_735_3 = explode('[comma]', $split_735_3);

        if($split_735_3[0] == $split_735_3[1]){
            $split_735_3 = $split_735_3[0];
        }
        else{
            $split_735_3 = implode('[comma]', $split_735_3);
        }

        $split_735_3 = str_replace(',', '%2C', $split_735_3); //Escape original
commas in URI.
        $split_735_3 = str_replace('[comma]', ',', $split_735_3); //Multiple
values are separated by commas.

        $wor_has_other_creator_per =
$split_701_3.','.$split_711_3.','.$split_725_3.','.$split_735_3;

        $wor_has_other_creator_per = explode(',', $wor_has_other_creator_per);

        if(array_key_exists($wor_has_other_creator_per[0],
$per_bnf_record_number_array)){
            $wor_has_other_creator_per[0] =
$per_bnf_record_number_array[$wor_has_other_creator_per[0]];
        }
        else{
            $wor_has_other_creator_per[0] = '';
```

```
        }

        if(array_key_exists($wor_has_other_creator_per[1],
$per_bnf_record_number_array)){
            $wor_has_other_creator_per[1] =
$per_bnf_record_number_array[$wor_has_other_creator_per[1]];
        }
        else{
            $wor_has_other_creator_per[1] = '';
        }

        if(array_key_exists($wor_has_other_creator_per[2],
$per_bnf_record_number_array)){
            $wor_has_other_creator_per[2] =
$per_bnf_record_number_array[$wor_has_other_creator_per[2]];
        }
        else{
            $wor_has_other_creator_per[2] = '';
        }

        if(array_key_exists($wor_has_other_creator_per[3],
$per_bnf_record_number_array)){
            $wor_has_other_creator_per[3] =
$per_bnf_record_number_array[$wor_has_other_creator_per[3]];
        }
        else{
            $wor_has_other_creator_per[3] = '';
        }

        if(array_key_exists($wor_has_other_creator_per[4],
$per_bnf_record_number_array)){
            $wor_has_other_creator_per[4] =
$per_bnf_record_number_array[$wor_has_other_creator_per[4]];
        }
        else{
            $wor_has_other_creator_per[4] = '';
        }

        if(array_key_exists($wor_has_other_creator_per[5],
$per_bnf_record_number_array)){
            $wor_has_other_creator_per[5] =
$per_bnf_record_number_array[$wor_has_other_creator_per[5]];
        }
        else{
            $wor_has_other_creator_per[5] = '';
        }

        if(array_key_exists($wor_has_other_creator_per[6],
$per_bnf_record_number_array)){
            $wor_has_other_creator_per[6] =
$per_bnf_record_number_array[$wor_has_other_creator_per[6]];
        }
        else{
            $wor_has_other_creator_per[6] = '';
        }

        if(array_key_exists($wor_has_other_creator_per[7],
$per_bnf_record_number_array)){
```

```
            $wor_has_other_creator_per[7] =
$per_bnf_record_number_array[$wor_has_other_creator_per[7]];
        }
        else{
            $wor_has_other_creator_per[7] = '';
        }

        if(array_key_exists($wor_has_other_creator_per[8],
$per_bnf_record_number_array)){
            $wor_has_other_creator_per[8] =
$per_bnf_record_number_array[$wor_has_other_creator_per[8]];
        }
        else{
            $wor_has_other_creator_per[8] = '';
        }

        if(array_key_exists($wor_has_other_creator_per[9],
$per_bnf_record_number_array)){
            $wor_has_other_creator_per[9] =
$per_bnf_record_number_array[$wor_has_other_creator_per[9]];
        }
        else{
            $wor_has_other_creator_per[9] = '';
        }

        if(array_key_exists($wor_has_other_creator_per[10],
$per_bnf_record_number_array)){
            $wor_has_other_creator_per[10] =
$per_bnf_record_number_array[$wor_has_other_creator_per[10]];
        }
        else{
            $wor_has_other_creator_per[10] = '';
        }

        if(array_key_exists($wor_has_other_creator_per[11],
$per_bnf_record_number_array)){
            $wor_has_other_creator_per[11] =
$per_bnf_record_number_array[$wor_has_other_creator_per[11]];
        }
        else{
            $wor_has_other_creator_per[11] = '';
        }

        if(array_key_exists($wor_has_other_creator_per[12],
$per_bnf_record_number_array)){
            $wor_has_other_creator_per[12] =
$per_bnf_record_number_array[$wor_has_other_creator_per[12]];
        }
        else{
            $wor_has_other_creator_per[12] = '';
        }

        if(array_key_exists($wor_has_other_creator_per[13],
$per_bnf_record_number_array)){
            $wor_has_other_creator_per[13] =
$per_bnf_record_number_array[$wor_has_other_creator_per[13]];
        }
        else{
            $wor_has_other_creator_per[13] = '';
```

```
        }

        if(array_key_exists($wor_has_other_creator_per[14],
$per_bnf_record_number_array)){
            $wor_has_other_creator_per[14] =
$per_bnf_record_number_array[$wor_has_other_creator_per[14]];
        }
        else{
            $wor_has_other_creator_per[14] = '';
        }

        if(array_key_exists($wor_has_other_creator_per[15],
$per_bnf_record_number_array)){
            $wor_has_other_creator_per[15] =
$per_bnf_record_number_array[$wor_has_other_creator_per[15]];
        }
        else{
            $wor_has_other_creator_per[15] = '';
        }

        if(array_key_exists($wor_has_other_creator_per[16],
$per_bnf_record_number_array)){
            $wor_has_other_creator_per[16] =
$per_bnf_record_number_array[$wor_has_other_creator_per[16]];
        }
        else{
            $wor_has_other_creator_per[16] = '';
        }

        if(array_key_exists($wor_has_other_creator_per[17],
$per_bnf_record_number_array)){
            $wor_has_other_creator_per[17] =
$per_bnf_record_number_array[$wor_has_other_creator_per[17]];
        }
        else{
            $wor_has_other_creator_per[17] = '';
        }

        if(array_key_exists($wor_has_other_creator_per[18],
$per_bnf_record_number_array)){
            $wor_has_other_creator_per[18] =
$per_bnf_record_number_array[$wor_has_other_creator_per[18]];
        }
        else{
            $wor_has_other_creator_per[18] = '';
        }

        if(array_key_exists($wor_has_other_creator_per[19],
$per_bnf_record_number_array)){
            $wor_has_other_creator_per[19] =
$per_bnf_record_number_array[$wor_has_other_creator_per[19]];
        }
        else{
            $wor_has_other_creator_per[19] = '';
        }

        if(array_key_exists($wor_has_other_creator_per[20],
$per_bnf_record_number_array)){
```

```
            $wor_has_other_creator_per[20] =
$per_bnf_record_number_array[$wor_has_other_creator_per[20]];
        }
        else{
            $wor_has_other_creator_per[20] = '';
        }

        if(array_key_exists($wor_has_other_creator_per[21],
$per_bnf_record_number_array)){
            $wor_has_other_creator_per[21] =
$per_bnf_record_number_array[$wor_has_other_creator_per[21]];
        }
        else{
            $wor_has_other_creator_per[21] = '';
        }

        if(array_key_exists($wor_has_other_creator_per[22],
$per_bnf_record_number_array)){
            $wor_has_other_creator_per[22] =
$per_bnf_record_number_array[$wor_has_other_creator_per[22]];
        }
        else{
            $wor_has_other_creator_per[22] = '';
        }

        if(array_key_exists($wor_has_other_creator_per[23],
$per_bnf_record_number_array)){
            $wor_has_other_creator_per[23] =
$per_bnf_record_number_array[$wor_has_other_creator_per[23]];
        }
        else{
            $wor_has_other_creator_per[23] = '';
        }

        if(array_key_exists($wor_has_other_creator_per[24],
$per_bnf_record_number_array)){
            $wor_has_other_creator_per[24] =
$per_bnf_record_number_array[$wor_has_other_creator_per[24]];
        }
        else{
            $wor_has_other_creator_per[24] = '';
        }

        if(array_key_exists($wor_has_other_creator_per[25],
$per_bnf_record_number_array)){
            $wor_has_other_creator_per[25] =
$per_bnf_record_number_array[$wor_has_other_creator_per[25]];
        }
        else{
            $wor_has_other_creator_per[25] = '';
        }

        if(array_key_exists($wor_has_other_creator_per[26],
$per_bnf_record_number_array)){
            $wor_has_other_creator_per[26] =
$per_bnf_record_number_array[$wor_has_other_creator_per[26]];
        }
        else{
            $wor_has_other_creator_per[26] = '';
```

```
        }

        if(array_key_exists($wor_has_other_creator_per[27],
$per_bnf_record_number_array)){
            $wor_has_other_creator_per[27] =
$per_bnf_record_number_array[$wor_has_other_creator_per[27]];
        }
        else{
            $wor_has_other_creator_per[27] = '';
        }

        if(array_key_exists($wor_has_other_creator_per[28],
$per_bnf_record_number_array)){
            $wor_has_other_creator_per[28] =
$per_bnf_record_number_array[$wor_has_other_creator_per[28]];
        }
        else{
            $wor_has_other_creator_per[28] = '';
        }

        if(array_key_exists($wor_has_other_creator_per[29],
$per_bnf_record_number_array)){
            $wor_has_other_creator_per[29] =
$per_bnf_record_number_array[$wor_has_other_creator_per[29]];
        }
        else{
            $wor_has_other_creator_per[29] = '';
        }

        if(array_key_exists($wor_has_other_creator_per[30],
$per_bnf_record_number_array)){
            $wor_has_other_creator_per[30] =
$per_bnf_record_number_array[$wor_has_other_creator_per[30]];
        }
        else{
            $wor_has_other_creator_per[30] = '';
        }

        if(array_key_exists($wor_has_other_creator_per[31],
$per_bnf_record_number_array)){
            $wor_has_other_creator_per[31] =
$per_bnf_record_number_array[$wor_has_other_creator_per[31]];
        }
        else{
            $wor_has_other_creator_per[31] = '';
        }

        if(array_key_exists($wor_has_other_creator_per[32],
$per_bnf_record_number_array)){
            $wor_has_other_creator_per[32] =
$per_bnf_record_number_array[$wor_has_other_creator_per[32]];
        }
        else{
            $wor_has_other_creator_per[32] = '';
        }

        if(array_key_exists($wor_has_other_creator_per[33],
$per_bnf_record_number_array)){
```

```
            $wor_has_other_creator_per[33] =
$per_bnf_record_number_array[$wor_has_other_creator_per[33]];
        }
        else{
            $wor_has_other_creator_per[33] = '';
        }

        if(array_key_exists($wor_has_other_creator_per[34],
$per_bnf_record_number_array)){
            $wor_has_other_creator_per[34] =
$per_bnf_record_number_array[$wor_has_other_creator_per[34]];
        }
        else{
            $wor_has_other_creator_per[34] = '';
        }

        if(array_key_exists($wor_has_other_creator_per[35],
$per_bnf_record_number_array)){
            $wor_has_other_creator_per[35] =
$per_bnf_record_number_array[$wor_has_other_creator_per[35]];
        }
        else{
            $wor_has_other_creator_per[35] = '';
        }

        if(array_key_exists($wor_has_other_creator_per[36],
$per_bnf_record_number_array)){
            $wor_has_other_creator_per[36] =
$per_bnf_record_number_array[$wor_has_other_creator_per[36]];
        }
        else{
            $wor_has_other_creator_per[36] = '';
        }

        if(array_key_exists($wor_has_other_creator_per[37],
$per_bnf_record_number_array)){
            $wor_has_other_creator_per[37] =
$per_bnf_record_number_array[$wor_has_other_creator_per[37]];
        }
        else{
            $wor_has_other_creator_per[37] = '';
        }

        if(array_key_exists($wor_has_other_creator_per[38],
$per_bnf_record_number_array)){
            $wor_has_other_creator_per[38] =
$per_bnf_record_number_array[$wor_has_other_creator_per[38]];
        }
        else{
            $wor_has_other_creator_per[38] = '';
        }

        if(array_key_exists($wor_has_other_creator_per[39],
$per_bnf_record_number_array)){
            $wor_has_other_creator_per[39] =
$per_bnf_record_number_array[$wor_has_other_creator_per[39]];
        }
        else{
            $wor_has_other_creator_per[39] = '';
```

```
        }

        if(array_key_exists($wor_has_other_creator_per[40],
$per_bnf_record_number_array)){
            $wor_has_other_creator_per[40] =
$per_bnf_record_number_array[$wor_has_other_creator_per[40]];
        }
        else{
            $wor_has_other_creator_per[40] = '';
        }

        if(array_key_exists($wor_has_other_creator_per[41],
$per_bnf_record_number_array)){
            $wor_has_other_creator_per[41] =
$per_bnf_record_number_array[$wor_has_other_creator_per[41]];
        }
        else{
            $wor_has_other_creator_per[41] = '';
        }

        if(array_key_exists($wor_has_other_creator_per[42],
$per_bnf_record_number_array)){
            $wor_has_other_creator_per[42] =
$per_bnf_record_number_array[$wor_has_other_creator_per[42]];
        }
        else{
            $wor_has_other_creator_per[42] = '';
        }

        if(array_key_exists($wor_has_other_creator_per[43],
$per_bnf_record_number_array)){
            $wor_has_other_creator_per[43] =
$per_bnf_record_number_array[$wor_has_other_creator_per[43]];
        }
        else{
            $wor_has_other_creator_per[43] = '';
        }

        if(array_key_exists($wor_has_other_creator_per[44],
$per_bnf_record_number_array)){
            $wor_has_other_creator_per[44] =
$per_bnf_record_number_array[$wor_has_other_creator_per[44]];
        }
        else{
            $wor_has_other_creator_per[44] = '';
        }

        if(array_key_exists($wor_has_other_creator_per[45],
$per_bnf_record_number_array)){
            $wor_has_other_creator_per[45] =
$per_bnf_record_number_array[$wor_has_other_creator_per[45]];
        }
        else{
            $wor_has_other_creator_per[45] = '';
        }

        if(array_key_exists($wor_has_other_creator_per[46],
$per_bnf_record_number_array)){
```

```
            $wor_has_other_creator_per[46] =
$per_bnf_record_number_array[$wor_has_other_creator_per[46]];
        }
        else{
            $wor_has_other_creator_per[46] = '';
        }

        if(array_key_exists($wor_has_other_creator_per[47],
$per_bnf_record_number_array)){
            $wor_has_other_creator_per[47] =
$per_bnf_record_number_array[$wor_has_other_creator_per[47]];
        }
        else{
            $wor_has_other_creator_per[47] = '';
        }

        if(array_key_exists($wor_has_other_creator_per[48],
$per_bnf_record_number_array)){
            $wor_has_other_creator_per[48] =
$per_bnf_record_number_array[$wor_has_other_creator_per[48]];
        }
        else{
            $wor_has_other_creator_per[48] = '';
        }

        if(array_key_exists($wor_has_other_creator_per[49],
$per_bnf_record_number_array)){
            $wor_has_other_creator_per[49] =
$per_bnf_record_number_array[$wor_has_other_creator_per[49]];
        }
        else{
            $wor_has_other_creator_per[49] = '';
        }

        if($wor_has_other_creator_per[50] !=''){
            $all_remark_verified = 'verified';
        }

        $wor_has_other_creator_per = implode(',', $wor_has_other_creator_per);

        $wor_has_other_creator_per = preg_replace('/,{2,}/', ',',
$wor_has_other_creator_per);

        if(str_ends_with($wor_has_other_creator_per, ',')){
            $wor_has_other_creator_per = substr($wor_has_other_creator_per, 0, -
1);
        }

        if(str_starts_with($wor_has_other_creator_per, ',')){
            $wor_has_other_creator_per = substr($wor_has_other_creator_per, 1);
        }

        if(stripos($wor_has_other_creator_per, ',')){
            $wor_has_other_creator_per = '"'.$wor_has_other_creator_per.'"';
        }

//      Create field wor_refers_to_per from 600$3 and 610$3.

        $split_600_3 = explode('600$3', $data);
```

```
        $split_600_3 = explode ($split_600_3[0].'600$3', $data);
        $split_600_3 = str_replace('600$3', '[comma]', $split_600_3[1]);
        $split_600_3 = preg_split('/(\d){3}\$./', $split_600_3);

        $split_600_3 = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_600_3[0]);

        $split_600_3 = explode('[comma]', $split_600_3);

        if($split_600_3[0] == $split_600_3[1]){
            $split_600_3 = $split_600_3[0];
        }
        else{
            $split_600_3 = implode('[comma]', $split_600_3);
        }

        $split_600_3 = str_replace(',', '%2C', $split_600_3); //Escape original
commas in URI.
        $split_600_3 = str_replace('[comma]', ',', $split_600_3); //Multiple
values are separated by commas.

        $split_610_3 = explode('610$3', $data);
        $split_610_3 = explode ($split_610_3[0].'610$3', $data);
        $split_610_3 = str_replace('610$3', '[comma]', $split_610_3[1]);
        $split_610_3 = preg_split('/(\d){3}\$./', $split_610_3);

        $split_610_3 = str_replace("<span style='color:blue'>[tab]</span>", '',
$split_610_3[0]);

        $split_610_3 = explode('[comma]', $split_610_3);

        if($split_610_3[0] == $split_610_3[1]){
            $split_610_3 = $split_610_3[0];
        }
        else{
            $split_610_3 = implode('[comma]', $split_610_3);
        }

        $split_610_3 = str_replace(',', '%2C', $split_610_3); //Escape original
commas in URI.
        $split_610_3 = str_replace('[comma]', ',', $split_610_3); //Multiple
values are separated by commas.

        $wor_refers_to_per = $split_600_3.','.$split_610_3;

        $wor_refers_to_per = explode(',', $wor_refers_to_per);

        if(array_key_exists($wor_refers_to_per[0], $per_bnf_record_number_array)){
            $wor_refers_to_per[0] =
$per_bnf_record_number_array[$wor_refers_to_per[0]];
        }
        else{
            $wor_refers_to_per[0] = '';
        }

        if(array_key_exists($wor_refers_to_per[1], $per_bnf_record_number_array)){
            $wor_refers_to_per[1] =
$per_bnf_record_number_array[$wor_refers_to_per[1]];
        }
```

```
        else{
            $wor_refers_to_per[1] = '';
        }

        if(array_key_exists($wor_refers_to_per[2], $per_bnf_record_number_array)){
            $wor_refers_to_per[2] =
$per_bnf_record_number_array[$wor_refers_to_per[2]];
        }
        else{
            $wor_refers_to_per[2] = '';
        }

        if(array_key_exists($wor_refers_to_per[3], $per_bnf_record_number_array)){
            $wor_refers_to_per[3] =
$per_bnf_record_number_array[$wor_refers_to_per[3]];
        }
        else{
            $wor_refers_to_per[3] = '';
        }

        if(array_key_exists($wor_refers_to_per[4], $per_bnf_record_number_array)){
            $wor_refers_to_per[4] =
$per_bnf_record_number_array[$wor_refers_to_per[4]];
        }
        else{
            $wor_refers_to_per[4] = '';
        }

        if(array_key_exists($wor_refers_to_per[5], $per_bnf_record_number_array)){
            $wor_refers_to_per[5] =
$per_bnf_record_number_array[$wor_refers_to_per[5]];
        }
        else{
            $wor_refers_to_per[5] = '';
        }

        if(array_key_exists($wor_refers_to_per[6], $per_bnf_record_number_array)){
            $wor_refers_to_per[6] =
$per_bnf_record_number_array[$wor_refers_to_per[6]];
        }
        else{
            $wor_refers_to_per[6] = '';
        }

        if(array_key_exists($wor_refers_to_per[7], $per_bnf_record_number_array)){
            $wor_refers_to_per[7] =
$per_bnf_record_number_array[$wor_refers_to_per[7]];
        }
        else{
            $wor_refers_to_per[7] = '';
        }

        if(array_key_exists($wor_refers_to_per[8], $per_bnf_record_number_array)){
            $wor_refers_to_per[8] =
$per_bnf_record_number_array[$wor_refers_to_per[8]];
        }
        else{
            $wor_refers_to_per[8] = '';
        }
```

```
        if(array_key_exists($wor_refers_to_per[9], $per_bnf_record_number_array)){
            $wor_refers_to_per[9] =
$per_bnf_record_number_array[$wor_refers_to_per[9]];
        }
        else{
            $wor_refers_to_per[9] = '';
        }

        if(array_key_exists($wor_refers_to_per[10],
$per_bnf_record_number_array)){
            $wor_refers_to_per[10] =
$per_bnf_record_number_array[$wor_refers_to_per[10]];
        }
        else{
            $wor_refers_to_per[10] = '';
        }

        if(array_key_exists($wor_refers_to_per[11],
$per_bnf_record_number_array)){
            $wor_refers_to_per[11] =
$per_bnf_record_number_array[$wor_refers_to_per[11]];
        }
        else{
            $wor_refers_to_per[11] = '';
        }

        if(array_key_exists($wor_refers_to_per[12],
$per_bnf_record_number_array)){
            $wor_refers_to_per[12] =
$per_bnf_record_number_array[$wor_refers_to_per[12]];
        }
        else{
            $wor_refers_to_per[12] = '';
        }

        if(array_key_exists($wor_refers_to_per[13],
$per_bnf_record_number_array)){
            $wor_refers_to_per[13] =
$per_bnf_record_number_array[$wor_refers_to_per[13]];
        }
        else{
            $wor_refers_to_per[13] = '';
        }

        if(array_key_exists($wor_refers_to_per[14],
$per_bnf_record_number_array)){
            $wor_refers_to_per[14] =
$per_bnf_record_number_array[$wor_refers_to_per[14]];
        }
        else{
            $wor_refers_to_per[14] = '';
        }

        if(array_key_exists($wor_refers_to_per[15],
$per_bnf_record_number_array)){
            $wor_refers_to_per[15] =
$per_bnf_record_number_array[$wor_refers_to_per[15]];
        }
```

```
        else{
            $wor_refers_to_per[15] = '';
        }

        if(array_key_exists($wor_refers_to_per[16],
$per_bnf_record_number_array)){
            $wor_refers_to_per[16] =
$per_bnf_record_number_array[$wor_refers_to_per[16]];
        }
        else{
            $wor_refers_to_per[16] = '';
        }

        if(array_key_exists($wor_refers_to_per[17],
$per_bnf_record_number_array)){
            $wor_refers_to_per[17] =
$per_bnf_record_number_array[$wor_refers_to_per[17]];
        }
        else{
            $wor_refers_to_per[17] = '';
        }

        if(array_key_exists($wor_refers_to_per[18],
$per_bnf_record_number_array)){
            $wor_refers_to_per[18] =
$per_bnf_record_number_array[$wor_refers_to_per[18]];
        }
        else{
            $wor_refers_to_per[18] = '';
        }

        if(array_key_exists($wor_refers_to_per[19],
$per_bnf_record_number_array)){
            $wor_refers_to_per[19] =
$per_bnf_record_number_array[$wor_refers_to_per[19]];
        }
        else{
            $wor_refers_to_per[19] = '';
        }

        if(array_key_exists($wor_refers_to_per[20],
$per_bnf_record_number_array)){
            $wor_refers_to_per[20] =
$per_bnf_record_number_array[$wor_refers_to_per[20]];
        }
        else{
            $wor_refers_to_per[20] = '';
        }

        if(array_key_exists($wor_refers_to_per[21],
$per_bnf_record_number_array)){
            $wor_refers_to_per[21] =
$per_bnf_record_number_array[$wor_refers_to_per[21]];
        }
        else{
            $wor_refers_to_per[21] = '';
        }
```

```
        if(array_key_exists($wor_refers_to_per[22],
$per_bnf_record_number_array)){
            $wor_refers_to_per[22] =
$per_bnf_record_number_array[$wor_refers_to_per[22]];
        }
        else{
            $wor_refers_to_per[22] = '';
        }

        if(array_key_exists($wor_refers_to_per[23],
$per_bnf_record_number_array)){
            $wor_refers_to_per[23] =
$per_bnf_record_number_array[$wor_refers_to_per[23]];
        }
        else{
            $wor_refers_to_per[23] = '';
        }

        if(array_key_exists($wor_refers_to_per[24],
$per_bnf_record_number_array)){
            $wor_refers_to_per[24] =
$per_bnf_record_number_array[$wor_refers_to_per[24]];
        }
        else{
            $wor_refers_to_per[24] = '';
        }

        if(array_key_exists($wor_refers_to_per[25],
$per_bnf_record_number_array)){
            $wor_refers_to_per[25] =
$per_bnf_record_number_array[$wor_refers_to_per[25]];
        }
        else{
            $wor_refers_to_per[25] = '';
        }

        if(array_key_exists($wor_refers_to_per[26],
$per_bnf_record_number_array)){
            $wor_refers_to_per[26] =
$per_bnf_record_number_array[$wor_refers_to_per[26]];
        }
        else{
            $wor_refers_to_per[26] = '';
        }

        if(array_key_exists($wor_refers_to_per[27],
$per_bnf_record_number_array)){
            $wor_refers_to_per[27] =
$per_bnf_record_number_array[$wor_refers_to_per[27]];
        }
        else{
            $wor_refers_to_per[27] = '';
        }

        if(array_key_exists($wor_refers_to_per[28],
$per_bnf_record_number_array)){
            $wor_refers_to_per[28] =
$per_bnf_record_number_array[$wor_refers_to_per[28]];
        }
```

```
        else{
            $wor_refers_to_per[28] = '';
        }

        if(array_key_exists($wor_refers_to_per[29],
$per_bnf_record_number_array)){
            $wor_refers_to_per[29] =
$per_bnf_record_number_array[$wor_refers_to_per[29]];
        }
        else{
            $wor_refers_to_per[29] = '';
        }

        if(array_key_exists($wor_refers_to_per[30],
$per_bnf_record_number_array)){
            $wor_refers_to_per[30] =
$per_bnf_record_number_array[$wor_refers_to_per[30]];
        }
        else{
            $wor_refers_to_per[30] = '';
        }

        if(array_key_exists($wor_refers_to_per[31],
$per_bnf_record_number_array)){
            $wor_refers_to_per[31] =
$per_bnf_record_number_array[$wor_refers_to_per[31]];
        }
        else{
            $wor_refers_to_per[31] = '';
        }

        if(array_key_exists($wor_refers_to_per[32],
$per_bnf_record_number_array)){
            $wor_refers_to_per[32] =
$per_bnf_record_number_array[$wor_refers_to_per[32]];
        }
        else{
            $wor_refers_to_per[32] = '';
        }

        if(array_key_exists($wor_refers_to_per[33],
$per_bnf_record_number_array)){
            $wor_refers_to_per[33] =
$per_bnf_record_number_array[$wor_refers_to_per[33]];
        }
        else{
            $wor_refers_to_per[33] = '';
        }

        if(array_key_exists($wor_refers_to_per[34],
$per_bnf_record_number_array)){
            $wor_refers_to_per[34] =
$per_bnf_record_number_array[$wor_refers_to_per[34]];
        }
        else{
            $wor_refers_to_per[34] = '';
        }
```

```
        if(array_key_exists($wor_refers_to_per[35],
$per_bnf_record_number_array)){
            $wor_refers_to_per[35] =
$per_bnf_record_number_array[$wor_refers_to_per[35]];
        }
        else{
            $wor_refers_to_per[35] = '';
        }

        if(array_key_exists($wor_refers_to_per[36],
$per_bnf_record_number_array)){
            $wor_refers_to_per[36] =
$per_bnf_record_number_array[$wor_refers_to_per[36]];
        }
        else{
            $wor_refers_to_per[36] = '';
        }

        if(array_key_exists($wor_refers_to_per[37],
$per_bnf_record_number_array)){
            $wor_refers_to_per[37] =
$per_bnf_record_number_array[$wor_refers_to_per[37]];
        }
        else{
            $wor_refers_to_per[37] = '';
        }

        if(array_key_exists($wor_refers_to_per[38],
$per_bnf_record_number_array)){
            $wor_refers_to_per[38] =
$per_bnf_record_number_array[$wor_refers_to_per[38]];
        }
        else{
            $wor_refers_to_per[38] = '';
        }

        if(array_key_exists($wor_refers_to_per[39],
$per_bnf_record_number_array)){
            $wor_refers_to_per[39] =
$per_bnf_record_number_array[$wor_refers_to_per[39]];
        }
        else{
            $wor_refers_to_per[39] = '';
        }

        if(array_key_exists($wor_refers_to_per[40],
$per_bnf_record_number_array)){
            $wor_refers_to_per[40] =
$per_bnf_record_number_array[$wor_refers_to_per[40]];
        }
        else{
            $wor_refers_to_per[40] = '';
        }

        if(array_key_exists($wor_refers_to_per[41],
$per_bnf_record_number_array)){
            $wor_refers_to_per[41] =
$per_bnf_record_number_array[$wor_refers_to_per[41]];
        }
```

```
        else{
            $wor_refers_to_per[41] = '';
        }

        if(array_key_exists($wor_refers_to_per[42],
$per_bnf_record_number_array)){
            $wor_refers_to_per[42] =
$per_bnf_record_number_array[$wor_refers_to_per[42]];
        }
        else{
            $wor_refers_to_per[42] = '';
        }

        if(array_key_exists($wor_refers_to_per[43],
$per_bnf_record_number_array)){
            $wor_refers_to_per[43] =
$per_bnf_record_number_array[$wor_refers_to_per[43]];
        }
        else{
            $wor_refers_to_per[43] = '';
        }

        if(array_key_exists($wor_refers_to_per[44],
$per_bnf_record_number_array)){
            $wor_refers_to_per[44] =
$per_bnf_record_number_array[$wor_refers_to_per[44]];
        }
        else{
            $wor_refers_to_per[44] = '';
        }

        if(array_key_exists($wor_refers_to_per[45],
$per_bnf_record_number_array)){
            $wor_refers_to_per[45] =
$per_bnf_record_number_array[$wor_refers_to_per[45]];
        }
        else{
            $wor_refers_to_per[45] = '';
        }

        if(array_key_exists($wor_refers_to_per[46],
$per_bnf_record_number_array)){
            $wor_refers_to_per[46] =
$per_bnf_record_number_array[$wor_refers_to_per[46]];
        }
        else{
            $wor_refers_to_per[46] = '';
        }

        if(array_key_exists($wor_refers_to_per[47],
$per_bnf_record_number_array)){
            $wor_refers_to_per[47] =
$per_bnf_record_number_array[$wor_refers_to_per[47]];
        }
        else{
            $wor_refers_to_per[47] = '';
        }
```

```
        if(array_key_exists($wor_refers_to_per[48],
$per_bnf_record_number_array)){
            $wor_refers_to_per[48] =
$per_bnf_record_number_array[$wor_refers_to_per[48]];
        }
        else{
            $wor_refers_to_per[48] = '';
        }

        if(array_key_exists($wor_refers_to_per[49],
$per_bnf_record_number_array)){
            $wor_refers_to_per[49] =
$per_bnf_record_number_array[$wor_refers_to_per[49]];
        }
        else{
            $wor_refers_to_per[49] = '';
        }

        if($wor_refers_to_per[50] !=''){
            $all_remark_verified = 'verified';
        }

        $wor_refers_to_per = implode(',', $wor_refers_to_per);

        $wor_refers_to_per = preg_replace('/,{2,}/', ',', $wor_refers_to_per);

        if(str_ends_with($wor_refers_to_per, ',')){
            $wor_refers_to_per = substr($wor_refers_to_per, 0, -1);
        }

        if(str_starts_with($wor_refers_to_per, ',')){
            $wor_refers_to_per = substr($wor_refers_to_per, 1);
        }

        if(stripos($wor_refers_to_per, ',')){
            $wor_refers_to_per = '"'.$wor_refers_to_per.'"';
        }

//      Create field all_validity.

        $all_validity = '84'; //Value matching "Vérifié" in taxonomy "Validité".

//      Create field all_remark from other fields (prepared above):
wor_has_author_per, wor_has_publisher_per, wor_has_other_creator_per and
wor_refers_to_per.

        if($all_remark_verified == 'verified'){
            $all_remark = '"Davantage de personnes ou de collectivités sont liées
à cette oeuvre en tant qu\'auteurs, créateurs, éditeurs, etc. Voir la notice de la
Bibliothèque nationale de France."';
        }

//      Combine the fields in CSV format.

        $output .=
$wor_record_type.','.$wor_document_type.','.$wor_title.','.$wor_volume_titles.','.
$wor_variant_of_title.','.$wor_responsibility.','.$wor_edition.','.$wor_address.',
'.$wor_date.','.$wor_assumed_date.','.$wor_end_date.','.$wor_assumed_end_date.','.
$wor_description.','.$wor_periodicity.','.$wor_subject.','.$wor_bnf_record.','.$wo
```

```
r_bnf_record_number.','.$wor_issn.','.$wor_data_origin_bnf.','.$all_remark_bnf.','
.$wor_has_author_per.','.$wor_has_publisher_per.','.$wor_has_other_creator_per.','
.$wor_refers_to_per.','.$all_validity.','.$all_remark.'<br/>';
        $total_displayed ++;

    }
}

//Echo the number of results.

echo 'Nombre de résultats : '.$total_displayed.'<br/>';
echo 'Affichage des résultats '.$start_row.' à '.$end_row.'<br/><br/>';

//Output in CSV format.

echo
'wor_record_type,wor_document_type,wor_title,wor_volume_titles,wor_variant_of_titl
e,wor_responsibility,wor_edition,wor_address,wor_date,wor_assumed_date,wor_end_dat
e,wor_assumed_end_date,wor_description,wor_periodicity,wor_subject,wor_bnf_record,
wor_bnf_record_number,wor_issn,wor_data_origin_bnf,all_remark_bnf,wor_has_author_p
er,wor_has_publisher_per,wor_has_other_creator_per,wor_refers_to_per,all_validity,
all_remark<br/>';
echo $output;

?>
```

c. Copier les données depuis un navigateur web et coller ces données dans de nouveaux fichiers CSV vides (20 000 lignes par fichier).
d. Importer les données dans *Drupal* par tranches de 20 000 lignes, grâce à l'outil d'importation du module Webform.

GeoNames

DONNÉES RELATIVES AUX LIEUX

Objectifs

Nous importons des données relatives à l'ensemble des lieux sur terre, à partir de la base *GeoNames* : https://www.geonames.org (7 avril 2023). Ces données sont destinées à composer des fiches de lieux auxquelles divers éléments de notre base de données sont liés pour signaler, par exemple, de quel lieu provient tel texte ou quel lieu représente telle illustration. Comprenant les coordonnées géographiques des lieux, les informations importées permettent en outre de générer des cartes.

Données concernées

L'ensemble des données disponibles en téléchargement gratuit, relatives à plus de onze millions de noms de lieux le 7 février 2023. Pour chaque lieu, les champs sont les suivants :

- admin1_code : division administrative de niveau un.
- admin2_code : division administrative de niveau deux.
- admin3_code : division administrative de niveau trois.
- admin4_code : division administrative de niveau quatre.
- alternatenames : noms alternatifs.

- asciiname : nom de lieu dans la norme de l'*American standard code for information interchange*.
- cc2 : code alternatif du pays, selon la norme ISO-3166.
- country_code : code du pays, selon la norme ISO-3166.
- dem : valeur d'un modèle numérique d'élévation (GTOPO30 ou SRTM3).
- elevation : altitude en mètres.
- feature_class : classe de lieu.
- feature_code : type de lieu.
- geonameid : identifiant *GeoNames*.
- latitude : la latitude en degrés décimaux.
- longitude : la longitude en degrés décimaux.
- modification_date : date de la dernière modification dans la base *GeoNames*.
- name : nom de lieu en format UTF-8.
- population : nombre d'habitants.
- timezone : fuseau horaire.

L'ensemble des données provient du fichier allCountries.zip. Dans la perspective lier les lieux aux divisions administratives qui les comprennent, nous récupérons également les fichiers admin1CodesASCII.txt et admin2Codes.txt.

Utilisation

Les données de *GeoNames* sont modifiées et enrichies avant l'importation. En particulier, nous explicitons des données qui apparaissent dans le fichier allCountries.zip sous forme de codes. Par exemple, nous explicitons en français et en toutes lettres les codes du champ feature_code, de manière qu'un utilisateur francophone les comprenne immédiatement. Les modifications apportées aux données de *GeoNames* sont systématiques : elles interviennent au moment de modifier en PHP ces données pour créer des fichiers CSV importables dans la base de données *Drupal* (voir la procédure détaillée ci-dessous). En principe, nous n'apportons pas d'autres modifications aux données concernées après leur importation.

Sur le site public, la plupart des données demeureront invisibles. Seules les données liées à d'autres entités (un texte, par exemple) seront accessibles. Une fois chargées dans la base *Drupal*, les données ne peuvent plus être réimportées et elles ne profitent plus des mises à jour auxquelles procède l'équipe de *GeoNames*. Que les données soient modifiées ou non, leur source est toujours indiquée et un lien hypertexte permet d'accéder à la fiche du lieu sur le site *GeoNames*.

Procédure détaillée

1. Exportation des données relatives aux œuvres.
    a. Accéder à http://download.geonames.org/export/dump
    b. Télécharger les fichiers allCountries.zip, admin1CodesASCII.txt, admin2Codes.txt et timeZones.txt. Extraire allCountries.txt du fichier Zip. Le contenu des trois autres fichiers sera intégré plus tard au code PHP de transformation des données.
2. Importer le fichier allCountries.txt sur un serveur MariaDB 10.
    a. Dans une base de données « geonames », préconfigurer une table « allCountries » contenant les colonnes qui vont accueillir les données de *GeoNames*. Exemple de commande SQL :

```
CREATE TABLE allCountries (
    geonameid INT(11),
```

```
    name VARCHAR (200),
    asciiname VARCHAR(200),
    alternatenames TEXT(100000),
    latitude DECIMAL(10,7),
    longitude DECIMAL(10,7),
    feature_class CHAR(1),
    feature_code VARCHAR(10),
    country_code CHAR(2),
    cc2 CHAR(255),
    admin1_code VARCHAR(20),
    admin2_code VARCHAR(80),
    admin3_code VARCHAR(20),
    admin4_code VARCHAR(20),
    population BIGINT(20),
    elevation VARCHAR(100),
    dem INT(11),
    timezone VARCHAR(100),
    modification_date DATE
);
```

b. Définir un index sur la colonne « geonameid ». Exemple de commande SQL :

```
CREATE INDEX geonameid ON allCountries (geonameid);
```

c. Modifier la valeur d'interclassement de la colonne « alternatenames » en utf8mb4_unicode_ci. Exemple de commande SQL :

```
ALTER TABLE allCountries MODIFY
    alternatenames TEXT(100000)
      CHARACTER SET utf8mb4
      COLLATE utf8mb4_unicode_ci;
```

d. Importer l'ensemble des données. Exemple de commande SQL :

```
LOAD DATA INFILE '/path/to/allCountries.txt'
    INTO TABLE allCountries
    CHARACTER SET utf8mb4
    FIELDS TERMINATED BY "\t"
    LINES TERMINATED BY '\n';
```

e. Devant la colonne « col_2 », ajouter une colonne « col_1 » de type INT.
f. Ajouter une valeur autoincrémentielle à « col_1 » et définir cette colonne comme la clef primaire de la table. Exemple de commande SQL :

```
ALTER TABLE allCountries
MODIFY col_1 INT NOT NULL AUTO_INCREMENT PRIMARY KEY;
```

3. À présent, nous sommes en mesure de créer un fichier CSV importable dans la base de données définitive. Il s'agit de retoucher les données brutes de *GeoNames*, de façon à permettre leur importation dans les champs correspondants de la base de données de notre site, grâce à la fonction d'importation du module Webform de *Drupal*. Pour ce faire, nous appelons les données brutes rassemblées dans la table « allCountries » de la base de données « geonames », nous les modifions en langage PHP, puis nous affichons sur une page web l'ensemble des données modifiées, de manière à pouvoir les copier et les coller dans un nouveau fichier CSV. Il convient de procéder en quatre temps :

nous importons d'abord les pays, ensuite les divisions administratives de niveau 1, puis les divisions administratives de niveau 2, enfin le reste des lieux.

a. À chaque étape de l'importation, nous allons utiliser un fichier « csv_geonames_place.php » dont voici un exemple de code :

```php
<?php
//Places

//Determine the first and last row to be echoed. For $end_row, the value 0 will
echo all the rows.

$start_row = 0;
$end_row = 0;

//Connect to the database.

$mysqli = new MySQLi('localhost','user','password','database');

if(mysqli_connect_errno())
    {
        printf('Échec de la connexion : base de données momentanément
inaccessible.');
        exit();
    }

//Total number of items.

$result_set_1 = $mysqli->query('SELECT COUNT(*) FROM allCountries');
$total = $result_set_1->fetch_row();

if($end_row == 0){
    $end_row = $total[0];
}

//Query the database.

$result_set_2 = $mysqli->query('SELECT * FROM allCountries WHERE col_1 >=
'.$start_row.' AND col_1 <= '.$end_row);

//Retrieve and manage data.

$tmpCount = 0;
$total_displayed = 0;
$output = NULL;

if($result_set_2->num_rows > 0){
    while(null !==($rows = $result_set_2->fetch_assoc()))
    {

//      Assign NULL to the variables.

        $all_validity = $combine_country_and_admin1_codes =
$combine_country_admin1_and_admin2_codes = $country_display = $data_display =
$feature_code_check_1 = $feature_code_check_2 = $feature_code_check_3 =
$feature_codes_to_exclude_1 = $feature_codes_to_exclude_2 =
$feature_codes_to_exclude_3 = $feature_code_value_1 = $feature_code_value_2 =
$feature_code_value_3 = $geonameid_check = $geonameid_check_2 =
$pla_geonames_admin1_code = $pla_geonames_admin2_code = $pla_geonames_admin3_code
```

```
= $pla_geonames_admin4_code = $pla_geonames_alternatenames =
$pla_geonames_asciiname = $pla_geonames_cc2 = $pla_geonames_dem =
$pla_geonames_country_code = $pla_geon_country_code_explicit =
$pla_geonames_elevation = $pla_geonames_feature_class = $pla_geonames_feature_code
= $pla_geon_feature_code_explicit = $pla_geonames_geonameid =
$pla_geonames_latitude = $pla_geonames_longitude = $pla_geonames_modification_date
= $pla_geonames_name = $pla_geonames_population = $pla_geonames_record =
$pla_geonames_timezone = $pla_is_part_of_admin1_pla = $pla_is_part_of_admin2_pla =
$pla_is_part_of_country_pla = $show_error_message = NULL;

//      Count the lines and retrieve data from the database.

        $tmpCount ++;
        $geonameid = $rows['geonameid'];
        $name = $rows['name'];
        $asciiname = $rows['asciiname'];
        $alternatenames = $rows['alternatenames'];
        $latitude = $rows['latitude'];
        $longitude = $rows['longitude'];
        $feature_class = $rows['feature_class'];
        $feature_code = $rows['feature_code'];
        $country_code = $rows['country_code'];
        $cc2 = $rows['cc2'];
        $admin1_code = $rows['admin1_code'];
        $admin2_code = $rows['admin2_code'];
        $admin3_code = $rows['admin3_code'];
        $admin4_code = $rows['admin4_code'];
        $population = $rows['population'];
        $elevation = $rows['elevation'];
        $dem = $rows['dem'];
        $timezone = $rows['timezone'];
        $modification_date = $rows['modification_date'];

//      Manage and escape double-quotes.

        $data =
$geonameid.'[tab]'.$name.'[tab]'.$asciiname.'[tab]'.$alternatenames.'[tab]'.$latit
ude.'[tab]'.$longitude.'[tab]'.$feature_class.'[tab]'.$feature_code.'[tab]'.$count
ry_code.'[tab]'.$cc2.'[tab]'.$admin1_code.'[tab]'.$admin2_code.'[tab]'.$admin3_cod
e.'[tab]'.$admin4_code.'[tab]'.$population.'[tab]'.$elevation.'[tab]'.$dem.'[tab]'
.$timezone.'[tab]'.$modification_date;

        $data = str_replace('"', '""', $data);
        $data = str_replace('« ', '""', $data);
        $data = str_replace('«', '""', $data);
        $data = str_replace(' »', '""', $data);
        $data = str_replace('»', '""', $data);

        $data = explode('[tab]', $data);

        $geonameid = $data[0];
        $name = $data[1];
        $asciiname = $data[2];
        $alternatenames = $data[3];
        $latitude = $data[4];
        $longitude = $data[5];
        $feature_class = $data[6];
        $feature_code = $data[7];
        $country_code = $data[8];
```

```
        $cc2 = $data[9];
        $admin1_code = $data[10];
        $admin2_code = $data[11];
        $admin3_code = $data[12];
        $admin4_code = $data[13];
        $population = $data[14];
        $elevation = $data[15];
        $dem = $data[16];
        $timezone = $data[17];
        $modification_date = $data[18];

//      Create field pla_geonames_geonameid from geonameid.

        $pla_geonames_geonameid = $geonameid;

//      Create field pla_geonames_record from previous field.

        $pla_geonames_record =
'https://www.geonames.org/'.$pla_geonames_geonameid;

//      Create field pla_geonames_name from name.

        $pla_geonames_name = $name;

        if(stripos($pla_geonames_name, ',') !== false ||
stripos($pla_geonames_name, '"') !== false){
            $pla_geonames_name = '"'.$pla_geonames_name.'"';
        }

//      Create field pla_geonames_asciiname from asciiname.

        $pla_geonames_asciiname = $asciiname;

        if(stripos($pla_geonames_asciiname, ',') !== false ||
stripos($pla_geonames_asciiname, '"') !== false){
            $pla_geonames_asciiname = '"'.$pla_geonames_asciiname.'"';
        }

//      Create field pla_geonames_alternatenames from asciiname.

        $pla_geonames_alternatenames = $alternatenames;

        if(stripos($pla_geonames_alternatenames, ',') !== false ||
stripos($pla_geonames_alternatenames, '"') !== false){
            $pla_geonames_alternatenames = '"'.$pla_geonames_alternatenames.'"';
        }

//      Create field pla_geonames_latitude from latitude.

        $pla_geonames_latitude = $latitude;

//      Create field pla_geonames_longitude from longitude.

        $pla_geonames_longitude = $longitude;

//      Create field pla_geonames_feature_class from feature_class.

        $pla_geonames_feature_class = $feature_class;
```

```
//      Create field pla_geonames_feature_code from feature_code.

        $pla_geonames_feature_code = $feature_code;

//      Create field pla_geon_feature_code_explicit from previous field, according
to "Type de lieu" taxonomy.

        if($pla_geonames_feature_code == 'ADM1H'){
            $pla_geon_feature_code_explicit = '8236'; //Value matching "Ancienne
division administrative de premier ordre".
        }
        elseif($pla_geonames_feature_code == 'ADM1'){
            $pla_geon_feature_code_explicit = '8237'; //Value matching "Division
administrative de premier ordre".
        }
        elseif($pla_geonames_feature_code == 'ADM2H'){
            $pla_geon_feature_code_explicit = '8238'; //Value matching "Ancienne
division administrative de deuxième ordre".
        }
        elseif($pla_geonames_feature_code == 'ADM2'){
            $pla_geon_feature_code_explicit = '8239'; //Value matching "Division
administrative de deuxième ordre".
        }
        elseif($pla_geonames_feature_code == 'ADM3H'){
            $pla_geon_feature_code_explicit = '8240'; //Value matching "Ancienne
division administrative de troisième ordre".
        }
        elseif($pla_geonames_feature_code == 'ADM3'){
            $pla_geon_feature_code_explicit = '8241'; //Value matching "Division
administrative de troisième ordre".
        }
        elseif($pla_geonames_feature_code == 'ADM4H'){
            $pla_geon_feature_code_explicit = '8242'; //Value matching "Ancienne
division administrative de quatrième ordre".
        }
        elseif($pla_geonames_feature_code == 'ADM4'){
            $pla_geon_feature_code_explicit = '8243'; //Value matching "Division
administrative de quatrième ordre".
        }
        elseif($pla_geonames_feature_code == 'ADM5H'){
            $pla_geon_feature_code_explicit = '8244'; //Value matching "Ancienne
division administrative de cinquième ordre".
        }
        elseif($pla_geonames_feature_code == 'ADM5'){
            $pla_geon_feature_code_explicit = '8245'; //Value matching "Division
administrative de cinquième ordre".
        }
        elseif($pla_geonames_feature_code == 'ADMDH'){
            $pla_geon_feature_code_explicit = '8246'; //Value matching "Ancienne
division administrative".
        }
        elseif($pla_geonames_feature_code == 'ADMD'){
            $pla_geon_feature_code_explicit = '8247'; //Value matching "Division
administrative".
        }
        elseif($pla_geonames_feature_code == 'LTER'){
            $pla_geon_feature_code_explicit = '8248'; //Value matching "Zone
louée".
        }
```

```
        elseif($pla_geonames_feature_code == 'PCLD'){
            $pla_geon_feature_code_explicit = '8249'; //Value matching "Entité
politique dépendante".
        }
        elseif($pla_geonames_feature_code == 'PCLF'){
            $pla_geon_feature_code_explicit = '8250'; //Value matching "État
librement associé".
        }
        elseif($pla_geonames_feature_code == 'PCLH'){
            $pla_geon_feature_code_explicit = '8251'; //Value matching "Ancienne
entité politique".
        }
        elseif($pla_geonames_feature_code == 'PCLIX'){
            $pla_geon_feature_code_explicit = '8252'; //Value matching "Section
d'une entité politique indépendante".
        }
        elseif($pla_geonames_feature_code == 'PCLI'){
            $pla_geon_feature_code_explicit = '8253'; //Value matching "Entité
politique indépendante".
        }
        elseif($pla_geonames_feature_code == 'PCLS'){
            $pla_geon_feature_code_explicit = '8254'; //Value matching "Entité
politique semi-indépendante".
        }
        elseif($pla_geonames_feature_code == 'PCL'){
            $pla_geon_feature_code_explicit = '8255'; //Value matching "Entité
politique".
        }
        elseif($pla_geonames_feature_code == 'PRSH'){
            $pla_geon_feature_code_explicit = '8256'; //Value matching "Paroisse".
        }
        elseif($pla_geonames_feature_code == 'TERR'){
            $pla_geon_feature_code_explicit = '8257'; //Value matching
"Territoire".
        }
        elseif($pla_geonames_feature_code == 'ZNB'){
            $pla_geon_feature_code_explicit = '8258'; //Value matching "Zone
tampon".
        }
        elseif($pla_geonames_feature_code == 'ZN'){
            $pla_geon_feature_code_explicit = '8259'; //Value matching "Zone".
        }
        elseif($pla_geonames_feature_code == 'AIRS'){
            $pla_geon_feature_code_explicit = '8260'; //Value matching "Aire
d'atterrissage pour hydravion".
        }
        elseif($pla_geonames_feature_code == 'ANCH'){
            $pla_geon_feature_code_explicit = '8261'; //Value matching "Zone
d'ancrage".
        }
        elseif($pla_geonames_feature_code == 'BAY'){
            $pla_geon_feature_code_explicit = '8262'; //Value matching "Baie".
        }
        elseif($pla_geonames_feature_code == 'BAYS'){
            $pla_geon_feature_code_explicit = '8263'; //Value matching "Baies".
        }
        elseif($pla_geonames_feature_code == 'BGHT'){
            $pla_geon_feature_code_explicit = '8264'; //Value matching "Baie(s)".
        }
```

```
        elseif($pla_geonames_feature_code == 'BNK'){
            $pla_geon_feature_code_explicit = '8265'; //Value matching "Banc(s)".
        }
        elseif($pla_geonames_feature_code == 'BNKR'){
            $pla_geon_feature_code_explicit = '8266'; //Value matching "Berge".
        }
        elseif($pla_geonames_feature_code == 'BNKX'){
            $pla_geon_feature_code_explicit = '8267'; //Value matching "Section
d'un banc".
        }
        elseif($pla_geonames_feature_code == 'BOG'){
            $pla_geon_feature_code_explicit = '8268'; //Value matching
"Tourbière".
        }
        elseif($pla_geonames_feature_code == 'CAPG'){
            $pla_geon_feature_code_explicit = '8269'; //Value matching "Calotte
glaciaire".
        }
        elseif($pla_geonames_feature_code == 'CHN'){
            $pla_geon_feature_code_explicit = '8270'; //Value matching "Chenal".
        }
        elseif($pla_geonames_feature_code == 'CHNL'){
            $pla_geon_feature_code_explicit = '8271'; //Value matching "Chenal
lacustre".
        }
        elseif($pla_geonames_feature_code == 'CHNM'){
            $pla_geon_feature_code_explicit = '8272'; //Value matching "Chenal
maritime".
        }
        elseif($pla_geonames_feature_code == 'CHNN'){
            $pla_geon_feature_code_explicit = '8273'; //Value matching "Chenal de
navigation".
        }
        elseif($pla_geonames_feature_code == 'CNFL'){
            $pla_geon_feature_code_explicit = '8274'; //Value matching
"Confluence".
        }
        elseif($pla_geonames_feature_code == 'CNL'){
            $pla_geon_feature_code_explicit = '8275'; //Value matching "Canal".
        }
        elseif($pla_geonames_feature_code == 'CNLA'){
            $pla_geon_feature_code_explicit = '8276'; //Value matching "Aqueduc".
        }
        elseif($pla_geonames_feature_code == 'CNLB'){
            $pla_geon_feature_code_explicit = '8277'; //Value matching "Coude de
canal".
        }
        elseif($pla_geonames_feature_code == 'CNLD'){
            $pla_geon_feature_code_explicit = '8278'; //Value matching "Canal de
drainage".
        }
        elseif($pla_geonames_feature_code == 'CNLI'){
            $pla_geon_feature_code_explicit = '8279'; //Value matching "Canal
d'irrigation".
        }
        elseif($pla_geonames_feature_code == 'CNLN'){
            $pla_geon_feature_code_explicit = '8280'; //Value matching "Canal de
navigation".
        }
```

```
        elseif($pla_geonames_feature_code == 'CNLQ'){
            $pla_geon_feature_code_explicit = '8281'; //Value matching "Canal
abandonné".
        }
        elseif($pla_geonames_feature_code == 'CNLSB'){
            $pla_geon_feature_code_explicit = '8282'; //Value matching "Canal
d'irrigation souterrain".
        }
        elseif($pla_geonames_feature_code == 'CNLX'){
            $pla_geon_feature_code_explicit = '8283'; //Value matching "Section de
canal".
        }
        elseif($pla_geonames_feature_code == 'COVE'){
            $pla_geon_feature_code_explicit = '8284'; //Value matching "Petite
baie".
        }
        elseif($pla_geonames_feature_code == 'CRKT'){
            $pla_geon_feature_code_explicit = '8285'; //Value matching "Crique de
marée".
        }
        elseif($pla_geonames_feature_code == 'CRNT'){
            $pla_geon_feature_code_explicit = '8286'; //Value matching "Courant".
        }
        elseif($pla_geonames_feature_code == 'CUTF'){
            $pla_geon_feature_code_explicit = '8287'; //Value matching "Chenal
traversant des méandres".
        }
        elseif($pla_geonames_feature_code == 'DCK'){
            $pla_geon_feature_code_explicit = '8288'; //Value matching "Dock(s)".
        }
        elseif($pla_geonames_feature_code == 'DCKB'){
            $pla_geon_feature_code_explicit = '8289'; //Value matching "Bassin
d'amarrage".
        }
        elseif($pla_geonames_feature_code == 'DOMG'){
            $pla_geon_feature_code_explicit = '8290'; //Value matching "Partie
élevée d'une calotte glaciaire".
        }
        elseif($pla_geonames_feature_code == 'DPRG'){
            $pla_geon_feature_code_explicit = '8291'; //Value matching "Partie
basse d'une calotte glaciaire".
        }
        elseif($pla_geonames_feature_code == 'DTCH'){
            $pla_geon_feature_code_explicit = '8292'; //Value matching "Fossé de
drainage ou d'irrigation".
        }
        elseif($pla_geonames_feature_code == 'DTCHD'){
            $pla_geon_feature_code_explicit = '8293'; //Value matching "Fossé de
drainage".
        }
        elseif($pla_geonames_feature_code == 'DTCHI'){
            $pla_geon_feature_code_explicit = '8294'; //Value matching "Fossé
d'irrigation".
        }
        elseif($pla_geonames_feature_code == 'DTCHM'){
            $pla_geon_feature_code_explicit = '8295'; //Value matching "Bouche
d'un fossé de drainage".
        }
        elseif($pla_geonames_feature_code == 'ESTY'){
```

```
            $pla_geon_feature_code_explicit = '8296'; //Value matching "Estuaire".
        }
        elseif($pla_geonames_feature_code == 'FISH'){
            $pla_geon_feature_code_explicit = '8297'; //Value matching "Zone de
pêche".
        }
        elseif($pla_geonames_feature_code == 'FJD'){
            $pla_geon_feature_code_explicit = '8298'; //Value matching "Fjord".
        }
        elseif($pla_geonames_feature_code == 'FJDS'){
            $pla_geon_feature_code_explicit = '8299'; //Value matching "Fjords".
        }
        elseif($pla_geonames_feature_code == 'FLLS'){
            $pla_geon_feature_code_explicit = '8300'; //Value matching "Chute(s)
d'eau".
        }
        elseif($pla_geonames_feature_code == 'FLLSX'){
            $pla_geon_feature_code_explicit = '8301'; //Value matching "Section de
chute(s) d'eau".
        }
        elseif($pla_geonames_feature_code == 'FLTM'){
            $pla_geon_feature_code_explicit = '8302'; //Value matching "Étendue de
boue".
        }
        elseif($pla_geonames_feature_code == 'FLTT'){
            $pla_geon_feature_code_explicit = '8303'; //Value matching "Grande
étendue de boue ou de sable produite par la marée".
        }
        elseif($pla_geonames_feature_code == 'GLCR'){
            $pla_geon_feature_code_explicit = '8304'; //Value matching
"Glacier(s)".
        }
        elseif($pla_geonames_feature_code == 'GULF'){
            $pla_geon_feature_code_explicit = '8305'; //Value matching "Golfe".
        }
        elseif($pla_geonames_feature_code == 'GYSR'){
            $pla_geon_feature_code_explicit = '8306'; //Value matching "Geyser".
        }
        elseif($pla_geonames_feature_code == 'HBR'){
            $pla_geon_feature_code_explicit = '8307'; //Value matching "Havre(s)".
        }
        elseif($pla_geonames_feature_code == 'HBRX'){
            $pla_geon_feature_code_explicit = '8308'; //Value matching "Section de
port".
        }
        elseif($pla_geonames_feature_code == 'INLT'){
            $pla_geon_feature_code_explicit = '8309'; //Value matching "Grau".
        }
        elseif($pla_geonames_feature_code == 'INLTQ'){
            $pla_geon_feature_code_explicit = '8310'; //Value matching "Ancien
grau".
        }
        elseif($pla_geonames_feature_code == 'LBED'){
            $pla_geon_feature_code_explicit = '8311'; //Value matching "Lac(s)
asséché(s)".
        }
        elseif($pla_geonames_feature_code == 'LGN'){
            $pla_geon_feature_code_explicit = '8312'; //Value matching "Lagune".
        }
```

```
        elseif($pla_geonames_feature_code == 'LGNS'){
            $pla_geon_feature_code_explicit = '8313'; //Value matching "Lagunes".
        }
        elseif($pla_geonames_feature_code == 'LGNX'){
            $pla_geon_feature_code_explicit = '8314'; //Value matching "Section de
lagune".
        }
        elseif($pla_geonames_feature_code == 'LK'){
            $pla_geon_feature_code_explicit = '8315'; //Value matching "Lac".
        }
        elseif($pla_geonames_feature_code == 'LKC'){
            $pla_geon_feature_code_explicit = '8316'; //Value matching "Lac de
cratère".
        }
        elseif($pla_geonames_feature_code == 'LKI'){
            $pla_geon_feature_code_explicit = '8317'; //Value matching "Lac
intermittent".
        }
        elseif($pla_geonames_feature_code == 'LKN'){
            $pla_geon_feature_code_explicit = '8318'; //Value matching "Lac salé".
        }
        elseif($pla_geonames_feature_code == 'LKNI'){
            $pla_geon_feature_code_explicit = '8319'; //Value matching "Lac salé
intermittent".
        }
        elseif($pla_geonames_feature_code == 'LKO'){
            $pla_geon_feature_code_explicit = '8320'; //Value matching "Bras-
mort".
        }
        elseif($pla_geonames_feature_code == 'LKOI'){
            $pla_geon_feature_code_explicit = '8321'; //Value matching "Bras-mort
intermittent".
        }
        elseif($pla_geonames_feature_code == 'LKS'){
            $pla_geon_feature_code_explicit = '8322'; //Value matching "Lacs".
        }
        elseif($pla_geonames_feature_code == 'LKSB'){
            $pla_geon_feature_code_explicit = '8323'; //Value matching "Lac
souterrain".
        }
        elseif($pla_geonames_feature_code == 'LKSC'){
            $pla_geon_feature_code_explicit = '8324'; //Value matching "Lacs de
cratère".
        }
        elseif($pla_geonames_feature_code == 'LKSI'){
            $pla_geon_feature_code_explicit = '8325'; //Value matching "Lacs
intermittents".
        }
        elseif($pla_geonames_feature_code == 'LKSN'){
            $pla_geon_feature_code_explicit = '8326'; //Value matching "Lacs
salés".
        }
        elseif($pla_geonames_feature_code == 'LKSNI'){
            $pla_geon_feature_code_explicit = '8327'; //Value matching "Lacs salés
intermittents".
        }
        elseif($pla_geonames_feature_code == 'LKX'){
            $pla_geon_feature_code_explicit = '8328'; //Value matching "Section de
lac".
```

```
        }
        elseif($pla_geonames_feature_code == 'MFGN'){
            $pla_geon_feature_code_explicit = '8329'; //Value matching "Bassins
d'évaporation de sel".
        }
        elseif($pla_geonames_feature_code == 'MGV'){
            $pla_geon_feature_code_explicit = '8330'; //Value matching "Mangrove".
        }
        elseif($pla_geonames_feature_code == 'MOOR'){
            $pla_geon_feature_code_explicit = '8331'; //Value matching "Lande(s)
(moor)".
        }
        elseif($pla_geonames_feature_code == 'MRSH'){
            $pla_geon_feature_code_explicit = '8332'; //Value matching "Marais".
        }
        elseif($pla_geonames_feature_code == 'MRSHN'){
            $pla_geon_feature_code_explicit = '8333'; //Value matching "Marais
salant".
        }
        elseif($pla_geonames_feature_code == 'NRWS'){
            $pla_geon_feature_code_explicit = '8334'; //Value matching "Goulets".
        }
        elseif($pla_geonames_feature_code == 'OCN'){
            $pla_geon_feature_code_explicit = '8335'; //Value matching "Océan".
        }
        elseif($pla_geonames_feature_code == 'OVF'){
            $pla_geon_feature_code_explicit = '8336'; //Value matching "Raz de
courant".
        }
        elseif($pla_geonames_feature_code == 'POND'){
            $pla_geon_feature_code_explicit = '8337'; //Value matching "Étang".
        }
        elseif($pla_geonames_feature_code == 'PONDI'){
            $pla_geon_feature_code_explicit = '8338'; //Value matching "Étang
intermittent".
        }
        elseif($pla_geonames_feature_code == 'PONDN'){
            $pla_geon_feature_code_explicit = '8339'; //Value matching "Étang
salé".
        }
        elseif($pla_geonames_feature_code == 'PONDNI'){
            $pla_geon_feature_code_explicit = '8340'; //Value matching "Étang(s)
salé(s) intermittent(s)".
        }
        elseif($pla_geonames_feature_code == 'PONDS'){
            $pla_geon_feature_code_explicit = '8341'; //Value matching "Étangs".
        }
        elseif($pla_geonames_feature_code == 'PONDSF'){
            $pla_geon_feature_code_explicit = '8342'; //Value matching "Étangs à
poissons".
        }
        elseif($pla_geonames_feature_code == 'PONDSI'){
            $pla_geon_feature_code_explicit = '8343'; //Value matching "Étangs
intermittents".
        }
        elseif($pla_geonames_feature_code == 'PONDSN'){
            $pla_geon_feature_code_explicit = '8344'; //Value matching "Étangs
salés".
        }
```

```
        elseif($pla_geonames_feature_code == 'POOL'){
            $pla_geon_feature_code_explicit = '8345'; //Value matching
"Bassin(s)".
        }
        elseif($pla_geonames_feature_code == 'POOLI'){
            $pla_geon_feature_code_explicit = '8346'; //Value matching "Bassin
intermittent".
        }
        elseif($pla_geonames_feature_code == 'RCH'){
            $pla_geon_feature_code_explicit = '8347'; //Value matching "Passage
navigable".
        }
        elseif($pla_geonames_feature_code == 'RDGG'){
            $pla_geon_feature_code_explicit = '8348'; //Value matching "Crête de
calotte glaciaire".
        }
        elseif($pla_geonames_feature_code == 'RDST'){
            $pla_geon_feature_code_explicit = '8349'; //Value matching "Rade".
        }
        elseif($pla_geonames_feature_code == 'RF'){
            $pla_geon_feature_code_explicit = '8350'; //Value matching "Récif(s)".
        }
        elseif($pla_geonames_feature_code == 'RFC'){
            $pla_geon_feature_code_explicit = '8351'; //Value matching "Récif(s)
corallien(s)".
        }
        elseif($pla_geonames_feature_code == 'RFX'){
            $pla_geon_feature_code_explicit = '8352'; //Value matching "Section de
récif".
        }
        elseif($pla_geonames_feature_code == 'RPDS'){
            $pla_geon_feature_code_explicit = '8353'; //Value matching "Rapides".
        }
        elseif($pla_geonames_feature_code == 'RSV'){
            $pla_geon_feature_code_explicit = '8354'; //Value matching
"Réservoir(s)".
        }
        elseif($pla_geonames_feature_code == 'RSVI'){
            $pla_geon_feature_code_explicit = '8355'; //Value matching "Réservoir
intermittent".
        }
        elseif($pla_geonames_feature_code == 'RSVT'){
            $pla_geon_feature_code_explicit = '8356'; //Value matching "Réservoir
d'eau, citerne".
        }
        elseif($pla_geonames_feature_code == 'RVN'){
            $pla_geon_feature_code_explicit = '8357'; //Value matching "Ravin(s)".
        }
        elseif($pla_geonames_feature_code == 'SBKH'){
            $pla_geon_feature_code_explicit = '8358'; //Value matching "Sebkha".
        }
        elseif($pla_geonames_feature_code == 'SD'){
            $pla_geon_feature_code_explicit = '8359'; //Value matching "Bras de
mer".
        }
        elseif($pla_geonames_feature_code == 'SEA'){
            $pla_geon_feature_code_explicit = '8360'; //Value matching "Mer".
        }
        elseif($pla_geonames_feature_code == 'SHOL'){
```

```
            $pla_geon_feature_code_explicit = '8361'; //Value matching "Banc(s) de
sable".
        }
        elseif($pla_geonames_feature_code == 'SILL'){
            $pla_geon_feature_code_explicit = '8362'; //Value matching "Seuil
aquatique".
        }
        elseif($pla_geonames_feature_code == 'SPNG'){
            $pla_geon_feature_code_explicit = '8363'; //Value matching
"Source(s)".
        }
        elseif($pla_geonames_feature_code == 'SPNS'){
            $pla_geon_feature_code_explicit = '8364'; //Value matching "Source(s)
sulfureuse(s)".
        }
        elseif($pla_geonames_feature_code == 'SPNT'){
            $pla_geon_feature_code_explicit = '8365'; //Value matching "Source(s)
thermale(s)".
        }
        elseif($pla_geonames_feature_code == 'STM'){
            $pla_geon_feature_code_explicit = '8366'; //Value matching "Cours
d'eau (stream)".
        }
        elseif($pla_geonames_feature_code == 'STMA'){
            $pla_geon_feature_code_explicit = '8367'; //Value matching
"Anabranche".
        }
        elseif($pla_geonames_feature_code == 'STMB'){
            $pla_geon_feature_code_explicit = '8368'; //Value matching "Coude d'un
cours d'eau".
        }
        elseif($pla_geonames_feature_code == 'STMC'){
            $pla_geon_feature_code_explicit = '8369'; //Value matching "Cours
d'eau canalisé".
        }
        elseif($pla_geonames_feature_code == 'STMD'){
            $pla_geon_feature_code_explicit = '8370'; //Value matching
"Défluent(s)".
        }
        elseif($pla_geonames_feature_code == 'STMH'){
            $pla_geon_feature_code_explicit = '8371'; //Value matching "Source
d'un cours d'eau".
        }
        elseif($pla_geonames_feature_code == 'STMI'){
            $pla_geon_feature_code_explicit = '8372'; //Value matching "Cours
d'eau intermittent".
        }
        elseif($pla_geonames_feature_code == 'STMIX'){
            $pla_geon_feature_code_explicit = '8373'; //Value matching "Section
d'un cours d'eau intermittent".
        }
        elseif($pla_geonames_feature_code == 'STMM'){
            $pla_geon_feature_code_explicit = '8374'; //Value matching
"Embouchure(s) d'un cours d'eau".
        }
        elseif($pla_geonames_feature_code == 'STMQ'){
            $pla_geon_feature_code_explicit = '8375'; //Value matching "Ancien
cours d'eau".
        }
```

```
        elseif($pla_geonames_feature_code == 'STMS'){
            $pla_geon_feature_code_explicit = '8376'; //Value matching "Plusieurs
cours d'eau".
        }
        elseif($pla_geonames_feature_code == 'STMSB'){
            $pla_geon_feature_code_explicit = '8377'; //Value matching "Perte
hydrologique".
        }
        elseif($pla_geonames_feature_code == 'STMX'){
            $pla_geon_feature_code_explicit = '8378'; //Value matching "Section
d'un cours d'eau".
        }
        elseif($pla_geonames_feature_code == 'STRT'){
            $pla_geon_feature_code_explicit = '8379'; //Value matching "Détroit".
        }
        elseif($pla_geonames_feature_code == 'SWMP'){
            $pla_geon_feature_code_explicit = '8380'; //Value matching "Marécage".
        }
        elseif($pla_geonames_feature_code == 'SYSI'){
            $pla_geon_feature_code_explicit = '8381'; //Value matching "Système
d'irrigation".
        }
        elseif($pla_geonames_feature_code == 'TNLC'){
            $pla_geon_feature_code_explicit = '8382'; //Value matching "Canal de
tunnel".
        }
        elseif($pla_geonames_feature_code == 'WAD'){
            $pla_geon_feature_code_explicit = '8383'; //Value matching "Oued".
        }
        elseif($pla_geonames_feature_code == 'WADB'){
            $pla_geon_feature_code_explicit = '8384'; //Value matching "Coude
d'oued".
        }
        elseif($pla_geonames_feature_code == 'WADJ'){
            $pla_geon_feature_code_explicit = '8385'; //Value matching "Jonction
d'oueds".
        }
        elseif($pla_geonames_feature_code == 'WADM'){
            $pla_geon_feature_code_explicit = '8386'; //Value matching "Embouchure
d'oued".
        }
        elseif($pla_geonames_feature_code == 'WADS'){
            $pla_geon_feature_code_explicit = '8387'; //Value matching "Oueds".
        }
        elseif($pla_geonames_feature_code == 'WADX'){
            $pla_geon_feature_code_explicit = '8388'; //Value matching "Section
d'oued".
        }
        elseif($pla_geonames_feature_code == 'WHRL'){
            $pla_geon_feature_code_explicit = '8389'; //Value matching
"Tourbillon".
        }
        elseif($pla_geonames_feature_code == 'WLL'){
            $pla_geon_feature_code_explicit = '8390'; //Value matching "Puits".
        }
        elseif($pla_geonames_feature_code == 'WLLQ'){
            $pla_geon_feature_code_explicit = '8391'; //Value matching "Puits
abandonné".
        }
```

```
        elseif($pla_geonames_feature_code == 'WLLS'){
            $pla_geon_feature_code_explicit = '8392'; //Value matching "Plusieurs
puits".
        }
        elseif($pla_geonames_feature_code == 'WTLD'){
            $pla_geon_feature_code_explicit = '8393'; //Value matching "Zone
humide".
        }
        elseif($pla_geonames_feature_code == 'WTLDI'){
            $pla_geon_feature_code_explicit = '8394'; //Value matching "Zone
humide intermittente".
        }
        elseif($pla_geonames_feature_code == 'WTRC'){
            $pla_geon_feature_code_explicit = '8395'; //Value matching "Cours
d'eau (watercourse)".
        }
        elseif($pla_geonames_feature_code == 'WTRH'){
            $pla_geon_feature_code_explicit = '8396'; //Value matching
"Abreuvoir(s)".
        }
        elseif($pla_geonames_feature_code == 'AGCR'){
            $pla_geon_feature_code_explicit = '8397'; //Value matching "Colonie
agricole".
        }
        elseif($pla_geonames_feature_code == 'AMUS'){
            $pla_geon_feature_code_explicit = '8398'; //Value matching "Parc
d'attraction".
        }
        elseif($pla_geonames_feature_code == 'AREA'){
            $pla_geon_feature_code_explicit = '8399'; //Value matching "Zone".
        }
        elseif($pla_geonames_feature_code == 'BSND'){
            $pla_geon_feature_code_explicit = '8400'; //Value matching "Bassin
versant".
        }
        elseif($pla_geonames_feature_code == 'BSNP'){
            $pla_geon_feature_code_explicit = '8401'; //Value matching "Bassin
pétrolier".
        }
        elseif($pla_geonames_feature_code == 'BTL'){
            $pla_geon_feature_code_explicit = '8402'; //Value matching "Champ de
bataille".
        }
        elseif($pla_geonames_feature_code == 'CLG'){
            $pla_geon_feature_code_explicit = '8403'; //Value matching
"Clairière".
        }
        elseif($pla_geonames_feature_code == 'CMN'){
            $pla_geon_feature_code_explicit = '8404'; //Value matching "Communal".
        }
        elseif($pla_geonames_feature_code == 'CNS'){
            $pla_geon_feature_code_explicit = '8405'; //Value matching "Zone de
concession".
        }
        elseif($pla_geonames_feature_code == 'COLF'){
            $pla_geon_feature_code_explicit = '8406'; //Value matching "Bassin
houiller".
        }
        elseif($pla_geonames_feature_code == 'CONT'){
```

```
            $pla_geon_feature_code_explicit = '8407'; //Value matching
"Continent".
        }
        elseif($pla_geonames_feature_code == 'CST'){
            $pla_geon_feature_code_explicit = '8408'; //Value matching "Côte".
        }
        elseif($pla_geonames_feature_code == 'CTRB'){
            $pla_geon_feature_code_explicit = '8409'; //Value matching "Centre
d'affaires".
        }
        elseif($pla_geonames_feature_code == 'DEVH'){
            $pla_geon_feature_code_explicit = '8410'; //Value matching
"Lotissement".
        }
        elseif($pla_geonames_feature_code == 'FLD'){
            $pla_geon_feature_code_explicit = '8411'; //Value matching "Champ(s)".
        }
        elseif($pla_geonames_feature_code == 'FLDI'){
            $pla_geon_feature_code_explicit = '8412'; //Value matching "Champ(s)
irrigué(s)".
        }
        elseif($pla_geonames_feature_code == 'GASF'){
            $pla_geon_feature_code_explicit = '8413'; //Value matching "Gisement
de gaz".
        }
        elseif($pla_geonames_feature_code == 'GRAZ'){
            $pla_geon_feature_code_explicit = '8414'; //Value matching "Zone de
pâturage".
        }
        elseif($pla_geonames_feature_code == 'GVL'){
            $pla_geon_feature_code_explicit = '8415'; //Value matching "Zone de
graviers".
        }
        elseif($pla_geonames_feature_code == 'INDS'){
            $pla_geon_feature_code_explicit = '8416'; //Value matching "Zone
industrielle".
        }
        elseif($pla_geonames_feature_code == 'LAND'){
            $pla_geon_feature_code_explicit = '8417'; //Value matching "Terre
arctique".
        }
        elseif($pla_geonames_feature_code == 'LCTY'){
            $pla_geon_feature_code_explicit = '8418'; //Value matching "Localité".
        }
        elseif($pla_geonames_feature_code == 'MILB'){
            $pla_geon_feature_code_explicit = '8419'; //Value matching "Base
militaire".
        }
        elseif($pla_geonames_feature_code == 'MNA'){
            $pla_geon_feature_code_explicit = '8420'; //Value matching "Zone
minière".
        }
        elseif($pla_geonames_feature_code == 'MVA'){
            $pla_geon_feature_code_explicit = '8421'; //Value matching "Zone
d'exercices militaires".
        }
        elseif($pla_geonames_feature_code == 'NVB'){
            $pla_geon_feature_code_explicit = '8422'; //Value matching "Base
navale".
```

```
        }
        elseif($pla_geonames_feature_code == 'OAS'){
            $pla_geon_feature_code_explicit = '8423'; //Value matching "Oasis".
        }
        elseif($pla_geonames_feature_code == 'OILF'){
            $pla_geon_feature_code_explicit = '8424'; //Value matching "Champ
pétrolifère".
        }
        elseif($pla_geonames_feature_code == 'PEAT'){
            $pla_geon_feature_code_explicit = '8425'; //Value matching "Zone de
récolte de la tourbe".
        }
        elseif($pla_geonames_feature_code == 'PRK'){
            $pla_geon_feature_code_explicit = '8426'; //Value matching "Parc".
        }
        elseif($pla_geonames_feature_code == 'PRT'){
            $pla_geon_feature_code_explicit = '8427'; //Value matching "Port".
        }
        elseif($pla_geonames_feature_code == 'QCKS'){
            $pla_geon_feature_code_explicit = '8428'; //Value matching "Sables
mouvants".
        }
        elseif($pla_geonames_feature_code == 'RES'){
            $pla_geon_feature_code_explicit = '8429'; //Value matching "Réserve".
        }
        elseif($pla_geonames_feature_code == 'RESA'){
            $pla_geon_feature_code_explicit = '8430'; //Value matching "Réserve
agricole".
        }
        elseif($pla_geonames_feature_code == 'RESF'){
            $pla_geon_feature_code_explicit = '8431'; //Value matching "Réserve
forestière".
        }
        elseif($pla_geonames_feature_code == 'RESH'){
            $pla_geon_feature_code_explicit = '8432'; //Value matching "Réserve de
chasse".
        }
        elseif($pla_geonames_feature_code == 'RESN'){
            $pla_geon_feature_code_explicit = '8433'; //Value matching "Réserve
naturelle".
        }
        elseif($pla_geonames_feature_code == 'RESP'){
            $pla_geon_feature_code_explicit = '8434'; //Value matching "Réserve de
palmiers".
        }
        elseif($pla_geonames_feature_code == 'RESV'){
            $pla_geon_feature_code_explicit = '8435'; //Value matching "Réserve
(pour des populations indigènes)".
        }
        elseif($pla_geonames_feature_code == 'RESW'){
            $pla_geon_feature_code_explicit = '8436'; //Value matching "Réserve
faunique".
        }
        elseif($pla_geonames_feature_code == 'RGN'){
            $pla_geon_feature_code_explicit = '8437'; //Value matching "Région".
        }
        elseif($pla_geonames_feature_code == 'RGNE'){
            $pla_geon_feature_code_explicit = '8438'; //Value matching "Région
économique".
```

```
        }
        elseif($pla_geonames_feature_code == 'RGNH'){
            $pla_geon_feature_code_explicit = '8439'; //Value matching "Région
historique".
        }
        elseif($pla_geonames_feature_code == 'RGNL'){
            $pla_geon_feature_code_explicit = '8440'; //Value matching "Région
lacustre".
        }
        elseif($pla_geonames_feature_code == 'RNGA'){
            $pla_geon_feature_code_explicit = '8441'; //Value matching "Zone de
tir".
        }
        elseif($pla_geonames_feature_code == 'SALT'){
            $pla_geon_feature_code_explicit = '8442'; //Value matching "Zone
saline".
        }
        elseif($pla_geonames_feature_code == 'SNOW'){
            $pla_geon_feature_code_explicit = '8443'; //Value matching "Champ de
neige".
        }
        elseif($pla_geonames_feature_code == 'TRB'){
            $pla_geon_feature_code_explicit = '8444'; //Value matching "Zone
tribale".
        }
        elseif($pla_geonames_feature_code == 'PPL'){
            $pla_geon_feature_code_explicit = '8445'; //Value matching "Lieu
habité".
        }
        elseif($pla_geonames_feature_code == 'PPLA'){
            $pla_geon_feature_code_explicit = '8446'; //Value matching "Siège
d'une division administrative de premier ordre".
        }
        elseif($pla_geonames_feature_code == 'PPLA2'){
            $pla_geon_feature_code_explicit = '8447'; //Value matching "Siège
d'une division administrative de deuxième ordre".
        }
        elseif($pla_geonames_feature_code == 'PPLA3'){
            $pla_geon_feature_code_explicit = '8448'; //Value matching "Siège
d'une division administrative de troisième ordre".
        }
        elseif($pla_geonames_feature_code == 'PPLA4'){
            $pla_geon_feature_code_explicit = '8449'; //Value matching "Siège
d'une division administrative de quatrième ordre".
        }
        elseif($pla_geonames_feature_code == 'PPLA5'){
            $pla_geon_feature_code_explicit = '8450'; //Value matching "Siège
d'une division administrative de cinquième ordre".
        }
        elseif($pla_geonames_feature_code == 'PPLC'){
            $pla_geon_feature_code_explicit = '8451'; //Value matching "Capitale
d'une entité politique".
        }
        elseif($pla_geonames_feature_code == 'PPLCH'){
            $pla_geon_feature_code_explicit = '8452'; //Value matching "Capitale
historique d'une entité politique".
        }
        elseif($pla_geonames_feature_code == 'PPLF'){
```

```
            $pla_geon_feature_code_explicit = '8453'; //Value matching "Village
agricole".
        }
        elseif($pla_geonames_feature_code == 'PPLG'){
            $pla_geon_feature_code_explicit = '8454'; //Value matching "Siège du
gouvernement d'une entité politique".
        }
        elseif($pla_geonames_feature_code == 'PPLH'){
            $pla_geon_feature_code_explicit = '8455'; //Value matching "Ancien
lieu habité".
        }
        elseif($pla_geonames_feature_code == 'PPLL'){
            $pla_geon_feature_code_explicit = '8456'; //Value matching "Localité
peuplée".
        }
        elseif($pla_geonames_feature_code == 'PPLQ'){
            $pla_geon_feature_code_explicit = '8457'; //Value matching "Lieu
anciennement habité".
        }
        elseif($pla_geonames_feature_code == 'PPLR'){
            $pla_geon_feature_code_explicit = '8458'; //Value matching "Lieu
habité religieux".
        }
        elseif($pla_geonames_feature_code == 'PPLS'){
            $pla_geon_feature_code_explicit = '8459'; //Value matching "Lieux
habités".
        }
        elseif($pla_geonames_feature_code == 'PPLW'){
            $pla_geon_feature_code_explicit = '8460'; //Value matching "Lieu
habité détruit".
        }
        elseif($pla_geonames_feature_code == 'PPLX'){
            $pla_geon_feature_code_explicit = '8461'; //Value matching "Section de
lieu habité".
        }
        elseif($pla_geonames_feature_code == 'STLMT'){
            $pla_geon_feature_code_explicit = '8462'; //Value matching "Colonie
israélienne".
        }
        elseif($pla_geonames_feature_code == 'CSWY'){
            $pla_geon_feature_code_explicit = '8463'; //Value matching "Chaussée
(terre-plein)".
        }
        elseif($pla_geonames_feature_code == 'OILP'){
            $pla_geon_feature_code_explicit = '8464'; //Value matching "Oléoduc".
        }
        elseif($pla_geonames_feature_code == 'PRMN'){
            $pla_geon_feature_code_explicit = '8465'; //Value matching
"Promenade".
        }
        elseif($pla_geonames_feature_code == 'PTGE'){
            $pla_geon_feature_code_explicit = '8466'; //Value matching "Portage".
        }
        elseif($pla_geonames_feature_code == 'RD'){
            $pla_geon_feature_code_explicit = '8467'; //Value matching "Route".
        }
        elseif($pla_geonames_feature_code == 'RDA'){
            $pla_geon_feature_code_explicit = '8468'; //Value matching "Ancienne
route".
```

```
        }
        elseif($pla_geonames_feature_code == 'RDB'){
            $pla_geon_feature_code_explicit = '8469'; //Value matching "Virage
routier".
        }
        elseif($pla_geonames_feature_code == 'RDCUT'){
            $pla_geon_feature_code_explicit = '8470'; //Value matching "Tranchée
routière".
        }
        elseif($pla_geonames_feature_code == 'RDJCT'){
            $pla_geon_feature_code_explicit = '8471'; //Value matching "Carrefour
routier".
        }
        elseif($pla_geonames_feature_code == 'RJCT'){
            $pla_geon_feature_code_explicit = '8472'; //Value matching "Noeud
ferroviaire".
        }
        elseif($pla_geonames_feature_code == 'RR'){
            $pla_geon_feature_code_explicit = '8473'; //Value matching "Voie
ferrée".
        }
        elseif($pla_geonames_feature_code == 'RRQ'){
            $pla_geon_feature_code_explicit = '8474'; //Value matching "Voie
ferrée abandonnée".
        }
        elseif($pla_geonames_feature_code == 'RTE'){
            $pla_geon_feature_code_explicit = '8475'; //Value matching "Route des
caravanes".
        }
        elseif($pla_geonames_feature_code == 'RYD'){
            $pla_geon_feature_code_explicit = '8476'; //Value matching "Gare de
triage".
        }
        elseif($pla_geonames_feature_code == 'ST'){
            $pla_geon_feature_code_explicit = '8477'; //Value matching "Rue".
        }
        elseif($pla_geonames_feature_code == 'STKR'){
            $pla_geon_feature_code_explicit = '8478'; //Value matching "Route du
bétail".
        }
        elseif($pla_geonames_feature_code == 'TNL'){
            $pla_geon_feature_code_explicit = '8479'; //Value matching "Tunnel".
        }
        elseif($pla_geonames_feature_code == 'TNLN'){
            $pla_geon_feature_code_explicit = '8480'; //Value matching "Tunnel
naturel".
        }
        elseif($pla_geonames_feature_code == 'TNLRD'){
            $pla_geon_feature_code_explicit = '8481'; //Value matching "Tunnel
routier".
        }
        elseif($pla_geonames_feature_code == 'TNLRR'){
            $pla_geon_feature_code_explicit = '8482'; //Value matching "Tunnel
ferroviaire".
        }
        elseif($pla_geonames_feature_code == 'TNLS'){
            $pla_geon_feature_code_explicit = '8483'; //Value matching "Tunnels".
        }
        elseif($pla_geonames_feature_code == 'TRL'){
```

```
            $pla_geon_feature_code_explicit = '8484'; //Value matching "Sentier".
        }
        elseif($pla_geonames_feature_code == 'ADMF'){
            $pla_geon_feature_code_explicit = '8485'; //Value matching "Bâtiment
administratif".
        }
        elseif($pla_geonames_feature_code == 'AGRF'){
            $pla_geon_feature_code_explicit = '8486'; //Value matching "Bâtiment
agricole".
        }
        elseif($pla_geonames_feature_code == 'AIRB'){
            $pla_geon_feature_code_explicit = '8487'; //Value matching "Base
aérienne".
        }
        elseif($pla_geonames_feature_code == 'AIRF'){
            $pla_geon_feature_code_explicit = '8488'; //Value matching
"Aérodrome".
        }
        elseif($pla_geonames_feature_code == 'AIRH'){
            $pla_geon_feature_code_explicit = '8489'; //Value matching "Héliport".
        }
        elseif($pla_geonames_feature_code == 'AIRP'){
            $pla_geon_feature_code_explicit = '8490'; //Value matching "Aéroport".
        }
        elseif($pla_geonames_feature_code == 'AIRQ'){
            $pla_geon_feature_code_explicit = '8491'; //Value matching "Aérodrome
abandonné".
        }
        elseif($pla_geonames_feature_code == 'AIRT'){
            $pla_geon_feature_code_explicit = '8492'; //Value matching "Terminal".
        }
        elseif($pla_geonames_feature_code == 'AMTH'){
            $pla_geon_feature_code_explicit = '8493'; //Value matching
"Amphithéâtre".
        }
        elseif($pla_geonames_feature_code == 'ANS'){
            $pla_geon_feature_code_explicit = '8494'; //Value matching "Site
archéologique ou préhistorique".
        }
        elseif($pla_geonames_feature_code == 'AQC'){
            $pla_geon_feature_code_explicit = '8495'; //Value matching
"Installation d'acquaculture".
        }
        elseif($pla_geonames_feature_code == 'ARCH'){
            $pla_geon_feature_code_explicit = '8496'; //Value matching "Arche".
        }
        elseif($pla_geonames_feature_code == 'ARCHV'){
            $pla_geon_feature_code_explicit = '8497'; //Value matching "Archives".
        }
        elseif($pla_geonames_feature_code == 'ART'){
            $pla_geon_feature_code_explicit = '8498'; //Value matching "Oeuvre
d'art".
        }
        elseif($pla_geonames_feature_code == 'ASTR'){
            $pla_geon_feature_code_explicit = '8499'; //Value matching "Station
astronomique".
        }
        elseif($pla_geonames_feature_code == 'ASYL'){
```

```
            $pla_geon_feature_code_explicit = '8500'; //Value matching "Asile
psychiatrique".
        }
        elseif($pla_geonames_feature_code == 'ATHF'){
            $pla_geon_feature_code_explicit = '8501'; //Value matching "Terrain de
sport".
        }
        elseif($pla_geonames_feature_code == 'ATM'){
            $pla_geon_feature_code_explicit = '8502'; //Value matching
"Distributeur automatique".
        }
        elseif($pla_geonames_feature_code == 'BANK'){
            $pla_geon_feature_code_explicit = '8503'; //Value matching "Banque".
        }
        elseif($pla_geonames_feature_code == 'BCN'){
            $pla_geon_feature_code_explicit = '8504'; //Value matching "Balise".
        }
        elseif($pla_geonames_feature_code == 'BDG'){
            $pla_geon_feature_code_explicit = '8505'; //Value matching "Pont".
        }
        elseif($pla_geonames_feature_code == 'BDGQ'){
            $pla_geon_feature_code_explicit = '8506'; //Value matching "Pont en
ruine".
        }
        elseif($pla_geonames_feature_code == 'BLDA'){
            $pla_geon_feature_code_explicit = '8507'; //Value matching "Immeuble
d'habitation".
        }
        elseif($pla_geonames_feature_code == 'BLDG'){
            $pla_geon_feature_code_explicit = '8508'; //Value matching
"Construction(s)".
        }
        elseif($pla_geonames_feature_code == 'BLDO'){
            $pla_geon_feature_code_explicit = '8509'; //Value matching "Immeuble
de bureaux".
        }
        elseif($pla_geonames_feature_code == 'BP'){
            $pla_geon_feature_code_explicit = '8510'; //Value matching "Borne
frontière".
        }
        elseif($pla_geonames_feature_code == 'BRKS'){
            $pla_geon_feature_code_explicit = '8511'; //Value matching "Caserne
militaire".
        }
        elseif($pla_geonames_feature_code == 'BRKW'){
            $pla_geon_feature_code_explicit = '8512'; //Value matching "Brise-
lames".
        }
        elseif($pla_geonames_feature_code == 'BSTN'){
            $pla_geon_feature_code_explicit = '8513'; //Value matching "Station de
mise en balles des produits agricoles".
        }
        elseif($pla_geonames_feature_code == 'BTYD'){
            $pla_geon_feature_code_explicit = '8514'; //Value matching "Chantier
naval (petits bateaux)".
        }
        elseif($pla_geonames_feature_code == 'BUR'){
            $pla_geon_feature_code_explicit = '8515'; //Value matching "Caverne(s)
funéraire(s)".
```

```
        }
        elseif($pla_geonames_feature_code == 'BUSTN'){
            $pla_geon_feature_code_explicit = '8516'; //Value matching "Station de
bus".
        }
        elseif($pla_geonames_feature_code == 'BUSTP'){
            $pla_geon_feature_code_explicit = '8517'; //Value matching "Arrêt de
bus".
        }
        elseif($pla_geonames_feature_code == 'CARN'){
            $pla_geon_feature_code_explicit = '8518'; //Value matching "Cairn".
        }
        elseif($pla_geonames_feature_code == 'CAVE'){
            $pla_geon_feature_code_explicit = '8519'; //Value matching
"Grotte(s)".
        }
        elseif($pla_geonames_feature_code == 'CH'){
            $pla_geon_feature_code_explicit = '8520'; //Value matching "Église".
        }
        elseif($pla_geonames_feature_code == 'CMP'){
            $pla_geon_feature_code_explicit = '8521'; //Value matching "Camp(s)".
        }
        elseif($pla_geonames_feature_code == 'CMPL'){
            $pla_geon_feature_code_explicit = '8522'; //Value matching "Camp de
bûcherons".
        }
        elseif($pla_geonames_feature_code == 'CMPLA'){
            $pla_geon_feature_code_explicit = '8523'; //Value matching "Camp de
travail".
        }
        elseif($pla_geonames_feature_code == 'CMPMN'){
            $pla_geon_feature_code_explicit = '8524'; //Value matching "Camp
minier".
        }
        elseif($pla_geonames_feature_code == 'CMPO'){
            $pla_geon_feature_code_explicit = '8525'; //Value matching "Camp
pétrolier".
        }
        elseif($pla_geonames_feature_code == 'CMPQ'){
            $pla_geon_feature_code_explicit = '8526'; //Value matching "Camp
abandonné".
        }
        elseif($pla_geonames_feature_code == 'CMPRF'){
            $pla_geon_feature_code_explicit = '8527'; //Value matching "Camp de
réfugiés".
        }
        elseif($pla_geonames_feature_code == 'CMTY'){
            $pla_geon_feature_code_explicit = '8528'; //Value matching
"Cimetière".
        }
        elseif($pla_geonames_feature_code == 'COMC'){
            $pla_geon_feature_code_explicit = '8529'; //Value matching "Centre de
communication".
        }
        elseif($pla_geonames_feature_code == 'CRRL'){
            $pla_geon_feature_code_explicit = '8530'; //Value matching
"Corral(s)".
        }
        elseif($pla_geonames_feature_code == 'CSNO'){
```

```
            $pla_geon_feature_code_explicit = '8531'; //Value matching "Casino".
        }
        elseif($pla_geonames_feature_code == 'CSTL'){
            $pla_geon_feature_code_explicit = '8532'; //Value matching "Château".
        }
        elseif($pla_geonames_feature_code == 'CSTM'){
            $pla_geon_feature_code_explicit = '8533'; //Value matching "Bureau de
douane".
        }
        elseif($pla_geonames_feature_code == 'CTHSE'){
            $pla_geon_feature_code_explicit = '8534'; //Value matching "Palais de
justice".
        }
        elseif($pla_geonames_feature_code == 'CTRA'){
            $pla_geon_feature_code_explicit = '8535'; //Value matching "Centre
atomique".
        }
        elseif($pla_geonames_feature_code == 'CTRCM'){
            $pla_geon_feature_code_explicit = '8536'; //Value matching "Centre
communautaire".
        }
        elseif($pla_geonames_feature_code == 'CTRF'){
            $pla_geon_feature_code_explicit = '8537'; //Value matching "Centre de
ressources".
        }
        elseif($pla_geonames_feature_code == 'CTRM'){
            $pla_geon_feature_code_explicit = '8538'; //Value matching "Centre
médical".
        }
        elseif($pla_geonames_feature_code == 'CTRR'){
            $pla_geon_feature_code_explicit = '8539'; //Value matching "Centre
religieux".
        }
        elseif($pla_geonames_feature_code == 'CTRS'){
            $pla_geon_feature_code_explicit = '8540'; //Value matching "Centre
spatial".
        }
        elseif($pla_geonames_feature_code == 'CVNT'){
            $pla_geon_feature_code_explicit = '8541'; //Value matching "Couvent".
        }
        elseif($pla_geonames_feature_code == 'DAM'){
            $pla_geon_feature_code_explicit = '8542'; //Value matching "Barrage".
        }
        elseif($pla_geonames_feature_code == 'DAMQ'){
            $pla_geon_feature_code_explicit = '8543'; //Value matching "Barrage
détruit ou endommagé".
        }
        elseif($pla_geonames_feature_code == 'DAMSB'){
            $pla_geon_feature_code_explicit = '8544'; //Value matching "Barrage
subaquatique".
        }
        elseif($pla_geonames_feature_code == 'DARY'){
            $pla_geon_feature_code_explicit = '8545'; //Value matching "Laiterie".
        }
        elseif($pla_geonames_feature_code == 'DCKD'){
            $pla_geon_feature_code_explicit = '8546'; //Value matching "Cale
sèche".
        }
        elseif($pla_geonames_feature_code == 'DCKY'){
```

```
            $pla_geon_feature_code_explicit = '8547'; //Value matching "Chantier
naval".
        }
        elseif($pla_geonames_feature_code == 'DIKE'){
            $pla_geon_feature_code_explicit = '8548'; //Value matching "Digue
(dike)".
        }
        elseif($pla_geonames_feature_code == 'DIP'){
            $pla_geon_feature_code_explicit = '8549'; //Value matching "Bâtiment
diplomatique".
        }
        elseif($pla_geonames_feature_code == 'DPOF'){
            $pla_geon_feature_code_explicit = '8550'; //Value matching "Dépôt de
carburant".
        }
        elseif($pla_geonames_feature_code == 'EST'){
            $pla_geon_feature_code_explicit = '8551'; //Value matching "Domaine
agricole".
        }
        elseif($pla_geonames_feature_code == 'ESTO'){
            $pla_geon_feature_code_explicit = '8552'; //Value matching "Plantation
de palmiers à huile".
        }
        elseif($pla_geonames_feature_code == 'ESTR'){
            $pla_geon_feature_code_explicit = '8553'; //Value matching "Plantation
d'hévéas".
        }
        elseif($pla_geonames_feature_code == 'ESTSG'){
            $pla_geon_feature_code_explicit = '8554'; //Value matching "Plantation
de canne à sucre".
        }
        elseif($pla_geonames_feature_code == 'ESTT'){
            $pla_geon_feature_code_explicit = '8555'; //Value matching "Plantation
de thé".
        }
        elseif($pla_geonames_feature_code == 'ESTX'){
            $pla_geon_feature_code_explicit = '8556'; //Value matching "Section de
domaine agricole".
        }
        elseif($pla_geonames_feature_code == 'FCL'){
            $pla_geon_feature_code_explicit = '8557'; //Value matching
"Établissement".
        }
        elseif($pla_geonames_feature_code == 'FNDY'){
            $pla_geon_feature_code_explicit = '8558'; //Value matching "Fonderie".
        }
        elseif($pla_geonames_feature_code == 'FRM'){
            $pla_geon_feature_code_explicit = '8559'; //Value matching "Ferme".
        }
        elseif($pla_geonames_feature_code == 'FRMQ'){
            $pla_geon_feature_code_explicit = '8560'; //Value matching "Ferme
abandonnée".
        }
        elseif($pla_geonames_feature_code == 'FRMS'){
            $pla_geon_feature_code_explicit = '8561'; //Value matching "Fermes".
        }
        elseif($pla_geonames_feature_code == 'FRMT'){
            $pla_geon_feature_code_explicit = '8562'; //Value matching
"Exploitation agricole".
```

```
        }
        elseif($pla_geonames_feature_code == 'FT'){
            $pla_geon_feature_code_explicit = '8563'; //Value matching "Fort".
        }
        elseif($pla_geonames_feature_code == 'FY'){
            $pla_geon_feature_code_explicit = '8564'; //Value matching "Ferry".
        }
        elseif($pla_geonames_feature_code == 'FYT'){
            $pla_geon_feature_code_explicit = '8565'; //Value matching "Terminal
de ferry".
        }
        elseif($pla_geonames_feature_code == 'GATE'){
            $pla_geon_feature_code_explicit = '8566'; //Value matching "Porte".
        }
        elseif($pla_geonames_feature_code == 'GDN'){
            $pla_geon_feature_code_explicit = '8567'; //Value matching
"Jardin(s)".
        }
        elseif($pla_geonames_feature_code == 'GHAT'){
            $pla_geon_feature_code_explicit = '8568'; //Value matching "Ghat".
        }
        elseif($pla_geonames_feature_code == 'GHSE'){
            $pla_geon_feature_code_explicit = '8569'; //Value matching "Maison
d'hôtes".
        }
        elseif($pla_geonames_feature_code == 'GOSP'){
            $pla_geon_feature_code_explicit = '8570'; //Value matching
"Installation de séparation gaz-huile".
        }
        elseif($pla_geonames_feature_code == 'GOVL'){
            $pla_geon_feature_code_explicit = '8571'; //Value matching "Bâtiment
de l'administration locale".
        }
        elseif($pla_geonames_feature_code == 'GRVE'){
            $pla_geon_feature_code_explicit = '8572'; //Value matching "Tombe".
        }
        elseif($pla_geonames_feature_code == 'HERM'){
            $pla_geon_feature_code_explicit = '8573'; //Value matching "Ermitage".
        }
        elseif($pla_geonames_feature_code == 'HLT'){
            $pla_geon_feature_code_explicit = '8574'; //Value matching "Aire de
repos pour les caravanes".
        }
        elseif($pla_geonames_feature_code == 'HMSD'){
            $pla_geon_feature_code_explicit = '8575'; //Value matching "Résidence
d'une station d'élevage (Australie, Nouvelle-Zélande)".
        }
        elseif($pla_geonames_feature_code == 'HSE'){
            $pla_geon_feature_code_explicit = '8576'; //Value matching
"Maison(s)".
        }
        elseif($pla_geonames_feature_code == 'HSEC'){
            $pla_geon_feature_code_explicit = '8577'; //Value matching "Maison de
campagne".
        }
        elseif($pla_geonames_feature_code == 'HSP'){
            $pla_geon_feature_code_explicit = '8578'; //Value matching "Hôpital".
        }
        elseif($pla_geonames_feature_code == 'HSPC'){
```

```
            $pla_geon_feature_code_explicit = '8579'; //Value matching "Clinique".
        }
        elseif($pla_geonames_feature_code == 'HSPD'){
            $pla_geon_feature_code_explicit = '8580'; //Value matching
"Dispensaire".
        }
        elseif($pla_geonames_feature_code == 'HSPL'){
            $pla_geon_feature_code_explicit = '8581'; //Value matching
"Léproserie".
        }
        elseif($pla_geonames_feature_code == 'HSTS'){
            $pla_geon_feature_code_explicit = '8582'; //Value matching "Site
historique".
        }
        elseif($pla_geonames_feature_code == 'HTL'){
            $pla_geon_feature_code_explicit = '8583'; //Value matching "Hôtel".
        }
        elseif($pla_geonames_feature_code == 'HUT'){
            $pla_geon_feature_code_explicit = '8584'; //Value matching "Hutte".
        }
        elseif($pla_geonames_feature_code == 'HUTS'){
            $pla_geon_feature_code_explicit = '8585'; //Value matching "Huttes".
        }
        elseif($pla_geonames_feature_code == 'INSM'){
            $pla_geon_feature_code_explicit = '8586'; //Value matching
"Installation militaire".
        }
        elseif($pla_geonames_feature_code == 'ITTR'){
            $pla_geon_feature_code_explicit = '8587'; //Value matching "Institut
de recherche".
        }
        elseif($pla_geonames_feature_code == 'JTY'){
            $pla_geon_feature_code_explicit = '8588'; //Value matching "Jetée".
        }
        elseif($pla_geonames_feature_code == 'LDNG'){
            $pla_geon_feature_code_explicit = '8589'; //Value matching
"Débarcadère".
        }
        elseif($pla_geonames_feature_code == 'LEPC'){
            $pla_geon_feature_code_explicit = '8590'; //Value matching "Colonie de
lépreux".
        }
        elseif($pla_geonames_feature_code == 'LIBR'){
            $pla_geon_feature_code_explicit = '8591'; //Value matching
"Bibliothèque".
        }
        elseif($pla_geonames_feature_code == 'LNDF'){
            $pla_geon_feature_code_explicit = '8592'; //Value matching "Décharge".
        }
        elseif($pla_geonames_feature_code == 'LOCK'){
            $pla_geon_feature_code_explicit = '8593'; //Value matching "Écluse(s)
navigable(s)".
        }
        elseif($pla_geonames_feature_code == 'LTHSE'){
            $pla_geon_feature_code_explicit = '8594'; //Value matching "Phare".
        }
        elseif($pla_geonames_feature_code == 'MALL'){
            $pla_geon_feature_code_explicit = '8595'; //Value matching "Centre
commercial".
```

```
        }
        elseif($pla_geonames_feature_code == 'MAR'){
            $pla_geon_feature_code_explicit = '8596'; //Value matching "Marina".
        }
        elseif($pla_geonames_feature_code == 'MFG'){
            $pla_geon_feature_code_explicit = '8597'; //Value matching "Usine".
        }
        elseif($pla_geonames_feature_code == 'MFGB'){
            $pla_geon_feature_code_explicit = '8598'; //Value matching
"Brasserie".
        }
        elseif($pla_geonames_feature_code == 'MFGC'){
            $pla_geon_feature_code_explicit = '8599'; //Value matching
"Conserverie".
        }
        elseif($pla_geonames_feature_code == 'MFGCU'){
            $pla_geon_feature_code_explicit = '8600'; //Value matching
"Cuivrerie".
        }
        elseif($pla_geonames_feature_code == 'MFGLM'){
            $pla_geon_feature_code_explicit = '8601'; //Value matching "Four à
chaux".
        }
        elseif($pla_geonames_feature_code == 'MFGM'){
            $pla_geon_feature_code_explicit = '8602'; //Value matching "Usine de
munitions".
        }
        elseif($pla_geonames_feature_code == 'MFGPH'){
            $pla_geon_feature_code_explicit = '8603'; //Value matching "Usine de
phosphates".
        }
        elseif($pla_geonames_feature_code == 'MFGQ'){
            $pla_geon_feature_code_explicit = '8604'; //Value matching "Usine
abandonnée".
        }
        elseif($pla_geonames_feature_code == 'MFGSG'){
            $pla_geon_feature_code_explicit = '8605'; //Value matching "Raffinerie
de sucre".
        }
        elseif($pla_geonames_feature_code == 'MKT'){
            $pla_geon_feature_code_explicit = '8606'; //Value matching "Marché".
        }
        elseif($pla_geonames_feature_code == 'ML'){
            $pla_geon_feature_code_explicit = '8607'; //Value matching
"Moulin(s)".
        }
        elseif($pla_geonames_feature_code == 'MLM'){
            $pla_geon_feature_code_explicit = '8608'; //Value matching "Usine de
traitement du minerai".
        }
        elseif($pla_geonames_feature_code == 'MLO'){
            $pla_geon_feature_code_explicit = '8609'; //Value matching "Moulin à
huile d'olive".
        }
        elseif($pla_geonames_feature_code == 'MLSG'){
            $pla_geon_feature_code_explicit = '8610'; //Value matching "Moulin à
sucre".
        }
        elseif($pla_geonames_feature_code == 'MLSGQ'){
```

```
            $pla_geon_feature_code_explicit = '8611'; //Value matching "Ancien
moulin à sucre".
        }
        elseif($pla_geonames_feature_code == 'MLSW'){
            $pla_geon_feature_code_explicit = '8612'; //Value matching "Scierie".
        }
        elseif($pla_geonames_feature_code == 'MLWND'){
            $pla_geon_feature_code_explicit = '8613'; //Value matching "Moulin à
vent".
        }
        elseif($pla_geonames_feature_code == 'MLWTR'){
            $pla_geon_feature_code_explicit = '8614'; //Value matching "Moulin à
eau".
        }
        elseif($pla_geonames_feature_code == 'MN'){
            $pla_geon_feature_code_explicit = '8615'; //Value matching "Mine(s)".
        }
        elseif($pla_geonames_feature_code == 'MNAU'){
            $pla_geon_feature_code_explicit = '8616'; //Value matching "Mine(s)
d'or".
        }
        elseif($pla_geonames_feature_code == 'MNC'){
            $pla_geon_feature_code_explicit = '8617'; //Value matching "Mine(s) de
charbon".
        }
        elseif($pla_geonames_feature_code == 'MNCR'){
            $pla_geon_feature_code_explicit = '8618'; //Value matching "Mine(s) de
chrome".
        }
        elseif($pla_geonames_feature_code == 'MNCU'){
            $pla_geon_feature_code_explicit = '8619'; //Value matching "Mine(s) de
cuivre".
        }
        elseif($pla_geonames_feature_code == 'MNFE'){
            $pla_geon_feature_code_explicit = '8620'; //Value matching "Mine(s) de
fer".
        }
        elseif($pla_geonames_feature_code == 'MNMT'){
            $pla_geon_feature_code_explicit = '8621'; //Value matching "Monument".
        }
        elseif($pla_geonames_feature_code == 'MNN'){
            $pla_geon_feature_code_explicit = '8622'; //Value matching "Mine(s) de
sel".
        }
        elseif($pla_geonames_feature_code == 'MNQ'){
            $pla_geon_feature_code_explicit = '8623'; //Value matching "Mine
abandonnée".
        }
        elseif($pla_geonames_feature_code == 'MNQR'){
            $pla_geon_feature_code_explicit = '8624'; //Value matching
"Carrière(s)".
        }
        elseif($pla_geonames_feature_code == 'MOLE'){
            $pla_geon_feature_code_explicit = '8625'; //Value matching "Môle".
        }
        elseif($pla_geonames_feature_code == 'MSQE'){
            $pla_geon_feature_code_explicit = '8626'; //Value matching "Mosquée".
        }
        elseif($pla_geonames_feature_code == 'MSSN'){
```

```
            $pla_geon_feature_code_explicit = '8627'; //Value matching "Mission".
        }
        elseif($pla_geonames_feature_code == 'MSSNQ'){
            $pla_geon_feature_code_explicit = '8628'; //Value matching "Mission
abandonnée".
        }
        elseif($pla_geonames_feature_code == 'MSTY'){
            $pla_geon_feature_code_explicit = '8629'; //Value matching
"Monastère".
        }
        elseif($pla_geonames_feature_code == 'MTRO'){
            $pla_geon_feature_code_explicit = '8630'; //Value matching "Station de
métro".
        }
        elseif($pla_geonames_feature_code == 'MUS'){
            $pla_geon_feature_code_explicit = '8631'; //Value matching "Musée".
        }
        elseif($pla_geonames_feature_code == 'NOV'){
            $pla_geon_feature_code_explicit = '8632'; //Value matching "Noviciat".
        }
        elseif($pla_geonames_feature_code == 'NSY'){
            $pla_geon_feature_code_explicit = '8633'; //Value matching
"Pépinière(s)".
        }
        elseif($pla_geonames_feature_code == 'OBPT'){
            $pla_geon_feature_code_explicit = '8634'; //Value matching "Point
d'observation".
        }
        elseif($pla_geonames_feature_code == 'OBS'){
            $pla_geon_feature_code_explicit = '8635'; //Value matching
"Observatoire".
        }
        elseif($pla_geonames_feature_code == 'OBSR'){
            $pla_geon_feature_code_explicit = '8636'; //Value matching "Radio-
observatoire".
        }
        elseif($pla_geonames_feature_code == 'OILJ'){
            $pla_geon_feature_code_explicit = '8637'; //Value matching "Jonction
d'oléoducs".
        }
        elseif($pla_geonames_feature_code == 'OILQ'){
            $pla_geon_feature_code_explicit = '8638'; //Value matching "Puits de
pétrole abandonné".
        }
        elseif($pla_geonames_feature_code == 'OILR'){
            $pla_geon_feature_code_explicit = '8639'; //Value matching "Raffinerie
de pétrole".
        }
        elseif($pla_geonames_feature_code == 'OILT'){
            $pla_geon_feature_code_explicit = '8640'; //Value matching "Réservoir
de pétrole".
        }
        elseif($pla_geonames_feature_code == 'OILW'){
            $pla_geon_feature_code_explicit = '8641'; //Value matching "Puits de
pétrole".
        }
        elseif($pla_geonames_feature_code == 'OPRA'){
            $pla_geon_feature_code_explicit = '8642'; //Value matching "Opéra".
        }
```

```
        elseif($pla_geonames_feature_code == 'PAL'){
            $pla_geon_feature_code_explicit = '8643'; //Value matching "Palais".
        }
        elseif($pla_geonames_feature_code == 'PGDA'){
            $pla_geon_feature_code_explicit = '8644'; //Value matching "Pagode".
        }
        elseif($pla_geonames_feature_code == 'PIER'){
            $pla_geon_feature_code_explicit = '8645'; //Value matching "Estacade".
        }
        elseif($pla_geonames_feature_code == 'PKLT'){
            $pla_geon_feature_code_explicit = '8646'; //Value matching "Parking".
        }
        elseif($pla_geonames_feature_code == 'PMPO'){
            $pla_geon_feature_code_explicit = '8647'; //Value matching "Station de
pompage de pétrole".
        }
        elseif($pla_geonames_feature_code == 'PMPW'){
            $pla_geon_feature_code_explicit = '8648'; //Value matching "Station de
pompage d'eau".
        }
        elseif($pla_geonames_feature_code == 'PO'){
            $pla_geon_feature_code_explicit = '8649'; //Value matching "Bureau de
poste".
        }
        elseif($pla_geonames_feature_code == 'PP'){
            $pla_geon_feature_code_explicit = '8650'; //Value matching "Poste de
police".
        }
        elseif($pla_geonames_feature_code == 'PPQ'){
            $pla_geon_feature_code_explicit = '8651'; //Value matching "Poste de
police abandonné".
        }
        elseif($pla_geonames_feature_code == 'PRKGT'){
            $pla_geon_feature_code_explicit = '8652'; //Value matching "Porte d'un
parc".
        }
        elseif($pla_geonames_feature_code == 'PRKHQ'){
            $pla_geon_feature_code_explicit = '8653'; //Value matching "Siège
administratif d'un parc".
        }
        elseif($pla_geonames_feature_code == 'PRN'){
            $pla_geon_feature_code_explicit = '8654'; //Value matching "Prison".
        }
        elseif($pla_geonames_feature_code == 'PRNJ'){
            $pla_geon_feature_code_explicit = '8655'; //Value matching "Maison de
correction".
        }
        elseif($pla_geonames_feature_code == 'PRNQ'){
            $pla_geon_feature_code_explicit = '8656'; //Value matching "Prison
abandonnée".
        }
        elseif($pla_geonames_feature_code == 'PS'){
            $pla_geon_feature_code_explicit = '8657'; //Value matching "Centrale
électrique".
        }
        elseif($pla_geonames_feature_code == 'PSH'){
            $pla_geon_feature_code_explicit = '8658'; //Value matching "Centrale
hydroélectrique".
        }
```

```
        elseif($pla_geonames_feature_code == 'PSN'){
            $pla_geon_feature_code_explicit = '8659'; //Value matching "Centrale
nucléaire".
        }
        elseif($pla_geonames_feature_code == 'PSTB'){
            $pla_geon_feature_code_explicit = '8660'; //Value matching "Poste
frontière".
        }
        elseif($pla_geonames_feature_code == 'PSTC'){
            $pla_geon_feature_code_explicit = '8661'; //Value matching "Poste de
douane".
        }
        elseif($pla_geonames_feature_code == 'PSTP'){
            $pla_geon_feature_code_explicit = '8662'; //Value matching "Poste de
patrouille".
        }
        elseif($pla_geonames_feature_code == 'PYR'){
            $pla_geon_feature_code_explicit = '8663'; //Value matching "Pyramide".
        }
        elseif($pla_geonames_feature_code == 'PYRS'){
            $pla_geon_feature_code_explicit = '8664'; //Value matching
"Pyramides".
        }
        elseif($pla_geonames_feature_code == 'QUAY'){
            $pla_geon_feature_code_explicit = '8665'; //Value matching "Quai
(quay)".
        }
        elseif($pla_geonames_feature_code == 'RDCR'){
            $pla_geon_feature_code_explicit = '8666'; //Value matching "Carrefour
giratoire".
        }
        elseif($pla_geonames_feature_code == 'RDIN'){
            $pla_geon_feature_code_explicit = '8667'; //Value matching "Échangeur
autoroutier".
        }
        elseif($pla_geonames_feature_code == 'RECG'){
            $pla_geon_feature_code_explicit = '8668'; //Value matching "Terrain de
golf".
        }
        elseif($pla_geonames_feature_code == 'RECR'){
            $pla_geon_feature_code_explicit = '8669'; //Value matching "Piste de
course".
        }
        elseif($pla_geonames_feature_code == 'REST'){
            $pla_geon_feature_code_explicit = '8670'; //Value matching
"Restaurant".
        }
        elseif($pla_geonames_feature_code == 'RET'){
            $pla_geon_feature_code_explicit = '8671'; //Value matching "Magasin".
        }
        elseif($pla_geonames_feature_code == 'RHSE'){
            $pla_geon_feature_code_explicit = '8672'; //Value matching "Maison de
repos".
        }
        elseif($pla_geonames_feature_code == 'RKRY'){
            $pla_geon_feature_code_explicit = '8673'; //Value matching "Rookery".
        }
        elseif($pla_geonames_feature_code == 'RLG'){
```

```
            $pla_geon_feature_code_explicit = '8674'; //Value matching "Site
religieux".
        }
        elseif($pla_geonames_feature_code == 'RLGR'){
            $pla_geon_feature_code_explicit = '8675'; //Value matching "Retraite".
        }
        elseif($pla_geonames_feature_code == 'RNCH'){
            $pla_geon_feature_code_explicit = '8676'; //Value matching "Ranch(s)".
        }
        elseif($pla_geonames_feature_code == 'RSD'){
            $pla_geon_feature_code_explicit = '8677'; //Value matching "Voie de
garage".
        }
        elseif($pla_geonames_feature_code == 'RSGNL'){
            $pla_geon_feature_code_explicit = '8678'; //Value matching "Signal
ferroviaire".
        }
        elseif($pla_geonames_feature_code == 'RSRT'){
            $pla_geon_feature_code_explicit = '8679'; //Value matching "Centre de
villégiature".
        }
        elseif($pla_geonames_feature_code == 'RSTN'){
            $pla_geon_feature_code_explicit = '8680'; //Value matching "Gare
ferroviaire".
        }
        elseif($pla_geonames_feature_code == 'RSTNQ'){
            $pla_geon_feature_code_explicit = '8681'; //Value matching "Gare
ferroviaire abandonnée".
        }
        elseif($pla_geonames_feature_code == 'RSTP'){
            $pla_geon_feature_code_explicit = '8682'; //Value matching "Arrêt
ferroviaire".
        }
        elseif($pla_geonames_feature_code == 'RSTPQ'){
            $pla_geon_feature_code_explicit = '8683'; //Value matching "Arrêt
ferroviaire abandonné".
        }
        elseif($pla_geonames_feature_code == 'RUIN'){
            $pla_geon_feature_code_explicit = '8684'; //Value matching "Ruine(s)".
        }
        elseif($pla_geonames_feature_code == 'SCH'){
            $pla_geon_feature_code_explicit = '8685'; //Value matching "École".
        }
        elseif($pla_geonames_feature_code == 'SCHA'){
            $pla_geon_feature_code_explicit = '8686'; //Value matching "École
agricole".
        }
        elseif($pla_geonames_feature_code == 'SCHC'){
            $pla_geon_feature_code_explicit = '8687'; //Value matching "Collège
universitaire".
        }
        elseif($pla_geonames_feature_code == 'SCHL'){
            $pla_geon_feature_code_explicit = '8688'; //Value matching "École de
langue".
        }
        elseif($pla_geonames_feature_code == 'SCHM'){
            $pla_geon_feature_code_explicit = '8689'; //Value matching "École
militaire".
        }
```

```
        elseif($pla_geonames_feature_code == 'SCHN'){
            $pla_geon_feature_code_explicit = '8690'; //Value matching "École
maritime".
        }
        elseif($pla_geonames_feature_code == 'SCHT'){
            $pla_geon_feature_code_explicit = '8691'; //Value matching "École
technique".
        }
        elseif($pla_geonames_feature_code == 'SECP'){
            $pla_geon_feature_code_explicit = '8692'; //Value matching "Centre de
préparation aux examens d'État".
        }
        elseif($pla_geonames_feature_code == 'SHPF'){
            $pla_geon_feature_code_explicit = '8693'; //Value matching "Bergerie".
        }
        elseif($pla_geonames_feature_code == 'SHRN'){
            $pla_geon_feature_code_explicit = '8694'; //Value matching
"Sanctuaire".
        }
        elseif($pla_geonames_feature_code == 'SHSE'){
            $pla_geon_feature_code_explicit = '8695'; //Value matching "Entrepôt".
        }
        elseif($pla_geonames_feature_code == 'SLCE'){
            $pla_geon_feature_code_explicit = '8696'; //Value matching "Écluse".
        }
        elseif($pla_geonames_feature_code == 'SNTR'){
            $pla_geon_feature_code_explicit = '8697'; //Value matching
"Sanatorium".
        }
        elseif($pla_geonames_feature_code == 'SPA'){
            $pla_geon_feature_code_explicit = '8698'; //Value matching "Spa".
        }
        elseif($pla_geonames_feature_code == 'SPLY'){
            $pla_geon_feature_code_explicit = '8699'; //Value matching
"Déversoir".
        }
        elseif($pla_geonames_feature_code == 'SQR'){
            $pla_geon_feature_code_explicit = '8700'; //Value matching "Square".
        }
        elseif($pla_geonames_feature_code == 'STBL'){
            $pla_geon_feature_code_explicit = '8701'; //Value matching "Étable".
        }
        elseif($pla_geonames_feature_code == 'STDM'){
            $pla_geon_feature_code_explicit = '8702'; //Value matching "Stade".
        }
        elseif($pla_geonames_feature_code == 'STNB'){
            $pla_geon_feature_code_explicit = '8703'; //Value matching "Base de
recherche scientifique".
        }
        elseif($pla_geonames_feature_code == 'STNC'){
            $pla_geon_feature_code_explicit = '8704'; //Value matching "Poste de
garde-côtes".
        }
        elseif($pla_geonames_feature_code == 'STNE'){
            $pla_geon_feature_code_explicit = '8705'; //Value matching "Station
d'expérimentation".
        }
        elseif($pla_geonames_feature_code == 'STNF'){
```

```
            $pla_geon_feature_code_explicit = '8706'; //Value matching "Station
forestière".
        }
        elseif($pla_geonames_feature_code == 'STNI'){
            $pla_geon_feature_code_explicit = '8707'; //Value matching "Station
d'inspection".
        }
        elseif($pla_geonames_feature_code == 'STNM'){
            $pla_geon_feature_code_explicit = '8708'; //Value matching "Station
météorologique".
        }
        elseif($pla_geonames_feature_code == 'STNR'){
            $pla_geon_feature_code_explicit = '8709'; //Value matching "Station de
radio".
        }
        elseif($pla_geonames_feature_code == 'STNS'){
            $pla_geon_feature_code_explicit = '8710'; //Value matching "Station
satellite".
        }
        elseif($pla_geonames_feature_code == 'STNW'){
            $pla_geon_feature_code_explicit = '8711'; //Value matching "Station
baleinière".
        }
        elseif($pla_geonames_feature_code == 'STPS'){
            $pla_geon_feature_code_explicit = '8712'; //Value matching "Marches".
        }
        elseif($pla_geonames_feature_code == 'SWT'){
            $pla_geon_feature_code_explicit = '8713'; //Value matching "Station
d'épuration des eaux usées".
        }
        elseif($pla_geonames_feature_code == 'SYG'){
            $pla_geon_feature_code_explicit = '8714'; //Value matching
"Synagogue".
        }
        elseif($pla_geonames_feature_code == 'THTR'){
            $pla_geon_feature_code_explicit = '8715'; //Value matching "Théâtre,
salle de cinéma, salle de spectacle".
        }
        elseif($pla_geonames_feature_code == 'TMB'){
            $pla_geon_feature_code_explicit = '8716'; //Value matching "Tombe(s)".
        }
        elseif($pla_geonames_feature_code == 'TMPL'){
            $pla_geon_feature_code_explicit = '8717'; //Value matching
"Temple(s)".
        }
        elseif($pla_geonames_feature_code == 'TNKD'){
            $pla_geon_feature_code_explicit = '8718'; //Value matching "Bac de
trempage pour le bétail".
        }
        elseif($pla_geonames_feature_code == 'TOLL'){
            $pla_geon_feature_code_explicit = '8719'; //Value matching "Barrière
de péage".
        }
        elseif($pla_geonames_feature_code == 'TOWR'){
            $pla_geon_feature_code_explicit = '8720'; //Value matching "Tour".
        }
        elseif($pla_geonames_feature_code == 'TRAM'){
            $pla_geon_feature_code_explicit = '8721'; //Value matching "Tramway".
        }
```

```
        elseif($pla_geonames_feature_code == 'TRANT'){
            $pla_geon_feature_code_explicit = '8722'; //Value matching "Terminal
de transit".
        }
        elseif($pla_geonames_feature_code == 'TRIG'){
            $pla_geon_feature_code_explicit = '8723'; //Value matching "Station de
triangulation".
        }
        elseif($pla_geonames_feature_code == 'TRMO'){
            $pla_geon_feature_code_explicit = '8724'; //Value matching "Terminal
d'oléoduc".
        }
        elseif($pla_geonames_feature_code == 'TWO'){
            $pla_geon_feature_code_explicit = '8725'; //Value matching "Bureau
d'intérim".
        }
        elseif($pla_geonames_feature_code == 'UNIP'){
            $pla_geon_feature_code_explicit = '8726'; //Value matching "École
préparatoire universitaire".
        }
        elseif($pla_geonames_feature_code == 'UNIV'){
            $pla_geon_feature_code_explicit = '8727'; //Value matching
"Université".
        }
        elseif($pla_geonames_feature_code == 'USGE'){
            $pla_geon_feature_code_explicit = '8728'; //Value matching
"Établissement gouvernemental des États-Unis au Panama".
        }
        elseif($pla_geonames_feature_code == 'VETF'){
            $pla_geon_feature_code_explicit = '8729'; //Value matching
"Établissement vétérinaire".
        }
        elseif($pla_geonames_feature_code == 'WALL'){
            $pla_geon_feature_code_explicit = '8730'; //Value matching "Mur".
        }
        elseif($pla_geonames_feature_code == 'WALLA'){
            $pla_geon_feature_code_explicit = '8731'; //Value matching "Ancien
mur".
        }
        elseif($pla_geonames_feature_code == 'WEIR'){
            $pla_geon_feature_code_explicit = '8732'; //Value matching "Seuil(s)
(barrage)".
        }
        elseif($pla_geonames_feature_code == 'WHRF'){
            $pla_geon_feature_code_explicit = '8733'; //Value matching "Quai(s)
(wharf)".
        }
        elseif($pla_geonames_feature_code == 'WRCK'){
            $pla_geon_feature_code_explicit = '8734'; //Value matching "Épave".
        }
        elseif($pla_geonames_feature_code == 'WTRW'){
            $pla_geon_feature_code_explicit = '8735'; //Value matching
"Installation d'alimentation en eau potable".
        }
        elseif($pla_geonames_feature_code == 'ZNF'){
            $pla_geon_feature_code_explicit = '8736'; //Value matching "Zone de
libre-échange".
        }
        elseif($pla_geonames_feature_code == 'ZOO'){
```

```
            $pla_geon_feature_code_explicit = '8737'; //Value matching "Zoo".
        }
        elseif($pla_geonames_feature_code == 'ASPH'){
            $pla_geon_feature_code_explicit = '8738'; //Value matching "Lac
d'asphalte".
        }
        elseif($pla_geonames_feature_code == 'ATOL'){
            $pla_geon_feature_code_explicit = '8739'; //Value matching "Atoll(s)".
        }
        elseif($pla_geonames_feature_code == 'BAR'){
            $pla_geon_feature_code_explicit = '8740'; //Value matching "Barre
(océanographie)".
        }
        elseif($pla_geonames_feature_code == 'BCH'){
            $pla_geon_feature_code_explicit = '8741'; //Value matching "Plage".
        }
        elseif($pla_geonames_feature_code == 'BCHS'){
            $pla_geon_feature_code_explicit = '8742'; //Value matching "Plages".
        }
        elseif($pla_geonames_feature_code == 'BDLD'){
            $pla_geon_feature_code_explicit = '8743'; //Value matching "Badlands".
        }
        elseif($pla_geonames_feature_code == 'BLDR'){
            $pla_geon_feature_code_explicit = '8744'; //Value matching "Chaos
(géologie)".
        }
        elseif($pla_geonames_feature_code == 'BLHL'){
            $pla_geon_feature_code_explicit = '8745'; //Value matching "Geyser
maritime".
        }
        elseif($pla_geonames_feature_code == 'BLOW'){
            $pla_geon_feature_code_explicit = '8746'; //Value matching "Creux de
déflation".
        }
        elseif($pla_geonames_feature_code == 'BNCH'){
            $pla_geon_feature_code_explicit = '8747'; //Value matching "Replat".
        }
        elseif($pla_geonames_feature_code == 'BUTE'){
            $pla_geon_feature_code_explicit = '8748'; //Value matching "Butte(s)".
        }
        elseif($pla_geonames_feature_code == 'CAPE'){
            $pla_geon_feature_code_explicit = '8749'; //Value matching "Cap
(cape)".
        }
        elseif($pla_geonames_feature_code == 'CFT'){
            $pla_geon_feature_code_explicit = '8750'; //Value matching "Fissure(s)
(cleft)".
        }
        elseif($pla_geonames_feature_code == 'CLDA'){
            $pla_geon_feature_code_explicit = '8751'; //Value matching "Caldera".
        }
        elseif($pla_geonames_feature_code == 'CLF'){
            $pla_geon_feature_code_explicit = '8752'; //Value matching
"Falaise(s)".
        }
        elseif($pla_geonames_feature_code == 'CNYN'){
            $pla_geon_feature_code_explicit = '8753'; //Value matching "Canyon".
        }
        elseif($pla_geonames_feature_code == 'CONE'){
```

```
            $pla_geon_feature_code_explicit = '8754'; //Value matching "Cône(s)
volcanique(s)".
        }
        elseif($pla_geonames_feature_code == 'CRDR'){
            $pla_geon_feature_code_explicit = '8755'; //Value matching "Corridor".
        }
        elseif($pla_geonames_feature_code == 'CRQ'){
            $pla_geon_feature_code_explicit = '8756'; //Value matching "Cirque".
        }
        elseif($pla_geonames_feature_code == 'CRQS'){
            $pla_geon_feature_code_explicit = '8757'; //Value matching "Cirques".
        }
        elseif($pla_geonames_feature_code == 'CRTR'){
            $pla_geon_feature_code_explicit = '8758'; //Value matching
"Cratère(s)".
        }
        elseif($pla_geonames_feature_code == 'CUET'){
            $pla_geon_feature_code_explicit = '8759'; //Value matching
"Cuesta(s)".
        }
        elseif($pla_geonames_feature_code == 'DLTA'){
            $pla_geon_feature_code_explicit = '8760'; //Value matching "Delta".
        }
        elseif($pla_geonames_feature_code == 'DPR'){
            $pla_geon_feature_code_explicit = '8761'; //Value matching
"Dépression(s)".
        }
        elseif($pla_geonames_feature_code == 'DSRT'){
            $pla_geon_feature_code_explicit = '8762'; //Value matching "Désert".
        }
        elseif($pla_geonames_feature_code == 'DUNE'){
            $pla_geon_feature_code_explicit = '8763'; //Value matching "Dune(s)".
        }
        elseif($pla_geonames_feature_code == 'DVD'){
            $pla_geon_feature_code_explicit = '8764'; //Value matching "Ligne de
partage des eaux".
        }
        elseif($pla_geonames_feature_code == 'ERG'){
            $pla_geon_feature_code_explicit = '8765'; //Value matching "Désert de
sable".
        }
        elseif($pla_geonames_feature_code == 'FAN'){
            $pla_geon_feature_code_explicit = '8766'; //Value matching "Cône(s) de
déjection".
        }
        elseif($pla_geonames_feature_code == 'FORD'){
            $pla_geon_feature_code_explicit = '8767'; //Value matching "Gué".
        }
        elseif($pla_geonames_feature_code == 'FSR'){
            $pla_geon_feature_code_explicit = '8768'; //Value matching "Fissure
volcanique".
        }
        elseif($pla_geonames_feature_code == 'GAP'){
            $pla_geon_feature_code_explicit = '8769'; //Value matching "Brèche".
        }
        elseif($pla_geonames_feature_code == 'GRGE'){
            $pla_geon_feature_code_explicit = '8770'; //Value matching "Gorge(s)".
        }
        elseif($pla_geonames_feature_code == 'HDLD'){
```

```
            $pla_geon_feature_code_explicit = '8771'; //Value matching "Cap
(headland)".
        }
        elseif($pla_geonames_feature_code == 'HLL'){
            $pla_geon_feature_code_explicit = '8772'; //Value matching "Colline".
        }
        elseif($pla_geonames_feature_code == 'HLLS'){
            $pla_geon_feature_code_explicit = '8773'; //Value matching "Collines".
        }
        elseif($pla_geonames_feature_code == 'HMCK'){
            $pla_geon_feature_code_explicit = '8774'; //Value matching
"Hammock(s)".
        }
        elseif($pla_geonames_feature_code == 'HMDA'){
            $pla_geon_feature_code_explicit = '8775'; //Value matching "Désert de
pierres (rock desert)".
        }
        elseif($pla_geonames_feature_code == 'INTF'){
            $pla_geon_feature_code_explicit = '8776'; //Value matching
"Interfluve".
        }
        elseif($pla_geonames_feature_code == 'ISL'){
            $pla_geon_feature_code_explicit = '8777'; //Value matching "Île".
        }
        elseif($pla_geonames_feature_code == 'ISLET'){
            $pla_geon_feature_code_explicit = '8778'; //Value matching "Îlot".
        }
        elseif($pla_geonames_feature_code == 'ISLF'){
            $pla_geon_feature_code_explicit = '8779'; //Value matching "Île
artificielle".
        }
        elseif($pla_geonames_feature_code == 'ISLM'){
            $pla_geon_feature_code_explicit = '8780'; //Value matching "Île de
mangrove".
        }
        elseif($pla_geonames_feature_code == 'ISLS'){
            $pla_geon_feature_code_explicit = '8781'; //Value matching "Îles".
        }
        elseif($pla_geonames_feature_code == 'ISLT'){
            $pla_geon_feature_code_explicit = '8782'; //Value matching
"Presqu'île".
        }
        elseif($pla_geonames_feature_code == 'ISLX'){
            $pla_geon_feature_code_explicit = '8783'; //Value matching "Section
d'île".
        }
        elseif($pla_geonames_feature_code == 'ISTH'){
            $pla_geon_feature_code_explicit = '8784'; //Value matching "Isthme".
        }
        elseif($pla_geonames_feature_code == 'KRST'){
            $pla_geon_feature_code_explicit = '8785'; //Value matching "Zone
karstique".
        }
        elseif($pla_geonames_feature_code == 'LAVA'){
            $pla_geon_feature_code_explicit = '8786'; //Value matching "Zone de
lave".
        }
        elseif($pla_geonames_feature_code == 'LEV'){
```

```
            $pla_geon_feature_code_explicit = '8787'; //Value matching "Digue
(levee)".
        }
        elseif($pla_geonames_feature_code == 'MESA'){
            $pla_geon_feature_code_explicit = '8788'; //Value matching "Mesa(s)".
        }
        elseif($pla_geonames_feature_code == 'MND'){
            $pla_geon_feature_code_explicit = '8789'; //Value matching
"Monticule(s) (mound)".
        }
        elseif($pla_geonames_feature_code == 'MRN'){
            $pla_geon_feature_code_explicit = '8790'; //Value matching "Moraine".
        }
        elseif($pla_geonames_feature_code == 'MT'){
            $pla_geon_feature_code_explicit = '8791'; //Value matching "Montagne".
        }
        elseif($pla_geonames_feature_code == 'MTS'){
            $pla_geon_feature_code_explicit = '8792'; //Value matching
"Montagnes".
        }
        elseif($pla_geonames_feature_code == 'NKM'){
            $pla_geon_feature_code_explicit = '8793'; //Value matching "Col de
méandre".
        }
        elseif($pla_geonames_feature_code == 'NTK'){
            $pla_geon_feature_code_explicit = '8794'; //Value matching "Nunatak".
        }
        elseif($pla_geonames_feature_code == 'NTKS'){
            $pla_geon_feature_code_explicit = '8795'; //Value matching "Nunataks".
        }
        elseif($pla_geonames_feature_code == 'PAN'){
            $pla_geon_feature_code_explicit = '8796'; //Value matching "Cuvette
(pan)".
        }
        elseif($pla_geonames_feature_code == 'PANS'){
            $pla_geon_feature_code_explicit = '8797'; //Value matching "Cuvettes
(pans)".
        }
        elseif($pla_geonames_feature_code == 'PASS'){
            $pla_geon_feature_code_explicit = '8798'; //Value matching "Col
(pass)".
        }
        elseif($pla_geonames_feature_code == 'PEN'){
            $pla_geon_feature_code_explicit = '8799'; //Value matching
"Péninsule".
        }
        elseif($pla_geonames_feature_code == 'PENX'){
            $pla_geon_feature_code_explicit = '8800'; //Value matching "Section de
péninsule".
        }
        elseif($pla_geonames_feature_code == 'PK'){
            $pla_geon_feature_code_explicit = '8801'; //Value matching "Pic".
        }
        elseif($pla_geonames_feature_code == 'PKS'){
            $pla_geon_feature_code_explicit = '8802'; //Value matching "Pics".
        }
        elseif($pla_geonames_feature_code == 'PLAT'){
            $pla_geon_feature_code_explicit = '8803'; //Value matching "Plateau".
        }
```

```
        elseif($pla_geonames_feature_code == 'PLATX'){
            $pla_geon_feature_code_explicit = '8804'; //Value matching "Section de
plateau".
        }
        elseif($pla_geonames_feature_code == 'PLDR'){
            $pla_geon_feature_code_explicit = '8805'; //Value matching "Polder".
        }
        elseif($pla_geonames_feature_code == 'PLN'){
            $pla_geon_feature_code_explicit = '8806'; //Value matching
"Plaine(s)".
        }
        elseif($pla_geonames_feature_code == 'PLNX'){
            $pla_geon_feature_code_explicit = '8807'; //Value matching "Section de
plaine".
        }
        elseif($pla_geonames_feature_code == 'PROM'){
            $pla_geon_feature_code_explicit = '8808'; //Value matching
"Promontoire(s)".
        }
        elseif($pla_geonames_feature_code == 'PT'){
            $pla_geon_feature_code_explicit = '8809'; //Value matching "Pointe".
        }
        elseif($pla_geonames_feature_code == 'PTS'){
            $pla_geon_feature_code_explicit = '8810'; //Value matching "Pointes".
        }
        elseif($pla_geonames_feature_code == 'RDGB'){
            $pla_geon_feature_code_explicit = '8811'; //Value matching "Crête de
sable".
        }
        elseif($pla_geonames_feature_code == 'RDGE'){
            $pla_geon_feature_code_explicit = '8812'; //Value matching "Crête(s)".
        }
        elseif($pla_geonames_feature_code == 'REG'){
            $pla_geon_feature_code_explicit = '8813'; //Value matching "Désert de
pierres (stony desert)".
        }
        elseif($pla_geonames_feature_code == 'RK'){
            $pla_geon_feature_code_explicit = '8814'; //Value matching "Rocher".
        }
        elseif($pla_geonames_feature_code == 'RKFL'){
            $pla_geon_feature_code_explicit = '8815'; //Value matching "Éboulement
(rockfall)".
        }
        elseif($pla_geonames_feature_code == 'RKS'){
            $pla_geon_feature_code_explicit = '8816'; //Value matching "Rochers".
        }
        elseif($pla_geonames_feature_code == 'SAND'){
            $pla_geon_feature_code_explicit = '8817'; //Value matching "Zone de
sable".
        }
        elseif($pla_geonames_feature_code == 'SBED'){
            $pla_geon_feature_code_explicit = '8818'; //Value matching "Lit d'un
cours d'eau sec".
        }
        elseif($pla_geonames_feature_code == 'SCRP'){
            $pla_geon_feature_code_explicit = '8819'; //Value matching
"Escarpement".
        }
        elseif($pla_geonames_feature_code == 'SDL'){
```

```
            $pla_geon_feature_code_explicit = '8820'; //Value matching "Col
(saddle)".
        }
        elseif($pla_geonames_feature_code == 'SHOR'){
            $pla_geon_feature_code_explicit = '8821'; //Value matching "Rivage".
        }
        elseif($pla_geonames_feature_code == 'SINK'){
            $pla_geon_feature_code_explicit = '8822'; //Value matching "Doline".
        }
        elseif($pla_geonames_feature_code == 'SLID'){
            $pla_geon_feature_code_explicit = '8823'; //Value matching "Éboulement
(slide)".
        }
        elseif($pla_geonames_feature_code == 'SLP'){
            $pla_geon_feature_code_explicit = '8824'; //Value matching "Pente(s)".
        }
        elseif($pla_geonames_feature_code == 'SPIT'){
            $pla_geon_feature_code_explicit = '8825'; //Value matching "Cordon
littoral".
        }
        elseif($pla_geonames_feature_code == 'SPUR'){
            $pla_geon_feature_code_explicit = '8826'; //Value matching
"Éperon(s)".
        }
        elseif($pla_geonames_feature_code == 'TAL'){
            $pla_geon_feature_code_explicit = '8827'; //Value matching "Talus
d'éboulis".
        }
        elseif($pla_geonames_feature_code == 'TRGD'){
            $pla_geon_feature_code_explicit = '8828'; //Value matching "Espace(s)
interdunaire(s)".
        }
        elseif($pla_geonames_feature_code == 'TRR'){
            $pla_geon_feature_code_explicit = '8829'; //Value matching "Terrasse".
        }
        elseif($pla_geonames_feature_code == 'UPLD'){
            $pla_geon_feature_code_explicit = '8830'; //Value matching "Hautes
terres".
        }
        elseif($pla_geonames_feature_code == 'VAL'){
            $pla_geon_feature_code_explicit = '8831'; //Value matching "Vallée".
        }
        elseif($pla_geonames_feature_code == 'VALG'){
            $pla_geon_feature_code_explicit = '8832'; //Value matching "Vallée
suspendue".
        }
        elseif($pla_geonames_feature_code == 'VALS'){
            $pla_geon_feature_code_explicit = '8833'; //Value matching "Vallées".
        }
        elseif($pla_geonames_feature_code == 'VALX'){
            $pla_geon_feature_code_explicit = '8834'; //Value matching "Section de
vallée".
        }
        elseif($pla_geonames_feature_code == 'VLC'){
            $pla_geon_feature_code_explicit = '8835'; //Value matching "Volcan".
        }
        elseif($pla_geonames_feature_code == 'APNU'){
            $pla_geon_feature_code_explicit = '8836'; //Value matching "Tablier
archipélagique".
```

```
        }
        elseif($pla_geonames_feature_code == 'ARCU'){
            $pla_geon_feature_code_explicit = '8837'; //Value matching "Arche
(Hawaï)".
        }
        elseif($pla_geonames_feature_code == 'ARRU'){
            $pla_geon_feature_code_explicit = '8838'; //Value matching "Arrugado
(Basse-Californie)".
        }
        elseif($pla_geonames_feature_code == 'BDLU'){
            $pla_geon_feature_code_explicit = '8839'; //Value matching "Marge
continentale".
        }
        elseif($pla_geonames_feature_code == 'BKSU'){
            $pla_geon_feature_code_explicit = '8840'; //Value matching "Bancs
sous-marins".
        }
        elseif($pla_geonames_feature_code == 'BNKU'){
            $pla_geon_feature_code_explicit = '8841'; //Value matching "Banc sous-
marin".
        }
        elseif($pla_geonames_feature_code == 'BSNU'){
            $pla_geon_feature_code_explicit = '8842'; //Value matching "Bassin
sous-marin".
        }
        elseif($pla_geonames_feature_code == 'CDAU'){
            $pla_geon_feature_code_explicit = '8843'; //Value matching "Cordillère
sous-marine".
        }
        elseif($pla_geonames_feature_code == 'CNSU'){
            $pla_geon_feature_code_explicit = '8844'; //Value matching "Canyons
sous-marins".
        }
        elseif($pla_geonames_feature_code == 'CNYU'){
            $pla_geon_feature_code_explicit = '8845'; //Value matching "Canyon
sous-marin".
        }
        elseif($pla_geonames_feature_code == 'CRSU'){
            $pla_geon_feature_code_explicit = '8846'; //Value matching "Glacis
continental".
        }
        elseif($pla_geonames_feature_code == 'DEPU'){
            $pla_geon_feature_code_explicit = '8847'; //Value matching
"Profondeurs".
        }
        elseif($pla_geonames_feature_code == 'EDGU'){
            $pla_geon_feature_code_explicit = '8848'; //Value matching "Bordure du
plateau continental".
        }
        elseif($pla_geonames_feature_code == 'ESCU'){
            $pla_geon_feature_code_explicit = '8849'; //Value matching
"Escarpement sous-marin".
        }
        elseif($pla_geonames_feature_code == 'FANU'){
            $pla_geon_feature_code_explicit = '8850'; //Value matching "Éventail
abyssal".
        }
        elseif($pla_geonames_feature_code == 'FLTU'){
```

```
            $pla_geon_feature_code_explicit = '8851'; //Value matching "Zone sous-
marine de faible relief et peu profonde (flat)".
        }
        elseif($pla_geonames_feature_code == 'FRZU'){
            $pla_geon_feature_code_explicit = '8852'; //Value matching "Zone de
fracture".
        }
        elseif($pla_geonames_feature_code == 'FURU'){
            $pla_geon_feature_code_explicit = '8853'; //Value matching "Sillon
sous-marin".
        }
        elseif($pla_geonames_feature_code == 'GAPU'){
            $pla_geon_feature_code_explicit = '8854'; //Value matching "Brèche
sous-marine (gap)".
        }
        elseif($pla_geonames_feature_code == 'GLYU'){
            $pla_geon_feature_code_explicit = '8855'; //Value matching "Ravine
sous-marine".
        }
        elseif($pla_geonames_feature_code == 'HLLU'){
            $pla_geon_feature_code_explicit = '8856'; //Value matching "Colline
sous-marine".
        }
        elseif($pla_geonames_feature_code == 'HLSU'){
            $pla_geon_feature_code_explicit = '8857'; //Value matching "Collines
sous-marines".
        }
        elseif($pla_geonames_feature_code == 'HOLU'){
            $pla_geon_feature_code_explicit = '8858'; //Value matching "Trou sous-
marin".
        }
        elseif($pla_geonames_feature_code == 'KNLU'){
            $pla_geon_feature_code_explicit = '8859'; //Value matching "Mont sous-
marin (knoll)".
        }
        elseif($pla_geonames_feature_code == 'KNSU'){
            $pla_geon_feature_code_explicit = '8860'; //Value matching "Monts
sous-marins (knolls)".
        }
        elseif($pla_geonames_feature_code == 'LDGU'){
            $pla_geon_feature_code_explicit = '8861'; //Value matching "Crête
rocheuse sous-marine (ledge)".
        }
        elseif($pla_geonames_feature_code == 'LEVU'){
            $pla_geon_feature_code_explicit = '8862'; //Value matching "Levée
sous-marine".
        }
        elseif($pla_geonames_feature_code == 'MESU'){
            $pla_geon_feature_code_explicit = '8863'; //Value matching "Mesa sous-
marine".
        }
        elseif($pla_geonames_feature_code == 'MNDU'){
            $pla_geon_feature_code_explicit = '8864'; //Value matching "Monticule
sous-marin (mound)".
        }
        elseif($pla_geonames_feature_code == 'MOTU'){
            $pla_geon_feature_code_explicit = '8865'; //Value matching "Fossé
sous-marin (moat)".
        }
```

```
        elseif($pla_geonames_feature_code == 'MTU'){
            $pla_geon_feature_code_explicit = '8866'; //Value matching "Mont sous-
marin (mountain)".
        }
        elseif($pla_geonames_feature_code == 'PKSU'){
            $pla_geon_feature_code_explicit = '8867'; //Value matching "Pics sous-
marins".
        }
        elseif($pla_geonames_feature_code == 'PKU'){
            $pla_geon_feature_code_explicit = '8868'; //Value matching "Pic sous-
marin".
        }
        elseif($pla_geonames_feature_code == 'PLNU'){
            $pla_geon_feature_code_explicit = '8869'; //Value matching "Plaine
sous-marine".
        }
        elseif($pla_geonames_feature_code == 'PLTU'){
            $pla_geon_feature_code_explicit = '8870'; //Value matching "Plateau
sous-marin".
        }
        elseif($pla_geonames_feature_code == 'PNLU'){
            $pla_geon_feature_code_explicit = '8871'; //Value matching "Pinacle
sous-marin".
        }
        elseif($pla_geonames_feature_code == 'PRVU'){
            $pla_geon_feature_code_explicit = '8872'; //Value matching "Région
sous-marine".
        }
        elseif($pla_geonames_feature_code == 'RDGU'){
            $pla_geon_feature_code_explicit = '8873'; //Value matching "Crête
sous-marine (ridge)".
        }
        elseif($pla_geonames_feature_code == 'RDSU'){
            $pla_geon_feature_code_explicit = '8874'; //Value matching "Crêtes
sous-marines (ridges)".
        }
        elseif($pla_geonames_feature_code == 'RFSU'){
            $pla_geon_feature_code_explicit = '8875'; //Value matching "Récifs
sous-marins".
        }
        elseif($pla_geonames_feature_code == 'RFU'){
            $pla_geon_feature_code_explicit = '8876'; //Value matching "Récif
sous-marin".
        }
        elseif($pla_geonames_feature_code == 'RISU'){
            $pla_geon_feature_code_explicit = '8877'; //Value matching "Élévation
sous-marine".
        }
        elseif($pla_geonames_feature_code == 'SCNU'){
            $pla_geon_feature_code_explicit = '8878'; //Value matching "Chenal
sous-marin (sea channel)".
        }
        elseif($pla_geonames_feature_code == 'SCSU'){
            $pla_geon_feature_code_explicit = '8879'; //Value matching "Chenaux
sous-marins (sea channels)".
        }
        elseif($pla_geonames_feature_code == 'SDLU'){
            $pla_geon_feature_code_explicit = '8880'; //Value matching "Col sous-
marin (saddle)".
```

```
        }
        elseif($pla_geonames_feature_code == 'SHFU'){
            $pla_geon_feature_code_explicit = '8881'; //Value matching "Plateau
continental".
        }
        elseif($pla_geonames_feature_code == 'SHLU'){
            $pla_geon_feature_code_explicit = '8882'; //Value matching "Haut-
fond".
        }
        elseif($pla_geonames_feature_code == 'SHSU'){
            $pla_geon_feature_code_explicit = '8883'; //Value matching "Hauts-
fonds".
        }
        elseif($pla_geonames_feature_code == 'SHVU'){
            $pla_geon_feature_code_explicit = '8884'; //Value matching "Vallée
sous-marine continentale (shelf valley)".
        }
        elseif($pla_geonames_feature_code == 'SILU'){
            $pla_geon_feature_code_explicit = '8885'; //Value matching "Seuil
aquatique (sous-marin)".
        }
        elseif($pla_geonames_feature_code == 'SLPU'){
            $pla_geon_feature_code_explicit = '8886'; //Value matching "Talus
continental".
        }
        elseif($pla_geonames_feature_code == 'SMSU'){
            $pla_geon_feature_code_explicit = '8887'; //Value matching "Monts
sous-marins (seamounts)".
        }
        elseif($pla_geonames_feature_code == 'SMU'){
            $pla_geon_feature_code_explicit = '8888'; //Value matching "Mont sous-
marin (seamount)".
        }
        elseif($pla_geonames_feature_code == 'SPRU'){
            $pla_geon_feature_code_explicit = '8889'; //Value matching "Éperon
sous-marin".
        }
        elseif($pla_geonames_feature_code == 'TERU'){
            $pla_geon_feature_code_explicit = '8890'; //Value matching "Terrasse
sous-marine".
        }
        elseif($pla_geonames_feature_code == 'TMSU'){
            $pla_geon_feature_code_explicit = '8891'; //Value matching "Guyots".
        }
        elseif($pla_geonames_feature_code == 'TMTU'){
            $pla_geon_feature_code_explicit = '8892'; //Value matching "Guyot".
        }
        elseif($pla_geonames_feature_code == 'TNGU'){
            $pla_geon_feature_code_explicit = '8893'; //Value matching "Extension
de la plaine sous-marine dans une région plus accidentée (tongue)".
        }
        elseif($pla_geonames_feature_code == 'TRGU'){
            $pla_geon_feature_code_explicit = '8894'; //Value matching "Couloir
sous-marin (trough)".
        }
        elseif($pla_geonames_feature_code == 'TRNU'){
            $pla_geon_feature_code_explicit = '8895'; //Value matching "Fosse
sous-marine".
        }
```

```
        elseif($pla_geonames_feature_code == 'VALU'){
            $pla_geon_feature_code_explicit = '8896'; //Value matching "Vallée
sous-marine (valley)".
        }
        elseif($pla_geonames_feature_code == 'VLSU'){
            $pla_geon_feature_code_explicit = '8897'; //Value matching "Vallées
sous-marines (valleys)".
        }
        elseif($pla_geonames_feature_code == 'BUSH'){
            $pla_geon_feature_code_explicit = '8898'; //Value matching
"Buisson(s)".
        }
        elseif($pla_geonames_feature_code == 'CULT'){
            $pla_geon_feature_code_explicit = '8899'; //Value matching "Zone
cultivée".
        }
        elseif($pla_geonames_feature_code == 'FRST'){
            $pla_geon_feature_code_explicit = '8900'; //Value matching "Forêt(s)".
        }
        elseif($pla_geonames_feature_code == 'FRSTF'){
            $pla_geon_feature_code_explicit = '8901'; //Value matching "Forêt
fossilisée".
        }
        elseif($pla_geonames_feature_code == 'GROVE'){
            $pla_geon_feature_code_explicit = '8902'; //Value matching "Bosquet".
        }
        elseif($pla_geonames_feature_code == 'GRSLD'){
            $pla_geon_feature_code_explicit = '8903'; //Value matching "Pelouse".
        }
        elseif($pla_geonames_feature_code == 'GRVC'){
            $pla_geon_feature_code_explicit = '8904'; //Value matching
"Cocoteraie".
        }
        elseif($pla_geonames_feature_code == 'GRVO'){
            $pla_geon_feature_code_explicit = '8905'; //Value matching
"Oliveraie".
        }
        elseif($pla_geonames_feature_code == 'GRVP'){
            $pla_geon_feature_code_explicit = '8906'; //Value matching
"Palmeraie".
        }
        elseif($pla_geonames_feature_code == 'GRVPN'){
            $pla_geon_feature_code_explicit = '8907'; //Value matching "Pinède".
        }
        elseif($pla_geonames_feature_code == 'HTH'){
            $pla_geon_feature_code_explicit = '8908'; //Value matching "Lande
(heath)".
        }
        elseif($pla_geonames_feature_code == 'MDW'){
            $pla_geon_feature_code_explicit = '8909'; //Value matching "Prairie".
        }
        elseif($pla_geonames_feature_code == 'OCH'){
            $pla_geon_feature_code_explicit = '8910'; //Value matching
"Verger(s)".
        }
        elseif($pla_geonames_feature_code == 'SCRB'){
            $pla_geon_feature_code_explicit = '8911'; //Value matching "Fruticée".
        }
        elseif($pla_geonames_feature_code == 'TREE'){
```

```
            $pla_geon_feature_code_explicit = '8912'; //Value matching "Arbre(s)".
        }
        elseif($pla_geonames_feature_code == 'TUND'){
            $pla_geon_feature_code_explicit = '8913'; //Value matching "Toundra".
        }
        elseif($pla_geonames_feature_code == 'VIN'){
            $pla_geon_feature_code_explicit = '8914'; //Value matching "Vignoble".
        }
        elseif($pla_geonames_feature_code == 'VINS'){
            $pla_geon_feature_code_explicit = '8915'; //Value matching
"Vignobles".
        }
        elseif($pla_geonames_feature_code == 'll'){
            $pla_geon_feature_code_explicit = '8916'; //Value matching "[Non
disponible]".
        }
        else{
            $pla_geon_feature_code_explicit = '';
        }

//      Create field pla_geonames_country_code from country_code.

        $pla_geonames_country_code = $country_code;

        if(stripos($pla_geonames_country_code, ',') !== false ||
stripos($pla_geonames_country_code, '"') !== false){
            $pla_geonames_country_code = '"'.$pla_geonames_country_code.'"';
        }

//      Create field pla_geon_country_code_explicit from previous field, according
to "Nom de pays" taxonomy.

        if($pla_geonames_country_code == 'AF'){
            $pla_geon_country_code_explicit = '8917'; //Value matching
"Afghanistan".
        }
        elseif($pla_geonames_country_code == 'AL'){
            $pla_geon_country_code_explicit = '8918'; //Value matching "Albanie".
        }
        elseif($pla_geonames_country_code == 'DZ'){
            $pla_geon_country_code_explicit = '8919'; //Value matching "Algérie".
        }
        elseif($pla_geonames_country_code == 'AS'){
            $pla_geon_country_code_explicit = '8920'; //Value matching "Samoa
américaines".
        }
        elseif($pla_geonames_country_code == 'AD'){
            $pla_geon_country_code_explicit = '8921'; //Value matching "Andorre".
        }
        elseif($pla_geonames_country_code == 'AO'){
            $pla_geon_country_code_explicit = '8922'; //Value matching "Angola".
        }
        elseif($pla_geonames_country_code == 'AI'){
            $pla_geon_country_code_explicit = '8923'; //Value matching "Anguilla".
        }
        elseif($pla_geonames_country_code == 'AQ'){
            $pla_geon_country_code_explicit = '8924'; //Value matching
"Antarctique".
        }
```

```
        elseif($pla_geonames_country_code == 'AG'){
            $pla_geon_country_code_explicit = '8925'; //Value matching "Antigua-
et-Barbuda".
        }
        elseif($pla_geonames_country_code == 'AR'){
            $pla_geon_country_code_explicit = '8926'; //Value matching
"Argentine".
        }
        elseif($pla_geonames_country_code == 'AM'){
            $pla_geon_country_code_explicit = '8927'; //Value matching "Arménie".
        }
        elseif($pla_geonames_country_code == 'AW'){
            $pla_geon_country_code_explicit = '8928'; //Value matching "Aruba".
        }
        elseif($pla_geonames_country_code == 'AU'){
            $pla_geon_country_code_explicit = '8929'; //Value matching
"Australie".
        }
        elseif($pla_geonames_country_code == 'AT'){
            $pla_geon_country_code_explicit = '8930'; //Value matching "Autriche".
        }
        elseif($pla_geonames_country_code == 'AZ'){
            $pla_geon_country_code_explicit = '8931'; //Value matching
"Azerbaïdjan".
        }
        elseif($pla_geonames_country_code == 'BS'){
            $pla_geon_country_code_explicit = '8932'; //Value matching "Bahamas".
        }
        elseif($pla_geonames_country_code == 'BH'){
            $pla_geon_country_code_explicit = '8933'; //Value matching "Bahreïn".
        }
        elseif($pla_geonames_country_code == 'BD'){
            $pla_geon_country_code_explicit = '8934'; //Value matching
"Bangladesh".
        }
        elseif($pla_geonames_country_code == 'BB'){
            $pla_geon_country_code_explicit = '8935'; //Value matching "Barbade".
        }
        elseif($pla_geonames_country_code == 'BY'){
            $pla_geon_country_code_explicit = '8936'; //Value matching
"Biélorussie".
        }
        elseif($pla_geonames_country_code == 'BE'){
            $pla_geon_country_code_explicit = '8937'; //Value matching "Belgique".
        }
        elseif($pla_geonames_country_code == 'BZ'){
            $pla_geon_country_code_explicit = '8938'; //Value matching "Belize".
        }
        elseif($pla_geonames_country_code == 'BJ'){
            $pla_geon_country_code_explicit = '8939'; //Value matching "Bénin".
        }
        elseif($pla_geonames_country_code == 'BM'){
            $pla_geon_country_code_explicit = '8940'; //Value matching "Bermudes".
        }
        elseif($pla_geonames_country_code == 'BT'){
            $pla_geon_country_code_explicit = '8941'; //Value matching "Bhoutan".
        }
        elseif($pla_geonames_country_code == 'BO'){
            $pla_geon_country_code_explicit = '8942'; //Value matching "Bolivie".
```

```
        }
        elseif($pla_geonames_country_code == 'BQ'){
            $pla_geon_country_code_explicit = '8943'; //Value matching "Bonaire,
Saint-Eustache et Saba".
        }
        elseif($pla_geonames_country_code == 'BA'){
            $pla_geon_country_code_explicit = '8944'; //Value matching "Bosnie-
Herzégovine".
        }
        elseif($pla_geonames_country_code == 'BW'){
            $pla_geon_country_code_explicit = '8945'; //Value matching "Botswana".
        }
        elseif($pla_geonames_country_code == 'BV'){
            $pla_geon_country_code_explicit = '8946'; //Value matching "Bouvet
(Île)".
        }
        elseif($pla_geonames_country_code == 'BR'){
            $pla_geon_country_code_explicit = '8947'; //Value matching "Brésil".
        }
        elseif($pla_geonames_country_code == 'IO'){
            $pla_geon_country_code_explicit = '8948'; //Value matching "Territoire
britannique de l'océan Indien".
        }
        elseif($pla_geonames_country_code == 'BN'){
            $pla_geon_country_code_explicit = '8949'; //Value matching "Brunéi
Darussalam".
        }
        elseif($pla_geonames_country_code == 'BG'){
            $pla_geon_country_code_explicit = '8950'; //Value matching "Bulgarie".
        }
        elseif($pla_geonames_country_code == 'BF'){
            $pla_geon_country_code_explicit = '8951'; //Value matching "Burkina
Faso".
        }
        elseif($pla_geonames_country_code == 'BI'){
            $pla_geon_country_code_explicit = '8952'; //Value matching "Burundi".
        }
        elseif($pla_geonames_country_code == 'CV'){
            $pla_geon_country_code_explicit = '8953'; //Value matching "Cap-Vert".
        }
        elseif($pla_geonames_country_code == 'KH'){
            $pla_geon_country_code_explicit = '8954'; //Value matching "Cambodge".
        }
        elseif($pla_geonames_country_code == 'CM'){
            $pla_geon_country_code_explicit = '8955'; //Value matching "Cameroun".
        }
        elseif($pla_geonames_country_code == 'CA'){
            $pla_geon_country_code_explicit = '8956'; //Value matching "Canada".
        }
        elseif($pla_geonames_country_code == 'KY'){
            $pla_geon_country_code_explicit = '8957'; //Value matching "Caïmans
(îles)".
        }
        elseif($pla_geonames_country_code == 'CF'){
            $pla_geon_country_code_explicit = '8958'; //Value matching "République
centrafricaine".
        }
        elseif($pla_geonames_country_code == 'TD'){
            $pla_geon_country_code_explicit = '8959'; //Value matching "Tchad".
```

```
        }
        elseif($pla_geonames_country_code == 'CL'){
            $pla_geon_country_code_explicit = '8960'; //Value matching "Chili".
        }
        elseif($pla_geonames_country_code == 'CN'){
            $pla_geon_country_code_explicit = '8961'; //Value matching "Chine".
        }
        elseif($pla_geonames_country_code == 'CX'){
            $pla_geon_country_code_explicit = '8962'; //Value matching "Christmas
(île)".
        }
        elseif($pla_geonames_country_code == 'CC'){
            $pla_geon_country_code_explicit = '8963'; //Value matching "Cocos
(îles)".
        }
        elseif($pla_geonames_country_code == 'CO'){
            $pla_geon_country_code_explicit = '8964'; //Value matching "Colombie".
        }
        elseif($pla_geonames_country_code == 'KM'){
            $pla_geon_country_code_explicit = '8965'; //Value matching "Comores".
        }
        elseif($pla_geonames_country_code == 'CD'){
            $pla_geon_country_code_explicit = '8966'; //Value matching "Congo
(République démocratique du)".
        }
        elseif($pla_geonames_country_code == 'CG'){
            $pla_geon_country_code_explicit = '8967'; //Value matching "Congo".
        }
        elseif($pla_geonames_country_code == 'CK'){
            $pla_geon_country_code_explicit = '8968'; //Value matching "Cook
(îles)".
        }
        elseif($pla_geonames_country_code == 'CR'){
            $pla_geon_country_code_explicit = '8969'; //Value matching "Costa
Rica".
        }
        elseif($pla_geonames_country_code == 'HR'){
            $pla_geon_country_code_explicit = '8970'; //Value matching "Croatie".
        }
        elseif($pla_geonames_country_code == 'CU'){
            $pla_geon_country_code_explicit = '8971'; //Value matching "Cuba".
        }
        elseif($pla_geonames_country_code == 'CW'){
            $pla_geon_country_code_explicit = '8972'; //Value matching "Curaçao".
        }
        elseif($pla_geonames_country_code == 'CY'){
            $pla_geon_country_code_explicit = '8973'; //Value matching "Chypre".
        }
        elseif($pla_geonames_country_code == 'CZ'){
            $pla_geon_country_code_explicit = '8974'; //Value matching "Tchéquie".
        }
        elseif($pla_geonames_country_code == 'CI'){
            $pla_geon_country_code_explicit = '8975'; //Value matching "Côte
d'Ivoire".
        }
        elseif($pla_geonames_country_code == 'DK'){
            $pla_geon_country_code_explicit = '8976'; //Value matching "Danemark".
        }
        elseif($pla_geonames_country_code == 'DJ'){
```

```
            $pla_geon_country_code_explicit = '8977'; //Value matching "Djibouti".
        }
        elseif($pla_geonames_country_code == 'DM'){
            $pla_geon_country_code_explicit = '8978'; //Value matching
"Dominique".
        }
        elseif($pla_geonames_country_code == 'DO'){
            $pla_geon_country_code_explicit = '8979'; //Value matching "République
dominicaine".
        }
        elseif($pla_geonames_country_code == 'EC'){
            $pla_geon_country_code_explicit = '8980'; //Value matching "Équateur".
        }
        elseif($pla_geonames_country_code == 'EG'){
            $pla_geon_country_code_explicit = '8981'; //Value matching "Égypte".
        }
        elseif($pla_geonames_country_code == 'SV'){
            $pla_geon_country_code_explicit = '8982'; //Value matching "El
Salvador".
        }
        elseif($pla_geonames_country_code == 'GQ'){
            $pla_geon_country_code_explicit = '8983'; //Value matching "Guinée
équatoriale".
        }
        elseif($pla_geonames_country_code == 'ER'){
            $pla_geon_country_code_explicit = '8984'; //Value matching "Érythrée".
        }
        elseif($pla_geonames_country_code == 'EE'){
            $pla_geon_country_code_explicit = '8985'; //Value matching "Estonie".
        }
        elseif($pla_geonames_country_code == 'SZ'){
            $pla_geon_country_code_explicit = '8986'; //Value matching "Eswatini".
        }
        elseif($pla_geonames_country_code == 'ET'){
            $pla_geon_country_code_explicit = '8987'; //Value matching "Éthiopie".
        }
        elseif($pla_geonames_country_code == 'FK'){
            $pla_geon_country_code_explicit = '8988'; //Value matching "Malouines
(îles) / Falkland (îles)".
        }
        elseif($pla_geonames_country_code == 'FO'){
            $pla_geon_country_code_explicit = '8989'; //Value matching "Féroé
(îles)".
        }
        elseif($pla_geonames_country_code == 'FJ'){
            $pla_geon_country_code_explicit = '8990'; //Value matching "Fidji".
        }
        elseif($pla_geonames_country_code == 'FI'){
            $pla_geon_country_code_explicit = '8991'; //Value matching "Finlande".
        }
        elseif($pla_geonames_country_code == 'FR'){
            $pla_geon_country_code_explicit = '8992'; //Value matching "France".
        }
        elseif($pla_geonames_country_code == 'GF'){
            $pla_geon_country_code_explicit = '8993'; //Value matching "Guyane
française".
        }
        elseif($pla_geonames_country_code == 'PF'){
```

```
            $pla_geon_country_code_explicit = '8994'; //Value matching "Polynésie
française".
        }
        elseif($pla_geonames_country_code == 'TF'){
            $pla_geon_country_code_explicit = '8995'; //Value matching "Terres
australes et antarctiques françaises".
        }
        elseif($pla_geonames_country_code == 'GA'){
            $pla_geon_country_code_explicit = '8996'; //Value matching "Gabon".
        }
        elseif($pla_geonames_country_code == 'GM'){
            $pla_geon_country_code_explicit = '8997'; //Value matching "Gambie".
        }
        elseif($pla_geonames_country_code == 'GE'){
            $pla_geon_country_code_explicit = '8998'; //Value matching "Géorgie".
        }
        elseif($pla_geonames_country_code == 'DE'){
            $pla_geon_country_code_explicit = '8999'; //Value matching
"Allemagne".
        }
        elseif($pla_geonames_country_code == 'GH'){
            $pla_geon_country_code_explicit = '9000'; //Value matching "Ghana".
        }
        elseif($pla_geonames_country_code == 'GI'){
            $pla_geon_country_code_explicit = '9001'; //Value matching
"Gibraltar".
        }
        elseif($pla_geonames_country_code == 'GR'){
            $pla_geon_country_code_explicit = '9002'; //Value matching "Grèce".
        }
        elseif($pla_geonames_country_code == 'GL'){
            $pla_geon_country_code_explicit = '9003'; //Value matching
"Groenland".
        }
        elseif($pla_geonames_country_code == 'GD'){
            $pla_geon_country_code_explicit = '9004'; //Value matching "Grenade".
        }
        elseif($pla_geonames_country_code == 'GP'){
            $pla_geon_country_code_explicit = '9005'; //Value matching
"Guadeloupe".
        }
        elseif($pla_geonames_country_code == 'GU'){
            $pla_geon_country_code_explicit = '9006'; //Value matching "Guam".
        }
        elseif($pla_geonames_country_code == 'GT'){
            $pla_geon_country_code_explicit = '9007'; //Value matching
"Guatemala".
        }
        elseif($pla_geonames_country_code == 'GG'){
            $pla_geon_country_code_explicit = '9008'; //Value matching
"Guernesey".
        }
        elseif($pla_geonames_country_code == 'GN'){
            $pla_geon_country_code_explicit = '9009'; //Value matching "Guinée".
        }
        elseif($pla_geonames_country_code == 'GW'){
            $pla_geon_country_code_explicit = '9010'; //Value matching "Guinée-
Bissau".
        }
```

```
        elseif($pla_geonames_country_code == 'GY'){
            $pla_geon_country_code_explicit = '9011'; //Value matching "Guyana".
        }
        elseif($pla_geonames_country_code == 'HT'){
            $pla_geon_country_code_explicit = '9012'; //Value matching "Haïti".
        }
        elseif($pla_geonames_country_code == 'HM'){
            $pla_geon_country_code_explicit = '9013'; //Value matching "Heard-et-
MacDonald (îles)".
        }
        elseif($pla_geonames_country_code == 'VA'){
            $pla_geon_country_code_explicit = '9014'; //Value matching "Vatican".
        }
        elseif($pla_geonames_country_code == 'HN'){
            $pla_geon_country_code_explicit = '9015'; //Value matching "Honduras".
        }
        elseif($pla_geonames_country_code == 'HK'){
            $pla_geon_country_code_explicit = '9016'; //Value matching "Hong
Kong".
        }
        elseif($pla_geonames_country_code == 'HU'){
            $pla_geon_country_code_explicit = '9017'; //Value matching "Hongrie".
        }
        elseif($pla_geonames_country_code == 'IS'){
            $pla_geon_country_code_explicit = '9018'; //Value matching "Islande".
        }
        elseif($pla_geonames_country_code == 'IN'){
            $pla_geon_country_code_explicit = '9019'; //Value matching "Inde".
        }
        elseif($pla_geonames_country_code == 'ID'){
            $pla_geon_country_code_explicit = '9020'; //Value matching
"Indonésie".
        }
        elseif($pla_geonames_country_code == 'IR'){
            $pla_geon_country_code_explicit = '9021'; //Value matching "Iran".
        }
        elseif($pla_geonames_country_code == 'IQ'){
            $pla_geon_country_code_explicit = '9022'; //Value matching "Iraq".
        }
        elseif($pla_geonames_country_code == 'IE'){
            $pla_geon_country_code_explicit = '9023'; //Value matching "Irlande".
        }
        elseif($pla_geonames_country_code == 'IM'){
            $pla_geon_country_code_explicit = '9024'; //Value matching "Île de
Man".
        }
        elseif($pla_geonames_country_code == 'IL'){
            $pla_geon_country_code_explicit = '9025'; //Value matching "Israël".
        }
        elseif($pla_geonames_country_code == 'IT'){
            $pla_geon_country_code_explicit = '9026'; //Value matching "Italie".
        }
        elseif($pla_geonames_country_code == 'JM'){
            $pla_geon_country_code_explicit = '9027'; //Value matching "Jamaïque".
        }
        elseif($pla_geonames_country_code == 'JP'){
            $pla_geon_country_code_explicit = '9028'; //Value matching "Japon".
        }
        elseif($pla_geonames_country_code == 'JE'){
```

```
            $pla_geon_country_code_explicit = '9029'; //Value matching "Jersey".
        }
        elseif($pla_geonames_country_code == 'JO'){
            $pla_geon_country_code_explicit = '9030'; //Value matching "Jordanie".
        }
        elseif($pla_geonames_country_code == 'KZ'){
            $pla_geon_country_code_explicit = '9031'; //Value matching
"Kazakhstan".
        }
        elseif($pla_geonames_country_code == 'KE'){
            $pla_geon_country_code_explicit = '9032'; //Value matching "Kenya".
        }
        elseif($pla_geonames_country_code == 'KI'){
            $pla_geon_country_code_explicit = '9033'; //Value matching "Kiribati".
        }
        elseif($pla_geonames_country_code == 'KP'){
            $pla_geon_country_code_explicit = '9034'; //Value matching "Corée du
Nord".
        }
        elseif($pla_geonames_country_code == 'KR'){
            $pla_geon_country_code_explicit = '9035'; //Value matching "Corée du
Sud".
        }
        elseif($pla_geonames_country_code == 'KW'){
            $pla_geon_country_code_explicit = '9036'; //Value matching "Koweït".
        }
        elseif($pla_geonames_country_code == 'KG'){
            $pla_geon_country_code_explicit = '9037'; //Value matching
"Kirghizistan".
        }
        elseif($pla_geonames_country_code == 'LA'){
            $pla_geon_country_code_explicit = '9038'; //Value matching "Laos".
        }
        elseif($pla_geonames_country_code == 'LV'){
            $pla_geon_country_code_explicit = '9039'; //Value matching "Lettonie".
        }
        elseif($pla_geonames_country_code == 'LB'){
            $pla_geon_country_code_explicit = '9040'; //Value matching "Liban".
        }
        elseif($pla_geonames_country_code == 'LS'){
            $pla_geon_country_code_explicit = '9041'; //Value matching "Lesotho".
        }
        elseif($pla_geonames_country_code == 'LR'){
            $pla_geon_country_code_explicit = '9042'; //Value matching "Libéria".
        }
        elseif($pla_geonames_country_code == 'LY'){
            $pla_geon_country_code_explicit = '9043'; //Value matching "Libye".
        }
        elseif($pla_geonames_country_code == 'LI'){
            $pla_geon_country_code_explicit = '9044'; //Value matching
"Liechtenstein".
        }
        elseif($pla_geonames_country_code == 'LT'){
            $pla_geon_country_code_explicit = '9045'; //Value matching "Lituanie".
        }
        elseif($pla_geonames_country_code == 'LU'){
            $pla_geon_country_code_explicit = '9046'; //Value matching
"Luxembourg".
        }
```

```
        elseif($pla_geonames_country_code == 'MO'){
            $pla_geon_country_code_explicit = '9047'; //Value matching "Macao".
        }
        elseif($pla_geonames_country_code == 'MG'){
            $pla_geon_country_code_explicit = '9048'; //Value matching
"Madagascar".
        }
        elseif($pla_geonames_country_code == 'MW'){
            $pla_geon_country_code_explicit = '9049'; //Value matching "Malawi".
        }
        elseif($pla_geonames_country_code == 'MY'){
            $pla_geon_country_code_explicit = '9050'; //Value matching "Malaisie".
        }
        elseif($pla_geonames_country_code == 'MV'){
            $pla_geon_country_code_explicit = '9051'; //Value matching "Maldives".
        }
        elseif($pla_geonames_country_code == 'ML'){
            $pla_geon_country_code_explicit = '9052'; //Value matching "Mali".
        }
        elseif($pla_geonames_country_code == 'MT'){
            $pla_geon_country_code_explicit = '9053'; //Value matching "Malte".
        }
        elseif($pla_geonames_country_code == 'MH'){
            $pla_geon_country_code_explicit = '9054'; //Value matching "Marshall
(îles)".
        }
        elseif($pla_geonames_country_code == 'MQ'){
            $pla_geon_country_code_explicit = '9055'; //Value matching
"Martinique".
        }
        elseif($pla_geonames_country_code == 'MR'){
            $pla_geon_country_code_explicit = '9056'; //Value matching
"Mauritanie".
        }
        elseif($pla_geonames_country_code == 'MU'){
            $pla_geon_country_code_explicit = '9057'; //Value matching "Maurice".
        }
        elseif($pla_geonames_country_code == 'YT'){
            $pla_geon_country_code_explicit = '9058'; //Value matching "Mayotte".
        }
        elseif($pla_geonames_country_code == 'MX'){
            $pla_geon_country_code_explicit = '9059'; //Value matching "Mexique".
        }
        elseif($pla_geonames_country_code == 'FM'){
            $pla_geon_country_code_explicit = '9060'; //Value matching "États
fédérés de Micronésie".
        }
        elseif($pla_geonames_country_code == 'MD'){
            $pla_geon_country_code_explicit = '9061'; //Value matching "Moldavie".
        }
        elseif($pla_geonames_country_code == 'MC'){
            $pla_geon_country_code_explicit = '9062'; //Value matching "Monaco".
        }
        elseif($pla_geonames_country_code == 'MN'){
            $pla_geon_country_code_explicit = '9063'; //Value matching "Mongolie".
        }
        elseif($pla_geonames_country_code == 'ME'){
            $pla_geon_country_code_explicit = '9064'; //Value matching
"Monténégro".
```

```
        }
        elseif($pla_geonames_country_code == 'MS'){
            $pla_geon_country_code_explicit = '9065'; //Value matching
"Montserrat".
        }
        elseif($pla_geonames_country_code == 'MA'){
            $pla_geon_country_code_explicit = '9066'; //Value matching "Maroc".
        }
        elseif($pla_geonames_country_code == 'MZ'){
            $pla_geon_country_code_explicit = '9067'; //Value matching
"Mozambique".
        }
        elseif($pla_geonames_country_code == 'MM'){
            $pla_geon_country_code_explicit = '9068'; //Value matching "Birmanie
(Myanmar)".
        }
        elseif($pla_geonames_country_code == 'NA'){
            $pla_geon_country_code_explicit = '9069'; //Value matching "Namibie".
        }
        elseif($pla_geonames_country_code == 'NR'){
            $pla_geon_country_code_explicit = '9070'; //Value matching "Nauru".
        }
        elseif($pla_geonames_country_code == 'NP'){
            $pla_geon_country_code_explicit = '9071'; //Value matching "Népal".
        }
        elseif($pla_geonames_country_code == 'NL'){
            $pla_geon_country_code_explicit = '9072'; //Value matching "Pays-Bas".
        }
        elseif($pla_geonames_country_code == 'NC'){
            $pla_geon_country_code_explicit = '9073'; //Value matching "Nouvelle-
Calédonie".
        }
        elseif($pla_geonames_country_code == 'NZ'){
            $pla_geon_country_code_explicit = '9074'; //Value matching "Nouvelle-
Zélande".
        }
        elseif($pla_geonames_country_code == 'NI'){
            $pla_geon_country_code_explicit = '9075'; //Value matching
"Nicaragua".
        }
        elseif($pla_geonames_country_code == 'NE'){
            $pla_geon_country_code_explicit = '9076'; //Value matching "Niger".
        }
        elseif($pla_geonames_country_code == 'NG'){
            $pla_geon_country_code_explicit = '9077'; //Value matching "Nigéria".
        }
        elseif($pla_geonames_country_code == 'NU'){
            $pla_geon_country_code_explicit = '9078'; //Value matching "Niue".
        }
        elseif($pla_geonames_country_code == 'NF'){
            $pla_geon_country_code_explicit = '9079'; //Value matching "Norfolk
(île)".
        }
        elseif($pla_geonames_country_code == 'MK'){
            $pla_geon_country_code_explicit = '9080'; //Value matching "Macédoine
du Nord".
        }
        elseif($pla_geonames_country_code == 'MP'){
```

```
            $pla_geon_country_code_explicit = '9081'; //Value matching "Mariannes
du Nord (îles)".
        }
        elseif($pla_geonames_country_code == 'NO'){
            $pla_geon_country_code_explicit = '9082'; //Value matching "Norvège".
        }
        elseif($pla_geonames_country_code == 'OM'){
            $pla_geon_country_code_explicit = '9083'; //Value matching "Oman".
        }
        elseif($pla_geonames_country_code == 'PK'){
            $pla_geon_country_code_explicit = '9084'; //Value matching "Pakistan".
        }
        elseif($pla_geonames_country_code == 'PW'){
            $pla_geon_country_code_explicit = '9085'; //Value matching "Palaos".
        }
        elseif($pla_geonames_country_code == 'PS'){
            $pla_geon_country_code_explicit = '9086'; //Value matching
"Palestine".
        }
        elseif($pla_geonames_country_code == 'PA'){
            $pla_geon_country_code_explicit = '9087'; //Value matching "Panama".
        }
        elseif($pla_geonames_country_code == 'PG'){
            $pla_geon_country_code_explicit = '9088'; //Value matching "Papouasie-
Nouvelle-Guinée".
        }
        elseif($pla_geonames_country_code == 'PY'){
            $pla_geon_country_code_explicit = '9089'; //Value matching "Paraguay".
        }
        elseif($pla_geonames_country_code == 'PE'){
            $pla_geon_country_code_explicit = '9090'; //Value matching "Pérou".
        }
        elseif($pla_geonames_country_code == 'PH'){
            $pla_geon_country_code_explicit = '9091'; //Value matching
"Philippines".
        }
        elseif($pla_geonames_country_code == 'PN'){
            $pla_geon_country_code_explicit = '9092'; //Value matching "Pitcairn
(îles)".
        }
        elseif($pla_geonames_country_code == 'PL'){
            $pla_geon_country_code_explicit = '9093'; //Value matching "Pologne".
        }
        elseif($pla_geonames_country_code == 'PT'){
            $pla_geon_country_code_explicit = '9094'; //Value matching "Portugal".
        }
        elseif($pla_geonames_country_code == 'PR'){
            $pla_geon_country_code_explicit = '9095'; //Value matching "Porto
Rico".
        }
        elseif($pla_geonames_country_code == 'QA'){
            $pla_geon_country_code_explicit = '9096'; //Value matching "Qatar".
        }
        elseif($pla_geonames_country_code == 'RO'){
            $pla_geon_country_code_explicit = '9097'; //Value matching "Roumanie".
        }
        elseif($pla_geonames_country_code == 'RU'){
            $pla_geon_country_code_explicit = '9098'; //Value matching "Russie".
        }
```

```
        elseif($pla_geonames_country_code == 'RW'){
            $pla_geon_country_code_explicit = '9099'; //Value matching "Rwanda".
        }
        elseif($pla_geonames_country_code == 'RE'){
            $pla_geon_country_code_explicit = '9100'; //Value matching "La
Réunion".
        }
        elseif($pla_geonames_country_code == 'BL'){
            $pla_geon_country_code_explicit = '9101'; //Value matching "Saint-
Barthélemy".
        }
        elseif($pla_geonames_country_code == 'SH'){
            $pla_geon_country_code_explicit = '9102'; //Value matching "Sainte-
Hélène, Ascension et Tristan da Cunha".
        }
        elseif($pla_geonames_country_code == 'KN'){
            $pla_geon_country_code_explicit = '9103'; //Value matching "Saint-
Christophe-et-Niévès".
        }
        elseif($pla_geonames_country_code == 'LC'){
            $pla_geon_country_code_explicit = '9104'; //Value matching "Sainte-
Lucie".
        }
        elseif($pla_geonames_country_code == 'MF'){
            $pla_geon_country_code_explicit = '9105'; //Value matching "Saint-
Martin (partie française)".
        }
        elseif($pla_geonames_country_code == 'PM'){
            $pla_geon_country_code_explicit = '9106'; //Value matching "Saint-
Pierre-et-Miquelon".
        }
        elseif($pla_geonames_country_code == 'VC'){
            $pla_geon_country_code_explicit = '9107'; //Value matching "Saint-
Vincent-et-les Grenadines".
        }
        elseif($pla_geonames_country_code == 'WS'){
            $pla_geon_country_code_explicit = '9108'; //Value matching "Samoa".
        }
        elseif($pla_geonames_country_code == 'SM'){
            $pla_geon_country_code_explicit = '9109'; //Value matching "Saint-
Marin".
        }
        elseif($pla_geonames_country_code == 'ST'){
            $pla_geon_country_code_explicit = '9110'; //Value matching "Sao Tomé-
et-Principe".
        }
        elseif($pla_geonames_country_code == 'SA'){
            $pla_geon_country_code_explicit = '9111'; //Value matching "Arabie
saoudite".
        }
        elseif($pla_geonames_country_code == 'SN'){
            $pla_geon_country_code_explicit = '9112'; //Value matching "Sénégal".
        }
        elseif($pla_geonames_country_code == 'RS'){
            $pla_geon_country_code_explicit = '9113'; //Value matching "Serbie".
        }
        elseif($pla_geonames_country_code == 'SC'){
            $pla_geon_country_code_explicit = '9114'; //Value matching
"Seychelles".
```

```
        }
        elseif($pla_geonames_country_code == 'SL'){
            $pla_geon_country_code_explicit = '9115'; //Value matching "Sierra
Leone".
        }
        elseif($pla_geonames_country_code == 'SG'){
            $pla_geon_country_code_explicit = '9116'; //Value matching
"Singapour".
        }
        elseif($pla_geonames_country_code == 'SX'){
            $pla_geon_country_code_explicit = '9117'; //Value matching "Saint-
Martin (partie néerlandaise)".
        }
        elseif($pla_geonames_country_code == 'SK'){
            $pla_geon_country_code_explicit = '9118'; //Value matching
"Slovaquie".
        }
        elseif($pla_geonames_country_code == 'SI'){
            $pla_geon_country_code_explicit = '9119'; //Value matching "Slovénie".
        }
        elseif($pla_geonames_country_code == 'SB'){
            $pla_geon_country_code_explicit = '9120'; //Value matching "Salomon
(îles)".
        }
        elseif($pla_geonames_country_code == 'SO'){
            $pla_geon_country_code_explicit = '9121'; //Value matching "Somalie".
        }
        elseif($pla_geonames_country_code == 'ZA'){
            $pla_geon_country_code_explicit = '9122'; //Value matching "Afrique du
Sud".
        }
        elseif($pla_geonames_country_code == 'GS'){
            $pla_geon_country_code_explicit = '9123'; //Value matching "Géorgie du
Sud-et-les îles Sandwich du Sud".
        }
        elseif($pla_geonames_country_code == 'SS'){
            $pla_geon_country_code_explicit = '9124'; //Value matching "Soudan du
Sud".
        }
        elseif($pla_geonames_country_code == 'ES'){
            $pla_geon_country_code_explicit = '9125'; //Value matching "Espagne".
        }
        elseif($pla_geonames_country_code == 'LK'){
            $pla_geon_country_code_explicit = '9126'; //Value matching "Sri
Lanka".
        }
        elseif($pla_geonames_country_code == 'SD'){
            $pla_geon_country_code_explicit = '9127'; //Value matching "Soudan".
        }
        elseif($pla_geonames_country_code == 'SR'){
            $pla_geon_country_code_explicit = '9128'; //Value matching "Suriname".
        }
        elseif($pla_geonames_country_code == 'SJ'){
            $pla_geon_country_code_explicit = '9129'; //Value matching "Svalbard
et Jan Mayen".
        }
        elseif($pla_geonames_country_code == 'SE'){
            $pla_geon_country_code_explicit = '9130'; //Value matching "Suède".
        }
```

```
        elseif($pla_geonames_country_code == 'CH'){
            $pla_geon_country_code_explicit = '9131'; //Value matching "Suisse".
        }
        elseif($pla_geonames_country_code == 'SY'){
            $pla_geon_country_code_explicit = '9132'; //Value matching "Syrie".
        }
        elseif($pla_geonames_country_code == 'TW'){
            $pla_geon_country_code_explicit = '9133'; //Value matching "Taïwan".
        }
        elseif($pla_geonames_country_code == 'TJ'){
            $pla_geon_country_code_explicit = '9134'; //Value matching
"Tadjikistan".
        }
        elseif($pla_geonames_country_code == 'TZ'){
            $pla_geon_country_code_explicit = '9135'; //Value matching "Tanzanie".
        }
        elseif($pla_geonames_country_code == 'TH'){
            $pla_geon_country_code_explicit = '9136'; //Value matching
"Thaïlande".
        }
        elseif($pla_geonames_country_code == 'TL'){
            $pla_geon_country_code_explicit = '9137'; //Value matching "Timor
oriental".
        }
        elseif($pla_geonames_country_code == 'TG'){
            $pla_geon_country_code_explicit = '9138'; //Value matching "Togo".
        }
        elseif($pla_geonames_country_code == 'TK'){
            $pla_geon_country_code_explicit = '9139'; //Value matching "Tokelau".
        }
        elseif($pla_geonames_country_code == 'TO'){
            $pla_geon_country_code_explicit = '9140'; //Value matching "Tonga".
        }
        elseif($pla_geonames_country_code == 'TT'){
            $pla_geon_country_code_explicit = '9141'; //Value matching "Trinité-
et-Tobago".
        }
        elseif($pla_geonames_country_code == 'TN'){
            $pla_geon_country_code_explicit = '9142'; //Value matching "Tunisie".
        }
        elseif($pla_geonames_country_code == 'TM'){
            $pla_geon_country_code_explicit = '9143'; //Value matching
"Turkménistan".
        }
        elseif($pla_geonames_country_code == 'TC'){
            $pla_geon_country_code_explicit = '9144'; //Value matching "Turques-
et-Caïques (îles)".
        }
        elseif($pla_geonames_country_code == 'TV'){
            $pla_geon_country_code_explicit = '9145'; //Value matching "Tuvalu".
        }
        elseif($pla_geonames_country_code == 'TR'){
            $pla_geon_country_code_explicit = '9146'; //Value matching "Turquie".
        }
        elseif($pla_geonames_country_code == 'UG'){
            $pla_geon_country_code_explicit = '9147'; //Value matching "Ouganda".
        }
        elseif($pla_geonames_country_code == 'UA'){
            $pla_geon_country_code_explicit = '9148'; //Value matching "Ukraine".
```

```
        }
        elseif($pla_geonames_country_code == 'AE'){
            $pla_geon_country_code_explicit = '9149'; //Value matching "Émirats
arabes unis".
        }
        elseif($pla_geonames_country_code == 'GB'){
            $pla_geon_country_code_explicit = '9150'; //Value matching "Royaume-
Uni".
        }
        elseif($pla_geonames_country_code == 'UM'){
            $pla_geon_country_code_explicit = '9151'; //Value matching "Îles
mineures éloignées des États-Unis".
        }
        elseif($pla_geonames_country_code == 'US'){
            $pla_geon_country_code_explicit = '9152'; //Value matching "États-Unis
d'Amérique".
        }
        elseif($pla_geonames_country_code == 'UY'){
            $pla_geon_country_code_explicit = '9153'; //Value matching "Uruguay".
        }
        elseif($pla_geonames_country_code == 'UZ'){
            $pla_geon_country_code_explicit = '9154'; //Value matching
"Ouzbékistan".
        }
        elseif($pla_geonames_country_code == 'VU'){
            $pla_geon_country_code_explicit = '9155'; //Value matching "Vanuatu".
        }
        elseif($pla_geonames_country_code == 'VE'){
            $pla_geon_country_code_explicit = '9156'; //Value matching
"Venezuela".
        }
        elseif($pla_geonames_country_code == 'VN'){
            $pla_geon_country_code_explicit = '9157'; //Value matching "Viet Nam".
        }
        elseif($pla_geonames_country_code == 'VG'){
            $pla_geon_country_code_explicit = '9158'; //Value matching "Vierges
britanniques (îles)".
        }
        elseif($pla_geonames_country_code == 'VI'){
            $pla_geon_country_code_explicit = '9159'; //Value matching "Vierges
des États-Unis (îles)".
        }
        elseif($pla_geonames_country_code == 'WF'){
            $pla_geon_country_code_explicit = '9160'; //Value matching "Wallis-et-
Futuna".
        }
        elseif($pla_geonames_country_code == 'EH'){
            $pla_geon_country_code_explicit = '9161'; //Value matching "Sahara
occidental".
        }
        elseif($pla_geonames_country_code == 'YE'){
            $pla_geon_country_code_explicit = '9162'; //Value matching "Yémen".
        }
        elseif($pla_geonames_country_code == 'ZM'){
            $pla_geon_country_code_explicit = '9163'; //Value matching "Zambie".
        }
        elseif($pla_geonames_country_code == 'ZW'){
            $pla_geon_country_code_explicit = '9164'; //Value matching "Zimbabwe".
        }
```

```
        elseif($pla_geonames_country_code == 'AX'){
            $pla_geon_country_code_explicit = '9165'; //Value matching "Åland".
        }
        elseif($pla_geonames_country_code == 'XK'){
            $pla_geon_country_code_explicit = '9166'; //Value matching "Kosovo".
        }
        else{
            $pla_geon_country_code_explicit = '';
        }

//      Create field pla_geonames_cc2 from cc2.

        $pla_geonames_cc2 = $cc2;

        if(stripos($pla_geonames_cc2, ',') !== false || stripos($pla_geonames_cc2,
'"') !== false){
            $pla_geonames_cc2 = '"'.$pla_geonames_cc2.'"';
        }

//      Create field pla_geonames_admin1_code from admin1_code.

        $pla_geonames_admin1_code = $admin1_code;

        if(stripos($pla_geonames_admin1_code, ',') !== false ||
stripos($pla_geonames_admin1_code, '"') !== false){
            $pla_geonames_admin1_code = '"'.$pla_geonames_admin1_code.'"';
        }

//      Create field pla_geonames_admin2_code from admin2_code.

        $pla_geonames_admin2_code = $admin2_code;

        if(stripos($pla_geonames_admin2_code, ',') !== false ||
stripos($pla_geonames_admin2_code, '"') !== false){
            $pla_geonames_admin2_code = '"'.$pla_geonames_admin2_code.'"';
        }

//      Create field pla_geonames_admin3_code from admin3_code.

        $pla_geonames_admin3_code = $admin3_code;

        if(stripos($pla_geonames_admin3_code, ',') !== false ||
stripos($pla_geonames_admin3_code, '"') !== false){
            $pla_geonames_admin3_code = '"'.$pla_geonames_admin3_code.'"';
        }

//      Create field pla_geonames_admin4_code from admin4_code.

        $pla_geonames_admin4_code = $admin4_code;

        if(stripos($pla_geonames_admin4_code, ',') !== false ||
stripos($pla_geonames_admin4_code, '"') !== false){
            $pla_geonames_admin4_code = '"'.$pla_geonames_admin4_code.'"';
        }

//      Create field pla_geonames_population from population.

        $pla_geonames_population = $population;
```

```
        if($pla_geonames_population == 0){
            $pla_geonames_population = NULL;
        }

//      Create field pla_geonames_elevation from elevation.

        $pla_geonames_elevation = $elevation;

        if(stripos($pla_geonames_elevation, ',') !== false ||
stripos($pla_geonames_elevation, '"') !== false){
            $pla_geonames_elevation = '"'.$pla_geonames_elevation.'"';
        }

//      Create field pla_geonames_dem from dem.

        $pla_geonames_dem = $dem;

        if($pla_geonames_dem == 0){
            $pla_geonames_dem = NULL;
        }

//      Create field pla_geonames_timezone from timezone.

        $pla_geonames_timezone = $timezone;

        if(stripos($pla_geonames_timezone, ',') !== false ||
stripos($pla_geonames_timezone, '"') !== false){
            $pla_geonames_timezone = '"'.$pla_geonames_timezone.'"';
        }

//      Create field pla_geonames_modification_date from modification_date.

        $pla_geonames_modification_date = $modification_date;

//      Create field pla_is_part_of_country_pla. To create this field, we first
need to import all the political entities into the Drupal database (using the
"conditions" below). Next, we need to generate an array linking the ISO code of
the country (pla_geonames_country code) with the identifier (NID) of the entity in
the database (see documentation). This array is stored in a separate PHP file.

        include 'pla_geonames_country_code_array.php';

        $feature_codes_to_exclude_1 = array('PCL', 'PCLD', 'PCLF', 'PCLI',
'PCLS');

        foreach($feature_codes_to_exclude_1 as $feature_code_value_1){
            if($feature_code === $feature_code_value_1){
                $feature_code_check_1 = true;
                break;
            }
        }

        if(!$feature_code_check_1){
            if(array_key_exists($country_code, $pla_geonames_country_code_array)){
                $pla_is_part_of_country_pla =
$pla_geonames_country_code_array[$country_code];
            }
            else{
                $pla_is_part_of_country_pla = '';
```

```
            }
        }
        else{
            $pla_is_part_of_country_pla = '';
        }

//      Create field pla_is_part_of_admin1_pla. To create this field, we first
need to import all the first-order administrative division into the Drupal
database (using the "conditions" below). Next, we need to generate an array
linking the GeoNames ID of the first-order administrative division with the
identifier (NID) of the corresponding entity in the database (see documentation).
This array is stored in a separate PHP file.

        include 'pla_geonames_admin1_array.php';

        $feature_codes_to_exclude_2 = array('PCL', 'PCLD', 'PCLF', 'PCLH', 'PCLI',
'PCLIX', 'PCLS', 'ADM1', 'ADM1H');

        foreach($feature_codes_to_exclude_2 as $feature_code_value_2){
            if($feature_code === $feature_code_value_2){
                $feature_code_check_2 = true;
                break;
            }
        }

        if(!$feature_code_check_2){
            $combine_country_and_admin1_codes = $country_code.'.'.$admin1_code;
        }
        else{
            $combine_country_and_admin1_codes = NULL;
        }

//      We now include a PHP file created from the file "admin1CodesASCII.txt"
that we downloaded from GeoNames website (see documentation).

        include 'pla_geonames_admin1_codes.php';

        if(array_key_exists($geonameid_check, $pla_geonames_admin1_array)){
            $pla_is_part_of_admin1_pla =
$pla_geonames_admin1_array[$geonameid_check];
        }
        else{
            $pla_is_part_of_admin1_pla = '';
        }

//      Create field pla_is_part_of_admin2_pla. To create this field, we first
need to import all the second-order administrative division into the Drupal
database (using the "conditions" below). Next, we need to generate an array
linking the GeoNames ID of the second-order administrative division with the
identifier (NID) of the corresponding entity in the database (see documentation).
This array is stored in a separate PHP file.

        include 'pla_geonames_admin2_array.php';

        $feature_codes_to_exclude_3 = array('PCL', 'PCLD', 'PCLF', 'PCLH', 'PCLI',
'PCLIX', 'PCLS', 'ADM1', 'ADM1H', 'ADM2', 'ADM2H');

        foreach($feature_codes_to_exclude_3 as $feature_code_value_3){
            if($feature_code === $feature_code_value_3){
```

```
                $feature_code_check_3 = true;
                break;
            }
        }

        if(!$feature_code_check_3){
            $combine_country_admin1_and_admin2_codes =
$country_code.'.'.$admin1_code.'.'.$admin2_code;
        }
        else{
            $combine_country_admin1_and_admin2_codes = NULL;
        }

//      We now include a PHP file created from the file "admin2Codes.txt" that we
downloaded from GeoNames website (see documentation).

        include 'pla_geonames_admin2_codes.php';

        if(array_key_exists($geonameid_check_2, $pla_geonames_admin2_array)){
            $pla_is_part_of_admin2_pla =
$pla_geonames_admin2_array[$geonameid_check_2];
        }
        else{
            $pla_is_part_of_admin2_pla = '';
        }

//      Create field all_validity.

        $all_validity = '84'; //Value matching "Vérifié" in taxonomy "Validité".

//      Conditions for displaying or excluding the places from a specific country.
Use the ISO code of the country, such as FR for France. Leave empty to display the
places from all countries.

        $country_to_display = '';
        $country_to_exclude = '';

//      Conditions for displaying only certain types of places (yes/no).

//      Show all (regardless of other conditions).

        $condition_show_all = 'no';

//      Political entities (class A).

        $condition_political_entity = 'yes';
        $condition_dependant_political_entity = 'yes';
        $condition_freely_associated_state = 'yes';
        $condition_historical_political_entity = 'yes';
        $condition_independant_political_entity = 'yes';
        $condition_section_of_independent_political_entity = 'yes';
        $condition_semi_independant_political_entity = 'yes';

//      Administrative divisions (class A).

        $condition_first_order_administrative_division = 'yes';
        $condition_historical_first_order_administrative_division = 'yes';
        $condition_second_order_administrative_division = 'yes';
        $condition_historical_second_order_administrative_division = 'yes';
```

```
        $condition_third_order_administrative_division = 'yes';
        $condition_historical_third_order_administrative_division = 'yes';
        $condition_fourth_order_administrative_division = 'yes';
        $condition_historical_fourth_order_administrative_division = 'yes';
        $condition_fifth_order_administrative_division = 'yes';
        $condition_historical_fifth_order_administrative_division = 'yes';
        $condition_administrative_division = 'yes';
        $condition_historical_administrative_division = 'yes';

//      Other country, state, region (class A).

        $condition_leased_area = 'yes';
        $condition_parish = 'yes';
        $condition_territory = 'yes';
        $condition_zone = 'yes';
        $condition_buffer_zone = 'yes';

//      Stream, lake, ... (class H).

        $condition_stream_lake = 'yes';

//      Parks, area, ... (class L).

        $condition_parks_area = 'yes';

//      City, village, ... (class P).

        $condition_city_village = 'yes';

//      Road, railroad, ... (class R).

        $condition_road_railroad = 'yes';

//      Spot, building, farm (class S).

        $condition_spot_building_farm = 'yes';

//      Mountain, hill, rock, ... (class T).

        $condition_mountain_hill_rock = 'yes';

//      Undersea (class U).

        $condition_undersea = 'yes';

//      Condition forest, heath, ... (class V).

        $condition_forest_heath = 'yes';

//      Check the conditions.

        if($condition_show_all == 'yes' || ($condition_political_entity == 'yes'
&& $feature_code == 'PCL') || ($condition_dependant_political_entity == 'yes' &&
$feature_code == 'PCLD') || ($condition_freely_associated_state == 'yes' &&
$feature_code == 'PCLF') || ($condition_historical_political_entity == 'yes' &&
$feature_code == 'PCLH') || ($condition_independant_political_entity == 'yes' &&
$feature_code == 'PCLI') || ($condition_section_of_independent_political_entity ==
'yes' && $feature_code == 'PCLIX') ||
($condition_semi_independant_political_entity == 'yes' && $feature_code == 'PCLS')
```

```
|| ($condition_first_order_administrative_division == 'yes' && $feature_code ==
'ADM1') || ($condition_historical_first_order_administrative_division == 'yes' &&
$feature_code == 'ADM1H') || ($condition_second_order_administrative_division ==
'yes' && $feature_code == 'ADM2') ||
($condition_historical_second_order_administrative_division == 'yes' &&
$feature_code == 'ADM2H') || ($condition_third_order_administrative_division ==
'yes' && $feature_code == 'ADM3') ||
($condition_historical_third_order_administrative_division == 'yes' &&
$feature_code == 'ADM3H') || ($condition_fourth_order_administrative_division ==
'yes' && $feature_code == 'ADM4') ||
($condition_historical_fourth_order_administrative_division == 'yes' &&
$feature_code == 'ADM4H') || ($condition_fifth_order_administrative_division ==
'yes' && $feature_code == 'ADM5') ||
($condition_historical_fifth_order_administrative_division == 'yes' &&
$feature_code == 'ADM5H') || ($condition_administrative_division == 'yes' &&
$feature_code == 'ADMD') || ($condition_historical_administrative_division ==
'yes' && $feature_code == 'ADMDH') || ($condition_leased_area == 'yes' &&
$feature_code == 'LTER') || ($condition_parish == 'yes' && $feature_code ==
'PRSH') || ($condition_territory == 'yes' && $feature_code == 'TERR') ||
($condition_zone == 'yes' && $feature_code == 'ZN') || ($condition_buffer_zone ==
'yes' && $feature_code == 'ZNB') || ($condition_stream_lake == 'yes' &&
$feature_class == 'H') || ($condition_parks_area == 'yes' && $feature_class ==
'L') || ($condition_city_village == 'yes' && $feature_class == 'P') ||
($condition_road_railroad == 'yes' && $feature_class == 'R') ||
($condition_spot_building_farm == 'yes' && $feature_class == 'S') ||
($condition_mountain_hill_rock == 'yes' && $feature_class == 'T') ||
($condition_undersea == 'yes' && $feature_class == 'U') ||
($condition_forest_heath == 'yes' && $feature_class == 'V')){
            $data_display = 'verified';
        }

        if($country_to_display == '' && $country_to_exclude == ''){
            $country_display = 'verified';
        }
        elseif($country_to_display == '' && $country_to_exclude == $country_code){
            $country_display = '';
        }
        elseif($country_to_display == '' && $country_to_exclude != $country_code){
            $country_display = 'verified';
        }
        elseif($country_to_display == $country_code && $country_to_exclude == ''){
            $country_display = 'verified';
        }
        elseif($country_to_display != $country_code && $country_to_exclude == ''){
            $country_display = '';
        }
        else{
            $country_display = '';
            $show_error_message = 'verifie';
        }

//      Combine the fields in CSV format.
        if($data_display == 'verified' && $country_display == 'verified'){

            $output .=
$pla_geonames_geonameid.','.$pla_geonames_record.','.$pla_geonames_name.','.$pla_g
eonames_asciiname.','.$pla_geonames_alternatenames.','.$pla_geonames_latitude.','.
$pla_geonames_longitude.','.$pla_geonames_feature_class.','.$pla_geonames_feature_
code.','.$pla_geon_feature_code_explicit.','.$pla_geonames_country_code.','.$pla_g
```

```
eon_country_code_explicit.','.$pla_geonames_cc2.','.$pla_geonames_admin1_code.','.
$pla_geonames_admin2_code.','.$pla_geonames_admin3_code.','.$pla_geonames_admin4_c
ode.','.$pla_geonames_population.','.$pla_geonames_elevation.','.$pla_geonames_dem
.','.$pla_geonames_timezone.','.$pla_geonames_modification_date.','.$pla_is_part_o
f_country_pla.','.$pla_is_part_of_admin1_pla.','.$pla_is_part_of_admin2_pla.','.$a
ll_validity.'<br/>';
            $total_displayed ++;

        }
    }
}

//Error message

if($show_error_message == 'verifie'){
    echo 'Error in the conditions: Choose a country to display or a country to
exclude, but not both options.';
}
else{

//  Echo the number of results.

    echo 'Nombre de résultats : '.$total_displayed.'<br/>';
    echo 'Affichage des résultats '.$start_row.' à '.$end_row.'<br/><br/>';

//  Output in CSV format.

    echo
'pla_geonames_geonameid,pla_geonames_record,pla_geonames_name,pla_geonames_asciina
me,pla_geonames_alternatenames,pla_geonames_latitude,pla_geonames_longitude,pla_ge
onames_feature_class,pla_geonames_feature_code,pla_geon_feature_code_explicit,pla_
geonames_country_code,pla_geon_country_code_explicit,pla_geonames_cc2,pla_geonames
_admin1_code,pla_geonames_admin2_code,pla_geonames_admin3_code,pla_geonames_admin4
_code,pla_geonames_population,pla_geonames_elevation,pla_geonames_dem,pla_geonames
_timezone,pla_geonames_modification_date,pla_is_part_of_country_pla,pla_is_part_of
_admin1_pla,pla_is_part_of_admin2_pla,all_validity<br/>';
    echo $output;

}

?>
```

b. Dans un premier temps, nous importons uniquement les lieux qui forment des pays ou des entités politiques assimilables à des pays. Dans cette perspective, nous utilisons le code du fichier « csv_geonames_place.php » ci-dessus qui permet d'afficher les données en format CSV. Vers la fin du fichier PHP, il convient de désactiver par trois blocs de commentaires successifs chacune des trois sections de code consacrées à la création des champs pla_is_part_of_country_pla, pla_is_part_of_admin1_pla et pla_is_part_of_admin2_pla. Toujours à la fin du fichier, nous paramétrons les « conditions » de sorte que seules les entités politiques soient prises en considération.
c. Copier les données depuis un navigateur web et coller ces données dans un nouveau fichier CSV vide.
d. Importer les données dans *Drupal*, grâce à l'outil d'importation du module Webform.

e. Il s'agit maintenant de créer un tableau associatif relatif aux entités politiques qui viennent d'être importées. Ce tableau doit réunir le code ISO du pays (pla_geonames_country_code) et l'identifiant (NID) de la fiche de lieu qui lui correspond dans la base de données du site *Drupal*. Nous envoyons une requête SQL à la table « node__pla_geonames_country_code » de la base « mercure » et nous générons le tableau grâce au PHP. Exemple de code (fichier « pla_geonames_country_code_id.php ») :

```php
<?php
//Array combining the ISO code of a country and the node ID (NID) of this country.

//Determine the first and last row to be echoed. For $end_row, the value 0 will
echo all the rows.

$start_row = 0;
$end_row = 0;

//Connect to the database.

$mysqli = new MySQLi('localhost','user','password','database');

if(mysqli_connect_errno())
    {
        printf('Échec de la connexion : base de données momentanément
inaccessible.');
        exit();
    }

//Total number of items.

$result_set_1 = $mysqli->query('SELECT COUNT(*) FROM
node__pla_geonames_country_code');
$total = $result_set_1->fetch_row();

if($end_row == 0){
    $end_row = $total[0];
}

//Query the database.

$result_set_2 = $mysqli->query('SELECT entity_id, pla_geonames_country_code_value
FROM node__pla_geonames_country_code');


//Retrieve and manage data.

$tmpCount = 0;
$total_displayed = 0;
$output = NULL;

if($result_set_2->num_rows > 0){
    while(null !==($rows = $result_set_2->fetch_assoc()))
    {

//      Count the lines and retrieve the two columns from the database.

        $tmpCount ++;
        $entity_id = $rows['entity_id'];
```

```
        $pla_geonames_country_code_value =
$rows['pla_geonames_country_code_value'];

//      Combine the fields.

        if(($tmpCount >= $start_row) && ($tmpCount <= $end_row)){
            $output .= '\''.$pla_geonames_country_code_value.'\' =>
'.$entity_id.', <br/>';
            $total_displayed ++;
        }
    }
}

//Echo the number of results.

echo 'Nombre de résultats : '.$total_displayed.'<br/>';
echo 'Affichage des résultats '.$start_row.' à '.$end_row.'<br/><br/>';

//Output.

echo '$pla_geonames_country_code_array = array(<br/>'.$output.');';
?>
```

f. Le résultat obtenu dans un navigateur web peut ensuite être copié et collé dans un fichier vierge « pla_geonames_country_code_array.php », fichier qui doit être placé dans le même répertoire que « csv_geonames_place.php ». Exemple de code (extrait) :

```
<?php
$pla_geonames_country_code_array = array(
'AD' => 892042,
'AE' => 892043,
'AF' => 892044,
'AG' => 892045,
/* [...] */
'ZM' => 892303,
'ZW' => 892304,
);
?>
```

g. Nous réitérons l'étape 3.b. ci-dessus. Cependant, nous paramétrons cette fois les « conditions » pour afficher les divisions administratives de niveau 1 uniquement. Il faut par ailleurs permettre la création du champ pla_is_part_of_country_pla en désactivant le bloc de commentaires autour du code relatif à ce champ. Nous importons les lieux concernés dans la base de données *Drupal*.

h. Dans la perspective d'importer les division administrative de niveau 2 et de les lier aux divisions administratives de niveau 1, nous créons un fichier PHP « pla_geonames_admin1_codes.php » à partir du fichier « admin1CodesASCII.txt » téléchargé sur le site de *GeoNames*. Ce fichier PHP permet d'obtenir l'identifiant *GeoNames* (geonameid) à partir de deux données de *GeoNames* : le code du pays (country_code) et le code de la division administrative de niveau 1 (admin1_code). Ce fichier est déposé dans le même répertoire que « csv_geonames_place.php ». Exemple de code (extrait) :

```php
<?php
if($combine_country_and_admin1_codes == 'AD.06'){
    $geonameid_check = 3039162; // GeoNames ID of Sant Julià de Loria / Sant Julia
de Loria.
}
elseif($combine_country_and_admin1_codes == 'AD.05'){
    $geonameid_check = 3039676; // GeoNames ID of Ordino / Ordino.
}
elseif($combine_country_and_admin1_codes == 'AD.04'){
    $geonameid_check = 3040131; // GeoNames ID of La Massana / La Massana.
}
elseif($combine_country_and_admin1_codes == 'AD.03'){
    $geonameid_check = 3040684; // GeoNames ID of Encamp / Encamp.
}
/* [...] */
elseif($combine_country_and_admin1_codes == 'ZW.09'){
    $geonameid_check = 1105843; // GeoNames ID of Bulawayo / Bulawayo.
}
elseif($combine_country_and_admin1_codes == 'ZW.10'){
    $geonameid_check = 1105844; // GeoNames ID of Harare / Harare.
}
else{
    $geonameid_check = NULL;
}
?>
```

i. Comme nous l'avons fait pour les entités politiques, nous créons à l'aide du PHP un tableau associatif qui réunit l'identifiant *Genonames* de la division administrative de niveau 1 (pla_geonames_geonameid) et l'identifiant *Drupal* (NID) correspondant. Pour créer ce tableau, il convient d'interroger la table « node__pla_geonames_geonameid ». Le code auquel nous recourons ci-dessous inclut aussi les pays, dont nous n'avons pas besoin à ce stade du travail, mais cela ne porte pas à conséquence. Exemple de code (fichier « pla_geonames_admin1_id.php ») :

```php
<?php
//Array combining the GeoNames ID of a first-order administrative division and the
node ID (NID) of this place.

//Determine the first and last row to be echoed. For $end_row, the value 0 will
echo all the rows.

$start_row = 0;
$end_row = 0;

//Connect to the database.

$mysqli = new MySQLi('localhost','user','password','database');

if(mysqli_connect_errno())
    {
        printf('Échec de la connexion : base de données momentanément
inaccessible.');
        exit();
    }

//Total number of items.
```

```php
$result_set_1 = $mysqli->query('SELECT COUNT(*) FROM
node__pla_geonames_geonameid');
$total = $result_set_1->fetch_row();

if($end_row == 0){
    $end_row = $total[0];
}

//Query the database.

$result_set_2 = $mysqli->query('SELECT entity_id, pla_geonames_geonameid_value
FROM node__pla_geonames_geonameid');


//Retrieve and manage data.

$tmpCount = 0;
$total_displayed = 0;
$output = NULL;

if($result_set_2->num_rows > 0){
    while(null !==($rows = $result_set_2->fetch_assoc()))
    {

//      Count the lines and retrieve the two columns from the database.

        $tmpCount ++;
        $entity_id = $rows['entity_id'];
        $pla_geonames_geonameid_value = $rows['pla_geonames_geonameid_value'];

//      Combine the fields.

        if(($tmpCount >= $start_row) && ($tmpCount <= $end_row)){
            $output .= $pla_geonames_geonameid_value.' => '.$entity_id.', <br/>';
            $total_displayed ++;
        }
    }
}

//Echo the number of results.

echo 'Nombre de résultats : '.$total_displayed.'<br/>';
echo 'Affichage des résultats '.$start_row.' à '.$end_row.'<br/><br/>';

//Output.

echo '$pla_geonames_admin1_array = array(<br/>'.$output.');';
?>
```

j. Le résultat obtenu dans un navigateur web peut ensuite être copié et collé dans un fichier vierge « pla_geonames_admin1_array.php », fichier qui doit être placé dans le même répertoire que « csv_geonames_place.php ». Exemple de code (extrait) :

```php
<?php
$pla_geonames_admin1_array = array(
3041565 => 892042,
290557 => 892043,
```

```
1149361 => 892044,
3576396 => 892045,
/* [...] */
8378500 => 912359,
12489745 => 912360,
);
?>
```

k. Nous pouvons maintenant importer dans la base *Drupal* les divisions administratives de niveau 2, comme nous l'avons fait pour les divisions administratives de niveau 1, en veillant à définir les « conditions » adéquates dans le fichier « csv_geonames_place.php » pour qu'il ne traite que les divisions administratives de niveau 2. Il faut par ailleurs permettre la création du champ pla_is_part_of_admin1_pla en désactivant le bloc de commentaires autour du code relatif à ce champ.

l. Dans la perspective d'importer tout le reste des données de *GeoNames* et de lier chaque lieu à sa division administrative de niveau 2 (lorsque ce lien existe), nous créons un fichier PHP « pla_geonames_admin2_codes.php » à partir du fichier « admin2Codes.txt » téléchargé sur le site de *GeoNames*. Ce fichier PHP permet d'obtenir l'identifiant *GeoNames* (geonameid) à partir de trois données de *GeoNames* : le code du pays (country_code), le code de la division administrative de niveau 1 (admin1_code) et le code de la division administrative de niveau 2 (admin2_code). Ce fichier est déposé dans le même répertoire que « csv_geonames_place.php ». Exemple de code (extrait) :

```
<?php
if($combine_country_admin1_and_admin2_codes == 'AE.01.101'){
    $geonameid_check_2 = 12047239; // GeoNames ID of Abu Dhabi Municipality / Abu
Dhabi Municipality.
}
elseif($combine_country_admin1_and_admin2_codes == 'AE.01.102'){
    $geonameid_check_2 = 12047240; // GeoNames ID of Al Ain Municipality / Al Ain
Municipality.
}
elseif($combine_country_admin1_and_admin2_codes == 'AE.01.103'){
    $geonameid_check_2 = 12047241; // GeoNames ID of Al Dhafra / Al Dhafra.
}
elseif($combine_country_admin1_and_admin2_codes == 'AE.04.701'){
    $geonameid_check_2 = 12047242; // GeoNames ID of Al Fujairah Municipality / Al
Fujairah Municipality.
}
/* [...] */
elseif($combine_country_admin1_and_admin2_codes == 'ZW.06.10800412'){
    $geonameid_check_2 = 10800412; // GeoNames ID of Umguza District / Umguza
District.
}
elseif($combine_country_admin1_and_admin2_codes == 'ZW.01.11204515'){
    $geonameid_check_2 = 11204515; // GeoNames ID of Mossurize District /
Mossurize District.
}
else{
    $geonameid_check_2 = NULL;
}
?>
```

m. Comme nous l'avons fait précédemment, il s'agit maintenant de créer un tableau associatif qui réunisse l'identifiant *GeoNames* de la division administrative de niveau 2 (pla_geonames_geonameid) et l'identifiant *Drupal* (NID) correspondant. Pour créer ce tableau, il convient d'interroger une nouvelle fois la table « node__pla_geonames_geonameid ». Le code que nous employons est presque identique à celui que nous avons utilisé à propos des divisions administratives de niveau 1. Le tableau associatif va donc inclure des lieux dont nous n'avons pas besoin à ce stade du travail, mais cela ne porte pas à conséquence. Exemple de code (fichier « pla_geonames_admin2_id.php ») :

```php
<?php
//Array combining the GeoNames ID of a second-order administrative division and
the node ID (NID) of this place.

//Determine the first and last row to be echoed. For $end_row, the value 0 will
echo all the rows.

$start_row = 0;
$end_row = 0;

//Connect to the database.

$mysqli = new MySQLi('localhost','user','password','database');

if(mysqli_connect_errno())
    {
        printf('Échec de la connexion : base de données momentanément
inaccessible.');
        exit();
    }

//Total number of items.

$result_set_1 = $mysqli->query('SELECT COUNT(*) FROM
node__pla_geonames_geonameid');
$total = $result_set_1->fetch_row();

if($end_row == 0){
    $end_row = $total[0];
}

//Query the database.

$result_set_2 = $mysqli->query('SELECT entity_id, pla_geonames_geonameid_value
FROM node__pla_geonames_geonameid');


//Retrieve and manage data.

$tmpCount = 0;
$total_displayed = 0;
$output = NULL;

if($result_set_2->num_rows > 0){
    while(null !==($rows = $result_set_2->fetch_assoc()))
    {

//      Count the lines and retrieve the two columns from the database.
```

```
        $tmpCount ++;
        $entity_id = $rows['entity_id'];
        $pla_geonames_geonameid_value = $rows['pla_geonames_geonameid_value'];

//      Combine the fields.

        if(($tmpCount >= $start_row) && ($tmpCount <= $end_row)){
            $output .= $pla_geonames_geonameid_value.' => '.$entity_id.', <br/>';
            $total_displayed ++;
        }
    }
}

//Echo the number of results.

echo 'Nombre de résultats : '.$total_displayed.'<br/>';
echo 'Affichage des résultats '.$start_row.' à '.$end_row.'<br/><br/>';

//Output.

echo '$pla_geonames_admin2_array = array(<br/>'.$output.');';
?>
```

n. Le résultat obtenu dans un navigateur web peut ensuite être copié et collé dans un fichier vierge « pla_geonames_admin2_array.php », fichier qui doit être placé dans le même répertoire que « csv_geonames_place.php ». Exemple de code (extrait) :

```
<?php
$pla_geonames_admin2_array = array(
3041565 => 892042,
290557 => 892043,
1149361 => 892044,
3576396 => 892045,
/* [...] */
10800412 => 974330,
11204515 => 974331,
);
?>
```

o. En recourant au fichier « csv_geonames_place.php », nous sommes désormais en mesure de modifier toutes les autres données en langage PHP, en veillant à paramétrer la section des « conditions » pour exclure les entités politiques, les divisions administratives de niveau 1 et les divisions administratives de niveau 2 qui ont déjà été importées. Il faut par ailleurs permettre la création du champ pla_is_part_of_admin2_pla en désactivant le bloc de commentaires autour du code relatif à ce champ. Au besoin, le code de « csv_geonames_place.php » permet d'afficher uniquement les données d'un seul pays ou, au contraire, d'exclure un seul pays. Voir en fin de fichier la section des « conditions ».

p. Copier les données depuis un navigateur web et coller ces données dans de nouveaux fichiers CSV vides (100 000 lignes par fichier).

q. Importer les données dans *Drupal* par tranches de 100 000 lignes, grâce à l'outil d'importation du module Webform.

Copies numériques

FICHIERS ET PROPRIÉTÉS

Objectifs

Il s'agit d'importer des numérisations en format PDF des volumes du *Mercure de France*. Ces fichiers PDF ont été préalablement téléchargés depuis des bibliothèques numériques comme *Gallica* et *Google Books*. Nous importons également certaines propriétés des fichiers PDF.

Données concernées

Les données sont les suivantes :
- Le fichier PDF lui-même.
- La taille du fichier (en mégaoctets).
- Le nombre de pages du fichier.
- Le nom du fichier (sans l'extension).

Utilisation

Les copies numériques donnent accès aux contenus du *Mercure de France*. Sur notre site, elles seront liées à des livraisons. Elles s'afficheront en regard des textes, illustrations, etc., saisis dans la base de données.

Nous utilisons des numérisations libres de droit, en indiquant la source du fichier téléchargé, le lien vers la numérisation concernée dans la bibliothèque numérique d'origine et l'année de téléchargement. Certains fichiers PDF ont subi des modifications après leur téléchargement. En particulier, nous lançons une reconnaissance optique de caractères sur les fichiers qui ne disposent pas nativement de cette ressource.

Procédure détaillée

1. Télécharger et nommer les fichiers PDF des livraisons du *Mercure de France* conformément au référentiel « Nom de fichier de la copie numérique ». Les fichiers sont regroupés dans un même dossier.
2. Au moyen de Windows PowerShell (et de son module Poppler), nous extrayons certaines données des fichiers PDF, en particulier le nom du fichier, sa taille en mégaoctets et le nombre de pages du document. Exemple de code :

```
# PowerShell command to analyze properties of a PDF file
# Specify the full path to the pdfinfo executable
$pdfinfoExecutable = "C:\Program Files\poppler-22.04.0\Library\bin\pdfinfo.exe"  # Replace with your actual path
# Specify the path to the folder containing PDF files
$folder = "C:\Path\to\folder\Mercure"
# Get the list of PDF files in the folder
$pdfFiles = Get-ChildItem -Path $folder -Filter *.pdf
# Initialize an array to store property data
$dataArray = @()
# Iterate through each PDF file
foreach ($file in $pdfFiles) {
```

```
    # Get the properties of the file
    $properties = Get-Item $file.FullName
    # Convert size from bytes to megabytes
    $sizeMB = $properties.Length / 1MB
    # Format the size with one decimal place
    $formattedSize = "{0:N1}" -f $sizeMB
    # Remove the last four characters from the file name (the .pdf extension)
    $fileNameWithoutLastCharacters = $file.Name.Substring(0, $file.Name.Length -
4)
    # Use pdfinfo to get the number of pages
    $pdfInfo = & $pdfinfoExecutable $file.FullName
    # Extract the number of pages from the text returned by pdfinfo
    $numberOfPages = ($pdfInfo -split "`n" | Where-Object { $_ -like "Pages:*"
}).Split(":")[1].Trim()
    # Create a custom object with relevant properties
    $objectProperties = [PSCustomObject]@{
        'name' = $fileNameWithoutLastCharacters
        'path' = $file.FullName
        'size' = $formattedSize
        'pages' = $numberOfPages
        'creation' = $properties.CreationTime
    }
    # Add the object to the array
    $dataArray += $objectProperties
}
# Export the data array to a CSV file with UTF-8 encoding
$dataArray | Export-Csv -Path "C:\Path\to\file\Propriétés des PDF.csv" -
NoTypeInformation -Encoding UTF8
```

3. Le fichier « Propriétés des PDF » peut être importé dans la table « digital_copies » d'une base de données que nous nommons également « digital_copies », base chargée sur un serveur MariaDB 10.
4. Dans notre cas, nous allons plus tard croiser ces données avec celles du site *Les énigmes de Mercure. Base de données des énigmes, logogriphes et charades du Mercure galant et du Mercure de France, 1673-1799* (voir plus loin). Cela nous permettra de générer automatiquement certaines données dont nous avons besoin, comme la source de la numérisation et le lien vers l'élément concerné dans la bibliothèque numérique d'où il est tiré. Cette partie de la procédure n'est possible qu'en disposant d'un accès de niveau administrateur à la base de données du site *Les énigmes de Mercure* qui n'a pas de fonctionnalité publique d'exportation. L'étape est donc optionnelle.
    a. Nous exportons la table « enigmes » de la base de données « enigmes » sous la forme d'un tableau associatif PHP qui sera utile plus tard. Voici un exemple partiel du résultat (fichier « tex_enigma_array.php ») :

```
$tex_enigma_array = array(
  array('id' => '42','date_volume' => '1724-01-01','numero_volume' =>
'0','page_debut' => '86','page_fin' => '86','titre' => 'PREMIERE
ENIGME.','titre_normalise' => 'Première énigme','genre_1' => 'Énigme','genre_2' =>
'','incipit' => 'Je sers & la nuit, & le jour,','incipit_normalise' => 'Je sers et
la nuit et le jour','lieu' => '','lieu_normalise_geonames' => '','latitude' =>
'','longitude' => '','date_texte' => '0000-00-00','auteur' =>
'','auteur_normalise' => '','activite' => '','activite_normalisee' => '','reponse'
=> 'Fontaine','reponse_normalisee' => 'Fontaine','lien' =>
'https://books.google.ch/books?id=VigXVrPJw5cC&hl=fr&pg=PA86#v=onepage&q&f=false',
'iframe' => '<iframe frameborder="0" scrolling="no" style="border:0px"
```

```
src="https://books.google.ch/books?id=VigXVrPJw5cC&hl=fr&pg=PA86&output=embed"
width=500 height=500></iframe>','note' => ''),
  array('id' => '43','date_volume' => '1724-01-01','numero_volume' =>
'0','page_debut' => '87','page_fin' => '87','titre' => 'SECONDE
ENIGME.','titre_normalise' => 'Seconde énigme','genre_1' => 'Énigme','genre_2' =>
'','incipit' => 'Dans un double & sombre parterre,','incipit_normalise' => 'Dans
un double et sombre parterre','lieu' => '','lieu_normalise_geonames' =>
'','latitude' => '','longitude' => '','date_texte' => '0000-00-00','auteur' =>
'','auteur_normalise' => '','activite' => '','activite_normalisee' => '','reponse'
=> 'Trictrac','reponse_normalisee' => 'Trictrac','lien' =>
'https://books.google.ch/books?id=VigXVrPJw5cC&hl=fr&pg=PA87#v=onepage&q&f=false',
'iframe' => '<iframe frameborder="0" scrolling="no" style="border:0px"
src="https://books.google.ch/books?id=VigXVrPJw5cC&hl=fr&pg=PA87&output=embed"
width=500 height=500></iframe>','note' => ''),
  array('id' => '44','date_volume' => '1724-01-01','numero_volume' =>
'0','page_debut' => '87','page_fin' => '88','titre' => 'TROISIÈME
ENIGME.','titre_normalise' => 'Troisième énigme','genre_1' => 'Énigme','genre_2'
=> '','incipit' => 'Tirée du sein de ma mere,','incipit_normalise' => 'Tirée du
sein de ma mère','lieu' => '','lieu_normalise_geonames' => '','latitude' =>
'','longitude' => '','date_texte' => '0000-00-00','auteur' =>
'','auteur_normalise' => '','activite' => '','activite_normalisee' => '','reponse'
=> 'Broche','reponse_normalisee' => 'Broche','lien' =>
'https://books.google.ch/books?id=VigXVrPJw5cC&hl=fr&pg=PA87#v=onepage&q&f=false',
'iframe' => '<iframe frameborder="0" scrolling="no" style="border:0px"
src="https://books.google.ch/books?id=VigXVrPJw5cC&hl=fr&pg=PA87&output=embed"
width=500 height=500></iframe>','note' => ''),
  array('id' => '73','date_volume' => '1724-02-01','numero_volume' =>
'0','page_debut' => '290','page_fin' => '290','titre' => 'PREMIERE
ENIGME.','titre_normalise' => 'Première énigme','genre_1' => 'Énigme','genre_2' =>
'','incipit' => 'Mon destin est des plus bizares.','incipit_normalise' => 'Mon
destin est des plus bizarres','lieu' => '','lieu_normalise_geonames' =>
'','latitude' => '','longitude' => '','date_texte' => '0000-00-00','auteur' =>
'','auteur_normalise' => '','activite' => '','activite_normalisee' => '','reponse'
=> 'Cloche','reponse_normalisee' => 'Cloche','lien' =>
'https://books.google.ch/books?id=qMBS2i6foRMC&hl=fr&pg=PA290#v=onepage&q&f=false'
,'iframe' => '<iframe frameborder="0" scrolling="no" style="border:0px"
src="https://books.google.ch/books?id=qMBS2i6foRMC&hl=fr&pg=PA290&output=embed"
width=500 height=500></iframe>','note' => ''),
  /** [...] **/
  array('id' => '6904','date_volume' => '1799-10-17','numero_volume' =>
'35','page_debut' => '199','page_fin' => '199','titre' =>
'ÉNIGME.','titre_normalise' => 'Énigme','genre_1' => 'Énigme','genre_2' =>
'','incipit' => 'De la boîte de Pandore,','incipit_normalise' => 'De la boîte de
Pandore','lieu' => '','lieu_normalise_geonames' => '','latitude' => '','longitude'
=> '','date_texte' => '0000-00-00','auteur' => 'B. J. V. L.
R......','auteur_normalise' => 'B. J. V. L. R......','activite' =>
'','activite_normalisee' => '','reponse' => '','reponse_normalisee' => '[Réponse
inconnue]','lien' =>
'https://books.google.ch/books?id=TlwDVotvF0kC&hl=fr&pg=PT7#v=onepage&q&f=false','
iframe' => '<iframe frameborder="0" scrolling="no" style="border:0px"
src="https://books.google.ch/books?id=TlwDVotvF0kC&hl=fr&pg=PT7&output=embed"
width=500 height=500></iframe>','note' => 'Le Mercure semble s\'interrompre à
cette date ou peu après pour plusieurs mois. Nous ignorons donc la réponse de ce
texte.'),
  array('id' => '6905','date_volume' => '1787-02-17','numero_volume' =>
'7','page_debut' => '99','page_fin' => '99','titre' => 'IV. ENIGME en Bouts-
rimés.','titre_normalise' => 'IV. Énigme en bouts-rimés','genre_1' =>
'Énigme','genre_2' => 'Autre','incipit' => 'Sans secours merveilleux de Sylphe ni
de Fée,','incipit_normalise' => 'Sans secours merveilleux de sylphe ni de
```

```
fée','lieu' => 'La Motte Chalançon','lieu_normalise_geonames' => 'La Motte-
Chalancon, FR, http://www.gnms.org/3008016','latitude' => '44.48333','longitude'
=> '5.38333','date_texte' => '0000-00-00','auteur' => '','auteur_normalise' =>
'','activite' => '','activite_normalisee' => '','reponse' =>
'Vin','reponse_normalisee' => 'Vin','lien' =>
'https://books.google.ch/books?id=Ty40AAAAMAAJ&hl=fr&pg=RA12-
PA99#v=onepage&q&f=false','iframe' => '<iframe frameborder="0" scrolling="no"
style="border:0px"
src="https://books.google.ch/books?id=Ty40AAAAMAAJ&hl=fr&pg=RA12-
PA99&output=embed" width=500 height=500></iframe>','note' => 'L\'auteur est "un
Abonné".')
);
?>
```

b. Ce fichier PHP doit être enregistré dans le même répertoire que le fichier « csv_digital_copies.php » (voir ci-après).

5. Nous modifions maintenant les données relatives aux numérisations à l'aide du langage PHP, dans la perspective de créer un fichier CSV et d'importer les données dans notre base de données *Drupal*. Exemple de code (fichier « csv_digital_copies.php ») :

```php
<?php
//Digital copies

//Determine the first and last row to be echoed. For $end_row, the value 0 will
echo all the rows.

$start_row = 0;
$end_row = 0;

//Connect to the database.

$mysqli = new MySQLi('localhost','user','password','database');

if(mysqli_connect_errno())
    {
        printf('Échec de la connexion : base de données momentanément
inaccessible.');
        exit();
    }

//Total number of items.

$result_set_1 = $mysqli->query('SELECT COUNT(*) FROM digital_copies');
$total = $result_set_1->fetch_row();

if($end_row == 0){
    $end_row = $total[0];
}

//Query the database.

$result_set_2 = $mysqli->query('SELECT * FROM digital_copies');


//Retrieve and manage data.

$tmpCount = 0;
$total_displayed = 0;
```

```
$output = NULL;

if($result_set_2->num_rows > 0){
    while(null !==($rows = $result_set_2->fetch_assoc()))
    {

//      Assign NULL to the variables.

        $all_remark_private = $all_validity = $all_validity_check =
$dig_cop_download_year = $dig_cop_file = $dig_cop_format = $dig_cop_name =
$dig_cop_number_of_pages = $dig_cop_ocr_quality = $dig_cop_size = $dig_cop_source
= $dig_cop_url = $day = $day_2 = $dummy_date = $dummy_date_2 = $month = $month_2 =
$volume = $year = $year_2 = NULL;

//      Count the lines and retrieve the data from the database.

        $tmpCount ++;
        $name = $rows['name'];
        $path = $rows['path'];
        $size = $rows['size'];
        $pages = $rows['pages'];
        $creation = $rows['creation'];

//      Create field dig_copy_file from name. This must be a public URL beginning
with http or https. The URL given below is only an example.

        $dig_cop_file =
'http://cadsandria.synology.me/evaluation/pdf/'.$name.'.pdf';

        $dig_cop_file = str_ireplace(',', '%2C', $dig_cop_file);
        $dig_cop_file = str_ireplace(' ', '%20', $dig_cop_file);

//      In our case, however, we have decided not to import files automatically,
because our file names contain spaces and commas which, when escaped, make the
file names unreadable. Delete the next line if you wish to import automatically.

        $dig_cop_file = '';

//      Create field dig_cop_name from name.

        if(stripos($name, ',') !== false){
            $dig_cop_name = '"'.$name.'"';
        }
        else{
            $dig_cop_name = $name;
        }

//      Create field dig_cop_number_of_pages from pages.

        $dig_cop_number_of_pages = $pages;

//      Create field dig_cop_format.

        $dig_cop_format = '246'; // Value matching "PDF" in the "Format de
fichier" taxonmy.

//      Create field dig_copy_size from size.

        $dig_cop_size = str_replace(',', '.', $size);
```

```
        if (substr($dig_cop_size, -2) === ".0"){
            $dig_cop_size = substr($dig_cop_size, 0, -2);
        }

//      Create field dig_cop_source when it is possible. To create this field, we
export the "Les énigmes de Mercure" database as an array. Without access to this
resource, this field must be entered manually in the Drupal database.

        include 'tex_enigma_array.php';

        $dummy_date = explode(', ', $name);
        $year = $dummy_date[0];
        $year = substr($year, 0, 4);
        $month = $dummy_date[1];
        $month = substr($month, 0, 2);

        if(isset($dummy_date[2]) || stripos($dummy_date[1], '(') !== false){
            $volume = 'volume or supplement'; // We will use this variable later.
        }

        $year_2 = intval($year);
        $month_2 = intval($month);

        if(($year_2 == 1778 && $month_2 > 5) || $year_2 >= 1779){

            $dummy_date_2 = explode(' (', $name);
            $day = substr($dummy_date_2[1], 0, 2);

            if(stripos($dummy_date_2[1], 'vendémiaire') !== false ||
stripos($dummy_date_2[1], 'brumaire') !== false || stripos($dummy_date_2[1],
'frimaire') !== false || stripos($dummy_date_2[1], 'nivôse') !== false ||
stripos($dummy_date_2[1], 'pluviôse') !== false || stripos($dummy_date_2[1],
'ventôse') !== false || stripos($dummy_date_2[1], 'germinal') !== false ||
stripos($dummy_date_2[1], 'floréal') !== false || stripos($dummy_date_2[1],
'prairial') !== false || stripos($dummy_date_2[1], 'messidor') !== false ||
stripos($dummy_date_2[1], 'thermidor') !== false || stripos($dummy_date_2[1],
'fructidor') !== false){
                $day = substr($dummy_date_2[2], 0, 2);
            }

            $dummy_date_2 = explode(' ('.$day, $name);


            if($dummy_date_2[1] != ' janvier)' || $dummy_date_2[1] != ' février)'
|| $dummy_date_2[1] != ' mars)' || $dummy_date_2[1] != ' avril)' ||
$dummy_date_2[1] != ' mai)' || $dummy_date_2[1] != ' juin)' || $dummy_date_2[1] !=
' juillet)' || $dummy_date_2[1] != ' août)' || $dummy_date_2[1] != ' septembre)'
|| $dummy_date_2[1] != ' octobre)' || $dummy_date_2[1] != ' novembre)' ||
$dummy_date_2[1] != ' décembre)'){
                $volume = 'volume or supplement';
            }

            $day_2 = intval($day);

            if($day_2 < 10){
                $day = '0'.$day_2;
            }
            elseif($day_2 > 31){
```

```
                $day = '01';
            }
            else{
                $day = strval($day_2);
            }
        }
        else{
            $day = '01';
        }

        $dummy_date = $year.'-'.$month.'-'.$day;

        foreach($tex_enigma_array as $value){
            if($value['date_volume'] === $dummy_date){
                $dig_cop_source = $value['lien'];
                if(stripos($dig_cop_source, 'google') !== false){
                    $dig_cop_source = '247'; // Value matching "Google Books" in
the "Source de la copie numérique" tayonomy.
                }
                elseif(stripos($dig_cop_source, 'gallica') !== false){
                    $dig_cop_source = '248'; // Value matching "Gallica" in the
"Source de la copie numérique" tayonomy.
                }
                elseif(stripos($dig_cop_source, 'reader') !== false){
                    $dig_cop_source = '249'; // Value matching "Bayerische
Staatsbibliothek" in the "Source de la copie numérique" tayonomy.
                }
                else{
                    $dig_cop_source = '';
                }
                break;
            }
        }

//      Create field dig_cop_url when it is possible. To create this field, we use
the array tex_enigma_array (see above). Without access to this resource, this
field must be entered manually in the Drupal database.

        foreach($tex_enigma_array as $value){
            if($value['date_volume'] === $dummy_date){
                $dig_cop_url = $value['lien'];
                if(stripos($dig_cop_url, 'google') !== false){
                    $dig_cop_url = explode('&hl=', $dig_cop_url);
                    $dig_cop_url = $dig_cop_url[0].'&printsec=frontcover';
                }
                elseif(stripos($dig_cop_url, 'gallica') !== false){
                    $dig_cop_url = explode('/', $dig_cop_url);
                    $dig_cop_url =
$dig_cop_url[0].'/'.$dig_cop_url[1].'/'.$dig_cop_url[2].'/'.$dig_cop_url[3].'/'.$d
ig_cop_url[4].'/'.$dig_cop_url[5];
                }
                elseif(stripos($dig_cop_url, 'reader') !== false){
                    $dig_cop_url = explode('/bsb', $dig_cop_url);
                    $dig_cop_url = explode('_', $dig_cop_url[1]);
                    $dig_cop_url = 'https://mdz-nbn-
resolving.de/details:bsb'.$dig_cop_url[0];
                }
                else{
                    $dig_cop_url = '';
```

```
                }
                break;
            }
        }

//      Create field dig_cop_download_year. This field depends on your own
download schedule. The information below is only an example.

        $year = intval($year);
        $month = intval($month);

        if($year <= 1709 || ($year == 1710 && $month <= 5)){ // Mercure galant
(1672-1710)
            if($year <= 1677 || ($year == 1678 && $month == 1) || ($year == 1678
&& $month >= 3 && stripos($name, '(') === false) || ($year == 1679 && $month == 1)
|| ($year == 1679 && $month >= 3 && $month <= 7) || ($year == 1679 && $month >= 8
&& stripos($name, '(') === false) || ($year == 1680 && $month == 1 &&
stripos($name, 'partie 1') !== false) || ($year == 1680 && $month == 2) || ($year
== 1680 && $month >= 5 && $month <= 6) || ($year == 1680 && $month >= 10) ||
($year == 1681 && $month >= 1 && $month <= 5) && ($year == 1681 && $month >= 8 &&
$month <= 9) || ($year == 1681 && $month >= 11 && $month <= 12) || ($year == 1682
&& $month == 1) || ($year == 1682 && $month == 3) || ($year == 1682 && $month >= 6
&& $month <= 8) || ($year == 1682 && $month >= 11 && $month <= 12) || ($year ==
1683 && stripos($name, 'supplément') === false && stripos($name, 'relation') ===
false)){
                $dig_cop_download_year = '2019-01-01';
            }
            else{
                $dig_cop_download_year = '2020-01-01';
            }
        }
        elseif(($year == 1710 && $month > 5) || ($year >= 1711 && $year <= 1713)
|| ($year == 1714 && $month <= 4)){ // Mercure galant (1710-1714).
            if(($year == 1711 && $month >= 4 && $month <= 5) || ($year == 1712 &&
stripos($name, '(') === false) || ($year == 1713 && $month >= 1 && $month <= 4)){
                $dig_cop_download_year = '2020-01-01';
            }
            else{
                $dig_cop_download_year = '2019-01-01';
            }
        }
        elseif(($year == 1714 && $month > 4) || $year == 1715 || ($year == 1716 &&
$month <= 10)){ // Nouveau Mercure galant (1714-1716).
            if(stripos($name, '(') !== false){
                $dig_cop_download_year = '2020-01-01';
            }
            else{
                $dig_cop_download_year = '2019-01-01';
            }
        }
        elseif(($year >= 1717 && $year <= 1720) || ($year == 1721 && $month <=
4)){ // Le Nouveau Mercure (1717-1721).
            $dig_cop_download_year = '2020-01-01';
        }
        elseif(($year == 1721 && $month >= 6) || $year <= 1723){ // Le Mercure
(1721-1723).
            if($year == 1722 && (($month >= 4 && $month <=6) || $month == 9)){
                $dig_cop_download_year = '2020-01-01';
            }
```

```
            else{
                $dig_cop_download_year = '2019-01-01';
            }
        }
        elseif($year > 1723 && $year < 1778 || ($year == 1778 && $month <= 5)){ //
Mercure de France (1724-1778).
            if(($year == 1745 && $month == 6) || ($year == 1749 && $month == 7 &&
stripos($name, '-') !== false) || ($year == 1755 && $month == 7 && stripos($name,
'-') !== false) || ($year == 1770 && $month == 4) || ($year == 1770 && $month ==
10)){
                $dig_cop_download_year = '2020-01-01';
            }
            else{
                $dig_cop_download_year = '2019-01-01';
            }
        }
        elseif(($year == 1778 && $month > 5) || ($year >= 1779 && $year <= 1791)){
// Mercure de France (1778-1791).
            $dig_cop_download_year = '2020-01-01';
        }
        elseif(($year > 1791 && $year < 1799) || ($year == 1799 && $month == 1 &&
stripos($name, 'prospectus') === false)){ // Mercure français (1792-1799).
            $dig_cop_download_year = '2020-01-01';
        }
        elseif($year == 1799){ // Mercure de France, journal politique, littéraire
et dramatique (1799)
            $dig_cop_download_year = '2020-01-01';
        }
        elseif($year > 1799 && $year <= 1816){ // Mercure de France, littéraire et
politique (1800-1816)
            $dig_cop_download_year = '2020-01-01';
        }
        elseif($year == 1817 || $year == 1818){ // Mercure de France (1817-1818)
            $dig_cop_download_year = '2020-01-01';
        }
        elseif($year == 1819 || $year == 1820){ // Mercure de France et chronique
de Paris (1819-1820)
            $dig_cop_download_year = '2020-01-01';
        }
        else{
            $dig_cop_download_year = '';
        }

//      Create field dig_cop_ocr_quality.

        $dig_cop_ocr_quality = '254'; // Value matching "Indéterminée" in the
"Qualité de la reconnaissance optique de caractères" taxonomy.

//      Create field all_remark_private.

        if($dig_cop_source == '247'){
            $source = 'Google Books';
        }
        elseif($dig_cop_source == '248'){
            $source = 'Gallica';
        }
        elseif($dig_cop_source == '249'){
            $source = 'Bayerische Staatsbibliothek';
        }
```

```
        if(isset($volume) && !empty($volume) && isset($dig_cop_source) &&
!empty($dig_cop_source)){
            $all_remark_private = 'Vérifier si le lien vers la numérisation et si
la source ('.$source.') sont corrects. Supprimer ensuite ce commentaire.';
            $all_validity_check = 'control';
        }
        elseif(empty($dig_cop_source)){
            $all_remark_private = 'Indiquer la source et le lien vers la
numérisation. Supprimer ensuite ce commentaire.';
            $all_validity_check = 'complete';
        }

//      Create field all_validity.

        if($all_validity_check == 'control'){
            $all_validity = '83'; // Value matching "En attente de validation" in
the "Validité" taxonomy.
        }
        elseif(stripos($all_validity_check, 'complete') !== false){
            $all_validity = '82'; // Value matching "À compléter" in the
"Validité" taxonomy.
        }
        else{
            $all_validity = '84'; // Value matching "Vérifié" in the "Validité"
taxonomy.
        }

//      Combine the fields in CSV format.

        if(($tmpCount >= $start_row) && ($tmpCount <= $end_row)){
            $output .=
$dig_cop_file.','.$dig_cop_name.','.$dig_cop_number_of_pages.','.$dig_cop_format.'
,'.$dig_cop_size.','.$dig_cop_source.','.$dig_cop_url.','.$dig_cop_download_year.'
,'.$dig_cop_ocr_quality.','.$all_remark_private.','.$all_validity.'<br/>';
            $total_displayed ++;
        }
    }
}


//Echo the number of results.

echo 'Nombre de résultats : '.$total_displayed.'<br/>';
echo 'Affichage des résultats '.$start_row.' à '.$end_row.'<br/><br/>';

//Output in CSV format.

echo
'dig_cop_file,dig_cop_name,dig_cop_number_of_pages,dig_cop_format,dig_cop_size,dig
_cop_source,dig_cop_url,dig_cop_download_year,dig_cop_ocr_quality,all_remark_priva
te,all_validity<br/>';
echo $output;
?>
```

6. Copier les données depuis un navigateur web et coller ces données dans un nouveau fichier CSV vide.

7. Importer les données dans *Drupal*, grâce à l'outil d'importation du module Webform. Certaines données doivent être complétées et/ou vérifiées manuellement, *via* l'interface du site *Drupal* en préparation.

Les énigmes de Mercure. Base de données des énigmes, logogriphes et charades du *Mercure galant* et du *Mercure de France*, 1673-1799

DONNÉES RELATIVES AUX TEXTES

Objectifs

Le site *Les énigmes de Mercure* (https://enigmes.othone.org, 24 octobre 2023) recense les énigmes, logogriphes et charades publiés dans le *Mercure galant* et le *Mercure de France* entre 1673 et 1799 avec, pour chaque texte, des données relatives au texte lui-même et, lorsque cela est possible, à son auteur. Nous récupérons sous la forme de fichiers CSV l'ensemble des données pertinentes pour les intégrer à notre base de données, après les avoir retouchées et enrichies.

Données concernées

L'importation concerne les énigmes, logogriphes et charades publiés dans le périodique entre 1673 et 1799, pour autant qu'ils forment des textes à part entière (nous excluons les énigmes qui sont proposées à l'intérieur d'un autre texte, par exemple un compte rendu de livre). Les données suivantes sont extraites :

- id : l'identifiant de l'énigme dans la base de données du site *Les énigmes de Mercure*.
- date_volume : la date de la livraison où paraît l'énigme.
- numero_volume : le numéro de la livraison.
- page_debut : la page de début de l'énigme dans la livraison.
- page_fin : la page de fin de l'énigme dans la livraison.
- titre : le titre de l'énigme tel qu'il apparaît dans la livraison.
- titre_normalise : le titre sous une forme standardisée du point de vue de la ponctuation et de la typographie.
- genre_1 : le genre littéraire (énigme, logogriphe, etc.).
- genre_2 : le second genre littéraire, si l'énigme appartient à plusieurs genres (dans le cas, par exemple, d'une énigme logogriphique ou d'une énigme en sonnet).
- incipit : le premier vers ou, si l'énigme est en prose, les premiers mots du texte.
- incipit_normalise : le premier vers ou les premiers mots, transcrits sous une forme standardisée du point de vue de la ponctuation et de la typographie.
- lieu : le lieu de provenance de l'énigme, tel qu'il apparaît dans le titre ou la signature du texte.
- lieu_normalise_geonames : le lieu sous une forme standardisée, construite d'après le toponyme moderne, le code ISO du pays et un lien externe vers la notice correspondante du site *GeoNames*.
- latitude : la latitude du lieu en degrés décimaux.
- longitude : la longitude du lieu en degrés décimaux.
- date_texte : la date de rédaction ou d'envoi du texte, lorsqu'elle est précisée.
- auteur : le nom de l'auteur, tel qu'il apparaît dans le titre ou la signature du texte.

- auteur_normalise : le nom de l'auteur sous une forme standardisée du point de vue de la ponctuation et de la typographie, et composé de façon à favoriser les recoupements avec d'autres textes du même auteur.
- activite : la profession de l'auteur ou autre activité indiquée dans la signature ou dans un autre élément de paratexte.
- activite_normalisee : la profession ou autre activité, indiquée sous une forme standardisée du point de vue de la ponctuation et de la typographie.
- reponse : la réponse de l'énigme, généralement indiquée dans la livraison suivante.
- reponse_normalisee : la réponse sous une forme standardisée du point de vue de la ponctuation et de la typographie.
- lien : le lien externe vers la numérisation du texte dans une bibliothèque numérique (*Google Books*, *Gallica*, etc.).
- iframe : un fragment de code permettant d'afficher la numérisation du texte sous la forme d'un iframe, à partir d'une bibliothèque numérique.
- note : d'éventuelles remarques relatives aux autres saisies, par exemple pour motiver l'attribution du texte à un auteur.

Utilisation

Dans la base de données de notre site, toutes ces données sont stockées, mais elles ne sont pas toutes affichées. La plupart des données qui sont affichées sont préalablement enrichies et/ou modifiées de façon à répondre aux besoins du site. Une fois chargées dans *Drupal*, les données ne peuvent plus être réimportées. Elles sont susceptibles d'être corrigées ou complétées par les collaborateurs du site. Que les données soient modifiées ou non, leur source est toujours indiquée et un lien hypertexte permet d'accéder à la fiche d'énigme correspondante sur le site *Les énigmes de Mercure*.

Procédure détaillée

1. Les données du site *Les énigmes de Mercure* sont rassemblées dans une base dont l'accès est réservé aux administrateurs, sans fonctionnalité publique d'exportation. Par conséquent, nous ne fournissons aucune procédure pour récupérer ces données. Il convient de s'adresser à l'administrateur.
2. La première étape consiste à importer les noms d'auteurs (champ auteur) dans la taxonomie « Nom » de notre site *Drupal*. Dans l'exemple suivant, nous nous connectons directement à la base de données des *Énigmes de Mercure*.
    a. Nous générons un fichier CSV au moyen du code suivant (fichier « csv_enigma_text_author.php ») :

```php
<?php
//Text, enigmas: author names

//Determine the first and last row to be echoed. For $end_row, the value 0 will
echo all the rows.

$start_row = 0;
$end_row = 0;

//Connect to the database.

$mysqli = new MySQLi('localhost','user','password','database');
```

```php
if(mysqli_connect_errno())
    {
        printf('Échec de la connexion : base de données momentanément
inaccessible.');
        exit();
    }

//Total number of items.

$result_set_1 = $mysqli->query('SELECT COUNT(*) FROM enigmes');
$total = $result_set_1->fetch_row();

if($end_row == 0){
    $end_row = $total[0];
}

//Query the database.

$result_set_2 = $mysqli->query('SELECT auteur FROM enigmes WHERE auteur != \'\'
AND id >= '.$start_row.' AND id <= '.$end_row.' GROUP BY auteur');

//Retrieve and manage data.

$total_displayed = 0;
$output = NULL;

if($result_set_2->num_rows > 0){
    while(null !==($rows = $result_set_2->fetch_assoc()))
    {

//      Retrieve "auteur" from the database.

        $author = $rows['auteur'];

//      Define the output.

        $output .= $author.'<br/>';
        $total_displayed ++;

    }
}

//Echo the number of results.

echo 'Nombre de résultats : '.$total_displayed.'<br/>';
echo 'Affichage des résultats '.$start_row.' à '.$end_row.'<br/><br/>';

//Output in CSV format.

echo 'author<br/>';
echo $output;

?>
```

b. Ces données peuvent ensuite être importées dans la taxonomie « Nom » de *Drupal* au moyen d'un outil d'importation comme le module Taxonomy Import.

3. Création d'un fichier CSV importable dans la base de données définitive. Il s'agit de retoucher les données brutes du site *Les énigmes de Mercure*, de façon à permettre leur importation dans les champs correspondants de la base de données de notre site, grâce à la fonction d'importation du module Webform de *Drupal*. Pour ce faire, nous exportons dans un tableau associatif PHP les données brutes rassemblées dans la base de données « enigmes », nous les modifions en langage PHP, puis nous affichons sur une page web l'ensemble des données modifiées, de manière à pouvoir les copier et les coller dans un nouveau fichier CSV. Cette procédure implique que toutes les copies numériques et les livraisons du *Mercure de France* aient été saisies dans la base de données de notre site, de façon à pouvoir lier chaque énigme à la livraison qui lui correspond et à indiquer les bons numéros de pages de l'énigme dans la copie numérique.
    a. Exporter les données de la base « enigmes », table « enigmes », sous forme d'un tableau associatif PHP : voir ci-dessus la procédure d'importation des copies numériques, point 4. a. Nous nommons le fichier « tex_enigma_array.php » et y recourrons plus tard.
    b. À partir de la base de données *Drupal*, créer un tableau associatif PHP qui réunisse pour chaque livraison son identifiant (NID), sa date (all_date_gregorian_calendar), l'identifiant du périodique dont il fait partie (par la relation iss_is_part_of_peri), l'identifiant de sa copie numérique principale (par la relation iss_has_dig_cop) et le lien vers la source de cette copie numérique (dig_cop_url). Ce tableau associatif peut être préparé en configurant une vue de *Drupal* spécialement conçue pour son affichage dans un navigateur. Il suffit ensuite de le copier dans un fichier PHP vierge. Nous y recourrons plus tard. Le tableau doit ressembler à ceci (fichier « iss_peri_dig_cop_array.php ») :

```php
<?php
$iss_peri_dig_cop_array = array(
    array('issue_nid' => '1382496','issue_all_date_gregorian_calendar' => '1672-
05-01','issue_volume' => '1','issue_number' => '','periodical_id' =>
'1080292','digital_copy_nid' => '1043910','dig_cop_url' =>
'https://gallica.bnf.fr/ark:/12148/bpt6k32948j'),
    array('issue_nid' => '1382497','issue_all_date_gregorian_calendar' => '1673-
01-01','issue_volume' => '2','issue_number' => '','periodical_id' =>
'1080292','digital_copy_nid' => '1043910','dig_cop_url' =>
'https://gallica.bnf.fr/ark:/12148/bpt6k32948j'),
    array('issue_nid' => '1382498','issue_all_date_gregorian_calendar' => '1673-
01-01','issue_volume' => '3','issue_number' => '','periodical_id' =>
'1080292','digital_copy_nid' => '1043910','dig_cop_url' =>
'https://gallica.bnf.fr/ark:/12148/bpt6k32948j'),
    array('issue_nid' => '1382499','issue_all_date_gregorian_calendar' => '1673-
06-01','issue_volume' => '4','issue_number' => '','periodical_id' =>
'1080292','digital_copy_nid' => '1043910','dig_cop_url' =>
'https://gallica.bnf.fr/ark:/12148/bpt6k32948j'),
    array('issue_nid' => '1382500','issue_all_date_gregorian_calendar' => '1674-
01-01','issue_volume' => '5','issue_number' => '','periodical_id' =>
'1080292','digital_copy_nid' => '1043910','dig_cop_url' =>
'https://gallica.bnf.fr/ark:/12148/bpt6k32948j'),
    /** [...] **/
    array('issue_nid' => '1386052','issue_all_date_gregorian_calendar' => '1819-
11-01','issue_volume' => '2','issue_number' => '','periodical_id' =>
'1105908','digital_copy_nid' => '1045330','dig_cop_url' =>
'https://books.google.fr/books?id=svIhuK0_j_oC&printsec=frontcover'),
```

```
    array('issue_nid' => '1386053','issue_all_date_gregorian_calendar' => '1819-
11-01','issue_volume' => '2','issue_number' => '','periodical_id' =>
'1105908','digital_copy_nid' => '1045330','dig_cop_url' =>
'https://books.google.fr/books?id=svIhuK0_j_oC&printsec=frontcover'),
    array('issue_nid' => '1386055','issue_all_date_gregorian_calendar' => '1819-
12-01','issue_volume' => '2','issue_number' => '','periodical_id' =>
'1105908','digital_copy_nid' => '1045331','dig_cop_url' =>
'https://gallica.bnf.fr/ark:/12148/bpt6k4246142x'),
    array('issue_nid' => '1386056','issue_all_date_gregorian_calendar' => '1819-
12-01','issue_volume' => '2','issue_number' => '','periodical_id' =>
'1105908','digital_copy_nid' => '1045330','dig_cop_url' =>
'https://books.google.fr/books?id=svIhuK0_j_oC&printsec=frontcover'),
    array('issue_nid' => '1386057','issue_all_date_gregorian_calendar' => '1820-
02-01','issue_volume' => '2','issue_number' => '','periodical_id' =>
'1105908','digital_copy_nid' => '1045330','dig_cop_url' =>
'https://books.google.fr/books?id=svIhuK0_j_oC&printsec=frontcover')
);
?>
```

c. Créer un tableau associatif PHP qui réunisse l'identifiant de l'énigme dans la base de données du site *Les énigmes de Mercure* et les pages de début et de fin de l'énigme dans le fichier PDF de la numérisation. Ces numéros de pages ne sont pas enregistrés dans la base de données du site des énigmes ; ils ont été saisis dans un fichier de travail pendant la phase de développement du site concerné. Nous recourrons plus tard à ce tableau associatif. Il devrait ressembler à ceci (fichier « enigma_pdf_pages_array.php ») :

```php
<?php
$enigma_pdf_pages_array = array(
    "4075" => array("pdf_start_page" => 123, "pdf_end_page" => 125),
    "270" => array("pdf_start_page" => 21, "pdf_end_page" => 22),
    "3539" => array("pdf_start_page" => 59, "pdf_end_page" => 59),
    "3540" => array("pdf_start_page" => 176, "pdf_end_page" => 176),
    "3541" => array("pdf_start_page" => 145, "pdf_end_page" => 147),
    /** [...] **/
    "6900" => array("pdf_start_page" => 28, "pdf_end_page" => 29),
    "6901" => array("pdf_start_page" => 74, "pdf_end_page" => 74),
    "6902" => array("pdf_start_page" => 75, "pdf_end_page" => 75),
    "6903" => array("pdf_start_page" => 123, "pdf_end_page" => 123),
    "6904" => array("pdf_start_page" => 124, "pdf_end_page" => 124),
);
?>
```

d. À partir de la liste des réponses standardisées d'énigmes (champ « reponse_normalisee »), ajouter dans la taxonomie de la base de données *Drupal* (vocabulaire « Mots clefs ») les éléments pertinents qui n'apparaissent pas encore comme des mots clefs. Il est possible de s'aider d'un outil d'importation comme le module Taxonomy import. L'élément « Dé à coudre » est pertinent, tandis que « [Réponse inconnue] » ne constitue pas un mot clef pertinent. Une réponse sandardisée d'énigme comme « Janot [personnage de comédie] » doit être transformée en « Janot ».

e. À partir de la liste des activités de la base de données des *Énigmes de Mercure* (champ « activite_normalisee »), ajouter les éléments qui n'apparaissent pas encore dans le vocabulaire « Activité » de la base de données *Drupal*. De même, à partir de la liste des lieux de la base des énigmes (champ « lieu »), ajouter les

éléments qui n'apparaissent pas encore dans le vocabulaire « Lieu » de la base *Drupal*. Enfin, à partir de la liste des auteurs de la base des énigmes (champ « auteur »), ajouter les éléments qui n'apparaissent pas encore dans le vocabulaire « Nom » de la base *Drupal*. Un module de *Drupal* comme Taxonomy import peut faciliter ces tâches d'importation. Toutefois, avant ou après l'importation, chacune des trois listes nécessite des retouches manuelles pour adapter certains termes aux nouveaux standards du site sur le *Mercure de France*.

f. Une fois ces ajouts terminés, extraire l'ensemble des termes de taxonomie de la base de données *Drupal* (table « taxonomy_term_field_data ») sous la forme d'un tableau associatif PHP. Nous recourrons à ce tableau associatif dans le code de préparation du fichier CSV (ci-dessous). Il devrait ressembler à ceci (fichier « taxonomy_term_array.php ») :

```
<?php
$taxonomy_term_array = array(
    array('tid' => '82','revision_id' => '82','vid' => 'validity','langcode' =>
'fr','status' => '1','name' => 'À compléter','description__value' =>
NULL,'description__format' => NULL,'weight' => '0','changed' =>
'1683048078','default_langcode' => '1','revision_translation_affected' => '1'),
    array('tid' => '83','revision_id' => '83','vid' => 'validity','langcode' =>
'fr','status' => '1','name' => 'En attente de validation','description__value' =>
NULL,'description__format' => NULL,'weight' => '1','changed' =>
'1683048190','default_langcode' => '1','revision_translation_affected' => '1'),
    array('tid' => '84','revision_id' => '84','vid' => 'validity','langcode' =>
'fr','status' => '1','name' => 'Vérifié','description__value' =>
NULL,'description__format' => NULL,'weight' => '2','changed' =>
'1683048190','default_langcode' => '1','revision_translation_affected' => '1'),
    array('tid' => '86','revision_id' => '86','vid' => 'section','langcode' =>
'fr','status' => '1','name' => 'Pièces fugitives en prose et en
vers','description__value' => NULL,'description__format' => NULL,'weight' =>
'0','changed' => '1722962609','default_langcode' =>
'1','revision_translation_affected' => '1'),
    array('tid' => '87','revision_id' => '87','vid' => 'section','langcode' =>
'fr','status' => '1','name' => 'Nouvelles littéraires, des beaux-arts,
etc.','description__value' => NULL,'description__format' => NULL,'weight' =>
'0','changed' => '1683167245','default_langcode' =>
'1','revision_translation_affected' => '1'),
    /** [...] **/
    array('tid' => '17696','revision_id' => '17696','vid' => 'keyword','langcode'
=> 'fr','status' => '1','name' => 'Voyelle E','description__value' =>
NULL,'description__format' => NULL,'weight' => '0','changed' =>
'1723845100','default_langcode' => '1','revision_translation_affected' => '1'),
    array('tid' => '17697','revision_id' => '17697','vid' => 'keyword','langcode'
=> 'fr','status' => '1','name' => 'Voyelle U','description__value' =>
NULL,'description__format' => NULL,'weight' => '0','changed' =>
'1723845100','default_langcode' => '1','revision_translation_affected' => '1'),
    array('tid' => '17698','revision_id' => '17698','vid' => 'keyword','langcode'
=> 'fr','status' => '1','name' => 'Voyelles','description__value' =>
NULL,'description__format' => NULL,'weight' => '0','changed' =>
'1723845100','default_langcode' => '1','revision_translation_affected' => '1'),
    array('tid' => '17699','revision_id' => '17699','vid' => 'keyword','langcode'
=> 'fr','status' => '1','name' => 'Vulpes','description__value' =>
NULL,'description__format' => NULL,'weight' => '0','changed' =>
'1723845100','default_langcode' => '1','revision_translation_affected' => '1'),
    array('tid' => '17700','revision_id' => '17700','vid' => 'keyword','langcode'
=> 'fr','status' => '1','name' => 'Zéphyr','description__value' =>
```

```
NULL,'description__format' => NULL,'weight' => '0','changed' =>
'1723845100','default_langcode' => '1','revision_translation_affected' => '1')
);
?>
```

g. Nous pouvons maintenant modifier en langage PHP les données issues du site *Les énigmes de Mercure*, de façon à les importer dans notre base de données *Drupal*. Exemple de code (fichier « csv_enigma_text.php ») :

```
<?php
//Enigmas: create a CSV file from "Les énigmes de Mercure" database.

include('tex_enigma_array.php'); //Export of "Les énigmes de Mercure" database.
See documentation.

//Determine the first and last row to be echoed. For $end_row, the value 0 will
echo all the rows.

$start_row = 0;
$end_row = 0;

//Total number of items.

$total = count($tex_enigma_array);

if($end_row == 0){
    $end_row = $total;
}

$tmpCount = 0;
$total_displayed = 0;
$output = NULL;

foreach ($tex_enigma_array as $enigma){

    //Assign NULL to the variables.

    $all_date_gregorian_calendar = $all_date_republican_calendar = $all_end_page =
$all_end_page_file = $all_remark = $all_remark_private = $all_start_page =
$all_start_page_file = $all_validity = $auteur_strtolower = $date_check =
$date_explode = $date_number_check = $day_check = $enigma_issue_check = $id_check
= $id_geonames = $id_url_dig_cop = $id_url_enigma = $iss_peri_dig_cop_value =
$issue_date = $issue_nid = $issue_number = $issue_number_check = $issue_volume =
$issue_volume_check = $month_check = $number_check = $place_check =
$tex_author_activity = $tex_author_collectivity = $tex_author_gender =
$tex_author_name = $tex_author_title_of_nobility = $tex_comes_from_pla =
$tex_concerns_wor = $tex_domain = $tex_enigma_answer = $tex_enigma_date =
$tex_enigma_digital_copy_iframe = $tex_enigma_digital_copy_url =
$tex_enigma_genre_one = $tex_enigma_genre_two = $tex_enigma_id =
$tex_enigma_issue_date = $tex_enigma_issue_number = $tex_enigma_place_geonames =
$tex_enigma_place_latitude = $tex_enigma_place_longitude = $tex_enigma_remark =
$tex_enigma_standard_answer = $tex_enigma_standard_author_name =
$tex_enigma_standard_incipit = $tex_enigma_standard_title =
$tex_enig_stand_author_occupation = $tex_incipit = $tex_is_addressed_to_per =
$tex_is_continuation_of_tex = $tex_is_part_of_fol = $tex_is_part_of_iss =
$tex_is_part_of_sec = $tex_is_probably_addressed_to_per =
$tex_is_probably_written_by_per = $tex_is_reproduction_of_tex =
$tex_is_response_to_tex = $tex_is_written_by_per = $tex_keyword = $tex_language =
```

```
$tex_literary_genre = $tex_ocr = $tex_paratextual_genre = $tex_place =
$tex_position_on_page = $tex_probably_comes_from_pla = $tex_probably_concerns_wor
= $tex_readers_letters = $tex_rec_article_short_title = $tex_signature =
$tex_title = $tex_title_table = $tex_type_of_journalistic_writing =
$tex_verse_prose = $url_matches = $year_check = $year_month_check = NULL;

    //Assign FALSE to two variables.

    $author_male = false;
    $author_female = false;

    //Retrieve and manage data.

    $tmpCount ++;
    $id = $enigma['id'];
    $date_volume = $enigma['date_volume'];
    $numero_volume = $enigma['numero_volume'];
    $page_debut = $enigma['page_debut'];
    $page_fin = $enigma['page_fin'];
    $titre = $enigma['titre'];
    $titre_normalise = $enigma['titre_normalise'];
    $genre_1 = $enigma['genre_1'];
    $genre_2 = $enigma['genre_2'];
    $incipit = $enigma['incipit'];
    $incipit_normalise = $enigma['incipit_normalise'];
    $lieu = $enigma['lieu'];
    $lieu_normalise_geonames = $enigma['lieu_normalise_geonames'];
    $latitude = $enigma['latitude'];
    $longitude = $enigma['longitude'];
    $date_texte = $enigma['date_texte'];
    $auteur = $enigma['auteur'];
    $auteur_normalise = $enigma['auteur_normalise'];
    $activite = $enigma['activite'];
    $activite_normalisee = $enigma['activite_normalisee'];
    $reponse = $enigma['reponse'];
    $reponse_normalisee = $enigma['reponse_normalisee'];
    $lien = $enigma['lien'];
    $iframe = $enigma['iframe'];
    $note = $enigma['note'];

    //Check the volume date, volume number and enigma ID.

    $date_explode = explode('-', $date_volume);
    $year_check = $date_explode[0];
    $year_month_check = $date_explode[0].'-'.$date_explode[1];
    $month_check = $date_explode[1];
    $date_check = $date_volume;
    $date_number_check = $date_volume.'-'.$numero_volume;
    $day_check = $date_explode[2];
    $number_check = $numero_volume;
    $id_check = $id;

    //Create field tex_enigma_id from id.

    $tex_enigma_id = $id;

    //Create field tex_enigma_issue_date from date_volume.

    $tex_enigma_issue_date = $date_volume;
```

```
    //Create field tex_enigma_issue_number from numero_volume.

    $tex_enigma_issue_number = $numero_volume;

    //Create field tex_enigma_standard_title from titre_normalise.

    $tex_enigma_standard_title = $titre_normalise;

    $tex_enigma_standard_title = str_replace('"', '""',
$tex_enigma_standard_title);
    if(stripos($tex_enigma_standard_title, ',') !== false){
        $tex_enigma_standard_title = '"'.$tex_enigma_standard_title.'"';
    }

    //Create field tex_enigma_genre_one from genre_1.

    $tex_enigma_genre_one = $genre_1;

    //Create field tex_enigma_genre_two from genre_2.

    $tex_enigma_genre_two = $genre_2;

    //Create field tex_enigma_standard_incipit from incipit_normalise.

    $tex_enigma_standard_incipit = $incipit_normalise;

    $tex_enigma_standard_incipit = str_replace('"', '""',
$tex_enigma_standard_incipit);
    if(stripos($tex_enigma_standard_incipit, ',') !== false){
        $tex_enigma_standard_incipit = '"'.$tex_enigma_standard_incipit.'"';
    }

    //Create field tex_enigma_place_geonames from lieu_normalise_geonames.

    $tex_enigma_place_geonames = $lieu_normalise_geonames;

    $tex_enigma_place_geonames = str_replace('"', '""',
$tex_enigma_place_geonames);
    if(stripos($tex_enigma_place_geonames, ',') !== false){
        $tex_enigma_place_geonames = '"'.$tex_enigma_place_geonames.'"';
    }

    //Create field tex_enigma_place_latitude from latitude.

    $tex_enigma_place_latitude = $latitude;

    //Create field tex_enigma_place_longitude from longitude.

    $tex_enigma_place_longitude = $longitude;

    //Create field tex_enigma_date from date_text.

    $tex_enigma_date = $date_texte;

    //Create field tex_enigma_standard_author_name from auteur_normalise.

    $tex_enigma_standard_author_name = $auteur_normalise;
```

```
    $tex_enigma_standard_author_name = str_replace('"', '""',
$tex_enigma_standard_author_name);
    if(stripos($tex_enigma_standard_author_name, ',') !== false){
        $tex_enigma_standard_author_name =
'"'.$tex_enigma_standard_author_name.'"';
    }

    //Create field tex_enig_stand_author_occupation from activite_normalisee.

    $tex_enig_stand_author_occupation = $activite_normalisee;

    $tex_enig_stand_author_occupation = str_replace('"', '""',
$tex_enig_stand_author_occupation);
    if(stripos($tex_enig_stand_author_occupation, ',') !== false){
        $tex_enig_stand_author_occupation =
'"'.$tex_enig_stand_author_occupation.'"';
    }

    //Create field tex_enigma_answer from reponse.

    $tex_enigma_answer = $reponse;

    $tex_enigma_answer = str_replace('"', '""', $tex_enigma_answer);
    if(stripos($tex_enigma_answer, ',') !== false){
        $tex_enigma_answer = '"'.$tex_enigma_answer.'"';
    }

    //Create field tex_enigma_standard_answer from reponse_normalisee.

    $tex_enigma_standard_answer = $reponse_normalisee;

    $tex_enigma_standard_answer = str_replace('"', '""',
$tex_enigma_standard_answer);
    if(stripos($tex_enigma_standard_answer, ',') !== false){
        $tex_enigma_standard_answer = '"'.$tex_enigma_standard_answer.'"';
    }

    //Create field tex_enigma_digital_copy_url from lien.

    $tex_enigma_digital_copy_url = $lien;

    //Create field tex_enigma_digital_copy_iframe from iframe.

    $tex_enigma_digital_copy_iframe = htmlspecialchars($iframe, ENT_QUOTES, 'UTF-
8');

    $tex_enigma_digital_copy_iframe = str_replace('"', '""',
$tex_enigma_digital_copy_iframe);
    if(stripos($tex_enigma_digital_copy_iframe, ',') !== false){
        $tex_enigma_digital_copy_iframe = '"'.$tex_enigma_digital_copy_iframe.'"';
    }

    //Create field tex_enigma_remark from note.

    $tex_enigma_remark = $note;

    $tex_enigma_remark = str_replace('"', '""', $tex_enigma_remark);
    if(stripos($tex_enigma_remark, ',') !== false){
        $tex_enigma_remark = '"'.$tex_enigma_remark.'"';
```

```
    }

    //Enigmas to remove (because they are no text, but a section of another text).

    $remove_array = array(4075, 4069, 4072, 4074, 3796, 3797, 3798, 3799, 3800,
3801, 3802, 3803, 3804, 3805, 3806, 3807, 4003, 4191, 4330, 4345, 132, 204, 289,
290, 291, 426, 643, 871, 886, 983, 1926, 2215, 2216, 2653, 2887, 2894, 2895, 4762,
4763, 4764, 4765, 4766, 4767, 5219, 5310, 6062, 2230);

    //Titles to remove (because they were invented or because they come from the
issue table).

    $remove_title_array = array(270, 3854, 4080, 604, 2652);

    //Italics: where to add italics to titles and first lines.

    //To titles.

    $tex_title_italics_array_year = array(1722, 1723, 1724, 1725, 1726, 1727,
1728, 1729, 1730, 1731, 1732, 1733, 1734, 1735, 1736, 1737, 1738, 1739, 1740,
1741, 1742, 1744, 1745, 1748, 1749, 1750, 1751, 1752, 1753, 1754, 1755, 1756,
1757, 1758, 1759, 1760, 1761, 1762, 1763, 1764, 1765, 1766, 1767, 1768, 1769,
1779, 1780, 1781, 1782, 1783, 1784, 1785, 1786, 1792, 1793, 1795, 1796, 1797,
1798); //The years of the enigmas concerned (YYYY).
    $tex_title_italics_array_year_month = array('1715-04', '1716-03', '1717-07',
'1718-09', '1718-10', '1718-11', '1718-12', '1719-01', '1719-02', '1719-03',
'1719-04', '1721-08', '1721-09', '1721-10', '1721-11', '1721-12', '1743-09',
'1743-10', '1743-11', '1743-12', '1747-03', '1747-04', '1747-05', '1747-06',
'1747-07', '1747-08', '1747-09', '1747-10', '1747-11', '1747-12', '1778-06',
'1778-07', '1778-08', '1778-09', '1778-10', '1778-11', '1778-12', '1787-01',
'1787-02', '1787-03', '1787-04', '1787-05', '1787-06', '1787-07', '1787-08',
'1787-09'); //The years and months of the enigmas concerned ('YYYY-MM').
    $tex_title_italics_array_date = array(); //The dates of the enigmas concerned
('YYYY-MM-DD').
    $tex_title_italics_array_date_number = array(); //The dates and volume numbers
of the enigmas concerned ('YYYY-MM-DD-N').
    $tex_title_italics_array_id = array(4071, 3937, 4133, 4176, 1049, 1052, 1054,
1055, 1057, 1060, 1061, 1063, 1064, 1066, 1067, 1182, 1184, 1191, 1193, 1198,
1202, 1203, 1205, 1207, 1214, 1215, 1216, 1222, 1227, 1228, 1231, 1234, 1238,
1243, 6820, 6821, 6822, 6823, 6824, 6825, 6826, 6827, 6828, 6829, 6830, 6831,
6832);

    //To first lines.

    $tex_incipit_italics_array_year = array(1677, 1678, 1679, 1680, 1683, 1684,
1685, 1686, 1687, 1688, 1689, 1690, 1691, 1692, 1693, 1694, 1695, 1696, 1697,
1698, 1699, 1700, 1701, 1702, 1703, 1704, 1705, 1706, 1707, 1708, 1709, 1710,
1711, 1713, 1714, 1716, 1720); //The years of the enigmas concerned (YYYY).
    $tex_incipit_italics_array_year_month = array('1681-02', '1681-03', '1681-04',
'1681-05', '1681-06', '1681-07', '1681-08', '1681-09', '1681-10', '1681-11',
'1681-12', '1682-02', '1682-03', '1682-04', '1682-05', '1682-06', '1682-08',
'1682-09', '1682-10', '1682-11', '1682-12', '1712-03', '1712-04', '1712-05',
'1712-06', '1712-07', '1712-08', '1712-09', '1712-10', '1712-11', '1712-12',
'1715-01', '1715-02', '1715-03', '1715-05', '1715-06', '1715-07', '1715-08',
'1715-09', '1715-10', '1715-11', '1715-12', '1717-05', '1717-06', '1717-08',
'1717-09', '1717-10', '1717-11', '1717-12', '1718-01', '1718-02', '1718-03',
'1718-04', '1718-05', '1718-06', '1718-07', '1718-08', '1719-05', '1719-06',
'1719-07', '1719-08', '1719-09', '1719-10', '1719-11', '1719-12', '1721-01',
```

```
'1721-02', '1721-03', '1721-04', '1721-05'); //The years and months of the enigmas
concerned ('YYYY-MM').
    $tex_incipit_italics_array_date = array(); //The dates of the enigmas
concerned ('YYYY-MM-DD').
    $tex_incipit_italics_array_date_number = array(); //The dates and volume
numbers of the enigmas concerned ('YYYY-MM-DD-N').
    $tex_incipit_italics_array_id = array(3615, 3616, 3651, 3652, 4099, 755, 756,
766, 761, 762, 770, 771, 772, 773, 774, 776, 790, 791, 792, 800, 809, 810, 818,
819, 824, 825, 829, 833, 834, 835, 840, 846, 853, 858, 862, 863, 869, 870, 875,
876, 879, 880, 883, 884, 885, 891, 895, 896, 897, 902, 906, 907, 910, 911, 916,
921, 924, 925, 926, 929, 934, 938, 951, 957, 977, 980, 1020, 1058, 1124, 1147,
1159, 1167, 1180, 1182, 1184, 1193, 1229, 1267, 1927, 2105, 2260, 2528, 2559,
2579, 2581, 2593, 2597);

    //Create field tex_title from titre.

    if(in_array($tex_enigma_id, $remove_title_array)){
    }else{

        $tex_title = $titre;

        if (in_array($year_check, $tex_title_italics_array_year) ||
in_array($year_month_check, $tex_title_italics_array_year_month) ||
in_array($date_check, $tex_title_italics_array_date) ||
in_array($date_number_check, $tex_title_italics_array_date_number) ||
in_array($id_check, $tex_title_italics_array_id)){
            $tex_title = '<em>'.$tex_title.'</em>';
        }

        if($year_check == 1770 || $year_check == 1771 || $year_check == 1772 ||
$year_check == 1773 || $year_check == 1774 || $year_check == 1775 || $year_check
== 1776 || $year_check == 1777 || ($year_check == 1778 && $month_check < 6)){
            if($titre == 'ÉNIGME' || $titre == 'ENIGME' || $titre == 'ÉNIGME.' ||
$titre == 'ENIGME.' || $titre == 'ÉNIGMES' || $titre == 'ENIGMES' || $titre ==
'ÉNIGMES.' || $titre == 'ENIGMES.' || $titre == 'LOGOGRYPHE' || $titre ==
'LOGOGRYPHE.' || $titre == 'LOGOGRYPHES' || $titre == 'LOGOGRYPHES.' || $titre ==
'LOGOGRIPHE' || $titre == 'LOGOGRIPHE.' || $titre == 'LOGOGRIPHES' || $titre ==
'LOGOGRIPHES.'){
            }else{
                $tex_title = '<em>'.$tex_title.'</em>';
            }
        }else{
        }

    }

    //Exceptions.

    if($id == 6861){
        $tex_title = "ÉNIGME. AIR : <em>La comédie est un miroir.</em>";
    }
    if($id == 6849){
        $tex_title = "ÉNIGME, A CHLOÉ. AIR : <em>Nous sommes précepteurs
d'amour.</em>";
    }
    if($id == 6844){
        $tex_title = "LOGOGRIPHE. AIR : <em>de la Croisée.</em>";
    }
    if($id == 6840){
```

```
        $tex_title = "ÉNIGME. AIR : <em>Je connais un berger discret.</em>";
    }
    if($id == 6836){
        $tex_title = "CHARADE QUI N'EN EST PAS UNE. / A ÉMILIE.";
    }
    if($id == 6685){
        $tex_title = "<em>CHARADE.</em> AIR : <em>Mon pere était pot</em>, étc.";
    }
    if($id == 6455){
        $tex_title = "<em>LOGOGRIPHE.</em> AIR : <em>Avec les jeux dans le
village.</em>";
    }
    if($id == 6399){
        $tex_title = "<em>CHARADE. AIR : Mon pere était pot, &c.</em>.";
    }
    if($id == 6391){
        $tex_title = "<em>ÉNIGME.</em> CHANSON sur l'Air du <em>Confiteor</em>.";
    }
    if($id == 6284){
        $tex_title = "LOGOGRIPHE <em>ANAGRAMMATIQUE.</em>";
    }
    if($id == 6278){
        $tex_title = "LOGOGRIPHE. <em>A Mlle. P.....</em>";
    }
    if($id == 6193){
        $tex_title = "ÉNIGME A COU. <em>Le Cou est la 2<sup>e</sup>. lettre du
mot.</em>";
    }
    if($id == 6178){
        $tex_title = "ÉNIGME. <em>ADRESSÉE A MÉLANIE.</em>";
    }
    if($id == 6161){
        $tex_title = "LOGOGRIPHE. <em>A une jeune Demoiselle de douze ans, qui en
demandait un à l'Auteur.</em>";
    }
    if($id == 6147){
        $tex_title = "A MLLE. DE M. DE LAUSANNE.";
    }
    if($id == 6146){
        $tex_title = "A MME. L... T... DE GENÊVE.";
    }
    if($id == 6145){
        $tex_title = "CHARADE, sur l'Air : <em>Vous qui d'amoureuse
aventure.</em>";
    }
    if($id == 6137){
        $tex_title = "ÉNIGME A FLANCS. <em>Par les</em> flancs, <em>on entend la
2e. & l'avant-derniere lettre du mot.</em>";
    }
    if($id == 6130){
        $tex_title = "CHARADE. <em>Conseil à Mademoiselle L..... D....</em>";
    }
    if($id == 6128){
        $tex_title = "ÉNIGME. <em>B.. J.. M. b.... S..., J. V... a. .. à. T... M..
C...</em>";
    }
    if($id == 6117){
        $tex_title = "LOGOGRIPHE. <em>A Mlle. ***.</em>";
    }
```

```
    if($id == 6115){
        $tex_title = "CHARADE. <em>A M. b.... S..., B.. J... d. T... M..
C...</em>";
    }
    if($id == 6105){
        $tex_title = "<em>Charade, Enigme & Logogriphe.</em>";
    }
    if($id == 6075){
        $tex_title = "CHARADE, <em>Conte.</em>";
    }
    if($id == 6063){
        $tex_title = "CHARADE. <em>CONTE.</em>";
    }
    if($id == 6029){
        $tex_title = "CHARADE. <em>A Mademoiselle N......</em>";
    }
    if($id == 6028){
        $tex_title = "LOGOGRIPHE. <em>A Mlle. L***.</em>";
    }
    if($id == 5978){
        $tex_title = "CHARADE. <em>A Mademoiselle de R........</em>";
    }
    if($id == 5968){
        $tex_title = "LOGOGRIPHE. <em>Le mot à M. l'Abbé R..., Vic. de B...</em>";
    }
    if($id == 5942){
        $tex_title = "CHARADE. <em>A Mademoiselle J*** de L******.</em>";
    }
    if($id == 5927){
        $tex_title = "CHARADE AIR : <em>Mon cher André.</em>";
    }
    if($id == 5861){
        $tex_title = "ÉNIGME <em>Proposée par des Pêcheurs à Homère.</em>";
    }
    if($id == 5812){
        $tex_title = "CHARADE. <em>A Madame</em> MOUTARD, <em>le jour de sa
Fête.</em>";
    }
    if($id == 5800){
        $tex_title = "CHARADE. <em>A Madame ***.</em>";
    }
    if($id == 5782){
        $tex_title = "CHARADE. Sur l'Air : <em>Coeurs sensibles.</em>";
    }
    if($id == 5770){
        $tex_title = "<em>A Mlle. de ***, qui avoit demandé à l'Auteur une
CHRADE.</em>";
    }
    if($id == 5764){
        $tex_title = "CHARADE <em>A Madame</em> DE LA MORINIÈRE, <em>qui m'en
demandoit une.</em>";
    }
    if($id == 5760){
        $tex_title = "LOGOGRIPHE <em>A Mlle. M....</em>";
    }
    if($id == 5733){
        $tex_title = "LOGOGRIPHE. A MADEMOISELLE DE B***.";
    }
    if($id == 5728){
```

```
        $tex_title = "CHARADE & LOGOGRIPHE.";
    }
    if($id == 5724){
        $tex_title = "LOGOGRIPHE. A MADAME B***.";
    }
    if($id == 5656){
        $tex_title = "CHARADE à M. <em>D...</em>";
    }
    if($id == 5631){
        $tex_title = "LOGOGRIPHE. <em>A Mlle. Sophie, P.... d'Ar....</em>";
    }
    if($id == 6905){
        $tex_title = "IV. <em>ENIGME en Bouts-rimés.</em>";
    }
    if($id == 5327){
        $tex_title = "<em>ÉNIGME.</em> Sur l'Air <em>du Confiteor.</em>";
    }
    if($id == 5172){
        $tex_title = "<em>ÉNIGME à corps, pour mettre entre les deux Enigmes à
tête & à queue des</em> N<sup>os</sup> 27 & 44.";
    }
    if($id == 5163){
        $tex_title = "<em>ÉNIGME à tête, pour répondre à l'Énigme à queue du</em>
N<sup>o</sup>. 27 <em>du Mercure.</em>";
    }
    if($id == 5138){
        $tex_title = "<em>CHARADE Faite au Château de Melleray, pour Mlle D.</em>
AIR : <em>Ah ! ma Voisine es tu fâchée ?....</em>";
    }
    if($id == 5126){
        $tex_title = "<em>CHARADE.</em> Sur l'Air <em>du Vaudeville de
Figaro.</em>";
    }
    if($id == 5090){
        $tex_title = "<em>CHARADE.</em> Sur l'Air : <em>Quoi ma Voisine es-tu
fâchée.</em>";
    }
    if($id == 5075){
        $tex_title = "<em>CHARADE.</em> Air : <em>N'est-il, Amour, sous ton
empire, &c.</em>";
    }
    if($id == 5054){
        $tex_title = "<em>CHARADE à Mademoiselle L. D...</em>";
    }
    if($id == 5053){
        $tex_title = "<em>LOGOGRYPHE Adressé à Madame DE LA VILLE, par M. S***,
Com.... de la Mar....</em>";
    }
    if($id == 5022){
        $tex_title = "<em>ÉNIGME à Mademoiselle ***.</em>";
    }
    if($id == 5000){
        $tex_title = "<em>CHARADE à Mlle ***.</em>";
    }
    if($id == 4996){
        $tex_title = "<em>LOGOGRYPHE A GRÉGOIRE.</em>";
    }
    if($id == 4986){
        $tex_title = "<em>CHARADE.</em> Air : <em>O ma tendre Musette.</em>";
```

```
    }
    if($id == 4977){
        $tex_title = "<em>AUTRE.</em>";
    }
    if($id == 4939){
        $tex_title = "<em>ÉNIGME à Madame du Q.....</em>";
    }
    if($id == 4973){
        $tex_title = "<em>CHARADE.</em> Sur l'Air : <em>Vous m'entendez
bien.</em>";
    }
    if($id == 4957){
        $tex_title = "<em>LOGOGRYPHE.</em> AIR : <em>Ce fut par la faute du
sort.</em>";
    }
    if($id == 4952){
        $tex_title = "<em>CHARADE.</em> AIR <em>de Marlbouroug.</em>";
    }
    if($id == 4939){
        $tex_title = "<em>CHARADE.</em> AIR : <em>O ma tendre Musette !</em>";
    }
    if($id == 4936){
        $tex_title = "III.";
    }
    if($id == 4935){
        $tex_title = "II.";
    }
    if($id == 4934){
        $tex_title = "I.";
    }
    if($id == 4931){
        $tex_title = "IV.";
    }
    if($id == 4930){
        $tex_title = "III.";
    }
    if($id == 4929){
        $tex_title = "II.";
    }
    if($id == 4928){
        $tex_title = "I.";
    }
    if($id == 4925){
        $tex_title = "II.";
    }
    if($id == 4924){
        $tex_title = "I.";
    }
    if($id == 4910){
        $tex_title = "III.";
    }
    if($id == 4909){
        $tex_title = "II.";
    }
    if($id == 4908){
        $tex_title = "I.";
    }
    if($id == 4904){
        $tex_title = "<em>CHARADE à Mlle P..... DE LA CH......</em>";
```

```
    }
    if($id == 4899){
        $tex_title = "V.";
    }
    if($id == 4898){
        $tex_title = "IV.";
    }
    if($id == 4897){
        $tex_title = "III.";
    }
    if($id == 4896){
        $tex_title = "II.";
    }
    if($id == 4895){
        $tex_title = "I.";
    }
    if($id == 4883){
        $tex_title = "<em>ÉNIGME à Mlle PA.... DE LA CHA....</em>";
    }
    if($id == 4882){
        $tex_title = "<em>CHARADE à Mlle ***.</em>";
    }
    if($id == 4876){
        $tex_title = "<em>AUTRE.</em>";
    }
    if($id == 4872){
        $tex_title = "<em>ÉNIGME OU LOGOGRYPHE.</em>";
    }
    if($id == 4870){
        $tex_title = "VI.";
    }
    if($id == 4869){
        $tex_title = "V.";
    }
    if($id == 4868){
        $tex_title = "IV.";
    }
    if($id == 4867){
        $tex_title = "III.";
    }
    if($id == 4866){
        $tex_title = "II.";
    }
    if($id == 4865){
        $tex_title = "I.";
    }
    if($id == 4833){
        $tex_title = "<em>ÉNIGME OU LOGOGRYPHE.</em>";
    }
    if($id == 4791){
        $tex_title = "IV.";
    }
    if($id == 4790){
        $tex_title = "III.";
    }
    if($id == 4789){
        $tex_title = "II.";
    }
    if($id == 4788){
```

```
        $tex_title = "I.";
    }
    if($id == 4701){
        $tex_title = "<em>LOGOGRYPHE, A PHILIS.</em>";
    }
    if($id == 4694){
        $tex_title = "<em>QUESTION ÉNIGMATIQUE.</em>";
    }
    if($id == 4547){
        $tex_title = "<em>LOGOGRYPHE A Madame DE MAD.**, à Guise.</em>";
    }
    if($id == 4412){
        $tex_title = "LOGOGRYPHE. <em>A Madame ***.</em>";
    }
    if($id == 3309){
        $tex_title = "<em>AUTRE. A Mademoiselle T.... RENARD.</em>EGLÉ.";
    }
    if($id == 3156){
        $tex_title = "<em>EPITRE LOGOGRYPHIQUE, à Mlle la C***, sous le nom
d'</em>EGLÉ.";
    }
    if($id == 3032){
        $tex_title = "<em>ÉNIGME GÉOMÉTRIQUE, par M. ***.</em>";
    }
    if($id == 3030){
        $tex_title = "<em>LOGOGRYPHE GRAMMAIRIEN.</em>";
    }
    if($id == 2923){
        $tex_title = "ÉNIGME <em>A Mademoiselle de ***.</em>";
    }
    if($id == 2904){
        $tex_title = "ÉNIGME <em>Traduite de l'Espagnol.</em>";
    }
    if($id == 2885){
        $tex_title = "LOGOGRYPHE LATIN.";
    }
    if($id == 2848){
        $tex_title = "<em>AUTRE ANGLOISE.</em>";
    }
    if($id == 2828){
        $tex_title = "<em>AUTRE A Mademoiselle REINE L* F***.</em>";
    }
    if($id == 2771){
        $tex_title = "ENIGME <em>Imitée de Rabelais.</em>";
    }
    if($id == 2761){
        $tex_title = "LOGOGRYPHE.";
    }
    if($id == 2756){
        $tex_title = "ÉNIGME.";
    }
    if($id == 2750){
        $tex_title = "LOGOGRYPHE.";
    }
    if($id == 2745){
        $tex_title = "ÉNIGME.";
    }
    if($id == 2742){
        $tex_title = "LOGOGRYPHE.";
```

```
    }
    if($id == 2738){
        $tex_title = "ÉNIGME.";
    }
    if($id == 2735){
        $tex_title = "LOGOGRYPHE.";
    }
    if($id == 2731){
        $tex_title = "ÉNIGME.";
    }
    if($id == 2728){
        $tex_title = "LOGOGRYPHE.";
    }
    if($id == 2724){
        $tex_title = "ÉNIGME.";
    }
    if($id == 2720){
        $tex_title = "LOGOGRYPHE. <em>Aux Dames.</em>";
    }
    if($id == 2716){
        $tex_title = "ÉNIGME.";
    }
    if($id == 2711){
        $tex_title = "LOGOGRYPHE. <em>A IRIS.</em>";
    }
    if($id == 2706){
        $tex_title = "ÉNIGME.";
    }
    if($id == 2703){
        $tex_title = "<em>AUTRE. A PHILIS.</em>";
    }
    if($id == 2702){
        $tex_title = "LOGOGRYPHE.";
    }
    if($id == 2697){
        $tex_title = "ÉNIGME.";
    }
    if($id == 2694){
        $tex_title = "LOGOGRYPHE.";
    }
    if($id == 2690){
        $tex_title = "ÉNIGME.";
    }
    if($id == 2687){
        $tex_title = "LOGOGRYPHE.";
    }
    if($id == 2683){
        $tex_title = "ÉNIGME.";
    }
    if($id == 2680){
        $tex_title = "LOGOGRYPHE.";
    }
    if($id == 2676){
        $tex_title = "ÉNIGME. <em>A Madame la Comtesse de R...</em>";
    }
    if($id == 2673){
        $tex_title = "LOGOGRYPHE.";
    }
    if($id == 2669){
```

```
        $tex_title = "ÉNIGME.";
    }
    if($id == 2664){
        $tex_title = "LOGOGRYPHE.";
    }
    if($id == 2659){
        $tex_title = "ÉNIGME ALLÉGORIQUE.";
    }
    if($id == 2657){
        $tex_title = "LOGOGRYPHE.";
    }
    if($id == 2651){
        $tex_title = "ÉNIGME.";
    }
    if($id == 2648){
        $tex_title = "LOGOGRYPHE.";
    }
    if($id == 2645){
        $tex_title = "ÉNIGME.";
    }
    if($id == 2641){
        $tex_title = "LOGOGRYPHE.";
    }
    if($id == 2637){
        $tex_title = "ÉNIGME.";
    }
    if($id == 2634){
        $tex_title = "LOGOGRYPHE.";
    }
    if($id == 2630){
        $tex_title = "ÉNIGME.";
    }
    if($id == 2626){
        $tex_title = "LOGOGRYPHE.";
    }
    if($id == 2622){
        $tex_title = "ÉNIGME.";
    }
    if($id == 2619){
        $tex_title = "LOGOGRYPHE.";
    }
    if($id == 2616){
        $tex_title = "ÉNIGME.";
    }
    if($id == 2613){
        $tex_title = "LOGOGRYPHE.";
    }
    if($id == 2609){
        $tex_title = "ÉNIGME.";
    }
    if($id == 2607){
        $tex_title = "LOGOGRYPHE.";
    }
    if($id == 2602){
        $tex_title = "LOGOGRYPHE. <em>A M. D... Conseiller en l'Election
d'Etampes.</em>";
    }
    if($id == 2599){
        $tex_title = "ÉNIGMES <em>A Mlle Cl... de la Gr..... à Rennes.</em>";
```

```
    }
    if($id == 2573){
        $tex_title = "<em>AUTRE à Mlle ***.</em>";
    }
    if($id == 2554){
        $tex_title = "<em>AUTRE.</em> Air : <em>Tous les bergers de
Chartres.</em>";
    }
    if($id == 2553){
        $tex_title = "<em>LOGOGRYPHE. A M. L***.</em> Air : <em>Ne vous laissez
jamais charmer, &c.</em>";
    }
    if($id == 2533){
        $tex_title = "<em>AUTRE.</em> Air : <em>Or dites-nous, Marie, &c.</em>";
    }
    if($id == 2531){
        $tex_title = "<em>AUTRE. A Mde DES C... par M. D. C. D. S. L.</em>";
    }
    if($id == 2527){
        $tex_title = "<em>ENIGME.</em> Air : <em>Réveillez-vous, belle endormie,
&c.</em>";
    }
    if($id == 2523){
        $tex_title = "<em>ENIGME.</em> Air <em>des folies d'Espagne.</em>";
    }
    if($id == 2518){
        $tex_title = "<em>AUTRE.</em> Air : <em>Par-tout où règne le chagrin,
&c.</em>";
    }
    if($id == 2517){
        $tex_title = "<em>LOGOGRYPHES.</em> Air : <em>Réveillez-vous, belle
endormie.</em>";
    }
    if($id == 2513){
        $tex_title = "<em>LOGOGRYPHE. A Mlle SYL... D'H... par M. D..</em>";
    }
    if($id == 2502){
        $tex_title = "<em>LOGOGRYPHES.</em> Air : <em>Ne v'la-t-il pas que
j'aime.</em>";
    }
    if($id == 2501){
        $tex_title = "<em>AUTRE.</em> Air : <em>Ton humeur est, Catherine,
&c.</em>";
    }
    if($id == 2500){
        $tex_title = "<em>ENIGMES.</em> Air : <em>Réveillez-vous, belle endormie,
&c.</em>";
    }
    if($id == 2497){
        $tex_title = "<em>AUTRE.</em> Air : <em>Ne v'la-t-il pas que
j'aime.</em>";
    }
    if($id == 2491){
        $tex_title = "<em>AUTRE.</em> Air : <em>Nous sommes précepteurs d'amour,
&c.</em>";
    }
    if($id == 2486){
        $tex_title = "<em>AUTRE. DÉDIÉ à des Dames qui s'étoient amusées quelques
jours de l'énigme & de la charade ou rébus.</em>";
```

```
    }
    if($id == 2476){
        $tex_title = "<em>AUTRE.</em> Air : <em>Ne v'la-t-il pas que
j'aime.</em>";
    }
    if($id == 2482){
        $tex_title = "<em>AUTRE à Madame .....</em>";
    }
    if($id == 2477){
        $tex_title = "<em>LOGOGRYPHES. A M. l'Abbé DE V...</em>";
    }
    if($id == 2476){
        $tex_title = "<em>AUTRE.</em> Air : <em>Tous les Bourgeois de
Chartres.</em>";
    }
    if($id == 2470){
        $tex_title = "<em>AUTRE.</em> Air : <em>Tous les Bergers de
Chartres.</em>";
    }
    if($id == 2466){
        $tex_title = "<em>AUTRE. A M. FABRE, à Strasbourg.</em>";
    }
    if($id == 2455){
        $tex_title = "<em>ENIGMES.</em> Air : <em>Réveillez-vous belle, &c.</em>";
    }
    if($id == 2447){
        $tex_title = "<em>PROVERBES en Énigme.</em>";
    }
    if($id == 2392){
        $tex_title = "A PHRYNÉ. <em>LOGOGRYPHE.</em>";
    }
    if($id == 2385){
        $tex_title = "<em>AUTRE.</em> A IRIS.";
    }
    if($id == 2369){
        $tex_title = "<em>ENIGME, par Madame la Marquise DE MONTBAZIN.</em>";
    }
    if($id == 2171){
        $tex_title = "<em>LOGOGRYPHE.</em> A <em>Mlle A... C... Q... J...</em>";
    }
    if($id == 2158){
        $tex_title = "<em>LOGOGRYPHE. A M. CHALUMEAU de S. Domingue, Auteur de
celui sur le mot</em> Réponse, <em>qu'il m'a adressé dans le Mercure de Juin
dernier.</em>";
    }
    if($id == 1998){
        $tex_title = "<em>ENIGME, EN CHANSON.</em> Air : <em>du Confiteor.</em>";
    }
    if($id == 1927){
        $tex_title = "LOGOGRYPHUS.";
    }
    if($id == 1896){
        $tex_title = "LOGOGRYPHE <em>A Madame de M***.</em>";
    }
    if($id == 1893){
        $tex_title = "ENIGME <em>A Madame de B***.</em>";
    }
    if($id == 1862){
        $tex_title = "ENIGME <em>A Madame le B....</em>";
```

```
    }
    if($id == 1827){
        $tex_title = "<em>LOGOGRYPHE En Sonnet, par B. D. B.</em>";
    }
    if($id == 1796){
        $tex_title = "ENIGME.";
    }
    if($id == 1794){
        $tex_title = "ENIGME.";
    }
    if($id == 1793){
        $tex_title = "<em>LOGOGRYPHE.</em> PAR P. L. F. PHILAGATHE, <em>de
Dunkerque.</em>";
    }
    if($id == 1788){
        $tex_title = "ENIGME.";
    }
    if($id == 1822){
        $tex_title = "ENIGME.";
    }
    if($id == 1787){
        $tex_title = "LOGOGRYPHE.";
    }
    if($id == 1786){
        $tex_title = "ENIGME.";
    }
    if($id == 1785){
        $tex_title = "<em>LOGOGRYPHE EN CHANSON.</em> Sur l'air : <em>De tous les
Capucins du monde.</em>";
    }
    if($id == 1780){
        $tex_title = "ENIGME.";
    }
    if($id == 1777){
        $tex_title = "ENIGME.";
    }
    if($id == 1774){
        $tex_title = "ENIGME.";
    }
    if($id == 1772){
        $tex_title = "<em>AUTRE.</em> Air : <em>Ne v'là t-il pas que
j'aime.</em>";
    }
    if($id == 1771){
        $tex_title = "AUTRE. Air : <em>V'là c'que c'est qu'd'aller au bois.</em>";
    }
    if($id == 1770){
        $tex_title = "ENIGMES EN VAUDEVILLES. Air : <em>Babet, que t'es
gentille.</em>";
    }
    if($id == 1768){
        $tex_title = "AUTRE Air <em>du Menuet d'Exaudet.</em>";
    }
    if($id == 1767){
        $tex_title = "ENIGMES EN VAUDEVILLES. Air : <em>A notre bonheur l'Amour
préside.</em>";
    }
    if($id == 1765){
        $tex_title = "ENIGME.";
```

```
    }
    if($id == 1763){
        $tex_title = "LOGOGRYPHE.";
    }
    if($id == 1759){
        $tex_title = "ENIGME.";
    }
    if($id == 1757){
        $tex_title = "LOGOGRYPHE.";
    }
    if($id == 1756){
        $tex_title = "ENIGME.";
    }
    if($id == 1754){
        $tex_title = "LOGOGRYPHE.";
    }
    if($id == 1753){
        $tex_title = "ENIGME.";
    }
    if($id == 1746){
        $tex_title = "<em>LOGOGRYPHE EN VAUDEVILLES.</em> Air : <em>De tous les
Capucins du monde.</em>";
    }
    if($id == 1737){
        $tex_title = "<em>ENIGME EN VAUDEVILLES.</em> Air : <em>On n'entend plus
dessous l'ormeau, &c.</em>";
    }
    if($id == 1733){
        $tex_title = "<em>AUTRE EN VAUDEVILLES.</em> Air : <em>Nous sommes
Précepteurs d'amour.</em>";
    }
    if($id == 1697){
        $tex_title = "AUTRE.";
    }
    if($id == 1680){
        $tex_title = "ESSAI <em>D'une espéce nouvelle d'Enigmes plus utiles que
celles qu'on lit ordinairement.</em>";
    }
    if($id == 1605){
        $tex_title = "STANCES LOGOGRIPHIQUES.";
    }
    if($id == 1369){
        $tex_title = "<em>LOGOGRYPHE. Par M. G*** de la Rochelle.</em>";
    }
    if($id == 1308){
        $tex_title = "<em>ENIGME. Par Madame des Forges Maillard.</em>";
    }
    if($id == 1291){
        $tex_title = "LOGOGRYPHUS.";
    }
    if($id == 1275){
        $tex_title = "LOGOGRYPHUS.";
    }
    if($id == 1267){
        $tex_title = "ENIGMA.";
    }
    if($id == 1235){
        $tex_title = "ENIGME <em>Par M. Desforges Maillard, Associé de l'Académie
des Belles Lettres de la Rochelle.</em>";
```

```
    }
    if($id == 1178){
        $tex_title = "AUTRE.";
    }
    if($id == 1177){
        $tex_title = "LOGOGRYPHE.";
    }
    if($id == 1175){
        $tex_title = "AUTRE. <em>Dans lequel on ne fait point entrer la voyelle
A.</em>";
    }
    if($id == 1174){
        $tex_title = "LOGOGRYPHE.";
    }
    if($id == 1173){
        $tex_title = "ENIGME.";
    }
    if($id == 1172){
        $tex_title = "AUTRE.";
    }
    if($id == 1171){
        $tex_title = "AUTRE.";
    }
    if($id == 1169){
        $tex_title = "AUTRE.";
    }
    if($id == 1168){
        $tex_title = "LOGOGRYPHE.";
    }
    if($id == 1166){
        $tex_title = "ENIGME.";
    }
    if($id == 1165){
        $tex_title = "AUTRE.";
    }
    if($id == 1164){
        $tex_title = "LOGOGRYPHE.";
    }
    if($id == 1163){
        $tex_title = "LOGOGRYPHE.";
    }
    if($id == 1160){
        $tex_title = "ENIGME.";
    }
    if($id == 1156){
        $tex_title = "LOGOGRYPHE.";
    }
    if($id == 1155){
        $tex_title = "ENIGME.";
    }
    if($id == 1153){
        $tex_title = "LOGOGRYPHE.";
    }
    if($id == 1152){
        $tex_title = "ENIGME.";
    }
    if($id == 1147){
        $tex_title = "ENIGMA.";
    }
```

```
    if($id == 1146){
        $tex_title = "ENIGME.";
    }
    if($id == 1143){
        $tex_title = "ENIGME.";
    }
    if($id == 1142){
        $tex_title = "LOGOGRYPHE.";
    }
    if($id == 1135){
        $tex_title = "LOGOGRYPHE.";
    }
    if($id == 1134){
        $tex_title = "ODE ENIGMATIQUE. <em>par le même.</em>";
    }
    if($id == 1133){
        $tex_title = "ENIGME EN LOGOGRYPHE. <em>Par M. Jacques</em>";
    }
    if($id == 1132){
        $tex_title = "LOGOGRYPHE.";
    }
    if($id == 1131){
        $tex_title = "ENIGME.";
    }
    if($id == 1124){
        $tex_title = "ENIGMA.";
    }
    if($id == 1044){
        $tex_title = "LOGOGRYPHE.";
    }
    if($id == 1043){
        $tex_title = "ENIGME.";
    }
    if($id == 1042){
        $tex_title = "LOGOGRYPHE.";
    }
    if($id == 1041){
        $tex_title = "ENIGME.";
    }
    if($id == 1039){
        $tex_title = "LOGOGRYPHE.";
    }
    if($id == 1038){
        $tex_title = "ENIGME.";
    }
    if($id == 1020){
        $tex_title = "LOGOGRYPHUS.";
    }
    if($id == 1013){
        $tex_title = "<em>AUTRE Logogryphe en Vaudevilles.</em> Air : <em>A
l'ombre de ce verd Bocage.</em>";
    }
    if($id == 1002){
        $tex_title = "<em>LOGOGRYPHE,</em> Qu'on peut chanter sur l'Air :
<em>Réveillez-vous, Belle endormie.</em>";
    }
    if($id == 982){
        $tex_title = "LOGOGRYPHE.";
    }
```

```
    if($id == 979){
        $tex_title = "LOGOGRYPHE.";
    }
    if($id == 978){
        $tex_title = "ENIGME.";
    }
    if($id == 977){
        $tex_title = "LOGOGRYPHUS.";
    }
    if($id == 976){
        $tex_title = "LOGOGRYPHE.";
    }
    if($id == 975){
        $tex_title = "ENIGME.";
    }
    if($id == 974){
        $tex_title = "LOGOGRYPHE.";
    }
    if($id == 971){
        $tex_title = "LOGOGRYPHE <em>A Mlle L....</em>";
    }
    if($id == 957){
        $tex_title = "LOGOGRYPHUS.";
    }
    if($id == 951){
        $tex_title = "LOGOGRYPHUS.";
    }
    if($id == 942){
        $tex_title = "ENIGME.";
    }
    if($id == 939){
        $tex_title = "ENIGME.";
    }
    if($id == 938){
        $tex_title = "LOGOGRYPHUS.";
    }
    if($id == 935){
        $tex_title = "ENIGME.";
    }
    if($id == 934){
        $tex_title = "LOGOGRYPHUS.";
    }
    if($id == 929){
        $tex_title = "LOGOGRYPHUS.";
    }
    if($id == 926){
        $tex_title = "ALIUS.";
    }
    if($id == 925){
        $tex_title = "ALIUS.";
    }
    if($id == 924){
        $tex_title = "LOGOGRYPHUS.";
    }
    if($id == 922){
        $tex_title = "ENIGME.";
    }
    if($id == 921){
        $tex_title = "LOGOGRYPHUS.";
```

```
    }
    if($id == 917){
        $tex_title = "ENIGME.";
    }
    if($id == 916){
        $tex_title = "LOGOGRYPHUS.";
    }
    if($id == 915){
        $tex_title = "<em>III. ENVOI.</em>";
    }
    if($id == 914){
        $tex_title = "<em>II. ENVOI.</em>";
    }
    if($id == 913){
        $tex_title = "LOGOGRYPHES <em>Adressés à Mlle ***. par A. X. Harduin
d'Arras. I. ENVOI.</em>";
    }
    if($id == 911){
        $tex_title = "ALIUS.";
    }
    if($id == 910){
        $tex_title = "LOGOGRYPHUS.";
    }
    if($id == 909){
        $tex_title = "LOGOGRYPHE.";
    }
    if($id == 908){
        $tex_title = "ENIGME.";
    }
    if($id == 907){
        $tex_title = "ALIUS.";
    }
    if($id == 906){
        $tex_title = "LOGOGRYPHUS.";
    }
    if($id == 903){
        $tex_title = "ENIGME.";
    }
    if($id == 902){
        $tex_title = "LOGOGRYPHUS.";
    }
    if($id == 898){
        $tex_title = "ENIGME.";
    }
    if($id == 897){
        $tex_title = "ALIUS.";
    }
    if($id == 896){
        $tex_title = "ALIUS.";
    }
    if($id == 895){
        $tex_title = "LOGOGRYPHUS.";
    }
    if($id == 891){
        $tex_title = "LOGOGRYPHUS.";
    }
    if($id == 887){
        $tex_title = "ENIGME.";
    }
```

```
    if($id == 885){
        $tex_title = "ALIUS.";
    }
    if($id == 884){
        $tex_title = "ALIUS.";
    }
    if($id == 883){
        $tex_title = "LOGOGRYPHUS.";
    }
    if($id == 881){
        $tex_title = "ENIGME.";
    }
    if($id == 880){
        $tex_title = "ALIUS.";
    }
    if($id == 879){
        $tex_title = "LOGOGRYPHUS.";
    }
    if($id == 876){
        $tex_title = "ALIUS.";
    }
    if($id == 875){
        $tex_title = "LOGOGRYPHUS.";
    }
    if($id == 872){
        $tex_title = "ENIGME.";
    }
    if($id == 870){
        $tex_title = "ALIUS.";
    }
    if($id == 869){
        $tex_title = "LOGOGRYPHUS.";
    }
    if($id == 865){
        $tex_title = "MCCCCXLIX.";
    }
    if($id == 864){
        $tex_title = "<em>DEUX LOGOGRYPHES Cronographiques, tirés d'un Manuscrit
des Poësies de</em> Jean Michel, <em>& autres Poëtes du XVI. siecle, N<sup>o</sup>
8060. de la Bibliotheque du Roy, F. 47.</em> MCCCCXL.";
    }
    if($id == 863){
        $tex_title = "ALIUS.";
    }
    if($id == 862){
        $tex_title = "LOGOGRYPHUS.";
    }
    if($id == 859){
        $tex_title = "ENIGME.";
    }
    if($id == 858){
        $tex_title = "LOGOGRYPHUS.";
    }
    if($id == 855){
        $tex_title = "LOGOGRYPHE.";
    }
    if($id == 854){
        $tex_title = "ENIGME.";
    }
```

```
    if($id == 853){
        $tex_title = "LOGOGRYPHUS.";
    }
    if($id == 846){
        $tex_title = "LOGOGRYPHUS.";
    }
    if($id == 840){
        $tex_title = "LOGOGRYPHUS.";
    }
    if($id == 835){
        $tex_title = "ALIUS.";
    }
    if($id == 834){
        $tex_title = "ALIUS.";
    }
    if($id == 833){
        $tex_title = "LOGOGRYPHUS.";
    }
    if($id == 829){
        $tex_title = "ENIGMA CUM LOGOGRYPHO.";
    }
    if($id == 825){
        $tex_title = "ALIUS.";
    }
    if($id == 824){
        $tex_title = "LOGOGRYPHUS.";
    }
    if($id == 820){
        $tex_title = "ENIGME.";
    }
    if($id == 819){
        $tex_title = "ALIUS.";
    }
    if($id == 818){
        $tex_title = "LOGOGRYPHUS.";
    }
    if($id == 812){
        $tex_title = "LOGOGRYPHE <em>A Mad. R....</em>";
    }
    if($id == 810){
        $tex_title = "ALIUS.";
    }
    if($id == 809){
        $tex_title = "LOGOGRYPHUS.";
    }
    if($id == 801){
        $tex_title = "ENIGME.";
    }
    if($id == 800){
        $tex_title = "LOGOGRYPHUS.";
    }
    if($id == 793){
        $tex_title = "ENIGME.";
    }
    if($id == 792){
        $tex_title = "ALIUS.";
    }
    if($id == 791){
        $tex_title = "ALIUS.";
```

```
    }
    if($id == 790){
        $tex_title = "LOGOGRYPHUS.";
    }
    if($id == 777){
        $tex_title = "ENIGME.";
    }
    if($id == 770){
        $tex_title = "LOGOGRYPHUS.";
    }
    if($id == 771){
        $tex_title = "ALIUS.";
    }
    if($id == 772){
        $tex_title = "ALIUS.";
    }
    if($id == 773){
        $tex_title = "ALIUS.";
    }
    if($id == 774){
        $tex_title = "ALIUS.";
    }
    if($id == 776){
        $tex_title = "ALIUS.";
    }
    if($id == 765){
        $tex_title = "ENIGME, <em>Demandée à l'Auteur par Mlle M. T...</em>";
    }
    if($id == 761){
        $tex_title = "ALTER.";
    }
    if($id == 762){
        $tex_title = "ALTER.";
    }
    if($id == 766){
        $tex_title = "LOGOGRYPHUS.";
    }
    if($id == 755){
        $tex_title = "LOGOGRYPHUS.";
    }
    if($id == 756){
        $tex_title = "ALIUS.";
    }
    if($id == 4089){
        $tex_title = "ENIGME par M<sup>r</sup> de Longueüil, de l'Academie Royale
d'Angers. SONNET.";
    }
    if($id == 4091){
        $tex_title = "ENIGME, <em>Par M<sup>r</sup> L***</em>";
    }
    if($id == 4093){
        $tex_title = "ENIGME par M. D. L. P.";
    }
    if($id == 4082){
        $tex_title = "ENIGME par M<sup>r</sup> de Souv....";
    }
    if($id == 4104){
        $tex_title = "AUTRE ENIGME. <em>Par Mr</em> MARTIN.";
    }
```

```
    if($id == 4112){
        $tex_title = "ENIGME. par l'Arch... Turpin.";
    }
    if($id == 4124){
        $tex_title = "<em>ENIGME ANCIENNE</em> de celles qu'on nommoit autrefois
<em>Logo grife,</em> on donne le mot avec l'Enigme, elle ne laisse pas pour cela
d'estre difficile à expliquer ; le mot de celle-cy est l'<em>Orange</em>.";
    }
    if($id == 4130){
        $tex_title = "ENIGME. <em>Par Monsieur le D...</em>.";
    }
    if($id == 4155){
        $tex_title = "AUTRE Du bel emplastre, qui fait plus de vers qu'un autre
n'en peut lire.";
    }
    if($id == 4185){
        $tex_title = "ENIGME de M. T. R. E. T.";
    }
    if($id == 4186){
        $tex_title = "AUTRE du Prisonnier volontaire.";
    }
    if($id == 4196){
        $tex_title = "ENIGME de l'Abbé Insulaire.";
    }
    if($id == 4197){
        $tex_title = "AUTRE du Solitaire malgré luy.";
    }
    if($id == 4200){
        $tex_title = "ENIGME de M. l'Abbé Tripié.";
    }
    if($id == 4203){
        $tex_title = "ENIGME Du Solitaire malgré lui de l'Isle Saint Loüis.";
    }
    if($id == 4204){
        $tex_title = "AUTRE De l'Abbé Insulaire.";
    }
    if($id == 4206){
        $tex_title = "AUTRE ENIGME Du fidele Berger de la Nymphe insulaire.";
    }
    if($id == 4207){
        $tex_title = "ENIGME Du Solitaire malgré lui de l'Isle S. Loüis.";
    }
    if($id == 4214){
        $tex_title = "AUTRE, Du Solitaire, malgré lui, de l'Isle S. Loüis.";
    }
    if($id == 4217){
        $tex_title = "<em>ENIGME Proposée par M. le Comte de Gilin, beau jeune
Vieillard de deux Lustres.</em>";
    }
    if($id == 4220){
        $tex_title = "AUTRE DU SOLITAIRE MALGRÉ LUI DE L'ISLE SAINT LOÜIS.";
    }
    if($id == 4222){
        $tex_title = "AUTRE DE M<sup>r</sup> LAUVIN.";
    }
    if($id == 4223){
        $tex_title = "<em>ENIGME. Par M. le Comte de S. Gilin</em>";
    }
    if($id == 4223){
```

```
        $tex_title = "AUTRE, DE MONSIEUR LAUVIN.";
    }
    if($id == 4235){
        $tex_title = "ENIGME, PAR M. LAUVIN.";
    }
    if($id == 4236){
        $tex_title = "AUTRE, PAR M. DE PILES.";
    }
    if($id == 4249){
        $tex_title = "ENIGME. Par M. CHANSIERGES.";
    }
    if($id == 4253){
        $tex_title = "<em>ENIGME. Par M. Constant, Ingenieur des menus plaisirs de
Sa Majesté.</em>";
    }
    if($id == 4256){
        $tex_title = "<em>AUTRE. Par</em> M. D. <em>O. L. V.</em>";
    }
    if($id == 4259){
        $tex_title = "<em>ENIGME De M. le Marquis de Neuvie.</em>";
    }
    if($id == 4261){
        $tex_title = "<em>ENIGME Par M. l'Abbé de Soüillac.</em>";
    }
    if($id == 4273){
        $tex_title = "ENIGME. <em>Par Madame la Comtesse d'Apr.</em>";
    }
    if($id == 4277){
        $tex_title = "<em>ENIGME.</em> Par M. Dumont, de Montbrisson en Forest.";
    }
    if($id == 4278){
        $tex_title = "AUTRE. Par le même.";
    }
    if($id == 4289){
        $tex_title = "ENIGME Par le même.";
    }
    if($id == 4290){
        $tex_title = "AUTRE Par M. de la T.";
    }
    if($id == 4295){
        $tex_title = "ENIGME. Par la belle Muse R. Bessiral.";
    }
    if($id == 4296){
        $tex_title = "AUTRE. Par M. B....";
    }
    if($id == 4302){
        $tex_title = "AUTRE. Par la Muse Prisonniere.";
    }
    if($id == 4373){
        $tex_title = "PREMIERE ENIGME. Par le P. P. M***";
    }
    if($id == 95){
        $tex_title = "<em>ENIGME Du même Auteur.</em>";
    }
    if($id == 99){
        $tex_title = "<em>PREMIERE ENIGME. Par M. ***</em>";
    }
    if($id == 100){
        $tex_title = "<em>SECONDE ENIGME. Du même.</em>";
```

```
    }
    if($id == 116){
        $tex_title = "NOUVELLES ENIGMES. <em>Premiere.</em>";
    }
    if($id == 116){
        $tex_title = "<em>Énigme sur l'Air</em>: Réveillez-vous, belle endormie.";
    }
    if($id == 160){
        $tex_title = "<em>ENIGME. Par M. F. A.</em>";
    }
    if($id == 193){
        $tex_title = "<em>DEUXIÈME ENIGME. Il y a un prix proposé pour ceux qui la
devineront.</em>";
    }
    if($id == 218){
        $tex_title = "LOGOGRYPHE.";
    }
    if($id == 236){
        $tex_title = "LOGOGRYPHE.";
    }
    if($id == 237){
        $tex_title = "LOGOGRYPHE.";
    }
    if($id == 244){
        $tex_title = "II.";
    }
    if($id == 245){
        $tex_title = "III.";
    }
    if($id == 246){
        $tex_title = "IV.";
    }
    if($id == 248){
        $tex_title = "II.";
    }
    if($id == 249){
        $tex_title = "III.";
    }
    if($id == 250){
        $tex_title = "IV.";
    }
    if($id == 269){
        $tex_title = "ENIGME.";
    }
    if($id == 281){
        $tex_title = "ENIGME sur l'air : <em>Reveillez-vous, Belle
endormie.</em>";
    }
    if($id == 281){
        $tex_title = "LE LOGOGRYPHE. <em>A Madame la Comtesse de B***</em>";
    }
    if($id == 281){
        $tex_title = "<em>LOGOGRYPHE à Madame de G. Par M. du P. d'Arras.</em>";
    }
    if($id == 398){
        $tex_title = "LOGOGRYPHE LATIN.";
    }
    if($id == 419){
        $tex_title = "LOGOGRYPHUS.";
```

```
    }
    if($id == 552){
        $tex_title = "<em>AUTRE Logogryphe proposé à S. A. S. Madame la Duchesse
du Maine, par M.</em> Don Cl. Hernandez Delvalle, <em>jeune Espagnol fort connu
dans le monde par son esprit.</em>";
    }
    if($id == 602){
        $tex_title = "<em>LOGOGRYPHE. A Mlle Diga. Par M. D. V.**.</em>";
    }
    if($id == 651){
        $tex_title = "ENIGMA.";
    }
    if($id == 669){
        $tex_title = "LOGOGRYPHUS.";
    }
    if($id == 670){
        $tex_title = "ALIUS.";
    }
    if($id == 695){
        $tex_title = "SONNET ENIGMATIQUE.";
    }
    if($id == 721){
        $tex_title = "LOGOGRYPHUS.";
    }
    if($id == 724){
        $tex_title = "AUTRE.";
    }
    if($id == 725){
        $tex_title = "AUTRE.";
    }
    if($id == 726){
        $tex_title = "SONNET ENIGMATIQUE. <em>Dont les Rimes ont été données dans
le Mercure.</em>";
    }
    if($id == 730){
        $tex_title = "LOGOGRYPHUS.";
    }
    if($id == 733){
        $tex_title = "AUTRE.";
    }
    if($id == 734){
        $tex_title = "AUTRE.";
    }
    if($id == 735){
        $tex_title = "LOGOGRYPHUS.";
    }
    if($id == 736){
        $tex_title = "ALTER.";
    }
    if($id == 741){
        $tex_title = "LOGOGRYPHUS.";
    }
    if($id == 746){
        $tex_title = "LOGOGRYPHUS.";
    }
    if($id == 751){
        $tex_title = "LOGOGRYPHUS.";
    }
```

```
    //Manage rich text field.

    if($tex_title != ''){

        $tex_title = preg_replace(
            '/"(\S)/',
            '« $1',
            $tex_title
        );

        $tex_title = preg_replace(
            '/(\S)"/',
            '$1 »',
            $tex_title
        );

        $tex_title = str_replace("’", "'", $tex_title);
        $tex_title = str_replace("&rsquo;", "'", $tex_title);
        $tex_title = '"{ format: full_html, value: ""<p>'.$tex_title.'</p>"" }"';
        $tex_title = htmlspecialchars($tex_title, ENT_QUOTES, 'UTF-8');
    }

    //Create field tex_title_table. (Remains empty.)

    $tex_title_table = '';

    //Create field tex_rec_article_short_title from titre_normalise.

    $tex_rec_article_short_title_enigma_array_id =
array(102,112,116,117,118,141,178,244,245,246,281,361,426,578,718,765,1071,1161,11
76,1680,2069,2240,2369,2417,2448,2456,2476,2484,2497,2501,2533,2612,2828,2848,3854
,4070,4071,4074,4080,4124,4133,4178,4182,4226,4254,4833,4872,5010,6391,6849,6905);
    $tex_rec_article_short_title_logogriph_array_id =
array(248,249,250,280,311,312,440,464,503,505,552,579,596,599,602,604,684,727,843,
844,864,865,994,1002,1175,1423,1643,2102,2154,2200,2201,2218,2224,2251,2321,2376,2
377,2378,2379,2380,2385,2392,2406,2407,2408,2409,2410,2466,2470,2482,2486,2491,251
8,2531,2535,2554,2573,2578,2554,2573,2703,2744,2872,4964);
    $tex_rec_article_short_title_charade_array_id =
array(4789,4790,4791,4866,4867,4868,4869,4870,4896,4897,4898,4899,4909,4910,4925,4
929,4930,4931,4935,4936,5138,5728,6105,6118);
    $tex_rec_article_short_title_enigma_vaudeville_array_id =
array(1768,1771,1772);

    if(in_array($id, $tex_rec_article_short_title_enigma_array_id) ||
($titre_normalise == 'Autre' && $genre_1 == 'Énigme' && $genre_2 == '') ||
(stripos($titre_normalise, 'Énigme à M') !== false) || (stripos($titre_normalise,
'Énigme à Iris') !== false) || (stripos($titre_normalise, 'Énigme à Philis') !==
false) || (stripos($titre_normalise, 'Énigme adressée à M') !== false) ||
((stripos($titre_normalise, 'Énigme') !== false) && (stripos($titre_normalise,
'Logogriphe') === false) && (stripos($titre_normalise, 'Charade') === false) &&
$genre_1 == 'Énigme' && $genre_2 == '') || ((stripos($titre_normalise, 'Autre')
!== false) && (stripos($titre_normalise, 'Logogriphe') === false) &&
(stripos($titre_normalise, 'Charade') === false) && $genre_1 == 'Énigme' &&
$genre_2 == '') || (stripos($titre_normalise, 'Fable énigmatique') !== false) ||
(stripos($titre_normalise, 'Feste de Chantilly') !== false) || $titre_normalise ==
'Énigme' || $titre_normalise == 'Autre énigme' || $titre_normalise == 'Première
énigme' || $titre_normalise == 'I. Énigme' || $titre_normalise == 'II. Énigme' ||
$titre_normalise == 'Seconde énigme' || $titre_normalise == 'Deuxième énigme' ||
$titre_normalise == 'Troisième énigme' || $titre_normalise == 'Quatrième énigme'
```

```
|| $titre_normalise == 'Cinquième énigme' || $titre_normalise == 'Sixième énigme'
|| $titre_normalise == 'Septième énigme' || $titre_normalise == 'Huitième énigme'
|| $titre_normalise == 'Neuvième énigme' || $titre_normalise == 'Dixième énigme'
|| $titre_normalise == 'Onzième énigme' || $titre_normalise == 'Douzième énigme'
|| $titre_normalise == 'Énigme, bouts-rimés et parodie' || $titre_normalise ==
'Autre énigme d\'un auteur moderne' || $titre_normalise == 'Énigme du même auteur'
|| $titre_normalise == 'Seconde énigme du même auteur' || $titre_normalise ==
'Quatrain énigmatique' || $titre_normalise == 'Énigme. Sonnet' || $titre_normalise
== 'Ode énigmatique' || $titre_normalise == 'Énigme irrégulière' ||
$titre_normalise == 'Vers énigmatiques' || $titre_normalise == 'Énigme en chanson'
|| $titre_normalise == 'Proverbes en énigme' || $titre_normalise == 'Épitaphe
énigmatique' || $titre_normalise == 'Énigme allégorique' || $titre_normalise ==
'Énigme imitée de Rabelais' || $titre_normalise == 'Énigme géométrique' ||
$titre_normalise == 'Énigme burlesque' || $titre_normalise == 'Rondeau
énigmatique' || $titre_normalise == 'Question énigmatique' || $titre_normalise ==
'Énigme en charade' || $titre_normalise == 'Énigme en rondeau'){
        $tex_rec_article_short_title = 13754; //Value matching "Énigme" in the
"Article récurrent" taxonomy.
    }
    elseif(in_array($id, $tex_rec_article_short_title_logogriph_array_id) ||
($titre_normalise == 'Autre' && $genre_1 == 'Logogriphe' && $genre_2 == '') ||
(stripos($titre_normalise, 'Logogriphe à M') !== false) ||
(stripos($titre_normalise, 'Logogryphe à M') !== false) ||
(stripos($titre_normalise, 'Logogriphe à Iris') !== false) ||
(stripos($titre_normalise, 'Logogriphe. À Iris') !== false) ||
(stripos($titre_normalise, 'Logogriphe à Philis') !== false) ||
(stripos($titre_normalise, 'Logogriphe adressé à M') !== false) ||
((stripos($titre_normalise, 'Logogriphe') !== false) && (stripos($titre_normalise,
'Énigme') === false) && (stripos($titre_normalise, 'Charade') === false) &&
$genre_1 == 'Logogriphe' && $genre_2 == '') || ((stripos($titre_normalise,
'Autre') !== false) && (stripos($titre_normalise, 'Énigme') === false) &&
(stripos($titre_normalise, 'Charade') === false) && $genre_1 == 'Logogriphe' &&
$genre_2 == '') || (stripos($titre_normalise, 'Épitre logogriphique') !== false)
|| $titre_normalise == 'Logogriphe' || $titre_normalise == 'Autre logogriphe' ||
$titre_normalise == 'Premier logogriphe' || $titre_normalise == 'Second
logogriphe' || $titre_normalise == 'Deuxième logogriphe' || $titre_normalise ==
'Troisième logogriphe' || $titre_normalise == 'Quatrième logogriphe' ||
$titre_normalise == 'Cinquième logogriphe' || $titre_normalise == 'Sixième
logogriphe' || $titre_normalise == 'Logogriphe par Mademoiselle ***' ||
$titre_normalise == 'Nouveau logogriphe' || $titre_normalise == 'Logogriphe latin'
|| $titre_normalise == 'Logogriphe en sonnet' || $titre_normalise == 'Logogriphe
renversé' || $titre_normalise == 'Stances logogriphiques' || $titre_normalise ==
'Sonnet en logogriphe' || $titre_normalise == 'Logogriphe en chanson' ||
$titre_normalise == 'Logogriphe arithmétique' || $titre_normalise == 'Anagramme
logogriphique' || $titre_normalise == 'Logogriphe aux dames' || $titre_normalise
== 'Logogriphe grammairien' || $titre_normalise == 'Logogriphe en stances' ||
$titre_normalise == 'Logogriphe anagrammatique'){
        $tex_rec_article_short_title = 13755; //Value matching "Logogriphe" in the
"Article récurrent" taxonomy.
    }
    elseif(in_array($id, $tex_rec_article_short_title_charade_array_id) ||
($titre_normalise == 'Autre' && $genre_1 == 'Charade' && $genre_2 == '') ||
(stripos($titre_normalise, 'Charade à M') !== false) || (stripos($titre_normalise,
'Charade à Iris') !== false) || (stripos($titre_normalise, 'Charade à Philis') !==
false) || (stripos($titre_normalise, 'Charade adressée à M') !== false) ||
((stripos($titre_normalise, 'Charade') !== false) && (stripos($titre_normalise,
'Énigme') === false) && (stripos($titre_normalise, 'Logogriphe') === false) &&
$genre_1 == 'Charade' && $genre_2 == '') || ((stripos($titre_normalise, 'Autre')
!== false) && (stripos($titre_normalise, 'Énigme') === false) &&
```

```
(stripos($titre_normalise, 'Logogriphe') === false) && $genre_1 == 'Charade' &&
$genre_2 == '') || $titre_normalise == 'Charade' || $titre_normalise == 'Autre
charade' || $titre_normalise == 'Première charade' || $titre_normalise == 'Seconde
charade' || $titre_normalise == 'Deuxième charade' || $titre_normalise ==
'Troisième charade' || $titre_normalise == 'Quatrième charade' || $titre_normalise
== 'Cinquième charade' || $titre_normalise == 'Sixième charade' ||
$titre_normalise == 'Charade [et logogriphe]' || $titre_normalise == 'Charade.
Conte' || $titre_normalise == 'Charade, conte'){
        $tex_rec_article_short_title = 13756; //Value matching "Charade" in the
"Article récurrent" taxonomy.
    }
    elseif((stripos($titre_normalise, 'Énigmes adressées à M') !== false) ||
$titre_normalise == 'Nouvelles énigmes' || $titre_normalise == 'Énigmes' ||
$titre_normalise == 'Énigmes et logogriphe' || $titre_normalise == 'Énigmes. À M.
L..... [Lagache fils]' || $titre_normalise == 'Énigmes à Mlle Cl... de la Gr.....
à Rennes'){
        $tex_rec_article_short_title = 11932; //Value matching "Énigmes" in the
"Article récurrent" taxonomy.
    }
    elseif((stripos($titre_normalise, 'Logogriphes adressés à M') !== false) ||
$titre_normalise == 'Nouveaux logogriphes' || $titre_normalise == 'Logogriphes' ||
$titre_normalise == 'Logogriphes. À M. l\'Abbé de V...'){
        $tex_rec_article_short_title = 13763; //Value matching "Logogriphes" in
the "Article récurrent" taxonomy.
    }
    elseif((stripos($titre_normalise, 'Charades adressées à M') !== false) ||
$titre_normalise == 'Nouvelles charades' || $titre_normalise == 'Charades' ||
$titre_normalise == 'Charades, sur l\'air : Vous qui d\'amoureuse aventure. À Mlle
E... H... de Bologne'){
        $tex_rec_article_short_title = 13764; //Value matching "Charades" in the
"Article récurrent" taxonomy.
    }
    elseif(in_array($id, $tex_rec_article_short_title_enigma_vaudeville_array_id)
|| ($titre_normalise == 'Autre en vaudevilles' && $genre_1 == 'Énigme' && $genre_2
== 'Chanson') || (stripos($titre_normalise, 'Énigme en vaudevilles') !== false) ||
(stripos($titre_normalise, 'Énigmes en vaudevilles') !== false)){
        $tex_rec_article_short_title = 13765; //Value matching "Énigme en
vaudevilles" in the "Article récurrent" taxonomy.
    }
    elseif(($titre_normalise == 'Autre en vaudevilles' && $genre_1 == 'Logogriphe'
&& $genre_2 == 'Chanson') || (stripos($titre_normalise, 'Logogriphe en
vaudevilles') !== false) || (stripos($titre_normalise, 'Logogriphes en
vaudevilles') !== false)){
        $tex_rec_article_short_title = 13766; //Value matching "Logogriphe en
vaudevilles" in the "Article récurrent" taxonomy.
    }
    elseif((($titre_normalise == 'Alius' || $titre_normalise == 'Alter') &&
$genre_1 == 'Énigme' && $genre_2 == '') || $titre_normalise == 'Enigma' ||
$titre_normalise == 'Aenigma' || $titre_normalise == 'Enigma cum logogripho'){
        $tex_rec_article_short_title = 13757; //Value matching "Enigma" in the
"Article récurrent" taxonomy.
    }
    elseif((($titre_normalise == 'Alius' || $titre_normalise == 'Alter') &&
$genre_1 == 'Logogriphe' && $genre_2 == '') || $titre_normalise == 'Logogriphus'){
        $tex_rec_article_short_title = 13758; //Value matching "Logogriphus" in
the "Article récurrent" taxonomy.
    }
    elseif($titre_normalise == 'Énigme-logogriphe' || $titre_normalise == 'Énigme
logogriphe'){
```

```
        $tex_rec_article_short_title = 13759; //Value matching "Énigme logogriphe"
in the "Article récurrent" taxonomy.
    }
    elseif($titre_normalise == 'Énigme en logogriphe' || $titre_normalise ==
'Énigme en logogriphe, sur une partie des rimes de celui de Monsieur Jacques sur
le mot conversation'){
        $tex_rec_article_short_title = 13767; //Value matching "Énigme en
logogriphe" in the "Article récurrent" taxonomy.
    }
    elseif($titre_normalise == 'Énigme logogriphique'){
        $tex_rec_article_short_title = 13760; //Value matching "Énigme
logogriphique" in the "Article récurrent" taxonomy.
    }
    elseif($titre_normalise == 'Logogriphe énigmatique'){
        $tex_rec_article_short_title = 13761; //Value matching "Logogriphe
énigmatique" in the "Article récurrent" taxonomy.
    }
    elseif($titre_normalise == 'Sonnet énigmatique' || $titre_normalise == 'Sonnet
énigmatique, dont les rimes ont été données dans le Mercure'){
        $tex_rec_article_short_title = 13762; //Value matching "Sonnet
énigmatique" in the "Article récurrent" taxonomy.
    }

    //Create field tex_incipit from incipit.

    $tex_incipit = $incipit;

    if (in_array($year_check, $tex_incipit_italics_array_year) ||
in_array($year_month_check, $tex_incipit_italics_array_year_month) ||
in_array($date_check, $tex_incipit_italics_array_date) ||
in_array($date_number_check, $tex_incipit_italics_array_date_number) ||
in_array($id_check, $tex_incipit_italics_array_id)){
        $tex_incipit = '<em>'.$tex_incipit.'</em>';
    }

    //Exceptions.

    if($id == 6891){
        $tex_incipit = "L'enfant, en doublant mon <em>premier</em>,";
    }
    if($id == 6868){
        $tex_incipit = "A MADRID, dans un cirque, où la foule se presse,";
    }
    if($id == 6841){
        $tex_incipit = "Je suis blonde et jamais je ne sers qu'<em>a la
brune,</em>";
    }
    if($id == 6744){
        $tex_incipit = "Je porte <em>lac</em>, je porte <em>cime</em>,";
    }
    if($id == 6315){
        $tex_incipit = "Je suis dans mon premier l'opposé de <em>mauvais</em>";
    }
    if($id == 6249){
        $tex_incipit = "LA FONTAINE était mon premier ;";
    }
    if($id == 6022){
        $tex_incipit = "Je ne suis point <em>abeille</em>,";
    }
```

```
    if($id == 5978){
        $tex_incipit = "MELCOUR en toi, PHILIS, admire mon entier ;";
    }
    if($id == 5728){
        $tex_incipit = "La <em>seconde</em> enrichit, & le <em>premier</em> dévore
:";
    }
    if($id == 5662){
        $tex_incipit = "Moineau, qui de l'<em>entier</em>";
    }
    if($id == 6905){
        $tex_incipit = "Sans secours merveilleux de Sylphe ni de <em>Fée,</em>";
    }
    if($id == 4742){
        $tex_incipit = "C'est avec quatre pieds que je suis <em>corps
solide</em>,";
    }
    if($id == 1567){
        $tex_incipit = "Lecteur, qu'est-ce qu'un mot où tu trouves deux
<em>U</em>,";
    }
    if($id == 2902){
        $tex_incipit = "Six <em>pieds de Roi</em>, Madame, <em>de carreau</em>";
    }
    if($id == 2652){
        $tex_incipit = "MONSIEUR, Il a été inséré dans votre Mercure de Novembre /
Je suis un fruit délicieux,";
    }
    if($id == 2573){
        $tex_incipit = "Aimable <em>Hébé</em>, vos desirs sont des loix ;";
    }
    if($id == 2486){
        $tex_incipit = "Semblables à <em>Vénus,</em> leur mère,";
    }
    if($id == 2456){
        $tex_incipit = "Je suis composé de deux . <em>mots,</em>";
    }
    if($id == 2438){
        $tex_incipit = "C'étoit pousser trop loin mainte <em>plaisanterie</em>,";
    }
    if($id == 2392){
        $tex_incipit = "Sans moi, blonde <em>Phryné</em>, seriez-vous aussi
belle,";
    }
    if($id == 2368){
        $tex_incipit = "Quoi ! vous voulez, <em>Hortense</em>, un logogryphe !";
    }
    if($id == 2356){
        $tex_incipit = "Aimable <em>Eglé</em>, tout céde à ma beauté,";
    }
    if($id == 2254){
        $tex_incipit = "Un peu de négligence, adorable <em>Clarice</em>,";
    }
    if($id == 2218){
        $tex_incipit = "Je suis nuit & jour sur pieds, toujours prêt à être utile.
Je travaille pour";
    }
    if($id == 2208){
        $tex_incipit = "Tout Rhéteur me connoît : <em>Cicéron</em> m'employa ;";
```

```
    }
    if($id == 2201){
        $tex_incipit = "Ma tête gouverne le monde ; ma queue est dans le Ciel &
dans les Enfers.";
    }
    if($id == 4071){
        $tex_incipit = "Dans les premiers temps, je n'estois apparemment employée
qu'à";
    }
    if($id == 4070){
        $tex_incipit = "Je suis François pour la vie ; & suis neanmoins comme les
Italiens & les";
    }
    if($id == 2200){
        $tex_incipit = "Présent du Nord, je préserve de ses rigueurs. Je contiens
ce qui embellit la nature ; une plaine";
    }
    if($id == 2199){
        $tex_incipit = "J'ai quatre pieds & deux lettres. Je renferme une
<em>Saison</em>, &, rien de plus. Quand je suis froide,";
    }
    if($id == 2189){
        $tex_incipit = "Dans mes six pieds, belle <em>Uranie</em>,";
    }
    if($id == 2175){
        $tex_incipit = "Contre moi, tendre <em>A</em>, ** pourquoi tant de colère
?";
    }
    if($id == 2137){
        $tex_incipit = "Voulez-vous me trouver, <em>Thémire ?</em>";
    }
    if($id == 2106){
        $tex_incipit = "Eléve de <em>Bellone</em>, au milieu des hasards";
    }
    if($id == 2088){
        $tex_incipit = "Je gagerois que la <em>Camille</em>,";
    }
    if($id == 2085){
        $tex_incipit = "Belle <em>Eglé</em>, si des lieux";
    }
    if($id == 1637){
        $tex_incipit = "Certaine piéce d'or, jointe à la lettre <em>T</em>,";
    }
    if($id == 1003){
        $tex_incipit = "Je porte <em>Mien</em>, je porte <em>Met</em>,";
    }
    if($id == 3641){
        $tex_incipit = "<em>C'est moy qui fais valoir un joly</em> Flageolet,";
    }
    if($id == 3642){
        $tex_incipit = "<em>J'ai le corps fait ainsi qu'un</em> Flageolet,";
    }
    if($id == 3643){
        $tex_incipit = "<em>J'ai mis au desespoir la Nymphe du Dieu</em> Pan,";
    }
    if($id == 3644){
        $tex_incipit = "<em>Pour défendre le Sexe-aimé jadis de</em> Pan,";
    }
    if($id == 3649){
```

```
        $tex_incipit = "<em>Je détourne aisément les coups de</em> Jupiter,";
    }
    if($id == 4133){
        $tex_incipit = "Ces admirables Forts, cette immense Machine,";
    }
    if($id == 4137){
        $tex_incipit = "<em>Je suis d'abord un premier</em> element,";
    }
    if($id == 4176){
        $tex_incipit = "En noirs cachots, <em>& hideuse clôture,</em>";
    }
    if($id == 4178){
        $tex_incipit = "<em>Quelquefois pour me prendre on fait sonner la</em> .
cloche,";
    }
    if($id == 4179){
        $tex_incipit = "<em>Je peux tout transporter sans aucun</em> . attelage,";
    }
    if($id == 4182){
        $tex_incipit = "<em>Mon pouvoir fait souvent trembler Mardix &</em>
Furne,";
    }
    if($id == 4267){
        $tex_incipit = "Un ami <em>de fort loin me vient rendre visite</em> ;";
    }
    if($id == 86){
        $tex_incipit = "Mon corps n'est point couvert d'écailles
<em>Jaunissantes</em>,";
    }
    if($id == 106){
        $tex_incipit = "Celui chez qui je suis est toûjours <em>Taciturne</em>,";
    }
    if($id == 141){
        $tex_incipit = "On me trouve en tout lieu, je suis dans la
<em>Calabre</em>,";
    }
    if($id == 280){
        $tex_incipit = "Pourriez-vous deviner l'Enigme que <em>voilà ?</em>";
    }
    if($id == 398){
        $tex_incipit = "<em>Si totus sumar, metiri tempora possum,</em>";
    }
    if($id == 419){
        $tex_incipit = "<em>Incubuit terris per me genus omne malorum,</em>";
    }
    if($id == 496){
        $tex_incipit = "<em>Astra colo soboles aterni virgo tonantis,</em>";
    }
    if($id == 651){
        $tex_incipit = "<em>Aeriis quondam volitans in nubibus basis,</em>";
    }
    if($id == 669){
        $tex_incipit = "<em>Tota sinistra vocor, terras mea cauda dealbat,</em>";
    }
    if($id == 670){
        $tex_incipit = "<em>Totius h[ae]c dos est ; hominesque ferasque
fatigo.</em>";
    }
    if($id == 707){
```

```
        $tex_incipit = "<em>Aures obtundo. Per partes, abdita signo :</em>";
    }
    if($id == 717){
        $tex_incipit = "<em>Exorno facies hominum, nunc signa decoris</em>";
    }
    if($id == 721){
        $tex_incipit = "<em>Uno complector diversum nomine sensum :</em>";
    }
    if($id == 726){
        $tex_incipit = "Ennemi de la <em>Pauvreté,</em>";
    }
    if($id == 727){
        $tex_incipit = "Quoi, l'on trouve dans <em>Charlatan</em>,";
    }
    if($id == 730){
        $tex_incipit = "<em>Partibus è septem conflor, miscere potentes</em>";
    }
    if($id == 735){
        $tex_incipit = "<em>Stagnantes undas limoso gurgite totus</em>";
    }
    if($id == 736){
        $tex_incipit = "<em>Noscere vis totum ? divina Volumina nosce,</em>";
    }
    if($id == 741){
        $tex_incipit = "<em>Quis fuerim, si scire voles ; en accipe Lector.</em>";
    }
    if($id == 746){
        $tex_incipit = "<em>Quinque meum pedibus nomen componitur, arbor,</em>";
    }
    if($id == 751){
        $tex_incipit = "<em>Si septem gradiar pedibus, sum doctor, et ecce</em>";
    }

    //Manage rich text field.

    if($tex_incipit != ''){

        $tex_incipit = preg_replace(
            '/"(\S)/',
            '« $1',
            $tex_incipit
        );

        $tex_incipit = preg_replace(
            '/(\S)"/',
            '$1 »',
            $tex_incipit
        );

        $tex_incipit = str_replace("’", "'", $tex_incipit);
        $tex_incipit = str_replace("&rsquo;", "'", $tex_incipit);
        $tex_incipit = str_replace('"', '""', $tex_incipit);
        $tex_incipit = '"{ format: full_html, value: ""<p>'.$tex_incipit.'</p>""
}"';
        $tex_incipit = htmlspecialchars($tex_incipit, ENT_QUOTES, 'UTF-8');
    }

    //Create field tex_position_on_page.
```

```
    //We first need to create an array with the IDs of the texts that start and
end on the same page as one or several other texts, along with their positions on
the page.

    $tex_position_on_page_array = [
        76 => 1, 77 => 2, 104 => 1, 105 => 2, 120 => 1, 121 => 2, 197 => 1, 198 =>
2, 214 => 1, 215 => 2, 224 => 1, 225 => 2, 226 => 3, 229 => 1, 230 => 2, 244 => 1,
245 => 2, 295 => 1, 296 => 2, 323 => 1, 324 => 2, 345 => 1, 346 => 2, 393 => 1,
394 => 2, 447 => 1, 448 => 2, 460 => 1, 461 => 2, 462 => 1, 463 => 2, 495 => 1,
496 => 2, 529 => 1, 530 => 2, 537 => 1, 538 => 2, 539 => 3, 540 => 1, 541 => 2,
565 => 1, 566 => 2, 572 => 1, 573 => 2, 574 => 1, 575 => 2, 585 => 1, 586 => 2,
591 => 1, 592 => 2, 637 => 1, 638 => 2, 673 => 1, 674 => 2, 675 => 3, 680 => 1,
681 => 2, 697 => 1, 698 => 2, 699 => 1, 700 => 2, 720 => 1, 721 => 2, 734 => 1,
735 => 2, 740 => 1, 741 => 2, 744 => 1, 745 => 2, 746 => 3, 761 => 1, 762 => 2,
767 => 1, 768 => 2, 769 => 1, 770 => 2, 771 => 3, 772 => 1, 773 => 2, 774 => 3,
778 => 1, 779 => 2, 781 => 1, 782 => 2, 783 => 3, 784 => 1, 785 => 2, 786 => 3,
788 => 1, 789 => 2, 791 => 1, 792 => 2, 796 => 1, 797 => 2, 798 => 3, 799 => 4,
804 => 1, 805 => 2, 806 => 3, 808 => 1, 809 => 2, 817 => 1, 818 => 2, 824 => 1,
825 => 2, 833 => 1, 834 => 2, 835 => 3, 839 => 1, 840 => 2, 855 => 1, 856 => 2,
857 => 3, 863 => 1, 864 => 2, 869 => 1, 870 => 2, 875 => 1, 876 => 2, 883 => 1,
884 => 2, 885 => 3, 905 => 1, 906 => 2, 910 => 1, 911 => 2, 924 => 1, 925 => 2,
932 => 1, 933 => 2, 956 => 1, 957 => 2, 1138 => 1, 1139 => 2, 1146 => 1, 1147 =>
2, 1159 => 1, 1160 => 2, 1166 => 1, 1167 => 2, 1170 => 1, 1171 => 2, 1183 => 1,
1184 => 2, 1225 => 1, 1226 => 2, 1248 => 1, 1249 => 2, 1251 => 1, 1252 => 2, 1258
=> 1, 1259 => 2, 1276 => 1, 1277 => 2, 1282 => 1, 1283 => 2, 1286 => 1, 1287 => 2,
1304 => 1, 1305 => 2, 1306 => 1, 1307 => 2, 1322 => 1, 1323 => 2, 1324 => 1, 1325
=> 2, 1339 => 1, 1340 => 2, 1346 => 1, 1347 => 2, 1386 => 1, 1387 => 2, 1388 => 3,
1389 => 4, 1418 => 1, 1419 => 2, 1490 => 1, 1491 => 2, 1508 => 1, 1509 => 2, 1526
=> 1, 1527 => 2, 1529 => 1, 1530 => 2, 1559 => 1, 1560 => 2, 1563 => 1, 1564 => 2,
1574 => 1, 1575 => 2, 1576 => 3, 1589 => 1, 1590 => 2, 1591 => 1, 1592 => 2, 1595
=> 1, 1596 => 2, 1652 => 1, 1653 => 2, 1669 => 1, 1670 => 2, 1692 => 1, 1693 => 2,
1770 => 1, 1771 => 2, 1927 => 1, 1928 => 2, 1931 => 1, 1932 => 2, 1945 => 1, 1946
=> 2, 1952 => 1, 1953 => 2, 1958 => 1, 1959 => 2, 1961 => 1, 1962 => 2, 1966 => 1,
1967 => 2, 1971 => 1, 1972 => 2, 1979 => 1, 1980 => 2, 1983 => 1, 1984 => 2, 1985
=> 3, 2006 => 1, 2007 => 2, 2008 => 3, 2026 => 1, 2027 => 2, 2036 => 1, 2037 => 2,
2055 => 1, 2056 => 2, 2063 => 1, 2064 => 2, 2079 => 1, 2080 => 2, 2111 => 1, 2112
=> 2, 2121 => 1, 2122 => 2, 2136 => 1, 2137 => 2, 2164 => 1, 2165 => 2, 2169 => 1,
2170 => 2, 2175 => 1, 2176 => 2, 2182 => 1, 2183 => 2, 2199 => 1, 2200 => 2, 2201
=> 3, 2225 => 1, 2226 => 2, 2230 => 1, 2231 => 2, 2235 => 1, 2236 => 2, 2248 => 1,
2249 => 2, 2265 => 1, 2266 => 2, 2271 => 1, 2272 => 2, 2278 => 1, 2279 => 2, 2285
=> 1, 2286 => 2, 2288 => 1, 2289 => 2, 2300 => 1, 2301 => 2, 2327 => 1, 2328 => 2,
2343 => 1, 2344 => 2, 2347 => 1, 2348 => 2, 2350 => 1, 2351 => 2, 2352 => 1, 2353
=> 2, 2359 => 1, 2360 => 2, 2361 => 1, 2362 => 2, 2376 => 1, 2377 => 2, 2378 => 1,
2379 => 2, 2380 => 3, 2386 => 1, 2387 => 2, 2393 => 1, 2394 => 2, 2406 => 1, 2407
=> 2, 2408 => 1, 2409 => 2, 2410 => 3, 2412 => 1, 2413 => 2, 2414 => 1, 2415 => 2,
2443 => 1, 2444 => 2, 2447 => 1, 2448 => 2, 2463 => 1, 2464 => 2, 2528 => 1, 2529
=> 2, 2542 => 1, 2543 => 2, 2547 => 1, 2548 => 2, 2556 => 1, 2557 => 2, 2558 => 1,
2559 => 2, 2562 => 1, 2563 => 2, 2564 => 1, 2565 => 2, 2579 => 1, 2580 => 2, 2581
=> 1, 2582 => 2, 2593 => 1, 2594 => 2, 2595 => 3, 2596 => 1, 2597 => 2, 2607 => 1,
2608 => 2, 2651 => 1, 2652 => 2, 2673 => 1, 2674 => 2, 2684 => 1, 2685 => 2, 2690
=> 1, 2691 => 2, 2700 => 1, 2701 => 2, 2769 => 1, 2770 => 2, 2792 => 1, 2793 => 2,
2801 => 1, 2802 => 2, 2807 => 1, 2808 => 2, 2821 => 1, 2822 => 2, 2826 => 1, 2827
=> 2, 2841 => 1, 2842 => 2, 2843 => 3, 2860 => 1, 2861 => 2, 2897 => 1, 2898 => 2,
2907 => 1, 2908 => 2, 2918 => 1, 2919 => 2, 2920 => 1, 2921 => 2, 2927 => 1, 2928
=> 2, 2975 => 1, 2976 => 2, 3015 => 1, 3016 => 2, 3017 => 3, 3036 => 1, 3037 => 2,
3049 => 1, 3050 => 2, 3068 => 1, 3069 => 2, 3076 => 1, 3077 => 2, 3089 => 1, 3090
=> 2, 3119 => 1, 3120 => 2, 3128 => 1, 3129 => 2, 3136 => 1, 3137 => 2, 3138 => 1,
3139 => 2, 3141 => 1, 3142 => 2, 3163 => 1, 3164 => 2, 3180 => 1, 3181 => 2, 3186
=> 1, 3187 => 2, 3194 => 1, 3195 => 2, 3237 => 1, 3238 => 2, 3246 => 1, 3247 => 2,
```

```
3249 => 1, 3250 => 2, 3254 => 1, 3255 => 2, 3260 => 1, 3261 => 2, 3276 => 1, 3277
=> 2, 3283 => 1, 3284 => 2, 3294 => 1, 3295 => 2, 3302 => 1, 3303 => 2, 3305 => 1,
3306 => 2, 3313 => 1, 3314 => 2, 3321 => 1, 3322 => 2, 3323 => 3, 3325 => 1, 3326
=> 2, 3328 => 1, 3329 => 2, 3347 => 1, 3348 => 2, 3362 => 1, 3363 => 2, 3364 => 1,
3365 => 2, 3379 => 1, 3380 => 2, 3383 => 1, 3384 => 2, 3479 => 1, 3480 => 2, 3502
=> 1, 3503 => 2, 3512 => 1, 3513 => 2, 3515 => 1, 3516 => 2, 3517 => 1, 3518 => 2,
3527 => 1, 3528 => 2, 3563 => 1, 3564 => 2, 4249 => 1, 4250 => 2, 4259 => 1, 4260
=> 2, 4297 => 1, 4298 => 2, 4301 => 1, 4302 => 2, 4312 => 1, 4313 => 2, 4318 => 1,
4319 => 2, 4332 => 1, 4333 => 2, 4359 => 1, 4360 => 2, 4365 => 1, 4366 => 2, 4388
=> 1, 4389 => 2, 4401 => 1, 4402 => 2, 4404 => 1, 4405 => 2, 4421 => 1, 4422 => 2,
4425 => 1, 4426 => 2, 4473 => 1, 4474 => 2, 4477 => 1, 4478 => 2, 4494 => 1, 4495
=> 2, 4598 => 1, 4599 => 2, 4600 => 1, 4601 => 2, 4614 => 1, 4615 => 2, 4632 => 1,
4633 => 2, 4660 => 1, 4661 => 2, 4666 => 1, 4667 => 2, 4674 => 1, 4675 => 2, 4680
=> 1, 4681 => 2, 4692 => 1, 4693 => 2, 4739 => 1, 4740 => 2, 4750 => 1, 4751 => 2,
4788 => 1, 4789 => 2, 4790 => 3, 4791 => 4, 4792 => 1, 4793 => 2, 4808 => 1, 4809
=> 2, 4810 => 1, 4811 => 2, 4818 => 1, 4819 => 2, 4820 => 1, 4821 => 2, 4832 => 1,
4833 => 2, 4834 => 3, 4842 => 1, 4843 => 2, 4844 => 3, 4851 => 1, 4852 => 2, 4865
=> 1, 4866 => 2, 4867 => 3, 4868 => 4, 4869 => 1, 4870 => 2, 4871 => 1, 4872 => 2,
4874 => 1, 4875 => 2, 4876 => 3, 4879 => 1, 4880 => 2, 4887 => 1, 4888 => 2, 4889
=> 3, 4891 => 1, 4892 => 2, 4893 => 3, 4895 => 1, 4896 => 2, 4897 => 3, 4898 => 4,
4899 => 5, 4905 => 1, 4906 => 2, 4908 => 1, 4909 => 2, 4910 => 1, 4911 => 2, 4915
=> 1, 4916 => 2, 4917 => 1, 4918 => 2, 4919 => 1, 4920 => 2, 4924 => 1, 4925 => 2,
4928 => 1, 4929 => 2, 4930 => 3, 4931 => 4, 4932 => 1, 4933 => 2, 4934 => 1, 4935
=> 2, 4936 => 3, 4937 => 1, 4938 => 2, 4940 => 1, 4941 => 2, 4943 => 1, 4944 => 2,
4949 => 1, 4950 => 2, 4955 => 1, 4956 => 2, 4963 => 1, 4964 => 2, 4967 => 1, 4968
=> 2, 4970 => 1, 4971 => 2, 4977 => 1, 4978 => 2, 4979 => 3, 4989 => 1, 4990 => 2,
4992 => 1, 4993 => 2, 4997 => 1, 4998 => 2, 4999 => 3, 5003 => 1, 5004 => 2, 5012
=> 1, 5013 => 2, 5014 => 3, 5021 => 1, 5022 => 2, 5025 => 1, 5026 => 2, 5031 => 1,
5032 => 2, 5033 => 1, 5034 => 2, 5042 => 1, 5043 => 2, 5045 => 1, 5046 => 2, 5051
=> 1, 5052 => 2, 5054 => 1, 5055 => 2, 5058 => 1, 5059 => 2, 5063 => 1, 5064 => 2,
5069 => 1, 5070 => 2, 5072 => 1, 5073 => 2, 5074 => 3, 5075 => 1, 5076 => 2, 5082
=> 1, 5083 => 2, 5087 => 1, 5088 => 2, 5090 => 1, 5091 => 2, 5096 => 1, 5097 => 2,
5098 => 3, 5099 => 1, 5100 => 2, 5105 => 1, 5106 => 2, 5109 => 1, 5110 => 2, 5114
=> 1, 5115 => 2, 5117 => 1, 5118 => 2, 5120 => 1, 5121 => 2, 5129 => 1, 5130 => 2,
5132 => 1, 5133 => 2, 5136 => 1, 5137 => 2, 5141 => 1, 5142 => 2, 5144 => 1, 5145
=> 2, 5150 => 1, 5151 => 2, 5153 => 1, 5154 => 2, 5162 => 1, 5163 => 2, 5172 => 1,
5173 => 2, 5174 => 1, 5175 => 2, 5178 => 1, 5179 => 2, 5184 => 1, 5185 => 2, 5189
=> 1, 5190 => 2, 5199 => 1, 5200 => 2, 5201 => 1, 5202 => 2, 5207 => 1, 5208 => 2,
5213 => 1, 5214 => 2, 5235 => 1, 5236 => 2, 5238 => 1, 5239 => 2, 5257 => 1, 5258
=> 2, 5280 => 1, 5281 => 2, 5286 => 1, 5287 => 2, 5292 => 1, 5293 => 2, 5294 => 3,
5295 => 1, 5296 => 2, 5301 => 1, 5302 => 2, 5304 => 1, 5305 => 2, 5314 => 1, 5315
=> 2, 5338 => 1, 5339 => 2, 5359 => 1, 5360 => 2, 5362 => 1, 5363 => 2, 5368 => 1,
5369 => 2, 5381 => 1, 5382 => 2, 5389 => 1, 5390 => 2, 5392 => 1, 5393 => 2, 5405
=> 1, 5406 => 2, 5407 => 1, 5408 => 2, 5411 => 1, 5412 => 2, 5419 => 1, 5420 => 2,
5432 => 1, 5433 => 2, 5452 => 1, 5453 => 2, 5455 => 1, 5456 => 2, 5464 => 1, 5465
=> 2, 5467 => 1, 5468 => 2, 5470 => 1, 5471 => 2, 5476 => 1, 5477 => 2, 5488 => 1,
5489 => 2, 5503 => 1, 5504 => 2, 5522 => 1, 5523 => 2, 5530 => 1, 5531 => 2, 5539
=> 1, 5540 => 2, 5572 => 1, 5573 => 2, 5575 => 1, 5576 => 2, 5578 => 1, 5579 => 2,
5581 => 1, 5582 => 2, 5584 => 1, 5585 => 2, 5614 => 1, 5615 => 2, 5623 => 1, 5624
=> 2, 5638 => 1, 5639 => 2, 5641 => 1, 5642 => 2, 5651 => 1, 5652 => 2, 5665 => 1,
5666 => 2, 5686 => 1, 5687 => 2, 5689 => 1, 5690 => 2, 5692 => 1, 5693 => 2, 5698
=> 1, 5699 => 2, 5701 => 1, 5702 => 2, 5703 => 3, 5734 => 1, 5735 => 2, 5737 => 1,
5738 => 2, 5752 => 1, 5753 => 2, 5755 => 1, 5756 => 2, 5758 => 1, 5759 => 2, 5767
=> 1, 5768 => 2, 5776 => 1, 5777 => 2, 5780 => 1, 5781 => 2, 5788 => 1, 5789 => 2,
5809 => 1, 5810 => 2, 5811 => 3, 5813 => 1, 5814 => 2, 5816 => 1, 5817 => 2, 5827
=> 1, 5828 => 2, 5836 => 1, 5837 => 2, 5840 => 1, 5841 => 2, 5845 => 1, 5846 => 2,
5848 => 1, 5849 => 2, 5851 => 1, 5852 => 2, 5860 => 1, 5861 => 2, 5863 => 1, 5864
=> 2, 5875 => 1, 5876 => 2, 5881 => 1, 5882 => 2, 5888 => 1, 5889 => 2, 5891 => 1,
5892 => 2, 5897 => 1, 5898 => 2, 5900 => 1, 5901 => 2, 5910 => 1, 5911 => 2, 5915
```

```
=> 1, 5916 => 2, 5917 => 3, 5937 => 1, 5938 => 2, 5940 => 1, 5941 => 2, 5948 => 1,
5949 => 2, 5950 => 3, 5951 => 1, 5952 => 2, 5953 => 3, 5960 => 1, 5961 => 2, 5963
=> 1, 5964 => 2, 5970 => 1, 5971 => 2, 5975 => 1, 5976 => 2, 5977 => 3, 5981 => 1,
5982 => 2, 5988 => 1, 5989 => 2, 5991 => 1, 5992 => 2, 5993 => 1, 5994 => 2, 6000
=> 1, 6001 => 2, 6002 => 1, 6003 => 2, 6004 => 3, 6005 => 1, 6006 => 2, 6011 => 1,
6012 => 2, 6013 => 3, 6029 => 1, 6030 => 2, 6032 => 1, 6033 => 2, 6038 => 1, 6039
=> 2, 6041 => 1, 6042 => 2, 6056 => 1, 6057 => 2, 6058 => 3, 6069 => 1, 6070 => 2,
6076 => 1, 6077 => 2, 6078 => 1, 6079 => 2, 6099 => 1, 6100 => 2, 6102 => 1, 6103
=> 2, 6112 => 1, 6113 => 2, 6115 => 1, 6116 => 2, 6122 => 1, 6123 => 2, 6124 => 1,
6125 => 2, 6126 => 3, 6133 => 1, 6134 => 2, 6146 => 1, 6147 => 2, 6168 => 1, 6169
=> 2, 6170 => 3, 6171 => 1, 6172 => 2, 6173 => 3, 6174 => 1, 6175 => 2, 6180 => 1,
6181 => 2, 6184 => 1, 6185 => 2, 6202 => 1, 6203 => 2, 6205 => 1, 6206 => 2, 6213
=> 1, 6214 => 2, 6217 => 1, 6218 => 2, 6223 => 1, 6224 => 2, 6225 => 1, 6226 => 2,
6228 => 1, 6229 => 2, 6234 => 1, 6235 => 2, 6236 => 3, 6238 => 1, 6239 => 2, 6246
=> 1, 6247 => 2, 6252 => 1, 6253 => 2, 6255 => 1, 6256 => 2, 6258 => 1, 6259 => 2,
6270 => 1, 6271 => 2, 6276 => 1, 6277 => 2, 6279 => 1, 6280 => 2, 6283 => 1, 6284
=> 2, 6294 => 1, 6295 => 2, 6296 => 3, 6307 => 1, 6308 => 2, 6316 => 1, 6317 => 2,
6321 => 1, 6322 => 2, 6325 => 1, 6326 => 2, 6333 => 1, 6334 => 2, 6340 => 1, 6341
=> 2, 6342 => 1, 6343 => 2, 6345 => 1, 6346 => 2, 6351 => 1, 6352 => 2, 6353 => 3,
6354 => 1, 6355 => 2, 6363 => 1, 6364 => 2, 6366 => 1, 6367 => 2, 6370 => 1, 6371
=> 2, 6375 => 1, 6376 => 2, 6381 => 1, 6382 => 2, 6384 => 1, 6385 => 2, 6386 => 3,
6388 => 1, 6389 => 2, 6396 => 1, 6397 => 2, 6398 => 3, 6399 => 1, 6400 => 2, 6401
=> 3, 6403 => 1, 6404 => 2, 6408 => 1, 6409 => 2, 6410 => 3, 6415 => 1, 6416 => 2,
6473 => 1, 6474 => 2, 6482 => 1, 6483 => 2, 6484 => 3, 6485 => 1, 6486 => 2, 6487
=> 3, 6495 => 1, 6496 => 2, 6497 => 1, 6498 => 2, 6499 => 3, 6500 => 1, 6501 => 2,
6502 => 3, 6503 => 1, 6504 => 2, 6505 => 3, 6507 => 1, 6508 => 2, 6509 => 1, 6510
=> 2, 6511 => 3, 6512 => 1, 6513 => 2, 6514 => 3, 6515 => 1, 6516 => 2, 6518 => 1,
6519 => 2, 6520 => 1, 6521 => 2, 6522 => 3, 6524 => 1, 6525 => 2, 6529 => 1, 6530
=> 2, 6531 => 3, 6533 => 1, 6534 => 2, 6537 => 1, 6538 => 2, 6539 => 1, 6540 => 2,
6541 => 1, 6542 => 2, 6547 => 1, 6548 => 2, 6556 => 1, 6557 => 2, 6561 => 1, 6562
=> 2, 6564 => 1, 6565 => 2, 6566 => 1, 6567 => 2, 6568 => 1, 6569 => 2, 6573 => 1,
6574 => 2, 6575 => 1, 6576 => 2, 6581 => 1, 6582 => 2, 6583 => 1, 6584 => 2, 6589
=> 1, 6590 => 2, 6591 => 1, 6592 => 2, 6594 => 1, 6595 => 2, 6596 => 3, 6597 => 1,
6598 => 2, 6599 => 3, 6600 => 1, 6601 => 2, 6602 => 3, 6603 => 1, 6604 => 2, 6605
=> 3, 6606 => 1, 6607 => 2, 6608 => 1, 6609 => 2, 6610 => 1, 6611 => 2, 6612 => 1,
6613 => 2, 6614 => 3, 6615 => 1, 6616 => 2, 6617 => 3, 6618 => 1, 6619 => 2, 6620
=> 3, 6621 => 1, 6622 => 2, 6623 => 3, 6628 => 1, 6629 => 2, 6630 => 1, 6631 => 2,
6632 => 1, 6633 => 2, 6634 => 3, 6636 => 1, 6637 => 2, 6638 => 1, 6639 => 2, 6640
=> 3, 6641 => 1, 6642 => 2, 6644 => 1, 6645 => 2, 6647 => 1, 6648 => 2, 6649 => 1,
6650 => 2, 6652 => 1, 6653 => 2, 6655 => 1, 6656 => 2, 6658 => 1, 6659 => 2, 6660
=> 1, 6661 => 2, 6666 => 1, 6667 => 2, 6668 => 1, 6669 => 2, 6670 => 1, 6671 => 2,
6677 => 1, 6678 => 2, 6683 => 1, 6684 => 2, 6685 => 1, 6686 => 2, 6692 => 1, 6693
=> 2, 6696 => 1, 6697 => 2, 6698 => 1, 6699 => 2, 6700 => 1, 6701 => 2, 6706 => 1,
6707 => 2, 6708 => 1, 6709 => 2, 6710 => 1, 6711 => 2, 6714 => 1, 6715 => 2, 6716
=> 1, 6717 => 2, 6718 => 1, 6719 => 2, 6720 => 1, 6721 => 2, 6722 => 1, 6723 => 2,
6727 => 1, 6728 => 2, 6732 => 1, 6733 => 2, 6734 => 1, 6735 => 2, 6739 => 1, 6740
=> 2, 6742 => 1, 6743 => 2, 6745 => 1, 6746 => 2, 6752 => 1, 6753 => 2, 6754 => 3,
6756 => 1, 6757 => 2, 6759 => 1, 6760 => 2, 6761 => 1, 6762 => 2, 6763 => 3, 6764
=> 1, 6765 => 2, 6766 => 1, 6767 => 2, 6768 => 1, 6769 => 2, 6770 => 3, 6771 => 1,
6772 => 2, 6773 => 1, 6774 => 2, 6775 => 1, 6776 => 2, 6777 => 1, 6778 => 2, 6779
=> 1, 6780 => 2, 6803 => 1, 6804 => 2, 6820 => 1, 6821 => 2, 6825 => 1, 6826 => 2,
6827 => 3, 6828 => 1, 6829 => 2, 6830 => 1, 6831 => 2, 6832 => 3, 6833 => 1, 6834
=> 2, 6836 => 1, 6837 => 2, 6838 => 3, 6839 => 1, 6840 => 2, 6845 => 1, 6846 => 2,
6848 => 1, 6849 => 2, 6857 => 1, 6858 => 2, 6863 => 1, 6864 => 2, 6865 => 3, 6866
=> 1, 6867 => 2, 6887 => 1, 6888 => 2, 6889 => 1, 6890 => 2, 6891 => 1, 6892 => 2,
6893 => 1, 6894 => 2, 6895 => 1, 6896 => 2
    ];

    if(array_key_exists($id, $tex_position_on_page_array)){
```

```
        $tex_position_on_page = $tex_position_on_page_array[$id];
    }
    else{
        $tex_position_on_page = NULL;
    }

    //Here, we need an array that combines Mercure issues, the periodicals to
which they belong and their reference digital copies. This data comes from the
Drupal database. See documentation.

    include 'iss_peri_dig_cop_array.php';

    //Using this array, we can link most of the enigmas to the issues of the
Mercure in the Drupal database by comparing the URLs of their digital copies.

    if(stripos($lien, 'google') !== false){
        $id_url_enigma = explode('id=', $lien);
        $id_url_enigma = explode('&', $id_url_enigma[1]);
        $id_url_enigma = $id_url_enigma[0];
    }
    elseif(stripos($lien, 'gallica') !== false){
        $id_url_enigma = explode('ark:/', $lien);
        $id_url_enigma = explode('/', $id_url_enigma[1]);
        $id_url_enigma = $id_url_enigma[0].'/'.$id_url_enigma[1];
    }
    elseif(stripos($lien, 'reader') !== false){
        $id_url_enigma = explode('/bsb', $lien);
        $id_url_enigma = explode('_', $id_url_enigma[1]);
        $id_url_enigma = $id_url_enigma[0];
    }
    elseif(stripos($lien, 'handle') !== false){
        $id_url_enigma = explode('/nyp.', $lien);
        $id_url_enigma = explode('?', $id_url_enigma[1]);
        $id_url_enigma = $id_url_enigma[0];
    }
    else{
        $id_url_enigma = NULL;
    }

    $iss_peri_exclude_array = array('1097012', '1104066', '1104181', '1104502',
'1104537', '1350967'); //NIDs of periodicals that don't include enigmas: Affaires
du temps, Journal politique de Bruxelles, Mercure historique et politique de
Bruxelles, Mercure historique et politique de l'Europe, L'Aviseur national and
Mercure étranger.

    foreach($iss_peri_dig_cop_array as $iss_peri_dig_cop_value){

        if(in_array($iss_peri_dig_cop_value['periodical_id'],
$iss_peri_exclude_array)){
        }
        else{

            if(stripos($iss_peri_dig_cop_value['dig_cop_url'], 'google') !==
false){
                $id_url_dig_cop = explode('id=',
$iss_peri_dig_cop_value['dig_cop_url']);
                $id_url_dig_cop = explode('&', $id_url_dig_cop[1]);
                $id_url_dig_cop = $id_url_dig_cop[0];
            }
```

```
            elseif(stripos($iss_peri_dig_cop_value['dig_cop_url'], 'gallica') !==
false){
                $id_url_dig_cop = explode('ark:/',
$iss_peri_dig_cop_value['dig_cop_url']);
                $id_url_dig_cop = explode('/', $id_url_dig_cop[1]);
                $id_url_dig_cop = $id_url_dig_cop[0].'/'.$id_url_dig_cop[1];
            }
            elseif(stripos($iss_peri_dig_cop_value['dig_cop_url'], 'mdz-nbn') !==
false){
                $id_url_dig_cop = explode(':bsb',
$iss_peri_dig_cop_value['dig_cop_url']);
                $id_url_dig_cop = $id_url_dig_cop[1];
            }
            elseif(stripos($iss_peri_dig_cop_value['dig_cop_url'], 'handle') !==
false){
                $id_url_dig_cop = explode('/nyp.',
$iss_peri_dig_cop_value['dig_cop_url']);
                $id_url_dig_cop = explode('?', $id_url_dig_cop[1]);
                $id_url_dig_cop = $id_url_dig_cop[0];
            }
            else{
                $id_url_dig_cop = NULL;
            }

            if(($id_url_enigma == $id_url_dig_cop) && ($date_volume ==
$iss_peri_dig_cop_value['issue_all_date_gregorian_calendar'])){
                if((($numero_volume !== '0') && (($numero_volume ==
$iss_peri_dig_cop_value['issue_volume']) || ($numero_volume ==
$iss_peri_dig_cop_value['issue_number']))) || (($numero_volume == '0') &&
($iss_peri_dig_cop_value['issue_volume'] == '') &&
($iss_peri_dig_cop_value['issue_number'] == ''))){
                    $url_matches = 1;
                    $issue_nid = $iss_peri_dig_cop_value['issue_nid'];
                }

            }

        }

    }

    //Create field all_start_page from page_debut.

    $all_start_page = $page_debut;

    //Create field all_end_page from page_fin.

    $all_end_page = $page_fin;

    //Create fields all_start_page_file and all_end_page_file.

    include('enigma_pdf_pages_array.php'); //See documentation.

    if(isset($enigma_pdf_pages_array[$id])){
        $all_start_page_file = $enigma_pdf_pages_array[$id]['pdf_start_page']-20;
        $all_end_page_file = $enigma_pdf_pages_array[$id]['pdf_end_page']-20;
    }
    if($year_check == 1724 || $year_check == 1680 || $year_check == 1679 ||
$year_check == 1678){
```

```
        $all_start_page_file = $all_start_page_file+20;
        $all_end_page_file = $all_end_page_file+20;
    }

    //Exceptions for all_start_page, all_start_page_file, all_end_page and
all_end_page_file.

    if($id == 270){
        $all_start_page = 17;
        $all_start_page_file = 27;
        $all_end_page = 20;
        $all_end_page_file = 30;
    }
    elseif($id == 3539){
        $all_start_page = 85;
        $all_start_page_file = 105;
        $all_end_page = 85;
        $all_end_page_file = 105;
    }
    elseif($id == 3540){
        $all_start_page = 331;
        $all_start_page_file = 341;
        $all_end_page = 331;
        $all_end_page_file = 341;
    }
    elseif($id == 3541){
        $all_start_page = 256;
        $all_start_page_file = 272;
        $all_end_page = 259;
        $all_end_page_file = 275;
    }
    elseif($id == 3542){
        $all_start_page_file = 264;
        $all_end_page_file = 266;
    }
    elseif($id == 3543){
        $all_start_page_file = 281;
        $all_end_page_file = 284;
    }
    elseif($id == 3544){
        $all_start_page_file = 285;
        $all_end_page_file = 285;
    }
    elseif($id == 3547){
        $all_start_page_file = 398;
        $all_end_page_file = 399;
    }
    elseif($id == 3548){
        $all_start_page_file = 399;
        $all_end_page_file = 399;
    }
    elseif($id == 3549){
        $all_start_page = 407;
        $all_start_page_file = 446;
        $all_end_page = 408;
        $all_end_page_file = 447;
    }
    elseif($id == 3550){
        $all_start_page = 408;
```

```
        $all_start_page_file = 447;
        $all_end_page = 408;
        $all_end_page_file = 447;
    }
    elseif($id == 3551){
        $all_start_page = 355;
        $all_start_page_file = 385;
        $all_end_page = 356;
        $all_end_page_file = 386;
    }
    elseif($id == 3552){
        $all_start_page = 357;
        $all_start_page_file = 387;
        $all_end_page = 357;
        $all_end_page_file = 387;
    }
    elseif($id == 3559){
        $all_start_page = 352;
        $all_start_page_file = 376;
        $all_end_page = 353;
        $all_end_page_file = 377;
    }
    elseif($id == 3560){
        $all_start_page = 353;
        $all_start_page_file = 377;
        $all_end_page = 354;
        $all_end_page_file = 378;
    }
    elseif($id == 3561){
        $all_start_page = 349;
        $all_start_page_file = 370;
        $all_end_page = 349;
        $all_end_page_file = 370;
    }
    elseif($id == 3562){
        $all_start_page = 350;
        $all_start_page_file = 371;
        $all_end_page = 351;
        $all_end_page_file = 372;
    }
    elseif($id == 3563){
        $all_start_page = 356;
        $all_start_page_file = 383;
        $all_end_page = 357;
        $all_end_page_file = 384;
    }
    elseif($id == 3564){
        $all_start_page = 357;
        $all_start_page_file = 384;
        $all_end_page = 358;
        $all_end_page_file = 385;
    }
    elseif($id == 3565){
        $all_start_page = 289;
        $all_start_page_file = 324;
        $all_end_page = 290;
        $all_end_page_file = 325;
    }
    elseif($id == 3566){
```

```
        $all_start_page = 290;
        $all_start_page_file = 325;
        $all_end_page = 291;
        $all_end_page_file = 326;
    }
    elseif($id == 4078){
        $all_start_page_file = 277;
        $all_end_page_file = 280;
    }
    elseif($id == 4079){
        $all_start_page_file = 332;
        $all_end_page_file = 337;
    }
    elseif($id == 4080){
        $all_start_page_file = 338;
        $all_end_page_file = 338;
    }
    elseif($id == 4081){
        $all_start_page_file = 301;
        $all_end_page_file = 305;
    }
    elseif($id == 4082){
        $all_start_page_file = 340;
        $all_end_page_file = 340;
    }
    elseif($id == 4083){
        $all_start_page_file = 340;
        $all_end_page_file = 342;
    }
    elseif($id == 4084){
        $all_start_page_file = 343;
        $all_end_page_file = 344;
    }
    elseif($id == 4085){
        $all_start_page_file = 233;
        $all_end_page_file = 234;
    }
    elseif($id == 4086){
        $all_start_page_file = 234;
        $all_end_page_file = 235;
    }
    elseif($id == 4087){
        $all_start_page_file = 235;
        $all_end_page_file = 235;
    }
    elseif($id == 4088){
        $all_start_page_file = 228;
        $all_end_page_file = 229;
    }
    elseif($id == 4091){
        $all_start_page_file = 210;
        $all_end_page_file = 213;
    }
    elseif($id == 4092){
        $all_start_page_file = 213;
        $all_end_page_file = 215;
    }
    elseif($id == 4093){
        $all_start_page_file = 224;
```

```
        $all_end_page_file = 226;
    }
    elseif($id == 4094){
        $all_start_page_file = 231;
        $all_end_page_file = 234;
    }
    elseif($id == 4095){
        $all_start_page_file = 196;
        $all_end_page_file = 198;
    }
    elseif($id == 4096){
        $all_start_page_file = 166;
        $all_end_page_file = 167;
    }
    elseif($id == 4097){
        $all_start_page_file = 169;
        $all_end_page_file = 170;
    }
    elseif($id == 4098){
        $all_start_page_file = 186;
        $all_end_page_file = 187;
    }
    elseif($id == 4099){
        $all_start_page_file = 326;
        $all_end_page_file = 327;
    }
    elseif($id == 4100){
        $all_start_page_file = 328;
        $all_end_page_file = 328;
    }
    elseif($id == 4101){
        $all_start_page_file = 172;
        $all_end_page_file = 173;
    }
    elseif($id == 4102){
        $all_start_page_file = 319;
        $all_end_page_file = 321;
    }
    elseif($id == 4103){
        $all_start_page_file = 93;
        $all_end_page_file = 93;
    }
    elseif($id == 4104){
        $all_start_page_file = 94;
        $all_end_page_file = 94;
    }
    elseif($id == 4105){
        $all_start_page_file = 267;
        $all_end_page_file = 269;
    }
    elseif($id == 4106){
        $all_start_page_file = 98;
        $all_end_page_file = 101;
    }
    elseif($id == 4107){
        $all_start_page_file = 213;
        $all_end_page_file = 215;
    }
    elseif($id == 4108){
```

```
        $all_start_page_file = 218;
        $all_end_page_file = 222;
    }
    elseif($id == 4109){
        $all_start_page_file = 198;
        $all_end_page_file = 201;
    }
    elseif($id == 4110){
        $all_start_page_file = 184;
        $all_end_page_file = 186;
    }
    elseif($id == 4111){
        $all_start_page_file = 212;
        $all_end_page_file = 215;
    }
    elseif($id == 4112){
        $all_start_page_file = 138;
        $all_end_page_file = 141;
    }
    elseif($id == 4113){
        $all_start_page_file = 250;
        $all_end_page_file = 252;
    }
    elseif($id == 4118){
        $all_start_page_file = 115;
        $all_end_page_file = 116;
    }
    elseif($id == 4119){
        $all_start_page_file = 117;
        $all_end_page_file = 118;
    }
    elseif($id == 4126){
        $all_start_page_file = 114;
        $all_end_page_file = 115;
    }
    elseif($id == 4127){
        $all_start_page_file = 227;
        $all_end_page_file = 229;
    }
    elseif($id == 4128){
        $all_start_page_file = 98;
        $all_end_page_file = 101;
    }
    elseif($id == 4129){
        $all_start_page_file = 104;
        $all_end_page_file = 106;
    }
    elseif($id == 4130){
        $all_start_page_file = 61;
        $all_end_page_file = 63;
    }
    elseif($id == 4131){
        $all_start_page_file = 76;
        $all_end_page_file = 78;
    }
    elseif($id == 4132){
        $all_start_page_file = 69;
        $all_end_page_file = 71;
    }
```

```
    elseif($id == 4133){
        $all_start_page_file = 168;
        $all_end_page_file = 169;
    }
    elseif($id == 4138){
        $all_start_page_file = 136;
        $all_end_page_file = 138;
    }
    elseif($id == 4139){
        $all_start_page_file = 138;
        $all_end_page_file = 139;
    }
    elseif($id == 4140){
        $all_start_page_file = 37;
        $all_end_page_file = 39;
    }
    elseif($id == 4141){
        $all_start_page_file = 40;
        $all_end_page_file = 43;
    }
    elseif($id == 4144){
        $all_start_page_file = 137;
        $all_end_page_file = 139;
    }
    elseif($id == 4145){
        $all_start_page_file = 139;
        $all_end_page_file = 142;
    }
    elseif($id == 4146){
        $all_start_page_file = 27;
        $all_end_page_file = 28;
    }
    elseif($id == 4147){
        $all_start_page_file = 28;
        $all_end_page_file = 30;
    }
    elseif($id == 4148){
        $all_start_page_file = 134;
        $all_end_page_file = 137;
    }
    elseif($id == 4149){
        $all_start_page_file = 138;
        $all_end_page_file = 140;
    }
    elseif($id == 4152){
        $all_start_page_file = 315;
        $all_end_page_file = 316;
    }
    elseif($id == 4153){
        $all_start_page_file = 341;
        $all_end_page_file = 345;
    }
    elseif($id == 312){
        $all_start_page_file = 355;
        $all_end_page_file = 358;
    }
    elseif($id == 313){
        $all_start_page_file = 358;
        $all_end_page_file = 359;
```

```
    }
    elseif($id == 314){
        $all_start_page_file = 332;
        $all_end_page_file = 332;
    }
    elseif($id == 330){
        $all_start_page_file = 107;
        $all_end_page_file = 107;
    }
    elseif($id == 331){
        $all_start_page_file = 108;
        $all_end_page_file = 108;
    }
    elseif($id == 332){
        $all_start_page_file = 332;
        $all_end_page_file = 332;
    }
    elseif($id == 333){
        $all_start_page_file = 332;
        $all_end_page_file = 334;
    }
    elseif($id == 350){
        $all_start_page_file = 116;
        $all_end_page_file = $all_start_page_file;
    }
    elseif($id == 351){
        $all_start_page_file = 117;
        $all_end_page_file = $all_start_page_file;
    }
    elseif($id == 352){
        $all_start_page_file = 118;
        $all_end_page_file = $all_start_page_file;
    }
    elseif($id == 353){
        $all_start_page_file = 325;
        $all_end_page_file = 326;
    }
    elseif($id == 354){
        $all_start_page_file = 326;
        $all_end_page_file = $all_start_page_file;
    }
    elseif($id == 355){
        $all_start_page_file = 326;
        $all_end_page_file = 327;
    }
    elseif($id == 356){
        $all_start_page_file = 327;
        $all_end_page_file = $all_start_page_file;
    }
    elseif($id == 357){
        $all_start_page_file = 541;
        $all_end_page_file = 542;
    }
    elseif($id == 358){
        $all_start_page_file = 542;
        $all_end_page_file = 544;
    }
    elseif($id == 359){
        $all_start_page_file = 544;
```

```
        $all_end_page_file = 545;
    }
    elseif($id == 376){
        $all_start_page_file = 328;
        $all_end_page_file = 329;
    }
    elseif($id == 377){
        $all_start_page_file = 329;
        $all_end_page_file = 331;
    }
    elseif($id == 378){
        $all_start_page_file = 332;
        $all_end_page_file = $all_start_page_file;
    }
    elseif($id == 379){
        $all_start_page_file = 332;
        $all_end_page_file = 333;
    }
    elseif($id == 389){
        $all_start_page_file = 332;
        $all_end_page_file = 333;
    }
    elseif($id == 390){
        $all_start_page_file = 333;
        $all_end_page_file = 334;
    }
    elseif($id == 391){
        $all_start_page_file = 334;
        $all_end_page_file = 336;
    }
    elseif($id == 392){
        $all_start_page_file = 547;
        $all_end_page_file = 548;
    }
    elseif($id == 393){
        $all_start_page_file = 548;
        $all_end_page_file = $all_start_page_file;
    }
    elseif($id == 394){
        $all_start_page_file = 548;
        $all_end_page_file = $all_start_page_file;
    }
    elseif($id == 395){
        $all_start_page_file = 549;
        $all_end_page_file = $all_start_page_file;
    }
    elseif($id == 396){
        $all_start_page_file = 122;
        $all_end_page_file = 123;
    }
    elseif($id == 397){
        $all_start_page_file = 123;
        $all_end_page_file = 124;
    }
    elseif($id == 398){
        $all_start_page_file = 124;
        $all_end_page_file = $all_start_page_file;
    }
    elseif($id == 399){
```

```
        $all_start_page_file = 353;
        $all_end_page_file = 354;
    }
    elseif($id == 400){
        $all_start_page_file = 354;
        $all_end_page_file = 355;
    }
    elseif($id == 401){
        $all_start_page_file = 355;
        $all_end_page_file = 356;
    }
    elseif($id == 406){
        $all_start_page_file = 346;
        $all_end_page_file = 347;
    }
    elseif($id == 407){
        $all_start_page_file = 347;
        $all_end_page_file = $all_start_page_file;
    }
    elseif($id == 408){
        $all_start_page_file = 348;
        $all_end_page_file = $all_start_page_file;
    }
    elseif($id == 409){
        $all_start_page_file = 352;
        $all_end_page_file = 354;
    }
    elseif($id == 438){
        $all_start_page_file = 319;
        $all_end_page_file = $all_start_page_file;
    }
    elseif($id == 439){
        $all_start_page_file = 319;
        $all_end_page_file = 321;
    }
    elseif($id == 440){
        $all_start_page_file = 321;
        $all_end_page_file = 322;
    }
    elseif($id == 441){
        $all_start_page_file = 322;
        $all_end_page_file = $all_start_page_file;
    }
    elseif($id == 442){
        $all_start_page_file = 323;
        $all_end_page_file = $all_start_page_file;
    }
    elseif($id == 443){
        $all_start_page_file = 534;
        $all_end_page_file = $all_start_page_file;
    }
    elseif($id == 444){
        $all_start_page_file = 534;
        $all_end_page_file = 536;
    }
    elseif($id == 445){
        $all_start_page_file = 536;
        $all_end_page_file = 537;
    }
```

```
    elseif($id == 454){
        $all_start_page_file = 105;
        $all_end_page_file = $all_start_page_file;
    }
    elseif($id == 455){
        $all_start_page_file = 105;
        $all_end_page_file = 106;
    }
    elseif($id == 456){
        $all_start_page_file = 106;
        $all_end_page_file = $all_start_page_file;
    }
    elseif($id == 457){
        $all_start_page_file = 106;
        $all_end_page_file = 107;
    }
    elseif($id == 458){
        $all_start_page_file = 107;
        $all_end_page_file = $all_start_page_file;
    }
    elseif($id == 459){
        $all_start_page_file = 341;
        $all_end_page_file = $all_start_page_file;
    }
    elseif($id == 460){
        $all_start_page_file = 342;
        $all_end_page_file = $all_start_page_file;
    }
    elseif($id == 461){
        $all_start_page_file = 342;
        $all_end_page_file = $all_start_page_file;
    }
    elseif($id == 462){
        $all_start_page_file = 343;
        $all_end_page_file = $all_start_page_file;
    }
    elseif($id == 463){
        $all_start_page_file = 343;
        $all_end_page_file = $all_start_page_file;
    }
    elseif($id == 464){
        $all_start_page_file = 343;
        $all_end_page_file = 344;
    }
    elseif($id == 472){
        $all_start_page_file = 114;
        $all_end_page_file = $all_start_page_file;
    }
    elseif($id == 473){
        $all_start_page_file = 114;
        $all_end_page_file = 115;
    }
    elseif($id == 474){
        $all_start_page_file = 115;
        $all_end_page_file = 117;
    }
    elseif($id == 475){
        $all_start_page_file = 117;
        $all_end_page_file = 119;
```

```
    }
    elseif($id == 476){
        $all_start_page_file = 119;
        $all_end_page_file = 120;
    }
    elseif($id == 477){
        $all_start_page_file = 120;
        $all_end_page_file = $all_start_page_file;
    }
    elseif($id == 478){
        $all_start_page_file = 344;
        $all_end_page_file = 345;
    }
    elseif($id == 479){
        $all_start_page_file = 345;
        $all_end_page_file = $all_start_page_file;
    }
    elseif($id == 480){
        $all_start_page_file = 345;
        $all_end_page_file = 347;
    }
    elseif($id == 481){
        $all_start_page_file = 347;
        $all_end_page_file = $all_start_page_file;
    }
    elseif($id == 482){
        $all_start_page_file = 347;
        $all_end_page_file = 348;
    }
    elseif($id == 483){
        $all_start_page_file = 349;
        $all_end_page_file = $all_start_page_file;
    }
    elseif($id == 540){
        $all_start_page_file = 333;
        $all_end_page_file = $all_start_page_file;
    }
    elseif($id == 541){
        $all_start_page_file = 333;
        $all_end_page_file = $all_start_page_file;
    }
    elseif($id == 542){
        $all_start_page_file = 334;
        $all_end_page_file = 335;
    }
    elseif($id == 543){
        $all_start_page_file = 335;
        $all_end_page_file = 336;
    }
    elseif($id == 544){
        $all_start_page_file = 569;
        $all_end_page_file = 570;
    }
    elseif($id == 545){
        $all_start_page_file = 570;
        $all_end_page_file = 572;
    }
    elseif($id == 546){
        $all_start_page_file = 572;
```

```
        $all_end_page_file = 574;
    }
    elseif($id == 553){
        $all_start_page_file = 110;
        $all_end_page_file = $all_start_page_file;
    }
    elseif($id == 554){
        $all_start_page_file = 111;
        $all_end_page_file = $all_start_page_file;
    }
    elseif($id == 555){
        $all_start_page_file = 112;
        $all_end_page_file = $all_start_page_file;
    }
    elseif($id == 556){
        $all_start_page_file = 330;
        $all_end_page_file = 331;
    }
    elseif($id == 557){
        $all_start_page_file = 331;
        $all_end_page_file = $all_start_page_file;
    }
    elseif($id == 558){
        $all_start_page_file = 332;
        $all_end_page_file = 333;
    }
    elseif($id == 559){
        $all_start_page_file = 333;
        $all_end_page_file = 335;
    }
    elseif($id == 208){
        $all_start_page_file = 112;
        $all_end_page_file = 113;
    }
    elseif($id == 209){
        $all_start_page_file = 113;
        $all_end_page_file = 113;
    }
    elseif($id == 210){
        $all_start_page_file = 114;
        $all_end_page_file = 115;
    }
    elseif($id == 1148){
        $all_start_page_file = 130;
        $all_end_page_file = 131;
    }
    elseif($id == 1149){
        $all_start_page_file = 131;
        $all_end_page_file = 132;
    }
    elseif($id == 1150){
        $all_start_page_file = 132;
        $all_end_page_file = 133;
    }
    elseif($id == 1151){
        $all_start_page_file = 133;
        $all_end_page_file = 133;
    }
    elseif($id == 1155){
```

```
        $all_start_page_file = 572;
        $all_end_page_file = 573;
    }
    elseif($id == 1156){
        $all_start_page_file = 573;
        $all_end_page_file = 573;
    }
    elseif($id == 1157){
        $all_start_page_file = 574;
        $all_end_page_file = 575;
    }
    elseif($id == 1158){
        $all_start_page_file = 575;
        $all_end_page_file = 576;
    }
    elseif($id == 1808){
        $all_start_page_file = 341;
        $all_end_page_file = 342;
    }
    elseif($id == 1809){
        $all_start_page_file = 342;
        $all_end_page_file = 343;
    }
    elseif($id == 1807){
        $all_start_page_file = 340;
        $all_end_page_file = 341;
    }
    elseif($id == 1806){
        $all_start_page_file = 339;
        $all_end_page_file = 340;
    }
    elseif($id == 3609){
        $all_start_page_file = 369;
        $all_end_page_file = 370;
    }
    elseif($id == 3610){
        $all_start_page_file = 370;
        $all_end_page_file = 371;
    }
    elseif($id == 3615){
        $all_start_page_file = 356;
        $all_end_page_file = 357;
    }
    elseif($id == 4070){
        $all_start_page_file = 231;
        $all_end_page_file = 233;
    }
    elseif($id == 4071){
        $all_start_page_file = 371;
        $all_end_page_file = 374;
    }
    elseif($id == 4073){
        $all_start_page_file = 345;
        $all_end_page_file = 346;
    }
    elseif($id == 4361){
        $all_start_page_file = 120;
        $all_end_page_file = 121;
    }
```

```
    elseif($id == 4362){
        $all_start_page_file = 121;
        $all_end_page_file = 121;
    }
    elseif($id == 4363){
        $all_start_page_file = 122;
        $all_end_page_file = 122;
    }
    elseif($id == 5189){
        $all_start_page_file = 16;
        $all_end_page_file = 16;
    }
    elseif($id == 5190){
        $all_start_page_file = 16;
        $all_end_page_file = 16;
    }
    elseif($id == 5191){
        $all_start_page_file = 17;
        $all_end_page_file = 17;
    }
    elseif($id == 4576){
        $all_start_page_file = 109;
        $all_end_page_file = 110;
    }

    if($all_end_page == '' || $all_end_page == NULL){
        $all_end_page = $all_start_page;
    }
    if($all_end_page_file == '' || $all_end_page_file == NULL){
        $all_end_page_file = $all_start_page_file;
    }

    //Create field tex_ocr. (Remains empty.)

    $tex_ocr = '';

    //Create field tex_keyword from reponse_normalisee.

    include('taxonomy_term_array.php'); //See documentation.

    //Exceptions.

    if($reponse_normalisee == 'Alliance [bague]'){
        $reponse_normalisee = 'Alliance';
    }
    if($reponse_normalisee == 'Ana [Anah]'){
        $reponse_normalisee = 'Ana';
    }
    if($reponse_normalisee == 'Ancre [et encre]'){
        $reponse_normalisee = 'Ancre';
    }
    if($reponse_normalisee == 'Angelus'){
        $reponse_normalisee = 'Angélus';
    }
    if($reponse_normalisee == 'Arthémise [Artémise II]'){
        $reponse_normalisee = 'Artémise';
    }
    if($reponse_normalisee == 'Avenio [Avignon]'){
        $reponse_normalisee = 'Avenio';
```

```
    }
    if($reponse_normalisee == 'Avoie [sainte Avoye]'){
        $reponse_normalisee = 'Avoye';
    }
    if($reponse_normalisee == 'Balle [de fusil]'){
        $reponse_normalisee = 'Balle';
    }
    if($reponse_normalisee == 'Barbarina [Barbara Campanini]'){
        $reponse_normalisee = 'Barbarina';
    }
    if($reponse_normalisee == 'Bâtonnier (chef de l\'ordre des avocats)'){
        $reponse_normalisee = 'Bâtonnier';
    }
    if($reponse_normalisee == 'Belle Poule, frégate'){
        $reponse_normalisee = 'Belle Poule';
    }
    if($reponse_normalisee == 'Bémol et le dièse'){
        $reponse_normalisee = 'Bémol et dièse';
    }
    if($reponse_normalisee == 'Berne [brimade]'){
        $reponse_normalisee = 'Berne';
    }
    if($reponse_normalisee == 'Bourdaloue [Louis Bourdaloue]'){
        $reponse_normalisee = 'Bourdaloue';
    }
    if($reponse_normalisee == 'Bouteille de savon [bulle de savon]'){
        $reponse_normalisee = 'Bouteille de savon';
    }
    if($reponse_normalisee == 'Brandebourg [parure]'){
        $reponse_normalisee = 'Brandebourg';
    }
    if($reponse_normalisee == 'Brid\'oison [personnage du Mariage de Figaro]'){
        $reponse_normalisee = 'Brid\'oison';
    }
    if($reponse_normalisee == 'Camargo [Marie-Anne de Camargo]'){
        $reponse_normalisee = 'Camargo';
    }
    if($reponse_normalisee == 'Cartouche [Louis Dominique Garthausen]'){
        $reponse_normalisee = 'Cartouche';
    }
    if($reponse_normalisee == 'Chastillon [Châtillon]'){
        $reponse_normalisee = 'Châtillon';
    }
    if($reponse_normalisee == 'Col [cou]'){
        $reponse_normalisee = 'Col';
    }
    if($reponse_normalisee == 'Conciergerie [prison]'){
        $reponse_normalisee = 'Conciergerie';
    }
    if($reponse_normalisee == 'Corps [de tailleur]'){
        $reponse_normalisee = 'Corps';
    }
    if($reponse_normalisee == 'Corps et l\'âme'){
        $reponse_normalisee = 'Corps et âme';
    }
    if($reponse_normalisee == 'Cuillère et la fourchette'){
        $reponse_normalisee = 'Cuillère et fourchette';
    }
    if($reponse_normalisee == 'Demoiselle de paveur [outil]'){
```

```
        $reponse_normalisee = 'Demoiselle de paveur';
    }
    if($reponse_normalisee == 'Donneau [de Visé]'){
        $reponse_normalisee = 'Donneau de Visé';
    }
    if($reponse_normalisee == 'Duchesse [meuble]'){
        $reponse_normalisee = 'Duchesse';
    }
    if($reponse_normalisee == 'Émigrant (jeu)'){
        $reponse_normalisee = 'Émigrant';
    }
    if($reponse_normalisee == 'Énigme et le Logogriphe'){
        $reponse_normalisee = 'Énigme et logogriphe';
    }
    if($reponse_normalisee == 'Énigme et le logogriphe du Mercure de France'){
        $reponse_normalisee = 'Énigme et logogriphe du Mercure de France';
    }
    if($reponse_normalisee == 'Falot [lanterne et personne]'){
        $reponse_normalisee = 'Falot';
    }
    if($reponse_normalisee == 'Faux [outil]'){
        $reponse_normalisee = 'Faux';
    }
    if($reponse_normalisee == 'Fleur de lys'){
        $reponse_normalisee = 'Fleur de lis';
    }
    if($reponse_normalisee == 'Flûte ou la musette'){
        $reponse_normalisee = 'Flûte ou musette';
    }
    if($reponse_normalisee == 'Flux et le reflux'){
        $reponse_normalisee = 'Flux et reflux';
    }
    if($reponse_normalisee == 'Fusil [aiguisoir]'){
        $reponse_normalisee = 'Fusil';
    }
    if($reponse_normalisee == 'Gage touché [jeu]'){
        $reponse_normalisee = 'Gage touché';
    }
    if($reponse_normalisee == 'Gaudron [goudron]'){
        $reponse_normalisee = 'Goudron';
    }
    if($reponse_normalisee == 'Gertrude [coiffure]'){
        $reponse_normalisee = 'Gertrude';
    }
    if($reponse_normalisee == 'Gigoudenne [gigoudaine]'){
        $reponse_normalisee = 'Gigoudaine';
    }
    if($reponse_normalisee == 'Gorge [sein d\'une femme]'){
        $reponse_normalisee = 'Gorge';
    }
    if($reponse_normalisee == 'Hérisson [coiffure]'){
        $reponse_normalisee = 'Hérisson';
    }
    if($reponse_normalisee == 'Immortelle [fleur]'){
        $reponse_normalisee = 'Immortelle';
    }
    if($reponse_normalisee == 'Janot [personnage de comédie]'){
        $reponse_normalisee = 'Janot';
    }
```

```
    if($reponse_normalisee == 'Jérôme Pointu [comédie d\'Alexandre-Louis-Bertrand
Robineau]'){
        $reponse_normalisee = 'Jérôme Pointu';
    }
    if($reponse_normalisee == 'Jour et la nuit'){
        $reponse_normalisee = 'Jour et nuit';
    }
    if($reponse_normalisee == 'La Folle Journée [Le Mariage de Figaro]'){
        $reponse_normalisee = 'La Folle Journée';
    }
    if($reponse_normalisee == 'La Harpe [Jean-François de La Harpe (1739-1803)]'){
        $reponse_normalisee = 'La Harpe';
    }
    if($reponse_normalisee == 'Lamie (monstre marin)'){
        $reponse_normalisee = 'Lamie';
    }
    if($reponse_normalisee == 'Les deux Comédies [à Paris]'){
        $reponse_normalisee = 'Les deux Comédies';
    }
    if($reponse_normalisee == 'Loi et la coutume'){
        $reponse_normalisee = 'Loi et coutume';
    }
    if($reponse_normalisee == 'M. CCC. VVVVVV. IIII. [an 1334]'){
        $reponse_normalisee = 'Année 1334';
    }
    if($reponse_normalisee == 'Madame Dugazon [Louise-Rosalie Lefebvre (1755-
1821)]'){
        $reponse_normalisee = 'Madame Dugazon';
    }
    if($reponse_normalisee == 'Maillard [Marie-Thérèse Davoux (1766-1818)]'){
        $reponse_normalisee = 'Maillard';
    }
    if($reponse_normalisee == 'Malle (du courrier)'){
        $reponse_normalisee = 'Malle';
    }
    if($reponse_normalisee == 'Matadors [jeu de l\'hombre]'){
        $reponse_normalisee = 'Matadors';
    }
    if($reponse_normalisee == 'Mâtin [chien]'){
        $reponse_normalisee = 'Mâtin';
    }
    if($reponse_normalisee == 'Mémoire [écrit]'){
        $reponse_normalisee = 'Mémoire';
    }
    if($reponse_normalisee == 'Mesmer [Franz Anton Mesmer (1734-1815)]'){
        $reponse_normalisee = 'Mesmer';
    }
    if($reponse_normalisee == 'Naissance de Monseigneur le duc de Bretagne'){
        $reponse_normalisee = 'Duc de Bretagne';
    }
    if($reponse_normalisee == 'Nombre de lettres de chaque mot'){
        $reponse_normalisee = 'Nombre de lettres';
    }
    if($reponse_normalisee == 'Nuit et le jour'){
        $reponse_normalisee = 'Nuit et jour';
    }
    if($reponse_normalisee == 'Orpin [minéral (orpiment) et plante]'){
        $reponse_normalisee = 'Orpin';
    }
```

```
    if($reponse_normalisee == 'Panckoucke [Charles-Joseph Panckoucke (1736-
1798)]'){
        $reponse_normalisee = 'Panckoucke';
    }
    if($reponse_normalisee == 'Parc [de brebis]'){
        $reponse_normalisee = 'Parc';
    }
    if($reponse_normalisee == 'Pêche [fruit]'){
        $reponse_normalisee = 'Pêche';
    }
    if($reponse_normalisee == 'Perronet [Jean-Rodolphe Perronet (1708-1794)]'){
        $reponse_normalisee = 'Perronet';
    }
    if($reponse_normalisee == 'Petit point [broderie]'){
        $reponse_normalisee = 'Petit point';
    }
    if($reponse_normalisee == 'Pied de roi [mesure]'){
        $reponse_normalisee = 'Pied de roi';
    }
    if($reponse_normalisee == 'Polignac [Melchior de Polignac (1661-1741)]'){
        $reponse_normalisee = 'Polignac';
    }
    if($reponse_normalisee == 'Porte de pierre et la porte de bois'){
        $reponse_normalisee = 'Porte de pierre et porte de bois';
    }
    if($reponse_normalisee == 'Poulet [poussin dans l\'oeuf]'){
        $reponse_normalisee = 'Poulet';
    }
    if($reponse_normalisee == 'Pouls ou l\'artère'){
        $reponse_normalisee = 'Pouls ou artère';
    }
    if($reponse_normalisee == 'Préville [Pierre-Louis Dubus (1721-1799)]'){
        $reponse_normalisee = 'Préville';
    }
    if($reponse_normalisee == 'Prise [de tabac]'){
        $reponse_normalisee = 'Prise de tabac';
    }
    if($reponse_normalisee == 'Quinola [valet de coeur au jeu du reversi]'){
        $reponse_normalisee = 'Quinola';
    }
    if($reponse_normalisee == 'Quinolas [valets de coeur au jeu du reversi]'){
        $reponse_normalisee = 'Quinolas';
    }
    if($reponse_normalisee == 'Quinte de valet au jeu du piquet, contre une
quatrième au roi'){
        $reponse_normalisee = 'Quinte de valet au jeu du piquet';
    }
    if($reponse_normalisee == 'Quinte major [majeure]'){
        $reponse_normalisee = 'Quinte majeure';
    }
    if($reponse_normalisee == 'Rossinante [cheval de Don Quichotte]'){
        $reponse_normalisee = 'Rossinante';
    }
    if($reponse_normalisee == 'Rouge [fard]'){
        $reponse_normalisee = 'Rouge';
    }
    if($reponse_normalisee == 'S. M. D. [Sa Majesté Danoise, Christian VII de
Danemark]'){
        $reponse_normalisee = 'Sa Majesté danoise';
```

```
    }
    if($reponse_normalisee == 'Sable [sablier]'){
        $reponse_normalisee = 'Sable';
    }
    if($reponse_normalisee == 'Serein [brouillard]'){
        $reponse_normalisee = 'Serein';
    }
    if($reponse_normalisee == 'Son (meunerie)'){
        $reponse_normalisee = 'Son';
    }
    if($reponse_normalisee == 'Surtout [vêtement]'){
        $reponse_normalisee = 'Surtout';
    }
    if($reponse_normalisee == 'Tableau qu\'on expose tous les ans à Notre-Dame le
premier jour de mai'){
        $reponse_normalisee = 'Tableau';
    }
    if($reponse_normalisee == 'Tabouret et la sellette'){
        $reponse_normalisee = 'Tabouret et sellette';
    }
    if($reponse_normalisee == 'Tarare [opéra]'){
        $reponse_normalisee = 'Tarare';
    }
    if($reponse_normalisee == 'Tierce et la sixième au piquet'){
        $reponse_normalisee = 'Tierce et sixième au piquet';
    }
    if($reponse_normalisee == 'Tonton [toton]'){
        $reponse_normalisee = 'Toton';
    }
    if($reponse_normalisee == 'Torre [Giovanni Battista Torre (mort en 1780)]'){
        $reponse_normalisee = 'Torre';
    }
    if($reponse_normalisee == 'Totum [toton]'){
        $reponse_normalisee = 'Toton';
    }
    if($reponse_normalisee == 'Tour [mécanisme]'){
        $reponse_normalisee = 'Tour';
    }
    if($reponse_normalisee == 'Turenne [Henri de La Tour d\'Auvergne (1616-
1675)]'){
        $reponse_normalisee = 'Turenne';
    }
    if($reponse_normalisee == 'Ver [chenille]'){
        $reponse_normalisee = 'Ver';
    }
    if($reponse_normalisee == 'Vergennes [Charles Gravier de Vergennes (1719-
1787)]'){
        $reponse_normalisee = 'Vergennes';
    }
    if($reponse_normalisee == 'Verrouil [verrou]'){
        $reponse_normalisee = 'Verrou';
    }
    if($reponse_normalisee == 'Chine, niche et chien'){
        $reponse_normalisee = 'Chine';
    }
    if($reponse_normalisee == 'Écriteau de maison à louer, soit premier, soit
second appartement'){
        $reponse_normalisee = 'Écriteau de maison à louer';
    }
```

```
    if($reponse_normalisee == 'Énigme inexplicable, ou qui n\'a point de sens'){
        $reponse_normalisee = 'Énigme inexplicable';
    }

    $tex_keyword = '';

    foreach($taxonomy_term_array as $taxonomy_term_value){

        if($taxonomy_term_value['vid'] == 'keyword' &&
$taxonomy_term_value['name'] == $reponse_normalisee){
            $tex_keyword = $taxonomy_term_value['tid'];
            break;
        }

    }

    //Create field tex_domain from reponse_normalisee.

    $domain_array_arts = array("Cariatides", "Colonne", "Corniche", "Façade du
Louvre", "Fontainebleau", "Louvre", "Palais", "Peintre", "Peinture", "Pinceau",
"Pont", "Pont Neuf", "Portrait", "Portrait d'une maîtresse", "Soliveau", "Statue",
"Tableau", "Tuileries", "Versailles");

    $domain_array_belles_lettres = array("Académie française", "Aenigma", "Affiche
de la Comédie-Française", "Affiche des spectacles", "Almanach", "Almanach de
cabinet", "Almanach des Muses", "Anagramme", "Apologue", "Arlequin", "Auteur",
"Bon mot", "Bouts-rimés", "Brid'oison", "Centurie", "Charade", "Comédie",
"Compilateur", "Cothurne", "Cyclope", "Danseur de corde", "Danseur de cordes",
"Décors", "Donneau de Visé", "Drame", "Éloquence", "Énigme", "Énigme et
logogriphe", "Énigme et logogriphe du Mercure de France", "Énigme inexplicable",
"Épigramme", "Épitaphe", "Épithalame", "Errata", "Fable", "Gazette", "Guillemins
de Machault", "Heautontimoroumenos", "Henriade", "Héroïdes", "Homère", "Horatius",
"Impromptu", "Janot", "Jérôme Pointu", "Journal", "La Folle Journée", "La Harpe",
"Lecteur", "Lecture", "Les deux Comédies", "Logogriphe", "Logogriphe du Mercure",
"Machiniste", "Madame Dugazon", "Madrigal", "Marionnettes", "Mélange",
"Melpomène", "Mercure", "Mercure de France", "Mercure français", "Métaphore",
"Métromanie", "Molière", "Montesquieu", "Ninon", "Oraison", "Panckoucke",
"Parterre d'un spectacle", "Poésie", "Préface", "Préville", "Prose", "Prosopopée",
"Quatrain", "Rime", "Roman", "Rondeau", "Rossinante", "Satire", "Sixain",
"Sonnet", "Souffleur de la comédie", "Théâtre", "Tragédie", "Vers", "Vers
alexandrin", "Vers de Clément Marot", "Vers de la poésie", "Virgile", "Voltaire",
"Volupté", "Zaïre");

    $domain_array_education = array("Apprentissage", "Collège", "École",
"Écolier", "Ergo", "Précepteur");

    $domain_array_history = array("An mille sept cent soixante-quinze", "An mille
sept cent vingt-huit", "Année 1334", "Année 1790", "Antiquité", "Armoirie",
"Armoiries", "Artémise", "Blason", "Calendrier", "Clovis", "Coriolan", "Écusson",
"Généalogie", "Histoire", "Historien", "Maharbal", "Minos", "Minotaure",
"Mithridate", "Pyramide", "Richelieu", "Rohan", "Thébain", "Troie", "Vespasianus",
"Pharaon");

    $domain_array_human_knowledge = array("Bibliothèque", "Démonstration",
"Encyclopédie", "Science");

    $domain_array_law = array("Avocat", "Bail", "Bandit", "Banqueroute",
"Bâtonnier", "Bûcher", "Calomniateur", "Calomnie", "Casuel", "Célibat", "Censure",
"Champart", "Chicane", "Chiourme", "Cocuage", "Commissaire", "Conciergerie",
```

```
"Créancier", "Crédit", "Crime", "Criminel", "Débiteur", "Délit", "Duel", "Écu
faux", "Fausse monnaie", "Filou", "Galère", "Héritage", "Innocence",
"Interdiction", "Jurisprudence", "Lettre de change", "Loi et coutume", "Oui du
mariage", "Parlement", "Prison", "Procédure", "Procès", "Procureur",
"Récrimination", "Sellette", "Sentence", "Sergent", "Testament", "Viol");

    $domain_array_mathematics = array("Aire", "Algèbre", "Angle", "Arithmétique",
"Cercle", "Chiffre", "Chiffre six", "Cône", "Étoile", "Géographie", "Géométrie",
"Graphomètre", "Losange", "Nombre un", "Point mathématique", "Récapitulation",
"Sept", "Sphère et le Cylindre", "Superficie", "Trente", "Triangle", "Trois",
"Zéro");

    $domain_array_mechanical_arts = array("Agriculture", "Aiguille", "Aiguille du
cadran solaire", "Aiguille d'un cadran", "Alambic", "Ancre", "Anneau ou bague",
"Anneaux de rideaux de lit", "Anneaux d'un rideau de lit", "Antichambre",
"Architecte", "Architecture", "Armoire", "Arrosoir", "Assiette", "Âtre", "Aune",
"Bac", "Bagnolette", "Balance", "Balances", "Balancier", "Balcon", "Baldaquin",
"Ballot", "Balustre", "Banc", "Barbier", "Bas", "Bas au métier", "Bateau", "Bateau
échappé", "Battants d'une porte", "Battoir", "Battoir des lavandières", "Berceau",
"Bijouterie", "Birloir", "Biscuit", "Boisseau", "Boîte à perruque ou à cheveux",
"Bonbon", "Bondon", "Bonnet", "Bonnet carré", "Bottes fortes", "Boucher",
"Boucherie", "Bouchon", "Bouchon de bouteille", "Boucle du col", "Boucles de
cheveux et de souliers", "Boucles d'oreille", "Bougie", "Bouteille", "Bouton
d'habit", "Bracelet", "Broche", "Broche tournée par un chien", "Brochure",
"Broderie", "Brouette", "Bûcheron", "Cabriolet", "Cabriolets", "Cadran solaire",
"Cafetière", "Camion", "Canapé", "Canevas", "Canif", "Canne à lorgnette",
"Caractère d'imprimerie", "Carillonneur", "Carrosse", "Carte de géographie",
"Carte du monde", "Carte géographique", "Carton", "Cartouche d'une fusée volante",
"Casserole", "Cave", "Ceinture", "Cerf-volant", "Chaise", "Chaise à porteur",
"Chaise à porteurs", "Chaise de commodité", "Chaise percée", "Chaloupe",
"Chandelier", "Chandelle", "Chapeau", "Char", "Charette", "Charpente",
"Charpentier", "Charroi", "Charron", "Charrue", "Chaux", "Cheminée", "Chemise",
"Chenet", "Chenets", "Chocolat", "Cidre", "Cierge", "Cire à cacheter", "Cire
d'Espagne", "Cire et le cachet", "Ciseaux", "Citerne", "Cloche", "Cloison",
"Clou", "Clou à soulier", "Coche", "Coche d'eau", "Cocher", "Coffre", "Cognée",
"Coiffe", "Coin à fendre le bois", "Collier", "Colombier", "Commode", "Compas",
"Contre-coeur de la cheminée", "Copeau", "Corbeille", "Corbillon de l'oublieux",
"Corbillon d'oublies", "Corde", "Cordeau", "Cordier", "Cordon", "Cordonnier",
"Corps de baleine", "Correspondance", "Corsage", "Corset", "Coton", "Courtage",
"Coussin", "Couteau", "Couteau de table", "Coutelas", "Couverture d'un livre",
"Cravate", "Crayon ou pastel", "Crèche", "Crémaillère", "Crible", "Crochet",
"Crocheteur", "Dé à coudre", "Décrotteur de souliers", "Demoiselle de paveur",
"Dentelle", "Deux aiguilles du cadran", "Diamant monté", "Diligence de terre ou
d'eau", "Drap rayé", "Duchesse", "Eau de la reine de Hongrie", "Échanson",
"Écharpe", "Échelle", "Écheveau", "Écrin", "Écritoire", "Écrivain", "Émail",
"Enclume", "Encrier", "Enseigne", "Enseigne de maison", "Enseigne de marchand",
"Enseigne d'un cabaret", "Enseignes", "Enseignes de Paris", "Épingle",
"Épouvantail", "Escalier", "Escarboucle", "Essieu", "Éventail", "Fabrique",
"Facteur de lettres", "Farine", "Fauteuil", "Fauteuils", "Fenêtre", "Fer à
cheval", "Fer à repasser", "Fer à repasser le linge", "Feu d'artifice", "Fiacre",
"Forge", "Forgeron", "Fossoyeur", "Four", "Fournier", "Fourreur", "Fripier",
"Fumier", "Fuseau", "Fuseaux à dentelle", "Fusée", "Fusée volante", "Gant",
"Gants", "Glace de miroir", "Goudron", "Hache", "Hachette", "Hameçon", "Horloge",
"Horloger", "Imprimerie", "Imprimeur", "Industrie", "Jardin", "Laboratoire",
"Laboureur", "Lanterne magique", "Libraire", "Ligne à pêcher", "Lime", "Lunette",
"Lunette d'approche", "Lunettes", "Lunettes pour le nez", "Lustre", "Manège",
"Manufacture", "Marchand", "Marchandises", "Marteau", "Marteau d'une porte",
"Meunier", "Moisson", "Mongolfier", "Monnaie", "Monnayeur", "Montgolfier",
"Montre", "Montre à répétition", "Montre ou pendule", "Mortaise", "Moulin",
```

```
"Moulin à farine", "Moulin à vent", "Mousseline", "Notaire", "Orfèvre", "Oripeau",
"Oublieux", "Papier", "Paquebot", "Parchemin", "Pelotte à épingles", "Pendule",
"Perruque", "Pèse-liqueur", "Pioche", "Porcelaine", "Portier", "Poste", "Poste aux
lettres", "Potager", "Poulie", "Poulies", "Poutre", "Presse", "Pressoir", "Prote",
"Quenouille", "Ramoneur", "Râpe à tabac", "Réverbère", "Roues", "Rouet", "Savon",
"Savonnette", "Secrétaire", "Selle", "Selle à cheval", "Selle de cheval",
"Serrure", "Soc de charrue", "Soubrette", "Soupape", "Tabatière", "Tamis", "Tapis
de Turquie", "Tapisserie", "Tenaille", "Tonneau", "Trame", "Tranchet",
"Treillage", "Truelle", "Tuyau", "Union", "Vase", "Velours", "Vendange", "Verre",
"Verre à boire", "Verrou", "Vignoble", "Vin", "Vitre", "Voie de bois", "Voile",
"Voile de navire", "Voiture");

    $domain_array_medicine = array("Accouchée", "Accoucheur", "Apoplexie",
"Apothicaire", "Aveuglement", "Béquilles", "Cal", "Charlatan", "Chirurgien",
"Clystère", "Collyre", "Cor d'un pied", "Crampe", "Dolor", "Douleur", "Fièvre",
"Gale", "Hémorroïde", "Hôpital", "Hôtel-Dieu", "Hydrocéphale", "Hypocondriaque",
"Inoculation", "Insomnie", "Lancette", "Lavement", "Léthargie", "Maladie",
"Médecin", "Médecine", "Medicus", "Mélancolie", "Migraine", "Opérateur",
"Panaris", "Petite vérole", "Pilule", "Piqûre", "Pouls", "Pouls ou artère",
"Quinze-vingts", "Remède", "Respiration", "Rhume", "Sage-femme", "Sang", "Santé",
"Seringue", "Vapeur");

    $domain_array_music = array("Archet", "Archet d'un violon", "Barbarina",
"Basse", "Basse de viole", "Basson", "Bémol et dièse", "Bourrée", "Camargo",
"Carillon", "Chanson", "Chant", "Chevalet de violon", "Chevilles d'un instrument",
"Clarinette", "Clavecin", "Cor", "Corde d'un instrument", "Cordes d'un violon",
"Cornemuse", "Danse", "Fifre", "Flûte", "Flûte allemande", "Flûte ou musette",
"Flûte, flageolet, hautbois ou chalumeau", "Guitare", "Harmonia", "Harmonie",
"Hautbois", "Luth", "Lutrin", "Lyre", "Maître de danse", "Mélodie", "Mi",
"Musette", "Musica", "Musique", "Note de musique", "Notes de musique", "Opéra",
"Orchestre", "Orgue", "Orgues de Barbarie", "Peau du tambour", "Psaltérion", "Sept
notes de la gamme", "Si et les autres notes de musique", "Symphonie", "Trompette",
"Tympanon", "Ut, ré, mi, fa, sol, la, si, ut", "Viole", "Violon");

    $domain_array_natural_history = array("Abeille", "Abeilles", "Abricot",
"Aigle", "Ail", "Aile", "Aileron", "Air", "Alpes", "Amarante", "Âne", "Anis",
"Ânon", "Anthropophage", "Araignée", "Arbre", "Arbres", "Arc-en-ciel", "Arête",
"Arête de poisson", "Argent", "Arsenic", "Artichaut", "Aspic", "Atome",
"Aubépine", "Aurore", "Averse", "Avoine", "Baleine", "Barbeau", "Baromètre",
"Basilic", "Becfigue", "Betterave", "Bise", "Blé", "Bocage", "Boeuf", "Bois",
"Bois de la vigne", "Bois de Sainte-Lucie", "Bois flotté", "Bombyx", "Bonde des
étangs", "Bosquet", "Bouis", "Bourgeon", "Boussole", "Brochet", "Bronze",
"Buisson", "Cabri", "Cacaoyer", "Café", "Calice de la rose", "Caméléon", "Canard",
"Cane", "Carotte et crotte", "Carpe", "Castor", "Cerf", "Cerise", "Cerneau",
"Cerneaux", "Cerveau", "Cervelle", "Chair", "Chameau", "Champignon",
"Chanterelle", "Chanvre", "Chapon", "Charbon", "Chardon", "Chardonneret",
"Charmille", "Chasselas", "Chat", "Châtaigne", "Chats", "Chatte", "Chauve-souris",
"Cheval", "Chevelure", "Cheveu", "Cheveux", "Chèvrefeuille", "Chien", "Chien
couchant", "Chiendent", "Chienne couchante", "Chou", "Chou-fleur", "Choux",
"Ciel", "Citron", "Citrouille", "Cloporte", "Clou de girofle", "Cochon", "Coco",
"Coeur", "Colimaçon", "Colombe", "Comète", "Coq", "Coq d'Inde", "Coquillage",
"Coquille", "Corail", "Corbeau", "Corme", "Corneille", "Cornes", "Cornichon",
"Coucou", "Coude", "Couperose", "Courlis", "Craie", "Crâne", "Cuivre", "Dauphin",
"Démon", "Dent", "Dents", "Diamant brut", "Dogue", "Eau", "Eau de la rivière",
"Écaille", "Échalote", "Écho", "Éclair", "Écrevisse", "Éléments", "Embryon",
"Épervier", "Épi", "Épi de blé", "Épi de froment", "Épine", "Éponge", "Érable",
"Escargot", "Espadon", "Estomac", "Étalon", "Feu ou la chaleur", "Feuillage",
"Feuille d'arbre", "Feuille du calice de la rose", "Feuilles des arbres", "Fève",
"Fleur", "Fleuve", "Flumen", "Flux et reflux", "Forêt", "Fougère", "Fourmi",
```

```
"Framboise", "Froment", "Génisse", "Gland", "Goret", "Grain", "Grain de blé",
"Grain de blé ou d'orge", "Gratte-cul", "Grêle", "Grenouille", "Grès", "Grès ou
rocher", "Gui", "Guimauve", "Hanneton", "Héliotrope", "Hellébore", "Hérisson",
"Hermaphrodite", "Hermine", "Héron", "Hirondelle", "Huître", "Huître à l'écaille",
"If", "Iris ou l'arc-en-ciel", "Ivoire", "Jasmin", "Jonquille", "Lait", "Laitue",
"Lamie", "Lamproie", "Lapin", "Lapis", "Lièvre", "Limaçon", "Limon", "Lin",
"Linotte", "Lion", "Lis", "Louveteau", "Lumière", "Lune", "Magicien", "Magie",
"Magnétisme", "Mappemonde", "Marbre", "Marguerite", "Marjolaine", "Marron",
"Marronier", "Marronnier", "Marrons", "Melon", "Ménagerie", "Mer", "Méridien",
"Merlan", "Miel", "Mineur et mine", "Moineau", "Monstre", "Montagne", "Moucheron",
"Moucherons", "Mouvement", "Mulet", "Mûre", "Mûrier", "Muscade", "Nacre",
"Nature", "Navet", "Nèfle", "Nègre", "Neige", "Neiges", "Nid d'oiseau", "Nielle",
"Noisette", "Noix", "Nostradamus", "Noyer", "Nuage", "Nuée", "Oie", "Oignon",
"Oiseau", "Oiseau-mouche", "Olivier", "Or", "Orage", "Orange douce", "Orme",
"Ormeau", "Orpin", "Ortie", "Ouragan", "Panthère", "Papillon", "Pêche", "Pelote de
neige", "Pépin", "Pépin de pomme", "Perdreaux", "Perle", "Perroquet", "Pierre
philosophale", "Pinson", "Pistache", "Pluie", "Poire", "Poisson", "Poivre",
"Pomme", "Pommier", "Porc", "Porcus", "Pou", "Poulain", "Poule", "Poulin",
"Prairie", "Prune", "Puce", "Quatre éléments", "Raisin", "Rat", "Rave", "Reine des
abeilles", "Reine-claude", "Reinette", "Rhinocéros", "Rivière", "Rocher", "Ronce",
"Rosa", "Rose", "Rose et le lis", "Roseau", "Rosée", "Rosier", "Rossignol",
"Ruisseau", "Sable", "Saison", "Saisons", "Sanglier", "Sapin", "Sarrasin",
"Saule", "Seigle", "Serin", "Serpent", "Sirène", "Soleil", "Source", "Spectre",
"Tabac", "Tarentule", "Taupe", "Taureau", "Tempestas", "Tempête", "Terre",
"Thermomètre", "Tigre", "Toile d'araignée", "Tonnerre", "Tourterelle", "Trèfle",
"Truffe", "Truite", "Univers", "Vanneau", "Veau", "Vent", "Vent du nord", "Ver",
"Ver à soie", "Ver de bois", "Ver luisant", "Verrat", "Verseau", "Vide", "Vif-
argent", "Vigne", "Vigne vierge", "Violette", "Vipère", "Volcan", "Volière",
"Vulpes");

    $domain_array_philology = array("Accent aigu", "Accents", "Accents aigu, grave
et circonflexe", "Alphabet", "Apostrophe", "Cinq voyelles", "Dictionnaire",
"Écriture", "Écriture bâtarde", "Langue", "Lettre A", "Lettre B", "Lettre C",
"Lettre D", "Lettre E", "Lettre F", "Lettre G", "Lettre H", "Lettre I", "Lettre
J", "Lettre L", "Lettre M", "Lettre N", "Lettre O", "Lettre P", "Lettre R",
"Lettre S", "Lettre T", "Lettre U", "Lettre V", "Lettre X", "Lettre Y", "Lettre
Z", "Lettres C, E, N, O", "Lettres de l'alphabet", "Lettres I et V",
"Monosyllabe", "Mot", "Mots", "Orthographe", "Parenthèse", "Parole", "Pléonasme",
"Point sur l'i", "Son de la voix", "Substantif", "Vingt-quatre lettres de
l'alphabet", "Virgule", "Voyelle", "Voyelle E", "Voyelle U", "Voyelles", "Voyelles
E, I, U");

    $domain_array_philosophy = array("Amour propre", "Conscience", "Corps et âme",
"Démocrite", "Descartes", "Épicure", "Être", "Modération", "Optimisme",
"Philosophie", "Platon", "Préjugé", "Pyrrhonisme", "Réflexion", "Sage", "Sagesse",
"Socrate");

    $domain_array_political_science = array("Aristocrate", "Aristocratie",
"Bourgeois", "Cocarde", "Commerce", "Confédération", "Conseiller", "Corsaire",
"Courtisane", "Décret", "Despote", "Despotisme", "Duc de Bretagne", "Économie",
"Émir", "Exil", "Fleur de lis", "Frontière", "Gênes soumise", "Liberté", "Loterie
royale", "Luxe", "Maire", "Misère", "Monarchie", "Nation", "Noblesse", "Paix",
"Politique", "Population", "Reine", "République", "Révolte", "Révolution", "Roi",
"Royaume", "Sa Majesté danoise", "Sceptre", "Souffrance", "Tiers état", "Tontine",
"Trône", "Vice-roi", "Vive le roi");

    $domain_array_religion = array("Abel", "Absolution", "Adam", "Adam et Ève",
"Alléluia", "Âme", "Ange", "Angélus", "Apothéose", "Aumusse", "Avoye", "Babel",
"Bénédicité", "Bigote", "Bon larron", "Bourdaloue", "Bréviaire", "Camail",
```

```
"Camerlingue", "Capucin", "Carême", "Carme", "Catéchisme", "Catéchumène", "Chaire
à prêcher", "Chanoine", "Chanoinesses", "Chantre", "Chapelet", "Charité",
"Cimetière", "Confessionnal", "Cordelier", "Couvent", "Croix", "Curé", "Devin",
"Diable", "Dieu", "Eau bénite", "Église", "Église romaine", "Eucharistie",
"Fanatisme", "Fève du gâteau des rois", "Fruit de vie", "Goupillon", "Idolâtrie",
"Idole", "Incarnation", "Jeûne", "Jourdain", "Jubilé", "Latro poenitens",
"Lazare", "Lévite", "Luthéranisme", "Magloire", "Marguillier", "Marie", "Martre",
"Martyrologe", "Melchisédech", "Miracle", "Miserere", "Mitre", "Moine",
"Monastère", "Nonce", "Nonnain", "Orémus", "Pain bénit de paroisse", "Parabole",
"Passage de la mer Rouge", "Pèlerin", "Pèlerinage", "Pentecôte", "Prédicateur",
"Prière", "Providence", "Psaume Miserere", "Psautier", "Religion", "Relique",
"Sacerdoce", "Sarah et Rachel", "Séminaire", "Sept psaumes pénitentiels",
"Synagogue", "Tonsure", "Tonsure de clerc", "Tour des couvents");

    $domain_array_sciences_of_war = array("Affût de canon", "Arme", "Armée",
"Armée des Confédérés", "Arsenal", "Artillerie", "Artilleur", "Assaut",
"Baïonnette", "Bannière", "Bataille", "Bataillon", "Bâton de maréchal de France",
"Belle Poule", "Bombe", "Bouclier", "Boulet", "Canon", "Carabine", "Carabinier",
"Castramétation", "Castrum", "Cavalerie", "Cavalier", "Ceinturon", "Chevaux-
léger", "Cimeterre", "Combat", "Escarmouche", "Cuirasse", "D'Estaing", "Front de
fortification", "Giberne", "Guerre", "Hallebarde", "Hausse-col", "Havresac",
"Infanterie", "Milice", "Mousquetaire", "Oriflamme", "Poudre à canon", "Régiment",
"Soldat", "Tactique", "Tambour et amour", "Tente d'armée", "Trêve", "Vergennes");

    $tex_domain = '';

    if(in_array($reponse_normalisee, $domain_array_law)){
        $tex_domain = "186"; //Value matching "Jurisprudence, droit" in the
"Domaine" taxonomy of the Drupal database.
    }
    if(in_array($reponse_normalisee, $domain_array_arts)){
        $tex_domain = "196"; //Value matching "Beaux-arts" in the "Domaine"
taxonomy of the Drupal database.
    }
    if(in_array($reponse_normalisee, $domain_array_belles_lettres)){
        $tex_domain = "197"; //Value matching "Belles-lettres, théâtre" in the
"Domaine" taxonomy of the Drupal database.
    }
    if(in_array($reponse_normalisee, $domain_array_human_knowledge)){
        $tex_domain = "183"; //Value matching "Système et histoire des
connaissances humaines" in the "Domaine" taxonomy of the Drupal database.
    }
    if(in_array($reponse_normalisee, $domain_array_history)){
        $tex_domain = "195"; //Value matching "Histoire" in the "Domaine" taxonomy
of the Drupal database.
    }
    if(in_array($reponse_normalisee, $domain_array_medicine)){
        $tex_domain = "187"; //Value matching "Médecine, chirurgie, pharmacie" in
the "Domaine" taxonomy of the Drupal database.
    }
    if(in_array($reponse_normalisee, $domain_array_music)){
        $tex_domain = "198"; //Value matching "Musique, danse" in the "Domaine"
taxonomy of the Drupal database.
    }
    if(in_array($reponse_normalisee, $domain_array_religion)){
        $tex_domain = "185"; //Value matching "Théologie, religion" in the
"Domaine" taxonomy of the Drupal database.
    }
    if(in_array($reponse_normalisee, $domain_array_philology)){
```

```
        $tex_domain = "184"; //Value matching "Philologie, connaissance des
langues" in the "Domaine" taxonomy of the Drupal database.
    }
    if(in_array($reponse_normalisee, $domain_array_philosophy)){
        $tex_domain = "188"; //Value matching "Philosophie" in the "Domaine"
taxonomy of the Drupal database.
    }
    if(in_array($reponse_normalisee, $domain_array_education)){
        $tex_domain = "189"; //Value matching "Pédagogie" in the "Domaine"
taxonomy of the Drupal database.
    }
    if(in_array($reponse_normalisee, $domain_array_mathematics)){
        $tex_domain = "194"; //Value matching "Mathématiques" in the "Domaine"
taxonomy of the Drupal database.
    }
    if(in_array($reponse_normalisee, $domain_array_mechanical_arts)){
        $tex_domain = "193"; //Value matching "Métiers, arts mécaniques" in the
"Domaine" taxonomy of the Drupal database.
    }
    if(in_array($reponse_normalisee, $domain_array_natural_history)){
        $tex_domain = "192"; //Value matching "Histoire naturelle" in the
"Domaine" taxonomy of the Drupal database.
    }
    if(in_array($reponse_normalisee, $domain_array_political_science)){
        $tex_domain = "190"; //Value matching "Sciences politiques" in the
"Domaine" taxonomy of the Drupal database.
    }
    if(in_array($reponse_normalisee, $domain_array_sciences_of_war)){
        $tex_domain = "191"; //Value matching "Sciences de la guerre" in the
"Domaine" taxonomy of the Drupal database.
    }

    //Exceptions.

    if($reponse_normalisee == 'Quinquina'){
        $tex_domain = '"187,192"';
    }
    if($reponse_normalisee == 'Capucine'){
        $tex_domain = '"185,192"';
    }
    if($reponse_normalisee == 'Cartouche'){
        $tex_domain = '"191,186,197"';
    }
    if($reponse_normalisee == 'Critique' && ($id == 1087 || $id == 5744 || $id ==
5866 || $id == 6291)){
        $tex_domain = '197';
    }
    if($reponse_normalisee == 'Émile'){
        $tex_domain = '"189,191"';
    }
    if($reponse_normalisee == 'Éternité' && $id == 1887){
        $tex_domain = '188';
    }
    if($reponse_normalisee == 'Harmonie' && $id == 1732){
        $tex_domain = '198';
    }
    if($reponse_normalisee == 'Hécatombe'){
        $tex_domain = '"195,185"';
    }
```

```
    if($reponse_normalisee == 'Malade imaginaire'){
        $tex_domain = '"197,187"';
    }
    if($reponse_normalisee == 'Martial'){
        $tex_domain = '"197,191"';
    }
    if($reponse_normalisee == 'Médaille'){
        $tex_domain = '"195,196"';
    }
    if($reponse_normalisee == 'Mercure' && ($id == 591 || $id == 788 || $id ==
4930 || $id == 4865)){
        $tex_domain = '';
    }
    if($reponse_normalisee == 'Mercure' && ($id == 852 || $id == 1911 || $id ==
4414 || $id == 4515 || $id == 5162)){
        $tex_domain = '"197,192"';
    }
    if($reponse_normalisee == 'Mercure' && $id == 6105){
        $tex_domain = '"197,192,187"';
    }
    if($reponse_normalisee == 'Mercure' && ($id == 3455 || $id == 6814)){
        $tex_domain = '192';
    }
    if($reponse_normalisee == 'Mercure' && $id == 6624){
        $tex_domain = '187';
    }
    if($reponse_normalisee == 'Mercure' && $id == 6702){
        $tex_domain = '"197,187"';
    }
    if($reponse_normalisee == 'Mercure' && $id == 6834){
        $tex_domain = '"192,187"';
    }
    if($reponse_normalisee == 'Mesmer' && $id == 5043){
        $tex_domain = '187';
    }
    if($reponse_normalisee == 'Ordonnance'){
        $tex_domain = '"191,186"';
    }
    if($reponse_normalisee == 'Pythagore'){
        $tex_domain = '"194,188"';
    }
    if($reponse_normalisee == 'Romance'){
        $tex_domain = '"198,197"';
    }
    if($reponse_normalisee == 'Saisons' && $id == 3326){
        $tex_domain = '"197,192"';
    }
    if($reponse_normalisee == 'Sifflet' && $id == 1767){
        $tex_domain = '"197,189"';
    }
    if($reponse_normalisee == 'Superstition'){
        $tex_domain = '"185,188"';
    }
    if($reponse_normalisee == 'Tabouret et sellette'){
        $tex_domain = '"186,190"';
    }
    if($reponse_normalisee == 'Tambour'){
        $tex_domain = '"198,191"';
    }
```

```
    if($reponse_normalisee == 'Tolérance'){
        $tex_domain = '"188,185"';
    }
    if($reponse_normalisee == 'Tour' && $id == 1150){
        $tex_domain = '193';
    }
    if($reponse_normalisee == 'Tour' && $id == 3382){
        $tex_domain = '191';
    }
    if($reponse_normalisee == 'Triangle' && ($id == 6501 || $id == 5381 || $id ==
1788 || $id == 1137)){
        $tex_domain = '"194,185"';
    }
    if($reponse_normalisee == 'Trompette' && ($id == 5082 || $id == 2939 || $id ==
2704)){
        $tex_domain = '"198,191"';
    }
    if($reponse_normalisee == 'Vaudeville'){
        $tex_domain = '"197,198"';
    }
    if($reponse_normalisee == 'Vierge'){
        $tex_domain = '"185,192"';
    }

    //Create field tex_signature. (Remains empty.)

    $tex_signature = '';

    //Create field tex_author_name from auteur.

    $tex_author_name = '';

    foreach($taxonomy_term_array as $taxonomy_term_value_2){

        if($taxonomy_term_value_2['vid'] === 'name'){

            if($auteur == $taxonomy_term_value_2['name']){
                $tex_author_name = $taxonomy_term_value_2['tid'];
            }

        }

    }

    //Exceptions.

    if($auteur == "E. M. J. D. L, Solitaire des bords de la Marne"){
        $tex_author_name = 9345;
    }
    if($auteur == "A. Barbery, le jeune."){
        $tex_author_name = 18331;
    }
    if($auteur == "M. de Broglio, de Martigues."){
        $tex_author_name = 9924;
    }
    if($auteur == "M. de Longpré, chés M. Boivin"){
        $tex_author_name = 10001;
    }
    if($auteur == "M. Barberi, le jeune."){
```

```
        $tex_author_name = 18330;
    }
    if($auteur == "M. Barbery, le jeune."){
        $tex_author_name = 18329;
    }
    if($auteur == "Gueroult"){
        $tex_author_name = 9404;
    }
    if($auteur == "M. Gueroult de Fécamp."){
        $tex_author_name = 10257;
    }
    if($auteur == "M. de la Grée, le fils."){
        $tex_author_name = 9969;
    }
    if($auteur == "Louis Coulard, l'Assesseur."){
        $tex_author_name = 9608;
    }
    if($auteur == "Roy, l'aîné."){
        $tex_author_name = 18327;
    }
    if($auteur == "..... de Falaise."){
        $tex_author_name = 18328;
    }
    if($auteur == "M. de la Roche, le jeune."){
        $tex_author_name = 9977;
    }
    if($auteur == "M. F***, fils"){
        $tex_author_name = 18332;
    }
    if($auteur == "M. de Lozières, fils"){
        $tex_author_name = 10004;
    }
    if($auteur == "M. de MONTPELLIER."){
        $tex_author_name = 10019;
    }
    if($auteur == "M. Castaing, fils"){
        $tex_author_name = 9776;
    }
    if($auteur == "M. Castaing fils"){
        $tex_author_name = 9776;
    }
    if($auteur == "L'Abbé de VINCHEGUERRE."){
        $tex_author_name = 9510;
    }
    if($auteur == "Madame Ourseau, chez Madame la Duchesse d'Ollone"){
        $tex_author_name = 10953;
    }
    if($auteur == "Mlle de Sauret, cadette"){
        $tex_author_name = 18333;
    }
    if($auteur == "Mademoiselle de Sauret, l'aînée"){
        $tex_author_name = 18334;
    }
    if($auteur == "L'Anonyme de Chartrait, par Melun."){
        $tex_author_name = 9535;
    }
    if($auteur == "Cardonne fils, C. A. C. G. D. M. L. D."){
        $tex_author_name = 18335;
    }
```

```
    if($auteur == "M. de Saint-Martin, Vicomte de Briouze"){
        $tex_author_name = 10045;
    }
    if($auteur == "M. de Groubental, fils."){
        $tex_author_name = 18336;
    }
    if($auteur == "De Groubentall, fils"){
        $tex_author_name = 18337;
    }
    if($auteur == "C. D. V., C. D. N."){
        $tex_author_name = 9231;
    }
    if($auteur == "M. Lagache, fils"){
        $tex_author_name = 10450;
    }
    if($auteur == "M. Levrier, fils."){
        $tex_author_name = 18338;
    }
    if($auteur == "Des Marais, du Chambon"){
        $tex_author_name = 18339;
    }
    if($auteur == "Mde Oublé, Li. R. S. J."){
        $tex_author_name = 18340;
    }
    if($auteur == "M. Lagache, fils."){
        $tex_author_name = 10450;
    }
    if($auteur == "M. l'Abbé de L. P***, auteur de Colas."){
        $tex_author_name = 10379;
    }
    if($auteur == "Papelart, A. A. M."){
        $tex_author_name = 18341;
    }
    if($auteur == "Par une Société de Gens de Lettres."){
        $tex_author_name = 11321;
    }
    if($auteur == "Par une société de gens de lettres."){
        $tex_author_name = 11321;
    }
    if($auteur == "M. Simon, fils"){
        $tex_author_name = 18342;
    }
    if($auteur == "M. le Baron des Hermeaux de la Valette, P. A. C R. D. N."){
        $tex_author_name = 18343;
    }
    if($auteur == "M. Poulhariez, écuyer."){
        $tex_author_name = 10758;
    }
    if($tex_author_name == 10759 || $tex_author_name == '10759'){
        $tex_author_name = 10758;
    }
    if($auteur == "M. R**, de l'Aquitaine"){
        $tex_author_name = 10786;
    }
    if($auteur == "M. Hubert, P. D. M. L. V. D. S. M."){
        $tex_author_name = 18344;
    }
    if($auteur == "Mde Seriés"){
        $tex_author_name = 11001;
```

```
    }
    if($auteur == "M. V. de P., fils."){
        $tex_author_name = 18345;
    }
    if($auteur == "M. Papelart, D."){
        $tex_author_name = 18346;
    }
    if($auteur == "M. Cott, P. de B."){
        $tex_author_name = 18347;
    }
    if($auteur == "M. l'Abbé Lau***, P. de C**."){
        $tex_author_name = 18348;
    }
    if($auteur == "M. de Lanevère"){
        $tex_author_name = 9990;
    }
    if($auteur == "M. Verkaven, fils."){
        $tex_author_name = 18349;
    }
    if($auteur == "M. Gazil, fils."){
        $tex_author_name = 10221;
    }
    if($auteur == "M. le Méteyer."){
        $tex_author_name = 10571;
    }
    if($auteur == "M. Pot, C. d. N."){
        $tex_author_name = 18350;
    }
    if($auteur == "M. Donnavi, Ga...de..."){
        $tex_author_name = 18363;
    }
    if($auteur == "M. Labrouche, fils"){
        $tex_author_name = 10444;
    }
    if($auteur == "Mantois Galant cher prisé, dans l'enceinte du Palais"){
        $tex_author_name = 10983;
    }
    if($auteur == "Mirtil, le Berger Fidelle d'Angouléme"){
        $tex_author_name = 18364;
    }
    if($auteur == "Madame de Saliez, Viguiere d'Alby"){
        $tex_author_name = 18365;
    }
    if($auteur == "Nymphes enjoüées, Clione, & Rozelinde"){
        $tex_author_name = 11288;
    }
    if($auteur == "Nympes enjoüées, Clione, & Roselinde"){
        $tex_author_name = 11288;
    }
    if($auteur == "Nimpes enjoüées, Clione & Rozelinde"){
        $tex_author_name = 11288;
    }
    if($auteur == "Mr Diéreville"){
        $tex_author_name = 11202;
    }
    if($auteur == "Mr Diereville"){
        $tex_author_name = 11202;
    }
    if($auteur == "Mr de Beaujour, Sr de Guilleranche de Mascon"){
```

```
        $tex_author_name = 18366;
    }
    if($auteur == "Madame d'Athie, connuë sous le nom de la grande Doris"){
        $tex_author_name = 18367;
    }
    if($auteur == "Mr de Longueüil, de l'Academie Royale d'Angers"){
        $tex_author_name = 11191;
    }
    if($auteur == "L'Abbé Insulaire à l'Anagramme, Rubé Caffé, & le jeune Logicien
de la ruë S. Loüis"){
        $tex_author_name = '"9514,18369,9474"';
    }
    if($auteur == "Du bel emplastre, qui fait plus de vers qu'un autre n'en peut
lire."){
        $tex_author_name = 9330;
    }
    if($auteur == "L'aimable Guibour, de la ruë S. Honoré"){
        $tex_author_name = 9527;
    }
    if($auteur == "Fidele Berger de la Nymphe insulaire"){
        $tex_author_name = 9367;
    }
    if($auteur == "Solitaire, malgré lui"){
        $tex_author_name = 11385;
    }
    if($auteur == "M. Bodin, C. D. F. D. R."){
        $tex_author_name = 18374;
    }
    if($auteur == "Quelqu'un qui auroit mieux aimé dormir, & qui eût peut-être
mieux fait"){
        $tex_author_name = 11341;
    }
    if($auteur == "M. Prévot de Moka"){
        $tex_author_name = 10769;
    }
    if($auteur == "M. L...y, R. D. G."){
        $tex_author_name = 18376;
    }
    if($auteur == "M. R....., fils aîné"){
        $tex_author_name = 18377;
    }
    if($auteur == "M. le Chevalier Brunet de la Martinière, fils aîné"){
        $tex_author_name = 18378;
    }
    if($auteur == "M. de la Sablonnière, C. R."){
        $tex_author_name = 9982;
    }
    if($auteur == "Une Société de Saint-Rambert, en Bugey"){
        $tex_author_name = 18379;
    }
    if($auteur == "M. Régnault"){
        $tex_author_name = 10798;
    }
    if($auteur == "MM. de Costebelle, Bruyer"){
        $tex_author_name = 18380;
    }
    if($auteur == "M. Sebire de Beauchesne"){
        $tex_author_name = 10844;
    }
```

```
    if($auteur == "M. Prevost"){
        $tex_author_name = 10765;
    }
    if($auteur == "M. Esmenard, fils aîné"){
        $tex_author_name = 18381;
    }
    if($auteur == "M. Emenard, fils aîné"){
        $tex_author_name = 18381;
    }
    if($auteur == "M. le Meteyer"){
        $tex_author_name = 10571;
    }
    if($auteur == "M. Prévost de Montigny"){
        $tex_author_name = 10767;
    }
    if($auteur == "M. d. P...., o... a. R... C... g... i..."){
        $tex_author_name = 18382;
    }
    if($auteur == "M. Prévost de Moka"){
        $tex_author_name = 10766;
    }
    if($auteur == "M. Prevôt de Moka"){
        $tex_author_name = 10769;
    }
    if($auteur == "M. Ch. M. D. V."){
        $tex_author_name = 18383;
    }
    if($auteur == "J. M. Baudin aîné"){
        $tex_author_name = 18384;
    }
    if($auteur == "Antonin Dauberjon"){
        $tex_author_name = 18385;
    }
    if($auteur == "L. D. B."){
        $tex_author_name = 18386;
    }
    if($auteur == "Germain Dauberjon"){
        $tex_author_name = 18387;
    }
    if($auteur == "Auguste M. de C......"){
        $tex_author_name = 18388;
    }
    if($auteur == "J. M. Baudin, aîné"){
        $tex_author_name = 18384;
    }
    if($auteur == "Houssaye, père"){
        $tex_author_name = 18389;
    }
    if($auteur == "Citoyen P......"){
        $tex_author_name = 18390;
    }
    if($auteur == "Mely Janin"){
        $tex_author_name = 18391;
    }
    if($auteur == "G. M."){
        $tex_author_name = 18392;
    }
    if($auteur == "B. J. V. L. R......"){
        $tex_author_name = 18393;
```

```
    }
    if($auteur == "L. H. D. G."){
        $tex_author_name = 9496;
    }
    if($auteur == "L. H. D. G. "){
        $tex_author_name = 9496;
    }
    if($auteur == "M. Emery, Sieur de l'Isle C. D. N."){
        $tex_author_name = 10139;
    }
    if($id == 3843 || $id == 3848){
        $tex_author_name = 11174;
    }
    if($auteur == "M. Lagache fils"){
        $tex_author_name = 10450;
    }

    //Other exception (mistake to be corrected in the "Les énigmes de Mercure"
database).

    if($id == 1827){
        $tex_author_name = 12929; //Value matching "B. D. B." in the "Name"
taxonomy.
    }

    //Create field tex_author_activity from activite_normalisee.

    $tex_author_activity = '';

    foreach($taxonomy_term_array as $taxonomy_term_value_3){

        if($taxonomy_term_value_3['vid'] === 'activity'){

            if($activite_normalisee == $taxonomy_term_value_3['name']){
                $tex_author_activity = $taxonomy_term_value_3['tid'];
            }

        }

    }

    //Exceptions.

    if($id == 2400){
        $tex_author_activity = '';
    }
    if($id == 6086){
        $tex_author_activity = '';
    }
    if($id == 6074){
        $tex_author_activity = '';
    }
    if($id == 6072){
        $tex_author_activity = '';
    }
    if($id == 5795){
        $tex_author_activity = '';
    }
    if($id == 5794){
```

```
        $tex_author_activity = '';
    }
    if($id == 5743){
        $tex_author_activity = '';
    }
    if($id == 3471){
        $tex_author_activity = '';
    }
    if($id == 2607 || $id == 2658 || $id == 2686){
        $tex_author_activity = 20150;
    }
    if($id == 209){
        $tex_author_activity = '';
    }
    if($id == 195){
        $tex_author_activity = '';
    }
    if($id == 194){
        $tex_author_activity = '';
    }
    if($id == 1990){
        $tex_author_activity = '';
    }
    if($id == 1755){
        $tex_author_activity = '';
    }
    if($activite_normalisee == 'Chantre, chanoine'){
        $tex_author_activity = '"18670,11503"';
        $author_male = true;
    }
    if($activite_normalisee == 'Notaire, directeur du bureau des postes aux
lettres'){
        $tex_author_activity = '"11503,18671"';
        $author_male = true;
    }
    if($activite_normalisee == 'Avocat au parlement, conseiller'){
        $tex_author_activity = '"18313,18672"';
        $author_male = true;
    }
    if($activite_normalisee == 'Libraire, marchand d\'estampes'){
        $tex_author_activity = '"18673,18674"';
        $author_male = true;
    }
    if($activite_normalisee == 'Maître de musique, organiste'){
        $tex_author_activity = '"18675,18441"';
        $author_male = true;
    }
    if($activite_normalisee == 'Ingénieur pour le roi, lieutenant'){
        $tex_author_activity = '"18676,12938"';
        $author_male = true;
    }
    if($activite_normalisee == 'Négociant, consul'){
        $tex_author_activity = '"18472,18677"';
        $author_male = true;
    }
    if($activite_normalisee == 'Subdélégué, juge'){
        $tex_author_activity = '"18678,18435"';
        $author_male = true;
    }
```

```
    if($activite_normalisee == 'Mestre de camp, lieutenant des gardes'){
        $tex_author_activity = '"18679,18680"';
        $author_male = true;
    }
    if($activite_normalisee == 'Professeur de géographie, maître d\'armes'){
        $tex_author_activity = '"18681,18682"';
        $author_male = true;
    }
    if($activite_normalisee == 'Peintre, professeur de dessin et d\'écriture'){
        $tex_author_activity = '"18466,18683"';
        $author_male = true;
    }
    if($activite_normalisee == 'Religieux, étudiant en théologie'){
        $tex_author_activity = '"18461,18684"';
        $author_male = true;
    }
    if($activite_normalisee == 'Maître d\'écriture, professeur d\'arithmétique'){
        $tex_author_activity = '"18685,18686"';
        $author_male = true;
    }
    if($activite_normalisee == 'Docteur en médecine, chirurgien-major'){
        $tex_author_activity = '"15243,18623"';
        $author_male = true;
    }
    if($activite_normalisee == 'Maître ès arts, maître de pension'){
        $tex_author_activity = '"18529,18453"';
        $author_male = true;
    }

    //Create field tex_author_title_of_nobility from auteur.

    $author_marquess_array = array("M. le marquis de ***", "M. le marquis de
C......", "M. le marquis de Cogners.", "M. le marquis de Fulvy", "M. le marquis de
la S**.", "M. le marquis de Neuvic", "M. le marquis des Six-Tours", "M. le mqs. de
la Parterie", "marquis de la Guesnerie d'Anjou", "Mr le marquis de S. Priest", "Mr
le marquis de Turlan");

    $author_marquess_array = array_map('strtolower', $author_marquess_array);

    $author_marchioness_array = array("Madame la marquise de Bert***", "Madame la
marquise de Montbazin.", "Mde la marquise de Noailles", "Mme la marquise de ****",
"Mme la marquise de B...");

    $author_marchioness_array = array_map('strtolower',
$author_marchioness_array);

    $author_count_array = array("M. le comte de Chemindieu", "M. le comte de
F***", "M. le comte de Gilin", "M. le comte de J.", "M. le comte de L. T.....",
"M. le comte de la Guerrande", "M. le comte de M.", "M. le comte de Pontevès", "M.
le comte de R. A. F.", "M. le comte de Rogles", "M. le comte de S. Gilin", "M. le
comte de V.......", "M. le comte de Vevure", "Mr le comte de Turlan");

    $author_count_array = array_map('strtolower', $author_count_array);

    $author_countess_array = array("La comtesse de ***", "Madame la comtesse
d'Apr.", "Madame la comtesse de B...", "Mde la comtesse de Chauvigny de Blot",
"Mme la comtesse de L**", "Mme la comtesse de St-Maximen Montclair", "Mme la
comtesse de Tréborras", "Mme. la comtesse de B***", "Mme. la comtesse de B...",
"Mme. la comtesse de M...");
```

```
    $author_countess_array = array_map('strtolower', $author_countess_array);

    $author_viscount_array = array("M. de Saint-Martin, Vicomte de Briouze", "M.
le Vicomte de B.", "M. le Vicomte de Ga....", "M. le Vicomte de Gal.", "M. le
Vicomte de Gal...", "M. le Vicomte de la Cressonniere", "M. le Vicomte de
Mélignan", "M. le Vicomte de Tilly", "M. le Vicomte Desfossés", "M. le Viconte de
Gal...");

    $author_viscount_array = array_map('strtolower', $author_viscount_array);

    $author_baron_array = array("M. le baron de B***", "M. le baron de P***", "M.
le baron de P.", "M. le baron de Thomassin de Juilly", "M. le baron de Walbert",
"M. le baron des Hermeaux de la Valette, P. A. C R. D. N.");

    $author_baron_array = array_map('strtolower', $author_baron_array);

    $author_baroness_array = array("Madame la baronne C....", "Madame la baronne
de Potelle");

    $author_baroness_array = array_map('strtolower', $author_baroness_array);

    $author_knight_array = array("Le chevalier de C****.", "Le chevalier de
l'Albarede.", "Le chevalier de Montador.","Le chevalier de T***.", "M. l[e]
chevalier de Meude Monpas", "M. le ch. de Chambon", "M. le ch. de L... St Just",
"M. le ch. de l'Huillier", "M. le ch. de Meude-Monpas", "M. le ch. de P***", "M.
le ch. de Saint-Gervais", "M. le chev. de Boisfontaine.", "M. le chev. de
Desezgaulx", "M. le chev. de Nerciat", "M. le chevalier Amyot", "M. le chevalier
Brunet de la martinière, fils aîné", "M. le chevalier D. B.", "M. le chevalier de
Bel-Orme", "M. le chevalier de Bergerac", "M. le chevalier de Blair", "M. le
chevalier de Breuvery", "M. le chevalier de Briois", "M. le chevalier de C. X.",
"M. le chevalier de C....", "M. le chevalier de Ch...", "M. le chevalier de
Chalignac.", "M. le chevalier de D**.", "M. le chevalier de L.... M.....", "M. le
chevalier de la doué", "M. le chevalier de la Grange", "M. le chevalier de la
mothe", "M. le chevalier de la Vente", "M. le chevalier de M. E. de Mad. D. F.",
"M. le chevalier de M....", "M. le chevalier de Meude-Monpas", "M. le chevalier de
P...", "M. le chevalier de S.... V...", "M. le chevalier de S.... V....", "M. le
chevalier de Susars", "M. le chevalier de T**.", "M. le chevalier de T....", "M.
le chevalier Desmarais du Chambon.", "M. le chevalier Hugot.", "M. le chevalier Le
Prévost", "M. le chevalier le Prevost.", "M. le chevalier Mesnard du Montelet");

    $author_knight_array = array_map('strtolower', $author_knight_array);

    $author_squire_array = array("M. Poulhariez, écuyer.");

    $author_squire_array = array_map('strtolower', $author_squire_array);

    $author_male_president_array = array("Mr le Président de la Tournelle", "Mr le
P. la Tournelle");

    $author_male_president_array = array_map('strtolower',
$author_male_president_array);

    $author_female_president_array = array("Madame la Présidente du T.....", "Me
la Presidente Baucheron");

    $author_female_president_array = array_map('strtolower',
$author_female_president_array);
```

```
    $auteur_strtolower = strtolower($auteur);

    if(in_array($auteur_strtolower, $author_marquess_array)){
        $tex_author_title_of_nobility = 230; //Value matching "Marquis" in the
"Tire de noblesse" taxonomy of the Drupal database.
        $author_male = true;
    }
    if(in_array($auteur_strtolower, $author_marchioness_array)){
        $tex_author_title_of_nobility = 13393; //Value matching "Marquise" in the
"Tire de noblesse" taxonomy of the Drupal database.
        $author_female = true;
    }
    if(in_array($auteur_strtolower, $author_count_array)){
        $tex_author_title_of_nobility = 12839; //Value matching "Comte" in the
"Tire de noblesse" taxonomy of the Drupal database.
        $author_male = true;
    }
    if(in_array($auteur_strtolower, $author_countess_array)){
        $tex_author_title_of_nobility = 12119; //Value matching "Comtesse" in the
"Tire de noblesse" taxonomy of the Drupal database.
        $author_female = true;
    }
    if(in_array($auteur_strtolower, $author_viscount_array)){
        $tex_author_title_of_nobility = 19587; //Value matching "Vicomte" in the
"Tire de noblesse" taxonomy of the Drupal database.
        $author_male = true;
    }
    if(in_array($auteur_strtolower, $author_baron_array)){
        $tex_author_title_of_nobility = 19588; //Value matching "Baron" in the
"Tire de noblesse" taxonomy of the Drupal database.
        $author_male = true;
    }
    if(in_array($auteur_strtolower, $author_baroness_array)){
        $tex_author_title_of_nobility = 19589; //Value matching "Baronne" in the
"Tire de noblesse" taxonomy of the Drupal database.
        $author_female = true;
    }
    if(in_array($auteur_strtolower, $author_knight_array)){
        $tex_author_title_of_nobility = 218; //Value matching "Chevalier" in the
"Tire de noblesse" taxonomy of the Drupal database.
        $author_male = true;
    }
    if(in_array($auteur_strtolower, $author_squire_array)){
        $tex_author_title_of_nobility = 13491; //Value matching "Écuyer" in the
"Tire de noblesse" taxonomy of the Drupal database.
        $author_male = true;
    }
    if(in_array($auteur_strtolower, $author_male_president_array)){
        $tex_author_title_of_nobility = 19590; //Value matching "Président" in the
"Tire de noblesse" taxonomy of the Drupal database.
        $author_male = true;
    }
    if(in_array($auteur_strtolower, $author_female_president_array)){
        $tex_author_title_of_nobility = 217; //Value matching "Présidente" in the
"Tire de noblesse" taxonomy of the Drupal database.
        $author_female = true;
    }

    //Exceptions.
```

```
    if($id == 3680 || $id == 3589 || $id == 3954){
        $tex_author_title_of_nobility = 11489;
        $author_male = true;
    }
    if($id == 2602){
        $tex_author_title_of_nobility = 13491;
        $author_male = true;
    }
    if($id == 3571 || $id == 5047 || $id == 95){
        $tex_author_title_of_nobility = 20227;
        $author_male = true;
    }

    //Create field tex_author_gender from auteur_normalise and
activite_normalisee.

    $tex_author_gender = 148; //Value matching "Indéterminé" in the "Genre"
taxonomy of the Drupal database.

    $author_male_array = array("Monsieur", "M. le ", "Abbé ", "Amant ", "Berger ",
"Chevalier ", "Comte ", "Financier ", "Malouin", "Inconnu ", " fils", "Le C. ",
"M. de ", "l'aîné", "le cadet", "le jeune", " comte", " marquis", "Mr ", "Rev. ",
"Sieur ", "Aimable Guibour de la rue Saint-Honoré", "Ampleman de la Cressonnière
(d'), Pierre Marc Antoine François (1733-1809)", "Antonin Dauberjon", "Arnault,
Antoine-Vincent (1766-1834)", "Baillio, Jean-Louis", "Beaubrun, Charles (1604-
1692)", "Belleville, Jean-Marc", "Benoit Lamothe", "Blaizot, Pierre (né vers 1740,
mort en 1808)", "Blanchet, François (1707-1784)", "Boileau, Nicolas (1636-1711)",
"Boussanelle (de), Louis (1720-1788)", "Brossard de Montaney, Jacques (1638-
1702)", "Bruc de Montplaisir (de), René (1610-1682)", "Bruno du Puget", "Campan
[Pierre Dominique François Berthollet (1749-1797)]", "Cerceau (du), Jean-Antoine
(1670-1730)", "Chemilly, François", "Comiers, Claude (mort en 1693)", "Cotin,
Charles (1604-1681)", "Coulard, Louis", "Courchon (abbé)", "Couret de Villeneuve,
Louis-Pierre (1749-1806)", "Coutouly, Alphonse-Henry (1730-1810)", "Del Valle y
Hernandez, Claudio (né en 1721)", "Desforges-Maillard, Paul (1699-1772)",
"Desfossés, Charles-Henri (1764-1851)", "Desmarais, Gaspard-François-Louis (mort
en 1779)", "Desnoyers, Pierre", "Dévot captif", "Docteur Lamarque", "Dufresny,
Charles (1657-1724)", "Duhamel de Rochefort, Luc", "Fontenelle (de), Bernard
(1657-1757)", "François Fournier", "Germain Dauberjon", "Grognet, Pierre (mort
vers 1540)", "Grouber de Groubentall de Linière, Marc-Ferdinand (1739-1815)",
"Guillaume, Jean", "Houdar de La Motte, Antoine (1672-1731)", "J. M. Baudin aîné",
"Jeune logicien de la rue Saint-Louis", "Josephus de Messimy", "La Borderie (de),
Bertrand (né vers 1507)", "La Condamine (de), Charles-Marie (1701-1774)",
"Labrousse de Veyrazet, Jean (1752-1793)", "Laffichard", "L'Affichard", "L'aimable
Guibour de la rue Saint-Honoré", "Lanevère (de), Bernard (1710-1785)", "Laujon,
Pierre (1727-1811)", "Le bel emplâtre", "Le Fèvre de Fontenay, Hardouin (né vers
1686, mort après 1736)", "Le Tellier d'Orvilliers, Charles (1692-1758)", "Le
Tenneur, Louis-Benjamin", "Le Ve Ch. Reg.", "Lesage, Alain-René (1668-1747)", "M.
*** de Fécamp", "M. 3e. J. de Montargis", "M. A. de N.", "M. A. F. Delandoulde",
"M. A. Mauger", "M. A. Perrin", "M. A-a", "M. Abril", "M. Agniel de Valabrix", "M.
Allain", "M. Allard Hurel", "M. Alléon des Goustes", "M. André Honoré", "M.
Anseaume", "M. Arnaud", "M. Arnauld", "M. As*****", "M. Aubry", "M. Aug. Ch...",
"M. Avisse", "M. B**", "M. B***", "M. B. de B.", "M. B. de Bureil..", "M. B. de
N.", "M. B..", "M. B... M.", "M. B...", "M. B.... A. D. C.", "M. B....", "M.
B..... A. D. C.", "M. B..... F.....", "M. B.....", "M. Ba**n", "M. Ba...", "M.
Balme", "M. Bandinucey", "M. Bandivey", "M. Baptestin", "M. Baraion", "M.
Baroche", "M. Barraud", "M. Basileuphile", "M. Baucheron de l'Echerolles", "M.
Baudrais", "M. Baverel", "M. Belle", "M. Benoist", "M. Benoît", "M. Bernard du
Montelet", "M. Berthier", "M. Berton de Chambelle", "M. Bezançon de la Percerie",
```

"M. Bézeau", "M. Bill***", "M. Bl....", "M. Blanchard", "M. Bodin, C. D. F. D. R.", "M. Boidin père", "M. Boinvilliers Foirestier", "M. Boinvilliers", "M. Boirel de Nerval", "M. Boivin du Fey", "M. Bonnay", "M. Bonnefin", "M. Bonnefoi", "M. Bonnefoy", "M. Boran", "M. Bordier", "M. Borel", "M. Borias", "M. Borjon de Cillery", "M. Bouchard", "M. Bouchet", "M. Bouilly", "M. Bouloux", "M. Bourdaloue de Blancafert", "M. Boutroue", "M. Bouvet", "M. Boyer", "M. Br...", "M. Bremont", "M. Brisoult", "M. Brissat", "M. Brondex", "M. Brosse C.", "M. Broüilhet du Rocq", "M. Brun sainte Catherine", "M. Bruyère", "M. Bucquet", "M. Busson", "M. C**", "M. C***", "M. C. de G.", "M. C. d'H.", "M. C. Germaise", "M. C... du Breuil", "M. C...", "M. C.... des Car....", "M. C....", "M. C.....", "M. C......", "M. C.......", "M. Cabanis", "M. Cabour", "M. Candelon", "M. Cassassolles", "M. Cat...", "M. Catimini", "M. Cauchy", "M. Cauville", "M. Ch. M. d. v.", "M. Ch....", "M. Chalumeau", "M. Champion", "M. Champy", "M. Chandorat", "M. Chansierges", "M. Charadin", "M. Chardon", "M. Charle", "M. Charnois", "M. Charron", "M. Chauvel", "M. Chemilly", "M. Chonrai", "M. Chouteau du Plantis", "M. Clément", "M. Clottereau", "M. Cloz", "M. Colas", "M. Colin", "M. Collet", "M. Compan", "M. Conjon", "M. Constant", "M. Contant", "M. Corbilly", "M. Cordier de la Houssaye", "M. Cornesse", "M. Cornu père", "M. Cornu", "M. Cott, P. de B.", "M. Cottereau", "M. Courtat", "M. Cresp. *", "M. Crom...", "M. D**", "M. D***", "M. D***** D.", "M. D*****", "M. d'A. D. C.", "M. Da... D. C...", "M. d'A...", "M. d'Abancourt", "M. d'Abloville", "M. Dalloz", "M. Damas", "M. Damise", "M. d'Angers", "M. Danzel", "M. Danzelle", "M. Darblay", "M. Dareau", "M. Dasnières", "M. Daumont D.", "M. Dauphin", "M. d'Avesne", "M. Davesnes", "M. Daviau", "M. David", "M. d'Azemar", "M. Del***** D. F. D. B.", "M. Delavigne", "M. d'Elbeuf", "M. Delisle. P. A. C. D. B.", "M. Demonron", "M. Demont", "M. d'Epineuil", "M. des Landes", "M. des Robardieres", "M. des Vauroux", "M. Descaz", "M. Descot de la Chesnaye", "M. Desflottes de l'Eichoisier", "M. Desgranges", "M. Desjardins de la Bonnardière", "M. d'Estremau", "M. Deville", "M. d'Évreux", "M. d'Har.", "M. d'Har....", "M. Doizon", "M. Domont", "M. Donnavi, Ga...de...", "M. Dorval de Chacenay", "M. Dracolff", "M. du ***", "M. Du Bled", "M. du Ch.", "M. Du L. D.....", "M. du Moulinet de Hardemare", "M. du P.", "M. du P...rat", "M. du Rion", "M. du Rosoy", "M. du Treüil", "M. Du***", "M. Dubled", "M. Dubosq", "M. Duchemin", "M. Duclauzeau", "M. Dul**", "M. Dul.", "M. Dulaurens", "M. Dumont", "M. Dupin", "M. Dupuis de Bansière", "M. Durion", "M. Duriou", "M. Duruchanoy", "M. Dusuchet", "M. Duvaudel", "M. Duvionau", "M. Elvi", "M. Esmangart de Bournonville", "M. F**", "M. F**. N**.", "M. F... G...", "M. F....", "M. Fabre", "M. Félix de Saint-Legerole", "M. Félix-Nogaret", "M. Ferdairon", "M. Féret", "M. Ferran", "M. Filbert", "M. Fillette", "M. Finot", "M. Fl**", "M. Fl....", "M. Flocard", "M. Follet", "M. Forestier", "M. Fournier de Villecerf", "M. Fournier", "M. Fr. Durruthy", "M. François de Saint-Michel", "M. Fretté", "M. Fréville", "M. Fron....", "M. G**", "M. G***", "M. G**. de V***", "M. G... d. C... p. M...", "M. G... D...", "M. G... de M...", "M. G...", "M. G....", "M. G.....", "M. G..... E. A. B. D. F. M.", "M. Gaillard", "M. Gaillon", "M. Gallays", "M. Gallet", "M. Garillaud", "M. Garnier", "M. Gas...", "M. Gaschignard", "M. Gasteblay Longville", "M. Gaudet.", "M. Gaudriot", "M. Gaultier", "M. Gélannor", "M. Gélede", "M. Gelhay", "M. Gendry", "M. Genneau", "M. Geoffroy", "M. Gervois", "M. Gibier", "M. Gillat", "M. Gillet du Coudray", "M. Gilson", "M. Gir. de Mont", "M. Giron", "M. Girot", "M. Gislain", "M. Glasson de la Severie", "M. Gobet", "M. Godard", "M. Gontier", "M. Gottereau", "M. Goubert", "M. Goudemetz", "M. Grainville", "M. Gratien", "M. Gratton de Saint-Gilles", "M. Grellier", "M. Guay", "M. Guérin du N***ois", "M. Guérin", "M. Gueroult", "M. Guibert", "M. Guichard", "M. Guignet", "M. H**", "M. H***", "M. H.. de F....", "M. H... D...", "M. H....", "M. H.....", "M. H.....", "M. Hal..", "M. Heiligenstein", "M. Helle", "M. Helliot", "M. Heud. Marg.", "M. Hide", "M. Hoch... d'Et...", "M. Holthousen", "M. Houllier de Saint-Remy", "M. Houllier de St Remi", "M. Houssaie", "M. Houssaye", "M. Hubert", "M. Hubert, P. D. M. L. V. D. S. M.", "M. Hubiez", "M. Huet de Longchamps", "M. Huet", "M. Huré", "M. Inhel", "M. Isambert du Freney", "M. Jacques Piron", "M. Jagault", "M. Jalaberd", "M. Jean-Baptiste Calvet", "M. Jeanot.", "M. Joly", "M. Juhel", "M. Jupin", "M. L** Michel", "M. L** R**", "M. L*** F***", "M. L***", "M. L****", "M.

L... C**", "M. L... Prevost", "M. L...", "M. L..... F.....", "M. L.....", "M. L...y, R. D. G.", "M. Laboissiere", "M. Labrouche", "M. Laffichard", "M. Lafite", "M. Lafon", "M. Lafournière", "M. Lambert", "M. Lan.", "M. Langlet", "M. Lapert", "M. Lapeyrade", "M. Lapleigné Ducoudray", "M. Lar...", "M. Larivière", "M. Lassors", "M. Laurent", "M. Laus de Boissy", "M. Lauterest", "M. Lavialle de Masmorel", "M. Lavielle", "M. Lebrun Tossa", "M. Lebrun", "M. Lecat", "M. Lecesne", "M. Leclerc Doze", "M. Lefebvre", "M. Lefevre", "M. Leg....", "M. Leleu d'Aubilly", "M. Lelong", "M. Lemaire", "M. L'Emozy d'Aurel", "M. Lepiniere", "M. Lerom..", "M. Lesbroussart", "M. Lescuyer", "M. Lhomandie", "M. Liégeois", "M. Lienart", "M. Lincoln", "M. Loubert", "M. Louis Guilbaut", "M. Lourdet", "M. Louvet", "M. Lubert", "M. Luneau...", "M. Lyon", "M. M*** de V.", "M. M***", "M. M****", "M. M.***", "M. M... D... V...", "M. M... de R...", "M. M... J...", "M. M...", "M. M....", "M. M.....", "M. M.....l", "M. M..ge", "M. Macé", "M. Maquin", "M. Marlé", "M. Martau", "M. Martel", "M. Martin du Pradeau", "M. Martin", "M. Martineau", "M. Massinot", "M. Mathieu Durousset", "M. Mauclerc", "M. Mazérou", "M. Medy de Seon", "M. Meslé", "M. Mesnard du Montelet", "M. Metairie", "M. Metaledot", "M. Metge", "M. Michellet", "M. Micro-Mégas", "M. Midavaine", "M. Mistelet", "M. Moinardeau", "M. Moisgas", "M. Moline", "M. Montmaur", "M. Moreau", "M. Moulet", "M. Mouron de Chatigny", "M. Mousseau de Malesset", "M. Mulot de la Ménardière", "M. Mulot", "M. Mustel", "M. N...", "M. N.... P.....", "M. N....", "M. Naissant", "M. Naret", "M. Navarre", "M. Niglais", "M. Noël P.....", "M. Noël", "M. Nt.", "M. Onfroy de Breville", "M. Orio", "M. Ortillon", "M. Oye", "M. P**", "M. P***", "M. P...", "M. P....", "M. P.....", "M. Pa...", "M. Paineau", "M. Palmer", "M. Papelart, D.", "M. Parron", "M. Parthon", "M. Pasqueau", "M. Pat*", "M. Pat.", "M. Pat...", "M. Pat....", "M. Patu", "M. Pechaux", "M. Périé de Montalric", "M. Perreaudeau", "M. Peyre", "M. Philosphinx", "M. Phoenix", "M. Picoche", "M. Pillet", "M. Pinault", "M. Piron", "M. Pitoy", "M. Plancher Delande", "M. Portier", "M. Pot, C. d. N.", "M. Pottier", "M. Poulhariez", "M. Pourtales", "M. Praxion", "M. Préaudeau du Pontdont", "M. Prevost de Montigny", "M. Prevost", "M. Prévost", "M. Prévôt de Moka", "M. Privat", "M. Propiac", "M. Pruadere", "M. Puidenot", "M. Quellier", "M. R**", "M. R... T.", "M. R...", "M. R....", "M. R.....", "M. Rafini", "M. Rag...", "M. Rangier", "M. Raoul", "M. Rarrio", "M. Rasicot", "M. Raynaud", "M. Refus", "M. Regnault", "M. Renoud", "M. Rétaux", "M. Révial", "M. Riboulleau", "M. Ricatte d'Huviller", "M. Ricatte", "M. Richard", "M. Rigollot", "M. Robert des Roches", "M. Robert", "M. Robinet", "M. Roger le boiteux et manchot", "M. Roger le boiteux", "M. Roger le manchot", "M. Romet", "M. Rouhier", "M. Rouyer", "M. Rozier", "M. Ruelle", "M. S***", "M. S**. B**.", "M. S... F...", "M. S....", "M. S.....", "M. S....au de S. V.", "M. Saint-G**", "M. San.....", "M. Sant....", "M. Santrin", "M. Sarrauton", "M. Satis", "M. Saute-Montagne", "M. Sauvageot", "M. Sebire de Beauchesne", "M. Subtil", "M. T***", "M. T...", "M. T...é", "M. Tachard", "M. Taconet", "M. Talot", "M. Tan.***", "M. Tarabastane", "M. Tarteron", "M. Tessier", "M. Th. M.", "M. Th. S. D. R.", "M. Thibaut Tanqueux", "M. Tissidre", "M. Trassart", "M. Travenol", "M. Triangle", "M. Troy", "M. Turpin de Cervignières", "M. V... de G...", "M. V...", "M. V....", "M. Valant", "M. Vallois", "M. Vandart", "M. Varé", "M. Vellifredi", "M. Verani de Varenne", "M. Verlhac", "M. Verninac", "M. Victor de Saint-Cl***", "M. Villiers", "M. Vincent", "M. Viniant", "M. Visinet", "M. Voisin", "M. W...", "M. War******", "M. Waton", "Machaut (de), Guillaume (né vers 1300, mort en 1377)", "Mantois galant cher prisé, dans l'enceinte du Palais", "Marquis de la Guesnerie d'Anjou", "Mellet (de), Antoine-Joseph (né vers 1661, mort en 1736)", "Michault, Pierre", "Mignard, Paul (1641-1691)", "Mitron poète", "MM. B. et L.", "MM. P...t C...r", "Mon oncle", "Mr. D. M. V.", "Mr. V.", "Musset de Cogners, Louis-Alexandre-Marie (1753-1839)", "Neveu du cousin Jacques", "Nouveau Lanois persécuté", "Orry de Fulvy, Philibert-Louis (1736-1823)", "Pajon, Henri (mort en 1776)", "Parisien parisien", "Père Agatange", "Père Ducoutau", "Père Maire", "Père Touvex", "Poisson, Raymond (né vers 1630, mort en 1690)", "Pontevès-Gien (de), Henri (1738-1790)", "Prisonnier volontaire", "Regnard, David", "Rousseau, Jean-Jacques (1712-1778)", "Saint-Martin (de), Georges-François (1729-1811)", "Seigneur de Monti", "Semillard des Ovillers (abbé)", "Sr **", "Sr Desmesnils", "Tilly (de),

```
Charles-Louis-Auguste-Casimir-Marie (né en 1754)", "Un amateur du Mercure de
France", "Un amateur impartial", "Un amateur", "Un An. Chan. d'Em.", "Un anonyme",
"Un ass. littér.", "Un C. de R.", "Un C....", "Un Ch.", "Un désoeuvré", "Un jeune
espiègle", "Un membre de la chambre littéraire de Chantepie", "Un membre de la
Société littéraire de Goven", "Un Pr. en P.", "Un solitaire", "Un songeur", "Un
sorcier", "Vienne Plancy (de), Antoine", "Vincheguerre (de), Adrien", "M. *", "M.
**", "M. ***", "M. ****", "M. ***...", "M. D. R***", "M. d. P...., o... a. R...
C... g... i...", "M. D. St. G.", "M. D. V.**", "M. D. V.......", "M. D.***", "M.
D..", "M. D... B...", "M. D... d'H.........", "M. D...", "M. D.....", "M.
D.....", "M. D...Z. G. D. C. D. R. C. E.", "M. L. C. de C.....", "M. L. C. de
P.", "M. L. C. L...", "M. L. C.**", "M. L. C... de B...", "M. L. ch. p. a. C. Q.
d. n.", "M. L. Cl.", "M. L. D. Villen-d'Esel", "M. L. de Rochemont", "M. L.
Demaserie", "M. L. G....", "M. L. M.***", "M. L. M... de F***", "M. L. P. Moline",
"M. L. V. de Ch.");

    $author_female_array = array("Madame", "Mademoiselle", "Mde", "Mme", "Mlle",
"Bergère ", "Comtesse ", "Mad.", "l'aînée", "la cadette", "la jeune", "Me la ", "
Comtesse", " Marquise", "Rév. mère ", "Soeur ", "Belle muse R. Bessiral",
"Blondine Guerin", "Catherine la Chapelle", "Clione et Roselinde", "Cosson de La
Cressonnière, Charlotte-Catherine (1740-1813)", "Dame de Pithiviers", "Dame
solitaire", "Dlles Le F.... et de G....", "Hallencourt de Dromesnil (de),
Charlotte-Françoise (1745-1806)", "Ismène", "Julie de Brulé", "Lorraine
espagnolette", "Marbet (de), Jeanne-Louise-Philippine (dame Mortemart de Boisse)
(1733-1796)", "Muse auxerroise", "Muse prisonnière", "Saliès (de), Antoinette
(1639-1730)", "Société des dames cloîtrées", "Suivante des Muses", "Sylvie du
Havre", "Une société de dames lettrées", "Une société de vendangeuses");

    foreach($author_male_array as $author_male_value){

        if(stripos($auteur_normalise, $author_male_value) !== false){
            $author_male = true;
        }

    }

    foreach($author_female_array as $author_female_value){

        if(stripos($auteur_normalise, $author_female_value)  !== false){
            $author_female = true;
        }

    }

    $author_male_activity_array = array("Conseiller au présidial", "Comédien du
roi", "Prêtre", "Juge", "Abbé", "Étudiant", "Contrôleur des garnisons et mortes-
payes", "Lieutenant général", "Chanoine", "Chantre, chanoine", "Ecclésiastique",
"Révérend père", "Organiste", "Avocat au parlement", "Musicien", "Lieutenant
particulier", "Greffier au présidial", "Maître des eaux et forêts", "Poète",
"Prisonnier", "Maître", "Écolier de rhétorique", "Professeur de rhétorique",
"Écrivain de la marine", "Capitaine général", "Écolier d'humanité", "Curé",
"Maître de pension", "Licencié ès lois", "Ancien mousquetaire du roi", "Commis",
"Notaire, directeur du bureau des postes aux lettres", "Étudiant en médecine",
"Avocat", "Assesseur", "Mousquetaire noir", "Religieux", "Officier de garnison",
"Officier d'artillerie", "Capitaine d'infanterie", "Vicaire", "Peintre",
"Contrôleur des fermes du roi", "Ancien officier de cavalerie", "Invalide",
"Brigadier des armées du roi", "Sous-lieutenant de cavalerie", "Négociant",
"Lieutenant", "Académicien", "Militaire", "Capitaine de cavalerie", "Pensionnaire
au collège", "Médecin", "Bénéficier", "Typographe", "Officier", "Frère
ignorantin", "Vicaire général", "Professeur des humanités", "Bedeau", "Volontaire
```

```
dans une brigade à cheval", "Avocat au parlement, conseiller", "Procureur fiscal",
"Marchand", "Étudiant en droit", "Élève de collège", "Enseigne de vaisseau",
"Licencié en droit", "Membre subalterne d'un bailliage", "Clerc régulier",
"Apothicaire chimiste", "Mestre de camp de cavalerie", "Écolier", "Officier
réformé", "Procureur", "Secrétaire du roi", "Professeur", "Notaire", "Gouverneur",
"Mousquetaire gris", "Distributeur du Mercure de France", "Praticien", "Ancien
maréchal des logis", "Diacre", "Parfumeur", "Receveur des fermes du roi",
"Lieutenant de marine marchande", "Contrôleur ambulant des domaines du roi",
"Étudiant de collège", "Clerc de procureur", "Architecte ingénieur", "Commis des
finances", "Ancien capitaine d'infanterie", "Officier d'infanterie", "Doyen",
"Financier", "Notaire royal", "Premier commis", "Mineur [militaire]", "Libraire,
marchand d'estampes", "Capitaine", "Officier de la chambre de la dauphine",
"Chapelain", "Perruquier", "Directeur de l'Académie royale d'écriture", "Maître de
musique, organiste", "Gendarme de la garde", "Sous-lieutenant de dragons", "Petit
clerc de notaire", "Officier de marine", "Prieur", "Maître ès arts", "Étudiant en
physique", "Contrôleur des tailles", "Contrôleur de la maison du roi", "Lieutenant
général du roi", "Prévôt", "Conseiller du roi", "Docteur en médecine", "Officier
de maison", "Garde des fermes du roi", "Ingénieur pour le roi, lieutenant",
"Maître de poste", "Professeur des mathématiques", "Principal de collège",
"Contrôleur général des finances", "Régent de collège", "Cavalier", "Receveur
général des faux nobles", "Président au parlement", "Homme de lettres", "Ingénieur
des menus plaisirs de Sa Majesté", "Commissaire provincial des guerres",
"Feudiste", "Frère convers", "Gendarme", "Écolier de philosophie", "Colonel
d'infanterie", "Capitaine au corps royal du génie", "Écolier de logique",
"Officier des chevau-légers", "Gendarme du roi", "Grainetier", "Receveur d'une
régie générale", "Souffleur de comédie", "Cordonnier", "Maître perruquier",
"Capitaine en second", "Peintre en éventails", "Savetier", "Ancien mousquetaire",
"Négociant, consul", "Étudiant de rhétorique", "Pandour", "Garde du corps",
"Garde-moulin", "Officier de cavalerie", "Garçon rôtisseur", "Secrétaire",
"Ingénieur", "Concierge général des bâtiments de Monsieur", "Imprimeur du roi",
"Ancien bas-officier de dragons", "Botaniste", "Officier de canonniers",
"Subdélégué, juge", "Rabbin", "Ancien bas-officier d'infanterie", "Officier de
dragons", "Conseiller honoraire au parlement", "Clerc de palais", "Ancien
capitaine aide-major", "Receveur des domaines du roi", "Ancien capitaine de
volontaires", "Visiteur", "Capitaine de dragons", "Capitaine de canonniers",
"Musicien du roi", "Commissaire des guerres", "Aumônier", "Caporal", "Docteur",
"Élève de l'École royale militaire", "Mestre de camp, lieutenant des gardes",
"Commis de la marine", "Chirurgien", "Professeur de géographie, maître d'armes",
"Peintre, professeur de dessin et d'écriture", "Compositeur d'imprimerie",
"Commissaire des classes, retiré", "Religieux étudiant", "Lieutenant des maréchaux
de France", "Commis de la douane", "Paysan", "Dragon", "Contrôleur de la régie",
"Ingénieur-feudiste", "Élève de l'École nationale des ponts et chaussées", "Commis
au bureau de la guerre", "Écoliers", "Commissaire feudiste", "Chirurgien-major",
"Élève", "Pensionnaire du roi", "Commis au bureau du domaine", "Ancien militaire",
"Sergent d'infanterie", "Religieux, étudiant en théologie", "Contrôleur des
aides", "Contrôleur des postes", "Capitaine d'artillerie", "Ingénieur des ponts et
chaussées", "Docteur en médecine, chirurgien-major", "Officier des gardes du
corps", "Maître d'écriture, professeur d'arithmétique", "Ancien vicaire",
"Correcteur d'imprimerie", "Ancien gendarme du roi", "Commis aux fermes",
"Fourrier", "Séminariste", "Secrétaire du cabinet de Madame", "Commissaire des
classes", "Directeur d'imprimerie", "Imprimeur en lettres", "Clerc", "Prieur-
curé", "Professeur de belles-lettres", "Chef d'escadron", "Marguillier", "Ancien
gendarme", "Maître ès arts, maître de pension", "Capitaine aide-major", "Orateur",
"Soldat volontaire", "Officier invalide", "Capitaine de navire", "Instituteur de
la jeunesse", "Homme de loi", "Ancien officier d'infanterie", "Procureur de la
commune", "Régisseur", "Juge de paix", "Receveur des domaines nationaux", "Vicaire
épiscopal", "Employé");
```

```
    $author_female_activity_array = array("Femme de médecin", "Religieuse",
"Trésorière de France", "Faïencière", "Danseuse à l'opéra", "Fille de militaire",
"Révérende mère", "Épouse d'avocat");

    foreach($author_male_activity_array as $author_male_activity_value){

        if(stripos($activite_normalisee, $author_male_activity_value) !== false){
            $author_male = true;
        }

    }

    foreach($author_female_activity_array as $author_female_activity_value){

        if(stripos($activite_normalisee, $author_female_activity_value)  !==
false){
            $author_female = true;
        }

    }

    if((stripos($note, 'un abonné') !== false) || (stripos($note, 'Abonné au
Mercure') !== false) || (stripos($note, 'abonné au Merc[ure]') !== false) ||
(stripos($note, 'un de nos abonnés') !== false) || (stripos($note, '"abonné."')
!== false) || (stripos($note, '"abonné"') !== false) || (stripos($note, 'un
citoyen') !== false) || (stripos($note, '"américain"') !== false)){
            $author_male = true;
        }
    if((stripos($note, 'une jeune Demoiselle de Lyon') !== false) ||
(stripos($note, 'une abonnée') !== false)){
            $author_female = true;
        }

    if($author_male){
        $tex_author_gender = 146; //Value matching "Homme" in the "Genre" taxonomy
of the Drupal database.
    }
    if($author_female){
        $tex_author_gender = 147; //Value matching "Femme" in the "Genre" taxonomy
of the Drupal database.
    }

    //Exceptions.

    $author_male_exception_array = [3719, 2142, 3738, 3749, 3845, 3848, 3924,
3977, 4017, 4061, 4571, 4584, 4589, 4990, 4991, 5076, 5145, 5283, 5335, 5539,
5595, 5931, 5907, 5994, 6153, 6154, 6155, 6187, 6523, 6552, 6688];

    $author_female_exception_array = [3728, 3720, 3587, 3579, 3563, 3545, 3497,
3495, 1855, 1924, 1935, 2149, 3739, 3748, 3970, 4043, 4486, 4495, 4543, 4745,
4919, 5056, 5397, 5515, 5930, 6068, 6155, 6307];

    if (in_array($id, $author_male_exception_array)){
        $tex_author_gender = 146;
    }elseif(in_array($id, $author_female_exception_array)){
        $tex_author_gender = 147;
    }
```

```
    //Create field tex_author_collectivity from auteur_normalise.

    $tex_author_collectivity = '';

    $collectivity_array = array("Deux inséparables d'Abbeville", "MM. B. et L.",
"MM. de Costebelle, Bruyer", "MM. P...t C...r", "Quatre imbéciles", "Société des
dames cloîtrées", "Trois jeunes gens", "Une société de dames lettrées", "Une
société de gens de lettres", "Une société de Saint-Rambert, en Bugey", "Une
société de vendangeuses");

    if(in_array($auteur_normalise, $collectivity_array)){
        $tex_author_collectivity = 1;
    }

    //Exception.

    if($id == 3860){
        $tex_author_collectivity = 1;
    }

    //Create field tex_place from lieu.

    $place_check = $lieu;

    $place_rtrim_exclude_array = array("[Le] N.", "Am.", "Am....", "Aquit.", "Aux
Dames de Sainte Foy. En Agenois.", "B.", "B...", "B....", "Bond....", "C. E. A.",
"C. E. N.", "C. G. E. N.", "C... p. M...", "Chem. E. A.", "Chemillé, E. A.",
"Collége des Q. N.", "Du Port-l'Evêque, en Norm.", "Fougères en Bret.", "G.",
"G...", "L.", "L...", "L.....", "L.......", "M.", "M....", "P.....", "Paroisse de
L.....", "Pondron de S.", "Q.", "R...", "S. H.", "St P.", "St. H. de Ch.", "St.
Jacq. d'Hev...", "T.", "T.......", "V. C.", "V.**.", "Ville d'Aramon, en Lang.");

    if(in_array($place_check, $place_rtrim_exclude_array)){
    }else{
        $place_check = rtrim($place_check, '.');
    }

    //Exceptions.

    if($place_check == 'Rochefort, près Saint-L[ô.]'){
        $place_check = 'Rochefort, près Saint-L[ô]';
    }
    if($place_check == 'V.**.'){
        $place_check = 'V.**';
    }
    if($place_check == 'Du Port-l\'Evêque, en Norm.'){
        $place_check = 'Port-l\'Evêque, en Norm.';
    }
    if($place_check == 'guise'){
        $place_check = 'Guise';
    }

    foreach($taxonomy_term_array as $taxonomy_term_value_4){

        if($taxonomy_term_value_4['vid'] === 'place'){

            if($place_check == $taxonomy_term_value_4['name']){
                $tex_place = $taxonomy_term_value_4['tid'];
            }
```

```
        }

    }

    //Other exceptions.

    if($place_check == 'Toul [...] Gy, en Comté'){
        $tex_place ='"19090,19363"';
    }
    if($place_check == 'Niort [...] Beauvais'){
        $tex_place ='"19125,15166"';
    }
    if($place_check == 'Toul [...] Gy'){
        $tex_place ='"19090,19364"';
    }

    //Create field all_date_gregorian_calendar from date_texte.

    if($date_texte != '0000-00-00'){
        $all_date_gregorian_calendar = $date_texte;
    }

    //Exception.

    if($id == 2159){
        $all_date_gregorian_calendar = '1762-05-05';
    }

    //Create field all_date_republican_calendar. (Remains empty: no text
concerned.)

    $all_date_republican_calendar = '';

    //Create field tex_language.

    $tex_language = 149; //Value matching "Français" in the "Langue" taxonomy.

    if($id == 398 || $id == 419 || $id == 496 || $id == 651 || $id == 669 || $id
== 670 || $id == 698 || $id == 694 || $id == 707 || $id == 717 || $id == 721 ||
$id == 730 || $id == 735 || $id == 736 || $id == 741 || $id == 746 || $id == 751
|| $id == 761 || $id == 762 || $id == 766 || $id == 755 || $id == 756 || $id ==
770 || $id == 771 || $id == 772 || $id == 774 || $id == 776 || $id == 790 || $id
== 791 || $id == 792 || $id == 800 || $id == 809 || $id == 810 || $id == 818 ||
$id == 819 || $id == 824 || $id == 825 || $id == 829 || $id == 833 || $id == 834
|| $id == 835 || $id == 840 || $id == 846 || $id == 853 || $id == 858 || $id ==
862 || $id == 863 || $id == 869 || $id == 870 || $id == 875 || $id == 876 || $id
== 879 || $id == 880 || $id == 883 || $id == 884 || $id == 885 || $id == 891 ||
$id == 895 || $id == 896 || $id == 897 || $id == 902 || $id == 906 || $id == 907
|| $id == 910 || $id == 911 || $id == 916 || $id == 921 || $id == 924 || $id ==
925 || $id == 926 || $id == 929 || $id == 934 || $id == 938 || $id == 951 || $id
== 957 || $id == 977 || $id == 980 || $id == 1020 || $id == 1058 || $id == 1124 ||
$id == 1147 || $id == 1159 || $id == 1167 || $id == 1180 || $id == 1182 || $id ==
1184 || $id == 1193 || $id == 1229 || $id == 1267 || $id == 1275 || $id == 1291 ||
$id == 1317 || $id == 1342 || $id == 1346 || $id == 1347 || $id == 1358 || $id ==
1380 || $id == 1443 || $id == 1927 || $id == 1982 || $id == 1988 || $id == 1997 ||
$id == 2009 || $id == 1936 || $id == 1944 || $id == 1953 || $id == 1954 || $id ==
1957 || $id == 1961 || $id == 1962 || $id == 1965 || $id == 1968 || $id == 1971 ||
$id == 1972 || $id == 1975 || $id == 2121 || $id == 2285 || $id == 2336 || $id ==
```

```
2528 || $id == 2559 || $id == 2579 || $id == 2581 || $id == 2593 || $id == 2597 ||
$id == 2644 || $id == 2664 || $id == 2705 || $id == 2737 || $id == 2749 || $id ==
2885 || $id == 6776){
        $tex_language = 150; //Value for "Latin" in the "Langue" taxonomy.
    }

    if($id == 2848){
        $tex_language = 154; //Value for "Anglais" in the "Langue" taxonomy.
    }

    //Create field tex_paratextual_genre. (Remains empty.)

    $tex_paratextual_genre = '';

    //Create field tex_verse_prose.

    $tex_verse_prose = 172; //Value for "Vers" in the "Vers et prose" taxonomy.

    if(stripos($note, 'prose') !== false){
        $tex_verse_prose = 173; //Value for "Prose" in the "Vers et prose"
taxonomy.
    }

    //Exceptions.

    if($id == 1680){
        $tex_verse_prose = 173; //Value for "Prose" in the "Vers et prose"
taxonomy.
    }
    if($id == 4073){
        $tex_verse_prose = 172; //Value for "Verse" in the "Vers et prose"
taxonomy.
    }

    //Create field tex_type_of_journalistic_writing.

    $tex_type_of_journalistic_writing = '"177,178"'; //Values matching "Jeu
d'esprit / Question" and "Poésie" in the "Type d'écrit journalistique" taxonomy.

    if($tex_verse_prose == 173){
        $tex_type_of_journalistic_writing = 177; //Value matching "Jeu d'esprit /
Question" in the "Type d'écrit journalistique" taxonomy.
    }

    //Create field tex_readers_letters. (Remains empty: no text concerned.)

    $tex_readers_letters = '';

    //Create field tex_literary_genre from genre_1 and genre_2. To give the proper
value, we refer to the "Genre littéraire" taxonomy in the Drupal database.

    if($genre_1 == 'Énigme'){
        $tex_literary_genre = 222; //Value matching "Énigme".
    }
    if($genre_1 == 'Logogriphe'){
        $tex_literary_genre = 13306; //Value matching "Logogriphe".
    }
    if($genre_1 == 'Charade'){
        $tex_literary_genre = 13307; //Value matching "Charade".
```

```
    }
    if($genre_1 == 'Sonnet'){
        if($titre_normalise == 'Sonnet'){
            $tex_literary_genre = 11546; //Value matching "Sonnet".
        }
        elseif($titre_normalise == 'Sonnet énigmatique'){
            $tex_literary_genre = 13308; //Value matching "Sonnet énigmatique".
        }
    }

    //Exceptions.

    if($id == 2574){
        $tex_literary_genre = "13361" ; //Value matching "Anagramme
logogriphique".
    }
    if($id == 2456){
        $tex_literary_genre = "13359" ; //Value matching "Bouts-rimés en énigme".
    }
    if($id == 6063 || $id == 6075){
        $tex_literary_genre = "13307,11816" ; //Value matching "Charade" and
"Conte".
    }
    if($id == 5728){
        $tex_literary_genre = "13307,13306" ; //Value matching "Charade" and
"Logogriphe".
    }
    if($id == 6105){
        $tex_literary_genre = "13307,222,13306" ; //Value matching "Charade",
"Énigme" and "Logogriphe".
    }
    if($id == 2659){
        $tex_literary_genre = "13363" ; //Value matching "Énigme allégorique".
    }
    if($id == 4109){
        $tex_literary_genre = "13546" ; //Value matching "Énigme burlesque".
    }
    if($id == 5010){
        $tex_literary_genre = "13552" ; //Value matching "Énigme-charade".
    }
    if($id == 829){
        $tex_literary_genre = "13329" ; //Value matching "Enigma cum logogripho".
    }
    if($id == 6905){
        $tex_literary_genre = "13557" ; //Value matching "Énigme en bouts-rimés".
    }
    if($id == 1998){
        $tex_literary_genre = "13356" ; //Value matching "Énigme en chanson".
    }
    if($id == 6301){
        $tex_literary_genre = "13555" ; //Value matching "Énigme en charade".
    }
    if($id == 1117 || $id == 1120 || $id == 1133 || $id == 1148 || $id == 1187 ||
$id == 1224 || $id == 1246 || $id == 1248 || $id == 1415 || $id == 2452){
        $tex_literary_genre = "13332" ; //Value matching "Énigme en logogriphe".
    }
    if($id == 4070 || $id == 4071){
        $tex_literary_genre = "13368" ; //Value matching "Énigme en prose".
    }
```

```
    if($id == 6873){
        $tex_literary_genre = "13556" ; //Value matching "Énigme en rondeau".
    }
    if($id == 1737 || $id == 1767 || $id == 1768 || $id == 1770 || $id == 1771 ||
$id == 1772){
        $tex_literary_genre = "13339" ; //Value matching "Énigme en vaudevilles".
    }
    if($id == 3032){
        $tex_literary_genre = "13365" ; //Value matching "Énigme géométrique".
    }
    if($id == 1593){
        $tex_literary_genre = "13336" ; //Value matching "Énigme irrégulière".
    }
    if($id == 961){
        $tex_literary_genre = "13331" ; //Value matching "Énigme logogriphe".
    }
    if($id == 455 || $id == 467 || $id == 2418 || $id == 2468 || $id == 2552 ||
$id == 5142 || $id == 6731){
        $tex_literary_genre = "13326" ; //Value matching "Énigme-logogriphe".
    }
    if($id == 1258 || $id == 1731 || $id == 2521 || $id == 2551 || $id == 2852 ||
$id == 6582){
        $tex_literary_genre = "13335" ; //Value matching "Énigme logogriphique".
    }
    if($id == 578){
        $tex_literary_genre = "13327" ; //Value matching "Énigme manquée".
    }
    if($id == 4833 || $id == 4872){
        $tex_literary_genre = "13550" ; //Value matching "Énigme ou logogriphe".
    }
    if($id == 1880){
        $tex_literary_genre = "13355" ; //Value matching "Énigmogogryphe".
    }
    if($id == 1161 || $id == 3831){
        $tex_literary_genre = "222,13334" ; //Values matching "Énigme" and
"Acrostiche".
    }
    if($id == 141 || $id == 3643 || $id == 3649 || $id == 3641 || $id == 3642 ||
$id == 4178 || $id == 4179 || $id == 4182){
        $tex_literary_genre = "222,11547" ; //Values matching "Énigme" and "Bouts-
rimés".
    }
    if($id == 6391){
        $tex_literary_genre = "222,11513" ; //Values matching "Énigme" and
"Chanson".
    }
    if($id == 4124){
        $tex_literary_genre = "222,13306" ; //Values matching "Énigme" and
"Logogriphe".
    }
    if($id == 472 || $id == 4027 || $id == 4089 || $id == 4167 || $id == 4226 ||
$id == 4254){
        $tex_literary_genre = "222,11546" ; //Values matching "Énigme" and
"Sonnet".
    }
    if($id == 86){
        $tex_literary_genre = "222,11547,12093" ; //Values matching "Énigme",
"Bouts-rimés" and "Parodie".
    }
```

```
    if($id == 726){
        $tex_literary_genre = "222,11546,11547" ; //Values matching "Énigme",
"Sonnet" and "Bouts-rimés".
    }
    if($id == 4133){
        $tex_literary_genre = "13547" ; //Value matching "Épigramme énigmatique".
    }
    if($id == 2581 || $id == 2582){
        $tex_literary_genre = "13362" ; //Value matching "Épitaphe énigmatique".
    }
    if($id == 3156){
        $tex_literary_genre = "13366" ; //Value matching "Épître logogriphique".
    }
    if($id == 4073){
        $tex_literary_genre = "13369" ; //Value matching "Explication
énigmatique".
    }
    if($id == 3721){
        $tex_literary_genre = "13367" ; //Value matching "Fable énigmatique".
    }
    if($id == 6284){
        $tex_literary_genre = "13554" ; //Value matching "Logogriphe
anagrammatique".
    }
    if($id == 2526){
        $tex_literary_genre = "13360" ; //Value matching "Logogriphe
arithmétique".
    }
    if($id == 4964){
        $tex_literary_genre = "13551" ; //Value matching "Logogriphe-charade en
calembours".
    }
    if($id == 1785){
        $tex_literary_genre = "13354" ; //Value matching "Logogriphe en chanson".
    }
    if($id == 842 || $id == 959 || $id == 968 || $id == 1320 || $id == 1323){
        $tex_literary_genre = "13330" ; //Value matching "Logogriphe énigmatique".
    }
    if($id == 748 || $id == 1827){
        $tex_literary_genre = "13328" ; //Value matching "Logogriphe en sonnet".
    }
    if($id == 5484){
        $tex_literary_genre = "13553" ; //Value matching "Logogriphe en stances".
    }
    if($id == 1733 || $id == 1746){
        $tex_literary_genre = "13338" ; //Value matching "Logogriphe en
vaudevilles".
    }
    if($id == 3030){
        $tex_literary_genre = "13364" ; //Value matching "Logogriphe grammairien".
    }
    if($id == 1821){
        $tex_literary_genre = "13306,13334" ; //Values matching "Logogriphe" and
"Acrostiche".
    }
    if($id == 684){
        $tex_literary_genre = "13306,12361" ; //Values matching "Logogriphe" and
"Bouquet".
    }
```

```
    if($id == 280){
        $tex_literary_genre = "13306,11547" ; //Values matching "Logogriphe" and
"Bouts-rimés".
    }
    if($id == 1134){
        $tex_literary_genre = "13333" ; //Value matching "Ode énigmatique".
    }
    if($id == 2447 || $id == 2448){
        $tex_literary_genre = "13358" ; //Value matching "Proverbe en énigme".
    }
    if($id == 379){
        $tex_literary_genre = "13325" ; //Value matching "Quatrain énigmatique".
    }
    if($id == 4694){
        $tex_literary_genre = "13549" ; //Value matching "Question énigmatique".
    }
    if($id == 4176){
        $tex_literary_genre = "13548" ; //Value matching "Rondeau énigmatique".
    }
    if($id == 507 || $id == 695 || $id == 2508 || $id == 3978 || $id == 4023 ||
$id == 4026 || $id == 4064){
        $tex_literary_genre = "13308" ; //Value matching "Sonnet énigmatique".
    }
    if($id == 1762){
        $tex_literary_genre = "13340" ; //Value matching "Sonnet en logogriphe".
    }
    if($id == 106 || $id == 726){
        $tex_literary_genre = "13308,11547" ; //Values matching "Sonnet
énigmatique", "Bouts-rimés".
    }
    if($id == 1605){
        $tex_literary_genre = "13337" ; //Value matching "Stances logogriphiques".
    }
    if($id == 2267){
        $tex_literary_genre = "13357" ; //Value matching "Vers énigmatiques".
    }

    //Manage commas.
    if(stripos($tex_literary_genre, ",") !== false){
        $tex_literary_genre = '"'.$tex_literary_genre.'"';
    }

    //Create field tex_is_part_of_iss from date_volume and numero_volume.

    $enigma_issue_check = $date_volume.'-'.$numero_volume;

    foreach($iss_peri_dig_cop_array as $iss_peri_dig_cop_value_2){

        $issue_date =
$iss_peri_dig_cop_value_2['issue_all_date_gregorian_calendar'];
        $issue_volume = $iss_peri_dig_cop_value_2['issue_volume'];
        $issue_number = $iss_peri_dig_cop_value_2['issue_number'];

        if($iss_peri_dig_cop_value_2['issue_volume'] == ''){
            $issue_volume = '0';
        }
        if($iss_peri_dig_cop_value_2['issue_number'] == ''){
            $issue_number = '0';
        }
```

```
        $issue_volume_check = $issue_date.'-'.$issue_volume;
        $issue_number_check = $issue_date.'-'.$issue_number;

        if($iss_peri_dig_cop_value_2['periodical_id'] == '1080293' ||
$iss_peri_dig_cop_value_2['periodical_id'] == '1097012' ||
$iss_peri_dig_cop_value_2['periodical_id'] == '1104066' ||
$iss_peri_dig_cop_value_2['periodical_id'] == '1104181' ||
$iss_peri_dig_cop_value_2['periodical_id'] == '1104502' ||
$iss_peri_dig_cop_value_2['periodical_id'] == '1104537' ||
$iss_peri_dig_cop_value_2['periodical_id'] == '1105906' ||
$iss_peri_dig_cop_value_2['periodical_id'] == '1350967' ||
$iss_peri_dig_cop_value_2['periodical_id'] == '1105907' ||
$iss_peri_dig_cop_value_2['periodical_id'] == '1105908'){
        }
        elseif($enigma_issue_check == $issue_volume_check || $enigma_issue_check
== $issue_number_check){
            $tex_is_part_of_iss = $iss_peri_dig_cop_value_2['issue_nid'];
        }

    }

    //Exceptions.

    if($id == 4037){
        $tex_is_part_of_iss = 1382966;
    }

    if($id == 4070){
        $tex_is_part_of_iss = 1382564;
    }

    if($id == 4071){
        $tex_is_part_of_iss = 1382589;
    }

    if($id == 4072){
        $tex_is_part_of_iss = 1382589;
    }

    if($id == 4073){
        $tex_is_part_of_iss = 1382594;
    }

    if($id == 4074){
        $tex_is_part_of_iss = 1382616;
    }

    if($date_check == '1683-07-01'){
        $tex_is_part_of_iss = 1382608;
    }

    if($date_check == '1684-06-01'){
        $tex_is_part_of_iss = 1382623;
    }

    if($date_check == '1686-10-01'){
        $tex_is_part_of_iss = 1382662;
    }
```

```
if($date_check == '1688-09-01'){
    if($id == 3794 || $id == 3795){
        $tex_is_part_of_iss = 1382696;
    }else{
        $tex_is_part_of_iss = 1382697;
    }
}

if($date_check == '1688-11-01'){
    $tex_is_part_of_iss = 1382700;
}

if($date_check == '1690-07-01'){
    $tex_is_part_of_iss = 1382730;
}

if($date_check == '1692-06-01'){
    $tex_is_part_of_iss = 1382766;
}

if($date_check == '1692-07-01'){
    $tex_is_part_of_iss = 1382768;
}

if($date_check == '1692-08-01'){
    $tex_is_part_of_iss = 1382770;
}

if($date_check == '1693-02-01'){
    $tex_is_part_of_iss = 1382777;
}

if($date_check == '1693-08-01'){
    $tex_is_part_of_iss = 1382784;
}

if($date_check == '1693-10-01'){
    $tex_is_part_of_iss = 1382787;
}

if($date_check == '1695-10-01'){
    $tex_is_part_of_iss = 1382812;
}

if($date_check == '1702-02-01'){
    $tex_is_part_of_iss = 1382891;
}

if($date_check == '1702-06-01'){
    $tex_is_part_of_iss = 1382896;
}

if($date_check == '1702-07-01'){
    $tex_is_part_of_iss = 1382898;
}

if($date_check == '1702-10-01'){
    $tex_is_part_of_iss = 1382902;
```

```
    }

    if($date_check == '1704-03-01'){
        $tex_is_part_of_iss = 1382921;
    }

    if($date_check == '1708-07-01'){
        $tex_is_part_of_iss = 1382977;
    }

    if($date_check == '1709-06-01'){
        $tex_is_part_of_iss = 1382989;
    }

    if($date_check == '1715-12-01'){
        $tex_is_part_of_iss = 1383065;
    }

    if($date_check == '1722-09-01'){
        if($id == 4364){
            $tex_is_part_of_iss = 1383148;
        }else{
            $tex_is_part_of_iss = 1383147;
        }
    }

    if($date_check == '1731-04-01'){
        if($id == 363){
            $tex_is_part_of_iss = 1383270;
        }else{
            $tex_is_part_of_iss = 1383269;
        }
    }

    //Create field tex_is_part_of_sec.

    $tex_is_part_of_sec = '';

    //Exceptions.

    if($id > 442 && $id < 446){
        $tex_is_part_of_sec = 2550246; //ID of the section "Pieces fugitives en
Vers et en Prose" (December 1732, vol. 2) in the Drupal database.
    }
    elseif($id > 437 && $id < 443){
        $tex_is_part_of_sec = 2550196; //ID of the section "Pieces fugitives en
Vers et en Prose" (December 1732, vol. 1) in the Drupal database.
    }
    elseif($id > 433 && $id < 438){
        $tex_is_part_of_sec = 2550153; //ID of the section "Pieces fugitives en
Vers et en Prose" (November 1732) in the Drupal database.
    }
    elseif($id > 430 && $id < 434){
        $tex_is_part_of_sec = 2550113; //ID of the section "Pieces fugitives en
Vers et en Prose" (October 1732) in the Drupal database.
    }
    elseif($id > 425 && $id < 431){
        $tex_is_part_of_sec = 2550101; //ID of the section "Pieces fugitives en
Vers et en Prose" (September 1732) in the Drupal database.
```

```
    }
    elseif($id > 422 && $id < 426){
        $tex_is_part_of_sec = 2549890; //ID of the section "Pieces fugitives en
Vers et en Prose" (August 1732) in the Drupal database.
    }
    elseif($id > 419 && $id < 423){
        $tex_is_part_of_sec = 2549601; //ID of the section "Pieces fugitives en
Vers et en Prose" (July 1732) in the Drupal database.
    }
    elseif($id > 415 && $id < 420){
        $tex_is_part_of_sec = 2549484; //ID of the section "Pieces fugitives en
Vers et en Prose" (June 1732, vol. 2) in the Drupal database.
    }
    elseif($id > 412 && $id < 416){
        $tex_is_part_of_sec = 2549321; //ID of the section "Pieces fugitives en
Vers et en Prose" (June 1732, vol. 1) in the Drupal database.
    }
    elseif($id > 409 && $id < 413){
        $tex_is_part_of_sec = 2549154; //ID of the section "Pieces fugitives en
Vers et en Prose" (May 1732) in the Drupal database.
    }
    elseif($id > 405 && $id < 410){
        $tex_is_part_of_sec = 2549078; //ID of the section "Pieces fugitives en
Vers et en Prose" (April 1732) in the Drupal database.
    }
    elseif($id > 401 && $id < 406){
        $tex_is_part_of_sec = 2548987; //ID of the section "Pieces fugitives en
Vers et en Prose" (March 1732) in the Drupal database.
    }
    elseif($id > 398 && $id < 402){
        $tex_is_part_of_sec = 2548920; //ID of the section "Pieces fugitives en
Vers et en Prose" (February 1732) in the Drupal database.
    }
    elseif($id > 395 && $id < 399){
        $tex_is_part_of_sec = 2548860; //ID of the section "Pieces fugitives en
Vers et en Prose" (January 1732) in the Drupal database.
    }
    elseif($id > 391 && $id < 396){
        $tex_is_part_of_sec = 2550330; //ID of the section "Pieces fugitives en
Vers et en Prose" (December 1731, vol. 2) in the Drupal database.
    }
    elseif($id > 385 && $id < 389){
        $tex_is_part_of_sec = 2550327; //ID of the section "Pieces fugitives en
Vers et en Prose" (November 1731) in the Drupal database.
    }
    elseif($id > 382 && $id < 386){
        $tex_is_part_of_sec = 2550301; //ID of the section "Pieces fugitives en
Vers et en Prose" (October 1731) in the Drupal database.
    }
    elseif($id > 372 && $id < 376){
        $tex_is_part_of_sec = 2550284; //ID of the section "Pieces fugitives en
Vers et en Prose" (July 1731) in the Drupal database.
    }
    elseif($id > 369 && $id < 373){
        $tex_is_part_of_sec = 2550277; //ID of the section "Pieces fugitives en
Vers et en Prose" (June 1731, vol. 2) in the Drupal database.
    }
    elseif($id > 366 && $id < 370){
```

```
        $tex_is_part_of_sec = 2550669; //ID of the section "Pieces fugitives en
Vers et en Prose" (June 1731, vol. 1) in the Drupal database.
    }
    elseif($id > 363 && $id < 367){
        $tex_is_part_of_sec = 2549954; //ID of the section "Pieces fugitives en
Vers et en Prose" (May 1731) in the Drupal database.
    }
    elseif($id > 359 && $id < 364){
        $tex_is_part_of_sec = 2549583; //ID of the section "Pieces fugitives en
Vers et en Prose" (April 1731) in the Drupal database.
    }
    elseif($id > 356 && $id < 360){
        $tex_is_part_of_sec = 2549313; //ID of the section "Pieces fugitives en
Vers et en Prose" (March 1731) in the Drupal database.
    }
    elseif($id > 352 && $id < 357){
        $tex_is_part_of_sec = 2548567; //ID of the section "Pieces fugitives en
Vers et en Prose" (February 1731) in the Drupal database.
    }
    elseif($id > 349 && $id < 353){
        $tex_is_part_of_sec = 2548383; //ID of the section "Pieces fugitives en
Vers et en Prose" (January 1731) in the Drupal database.
    }
    elseif($id > 346 && $id < 350){
        $tex_is_part_of_sec = 2550638; //ID of the section "Pieces fugitives en
Vers et en Prose" (December 1730, vol. 2) in the Drupal database.
    }
    elseif($id > 342 && $id < 347){
        $tex_is_part_of_sec = 2550604; //ID of the section "Pieces fugitives en
Vers et en Prose" (December 1730, vol. 1) in the Drupal database.
    }
    elseif($id > 339 && $id < 343){
        $tex_is_part_of_sec = 2550410; //ID of the section "Pieces fugitives en
Vers et en Prose" (November 1730) in the Drupal database.
    }
    elseif($id > 336 && $id < 340){
        $tex_is_part_of_sec = 2550387; //ID of the section "Pieces fugitives en
Vers et en Prose" (October 1730) in the Drupal database.
    }
    elseif($id > 333 && $id < 337){
        $tex_is_part_of_sec = 2550299; //ID of the section "Pieces fugitives en
Vers et en Prose" (September 1730) in the Drupal database.
    }
    elseif($id > 331 && $id < 334){
        $tex_is_part_of_sec = 2550212; //ID of the section "Pieces fugitives en
Vers et en Prose" (August 1730) in the Drupal database.
    }
    elseif($id > 329 && $id < 332){
        $tex_is_part_of_sec = 2550091; //ID of the section "Pieces fugitives en
Vers et en Prose" (July 1730) in the Drupal database.
    }
    elseif($id == 328 || $id == 329){
        $tex_is_part_of_sec = 2549686; //ID of the section "Pieces fugitives en
Vers et en Prose" (June 1730, vol. 2) in the Drupal database.
    }
    elseif($id == 326 || $id == 327){
        $tex_is_part_of_sec = 2549419; //ID of the section "Pieces fugitives en
Vers et en Prose" (June 1730, vol. 1) in the Drupal database.
    }
```

```
    elseif($id == 323 || $id == 324 || $id == 325){
        $tex_is_part_of_sec = 2549212; //ID of the section "Pieces fugitives en
Vers et en Prose" (May 1730) in the Drupal database.
    }
    elseif($id == 522 || $id == 523 || $id == 524 || $id == 525 || $id == 526){
        $tex_is_part_of_sec = 2550509; //ID of the section "Pieces fugitives en
Vers et en Prose" (February 1734) in the Drupal database.
    }
    elseif($id == 518 || $id == 519 || $id == 520 || $id == 521){
        $tex_is_part_of_sec = 2550485; //ID of the section "Pieces fugitives en
Vers et en Prose" (January 1734) in the Drupal database.
    }
    elseif($id == 515 || $id == 516 || $id == 517){
        $tex_is_part_of_sec = 2550480; //ID of the section "Pieces fugitives en
Vers et en Prose" (December 1733, vol. 2) in the Drupal database.
    }
    elseif($id == 510 || $id == 511 || $id == 512 || $id == 513 || $id == 514){
        $tex_is_part_of_sec = 2550477; //ID of the section "Pieces fugitives en
Vers et en Prose" (December 1733, vol. 1) in the Drupal database.
    }
    elseif($id == 506 || $id == 507 || $id == 508 || $id == 509){
        $tex_is_part_of_sec = 2550474; //ID of the section "Pieces fugitives en
Vers et en Prose" (November 1733) in the Drupal database.
    }
    elseif($id == 501 || $id == 502 || $id == 503 || $id == 504 || $id == 505){
        $tex_is_part_of_sec = 2550454; //ID of the section "Pieces fugitives en
Vers et en Prose" (October 1733) in the Drupal database.
    }
    elseif($id == 497 || $id == 498 || $id == 499 || $id == 500){
        $tex_is_part_of_sec = 2550450; //ID of the section "Pieces fugitives en
Vers et en Prose" (September 1733) in the Drupal database.
    }
    elseif($id == 490 || $id == 491 || $id == 492 || $id == 493 || $id == 494 ||
$id == 495 || $id == 496){
        $tex_is_part_of_sec = 2550448; //ID of the section "Pieces fugitives en
Vers et en Prose" (August 1733) in the Drupal database.
    }
    elseif($id == 484 || $id == 485 || $id == 486 || $id == 487 || $id == 488 ||
$id == 489){
        $tex_is_part_of_sec = 2550413; //ID of the section "Pieces fugitives en
Vers et en Prose" (July 1733) in the Drupal database.
    }
    elseif($id == 478 || $id == 479 || $id == 480 || $id == 481 || $id == 482 ||
$id == 483){
        $tex_is_part_of_sec = 2550308; //ID of the section "Pieces fugitives en
Vers et en Prose" (June 1733, vol. 2) in the Drupal database.
    }
    elseif($id == 472 || $id == 473 || $id == 474 || $id == 475 || $id == 476 ||
$id == 477){
        $tex_is_part_of_sec = 2550223; //ID of the section "Pieces fugitives en
Vers et en Prose" (June 1733, vol. 1) in the Drupal database.
    }
    elseif($id == 465 || $id == 466 || $id == 467 || $id == 468 || $id == 469 ||
$id == 470 || $id == 471){
        $tex_is_part_of_sec = 2550121; //ID of the section "Pieces fugitives en
Vers et en Prose" (May 1733) in the Drupal database.
    }
    elseif($id == 459 || $id == 460 || $id == 461 || $id == 462 || $id == 463 ||
$id == 464){
```

```
        $tex_is_part_of_sec = 2549958; //ID of the section "Pieces fugitives en
Vers et en Prose" (April 1733) in the Drupal database.
    }
    elseif($id == 454 || $id == 455 || $id == 456 || $id == 457 || $id == 458){
        $tex_is_part_of_sec = 2549575; //ID of the section "Pièces fugitives en
vers et en prose" (March 1733) in the Drupal database.
    }
    elseif($id == 450 || $id == 451 || $id == 452 || $id == 453){
        $tex_is_part_of_sec = 2547865; //ID of the section "Pièces fugitives en
Vers et en Prose" (February 1733) in the Drupal database.
    }
    elseif($id == 446 || $id == 447 || $id == 448 || $id == 449){
        $tex_is_part_of_sec = 2549048; //ID of the section "Pieces fugitives, en
Vers et en Prose" (January 1733) in the Drupal database.
    }
    elseif($id == 4144 || $id == 4145){
        $tex_is_part_of_sec = 2548414; //ID of the section "Article des énigmes"
(February 1713) in the Drupal database.
    }
    elseif($id == 4101 || $id == 4102){
        $tex_is_part_of_sec = 2548341; //ID of the section "Énigmes" (March 1712)
in the Drupal database.
    }
    elseif($id == 4088){
        $tex_is_part_of_sec = 2548115; //ID of the section "Article des énigmes"
(March 1711) in the Drupal database.
    }
    elseif($id == 243 || $id == 244 || $id == 245 || $id == 246 || $id == 247 ||
$id == 248 || $id == 249 || $id == 250){
        $tex_is_part_of_sec = 2547336; //ID of the section "Énigmes et
logogryphes" (July 1728) in the Drupal database.
    }
    elseif($id == 280){
        $tex_is_part_of_sec = 2547337; //ID of the section "Bouts-rimez remplis"
(April 1729) in the Drupal database.
    }
    elseif($id == 1143 || $id == 1144 || $id == 1145 || $id == 1146 || $id ==
1147){
        $tex_is_part_of_sec = 2547555; //ID of the section "Énigmes et logogryphe"
(May 1745) in the Drupal database.
    }
    elseif($id == 1152 || $id == 1153 || $id == 1154){
        $tex_is_part_of_sec = 2547556; //ID of the section "Énigme et logogryphes"
(June 1745, vol. 2) in the Drupal database.
    }
    elseif($id == 1155 || $id == 1156 || $id == 1157 || $id == 1158){
        $tex_is_part_of_sec = 2547557; //ID of the section "Énigme et logogriphes"
(July 1745) in the Drupal database.
    }
    elseif($id == 1159 || $id == 1160 || $id == 1161 || $id == 1162 || $id ==
1163){
        $tex_is_part_of_sec = 2547558; //ID of the section "Énigmes et
logogriphes" (August 1745) in the Drupal database.
    }
    elseif($id == 1166 || $id == 1167 || $id == 1168 || $id == 1169){
        $tex_is_part_of_sec = 2547559; //ID of the section "Énigmes et
logogryphes" (October 1745) in the Drupal database.
    }
    elseif($id == 1173 || $id == 1174 || $id == 1175){
```

```
        $tex_is_part_of_sec = 2547560; //ID of the section "Énigme et logogryphes"
(December 1745, vol. 1) in the Drupal database.
    }
    elseif($id == 1179 || $id == 1180 || $id == 1181){
        $tex_is_part_of_sec = 2547561; //ID of the section "Énigme et logogryphes"
(January 1746) in the Drupal database.
    }
    elseif($id == 4788 || $id == 4789 || $id == 4790 || $id == 4791){
        $tex_is_part_of_sec = 2547656; //ID of the section "Charades" (4 May 1782)
in the Drupal database.
    }
    elseif($id == 4865 || $id == 4866 || $id == 4867 || $id == 4868 || $id == 4869
|| $id == 4870){
        $tex_is_part_of_sec = 2547657; //ID of the section "Charades" (28 December
1782) in the Drupal database.
    }
    elseif($id == 4895 || $id == 4896 || $id == 4897 || $id == 4898 || $id ==
4899){
        $tex_is_part_of_sec = 2547679; //ID of the section "Charades" (1 March
1783) in the Drupal database.
    }
    elseif($id == 4908 || $id == 4909 || $id == 4910){
        $tex_is_part_of_sec = 2547680; //ID of the section "Charades" (22 March
1783) in the Drupal database.
    }
    elseif($id == 4924 || $id == 4925){
        $tex_is_part_of_sec = 2547681; //ID of the section "Charades" (3 May 1783)
in the Drupal database.
    }
    elseif($id == 4928 || $id == 4929 || $id == 4930 || $id == 4931){
        $tex_is_part_of_sec = 2547682; //ID of the section "Charades" (10 May
1783) in the Drupal database.
    }
    elseif($id == 4934 || $id == 4935 || $id == 4936){
        $tex_is_part_of_sec = 2547683; //ID of the section "Charades" (17 May
1783) in the Drupal database.
    }
    elseif($id == 4085 || $id == 4086 || $id == 4087){
        $tex_is_part_of_sec = 2547735; //ID of the section "Article des énigmes"
(February 1711) in the Drupal database.
    }
    elseif($id == 6429){
        $tex_is_part_of_sec = 2547841; //ID of the section "Pièces fugitives en
vers et en prose" (16 December 1792) in the Drupal database.
    }
    elseif($id == 6430){
        $tex_is_part_of_sec = 2547842; //ID of the section "Pièces fugitives en
vers et en prose" (17 December 1792) in the Drupal database.
    }
    elseif($id == 6431){
        $tex_is_part_of_sec = 2547843; //ID of the section "Pièces fugitives en
vers et en prose" (18 December 1792) in the Drupal database.
    }
    elseif($id == 527 || $id == 528 || $id == 529 || $id == 530){
        $tex_is_part_of_sec = 2551101; //ID of the section "Pièces fugitives en
vers et en prose" (March 1734) in the Drupal database.
    }
    elseif($id == 531 || $id == 532 || $id == 533 || $id == 534){
```

```
        $tex_is_part_of_sec = 2551102; //ID of the section "Pièces fugitives en
vers et en prose" (April 1734) in the Drupal database.
    }
    elseif($id == 535 || $id == 536 || $id == 537 || $id == 538 || $id == 539){
        $tex_is_part_of_sec = 2551103; //ID of the section "Pièces fugitives en
vers et en prose" (May 1734) in the Drupal database.
    }
    elseif($id == 540 || $id == 541 || $id == 542 || $id == 543){
        $tex_is_part_of_sec = 2551104; //ID of the section "Pièces fugitives en
vers et en prose" (June 1734, vol. 1) in the Drupal database.
    }
    elseif($id == 544 || $id == 545 || $id == 546){
        $tex_is_part_of_sec = 2551105; //ID of the section "Pièces fugitives en
vers et en prose" (June 1734, vol. 2) in the Drupal database.
    }
    elseif($id == 376 || $id == 377 || $id == 378 || $id == 379){
        $tex_is_part_of_sec = 2550982; //ID of the section "Pièces fugitives en
vers et en prose" (August 1731) in the Drupal database.
    }
    elseif($id == 380 || $id == 381 || $id == 382){
        $tex_is_part_of_sec = 2551089; //ID of the section "Pièces fugitives en
vers et en prose" (September 1731) in the Drupal database.
    }
    elseif($id == 389 || $id == 390 || $id == 391){
        $tex_is_part_of_sec = 2551130; //ID of the section "Pièces fugitives en
vers et en prose" (December 1731, vol. 1) in the Drupal database.
    }

    //Create field tex_is_reproduction_of_tex. (Remains empty.)

    $tex_is_reproduction_of_tex = '';

    //Create field tex_is_continuation_of_tex. (Remains empty.)

    $tex_is_continuation_of_tex = '';

    //Create field tex_is_response_to_tex. (Remains empty.)

    $tex_is_response_to_tex = '';

    //Create field tex_is_written_by_per from auteur_normalise.

    if($auteur_normalise == 'Ampleman de la Cressonnière (d\'), Pierre Marc
Antoine François (1733-1809)'){
        $tex_is_written_by_per = 2550975;
    }
    elseif($auteur_normalise == 'Arnault, Antoine-Vincent (1766-1834)'){
        $tex_is_written_by_per = 209557;
    }
    elseif($auteur_normalise == 'Beaubrun, Charles (1604-1692)'){
        $tex_is_written_by_per = 170043;
    }
    elseif($auteur_normalise == 'Blaizot, Pierre (né vers 1740, mort en 1808)'){
        $tex_is_written_by_per = 197587;
    }
    elseif($auteur_normalise == 'Blanchet, François (1707-1784)'){
        $tex_is_written_by_per = 198504;
    }
    elseif($auteur_normalise == 'Boileau, Nicolas (1636-1711)'){
```

```
        $tex_is_written_by_per = 163497;
    }
    elseif($auteur_normalise == 'Boussanelle (de), Louis (1720-1788)'){
        $tex_is_written_by_per = 199885;
    }
    elseif($auteur_normalise == 'Brossard de Montaney, Jacques (1638-1702)'){
        $tex_is_written_by_per = 167329;
    }
    elseif($auteur_normalise == 'Bruc de Montplaisir (de), René (1610-1682)'){
        $tex_is_written_by_per = 167190;
    }
    elseif($auteur_normalise == 'Campan [Pierre Dominique François Berthollet
(1749-1797)]'){
        $tex_is_written_by_per = 2550976;
    }
    elseif($auteur_normalise == 'Cerceau (du), Jean-Antoine (1670-1730)'){
        $tex_is_written_by_per = 162803;
    }
    elseif($auteur_normalise == 'Comiers, Claude (mort en 1693)'){
        $tex_is_written_by_per = 136198;
    }
    elseif($auteur_normalise == 'Cosson de La Cressonnière, Charlotte-Catherine
(1740-1813)'){
        $tex_is_written_by_per = 192068;
    }
    elseif($auteur_normalise == 'Cotin, Charles (1604-1681)'){
        $tex_is_written_by_per = 171963;
    }
    elseif($auteur_normalise == 'Couret de Villeneuve, Louis-Pierre (1749-1806)'){
        $tex_is_written_by_per = 227095;
    }
    elseif($auteur_normalise == 'Coutouly, Alphonse-Henry (1730-1810)'){
        $tex_is_written_by_per = 2551001;
    }
    elseif($auteur_normalise == 'Del Valle y Hernandez, Claudio (né en 1721)'){
        $tex_is_written_by_per = 2551005;
    }
    elseif($auteur_normalise == 'Desforges-Maillard, Paul (1699-1772)'){
        $tex_is_written_by_per = 174458;
    }
    elseif($auteur_normalise == 'Desfossés, Charles-Henri (1764-1851)'){
        $tex_is_written_by_per = 2551008;
    }
    elseif($auteur_normalise == 'Desmarais, Gaspard-François-Louis (mort en
1779)'){
        $tex_is_written_by_per = 2551012;
    }
    elseif($auteur_normalise == 'Dufresny, Charles (1657-1724)'){
        $tex_is_written_by_per = 165667;
    }
    elseif($auteur_normalise == 'Fontenelle (de), Bernard (1657-1757)'){
        $tex_is_written_by_per = 172503;
    }
    elseif($auteur_normalise == 'Grognet, Pierre (mort vers 1540)'){
        $tex_is_written_by_per = 149852;
    }
    elseif($auteur_normalise == 'Grouber de Groubentall de Linière, Marc-Ferdinand
(1739-1815)'){
        $tex_is_written_by_per = 178014;
```

```
    }
    elseif($auteur_normalise == 'Hallencourt de Dromesnil (de), Charlotte-
Françoise (1745-1806)'){
        $tex_is_written_by_per = 2551018;
    }
    elseif($auteur_normalise == 'Houdar de La Motte, Antoine (1672-1731)'){
        $tex_is_written_by_per = 165599;
    }
    elseif($auteur_normalise == 'La Borderie (de), Bertrand (né vers 1507)'){
        $tex_is_written_by_per = 140423;
    }
    elseif($auteur_normalise == 'La Condamine (de), Charles-Marie (1701-1774)'){
        $tex_is_written_by_per = 199802;
    }
    elseif($auteur_normalise == 'Labrousse de Veyrazet, Jean (1752-1793)'){
        $tex_is_written_by_per = 2551019;
    }
    elseif($auteur_normalise == 'Lanevère (de), Bernard (1710-1785)'){
        $tex_is_written_by_per = 2550819;
    }
    elseif($auteur_normalise == 'Larcher, Jean-Chrysostôme (comte de La Touraille)
(1720-1794)'){
        $tex_is_written_by_per = 191562;
    }
    elseif($auteur_normalise == 'Laujon, Pierre (1727-1811)'){
        $tex_is_written_by_per = 200970;
    }
    elseif($auteur_normalise == 'Le Fèvre de Fontenay, Hardouin (né vers 1686,
mort après 1736)'){
        $tex_is_written_by_per = 1295653;
    }
    elseif($auteur_normalise == 'Le Tellier d\'Orvilliers, Charles (1692-1758)'){
        $tex_is_written_by_per = 2549985;
    }
    elseif($auteur_normalise == 'Lesage, Alain-René (1668-1747)'){
        $tex_is_written_by_per = 172487;
    }
    elseif($auteur_normalise == 'Machaut (de), Guillaume (né vers 1300, mort en
1377)'){
        $tex_is_written_by_per = 319085;
    }
    elseif($auteur_normalise == 'Marbet (de), Jeanne-Louise-Philippine (dame
Mortemart de Boisse) (1733-1796)'){
        $tex_is_written_by_per = 2551020;
    }
    elseif($auteur_normalise == 'Mellet (de), Antoine-Joseph (né vers 1661, mort
en 1736)'){
        $tex_is_written_by_per = 2551021;
    }
    elseif($auteur_normalise == 'Mignard, Paul (1641-1691)'){
        $tex_is_written_by_per = 170449;
    }
    elseif($auteur_normalise == 'Musset de Cogners, Louis-Alexandre-Marie (1753-
1839)'){
        $tex_is_written_by_per = 231164;
    }
    elseif($auteur_normalise == 'Orry de Fulvy, Philibert-Louis (1736-1823)'){
        $tex_is_written_by_per = 2551022;
    }
```

```
    elseif($auteur_normalise == 'Pajon, Henri (mort en 1776)'){
        $tex_is_written_by_per = 180009;
    }
    elseif($auteur_normalise == 'Poisson, Raymond (né vers 1630, mort en 1690)'){
        $tex_is_written_by_per = 162505;
    }
    elseif($auteur_normalise == 'Pontevès-Gien (de), Henri (1738-1790)'){
        $tex_is_written_by_per = 199049;
    }
    elseif($auteur_normalise == 'Rousseau, Jean-Jacques (1712-1778)'){
        $tex_is_written_by_per = 199708;
    }
    elseif($auteur_normalise == 'Saint-Martin (de), Georges-François (1729-
1811)'){
        $tex_is_written_by_per = 2551023;
    }
    elseif($auteur_normalise == 'Saliès (de), Antoinette (1639-1730)'){
        $tex_is_written_by_per = 165174;
    }
    elseif($auteur_normalise == 'Tilly (de), Charles-Louis-Auguste-Casimir-Marie
(né en 1754)'){
        $tex_is_written_by_per = 2551024;
    }
    elseif($auteur_normalise == 'Vienne Plancy (de), Antoine'){
        $tex_is_written_by_per = 2551337;
    }

    //We adjust the author's gender here.

    if($auteur == '' && $activite == '' && $tex_is_written_by_per == ''){
        $tex_author_gender = 148; //Value matching "Indéterminé" in the "Genre"
taxonomy of the Drupal database.
    }

    //Create field tex_is_probably_written_by_per.

    if($id == 218){
        $tex_is_probably_written_by_per = 2550818;
    } //Charles François de Charbonnier.
    elseif($id == 354 || $id == 365){
        $tex_is_probably_written_by_per = 165599;
    } //Houdar de La Motte.
    elseif($id == 363){
        $tex_is_probably_written_by_per = 185549;
    } //Jean-Florent-Joseph de Neufville de Brunaubois-Montador.
    elseif($id == 563){
        $tex_is_probably_written_by_per = 2549980;
    } //Jean-Baptiste Poncy de Neuville.
    elseif($id == 1409){
        $tex_is_probably_written_by_per = 192071;
    } //Jean-Nicolas Jouin de Sauseuil.
    elseif($id == 1410){
        $tex_is_probably_written_by_per = 2550974;
    } //François Lorieux.
    elseif($id == 1646){
        $tex_is_probably_written_by_per = 2550819;
    } //Bernard de Lanevère.
    elseif($id == 1888){
        $tex_is_probably_written_by_per = 199885;
```

```
    } //Louis de Boussanelle.
    elseif($id == 1896 || $id == 2136 || $id == 2141){
        $tex_is_probably_written_by_per = 197100;
    } //Jean-François Guichard.
    elseif($id == 2781 || $id == 3087){
        $tex_is_probably_written_by_per = 2550960;
    } //Alexis-Antoine Valette des Hermeaux.
    elseif($id == 3114){
        $tex_is_probably_written_by_per = 2550961;
    } //Claude Louis-Denis Mulot de La Ménardière.
    elseif($id == 4153){
        $tex_is_probably_written_by_per = 1295653;
    } //Hardouin Le Fèvre de Fontenay.
    elseif($id == 4625 || $id == 4631 || $id == 4668){
        $tex_is_probably_written_by_per = 2550962;
    } //Jacques-Eustache de Louvencourt.
    elseif($id == 4677){
        $tex_is_probably_written_by_per = 231007;
    } //Pierre Fridolin Piet-Chambelle.
    elseif($id == 4913){
        $tex_is_probably_written_by_per = 201383;
    } //François-Félix Nogaret.
    elseif($id == 5180 || $id == 5332){
        $tex_is_probably_written_by_per = 210960;
    } //Jean-Étienne-Judith Forestier, dit Boinvilliers.
    elseif($id == 5903 || $id == 5946 || $id == 5956 || $id == 6021){
        $tex_is_probably_written_by_per = 226976;
    } //Jean-Antoine Lebrun-Tossa.
    elseif($id == 5907 || $id == 5994){
        $tex_is_probably_written_by_per = 208753;
    } //André-Joseph Grétry.
    elseif($id == 2870 || $id == 2871){
        $tex_is_probably_written_by_per = 185414;
    } //Jean-Baptiste Cabanis.

    //Create field tex_is_addressed_to_per.

    if($id == 552){
        $tex_is_addressed_to_per = 161220; //Anne Louise Bénédicte de Bourbon-
Condé Maine.
    }
    elseif($id == 5386){
        $tex_is_addressed_to_per = 225480; //Adrienne de La Fayette.
    }
    elseif($id == 3032){
        $tex_is_addressed_to_per = 185286; //Jean Le Rond D'Alembert.
    }

    //Create field tex_is_probably_addressed_to_per. (Remains empty: no text
concerned.)

    $tex_is_probably_addressed_to_per = '';

    //Create field tex_concerns_wor from reponse_normalisee.

    if($reponse_normalisee == 'Almanach des Muses'){
        $tex_concerns_wor = 1646884;
    }
```

```
    elseif($reponse_normalisee == 'Brid\'oison [personnage du Mariage de Figaro]'
|| $reponse_normalisee == 'Figaro' || $reponse_normalisee == 'La Folle Journée [Le
Mariage de Figaro]'){
        $tex_concerns_wor = 1740312;
    }
    elseif($id == 5062){
        $tex_concerns_wor = 1595566;
    }
    elseif($reponse_normalisee == 'Émile'){
        $tex_concerns_wor = 1646876;
    }
    elseif($reponse_normalisee == 'Gazette'){
        $tex_concerns_wor = 716710;
    }
    elseif($reponse_normalisee == 'Henriade'){
        $tex_concerns_wor = 1481061;
    }
    elseif($reponse_normalisee == 'Jérôme Pointu'){
        $tex_concerns_wor = 1759927;
    }
    elseif($reponse_normalisee == 'Malade imaginaire'){
        $tex_concerns_wor = 818881;
    }
    elseif($reponse_normalisee == 'Métromanie'){
        $tex_concerns_wor = 1525352;
    }
    elseif($reponse_normalisee == 'Peau d\'âne'){
        $tex_concerns_wor = 887504;
    }
    elseif($id == 2854){
        $tex_concerns_wor = 1656574;
    }
    elseif($reponse_normalisee == 'Tarare'){
        $tex_concerns_wor = 1785738;
    }
    elseif($reponse_normalisee == 'Zaïre'){
        $tex_concerns_wor = 1500260;
    }

    //Create field tex_probably_concerns_wor.

    if($id == 2060 || $id == 5005){
        $tex_probably_concerns_wor = 1595566;
    }

    //Create field tex_comes_from_pla from lieu_normalise_geonames.

    $id_geonames = explode('/', $lieu_normalise_geonames);
    $id_geonames = $id_geonames[3];

    if($id_geonames == 2660646){
        $tex_comes_from_pla = 1088175; //NID of "Genève" in the Drupal database.
    }
    elseif($id_geonames == 3032213){
        $tex_comes_from_pla = 1043216; //NID of "Blois" in the Drupal database.
    }
    elseif($id_geonames == 2988507){
        $tex_comes_from_pla = 1007953; //NID of "Paris" in the Drupal database.
    }
```

```
    elseif($id_geonames == 2968662){
        $tex_comes_from_pla = 992060; //NID of "Villemeux-sur-Eure" in the Drupal
database.
    }
    elseif($id_geonames == 3020810){
        $tex_comes_from_pla = 1034356; //NID of "Dreux" in the Drupal database.
    }
    elseif($id_geonames == 3021372){
        $tex_comes_from_pla = 1034781; //NID of "Dijon" in the Drupal database.
    }
    elseif($id_geonames == 3018794){
        $tex_comes_from_pla = 1032829; //NID of "Fécamp" in the Drupal database.
    }
    elseif($id_geonames == 2995469){
        $tex_comes_from_pla = 1013367; //NID of "Marseille" in the Drupal
database.
    }
    elseif($id_geonames == 2982652){
        $tex_comes_from_pla = 1003608; //NID of "Rouen" in the Drupal database.
    }
    elseif($id_geonames == 2969766){
        $tex_comes_from_pla = 992962; //NID of "Vernon" in the Drupal database.
    }
    elseif($id_geonames == 2969679){
        $tex_comes_from_pla = 992892; //NID of "Versailles" in the Drupal
database.
    }
    elseif($id_geonames == 3026286){
        $tex_comes_from_pla = 1038601; //NID of "Château-du-Loir" in the Drupal
database.
    }
    elseif($id_geonames == 3028542){
        $tex_comes_from_pla = 1040383; //NID of "Carpentras" in the Drupal
database.
    }
    elseif($id_geonames == 2985659){
        $tex_comes_from_pla = 1005860; //NID of "Pradelles" in the Drupal
database.
    }
    elseif($id_geonames == 3009526){
        $tex_comes_from_pla = 1025531; //NID of "La Fère" in the Drupal database.
    }
    elseif($id_geonames == 3036784){
        $tex_comes_from_pla = 1048606; //NID of "Arras" in the Drupal database.
    }
    elseif($id_geonames == 3016265){
        $tex_comes_from_pla = 1030897; //NID of "Gerberoy" in the Drupal database.
    }
    elseif($id_geonames == 3038354){
        $tex_comes_from_pla = 1049708; //NID of "Aix-en-Provence" in the Drupal
database.
    }
    elseif($id_geonames == 3036938){
        $tex_comes_from_pla = 1048703; //NID of "Arles" in the Drupal database.
    }
    elseif($id_geonames == 3014867){
        $tex_comes_from_pla = 1029950; //NID of "Granville" in the Drupal
database.
    }
```

```
    elseif($id_geonames == 2975088){
        $tex_comes_from_pla = 996989; //NID of "Senlis" in the Drupal database.
    }
    elseif($id_geonames == 2990363){
        $tex_comes_from_pla = 1009325; //NID of "Nîmes" in the Drupal database.
    }
    elseif($id_geonames == 2981041){
        $tex_comes_from_pla = 1002242; //NID of "Saint-Cloud" in the Drupal
database.
    }
    elseif($id_geonames == 2993024){
        $tex_comes_from_pla = 1011441; //NID of "Montargis" in the Drupal
database.
    }
    elseif($id_geonames == 2996944){
        $tex_comes_from_pla = 1014491; //NID of "Lyon" in the Drupal database.
    }
    elseif($id_geonames == 2975050){
        $tex_comes_from_pla = 996956; //NID of "Sens" in the Drupal database.
    }
    elseif($id_geonames == 2985821){
        $tex_comes_from_pla = 1005994; //NID of "Poulainville" in the Drupal
database.
    }
    elseif($id_geonames == 2994798){
        $tex_comes_from_pla = 1012833; //NID of "Meaux" in the Drupal database.
    }
    elseif($id_geonames == 2968545){
        $tex_comes_from_pla = 991946; //NID of "Villeneuve-lès-Avignon" in the
Drupal database.
    }
    elseif($id_geonames == 3038638){
        $tex_comes_from_pla = 1049880; //NID of "Agde" in the Drupal database.
    }
    elseif($id_geonames == 3021100){
        $tex_comes_from_pla = 1034559; //NID of "Donchery" in the Drupal database.
    }
    elseif($id_geonames == 3028337){
        $tex_comes_from_pla = 1040233; //NID of "Castelnau-le-Lez" in the Drupal
database.
    }
    elseif($id_geonames == 3024026){
        $tex_comes_from_pla = 1036834; //NID of "Conches-en-Ouche" in the Drupal
database.
    }
    elseif($id_geonames == 2986140){
        $tex_comes_from_pla = 1006234; //NID of "Pontoise" in the Drupal database.
    }
    elseif($id_geonames == 3035681){
        $tex_comes_from_pla = 1047724; //NID of "Avignon" in the Drupal database.
    }
    elseif($id_geonames == 3012315){
        $tex_comes_from_pla = 1028051; //NID of "Joinville" in the Drupal
database.
    }
    elseif($id_geonames == 2983489){
        $tex_comes_from_pla = 1004194; //NID of "Riom" in the Drupal database.
    }
    elseif($id_geonames == 3025602){
```

```
        $tex_comes_from_pla = 1038037; //NID of "Chemillé-en-Anjou" in the Drupal
database.
    }
    elseif($id_geonames == 3037656){
        $tex_comes_from_pla = 1049211; //NID of "Angers" in the Drupal database.
    }
    elseif($id_geonames == 3026467){
        $tex_comes_from_pla = 1038745; //NID of "Chartres" in the Drupal database.
    }
    elseif($id_geonames == 2991481){
        $tex_comes_from_pla = 1010171; //NID of "Moulins" in the Drupal database.
    }
    elseif($id_geonames == 3027487){
        $tex_comes_from_pla = 1039551; //NID of "Châlons-en-Champagne" in the
Drupal database.
    }
    elseif($id_geonames == 3023943){
        $tex_comes_from_pla = 1036760; //NID of "Condom" in the Drupal database.
    }
    elseif($id_geonames == 3019459){
        $tex_comes_from_pla = 1033313; //NID of "Étampes" in the Drupal database.
    }
    elseif($id_geonames == 2967879){
        $tex_comes_from_pla = 991333; //NID of "Vitré" in the Drupal database.
    }
    elseif($id_geonames == 2984701){
        $tex_comes_from_pla = 1005080; //NID of "Quimper" in the Drupal database.
    }
    elseif($id_geonames == 3031582){
        $tex_comes_from_pla = 1042714; //NID of "Bordeaux" in the Drupal database.
    }
    elseif($id_geonames == 2975758){
        $tex_comes_from_pla = 997495; //NID of "Saumur" in the Drupal database.
    }
    elseif($id_geonames == 2989317){
        $tex_comes_from_pla = 1008515; //NID of "Orléans" in the Drupal database.
    }
    elseif($id_geonames == 3026303){
        $tex_comes_from_pla = 1038613; //NID of "Châteaubriant" in the Drupal
database.
    }
    elseif($id_geonames == 2972468){
        $tex_comes_from_pla = 995014; //NID of "Tonneins" in the Drupal database.
    }
    elseif($id_geonames == 3033308){
        $tex_comes_from_pla = 1045926; //NID of "Bernay" in the Drupal database.
    }
    elseif($id_geonames == 2979309){
        $tex_comes_from_pla = 1000650; //NID of "Saint-Jean-de-Luz" in the Drupal
database.
    }
    elseif($id_geonames == 3012655){
        $tex_comes_from_pla = 1028297; //NID of "Issoudun" in the Drupal database.
    }
    elseif($id_geonames == 3012319){
        $tex_comes_from_pla = 1028053; //NID of "Joigny" in the Drupal database.
    }
    elseif($id_geonames == 2972360){
```

```
        $tex_comes_from_pla = 994929; //NID of "Touffreville-la-Cable" in the
Drupal database.
    }
    elseif($id_geonames == 2968748){
        $tex_comes_from_pla = 992140; //NID of "Villefranche-sur-Saône" in the
Drupal database.
    }
    elseif($id_geonames == 2983990){
        $tex_comes_from_pla = 1004546; //NID of "Rennes" in the Drupal database.
    }
    elseif($id_geonames == 3008686){
        $tex_comes_from_pla = 1024774; //NID of "L'Aigle" in the Drupal database.
    }
    elseif($id_geonames == 2972191){
        $tex_comes_from_pla = 994816; //NID of "Tours" in the Drupal database.
    }
    elseif($id_geonames == 3020119){
        $tex_comes_from_pla = 1033840; //NID of "Entrevaux" in the Drupal
database.
    }
    elseif($id_geonames == 2972328){
        $tex_comes_from_pla = 994908; //NID of "Toulon" in the Drupal database.
    }
    elseif($id_geonames == 2991772){
        $tex_comes_from_pla = 1010376; //NID of "Morlaix" in the Drupal database.
    }
    elseif($id_geonames == 2994160){
        $tex_comes_from_pla = 1012314; //NID of "Metz" in the Drupal database.
    }
    elseif($id_geonames == 2970838){
        $tex_comes_from_pla = 993812; //NID of "Valréas" in the Drupal database.
    }
    elseif($id_geonames == 3021670){
        $tex_comes_from_pla = 1034981; //NID of "Dax" in the Drupal database.
    }
    elseif($id_geonames == 2998228){
        $tex_comes_from_pla = 1015480; //NID of "Lingèvres" in the Drupal
database.
    }
    elseif($id_geonames == 2990474){
        $tex_comes_from_pla = 1009403; //NID of "Nevers" in the Drupal database.
    }
    elseif($id_geonames == 2972695){
        $tex_comes_from_pla = 995186; //NID of "Thouars" in the Drupal database.
    }
    elseif($id_geonames == 2973338){
        $tex_comes_from_pla = 995695; //NID of "Tartas" in the Drupal database.
    }
    elseif($id_geonames == 2972315){
        $tex_comes_from_pla = 994900; //NID of "Toulouse" in the Drupal database.
    }
    elseif($id_geonames == 2975143){
        $tex_comes_from_pla = 997035; //NID of "Semur-en-Auxois" in the Drupal
database.
    }
    elseif($id_geonames == 3031030){
        $tex_comes_from_pla = 1042306; //NID of "Bourg-Argental" in the Drupal
database.
    }
```

```
    elseif($id_geonames == 3038230){
        $tex_comes_from_pla = 1049623; //NID of "Alençon" in the Drupal database.
    }
    elseif($id_geonames == 3026285){
        $tex_comes_from_pla = 1038600; //NID of "Châteaudun" in the Drupal
database.
    }
    elseif($id_geonames == 3006787){
        $tex_comes_from_pla = 1023127; //NID of "La Rochelle" in the Drupal
database.
    }
    elseif($id_geonames == 3007680){
        $tex_comes_from_pla = 1023900; //NID of "Langres" in the Drupal database.
    }
    elseif($id_geonames == 2990999){
        $tex_comes_from_pla = 1009810; //NID of "Nancy" in the Drupal database.
    }
    elseif($id_geonames == 3033123){
        $tex_comes_from_pla = 1045775; //NID of "Besançon" in the Drupal database.
    }
    elseif($id_geonames == 2994651){
        $tex_comes_from_pla = 1012725; //NID of "Melun" in the Drupal database.
    }
    elseif($id_geonames == 745044){
        $tex_comes_from_pla = 1330550; //NID of "Constantinople" in the Drupal
database.
    }
    elseif($id_geonames == 3038634){
        $tex_comes_from_pla = 1049878; //NID of "Agen" in the Drupal database.
    }
    elseif($id_geonames == 3036067){
        $tex_comes_from_pla = 1048027; //NID of "Aups" in the Drupal database.
    }
    elseif($id_geonames == 3015926){
        $tex_comes_from_pla = 1030643; //NID of "Gisors" in the Drupal database.
    }
    elseif($id_geonames == 2987914){
        $tex_comes_from_pla = 1007546; //NID of "Perpignan" in the Drupal
database.
    }
    elseif($id_geonames == 2974640){
        $tex_comes_from_pla = 996642; //NID of "Sézanne" in the Drupal database.
    }
    elseif($id_geonames == 2988896){
        $tex_comes_from_pla = 1008220; //NID of "Oyé" in the Drupal database.
    }
    elseif($id_geonames == 2967972){
        $tex_comes_from_pla = 991409; //NID of "Vire" in the Drupal database.
    }
    elseif($id_geonames == 2996905){
        $tex_comes_from_pla = 1014471; //NID of "Machecoul" in the Drupal
database.
    }
    elseif($id_geonames == 3035843){
        $tex_comes_from_pla = 1047849; //NID of "Auxerre" in the Drupal database.
    }
    elseif($id_geonames == 3019626){
        $tex_comes_from_pla = 1033441; //NID of "Essay" in the Drupal database.
    }
```

```
    elseif($id_geonames == 3035828){
        $tex_comes_from_pla = 1047838; //NID of "Auxonne" in the Drupal database.
    }
    elseif($id_geonames == 3037051){
        $tex_comes_from_pla = 1048771; //NID of "Argentan" in the Drupal database.
    }
    elseif($id_geonames == 2998614){
        $tex_comes_from_pla = 1015797; //NID of "L'Hermenault" in the Drupal
database.
    }
    elseif($id_geonames == 2987928){
        $tex_comes_from_pla = 1007558; //NID of "Péronne" in the Drupal database.
    }
    elseif($id_geonames == 2984114){
        $tex_comes_from_pla = 1004641; //NID of "Reims" in the Drupal database.
    }
    elseif($id_geonames == 3022158){
        $tex_comes_from_pla = 1035393; //NID of "Cuers" in the Drupal database.
    }
    elseif($id_geonames == 3034006){
        $tex_comes_from_pla = 1046460; //NID of "Beauvais" in the Drupal database.
    }
    elseif($id_geonames == 2982303){
        $tex_comes_from_pla = 1003356; //NID of "Rozières-en-Beauce" in the Drupal
database.
    }
    elseif($id_geonames == 3021348){
        $tex_comes_from_pla = 1034766; //NID of "Dingé" in the Drupal database.
    }
    elseif($id_geonames == 3033140){
        $tex_comes_from_pla = 1045788; //NID of "Berville-en-Caux" in the Drupal
database.
    }
    elseif($id_geonames == 2993002){
        $tex_comes_from_pla = 1011422; //NID of "Montauban" in the Drupal
database.
    }
    elseif($id_geonames == 3005866){
        $tex_comes_from_pla = 1022352; //NID of "Laval" in the Drupal database.
    }
    elseif($id_geonames == 3022901){
        $tex_comes_from_pla = 1035951; //NID of "Courteuil" in the Drupal
database.
    }
    elseif($id_geonames == 3035088){
        $tex_comes_from_pla = 1047274; //NID of "Bapaume" in the Drupal database.
    }
    elseif($id_geonames == 2990263){
        $tex_comes_from_pla = 1009246; //NID of "Nogent-sur-Seine" in the Drupal
database.
    }
    elseif($id_geonames == 3034309){
        $tex_comes_from_pla = 1046701; //NID of "Beaufort-en-Vallée" in the Drupal
database.
    }
    elseif($id_geonames == 3020944){
        $tex_comes_from_pla = 1034452; //NID of "Doullens" in the Drupal database.
    }
    elseif($id_geonames == 3023763){
```

```
        $tex_comes_from_pla = 1036615; //NID of "Corbeil-Essonnes" in the Drupal
database.
    }
    elseif($id_geonames == 3020686){
        $tex_comes_from_pla = 1034274; //NID of "Dunkerque" in the Drupal
database.
    }
    elseif($id_geonames == 3022569){
        $tex_comes_from_pla = 1035691; //NID of "Crépy-en-Valois" in the Drupal
database.
    }
    elseif($id_geonames == 2976007){
        $tex_comes_from_pla = 997689; //NID of "Sarlat-la-Canéda" in the Drupal
database.
    }
    elseif($id_geonames == 3038169){
        $tex_comes_from_pla = 1049583; //NID of "Allanche" in the Drupal database.
    }
    elseif($id_geonames == 2985229){
        $tex_comes_from_pla = 1005519; //NID of "Provins" in the Drupal database.
    }
    elseif($id_geonames == 2986428){
        $tex_comes_from_pla = 1006468; //NID of "Pommard" in the Drupal database.
    }
    elseif($id_geonames == 3025466){
        $tex_comes_from_pla = 1037924; //NID of "Cherbourg" in the Drupal
database.
    }
    elseif($id_geonames == 2980589){
        $tex_comes_from_pla = 1001824; //NID of "Sainte-Foy-la-Grande" in the
Drupal database.
    }
    elseif($id_geonames == 2986151){
        $tex_comes_from_pla = 1006243; //NID of "Pont-l'Évêque" in the Drupal
database.
    }
    elseif($id_geonames == 2971033){
        $tex_comes_from_pla = 993947; //NID of "Valensole" in the Drupal database.
    }
    elseif($id_geonames == 2980111){
        $tex_comes_from_pla = 1001384; //NID of "Saint-Florentin" in the Drupal
database.
    }
    elseif($id_geonames == 3026464){
        $tex_comes_from_pla = 1038743; //NID of "Chartrettes" in the Drupal
database.
    }
    elseif($id_geonames == 2979013){
        $tex_comes_from_pla = 1000369; //NID of "Saint-Just-des-Marais" in the
Drupal database.
    }
    elseif($id_geonames == 2970777){
        $tex_comes_from_pla = 993765; //NID of "Vannes" in the Drupal database.
    }
    elseif($id_geonames == 3029984){
        $tex_comes_from_pla = 1041456; //NID of "Brissac-Quincé" in the Drupal
database.
    }
    elseif($id_geonames == 2990969){
```

```
        $tex_comes_from_pla = 1009789; //NID of "Nantes" in the Drupal database.
    }
    elseif($id_geonames == 3024816){
        $tex_comes_from_pla = 1037426; //NID of "Clairac" in the Drupal database.
    }
    elseif($id_geonames == 3005524){
        $tex_comes_from_pla = 1022041; //NID of "La Ville-du-Bois" in the Drupal
database.
    }
    elseif($id_geonames == 2972203){
        $tex_comes_from_pla = 994827; //NID of "Tournus" in the Drupal database.
    }
    elseif($id_geonames == 3019265){
        $tex_comes_from_pla = 1033168; //NID of "Évreux" in the Drupal database.
    }
    elseif($id_geonames == 2992166){
        $tex_comes_from_pla = 1010693; //NID of "Montpellier" in the Drupal
database.
    }
    elseif($id_geonames == 3004746){
        $tex_comes_from_pla = 1021326; //NID of "Le Charmel" in the Drupal
database.
    }
    elseif($id_geonames == 3026298){
        $tex_comes_from_pla = 1038610; //NID of "Château-Chinon" in the Drupal
database.
    }
    elseif($id_geonames == 3029241){
        $tex_comes_from_pla = 1040871; //NID of "Caen" in the Drupal database.
    }
    elseif($id_geonames == 2986094){
        $tex_comes_from_pla = 1006196; //NID of "Pont-sur-Seine" in the Drupal
database.
    }
    elseif($id_geonames == 3029974){
        $tex_comes_from_pla = 1041450; //NID of "Brive-la-Gaillarde" in the Drupal
database.
    }
    elseif($id_geonames == 3029162){
        $tex_comes_from_pla = 1040813; //NID of "Calais" in the Drupal database.
    }
    elseif($id_geonames == 3030300){
        $tex_comes_from_pla = 1041704; //NID of "Brest" in the Drupal database.
    }
    elseif($id_geonames == 2999619){
        $tex_comes_from_pla = 1016681; //NID of "Lessay" in the Drupal database.
    }
    elseif($id_geonames == 3031871){
        $tex_comes_from_pla = 1042931; //NID of "Bolbec" in the Drupal database.
    }
    elseif($id_geonames == 2998132){
        $tex_comes_from_pla = 1015406; //NID of "L'Isle-en-Dodon" in the Drupal
database.
    }
    elseif($id_geonames == 2998269){
        $tex_comes_from_pla = 1015511; //NID of "Limours" in the Drupal database.
    }
    elseif($id_geonames == 3007841){
        $tex_comes_from_pla = 1024046; //NID of "Landivy" in the Drupal database.
```

```
    }
    elseif($id_geonames == 2973783){
        $tex_comes_from_pla = 995990; //NID of "Strasbourg" in the Drupal
database.
    }
    elseif($id_geonames == 3013510){
        $tex_comes_from_pla = 1028939; //NID of "Henrichemont" in the Drupal
database.
    }
    elseif($id_geonames == 2974389){
        $tex_comes_from_pla = 996454; //NID of "Soissons" in the Drupal database.
    }
    elseif($id_geonames == 2972742){
        $tex_comes_from_pla = 995226; //NID of "Thonon-les-Bains" in the Drupal
database.
    }
    elseif($id_geonames == 2998286){
        $tex_comes_from_pla = 1015521; //NID of "Limoges" in the Drupal database.
    }
    elseif($id_geonames == 3037854){
        $tex_comes_from_pla = 1049356; //NID of "Amiens" in the Drupal database.
    }
    elseif($id_geonames == 3026141){
        $tex_comes_from_pla = 1038486; //NID of "Châtellerault" in the Drupal
database.
    }
    elseif($id_geonames == 2989920){
        $tex_comes_from_pla = 1008987; //NID of "Noyal-Muzillac" in the Drupal
database.
    }
    elseif($id_geonames == 3014383){
        $tex_comes_from_pla = 1029627; //NID of "Guéret" in the Drupal database.
    }
    elseif($id_geonames == 2998324){
        $tex_comes_from_pla = 1015555; //NID of "Lille" in the Drupal database.
    }
    elseif($id_geonames == 3032440){
        $tex_comes_from_pla = 1043371; //NID of "Blaison-Gohier" in the Drupal
database.
    }
    elseif($id_geonames == 3037598){
        $tex_comes_from_pla = 1049161; //NID of "Angoulême" in the Drupal
database.
    }
    elseif($id_geonames == 2998264){
        $tex_comes_from_pla = 1015508; //NID of "Limoux" in the Drupal database.
    }
    elseif($id_geonames == 2975340){
        $tex_comes_from_pla = 997167; //NID of "Sées" in the Drupal database.
    }
    elseif($id_geonames == 3021993){
        $tex_comes_from_pla = 1035251; //NID of "Cusset" in the Drupal database.
    }
    elseif($id_geonames == 3019329){
        $tex_comes_from_pla = 1033210; //NID of "Eu" in the Drupal database.
    }
    elseif($id_geonames == 2987003){
        $tex_comes_from_pla = 1006932; //NID of "Pithiviers" in the Drupal
database.
```

```
    }
    elseif($id_geonames == 2992760){
        $tex_comes_from_pla = 1011218; //NID of "Montdidier" in the Drupal
database.
    }
    elseif($id_geonames == 3029030){
        $tex_comes_from_pla = 1040726; //NID of "Cambrai" in the Drupal database.
    }
    elseif($id_geonames == 3023141){
        $tex_comes_from_pla = 1036171; //NID of "Courbevoie" in the Drupal
database.
    }
    elseif($id_geonames == 2971549){
        $tex_comes_from_pla = 994304; //NID of "Troyes" in the Drupal database.
    }
    elseif($id_geonames == 3029706){
        $tex_comes_from_pla = 1041247; //NID of "Bry-sur-Marne" in the Drupal
database.
    }
    elseif($id_geonames == 2998150){
        $tex_comes_from_pla = 1015422; //NID of "Lisieux" in the Drupal database.
    }
    elseif($id_geonames == 2983771){
        $tex_comes_from_pla = 1004405; //NID of "Rezay" in the Drupal database.
    }
    elseif($id_geonames == 2971929){
        $tex_comes_from_pla = 994605; //NID of "Tréguier" in the Drupal database.
    }
    elseif($id_geonames == 3036079){
        $tex_comes_from_pla = 1048032; //NID of "Auneau" in the Drupal database.
    }
    elseif($id_geonames == 3034554){
        $tex_comes_from_pla = 1046893; //NID of "Baugé-en-Anjou" in the Drupal
database.
    }
    elseif($id_geonames == 3026177){
        $tex_comes_from_pla = 1038510; //NID of "Châtel" in the Drupal database.
    }
    elseif($id_geonames == 3008225){
        $tex_comes_from_pla = 1024367; //NID of "Lamballe" in the Drupal database.
    }
    elseif($id_geonames == 3013521){
        $tex_comes_from_pla = 1028950; //NID of "Hennebont" in the Drupal
database.
    }
    elseif($id_geonames == 3035639){
        $tex_comes_from_pla = 1047687; //NID of "Avranches" in the Drupal
database.
    }
    elseif($id_geonames == 2971298){
        $tex_comes_from_pla = 994127; //NID of "Ussel" in the Drupal database.
    }
    elseif($id_geonames == 3030276){
        $tex_comes_from_pla = 1041687; //NID of "Breteuil" in the Drupal database.
    }
    elseif($id_geonames == 2873891){
        $tex_comes_from_pla = 1135208; //NID of "Mannheim" in the Drupal database.
    }
    elseif($id_geonames == 3022826){
```

```
        $tex_comes_from_pla = 1035891; //NID of "Coutances" in the Drupal
database.
    }
    elseif($id_geonames == 3007477){
        $tex_comes_from_pla = 1023730; //NID of "Laon" in the Drupal database.
    }
    elseif($id_geonames == 3024635){
        $tex_comes_from_pla = 1037286; //NID of "Clermont-Ferrand" in the Drupal
database.
    }
    elseif($id_geonames == 3023484){
        $tex_comes_from_pla = 1036410; //NID of "Cosnac" in the Drupal database.
    }
    elseif($id_geonames == 3019113){
        $tex_comes_from_pla = 1033056; //NID of "Falaise" in the Drupal database.
    }
    elseif($id_geonames == 2792413){
        $tex_comes_from_pla = 1078995; //NID of "Liège" in the Drupal database.
    }
    elseif($id_geonames == 3027977){
        $tex_comes_from_pla = 1039932; //NID of "Celon" in the Drupal database.
    }
    elseif($id_geonames == 3031702){
        $tex_comes_from_pla = 1042798; //NID of "Bonneval" in the Drupal database.
    }
    elseif($id_geonames == 3003796){
        $tex_comes_from_pla = 1020509; //NID of "Le Havre" in the Drupal database.
    }
    elseif($id_geonames == 2975349){
        $tex_comes_from_pla = 997172; //NID of "Sedan" in the Drupal database.
    }
    elseif($id_geonames == 3009481){
        $tex_comes_from_pla = 1025491; //NID of "La Ferté-sous-Jouarre" in the
Drupal database.
    }
    elseif($id_geonames == 2969799){
        $tex_comes_from_pla = 992992; //NID of "Verneuil-sur-Avre" in the Drupal
database.
    }
    elseif($id_geonames == 2972058){
        $tex_comes_from_pla = 994711; //NID of "Trans-en-Provence" in the Drupal
database.
    }
    elseif($id_geonames == 2990293){
        $tex_comes_from_pla = 1009273; //NID of "Noé" in the Drupal database.
    }
    elseif($id_geonames == 3028512){
        $tex_comes_from_pla = 1040368; //NID of "Carrouges" in the Drupal
database.
    }
    elseif($id_geonames == 2991704){
        $tex_comes_from_pla = 1010328; //NID of "Mortagne-au-Perche" in the Drupal
database.
    }
    elseif($id_geonames == 3020925){
        $tex_comes_from_pla = 1034438; //NID of "Dourdan" in the Drupal database.
    }
    elseif($id_geonames == 3012308){
        $tex_comes_from_pla = 1028046; //NID of "Jolivet" in the Drupal database.
```

```
    }
    elseif($id_geonames == 2972811){
        $tex_comes_from_pla = 995284; //NID of "Thionville" in the Drupal
database.
    }
    elseif($id_geonames == 2975774){
        $tex_comes_from_pla = 997508; //NID of "Saulzet" in the Drupal database.
    }
    elseif($id_geonames == 3176959){
        $tex_comes_from_pla = 1245336; //NID of "Florence" in the Drupal database.
    }
    elseif($id_geonames == 2990273){
        $tex_comes_from_pla = 1009255; //NID of "Nogent-le-Roi" in the Drupal
database.
    }
    elseif($id_geonames == 2968755){
        $tex_comes_from_pla = 992146; //NID of "Villefranche-de-Rouergue" in the
Drupal database.
    }
    elseif($id_geonames == 2986294){
        $tex_comes_from_pla = 1006361; //NID of "Pont-Audemer" in the Drupal
database.
    }
    elseif($id_geonames == 3031009){
        $tex_comes_from_pla = 1042285; //NID of "Bourg-en-Bresse" in the Drupal
database.
    }
    elseif($id_geonames == 2979658){
        $tex_comes_from_pla = 1000962; //NID of "Saint-Girod" in the Drupal
database.
    }
    elseif($id_geonames == 2980291){
        $tex_comes_from_pla = 1001550; //NID of "Saint-Étienne" in the Drupal
database.
    }
    elseif($id_geonames == 3027422){
        $tex_comes_from_pla = 1039500; //NID of "Chambéry" in the Drupal database.
    }
    elseif($id_geonames == 2972467){
        $tex_comes_from_pla = 995013; //NID of "Tonnerre" in the Drupal database.
    }
    elseif($id_geonames == 3038224){
        $tex_comes_from_pla = 1049619; //NID of "Alès" in the Drupal database.
    }
    elseif($id_geonames == 2983652){
        $tex_comes_from_pla = 1004322; //NID of "Richelieu" in the Drupal
database.
    }
    elseif($id_geonames == 3038789){
        $tex_comes_from_pla = 1049995; //NID of "Abbeville" in the Drupal
database.
    }
    elseif($id_geonames == 2986495){
        $tex_comes_from_pla = 1006517; //NID of "Poitiers" in the Drupal database.
    }
    elseif($id_geonames == 3024066){
        $tex_comes_from_pla = 1036858; //NID of "Compiègne" in the Drupal
database.
    }
```

```
    elseif($id_geonames == 3037021){
        $tex_comes_from_pla = 1048755; //NID of "Argenton-les-Vallées" in the
Drupal database.
    }
    elseif($id_geonames == 3023688){
        $tex_comes_from_pla = 1036558; //NID of "Cordey" in the Drupal database.
    }
    elseif($id_geonames == 2970797){
        $tex_comes_from_pla = 993776; //NID of "Vandœuvre-lès-Nancy" in the Drupal
database.
    }
    elseif($id_geonames == 2967108){
        $tex_comes_from_pla = 990672; //NID of "Blaye" in the Drupal database.
    }
    elseif($id_geonames == 3014172){
        $tex_comes_from_pla = 1029467; //NID of "Gumery" in the Drupal database.
    }
    elseif($id_geonames == 3026827){
        $tex_comes_from_pla = 1039007; //NID of "Chantilly" in the Drupal
database.
    }
    elseif($id_geonames == 2997457){
        $tex_comes_from_pla = 1014896; //NID of "Loudun" in the Drupal database.
    }
    elseif($id_geonames == 3007874){
        $tex_comes_from_pla = 1024069; //NID of "Landerneau" in the Drupal
database.
    }
    elseif($id_geonames == 3174659){
        $tex_comes_from_pla = 1243824; //NID of "Livorno" in the Drupal database.
    }
    elseif($id_geonames == 2971258){
        $tex_comes_from_pla = 994094; //NID of "Uzès" in the Drupal database.
    }
    elseif($id_geonames == 2986158){
        $tex_comes_from_pla = 1006248; //NID of "Pont-l'Abbé" in the Drupal
database.
    }
    elseif($id_geonames == 2969447){
        $tex_comes_from_pla = 992725; //NID of "Viala-du-Tarn" in the Drupal
database.
    }
    elseif($id_geonames == 3010154){
        $tex_comes_from_pla = 1026104; //NID of "La Châtre" in the Drupal
database.
    }
    elseif($id_geonames == 2981280){
        $tex_comes_from_pla = 1002463; //NID of "Saint-Brieuc" in the Drupal
database.
    }
    elseif($id_geonames == 3018074){
        $tex_comes_from_pla = 1032243; //NID of "Fontainebleau" in the Drupal
database.
    }
    elseif($id_geonames == 2992890){
        $tex_comes_from_pla = 1011329; //NID of "Montbrison" in the Drupal
database.
    }
    elseif($id_geonames == 2977295){
```

```
        $tex_comes_from_pla = 998773; //NID of "Saint-Quentin" in the Drupal
database.
    }
    elseif($id_geonames == 3027082){
        $tex_comes_from_pla = 1039215; //NID of "Champlitte" in the Drupal
database.
    }
    elseif($id_geonames == 3035409){
        $tex_comes_from_pla = 1047511; //NID of "Bagneux" in the Drupal database.
    }
    elseif($id_geonames == 2997996){
        $tex_comes_from_pla = 1015307; //NID of "Loches" in the Drupal database.
    }
    elseif($id_geonames == 3028162){
        $tex_comes_from_pla = 1040079; //NID of "Caussade" in the Drupal database.
    }
    elseif($id_geonames == 2800866){
        $tex_comes_from_pla = 1084328; //NID of "Bruxelles" in the Drupal
database.
    }
    elseif($id_geonames == 3014728){
        $tex_comes_from_pla = 1029868; //NID of "Grenoble" in the Drupal database.
    }
    elseif($id_geonames == 2971053){
        $tex_comes_from_pla = 993959; //NID of "Valence" in the Drupal database.
    }
    elseif($id_geonames == 2980816){
        $tex_comes_from_pla = 1002027; //NID of "Saint-Dizier" in the Drupal
database.
    }
    elseif($id_geonames == 3010404){
        $tex_comes_from_pla = 1026346; //NID of "La Chapelle-aux-Filtzméens" in
the Drupal database.
    }
    elseif($id_geonames == 3021193){
        $tex_comes_from_pla = 1034646; //NID of "Domfront" in the Drupal database.
    }
    elseif($id_geonames == 2987967){
        $tex_comes_from_pla = 1007584; //NID of "Périgueux" in the Drupal
database.
    }
    elseif($id_geonames == 3021264){
        $tex_comes_from_pla = 1034701; //NID of "Dol-de-Bretagne" in the Drupal
database.
    }
    elseif($id_geonames == 3027699){
        $tex_comes_from_pla = 1039721; //NID of "Chablis" in the Drupal database.
    }
    elseif($id_geonames == 3028263){
        $tex_comes_from_pla = 1040167; //NID of "Castres" in the Drupal database.
    }
    elseif($id_geonames == 2972350){
        $tex_comes_from_pla = 994920; //NID of "Toul" in the Drupal database.
    }
    elseif($id_geonames == 2973966){
        $tex_comes_from_pla = 996123; //NID of "Soupir" in the Drupal database.
    }
    elseif($id_geonames == 2985361){
```

```
        $tex_comes_from_pla = 1005628; //NID of "Preuilly-sur-Claise" in the
Drupal database.
    }
    elseif($id_geonames == 3030101){
        $tex_comes_from_pla = 1041550; //NID of "Brie-Comte-Robert" in the Drupal
database.
    }
    elseif($id_geonames == 2995814){
        $tex_comes_from_pla = 1013645; //NID of "Marfontaine" in the Drupal
database.
    }
    elseif($id_geonames == 2980671){
        $tex_comes_from_pla = 1001904; //NID of "Sainte-Colombe-sur-l'Hers" in the
Drupal database.
    }
    elseif($id_geonames == 3010025){
        $tex_comes_from_pla = 1025982; //NID of "La Ciotat" in the Drupal
database.
    }
    elseif($id_geonames == 2990355){
        $tex_comes_from_pla = 1009320; //NID of "Niort" in the Drupal database.
    }
    elseif($id_geonames == 2986150){
        $tex_comes_from_pla = 1006242; //NID of "Pont-l'Évêque" in the Drupal
database.
    }
    elseif($id_geonames == 3025892){
        $tex_comes_from_pla = 1038275; //NID of "Chaumont" in the Drupal database.
    }
    elseif($id_geonames == 3019858){
        $tex_comes_from_pla = 1033628; //NID of "Ervy-le-Châtel" in the Drupal
database.
    }
    elseif($id_geonames == 2979657){
        $tex_comes_from_pla = 1000961; //NID of "Saint-Girons" in the Drupal
database.
    }
    elseif($id_geonames == 3012148){
        $tex_comes_from_pla = 1027921; //NID of "Joyeuse" in the Drupal database.
    }
    elseif($id_geonames == 2975026){
        $tex_comes_from_pla = 996940; //NID of "Sépeaux" in the Drupal database.
    }
    elseif($id_geonames == 3031743){
        $tex_comes_from_pla = 1042834; //NID of "Bonnemain" in the Drupal
database.
    }
    elseif($id_geonames == 3037331){
        $tex_comes_from_pla = 1048952; //NID of "Aramon" in the Drupal database.
    }
    elseif($id_geonames == 2990064){
        $tex_comes_from_pla = 1009106; //NID of "Notre-Dame-de-Bondeville" in the
Drupal database.
    }
    elseif($id_geonames == 3029748){
        $tex_comes_from_pla = 1041280; //NID of "Brunoy" in the Drupal database.
    }
    elseif($id_geonames == 3002172){
```

```
        $tex_comes_from_pla = 1019036; //NID of "Les Andelys" in the Drupal
database.
    }
    elseif($id_geonames == 2975469){
        $tex_comes_from_pla = 997263; //NID of "Sceaux" in the Drupal database.
    }
    elseif($id_geonames == 3023466){
        $tex_comes_from_pla = 1036397; //NID of "Cossé-le-Vivien" in the Drupal
database.
    }
    elseif($id_geonames == 2982317){
        $tex_comes_from_pla = 1003368; //NID of "Rozay-en-Brie" in the Drupal
database.
    }
    elseif($id_geonames == 3034140){
        $tex_comes_from_pla = 1046569; //NID of "Beaumont-sur-Sarthe" in the
Drupal database.
    }
    elseif($id_geonames == 3016714){
        $tex_comes_from_pla = 1031201; //NID of "Ganges" in the Drupal database.
    }
    elseif($id_geonames == 2994935){
        $tex_comes_from_pla = 1012945; //NID of "Mayenne" in the Drupal database.
    }
    elseif($id_geonames == 2995375){
        $tex_comes_from_pla = 1013297; //NID of "Martincourt" in the Drupal
database.
    }
    elseif($id_geonames == 3026613){
        $tex_comes_from_pla = 1038850; //NID of "Charleville-Mézières" in the
Drupal database.
    }
    elseif($id_geonames == 2990919){
        $tex_comes_from_pla = 1009751; //NID of "Narbonne" in the Drupal database.
    }
    elseif($id_geonames == 2979781){
        $tex_comes_from_pla = 1001079; //NID of "Saint-Germain-et-Mons" in the
Drupal database.
    }
    elseif($id_geonames == 3014029){
        $tex_comes_from_pla = 1029359; //NID of "Ham" in the Drupal database.
    }
    elseif($id_geonames == 3005270){
        $tex_comes_from_pla = 1021813; //NID of "Le Blanc" in the Drupal database.
    }
    elseif($id_geonames == 3027484){
        $tex_comes_from_pla = 1039549; //NID of "Chalon-sur-Saône" in the Drupal
database.
    }
    elseif($id_geonames == 2970977){
        $tex_comes_from_pla = 993904; //NID of "Vallabrix" in the Drupal database.
    }
    elseif($id_geonames == 2980885){
        $tex_comes_from_pla = 1002094; //NID of "Saint-Denis-en-Val" in the Drupal
database.
    }
    elseif($id_geonames == 3013922){
        $tex_comes_from_pla = 1029268; //NID of "Harfleur" in the Drupal database.
    }
```

```
    elseif($id_geonames == 2983362){
        $tex_comes_from_pla = 1004110; //NID of "Roanne" in the Drupal database.
    }
    elseif($id_geonames == 2992292){
        $tex_comes_from_pla = 1010800; //NID of "Montluçon" in the Drupal
database.
    }
    elseif($id_geonames == 3017921){
        $tex_comes_from_pla = 1032105; //NID of "Fontenay-le-Comte" in the Drupal
database.
    }
    elseif($id_geonames == 2979363){
        $tex_comes_from_pla = 1000703; //NID of "Saint-Jean-d'Angély" in the
Drupal database.
    }
    elseif($id_geonames == 3009477){
        $tex_comes_from_pla = 1025487; //NID of "La Ferté-Vidame" in the Drupal
database.
    }
    elseif($id_geonames == 2977250){
        $tex_comes_from_pla = 998736; //NID of "Saint-Rambert-en-Bugey" in the
Drupal database.
    }
    elseif($id_geonames == 2984284){
        $tex_comes_from_pla = 1004772; //NID of "Rebais" in the Drupal database.
    }
    elseif($id_geonames == 2994085){
        $tex_comes_from_pla = 1012258; //NID of "Meymac" in the Drupal database.
    }
    elseif($id_geonames == 3034483){
        $tex_comes_from_pla = 1046845; //NID of "Bayeux" in the Drupal database.
    }
    elseif($id_geonames == 3012526){
        $tex_comes_from_pla = 1028201; //NID of "Jambville" in the Drupal
database.
    }
    elseif($id_geonames == 311046){
        $tex_comes_from_pla = 1309850; //NID of "İzmir" in the Drupal database.
    }
    elseif($id_geonames == 2972857){
        $tex_comes_from_pla = 995326; //NID of "Thiers" in the Drupal database.
    }
    elseif($id_geonames == 3015434){
        $tex_comes_from_pla = 1030247; //NID of "Goven" in the Drupal database.
    }
    elseif($id_geonames == 2981843){
        $tex_comes_from_pla = 1002969; //NID of "Saint-Amand-en-Puisaye" in the
Drupal database.
    }
    elseif($id_geonames == 2983276){
        $tex_comes_from_pla = 1004059; //NID of "Rochefort" in the Drupal
database.
    }
    elseif($id_geonames == 2982200){
        $tex_comes_from_pla = 1003267; //NID of "Rugles" in the Drupal database.
    }
    elseif($id_geonames == 3032621){
        $tex_comes_from_pla = 1045370; //NID of "Billy-Chevannes" in the Drupal
database.
```

```
    }
    elseif($id_geonames == 3038088){
        $tex_comes_from_pla = 1049518; //NID of "Allonne" in the Drupal database.
    }
    elseif($id_geonames == 3022164){
        $tex_comes_from_pla = 1035399; //NID of "Cucuron" in the Drupal database.
    }
    elseif($id_geonames == 2985118){
        $tex_comes_from_pla = 1005430; //NID of "Puiseaux" in the Drupal database.
    }
    elseif($id_geonames == 3017563){
        $tex_comes_from_pla = 1031844; //NID of "Fouquenies" in the Drupal
database.
    }
    elseif($id_geonames == 3037352){
        $tex_comes_from_pla = 1048959; //NID of "Apt" in the Drupal database.
    }
    elseif($id_geonames == 2991102){
        $tex_comes_from_pla = 1009893; //NID of "Mussidan" in the Drupal database.
    }
    elseif($id_geonames == 2993384){
        $tex_comes_from_pla = 1011736; //NID of "Monclar" in the Drupal database.
    }
    elseif($id_geonames == 3035883){
        $tex_comes_from_pla = 1047878; //NID of "Autun" in the Drupal database.
    }
    elseif($id_geonames == 3003603){
        $tex_comes_from_pla = 1020336; //NID of "Le Mans" in the Drupal database.
    }
    elseif($id_geonames == 3013365){
        $tex_comes_from_pla = 1028822; //NID of "Hesdin" in the Drupal database.
    }
    elseif($id_geonames == 2973013){
        $tex_comes_from_pla = 995459; //NID of "Thann" in the Drupal database.
    }
    elseif($id_geonames == 3017609){
        $tex_comes_from_pla = 1031881; //NID of "Fougères" in the Drupal database.
    }
    elseif($id_geonames == 3026842){
        $tex_comes_from_pla = 1039020; //NID of "Chantepie" in the Drupal
database.
    }
    elseif($id_geonames == 3005791){
        $tex_comes_from_pla = 1022286; //NID of "Laval-Roquecezière" in the Drupal
database.
    }
    elseif($id_geonames == 2999683){
        $tex_comes_from_pla = 1016743; //NID of "Les Sables-d'Olonne" in the
Drupal database.
    }
    elseif($id_geonames == 3025314){
        $tex_comes_from_pla = 1037805; //NID of "Chevilly-Larue" in the Drupal
database.
    }
    elseif($id_geonames == 2984192){
        $tex_comes_from_pla = 1004701; //NID of "Redon" in the Drupal database.
    }
    elseif($id_geonames == 3034475){
        $tex_comes_from_pla = 1046840; //NID of "Bayonne" in the Drupal database.
```

```
    }
    elseif($id_geonames == 2990550){
        $tex_comes_from_pla = 1009477; //NID of "Neuville-aux-Bois" in the Drupal
database.
    }
    elseif($id_geonames == 3023919){
        $tex_comes_from_pla = 1036745; //NID of "Confolens" in the Drupal
database.
    }
    elseif($id_geonames == 3030093){
        $tex_comes_from_pla = 1041544; //NID of "Brienne-le-Château" in the Drupal
database.
    }
    elseif($id_geonames == 3016953){
        $tex_comes_from_pla = 1031359; //NID of "Fronton" in the Drupal database.
    }
    elseif($id_geonames == 2996983){
        $tex_comes_from_pla = 1014522; //NID of "Luzancy" in the Drupal database.
    }
    elseif($id_geonames == 2995653){
        $tex_comes_from_pla = 1013517; //NID of "Marly-la-Ville" in the Drupal
database.
    }
    elseif($id_geonames == 3021411){
        $tex_comes_from_pla = 1034814; //NID of "Dieppe" in the Drupal database.
    }
    elseif($id_geonames == 3034800){
        $tex_comes_from_pla = 1047068; //NID of "Bar-sur-Seine" in the Drupal
database.
    }
    elseif($id_geonames == 3016824){
        $tex_comes_from_pla = 1031264; //NID of "Gaillac" in the Drupal database.
    }
    elseif($id_geonames == 2993372){
        $tex_comes_from_pla = 1011724; //NID of "Moncy" in the Drupal database.
    }
    elseif($id_geonames == 2968104){
        $tex_comes_from_pla = 991518; //NID of "Villy-Bocage" in the Drupal
database.
    }
    elseif($id_geonames == 2986060){
        $tex_comes_from_pla = 1006171; //NID of "Pornic" in the Drupal database.
    }
    elseif($id_geonames == 2983267){
        $tex_comes_from_pla = 1004054; //NID of "Rochefort-en-Terre" in the Drupal
database.
    }
    elseif($id_geonames == 2980400){
        $tex_comes_from_pla = 1001648; //NID of "Sainte-Menehould" in the Drupal
database.
    }
    elseif($id_geonames == 3036878){
        $tex_comes_from_pla = 1048667; //NID of "Arnac-Pompadour" in the Drupal
database.
    }
    elseif($id_geonames == 2978641){
        $tex_comes_from_pla = 1000025; //NID of "Saint-Maixme-Hauterive" in the
Drupal database.
    }
```

```
    elseif($id_geonames == 3014128){
        $tex_comes_from_pla = 1029433; //NID of "Gy" in the Drupal database.
    }
    elseif($id_geonames == 2972401){
        $tex_comes_from_pla = 994963; //NID of "Torteron" in the Drupal database.
    }
    elseif($id_geonames == 3014195){
        $tex_comes_from_pla = 1029486; //NID of "Guise" in the Drupal database.
    }
    elseif($id_geonames == 2990308){
        $tex_comes_from_pla = 1009286; //NID of "Noailles" in the Drupal database.
    }
    elseif($id_geonames == 2969836){
        $tex_comes_from_pla = 993022; //NID of "Verne" in the Drupal database.
    }
    elseif($id_geonames == 2985310){
        $tex_comes_from_pla = 1005584; //NID of "Princé" in the Drupal database.
    }
    elseif($id_geonames == 2984699){
        $tex_comes_from_pla = 1005079; //NID of "Quimperlé" in the Drupal
database.
    }
    elseif($id_geonames == 3014421){
        $tex_comes_from_pla = 1029658; //NID of "Guégon" in the Drupal database.
    }
    elseif($id_geonames == 2984718){
        $tex_comes_from_pla = 1005095; //NID of "Quillebeuf-sur-Seine" in the
Drupal database.
    }
    elseif($id_geonames == 3034141){
        $tex_comes_from_pla = 1046570; //NID of "Beaumont-sur-Oise" in the Drupal
database.
    }
    elseif($id_geonames == 3025762){
        $tex_comes_from_pla = 1038175; //NID of "Chavanat" in the Drupal database.
    }
    elseif($id_geonames == 3021062){
        $tex_comes_from_pla = 1034528; //NID of "Donzenac" in the Drupal database.
    }
    elseif($id_geonames == 2983568){
        $tex_comes_from_pla = 1004263; //NID of "Rignac" in the Drupal database.
    }
    elseif($id_geonames == 3025132){
        $tex_comes_from_pla = 1037659; //NID of "Chinon" in the Drupal database.
    }
    elseif($id_geonames == 2992667){
        $tex_comes_from_pla = 1011141; //NID of "Monterrein" in the Drupal
database.
    }
    elseif($id_geonames == 2988888){
        $tex_comes_from_pla = 1008215; //NID of "Oyonnax" in the Drupal database.
    }
    elseif($id_geonames == 3031327){
        $tex_comes_from_pla = 1042524; //NID of "Boudeville" in the Drupal
database.
    }
    elseif($id_geonames == 3021356){
        $tex_comes_from_pla = 1034769; //NID of "Dinan" in the Drupal database.
    }
```

```
    elseif($id_geonames == 2842632){
        $tex_comes_from_pla = 1118195; //NID of "Saarlouis" in the Drupal
database.
    }
    elseif($id_geonames == 2986092){
        $tex_comes_from_pla = 1006194; //NID of "Pont-sur-Yonne" in the Drupal
database.
    }
    elseif($id_geonames == 3008016){
        $tex_comes_from_pla = 1024187; //NID of "La Motte-Chalancon" in the Drupal
database.
    }
    elseif($lieu_normalise_geonames == 'Saint-Maurice, FR' && $auteur_normalise ==
'M. Metairie'){
        $tex_comes_from_pla = 999569; //NID of "Saint-Maurice-du-Désert" in the
Drupal database.
    }
    elseif($lieu_normalise_geonames == 'Quincey, FR'){
        $tex_comes_from_pla = 1005071; //NID of "Quincey (Aube)" in the Drupal
database.
    }

    //Exceptions.

    if($id == 2288){
        $tex_comes_from_pla = '';
    }
    if($id == 3843){
        $tex_comes_from_pla = '';
    }

    //Create field tex_probably_comes_from_pla.

    if(stripos($lieu, 'Chambon') !== false){
        $tex_probably_comes_from_pla = 1045862; //NID of "Bersac-sur-Rivalier" in
the Drupal database.
    }
    elseif($lieu_normalise_geonames == 'Saint-Domingue'){
        $tex_probably_comes_from_pla = 1378217; //NID of "Santo Domingo" in the
Drupal database.
    }
    elseif($lieu_normalise_geonames == 'Charenton'){
        $tex_probably_comes_from_pla = 1038870; //NID of "Charenton-le-Pont" in
the Drupal database.
    }
    elseif($lieu_normalise_geonames == 'Donnemarie'){
        $tex_probably_comes_from_pla = 1034543; //NID of "Donnemarie-Dontilly" in
the Drupal database.
    }
    elseif($lieu_normalise_geonames == 'Braine'){
        $tex_probably_comes_from_pla = 1041941; //NID of "Braine (Aisne)" in the
Drupal database.
    }
    elseif($id == 1875 || $id == 1883){
        $tex_probably_comes_from_pla = 1042798; //NID of "Bonneval" in the Drupal
database.
    }
    elseif($id == 2242){
```

```
        $tex_probably_comes_from_pla = 1022352; //NID of "Laval" in the Drupal
database.
    }
    elseif($id == 2468 || $id == 2474 || $id == 2535){
        $tex_probably_comes_from_pla = 1000634; //NID of "Saint-Jean-de-Savigny"
in the Drupal database.
    }
    elseif($id == 5587){
        $tex_probably_comes_from_pla = 2167124; //NID of "Arrondissement de
Rochefort" in the Drupal database.
    }
    elseif($id == 2638){
        $tex_probably_comes_from_pla = 995943; //NID of "Sully-sur-Loire" in the
Drupal database.
    }
    elseif($lieu_normalise_geonames == 'Saint-Valery, FR'){
        $tex_probably_comes_from_pla = 998219; //NID of "Saint-Valery-sur-Somme"
in the Drupal database.
    }
    elseif($lieu_normalise_geonames == 'Saint-Quentin, FR'){
        $tex_probably_comes_from_pla = 998773; //NID of "Saint-Quentin (Aisne)" in
the Drupal database.
    }
    elseif($lieu_normalise_geonames == 'Harcourt, FR'){
        $tex_probably_comes_from_pla = 995143; //NID of "Thury-Harcourt" in the
Drupal database.
    }
    elseif($lieu_normalise_geonames == 'Thorigné, FR'){
        $tex_probably_comes_from_pla = 995206; //NID of "Thorigné-sur-Dué" in the
Drupal database.
    }
    elseif($lieu_normalise_geonames == 'Tournon, FR'){
        $tex_probably_comes_from_pla = 994835; //NID of "Tournon-sur-Rhône" in the
Drupal database.
    }
    elseif($lieu_normalise_geonames == 'Vaucluse, FR'){
        $tex_probably_comes_from_pla = 1032236; //NID of "Fontaine-de-Vaucluse" in
the Drupal database.
    }
    elseif($lieu_normalise_geonames == 'Saint-Pourçain, FR'){
        $tex_probably_comes_from_pla = 998838; //NID of "Saint-Pourçain-sur-
Sioule" in the Drupal database.
    }
    elseif($id == 692){
        $tex_probably_comes_from_pla = 1007546; //NID of "Perpignan" in the Drupal
database.
    }
    elseif($id == 2638){
        $tex_probably_comes_from_pla = 995943; //NID of "Sully-sur-Loire" in the
Drupal database.
    }
    elseif($id == 2863){
        $tex_probably_comes_from_pla = 996670; //NID of "Sèvres" in the Drupal
database.
    }
    elseif($id == 3011){
        $tex_probably_comes_from_pla = 1029868; //NID of "Grenoble" in the Drupal
database.
    }
```

```
    elseif($id == 3095){
        $tex_probably_comes_from_pla = 1033056; //NID of "Falaise" in the Drupal
database.
    }
    elseif($id == 3102 || $id == 3103){
        $tex_probably_comes_from_pla = 1019533; //NID of "Le Plantis" in the
Drupal database.
    }
    elseif($id == 3504){
        $tex_probably_comes_from_pla = 1009466; //NID of "Neuville-les-Dames" in
the Drupal database.
    }
    elseif($id == 3965){
        $tex_probably_comes_from_pla = 1011242; //NID of "Montcuq" in the Drupal
database.
    }
    elseif($id == 4030 || $id == 4033){
        $tex_probably_comes_from_pla = 1000024; //NID of "Saint-Malo" in the
Drupal database.
    }
    elseif($id == 4062){
        $tex_probably_comes_from_pla = 1032812; //NID of "Feings" in the Drupal
database.
    }
    elseif($id == 4152){
        $tex_probably_comes_from_pla = 1007953; //NID of "Paris" in the Drupal
database.
    }
    elseif($id == 5046 || $id == 5047){
        $tex_probably_comes_from_pla = 1001079; //NID of "Saint-Germain-et-Mons"
in the Drupal database.
    }
    elseif($id == 5174){
        $tex_probably_comes_from_pla = 1002799; //NID of "Saint-Aquilin-de-Pacy"
in the Drupal database.
    }
    elseif($id == 5592){
        $tex_probably_comes_from_pla = 1009412; //NID of "Neuvy-le-Roi" in the
Drupal database.
    }
    elseif($id == 5769){
        $tex_probably_comes_from_pla = 1032236; //NID of "Fontaine-de-Vaucluse" in
the Drupal database.
    }
    elseif($id == 5955){
        $tex_probably_comes_from_pla = 1000059; //NID of "Saint-Luc (Eure)" in the
Drupal database.
    }
    elseif($id == 6185){
        $tex_probably_comes_from_pla = 1043356; //NID of "Blancafort (Cher)" in
the Drupal database.
    }
    elseif($id == 4704 || $id == 4721){
        $tex_probably_comes_from_pla = 1042524; //NID of "Boudeville (Seine-
Maritime)" in the Drupal database.
    }

    //Create field tex_is_part_of_fol.
```

```
    $tex_is_part_of_fol = '';

    if($id == 685){
        $tex_is_part_of_fol = 2547626; //Value matching the folder "Interventions
manuscrites" in the Drupal database.
    }
    if($id == 489){
        $tex_is_part_of_fol = 2547879; //Value matching the folder "Affaire de
Mademoiselle Malcrais de La Vigne" in the Drupal database.
    }
    if($reponse_normalisee == 'Vin'){
        $tex_is_part_of_fol = 2548651; //Value matching the folder "Vin" in the
Drupal database.
    }
    if($auteur_normalise == '+Δ+++' || $auteur_normalise == 'M. d. P...., o... a.
R... C... g... i...' || $auteur_normalise == 'M. L. B. de B. B. D. A. D. R. E. C.
A. G. S.' || $auteur_normalise == 'Mlle de Car..... [M. Le Riche]' ||
$auteur_normalise == 'Mlle de la Boissierre [anonyme]' || $auteur_normalise == 'Mr
Ztedroc' || $auteur_normalise == 'Pajon, Henri (mort en 1776)' ||
$auteur_normalise == 'Père Maire' || $auteur_normalise == 'Petit Boileau' ||
$auteur_normalise == 'Un amateur du Mercure de France' || $auteur_normalise == 'Un
anonyme' || $auteur_normalise == 'Une femme grosse' || $auteur_normalise ==
'Zorobabel de Roqueventoouse' || $id == 1070 || $id == 2980 || $id == 1903 || $id
== 2692 || $id == 3318 || $id == 4189 || $id == 4190){
        $tex_is_part_of_fol = 2547173; //Value matching the folder "Pratiques du
pseudonyme et de l'anonymat" in the Drupal database.
    }
    if($auteur_normalise == 'Marbet (de), Jeanne-Louise-Philippine (dame Mortemart
de Boisse) (1733-1796)'){
        $tex_is_part_of_fol = 2547118; //Value matching the folder "Paroles
féminines" in the Drupal database.
    }
    if($id == 685){
        $tex_is_part_of_fol = '"2547173,2547626"'; //Value matching the folders
"Pratiques du pseudonyme et de l'anonymat" and "Interventions manuscrites" in the
Drupal database.
    }
    if($id == 2596){
        $tex_is_part_of_fol = 1416123; //Value matching the folder "La fonction de
commis" in the Drupal database.
    }

    //Create field all_remark. We rewrite the data from the field note.

    if($id == 104){
        $all_remark = 'Énigme inspirée d’un sonnet énigmatique de Mr d’Aubicourt
paru dans le Mercure d’août 1706, p. 426-427.';
    }
    elseif($id == 119){
        $all_remark = 'Auteur présumé : Petit Jean Mercier de Blois.';
    }
    elseif($id == 120){
        $all_remark = 'Ce texte provient d’une énigme beaucoup plus longue de
"M<sup>r</sup> Rault" publiée dans le Mercure de décembre 1684, p. 315-316.';
    }
    elseif($id == 173){
        $all_remark = 'Reprise abrégée d’une énigme de « M<sup>r</sup> Gardien »
parue dans le Mercure de juin 1679, p. 323-324.';
    }
```

```
    elseif($id == 206){
        $all_remark = 'Cette énigme constitue la reprise, avec une fin différente,
d’un texte publié dans le Mercure de janvier 1680, vol. 1, p. 293-294, et composé
par « M<sup>r</sup> de Préfontaine ».';
    }
    elseif($id == 270){
        $all_remark = 'Le nom de l’auteur, "M<sup>r</sup> Robinet", est donné dans
la livraison de janvier 1678, p. 241.';
    }
    elseif($id == 283){
        $all_remark = 'Le signataire « J. B. D. » reprend manifestement une énigme
qui a déjà paru (avec d’infimes variantes) dans la livraison de juillet 1717, p.
184-185, signée par « M. le Comte de S. Gilin ».';
    }
    elseif($id == 296){
        $all_remark = 'Le signataire « De Bellefosse » reprend manifestement un
texte déjà paru sous le nom de la « Dame Solitaire » dans le Mercure de juin 1684,
p. 258. La réponse de l’énigme était alors l’argent.';
    }
    elseif($id == 466){
        $all_remark = 'Ce texte constitue la reprise partielle d’une énigme plus
longue de Mr d’Aubicourt parue dans le Mercure de septembre 1706, p. 357-359.';
    }
    elseif($id == 478){
        $all_remark = 'Ce texte constitue la reprise, avec des variantes, d’une
énigme publiée par Mr d’Aubicourt dans le Mercure d’août 1706, p. 426-427.';
    }
    elseif($id == 487){
        $all_remark = 'La provenance de l’auteur est déductible du texte et de sa
réponse « Meaux ».';
    }
    elseif($id == 596){
        $all_remark = 'Note des éditeurs en juillet 1737, p. 1579 : "On a fait une
faute considérable dans le Mercure de May, p. 921. en confondant par inadvertance
deux Logogryphes en un seul."';
    }
    elseif($id == 604){
        $all_remark = "Ce logogriphe est collé par erreur au texte précédent.";
    }
    elseif($id == 744 || $id == 745 || $id == 746 || $id == 983 || $id == 1248 ||
$id == 1249){
        $all_remark = 'Le numéro de page imprimé est erroné.';
    }
    elseif($id == 1030){
        $all_remark = 'L’auteur est un « Galant Maloüin » d’après une pulication
antérieure du même texte dans le Mercure de mars 1707, p. 374-375.';
    }
    elseif($id == 1100){
        $all_remark = 'L’auteur Bernard de Lanevère est identifiable par son nom,
par sa profession d’ancien mousquetaire du roi et par le fait qu’il réside à
Dax.';
    }
    elseif($id == 1131){
        $all_remark = 'Reprise, avec beaucoup de variantes, d’une énigme parue
anonymement dans le Mercure d’avril 1710, p. 345-346.';
    }
    elseif($id == 1239){
```

```
        $all_remark = 'Ce texte paraîtra de nouveau dans le Mercure d’avril 1769,
vol. 2, p. 61-62, avec une signature vraisemblablement apocryphe : « M. Mulot,
praticien. »';
    }
    elseif($id == 1492 || $id == 1493 || $id == 1494 || $id == 1495 || $id == 1496
|| $id == 1497 || $id == 1498){
        $all_remark = 'Auteur identifié d’après une lettre de Charles-Marie de La
Condamine à Johann II Bernoulli, Paris, 18 avril 1772, Bâle,
Universitätsbibliothek, L la 685, p. 625-628. Nous remercions Hugues Marchal
d’avoir porté ce document à notre connaissance.';
    }
    elseif($id == 1515 || $id == 1516 || $id == 1517 || $id == 1518 || $id == 1519
|| $id == 1520){
        $all_remark = 'Les énigmes et logogriphes de cette livraison sont
présentés comme provenant de l’auteur de l’énigme sur l’éternuement, parue dans le
Mercure de janvier 1772, vol. 2, p. 72-73, ce qui permet de les attribuer à La
Condamine. Voir aussi la lettre de Charles-Marie de La Condamine à Johann II
Bernoulli, Paris, 18 avril 1772, Bâle, Universitätsbibliothek, L la 685, p. 625-
628. Nous remercions Hugues Marchal d’avoir porté ce document à notre
connaissance.';
    }
    elseif($id == 1528 || $id == 1529 || $id == 1530 || $id == 1531 || $id == 1532
|| $id == 1533){
        $all_remark = 'La Condamine se présente comme l’auteur de ce texte dans
une lettre publiée dans le <em>Nouveau Journal helvétique</em>, Neuchâtel,
Imprimerie de la Société Typographique, février 1773, p. 104-113.';
    }
    elseif($id == 1571 || $id == 1572 || $id == 1573){
        $all_remark = 'Nous lisons dans le Mercure de décembre 1772, p. 213 : "M.
de la Garde d’Auberty, ancien trésorier de France de la généralité de Paris,
demeurant à Tulle, écrit que c’est à tort qu’on a mis son nom au bas de deux
énigmes & d’un logogryphe du second volume du ---Mercure--- d’Octobre dernier, &
qu’il n’en est point l’auteur."';
    }
    elseif($id == 1777 || $id == 1778){
        $all_remark = 'Auteur présumé : Adrien de Vincheguerre.';
    }
    elseif($id == 1734 || $id == 1735){
        $all_remark = 'Le nom de l’auteur est donné dans le Mercure de juin 1754,
vol. 1, p. 105.';
    }
    elseif($id == 1750 || $id == 1753){
        $all_remark = 'L’auteur est vraisemblablement « M. Martin ». Cet auteur
publie en effet une énigme signée « <em>M. M.*****. de Paris</em> » en juillet
1754, p. 93-94, qui sera republiée sous le nom de « <em>M. Martin de Savigny</em>
» en septembre 1771, p. 80.';
    }
    elseif($id == 1823){
        $all_remark = 'Auteur présumé : Abbé Renaudor.';
    }
    elseif($id == 1842){
        $all_remark = 'Le véritable auteur de ce logogriphe est démasqué par un
lecteur du Mercure dans la livraison d’août 1756, p. 79-80.';
    }
    elseif($id == 1843){
        $all_remark = 'Ce texte constitue la reprise, avec passablement de petites
variantes, d’une énigme de « M<sup>r</sup> Adam » parue dans le Mercure de mars
1688, p. 339-342.';
    }
```

```
    elseif($id == 1864){
        $all_remark = 'Le signataire n’est certainement pas l’auteur de l’énigme,
celle-ci ayant déjà paru anonymement dans les Mercure de décembre 1677, vol. 10,
p. 244-246, et de septembre 1712, p. 205-208.';
    }
    elseif($id == 1889){
        $all_remark = 'L’auteur présumé est M. Cassassolles qui publie un texte
presque identique dans le Mercure de mars 1762, p. 63.';
    }
    elseif($id == 1903){
        $all_remark = 'Nous lisons dans le Mercure de mai 1758, p. 72 : "La
personne qui nous a envoyé l’Enigme du premier volume d’Avril, l’a mise sous le
nom de Mademoiselle de Boissierre, de Moulins, qui nous a chargé de publier son
désaveu. Nous prions très-instamment ceux qui nous feront l’honneur de nous
adresser leurs productions de ne pas commettre de semblables infidélités ; car
nous ne pourrions nous dispenser de démasquer aux yeux du Public ces faussaires,
si nous parvenions à les reconnoître."';
    }
    elseif($id == 1941){
        $all_remark = 'Ce texte est attribuable à La Condamine par sa
republication dans le Mercure de janvier 1772, vol. 2, p. 75-76.';
    }
    elseif($id == 1992){
        $all_remark = 'Le lieu de provenance est confirmé par une des anagrammes
(Turnus) du mot à deviner.';
    }
    elseif($id == 2001 || $id == 2016){
        $all_remark = 'L’auteur a été identifié d’après des notices biographiques
anciennes.';
    }
    elseif($id == 2009){
        $all_remark = 'L’attribution erronée du texte à "M. Duberoi" est corrigée
en juillet 1768, vol. 2, p. 69. À l’occasion de cette correction tardive, on
apprend que l’auteur vient de Paris.';
    }
    elseif($id == 2033){
        $all_remark = 'L’auteur est « Rabelleau » d’après une publication
ultérieure du même texte dans le Mercure de mai 1761, p. 49-50.';
    }
    elseif($id == 2036){
        $all_remark = 'Redonné sous une forme sensiblement différente, ce texte
sera attribué à « M. D..... de Paris » dans le Mercure de juillet 1768, t. 2, p.
69.';
    }
    elseif($note == 'Le numéro de page imprimé est erroné.'){
        $all_remark = 'Le numéro de page imprimé est erroné.';
    }
    elseif($id == 2060){
        $all_remark = 'À l’égard de ce logogriphe sur l’<em>Encyclopédie</em>, la
signature « M. D…… » apparaît comme un clin d’oeil à Denis Diderot.';
    }
    elseif($id == 2069){
        $all_remark = 'Ce texte sera republié sous le nom de « M. M. Avocat au
Parlement » dans le Mercure de septembre 1764, p. 101-102.';
    }
    elseif($id == 2112){
        $all_remark = 'Ce texte sera republié sous le nom de « M. D. L. P. » dans
le Mercure de février 1774, p. 46-47. Toutefois, il constitue la reprise abrégée
d’un texte paru dans le Mercure de mars 1679, p. 353-354.';
```

```
    }
    elseif($id == 2125){
        $all_remark = 'Ce texte constitue la reprise, avec des variantes, d’une
énigme de Mr d’Aubicourt parue dans le Mercure de mars 1704, p. 357-358.';
    }
    elseif($id == 2192 || $id == 4500){
        $all_remark = 'Cette célèbre énigme est attribuée à François Blanchet dans
le Mercure d’avril 1763, p. 69. Toutefois, une énigme très semblable est signée
par « Mademoiselle D. B. » dans le Mercure de mars 1749, p. 118.';
    }
    elseif($id == 2232){
        $all_remark = 'Le signataire « J. D. ou l’Inconnu de Dampierre »
s’approprie une énigme du père Maire, professeur de rhétorique à Lyon, parue dans
le Mercure de mai 1722, vol. 2, p. 130-131.';
    }
    elseif($id == 2310){
        $all_remark = 'Ce texte constitue la reprise, avec des variantes, d’une
énigme de juin 1678, p. 331-332, composée par « une Personne de la premiere
qualité ».';
    }
    elseif($id == 2359){
        $all_remark = 'Ce texte constitue la reprise, avec passablement de
variantes, d’une énigme de Mr d’Aubicourt parue dans le Mercure d’avril 1706, p.
376-377.';
    }
    elseif($id == 2381){
        $all_remark = 'L’auteur est « J. B. D. » de Versailles, d’après une
publication antérieure du même texte dans le Mercure de mai 1729, p. 944.';
    }
    elseif($id == 2386){
        $all_remark = 'L’auteure est la « Bergere Fleurette », d’après une
publication antérieure du même texte dans le Mercure d’avril 1689, p. 316-317.';
    }
    elseif($id == 2395){
        $all_remark = 'Cette énigme paraît plusieurs fois anonymement. Elle est
signée « Flote, d’Angers » dans le Mercure de juin 1748, vol. 2, p. 134.';
    }
    elseif($id == 2400){
        $all_remark = 'L’auteur est « L’Abbé Pagés », d’après une publication
antérieure du même texte dans le Mercure de novembre 1739, p. 2627-2628.';
    }
    elseif($id == 2417){
        $all_remark = "Cette poésie qui forme un losange repose sur un procédé de
versification proche des vers rhopaliques. Ce procédé est décrit dans une note qui
accompagne le texte : « La singulière composition de cette Enigme a seule engagé
l'Auteur à l'envoyer au Mercure, 1<sup>o</sup>. parce que des vingt-six vers qui
composent cet ouvrage, les treize derniers répondent parfaitement aux treize
premiers par le nombre des pieds, 2<sup>o</sup>. par le nombre des mêmes rimes
distribuées également, depuis le milieu de l'ouvrage jusqu'à la fin, comme du
milieu au commencement, 3<sup>o</sup>. parce que la structure de tous les vers &
leur arrangement forment ensemble un lozange parfait. » (p. 79).";
    }
    elseif($id == 2425){
        $all_remark = 'L’auteur est « M.... » de Rouen, d’après une publication
antérieure du même texte dans le Mercure de janvier 1745, p. 115-116.';
    }
    elseif($id == 2455){
        $all_remark = 'À propos de l’auteur, on lit dans le Mercure d’octobre
1766, vol. 1, p. 64, que « M. B.... » est le « véritable auteur de l’énigme de la
```

```
carte, dans le second volume de Juillet, à l’occasion de la fête de Saint Augustin
».';
    }
    elseif($id == 2472){
        $all_remark = 'L’auteure est « l’aimable Caliste », d’après une
publication antérieure du même texte dans le Mercure d’avril 1686, p. 348.';
    }
    elseif($id == 2488){
        $all_remark = 'L’identification de l’auteur est permise par une
publication antérieure du même texte dans le Mercure de janvier 1728, p. 106.';
    }
    elseif($id == 2489){
        $all_remark = 'L’auteur est « Blezin » d’Arles, d’après une publication
antérieure du même texte dans le Mercure de novembre 1733, p. 2409-2410.';
    }
    elseif($id == 2492){
        $all_remark = 'L’auteur est « M<sup>r</sup> David », d’après une
publication antérieure du même texte dans le Mercure d’octobre 1694, p. 305-306.';
    }
    elseif($id == 2534){
        $all_remark = '« M. T… » est « M. T. D. M. », d’après un texte publié dans
le Mercure de novembre 1767, p. 61-62.';
    }
    elseif($id == 2535){
        $all_remark = "Cette énigme sur le triangle forme un calligramme.";
    }
    elseif($id == 2596){
        $all_remark = 'La correction qui accompagne la signature du texte n’est
pas relative au Mercure d’avril 1768, mais à la livraison d’avril 1760. Les
responsables du Mercure ont dû se tromper en lisant la date sur la lettre qui
l’accompagnait. Par conséquent, la présente énigme doit dater de l’année 1760 et
être restée huit ans dans les papiers du Mercure. Elle est d’ailleurs publiée sous
une forme sensiblement différente en octobre 1760, vol. 1, p. 47.';
    }
    elseif($id == 2643){
        $all_remark = 'Dans la signature, le lieu « Pluviers » compte parmi les
noms donnés à Pithiviers, où se trouve le château de Bardy.';
    }
    elseif($id == 2665){
        $all_remark = 'À l’égard du lieu de provenance, l’identification de Châtel
dans l’actuelle Haute-Savoie semble confirmée par la réponse du logogriphe qui est
« Alpes ».';
    }
    elseif($id == 2681){
        $all_remark = 'Ce texte ayant déjà paru de manière anonyme dans le Mercure
de janvier 1747, p. 156, la signature « M. Mulot » constitue probablement une
imposture. Le même M. Mulot s’approprie également le logogriphe qui suit
directement celui-ci dans le volume et qui a paru en 1730 sous une autre
signature.';
    }
    elseif($id == 2682){
        $all_remark = '« M. Mulot » s’approprie un logogriphe paru dans le Mercure
de décembre 1730, vol. 1, p. 2653, signé par « D’Orvilliers, de Vernon », c’est-à-
dire Charles Le Tellier d’Orvilliers (1692-1758).';
    }
    elseif($id == 2691){
        $all_remark = 'Le signataire « M. J. V. Tallard » s’approprie une énigme
de J. F. Guichard parue dans le Mercure de décembre 1750, vol. 2, p. 102-103.';
    }
```

```
    elseif($id == 2692){
        $all_remark = 'Le signataire « M. J. V. Tallard » s’approprie une énigme
de J. F. Guichard parue dans le Mercure de novembre 1750, p. 123.';
    }
    elseif($id == 2710){
        $all_remark = 'Cette énigme a paru sans nom d’auteur dans le Mercure de
mars 1737, p. 508. Il est peu probable que le signataire « M. Desnoyers »
(vraisemblablement Pierre Desnoyers) en soit l’auteur.';
    }
    elseif($id == 2760){
        $all_remark = '« M. T... » s’approprie ici une énigme qui était signée par
« M<sup>lle</sup> de Bellefont, de Vernon » dans le Mercure de juin 1729, vol. 1,
p. 1155-1156. Voir aussi l’énigme prétendument écrite par « M. T. » dans le
Mercure de janvier 1770, p. 80.';
    }
    elseif($id == 2766){
        $all_remark = '« M. T. » s’approprie ici une énigme qui était signée par «
M. V. J. A. L. » dans le Mercure d’avril 1732, p. 737-738. Voir aussi l’énigme
prétendument écrite par « M. T... » dans le Mercure de décembre 1769, vol. 1, p.
69.';
    }
    elseif($id == 2870 || $id == 2871){
        $all_remark = 'Le lieu de provenance « Salaignac » désigne le manoir de
Salagnac à Cosnac, fief de la famille Cabanis depuis le mariage de Jean-Baptiste
(1725-1786) avec Marie-Hélène d’Escarolle de Souleyrac. Voir André Role, Luc
Boulet, <em>Georges Cabanis. Le médecin de Brumaire</em>, Paris, Fernand Lanore,
1994, p. 13. Il n’est pas exclu que l’auteur du texte soit le jeune Pierre-Jean-
Georges Cabanis (1757-1808), fils de Jean-Baptiste et de Marie-Hélène, qui
développe très tôt un goût pour la poésie.';
    }
    elseif($id == 2887){
        $all_remark = 'Cette célèbre énigme de Houdar de La Motte paraît dans le
Mercure de janvier 1728, p. 103.';
    }
    elseif($id == 2919 || $id == 2920 || $id == 2921){
        $all_remark = 'L’identification du lieu est permise par la signature d’un
texte ultérieur du même auteur, dans le Mercure de juin 1771, p. 81-82, qui
précise « Bonneval en Beauce ».';
    }
    elseif($id == 2945){
        $all_remark = 'L’auteur reprend manifestement, en introduisant des
variantes, un texte de « Madame Rambey » paru dans le Mercure de novembre 1678, p.
366-357.';
    }
    elseif($id == 3027 || $id == 3028){
        $all_remark = '« M. Danzelle » est sans doute la même personne que « M.
Danzel » qui publie deux énigmes dans le Mercure d’octobre 1772, vol. 1, p. 69.';
    }
    elseif($id == 3059){
        $all_remark = 'Dans la signature, « D... de la V... » désigne
vraisemblablement le « Dieu de la Vigne ».';
    }
    elseif($id == 3113){
        $all_remark = 'Le même « chapelain de Dourdan » enverra des énigmes de
Senlis au mois de février de la même année (p. 48 et 49). « Dourdan » désigne donc
vraisemblablement le lieu d’origine de l’auteur, plutôt que son lieu de
résidence.';
    }
    elseif($id == 3134){
```

```
        $all_remark = 'Le même auteur a envoyé des énigmes de Senlis deux ans plus
tôt, en février 1772, p. 48 et 49. « Dourdan » désigne donc peut-être le lieu
d'origine de l'auteur, plutôt que son lieu de résidence.';
    }
    elseif($id == 3142){
        $all_remark = 'Lieu indiqué dans la signature, Huviller est le nom de
Jolivet avant la Révolution.';
    }
    elseif($id == 3166){
        $all_remark = 'Le signataire « M. L. Pons » s'approprie ici un texte paru
plusieurs fois dans le Mercure, en janvier 1712, p. 316-317, et en septembre 1728,
p. 2009 (sous une forme différente), puis en avril 1761, vol. 1, p. 62, et en
septembre 1764, p. 101-102. Voir aussi l'énigme qui suit directement celle-ci dans
la présente livraison.';
    }
    elseif($id == 3167){
        $all_remark = 'Le signataire « M. L. Pons » s'approprie ici un texte paru
sous le nom de « L'Inconnu de la ruë Saint Jean de Beauvais » dans le Mercure
d'avril 1703, p. 377. Voir aussi l'énigme qui précède directement celle-ci dans la
présente livraison.';
    }
    elseif($id == 3179){
        $all_remark = 'À propos de l'auteur de cette énigme, nous lisons dans le
Mercure d'octobre 1774, vol. 2, p. 213 : « Nous nous sommes engagés d'annoncer que
l'énigme <em>le Portrait</em>, insérée dans le Mercure de Septembre 1774, n'est
point de Mlle Parent de Melun, & que c'est à tort que cette énigme a été envoyée &
mise sous son nom. »';
    }
    elseif($id == 3203){
        $all_remark = 'L'auteure est présentée comme la « Collactane [soeur de
lait] de Mgr le Comte d'A... », probablement le comte d'Artois.';
    }
    elseif($id == 3266){
        $all_remark = 'Le signataire s'approprie manifestement un texte qui a déjà
paru anonymement dans la livraison d'octobre 1723, p. 746. Voir aussi l'énigme qui
paraît sous le même nom en avril 1776, vol. 1, p. 48.';
    }
    elseif($id == 3335){
        $all_remark = 'Le signataire s'approprie manifestement un texte qui a déjà
paru anonymement, sous une forme un peu différente, dans la livraison d'octobre
1723, p. 747. Voir aussi l'énigme qui paraît sous le même nom d'auteur en juillet
1775, vol. 1, p. 54.';
    }
    elseif($id == 3318){
        $all_remark = 'Le signataire « M. Dracolff » s'approprie ou republie un
texte déjà paru dans le Mercure de mars 1740, p. 514-515, sous le nom de «
Flocard, à Paris ». Le pseudonyme Dracolff correspond au nom Flocard lu de droite
à gauche.';
    }
    elseif($id == 3399){
        $all_remark = 'Ce texte constitue la reprise, avec peu de variantes, d'une
énigme publiée anonymement dans le Mercure de septembre 1690, p. 275-278. Le
signataire du texte n'est donc pas son auteur.';
    }
    elseif($id == 3405){
        $all_remark = 'Le signataire s'approprie ici un texte paru anonymement
dans les Mercure de janvier 1706, p. 431-432, et d'octobre 1731, p. 2364-2365,
tout en introduisant de petites variantes.';
    }
```

```
    elseif($id == 3418 || $id == 3437 || $id == 3449 || $id == 3455 || $id == 3456
|| $id == 3462){
        $all_remark = 'Abrégés dans la signature, l’activité (curé) et le lieu
(Quincey) sont déduits d’après des publications antérieures du même auteur, par
exemple son énigme parue dans le Mercure de décembre 1775, p. 55-58.';
    }
    elseif($id == 3421){
        $all_remark = 'L’auteur s’inspire très vraisemblablement d’une énigme déjà
parue. On la trouve déjà dans la <em>Suite de la clef ou journal historique sur
les matières du tems. Contenant quelques Nouvelles de Littérature, & autres
Remarques curieuses. Par le Sieur C. J.</em>, Paris, Etienne Ganeau, t. 35, avril
1734, p. 263.';
    }
    elseif($id == 3422){
        $all_remark = 'Ce texte a déjà paru dans la <em>Suite de la clef ou
journal historique sur les matières du tems. Contenant quelques Nouvelles de
Littérature, & autres Remarques curieuses. Par le Sieur C. J.</em>, Paris, Etienne
Ganeau, t. 35, juin 1734, p. 407. L’auteur anonyme provenait alors de Tours.';
    }
    elseif($id == 3479){
        $all_remark = 'L’auteur s’inspire très vraisemblablement d’un logogriphe
paru dans le Mercure de juillet 1775, vol. 1, p. 56, sous le nom de « M. Vincent,
Curé de Quincey ».';
    }
    elseif($id == 3485){
        $all_remark = 'L’auteur s’inspire très vraisemblablement d’un logogriphe
paru dans le Mercure de septembre 1775, p. 65, sous le nom de « M. Vincent, Curé
de Quincey ».';
    }
    elseif($id == 3524){
        $all_remark = '« M. Labrouche fils » étant associé à la ville de Dax dans
les autres textes qu’il donne au Mercure, il est possible qu’une erreur se soit
glissée dans la signature de ce texte-ci.';
    }
    elseif($id == 3544){
        $all_remark = 'La numérotation des pages est erronée (inversion des
chiffres 2 et 4).';
    }
    elseif($id == 3562){
        $all_remark = 'L’identification de l’auteur est permise par d’autres
signatures de poésies dans le Mercure. Voir aussi Geoffrey Turnovsky, « Les
Lecteurs du Mercure galant. Trois aperçus », <em>Dix-septième siècle</em>,
n<sup>o</sup> 270, 2016, p. 65-80.';
    }
    elseif($id == 3575){
        $all_remark = 'L’<em>Extraordinaire du ---Mercure--- galant</em> d’octobre
1678, p. 178, précise que cette auteure vient d’Espagne.';
    }
    elseif($id == 3587){
        $all_remark = 'À propos de l’auteure, le Mercure précise « Dame de Paris
demeurant à Mets ». Cependant, dans l’<em>Extraordinaire du ---Mercure---
galant</em> d’octobre 1679, p. 274, nous lisons : « Le Bois de Sainte Lucie estoit
donc le Mot de cette Enigme, faite par Madame de Taillandier de Nancy, & non de
Mets, comme je vous l’avois d’abord marqué. » S’y trouve également une explication
détaillée de l’énigme.';
    }
    elseif($id == 3619){
```

```
        $all_remark = 'L’activité « docteur en médecine » est déductible des
lettres « D. M. » qui accompagnent le nom de l’auteur et d’une autre mention de
cet auteur dans le Mercure de novembre 1680, p. 341.';
    }
    elseif($id == 3651){
        $all_remark = 'Le nom de l’auteur est dévoilé dans le Mercure de septembre
1682, 2<sup>nde</sup> partie, p. 354-356 : « Le mot de la premiere [énigme] estoit
<em>le Portrait</em>, M<sup>r</sup> Mignard, Peintre fameux, & qui est du nombre
de ceux qui composent l’Académie Royale de Peinture & Sculpture, en est l’Autheur.
Il est Fils de feu M<sup>r</sup> Mignard, dit d’Avignon, que le Roy fit venir en
1659. pour faire son Portrait, & celuy de la Reyne, & qui mourut en 1669. Il
estoit alors Recteur de l’Académie de Peinture, & venoit d’achever le bel
Apartement des Tuilleries, qui fait aujourd’huy l’admiration de tous les Curieux.
Le Fils d’un Homme si habile en son Art, & qui estoit universel dans sa
Profession, ayant herité de la maniere de faire des Portraits, qui avoit mis feu
son Pere dans une si haute réputation, le bruit s’en répandit dans la Cour de
Baviere. Il y fut mandé en 1673. par Madame l’Electrice, pour peindre toute sa
Famille. Il y réüssit au gré de cette Princesse, beaucoup mieux que plusieurs
fameux Peintres d’Italie qu’elle avoit fait venir pour le mesme sujet. Elle
reconnut son travail d’une maniere digne d’elle. Il se peignit luy-mesme par son
ordre, & cette Princesse luy ordonna de mettre autour de son Portrait, <em>Adelaïs
Apelles unus Mignardus</em>. Elle fit en suite placer ce Portrait dans l’une des
Galeries du riche Palais de Munich ; ce qui obligea beaucoup d’Italiens qui se
trouverent alors en cette Cour, de dire apres M<sup>r</sup> le Comte Nogarolli,
<em>Veramente à rotto i penelli alli primi d’Italia</em>. »';
    }
    elseif($id == 3652){
        $all_remark = 'Auteur du texte, le « Berger Alcidon » dévoile son nom dans
l’<em>Extraordinaire du ---Mercure--- galant</em> de juillet 1682, p. 394 de
l’édition lyonnaise de Thomas Amaulry : « DIEREVILLE, <em>du Pont-Levesque. cy-
devant le Berger Alcidon, du Fauxbourg S. Victor.</em> »';
    }
    elseif($id == 3670){
        $all_remark = 'L’auteur est présenté comme « un Homme de mérite & de
réputation » (p. 353). Son identité est révélée dans le Mercure de mai 1683, p.
315 : « Mr Beaubrun, ancien Recteur, & Trésorier de l’Académie de Peinture &
Sculpture ».';
    }
    elseif($id == 3721){
        $all_remark = 'Le nom de l’auteur et son domicile sont donnés dans le
Mercure d’août 1685, p. 309 : « Je vous ay mandé que la Fable énigmatique que je
vous envoyay le dernier mois, estoit de M. B. D. de Thoulouze. Je vous l’ay dit
sur un faux Memoire. Elle est de M. Broüilhet du Rocq, qui est de Blaye, & non de
Thoulouze. »';
    }
    elseif($id == 3733){
        $all_remark = 'Le Mercure précise dans la livraison du mois de février
1686, 1<em>ère</em> partie, p. 304-305 : « On m’a fait connoistre que l’une des
deux Enigmes du dernier mois estoit imprimée dans un recüeil de M<sup>r</sup>
Cotin. Comme il y a des Devineurs de bonne foy qui ne l’ont pas veuë dans ce
recüeil, j’en vais mettre icy les noms. »';
    }
    elseif($id == 3781){
        $all_remark = 'Le nom de l’auteur n’est donné que dans la livraison du
mois suivant, p. 333.';
    }
    elseif($id == 3783){
        $all_remark = 'L’activité de l’auteur est dévoilée dans le Mercure du mois
suivant, p. 334-335 : « La premiere [énigme du mois d’avril] est du mesme Autheur
```

```
que celle du <em>Canevas</em>. C’est un Abbé d’une Famille toute pleine d’esprit.
»';
    }
    elseif($id == 3810 || $id == 3811){
        $all_remark = 'Nous apprenons dans la livraison de janvier 1689, p. 263,
que les mots des énigmes de novembre 1688 « n’ont esté trouvez de personne ».';
    }
    elseif($id == 3815){
        $all_remark = 'Dans les lignes qui précèdent immédiatement l’énigme, p.
316, nous apprenons que la « Bergere Fleurette » est « Fille du défunt Berger
Fleuriste », autre auteur d’énigmes.';
    }
    elseif($id == 3831){
        $all_remark = 'La réponse de l’énigme, « un balay », est donnée dans
l’énigme elle-même sous forme d’acrostiche.';
    }
    elseif($id == 3843){
        $all_remark = 'Le nom du « Cavalier Angevin » auteur de cette énigme est
donné par le rédacteur du Mercure dans la livraison d’octobre 1692, p. 308.';
    }
    elseif($id == 3848){
        $all_remark = 'Le nom du « Cavalier d’Angers » auteur de cette énigme est
donné par le rédacteur du Mercure dans la livraison d’octobre 1692, p. 308.';
    }
    elseif($id == 3849){
        $all_remark = 'Réponse dans le Mercure du mois suivant, p. 329 : « Le mot
de l’Enigme du mois passé a esté trouvé de peu de personnes, & je croy que le seul
nom de l’Auteur leur a donné lieu de le deviner. Elle est de M. de Comiers, &
comme il est aveugle, & retiré dans les Quinze-vingts, on a pensé que le sens de
cette Enigme convenoit au petit Garçon qui le conduit. » L’énigme est ensuite
expliquée en détail.';
    }
    elseif($id == 3876){
        $all_remark = 'Nous lisons dans la livraison du mois suivant, p. 358 : «
L’Enigme du mois passé, qui estoit seulement de quatre Vers, n’a esté expliquée
dans son vray sens que par une seule personne, sous le nom du Jardinier de la
belle Tour de Rheins[.] C’estoit <em>le Songe</em>. »';
    }
    elseif($id == 3877){
        $all_remark = 'Dans la livraison du mois suivant, p. 301, nous apprenons
qu’une seule personne a trouvé la réponse de l’énigme.';
    }
    elseif($id == 3928){
        $all_remark = 'Le domicile de l’auteur, rue Barbette à Paris, est indiqué
plus haut, p. 274, dans la liste des personnes qui ont deviné l’énigme du mois
précédent.';
    }
    elseif($id == 3960){
        $all_remark = 'Dans le Mercure de juin 1701, p. 381, nous apprenons que «
quantité de personnes ont expliqué » cette énigme « sur l’esprit », alors que sa
réponse est la parole.';
    }
    elseif($id == 3975){
        $all_remark = 'Dans le Mercure de septembre 1702, p. 424-425, la réponse
donnée par le rédacteur du Mercure est « Or ». Toutefois, il s’agit d’une erreur
que le rédacteur corrige dans la livraison d’octobre, p. 420-421.';
    }
    elseif($id == 3994){
```

```
        $all_remark = 'L’unique énigme du mois précédent portait également sur la
jarretière. Au moment de donner la réponse de la présente énigme, le rédacteur du
Mercure note dans la livraison de mai 1704, p. 353 : « Le mot de la derniere
Enigme estoit <em>la jarretiere</em>, il a été trouvé par plusieurs personnes,
mais peu ont donné leurs noms, aïant deviné la jarretiere sans croire avoir trouvé
le veritable mot, ne s’imaginant pas qu’on donnât deux fois de suite une Enigme,
dont le mot fût le même. »';
    }
    elseif($id == 3995){
        $all_remark = 'Pagination erronée : la présente énigme devrait commencer à
la page 454 et non 354.';
    }
    elseif($id == 4000){
        $all_remark = 'Nous lisons dans la livraison du mois suivant, p. 378 : «
Quoique cette Enigme soit fort belle & que les rapports en soient fort justes,
elle est si peu connuë qu’il ne faut pas s’étonner si les Devineurs ne sont pas en
grand nombre, & si plusieurs ont crû que le veritable mot étoit le Songe ou le
Soleil. »';
    }
    elseif($id == 4011){
        $all_remark = 'L’énigme du mois précédent portait déjà sur l’oignon. Au
moment de donner la réponse de la présente énigme, le rédacteur du Mercure écrit
dans la livraison de septembre 1705, p. 368-369 : « L’Enigme du mois passé estoit
encore l’<em>Oignon</em> ; mais quoi qu’elle ait esté dévinée par plusieurs
personnes, il s’en est peu trouvé qui ayent osé en envoyer le mot, persuadez
qu’ils se trompoient, parce que ce n’estoit pas l’usage de donner deux fois de
faire une Enigme sur le même sujet. De maniere que j’ai réussi, en leur causant
l’agreable embarras que je m’estois proposé. »';
    }
    elseif($id == 4019){
        $all_remark = 'Le nom de l’auteur « Mr d’Aubicourt » n’est donné que dans
la livraison du mois suivant, p. 395.';
    }
    elseif($id == 4024){
        $all_remark = 'Au moment de donner la réponse dans le Mercure du mois
suivant, p. 384, le rédacteur indique que beaucoup de lecteurs ont cru que cette
énigme sur l’écriture portait sur l’imprimerie.';
    }
    elseif($id == 4031){
        $all_remark = 'Nous lisons dans la livraison du mois suivant, p. 410-411 :
« Le nombre de Devineurs seroit plus grand, si plusieurs n’avoient pas pris le
change, en croyant que l’Enigme estoit le Jeu de Dames. »';
    }
    elseif($id == 4034){
        $all_remark = 'Dans la livraison du mois suivant, p. 388, nous lisons que
plusieurs « Devineurs » ont expliqué cette énigme sur la nuit et le jour par « le
Soleil & la Lune ».';
    }
    elseif($id == 4052){
        $all_remark = 'Au moment de donner la réponse, le rédacteur du Mercure
précise dans la livraison du mois suivant, p. 363-364 : « Le mot de l’Enigme du
mois passé qui estoit <em>le Bois</em>, a paru si difficile, qu’il n’a esté trouvé
que par cinq ou six personnes seulement. Si personne ne l’avoit trouvé, on
pourroit dire que c’est la faute de l’Enigme, mais on ne s’en peut plaindre quand
le mot ne seroit trouvé que par une seule personne, puisque l’Enigme est juste ;
mais qu’elle est difficile. »';
    }
    elseif($id == 4053){
```

```
        $all_remark = 'Nous apprenons dans la livraison du mois suivant, p. 344,
que cette énigme sur l’esprit n’a été résolue que par « tres-peu de personnes ».';
    }
    elseif($id == 4063){
        $all_remark = 'L’unique énigme du mois précédent portait déjà sur la
cloche. Le rédacteur du Mercure commente ainsi cette répétition au moment de
donner la réponse dans la livraison du mois de janvier 1710, p. 344 : « [ces
énigmes] ayant le mesme mot, sont neanmoins entierement differentes ; ce qui
marque la diversité du genie des hommes, qui en travaillant sur la mesme matiere
ne rencontrent en rien. »';
    }
    elseif($id == 4064){
        $all_remark = 'À propos de cette énigme, nous lisons dans le Mercure de
février 1710, p. 333 : « [l’énigme] du mois dernier estoit <em>l’Ortographe</em>,
de la maniere dont plusieurs personnent tâchent aujourd’huy d’introduire l’usage,
ce qui a embarassé particulierement ceux qui ne se sont pas attachez avec assez de
soin à remarquer comment cette Enigme estoit écrite, ce qui fait que plusieurs se
sont trompez dans l’explication qu’ils luy ont donnée. »';
    }
    elseif($id == 4070){
        $all_remark = 'À propos de l’auteur de cette énigme en prose, voir Fabien
Simon, « Antoine de Vienne Plancy ou un feuilleton de la langue universelle dans
la France du XVII<sup>e</sup> siècle », in Paule Petitier, Fabien Simon (dir.),
<em>L’historien et les langues</em>, <em>Écrire l’histoire</em>, n<sup>o</sup> 19,
2019, p. 127-137.';
    }
    elseif($id == 4082){
        $all_remark = 'L’auteur « M<sup>r</sup> de Souv..... » est probablement «
Mr de Souvenel, de Rennes en Bretagne » qui publie une énigme sur le miroir dans
le Mercure d’octobre 1708, p. 362.';
    }
    elseif($id == 4083){
        $all_remark = 'Le nom de l’auteur (qui est le rédacteur du Mercure) peut
être déduit d’une note qu’il laisse dans la livraison de janvier 1711, p. 322-
323.';
    }
    elseif($id == 4087){
        $all_remark = 'Cette énigme sur le sang fait référence à une anecdote que
le rédacteur du Mercure raconte dans la livraison du mois suivant, 3<sup>e</sup>
partie, p. 147-154.';
    }
    elseif($id == 4106){
        $all_remark = 'Le rédacteur du Mercure précise dans la livraison du mois
suivant que personne n’a trouvé la solution de cette énigme, ce qui prive la «
Bourgeoisie subalterne » (p. 185) du plaisir de lire la liste des noms des
devineurs.';
    }
    elseif($id == 4113){
        $all_remark = 'Auteur présumé, d’après le texte qui précède directement
l’énigme : Jerico le Brutal.';
    }
    elseif($id == 4152){
        $all_remark = 'Nous lisons dans la livraison du mois suivant, p. 279 : «
Mr D. L. autheur de cette Enigme, me mande qu’on la luy a dérobée, & qu’il
desavoüe la lettre qu’on m’a écrite sous son nom en me l’envoyant. »';
    }
    elseif($id == 4160 || $id == 4161){
        $all_remark = 'L’auteur est présenté comme « M. Dumoulin » dans la
livraison du mois suivant, p. 348.';
```

```
    }
    elseif($id == 4163){
        $all_remark = 'Le rédacteur du <em>Nouveau ---Mercure--- galant</em>
Hardouin Le Fèvre de Fontenay se déclare d’abord l’auteur de l’énigme. Toutefois,
il la désavoue dans la livraison du mois suivant, p. 275 : « Je sçay que la
premiere [énigme] du mois passé estoit bonne, & l’autre mauvaise ; mais ce n’est
ma faute que par la complaisance que j’ay euë pour Madame de N... qui l’a faite &
qui m’a pressé malgré moy de m’en dire l’auteur : je l’ay fait, & je me garderois
bien de parler de cette avanture, si, lorsque je luy ay dit aprés tout le monde,
qu’elle ne valoit rien, elle ne s’estoit pas mocquée de moy, & ne s’en mocquoit
pas encore. Au reste son Enigme ressemble à une poire comme à un moulin à vent.
»';
    }
    elseif($id == 4165){
        $all_remark = 'À propos de l’auteur de ce texte, le rédacteur du Mercure
précise, dans les lignes qui précèdent immédiatement l’énigme à la page 279 : « Un
de mes amis m’a donné cette autre Enigme qu’il a dérobé [<em>sic</em>] à un des
siens. Celuy qui en est l’Auteur est de Villeneuve d’Avignon ; c’est un homme de
Lettre, connu non seulement par un Journal qu’il a fait des nouvelles de l’ordre
de la boisson ; mais encore par plusieurs autres ouvrages d’esprit. » Cet auteur
est donc un des deux rédacteurs du périodique <em>Nouvelles de l’ordre de la
boisson</em>. Il peut s’agir de François de Morgier (mort en 1723) ou de l’abbé
Jean-Antoine de Charnes (1641-1728).';
    }
    elseif($id == 4189 || $id == 4190){
        $all_remark = 'Dans la livraison du mois suivant, p. 254-255, le rédacteur
du Mercure précise que personne n’a deviné les deux énigmes du mois de décembre,
qui sont du même auteur (sous deux pseudonymes différents).';
    }
    elseif($id == 4229){
        $all_remark = 'Donnée dans la livraison de novembre 1717, p. 166, la
réponse à cette énigme est d’abord présentée comme le miroir, mais le rédacteur du
<em>Nouveau ---Mercure---</em> annonce dans la livraison de décembre, p. 181,
qu’il s’agit d’une erreur et que la véritable réponse est la quenouille.';
    }
    elseif($id == 4259){
        $all_remark = 'L’identité de l’auteur est déduite de Nicolas Viton de
Saint-Allais, <em>Nobiliaire universel de France ou recueil général des
généalogies historiques des maisons nobles de ce royaume</em> [...], Paris, chez
l’auteur, t. 11, 1817, p. 148.';
    }
    elseif($id == 4268){
        $all_remark = 'La page de l’exemplaire est numérotée par erreur 158.';
    }
    elseif($id == 4273){
        $all_remark = 'Il est peu probable que la signataire soit le véritable
auteur du texte, cette énigme ayant déjà paru anonymement dans les livraisons de
décembre 1695, p. 310-311, et mars 1706, p. 403-404.';
    }
    elseif($id == 4349){
        $all_remark = 'Les informations relatives à l’auteur sont données dans la
livraison du mois suivant, à la page 70. Nous y apprenons également que « M.
Cholier, fils de M. le Prevôt des Marchands, l’explica l’année passée avec
beaucoup de succès, devant une Assemblée des plus nombreuses & des plus illustres
qu’on ait vuës à Lyon ».';
    }
    elseif($id == 4362){
        $all_remark = 'La page de l’exemplaire est numérotée par erreur 219.';
    }
```

```
    elseif($id == 4418){
        $all_remark = 'L’activité de l’auteur (chanoine) est donnée en toutes
lettres dans le Mercure du 5 juin 1779, p. 13.';
    }
    elseif($id == 4419){
        $all_remark = 'L’énigme est republiée dans les <em>Oeuvres choisies de P.
Laujon, membre de l’Institut</em> [...], Paris, Léopold Collin, Patris et
C<em>ie</em>, t. 1, 1811, p. 432.';
    }
    elseif($id == 4540){
        $all_remark = 'Le nom de famille de l’auteur existe également sous les
formes « Lemozy d’Orel » et « Lemosy d’Orel ».';
    }
    elseif($id == 4572){
        $all_remark = 'Cette énigme constitue manifestement la réécriture d’un
autre texte paru anonymement dans le Mercure de février 1772, p. 72-74. L’auteur
anonyme venait alors de Perpignan.';
    }
    elseif($id == 4625){
        $all_remark = 'L’auteur est M. de Louvencourt. Son patronyme apparaît en
toutes lettres dans la signature d’un logogriphe de la livraison du 25 novembre
1780, p. 160.';
    }
    elseif($id == 4660 || $id == 4661){
        $all_remark = 'Le numéro de page imprimé est erroné.';
    }
    elseif($id == 4670){
        $all_remark = 'Le texte provient de « Sounalon », toponyme que nous ne
parvenons guère à identifier. Il faut peut-être lire « Sounalou », hameau de la
commune de Sumène dans l’actuel département du Gard.';
    }
    elseif($id == 4678){
        $all_remark = 'À propos du lieu de provenance de l’auteur, indiqué dans la
signature, nous supposons qu’il faut lire « Pont-l’Évêque, en Normandie » et non «
Port-l’Evêque ».';
    }
    elseif($id == 4704 || $id == 4721){
        $all_remark = 'L’auteur présumé est M. le Barbier C., de Boudeville près
Rouen. Voir la signature du logogriphe qu’il publie dans le Mercure du 6 août
1791, n<sup>o</sup> 32, p. 35.';
    }
    elseif($id == 4706){
        $all_remark = '« Mde de Trignolles » s’approprie une énigme parue sous le
nom de « J. A. F. C. D. N. » dans le Mercure d’avril 1739, p. 713-714.';
    }
    elseif($id == 4740){
        $all_remark = 'L’auteur de ce logogriphe est « le jeune Auteur du
Logogryphe de la Vitre », paru dans le Mercure du 21 avril 1781, n<sup>o</sup> 16,
p. 103, qui signait alors « M. Marin fils, âgé de 13 ans, à la Ciotat ».';
    }
    elseif($id == 4805){
        $all_remark = 'L’auteur présumé est M. l’Abbé Lacolley, signataire d’une
énigme dans la livraison du 9 février 1788, n<sup>o</sup> 6, p. 56-57.';
    }
    elseif($id == 4839){
        $all_remark = 'Le signataire est aussi l’« Auteur du Logogryphe
<em>Pistolet</em>, inséré dans le ---Mercure--- du 7 Septembre » 1782,
n<sup>o</sup> 36, p. 7-8.';
    }
```

```
    elseif($id == 4843){
        $all_remark = 'L’auteur est peut-être « M. le Chevalier de la Mothe », signataire d’un logogriphe dans le Mercure du 23 février 1782, n<sup>o</sup> 8, p. 153-154.';
    }
    elseif($id == 4924 || $id == 4925){
        $all_remark = '« M. H** » est vraisemblablement le même auteur que « M. H*** » qui fournit d’autres charades au Mercure.';
    }
    elseif($id == 4941){
        $all_remark = 'L’auteur est peut-être « M. Verani de Varenne » qui publiera une énigme dans le Mercure du 10 avril 1784, n<sup>o</sup> 15, p. 56, alors qu’il sera « âgé de 14 ans ».';
    }
    elseif($id == 5004 || $id == 5005){
        $all_remark = 'L’auteure présumée est « Mlle Brisoult l’aînée, à S. Dizier ». Voir sa signature dans le Mercure du 20 mai 1780, n<sup>o</sup> 21, p. 110.';
    }
    elseif($id == 5007 || $id == 5017 || $id == 5079 || $id == 5100 || $id == 5017 || $id == 5146 || $id == 5370){
        $all_remark = 'L’auteur présumé est « M. Santrin », signature qui apparaît dans le Mercure du 31 mai 1783, n<sup>o</sup> 22, p. 210.';
    }
    elseif($id == 5084){
        $all_remark = 'Antoine-Vincent Arnault s’attribue cette charade dans ses <em>Souvenirs d’un sexagénaire</em>, Paris, Dufey, 1833, t. 1, p. 52. Dans la livraison du 8 mai 1784, n<sup>o</sup> 19, p. 54, nous apprenons que cette charade « fait allusion à la première représentation de <em>Coriolan</em>, au profit des Pauvres ».';
    }
    elseif($id == 5090){
        $all_remark = 'Dans les années 1780, les signatures « M. de G. » et « M. C. de G. » renvoient très probablement au même auteur, officier au régiment de Boulonnais.';
    }
    elseif($id == 5102 || $id == 5103 || $id == 5503 || $id == 5504 || $id == 5505 || $id == 5243){
        $all_remark = 'Erreur dans la numérotation des pages.';
    }
    elseif($id == 5113){
        $all_remark = 'L’auteur en peut-être « M. Noël P..... » à Chalon-sur-Saône. Voir le logogriphe publié sous ce nom dans le Mercure du 31 juillet 1784, n<sup>o</sup> 31, p. 198-199.';
    }
    elseif($id == 5131 || $id == 5331 || $id == 5430){
        $all_remark = 'Cet auteur signe d’autres textes « M. Révial ». Voir le logogriphe qu’il publie sous ce nom dans le Mercure du 2 septembre 1786, n<sup>o</sup> 35, p. 20.';
    }
    elseif($id == 5166){
        $all_remark = 'Le signataire « M. l’Abbé le Dru » s’approprie une énigme de Charles Cotin parue dans les recueils de cet auteur et dans les Mercure de janvier 1686, p. 200, et de septembre 1725, vol. 2, p. 2248.';
    }
    elseif($id == 5190 || $id == 5320){
        $all_remark = 'L’auteur est probablement « M. Larivière, de Falaise » qui, en 1780, était « Écolier de Logique sous M. Adam, au Collége du Bois, en l’Université de Caen ». Voir le logogriphe qu’il publie dans le Mercure du 22 juillet 1780, n<sup>o</sup> 30, p. 165-166.';
```

```
    }
    elseif($id == 5316){
        $all_remark = 'Dans le texte du logogriphe, l’auteur précise que le mot de la réponse (qui est « Olivier ») forme aussi son nom.';
    }
    elseif($id == 5362){
        $all_remark = 'Dans la livraison suivante du Mercure, p. 51, la réponse proposée est « bonsoir ». La correction de « bonsoir » en « bonjour » intervient dans la livraison du 18 février 1786, n<sup>o</sup> 7, p. 107.';
    }
    elseif($id == 5388){
        $all_remark = 'L’auteur présumé est « M. le Ch. de Saint-Gervais », officier d’infanterie au régiment de Picardie. Voir les textes qu’il publie sous ce nom dans les Mercure du 29 mars 1783, n<sup>o</sup> 13, p. 221, et du 1<sup>er</sup> novembre 1783, n<sup>o</sup> 44, p. 12-13.';
    }
    elseif($id == 5391){
        $all_remark = 'L’auteur de ce texte est probablement le même qu’un certain « M. B... » qui provient de Saint-Quentin et qui donnera des textes au Mercure dans les années suivantes. Voir par exemple sa charade du 10 septembre 1791, n<sup>o</sup> 37, p. 52.';
    }
    elseif($id == 5396 || $id == 5432){
        $all_remark = 'Le signataire « M. Guérin du N** » s’approprie une énigme parue anonymement dans le Mercure galant de septembre 1710, p. 310-315. Ce signataire est probablement le même « M. Guérin » que celui qui provient de « Chevanne Gazeau, en Nivernois ». Voir l’énigme que ce dernier publie dans le Mercure du 22 juillet 1786, n<sup>o</sup> 29, p. 163-164.';
    }
    elseif($id == 5436){
        $all_remark = 'L’auteur est probablement la même personne que « M. le Vicomte de Gal... ». Ce dernier est notamment l’auteur d’une énigme parue dans le Mercure du 10 juin 1786, n<sup>o</sup> 23, p. 54-55.';
    }
    elseif($id == 5460){
        $all_remark = 'Republié dans le Mercure du 26 mai 1792, n<sup>o</sup> 21, p. 89, ce texte portera alors la signature de « Mad. A... ».';
    }
    elseif($id == 5466){
        $all_remark = 'L’auteur est peut-être le poète Jean-François Guichard (1731-1811), né et mort à Chartrettes.';
    }
    elseif($id == 5534){
        $all_remark = 'Le lieu de provenance « Fouquenies » est déduit d’après l’existence d’un chemin du Bréda à proximité de Monmille, hameau intégré à la commune de Fouquenies.';
    }
    elseif($id == 5560){
        $all_remark = 'Dans un compte rendu du <em>Manuel des oisifs</em>, page 81 du présent numéro du Mercure, le lecteur apprend que la charade provient de cet ouvrage. L’auteur du <em>Manuel des oisifs</em> est d’après les bibliographes l’abbé Semillard des Ovillers, curé de Tremblay.';
    }
    elseif($id == 5596){
        $all_remark = 'Mentionné dans la signature, le « Couvent d’Orient en Rouergue » doit être le monastère Notre-Dame d’Orient à Laval-Roquecezière.';
    }
    elseif($id == 5630 || $id == 5631){
```

```
        $all_remark = 'L’auteur est « un natif de Saint-Just des Marais ». L’indication « près Béarn » est peut-être fautive. Il pourrait s’agir de Saint-Just-des-Marais près de Beauvais.';
    }
    elseif($id == 5647 || $id == 5648 || $id == 5659){
        $all_remark = 'L’auteur présumé est « M. Busson » qui se présente comme un « Elève des Ponts & Chaussées » dans le Mercure du 12 août 1786, n<sup>o</sup> 32, p. 56.';
    }
    elseif($id == 5674){
        $all_remark = 'Nous lisons dans le Mercure du 16 février 1788, n<sup>o</sup> 7, p. 144 : « Au titre de la Charade du N<sup>o</sup>. 5, dont les mots sont <em>Vermine</em> & <em>Minerve</em> ; au lieu de <em>Charade</em>, lisez, <em>Charade & Logogriphe</em>. ».';
    }
    elseif($id == 5703){
        $all_remark = 'Officier au régiment de Neustrie, l’auteur signe de son nom complet « M. de Chateauvieux » un logogriphe dans le Mercure du 23 mai 1789, n<sup>o</sup> 21, p. 149-150.';
    }
    elseif($id == 5707){
        $all_remark = 'L’écolier qui signe cette énigme l’usurpe manifestement à l’abbé Métrail, aumônier de l’hospice royal, qui a déjà publié ce texte dans le Mercure du 2 avril 1785, n<sup>o</sup> 14, p. 9-10.';
    }
    elseif($id == 5738){
        $all_remark = 'Le lieu de provenance est peut-être Thiers (qui aurait été transcrit par erreur « Chiers »). À Thiers se trouve un autre auteur d’énigme du nom de Leclerc en 1785. Voir le texte publié par ce dernier dans le Mercure du 17 décembre 1785, n<sup>o</sup> 51, p. 101-102.';
    }
    elseif($id == 5742){
        $all_remark = 'L’activité de l’auteur est indiquée dans la signature de la charade, à la page précédente.';
    }
    elseif($id == 5745){
        $all_remark = 'L’auteur est sans doute le même que le garde du corps du comte d’Artois qui signe « M. Prévost ». Voir le logogriphe que ce dernier publie dans le Mercure du 14 juillet 1787, n<sup>o</sup> 28, p. 54-55.';
    }
    elseif($id == 5761){
        $all_remark = 'L’auteur se présentera comme un « Com. aux Ferm. » dans la signature d’une énigme ultérieure, dans le Mercure du 17 janvier 1789, n<sup>o</sup> 3, p. 108, ce qui permet de déduire son activité de commis aux fermes.';
    }
    elseif($id == 5762){
        $all_remark = 'Le signataire s’approprie une énigme qui a déjà paru anonymement dans le Mercure galant de janvier 1689, p. 265, et dans le Mercure de France d’octobre 1766, vol. 2, p. 57.';
    }
    elseif($id == 5770){
        $all_remark = 'Outre le premier vers, le signataire reprend exactement le texte d’une charade de « M. Berthier, Officier au Rég. de Picardie », parue dans le Mercure du 18 décembre 1784, n<sup>o</sup> 51, p. 118.';
    }
    elseif($id == 5798){
        $all_remark = 'L’auteur présumé est « M. Pillet, Lyonnois », qui signera un logogriphe dans le Mercure du 22 août 1789, n<sup>o</sup> 34, p. 101.';
```

```
    }
    elseif($id == 5800){
        $all_remark = 'Dans la signature, au lieu de « Montade », il faut peut-
être lire Montoire qui désignerait l’actuelle commune de Montoire-sur-le-Loir.';
    }
    elseif($id == 5864){
        $all_remark = 'Le signataire se présente comme « un Abonné », mais cet
abonné ne peut être l’auteur de l’énigme, puisque celle-ci paraît pour la première
fois sous le pseudonyme de « L’Inconnu de la rue Saint Jean de Beauvais » dans Le
Mercure galant d’avril 1703, p. 377.';
    }
    elseif($id == 5896){
        $all_remark = 'Nous n’avons pas identifié le lieu indiqué dans la
signature, mais il existe en France plusieurs lieux-dits « Fromenteau » (plutôt
que « Fromenteault »).';
    }
    elseif($id == 5904 || $id == 6048){
        $all_remark = 'L’auteur présumé est « M. Sebire de Beauchêne, Officier de
Marine », qui publie sous ce nom des énigmes et quelques logogriphes depuis la
livraison du 4 novembre 1786, n<sup>o</sup> 44, p. 6.';
    }
    elseif($id == 5923){
        $all_remark = 'Le texte a déjà paru dans le Mercure du 3 février 1781,
<sup>o</sup> 5, p. 21, où il était signé par « M. Dulaurens, Avocat au Parlement
».';
    }
    elseif($id == 6027){
        $all_remark = 'Lieu indiqué dans la signature, « Patinges » est un ancien
nom du village de Torteron.';
    }
    elseif($id == 6044 || $id == 6047 || $id == 6056){
        $all_remark = 'L’auteur est peut-être « M. Damise », qui se présente comme
un « Av[ocat] en Parl[ement] » dans la signature d’une charade, dans le Mercure du
10 avril 1790, n<sup>o</sup> 15, p. 56.';
    }
    elseif($id == 6050 || $id == 6051){
        $all_remark = 'L’auteur du présent texte semble le même que celui qui, à
partir de la livraison du 13 octobre 1792, publiera de nombreux jeux d’esprit dans
le Mercure sous le nom de « M. Ch. M. d. v. ».';
    }
    elseif($id == 6057){
        $all_remark = 'Dans la signature, au lieu de « Vorne », il faut lire «
Verne », village de l’actuel département du Doubs, où un certain Baverel est actif
comme curé pendant la Révolution.';
    }
    elseif($id == 6072 || $id == 6074){
        $all_remark = 'Dans la livraison précédente du Mercure (14 août 1790,
n<sup>o</sup> 33, p. 55), cet auteur est présenté comme le « Curé de St-Maixme
».';
    }
    elseif($id == 6086){
        $all_remark = 'Dans la livraison du 14 août 1790, n<sup>o</sup> 33, p. 55,
cet auteur est présenté comme le « Curé de St-Maixme ».';
    }
    elseif($id == 6091 || $id == 6095){
        $all_remark = 'Dans le Mercure, cet auteur est tantôt présenté comme «
Daviau », tantôt comme « Daveau ».';
    }
    elseif($id == 6105){
```

```
        $all_remark = 'Sous un même titre et après une introduction en vers, ce
texte rassemble une charade, une énigme et un logogriphe ayant la même réponse.';
    }
    elseif($id == 6113){
        $all_remark = 'Apparaissant dans la signature, le lieu-dit Trégranteur se
trouve sur la commune de Guégon, près de Josselin, dans l'actuel département du
Morbihan. Le signataire s'approprie un texte paru anonymement dans le Mercure
galant de janvier 1695, p. 343, et souvent republié par la suite.';
    }
    elseif($id == 6114){
        $all_remark = 'Présenté comme un « Américain », cet auteur est
probablement le même qui signe plusieurs textes « M. Prévôt de Moka, Américain »,
à partir de la livraison du 11 septembre 1779, p. 58.';
    }
    elseif($id == 6118){
        $all_remark = 'Cet officier d'infanterie signera « le Ch. De Pré.. » une
énigme dans le Mercure du 1<sup>er</sup> janvier 1791, n<sup>o</sup> 1, p. 28-
30.';
    }
    elseif($id == 6124){
        $all_remark = 'Dans le Mercure du 11 février 1792, n<sup>o</sup> 6, p. 42,
nous apprenons que M. Gaillard est docteur en médecine à Richelieu.';
    }
    elseif($id == 6131){
        $all_remark = 'Ce texte est fortement inspiré d'une énigme parue
anonymement dans le Mercure galant d'août 1697, p. 282-283.';
    }
    elseif($id == 6143){
        $all_remark = 'Le signataire « M. Verlhac » s'approprie une énigme parue
anonymement dans le Mercure galant de novembre 1683, p. 279-280.';
    }
    elseif($id == 6163){
        $all_remark = 'Le prénom de l'auteur est « François », d'après la
signature du logogriphe qu'il publie dans le Mercure du 8 janvier 1791,
n<sup>o</sup> 2, p. 57.';
    }
    elseif($id == 6177 || $id == 6178){
        $all_remark = 'L'auteur se présente comme « le Berger de Monterrin ». Or
l'orthographe « Monterrin » est attestée pour désigner « Monterrein », dans le
département du Morbihan, pendant la période révolutionnaire.';
    }
    elseif($id == 6199){
        $all_remark = 'Cette énigme a déjà paru dans le Mercure galant de mai
1682, p. 232-234. Elle était alors présentée comme l'œuvre du « Geometre par
hazard ».';
    }
    elseif($id == 6211){
        $all_remark = 'La signataire s'approprie une énigme parue dans le Mercure
du 15 mars 1788, n<sup>o</sup> 11, p. 101, sous le nom de « M. de L***** ».';
    }
    elseif($id == 6217 || $id == 6219){
        $all_remark = 'La signataire s'approprie une énigme parue dans le Mercure
du 11 juillet 1789, n<sup>o</sup> 28, p. 53, sous le nom de « M. l'Abbé Dubois,
Et[udiant] à Brive ».';
    }
    elseif($id == 6244){
        $all_remark = 'Une publication antérieure du même texte (avec un premier
vers sensiblement différent) dans le Mercure du 25 septembre 1778, p. 260,
présente l'auteur comme un certain « M. *** Abonné au Mercure ».';
```

```
    }
    elseif($id == 6268){
        $all_remark = 'Le signataire se présente comme l’auteur du bref
<em>Bouquet présenté à M. Necker à son entrée à Paris le 30 juillet 1789</em>,
paru chez Valleyre l’aîné. Cet auteur est un certain abbé de Courchon.';
    }
    elseif($id == 6281){
        $all_remark = 'Déjà publié dans le Mercure du 15 juin 1779, p. 141-142, ce
texte était alors signé « Par M. Joly de Bagneux ».';
    }
    elseif($id == 6283){
        $all_remark = 'Le signataire se présente comme « un Abonné », mais ce
texte bien connu du poète Houdar de La Motte a déjà paru plusieurs fois dans le
Mercure à partir de la livraison de janvier 1728, p. 103.';
    }
    elseif($id == 6286){
        $all_remark = 'Dans la signature du texte qui précède celui-ci, le
signataire est présenté comme « un Abonné », mais l’énigme a déjà paru sous le nom
de « M. F. de Montferrand » dans le Mercure du 22 avril 1780, n<sup>o</sup> 17, p.
161.';
    }
    elseif($id == 6287){
        $all_remark = 'Dans la signature de la charade publiée un peu plus haut,
p. 122, le signataire est présenté comme « un Abonné », mais le logogriphe a déjà
paru dans le Mercure du 28 octobre 1780, n<sup>o</sup> 44, p. 162-163. L’auteur se
définissait alors comme un cordonnier du village de Soupir.';
    }
    elseif($id == 6288){
        $all_remark = 'L’auteur présumé est M. Gaillard, docteur en médecine à
Richelieu, qui publie plusieurs jeux d’esprit dans le Mercure à partir de la
livraison du 25 décembre 1790, n<sup>o</sup> 52, p. 139, et jusqu’en 1792.';
    }
    elseif($id == 6289){
        $all_remark = 'Ce texte a déjà paru sous le nom de « M. Joly » dans le
Mercure du 18 mars 1780, n<sup>o</sup> 12, p. 104.';
    }
    elseif($id == 6292){
        $all_remark = 'Déjà publié dans les Mercure du 15 décembre 1781,
n<sup>o</sup> 50, p. 169, et du 15 octobre 1791, n<sup>o</sup> 42, p. 89, ce texte
était alors présenté comme l’œuvre d’un certain « M. Pruadere de Saint-Girons ».';
    }
    elseif($id == 6293){
        $all_remark = 'Ce texte a déjà paru sous le nom de « M. M*** » dans le
Mercure du 29 octobre 1791, n<sup>o</sup> 44, p. 162.';
    }
    elseif($id == 6298){
        $all_remark = 'Ce texte a déjà paru sous le nom de « M. C.... Avocat »
dans le Mercure du 21 juillet 1781, n<sup>o</sup> 29, p. 106-107.';
    }
    elseif($id == 6299){
        $all_remark = 'L’auteur est présenté comme « le même » que celui du texte
précédent, lui-même présenté comme « un Abonné ». C’est toutefois peu probable,
les deux textes ayant déjà paru dans le Mercure du 21 juillet 1781, n<sup>o</sup>
29, p. 106-108, le premier sous le nom de « M. C.... », le second sous celui de «
M. Fréville ».';
    }
    elseif($id == 6307){
```

```
        $all_remark = 'L’auteure est présentée comme « une Demoiselle », mais ce
texte est la reprise d’une énigme anonyme parue dans le Mercure du 15 juillet
1778, p. 141.';
    }
    elseif($id == 6319){
        $all_remark = 'Ce texte constitue la reprise d’une énigme publiée dans le
Mercure du 22 juillet 1786, n<sup>o</sup> 29, p. 163-164. L’auteur se présentait
alors comme « M. Guérin de Chevanne Gazeau, en Nivernois ».';
    }
    elseif($id == 6334){
        $all_remark = 'L’auteure est « Mde ..... à Valence, en Dauphiné », d’après
une publication antérieure du même texte dans le Mercure du 23 septembre 1780,
n<sup>o</sup> 39, p. 150.';
    }
    elseif($id == 6349){
        $all_remark = 'L’auteur est « M. Dracolf, à Strasbourg », d’après une
publication antérieure du même texte dans le Mercure du 15 janvier 1779, p. 148-
149.';
    }
    elseif($id == 6358){
        $all_remark = 'Une publication antérieure de ce texte dans le Mercure du
12 août 1780, n<sup>o</sup> 33, p. 59, présente l’auteur comme « M. d’A…, Officier
au troisième Régiment des Chevaux-Légers ».';
    }
    elseif($id == 6362){
        $all_remark = 'L’auteur présumé est « M. Delorme l’aîné, de Lyon », un des
rares auteurs d’énigmes qui, à cette époque, indiquent un nom de ville dans leurs
signatures. Voir par exemple le logogriphe qu’il publie dans le Mercure du 29
septembre 1792, n<sup>o</sup> 39, p. 91.';
    }
    elseif($id == 6364){
        $all_remark = 'L’auteur est « M. Richard, Avocat en Parlement », d’après
une publication antérieure du même texte dans le Mercure du 16 décembre 1780,
n<sup>o</sup> 51, p. 110.';
    }
    elseif($id == 6366){
        $all_remark = 'L’auteur est « M. Courtat », d’après une publication
antérieure du même texte dans le Mercure du 5 mars 1785, n<sup>o</sup> 10, p. 7.';
    }
    elseif($id == 6369){
        $all_remark = 'L’auteur est « M. le Th.*** », d’après une publication
antérieure du même texte dans le Mercure du 18 juin 1785, n<sup>o</sup> 25, p.
102.';
    }
    elseif($id == 6370){
        $all_remark = 'L’auteur est « M. Houllier de Saint-Remy », d’après une
publication antérieure du même texte dans le Mercure du 9 décembre 1780,
n<sup>o</sup> 50, p. 63.';
    }
    elseif($id == 6373){
        $all_remark = 'Ce texte, qui a déjà paru dans le Mercure du 7 octobre
1780, n<sup>o</sup> 41, p. 23, était alors signé « Par un Grainetier de la rue de
la Mortellerie, à Paris ».';
    }
    elseif($id == 6376){
        $all_remark = 'L’auteur est « M. Parthon », d’après une publication
antérieure du même texte dans le Mercure du 16 septembre 1780, n<sup>o</sup> 38,
p. 116.';
    }
```

```
    elseif($id == 6378){
        $all_remark = 'L’auteur est « M. D***** D. », d’après une publication antérieure du même texte dans le Mercure du 3 juillet 1784, n<sup>o</sup> 27, p. 8.';
    }
    elseif($id == 6381){
        $all_remark = 'L’auteur est « M. J[ean-]B[aptiste] Calvet, de la Société Littéraire du Collège de Reims », d’après une publication antérieure du même texte dans le Mercure du 2 juillet 1791, n<sup>o</sup> 27, p. 29.';
    }
    elseif($id == 6391){
        $all_remark = 'L’auteure est « Mme la Marquise de **** », d’après une publication antérieure du même texte dans le Mercure du 12 novembre 1785, n<sup>o</sup> 46, p. 68-69.';
    }
    elseif($id == 6394){
        $all_remark = 'L’auteur est « le M*** », d’après une publication antérieure du même texte dans le Mercure du 19 février 1780, n<sup>o</sup> 8, p. 119.';
    }
    elseif($id == 6397){
        $all_remark = 'Cette énigme est fortement inspirée d’un texte « de M. de Conjon fils, de Bayeux », paru dans le Mercure du 3 septembre 1785, n<sup>o</sup> 36, p. 7.';
    }
    elseif($id == 6400){
        $all_remark = 'L’auteur est « M. P. D., de Dol, en Bretagne », d’après une publication antérieure du même texte dans le Mercure du 19 août 1780, n<sup>o</sup> 34, p. 120.';
    }
    elseif($id == 6445){
        $all_remark = 'Le signataire « M. Ch. M. D. V. » s’approprie une énigme parue sous le nom de « M. Verani de Varenne, âgé de 14 ans » dans le Mercure du 10 avril 1784, n<sup>o</sup> 15, p. 56.';
    }
    elseif($id == 6460){
        $all_remark = 'Le Mercure indique avec prudence : « Cette énigme est attribuée à J. J. Rousseau. » Aujourd’hui, cette attribution est confirmée. La réponse, qui devrait se trouver dans le numéro 58 du Mercure (27 février 1793), n’y est pas donnée. On la trouve cependant au numéro 59 du 28 février 1793, p. 465. Voir Timothée Léchot, « L’énigme et le portrait : sur un quatrain de Jean-Jacques Rousseau », in Lionel Bartolini, Isaline Deléderray-Oguey, Grégoire Oguey (dir.), <em>Chanter l’histoire en réveillant les sources. Mélanges d’histoire(s) offerts au professeur Jean-Daniel Morerod par ses amis, collègues et anciens étudiants</em>, Neuchâtel, Alphil – Presses universitaires suisses, 2023, p. 397-418.';
    }
    elseif($id == 6465){
        $all_remark = 'L’auteur est « M. le Th.*** », d’après une publication antérieure du même texte dans le Mercure du 18 juin 1785, n<sup>o</sup> 25, p. 102.';
    }
    elseif($id == 6468){
        $all_remark = 'L’auteur est « M. le Chevalier Amyot », d’après une publication antérieure du même texte dans le Mercure du 11 juin 1785, n<sup>o</sup> 24, p. 52.';
    }
    elseif($id == 6469){
```

```
        $all_remark = 'L’auteur est « M. de Roucelle, Commissaire des Guerres de
la Maison militaire de Monsieur », d’après une publication antérieure du même
texte dans le Mercure du 12 mars 1785, n<sup>o</sup> 11, p. 55-56.';
    }
    elseif($id == 6470){
        $all_remark = 'L’auteure est « Mlle de Malerme, de Bruxelles », d’après
une publication antérieure du même texte dans le Mercure du 9 décembre 1786,
n<sup>o</sup> 49, p. 53-54.';
    }
    elseif($id == 6471){
        $all_remark = 'L’auteur est « M. Duruchanoy, de Saint-Quentin », d’après
une publication antérieure du même texte dans le Mercure du 8 avril 1786,
n<sup>o</sup> 14, p. 89.';
    }
    elseif($id == 6474){
        $all_remark = 'Cette énigme a paru sous une forme sensiblement différente
dans le Mercure du 9 décembre 1786, n<sup>o</sup> 49, p. 53. Elle était alors
signée : « M. N. Delisle ».';
    }
    elseif($id == 6475){
        $all_remark = 'Ce texte constitue manifestement la réécriture de deux
textes antérieurs parus respectivement dans les Mercure des 11 mars 1786,
n<sup>o</sup> 10, p. 55-56, et 26 août 1786, n<sup>o</sup> 34, p. 153-154.
L’auteur du texte du 26 août 1786 était « M. Robert des Roches ».';
    }
    elseif($id == 6477){
        $all_remark = 'L’auteure est « Mlle B.... de Toul », d’après une
publication antérieure et presque identique de ce texte dans le Mercure du 12 août
1786, n<sup>o</sup> 32, p. 53-54.';
    }
    elseif($id == 6478){
        $all_remark = 'L’auteur est « M. D. R. M. A M. D. R. D. B. », d’après une
publication antérieure du même texte (avec un distique supplémentaire) dans le
Mercure du 15 avril 1786, n<sup>o</sup> 15, p. 127-128.';
    }
    elseif($id == 6479){
        $all_remark = 'Ce texte a déjà paru dans le Mercure du 17 juin 1786,
n<sup>o</sup> 24, p. 105. L’auteure anonyme se présentait alors comme la fille
d’un militaire.';
    }
    elseif($id == 6481){
        $all_remark = 'Ce logogriphe a paru sous une forme sensiblement différente
dans le Mercure du 6 mai 1786, n<sup>o</sup> 18, p. 9-10. Son auteur était alors «
M. le Chevalier de Breuvery, Officier au Régiment de Provence ».';
    }
    elseif($id == 6482){
        $all_remark = 'L’auteure est « Madame A** », d’après une publication
antérieure du même texte dans le Mercure du 30 septembre 1786, n<sup>o</sup> 39,
p. 197.';
    }
    elseif($id == 6487){
        $all_remark = 'Ce logogriphe semble reprendre, en l’abrégeant, un texte
publié par « M. Saint-G**, Officier au Régiment de Picardie » dans le Mercure du
1<sup>er</sup> avril 1786, n<sup>o</sup> 13, p. 44-45.';
    }
    elseif($id == 6489){
        $all_remark = 'Cette énigme est très similaire à un texte de « M. le
Vicomte de Gal... » paru dans le Mercure du 10 juin 1786, n<sup>o</sup> 23, p. 54-
55.';
```

```
    }
    elseif($id == 6493){
        $all_remark = 'L’auteur est « M. Gillat, Etudiant en Droit des Facultés de
Rennes », d’après une publication antérieure du même texte dans le Mercure du 10
juin 1786, <sup>o</sup> 23, p. 55-56.';
    }
    elseif($id == 6494){
        $all_remark = 'L’auteur est « M. Robert des Roches », d’après une
publication antérieure du même texte dans le Mercure du 16 septembre 1786,
n<sup>o</sup> 37, p. 103.';
    }
    elseif($id == 6495){
        $all_remark = 'L’auteur est « Un membre de la Société littéraire de Goven
en Bretagne », d’après une publication antérieure (avec un vers de plus) du même
texte dans le Mercure du 16 décembre 1786, n<sup>o</sup> 50, p. 99.';
    }
    elseif($id == 6496){
        $all_remark = 'L’auteur est « M. Bouchet », d’après une publication
antérieure et presque identique du même texte dans le Mercure du 7 janvier 1786,
n<sup>o</sup> 1, p. 22.';
    }
    elseif($id == 6501){
        $all_remark = 'L’auteur est « M. S. D. B., Officier de Marine », d’après
une publication antérieure du même texte dans le Mercure du 18 mars 1786,
n<sup>o</sup> 11, p. 102.';
    }
    elseif($id == 6502){
        $all_remark = 'Ce texte a déjà paru dans le Mercure du 3 juin 1786,
n<sup>o</sup> 22, p. 8. Son auteur anonyme était alors « un Dragon du Régiment de
Noailles ».';
    }
    elseif($id == 6504){
        $all_remark = 'L’auteur est « M. Barraud », d’après une publication
antérieure du même texte dans le Mercure du 11 octobre 1783, n<sup>o</sup> 41, p.
55.';
    }
    elseif($id == 6507){
        $all_remark = 'L’auteur est « M. Bl..., d’Arras », d’après une publication
antérieure du même texte dans le Mercure du 4 octobre 1783, n<sup>o</sup> 40, p.
7-8.';
    }
    elseif($id == 6508){
        $all_remark = 'L’auteur est « M. Berthier, Officier au Rég[iment] de
Picardie », d’après une publication antérieure du même texte dans le Mercure du 16
août 1783, n<sup>o</sup> 33, p. 104-105.';
    }
    elseif($id == 6509){
        $all_remark = 'L’auteur est « M. Duclauzeau », d’après une publication
antérieure du même texte dans le Mercure du 22 avril 1786, n<sup>o</sup> 16, p.
173.';
    }
    elseif($id == 6512){
        $all_remark = 'L’auteur est « M. ***, de Beaumont-le-Vicomte », d’après
une publication antérieure du même texte dans le Mercure du 4 octobre 1783,
n<sup>o</sup> 40, p. 7.';
    }
    elseif($id == 6516){
        $all_remark = 'L’auteur est « C.... », d’après une publication antérieure
du même texte dans le Mercure du 8 décembre 1781, n<sup>o</sup> 49, p. 50.';
```

```
    }
    elseif($id == 6518){
        $all_remark = 'L’auteur est « M. de S. G. », d’après une publication
antérieure du même texte dans le Mercure du 27 mai 1786, n<sup>o</sup> 21, p.
149.';
    }
    elseif($id == 6519){
        $all_remark = 'Déjà parue dans le Mercure du 16 mars 1782, n<sup>o</sup>
11, p. 110, cette énigme portait alors, en guise de signature, les mots « Au
Palais Bourbon ».';
    }
    elseif($id == 6520){
        $all_remark = 'L’auteur est « M. de Châteaugiron, Cap[itaine] au
Rég[iment] de Normandie », d’après une publication antérieure du même texte dans
le Mercure du 1<sup>er</sup> février 1783, n<sup>o</sup> 5, p. 5.';
    }
    elseif($id == 6524){
        $all_remark = 'L’auteur est « M. Philosphinx », d’après une publication
antérieure du même texte dans le Mercure du 18 décembre 1784, n<sup>o</sup> 51, p.
119.';
    }
    elseif($id == 6526){
        $all_remark = 'Déjà paru dans le Mercure du 4 février 1786, n<sup>o</sup>
5, p. 9, ce texte était alors publié par un anonyme se présentant comme un commis
de la douane.';
    }
    elseif($id == 6527){
        $all_remark = 'Cette énigme est très similaire à texte de « M. le
Chevalier de Bergerac » paru dans le Mercure du 15 juin 1782, n<sup>o</sup> 24, p.
106.';
    }
    elseif($id == 6529){
        $all_remark = 'L’auteur est « M. Berthier, Officier au Rég[iment] de
Picardie », d’après une publication antérieure du même texte dans le Mercure du 18
décembre 1784, n<sup>o</sup> 51, p. 118.';
    }
    elseif($id == 6530){
        $all_remark = 'L’auteur est « M. de la Sablonnière, C. R. », d’après une
publication antérieure du même texte dans le Mercure du 19 février 1785,
n<sup>o</sup> 8, p. 101-102.';
    }
    elseif($id == 6531){
        $all_remark = 'L’auteur est « M. Gratton [de Saint-Gilles], Capitaine de
Canonniers », d’après une publication antérieure du même texte dans le Mercure du
18 décembre 1784, n<sup>o</sup> 51, p. 119.';
    }
    elseif($id == 6533){
        $all_remark = 'L’auteur est « M. N.... d’Arras », d’après une publication
antérieure du même texte dans le Mercure du 12 octobre 1782, n<sup>o</sup> 41, p.
52-53.';
    }
    elseif($id == 6534){
        $all_remark = 'L’auteur est « M. Conjon », d’après une publication
antérieure du même texte dans le Mercure du 9 juillet 1785, n<sup>o</sup> 28, p.
54.';
    }
    elseif($id == 6535){
```

```
        $all_remark = 'L’auteur est « M. M*** », d’après une publication
antérieure du même texte dans le Mercure du 12 octobre 1782, n<sup>o</sup> 41, p.
53-54.';
    }
    elseif($id == 6536){
        $all_remark = 'L’auteur est « M. Bouloux », d’après une publication
antérieure du même texte dans le Mercure du 25 juin 1785, n<sup>o</sup> 26, p.
150.';
    }
    elseif($id == 6538){
        $all_remark = 'Ce logogriphe est très similaire à un texte de « M. Lagache
fils, à Amiens » paru dans le Mercure du 31 août 1782, n<sup>o</sup> 35, p. 198.';
    }
    elseif($id == 6539){
        $all_remark = 'La charade est composée par « une Société de Saint-Rambert,
en Bugey », d’après une publication antérieure du même texte dans le Mercure du 16
juillet 1785, n<sup>o</sup> 29, p. 102.';
    }
    elseif($id == 6540){
        $all_remark = 'L’auteur est « M. André Honoré », d’après une publication
antérieure du même texte dans le Mercure du 8 janvier 1785, n<sup>o</sup> 2, p.
64.';
    }
    elseif($id == 6541){
        $all_remark = 'Cette charade a déjà paru dans le Mercure du 8 janvier
1785, n<sup>o</sup> 2, p. 63. L’auteur anonyme se présentait alors comme « un
Abonné d’Andely ».';
    }
    elseif($id == 6542){
        $all_remark = 'L’auteur est « M. M*** de V. », d’après une publication
antérieure du même texte dans le Mercure du 8 janvier 1785, n<sup>o</sup> 2, p.
64-65. Le texte était alors adressé à une demoiselle de 28 ans (et non de 30
ans).';
    }
    elseif($id == 6544){
        $all_remark = 'Cette énigme est très similaire à un texte de « M. l’Abbé
Lassagne, Vicaire de Meymac, en Limousin », paru dans le Mercure du 20 août 1785,
n<sup>o</sup> 34, p. 99-100.';
    }
    elseif($id == 6545){
        $all_remark = 'Ce logogriphe est très similaire à un texte de « M. R.....
de Narbonne, Ancien Capitaine d’Infanterie », paru dans le Mercure du 20 août
1785, n<sup>o</sup> 34, p. 101, mais il introduit de petites variantes.';
    }
    elseif($id == 6546){
        $all_remark = 'L’auteur est « M. l’Abbé de C..... », d’après une
publication antérieure du même texte dans le Mercure du 30 avril 1785,
n<sup>o</sup> 18, p. 199.';
    }
    elseif($id == 6547){
        $all_remark = 'L’auteur est « M. l’Abbé de C..... », d’après une
publication antérieure du même texte dans le Mercure du 30 avril 1785,
n<sup>o</sup> 18, p. 199-200.';
    }
    elseif($id == 6548){
        $all_remark = 'L’auteur est « M. l’Abbé de C..... », d’après une
publication antérieure du même texte dans le Mercure du 30 avril 1785,
n<sup>o</sup> 18, p. 200.';
    }
```

```
    elseif($id == 6552){
        $all_remark = 'Le dernier mot de la signature est illisible sur
l’exemplaire que nous avons consulté.';
    }
    elseif($id == 6561){
        $all_remark = 'L’auteur est « M. Robert des Roches, de Fontenay-le-Comte
», d’après une publication antérieure du même texte dans le Mercure du 23 avril
1785, n<sup>o</sup> 17, p. 150.';
    }
    elseif($id == 6662){
        $all_remark = 'Ce logogriphe est très similaire à un texte de « M.
Candelon » paru dans le Mercure de mai 1772, p. 56-57.';
    }
    elseif($id == 6675){
        $all_remark = 'Avec des variantes, cette énigme a déjà paru dans le
Mercure de janvier 1778, vol. 1, p. 62-63. Elle était alors signée « Par une Dlle
de Nogent-le-Roi ».';
    }
    elseif($id == 6684){
        $all_remark = 'L’auteur est « M. Lagache fils, à G. », d’après une
publication antérieure du même texte dans le Mercure du 8 juillet 1780,
n<sup>o</sup> 28, p. 59.';
    }
    elseif($id == 6685){
        $all_remark = 'L’auteur est « M. Lagache fils, d’Amiens », d’après une
publication antérieure du même texte dans le Mercure du 6 octobre 1792 (15
vendémiaire an I), n<sup>o</sup> 40, p. 7. M. Lagache fils signe par ailleurs la
poésie qui précède immédiatement cette charade-ci dans le présent numéro du
Mercure.';
    }
    elseif($id == 6686){
        $all_remark = 'L’auteur est « M. Lagache fils, Rec[eveur] des Dom[aines]
Nat[ionaux] », d’après une publication antérieure du même texte dans le Mercure du
28 avril 1792, n<sup>o</sup> 17, p. 87. M. Lagache fils signe par ailleurs la
poésie qui précède immédiatement la charade et le logogriphe dans le présent
numéro du Mercure.';
    }
    elseif($id == 6693){
        $all_remark = 'L’auteur est « M. Catimini, de Rouen », d’après une
publication antérieure du même texte dans le Mercure de juillet 1773, vol. 1, p.
73.';
    }
    elseif($id == 6713){
        $all_remark = 'L’auteur est « M. Catimini, de Rouen », d’après une
publication antérieure du même texte (avec une variante) dans le Mercure de
juillet 1773, vol. 1, p. 73.';
    }
    elseif($id == 6715){
        $all_remark = 'L’auteur est « M. Houllier de St Remi » (ou Saint-Rémy),
d’après une publication antérieure du même logogriphe dans le Mercure d’avril
1773, vol. 1, p. 67.';
    }
    elseif($id == 6727){
        $all_remark = 'Ce texte a déjà paru dans le Mercure de janvier 1774, vol.
2, p. 69. Il était alors signé « Par un Chapelain de Dourdan ».';
    }
    elseif($id == 6734){
```

```
        $all_remark = 'L’auteur est « M. Houllier de Saint Remy, à Sezanne », d’après une publication antérieure du même texte dans le Mercure d’octobre 1775, vol. 1, p. 86.';
    }
    elseif($id == 6735){
        $all_remark = 'L’auteur est « M. Houllier de Saint Remy, à Sezanne », d’après une publication antérieure du même texte dans le Mercure d’octobre 1775, vol. 2, p. 59.';
    }
    elseif($id == 6772){
        $all_remark = 'L’auteur est « M. Houllier de Saint Remy, à Sezanne », d’après une publication antérieure du même texte dans le Mercure d’octobre 1775, vol. 1, p. 85.';
    }
    elseif($id == 6795){
        $all_remark = 'Ce logogriphe s’inspire manifestement d’un texte du « cit[oyen] du T**., de la ville de Caen », paru dans le Mercure du 30 novembre 1796 (10 frimaire an V), n<sup>o</sup> 7, p. 40-41.';
    }
    elseif($id == 6798){
        $all_remark = 'Ce logogriphe s’inspire manifestement d’un texte de « M. L. C... de B..., à la Source de la Dive », paru dans le Mercure du 22 mars 1783, n<sup>o</sup> 12, p. 147-148.';
    }
    elseif($id == 6804){
        $all_remark = 'L’auteur est « M. Houssaye, d’Evreux », d’après une publication antérieure du même texte dans le Mercure du 11 février 1792, n<sup>o</sup> 6, p. 41.';
    }
    elseif($id == 6816){
        $all_remark = 'L’auteur est « M. Gar.... le jeune, d’Auxerre », d’après une publication antérieure du même texte dans le Mercure d’octobre 1771, vol. 1, p. 65-66.';
    }
    elseif($id == 6827){
        $all_remark = 'La solution au logogriphe proposée dans le numéro 14 du 8 février 1799, p. 104, est « Étable », mais il s’agit d’une erreur que corrige le numéro 15 du 18 février 1799, p. 168, pour donner la réponse de « Lièvre ».';
    }
    elseif($id == 6832){
        $all_remark = 'L’auteur est « M. Houllier de S. Remi » (ou Saint-Rémy), d’après une publication antérieure du même texte dans <em>La Suite de la clef, ou journal historique sur les matières du tems</em>, Paris, Ruault, février 1774, t. 115, p. 137. C’est aussi dans ce périodique (mars 1774, p. 213) que nous trouvons la réponse « Étable ».';
    }
    elseif($id == 6839){
        $all_remark = 'Dans l’exemplaire consulté du Mercure de France suivant (numéro 4), les pages consacrées aux nouvelles énigmes et à l’explication des énigmes précédentes semblent manquantes. Nous trouvons en revanche la réponse de ce texte dans le recueil d’Hilaire Le Gai, <em>Un million d’énigmes, charades et logogriphes suivi d’un choix des plus jolies énigmes italiennes, espagnoles, anglaises et allemandes avec la traduction en regard</em>, Paris, Passard, 1850, qui donne ce texte à la page 355.';
    }
    elseif($id == 6840){
        $all_remark = 'Dans l’exemplaire consulté du Mercure de France suivant (numéro 4), les pages consacrées aux nouvelles énigmes et à l’explication des énigmes précédentes semblent manquantes. Nous trouvons en revanche la réponse de
```

```
ce texte dans <em>Les Affiches, annonces judiciaires, avis divers du Mans, et du département de la Sarthe</em>, Le Mans, Monnoyer, 69<sup>e</sup> année, 4 janvier 1839, n<sup>o</sup> 2, p. 15.';
    }
    elseif($id == 6845){
        $all_remark = 'À défaut d'avoir pu consulter le numéro suivant du Mercure (numéro 7), où se trouve probablement la réponse de ce texte, nous trouvons cette réponse dans le recueil d'Hilaire Le Gai, <em>Un million d'énigmes, charades et logogriphes suivi d'un choix des plus jolies énigmes italiennes, espagnoles, anglaises et allemandes avec la traduction en regard</em>, Paris, Passard, 1850, qui donne ce texte à la page 267 (et la réponse à la page 532).';
    }
    elseif($id == 6846){
        $all_remark = 'À défaut d'avoir pu consulter le numéro suivant du Mercure (numéro 7), où se trouve probablement la réponse de ce texte, nous trouvons cette réponse dans un autre périodique : <em>Les Affiches, annonces judiciaires, avis divers du Mans, et du département de la Sarthe</em>, Le Mans, Monnoyer, 68<sup>e</sup> année, 15 juin 1838, n<sup>o</sup> 48, p. 392.';
    }
    elseif($id == 6847){
        $all_remark = 'À défaut d'avoir pu consulter le numéro suivant du Mercure (numéro 7), où se trouve probablement la réponse de ce texte, nous trouvons cette réponse dans un autre périodique : <em>Les Affiches, annonces judiciaires, avis divers du Mans, et du département de la Sarthe</em>, Le Mans, Monnoyer, 68<sup>e</sup> année, 11 mai 1838, n<sup>o</sup> 38, p. 304.';
    }
    elseif($id == 6850){
        $all_remark = 'L'auteur est « M. Pillet, Lyonnois », d'après une publication antérieure du même texte dans le Mercure du 22 août 1789, n<sup>o</sup> 34, p. 101.';
    }
    elseif($id == 6870){
        $all_remark = 'L'auteur est « M. H..... », d'après une publication antérieure du même texte dans le Mercure du 6 août 1785, n<sup>o</sup> 32, p. 8.';
    }
    elseif($id == 6883){
        $all_remark = 'L'auteur est « M. L... de Moncy », d'après une publication antérieure du même texte dans le Mercure du 5 septembre 1789, n<sup>o</sup> 36, p. 7.';
    }


    $all_remark = str_replace(' Mercure galant', ' <em>Mercure galant</em>', $all_remark);
    $all_remark = str_replace(' Mercure français', ' <em>Mercure français</em>', $all_remark);
    $all_remark = str_replace(' Mercure de France', ' <em>Mercure de France</em>', $all_remark);
    $all_remark = str_replace(' Mercure', ' <em>Mercure</em>', $all_remark);
    $all_remark = str_replace('---Mercure---', 'Mercure', $all_remark);
    $all_remark = str_replace(' Extraordinaire', ' <em>Extraordinaire</em>', $all_remark);

    $all_remark = preg_replace('/(?<=\S)"/', ' »', $all_remark);
    $all_remark = preg_replace('/"(?=\S)/', '« ', $all_remark);

    $remove_note = array(86);

    if(in_array($tex_enigma_id, $remove_note)){
```

```
        $all_remark = '';
    }

    //Manage rich text field.

    if($all_remark != ''){

        $all_remark = preg_replace(
            '/"(\S)/',
            '« $1',
            $all_remark
        );

        $all_remark = preg_replace(
            '/(\S)"/',
            '$1 »',
            $all_remark
        );

        $all_remark = str_replace("’", "'", $all_remark);
        $all_remark = str_replace("&rsquo;", "'", $all_remark);
        $all_remark = '"{ format: full_html, value: ""<p>'.$all_remark.'</p>""
}"';
        $all_remark = htmlspecialchars($all_remark, ENT_QUOTES, 'UTF-8');
    }

    //Create field all_remark_private.

    $all_remark_private = "Les données relatives à ce texte proviennent du site
<em>Les énigmes de Mercure</em>. Elles ont été enrichies et importées en suivant
la procédure d'importation des données externes correspondante.</p>
    <p>Modifications et vérifications à effectuer par le collaborateur :</p>
    <p><strong>Données de base</strong></p>
    <ul>
    <li>Compléter au besoin le champ « Fait partie d'une section » ;</li>
    <li>Corriger au besoin le champ « Position sur la page » ;</li>
    <li>Corriger au besoin les champs « Page de début dans la numérisation » et «
Page de fin dans la numérisation » ;</li>
    <li>Copier et coller le texte océrisé dans le champ « Texte ».</li>
    </ul>
    <p><strong>Provenance</strong></p>
    <ul>
    <li>Si le texte est signé, compléter le champ « Signature ».</li>
    </ul>
    <p><strong>Langue et genre</strong></p>
    <p>[Aucune modification.]</p>
    <p><strong>Autres relations, titre dans la table des matières</strong></p>
    <ul>
    <li>Si le texte a un titre dans la table des matières, compléter le champ «
Titre d'après la table ».</li>
    </ul>
    <p><strong>Remarques et validité</strong></p>
    <ul>
    <li>Laisser une « Remarque privée » ; si les modifications demandées ont été
réalisées sans difficulté, écrire simplement « Complété. » avec les initiales du
collaborateur et la date ;</li>
    <li>Indiquer dans le champ « Validité » la valeur « En attente de validation
».</li>
    </ul>
```

```
    <p><strong>N.B.</strong> Si des erreurs ou des lacunes apparaissent dans
d'autres champs, merci de faire les modifications qui s'imposent ou de laisser une
remarque privée, en maintenant le statut « À compléter ». (TL, 15 octobre 2024.)";

    //Manage rich text field.

    if($all_remark_private != ''){

        $all_remark_private = preg_replace(
            '/"(\S)/',
            '« $1',
            $all_remark_private
        );

        $all_remark_private = preg_replace(
            '/(\S)"/',
            '$1 »',
            $all_remark_private
        );

        $all_remark_private = str_replace("’", "'", $all_remark_private);
        $all_remark_private = str_replace("&rsquo;", "'", $all_remark_private);
        $all_remark_private = '"{ format: full_html, value:
""<p>'.$all_remark_private.'</p>"" }"';
        $all_remark_private = htmlspecialchars($all_remark_private, ENT_QUOTES,
'UTF-8');
    }

    //Create field all_validity.

    $all_validity = 82; //Value matching "À compléter" in the "Validité" taxonomy.

    //Combine the fields in CSV format.

    if(in_array($tex_enigma_id, $remove_array)){
    }else{

        if(($tmpCount >= $start_row) && ($tmpCount <= $end_row)){
            $output .=
$tex_title.','.$tex_title_table.','.$tex_rec_article_short_title.','.$tex_incipit.
','.$all_start_page.','.$all_start_page_file.','.$all_end_page.','.$all_end_page_f
ile.','.$tex_position_on_page.','.$tex_ocr.','.$tex_keyword.','.$tex_domain.','.$t
ex_signature.','.$tex_author_name.','.$tex_author_activity.','.$tex_author_title_o
f_nobility.','.$tex_author_gender.','.$tex_author_collectivity.','.$tex_place.','.
$all_date_gregorian_calendar.','.$all_date_republican_calendar.','.$tex_language.'
','.$tex_paratextual_genre.','.$tex_type_of_journalistic_writing.','.$tex_readers_l
etters.','.$tex_enigma_id.','.$tex_enigma_issue_date.','.$tex_enigma_issue_number.
','.$tex_enigma_standard_title.','.$tex_enigma_genre_one.','.$tex_enigma_genre_two
.','.$tex_enigma_standard_incipit.','.$tex_enigma_place_geonames.','.$tex_enigma_p
lace_latitude.','.$tex_enigma_place_longitude.','.$tex_enigma_date.','.$tex_enigma
_standard_author_name.','.$tex_enig_stand_author_occupation.','.$tex_enigma_answer
.','.$tex_enigma_standard_answer.','.$tex_enigma_digital_copy_url.','.$tex_enigma_
digital_copy_iframe.','.$tex_enigma_remark.','.$tex_verse_prose.','.$tex_literary_
genre.','.$tex_is_part_of_iss.','.$tex_is_part_of_sec.','.$tex_is_reproduction_of_
tex.','.$tex_is_continuation_of_tex.','.$tex_is_response_to_tex.','.$tex_is_writte
n_by_per.','.$tex_is_probably_written_by_per.','.$tex_is_addressed_to_per.','.$tex
_is_probably_addressed_to_per.','.$tex_concerns_wor.','.$tex_probably_concerns_wor
.','.$tex_comes_from_pla.','.$tex_probably_comes_from_pla.','.$tex_is_part_of_fol.
','.$all_remark.','.$all_remark_private.','.$all_validity.'<br/>';
```

```
            $total_displayed ++;

        }

    }

}


//Echo the number of results.

echo 'Nombre de résultats : '.$total_displayed.'<br/>';
echo 'Affichage des résultats '.$start_row.' à '.$end_row.'<br/><br/>';

//Output in CSV format.

echo
'tex_title,tex_title_table,tex_rec_article_short_title,tex_incipit,all_start_page,
all_start_page_file,all_end_page,all_end_page_file,tex_position_on_page,tex_ocr,te
x_keyword,tex_domain,tex_signature,tex_author_name,tex_author_activity,tex_author_
title_of_nobility,tex_author_gender,tex_author_collectivity,tex_place,all_date_gre
gorian_calendar,all_date_republican_calendar,tex_language,tex_paratextual_genre,te
x_type_of_journalistic_writing,tex_readers_letters,tex_enigma_id,tex_enigma_issue_
date,tex_enigma_issue_number,tex_enigma_standard_title,tex_enigma_genre_one,tex_en
igma_genre_two,tex_enigma_standard_incipit,tex_enigma_place_geonames,tex_enigma_pl
ace_latitude,tex_enigma_place_longitude,tex_enigma_date,tex_enigma_standard_author
_name,tex_enig_stan_author_occupation,tex_enigma_answer,tex_enigma_standard_answer
,tex_enigma_digital_copy_url,tex_enigma_digital_copy_iframe,tex_enigma_remark,tex_
verse_prose,tex_literary_genre,tex_is_part_of_iss,tex_is_part_of_sec,tex_is_reprod
uction_of_tex,tex_is_continuation_of_tex,tex_is_response_to_tex,tex_is_written_by_
per,tex_is_probably_written_by_per,tex_is_addressed_to_per,tex_is_probably_address
ed_to_per,tex_concerns_wor,tex_probably_concerns_wor,tex_comes_from_pla,tex_probab
ly_comes_from_pla,tex_is_part_of_fol,all_remark,all_remark_private,all_validity<br
/>';
echo $output;
```

h. Copier les données depuis un navigateur web et coller l'ensemble dans un nouveau fichier CSV vide.

4. Importer les données dans *Drupal*, grâce à l'outil d'importation du module Webform.
5. Des modifications doivent être faites manuellement après l'importation. Premièrement, il s'agit de compléter des données manquantes, à savoir la section (champ « tex_is_part_of_sec »), la reconnaissance textuelle (« tex_ocr »), la signature (« tex_signature ») et le titre d'après la table (« tex_title_table »). Deuxièmement, les autres champs importés doivent être vérifiés et corrigés au besoin, en particulier la position sur la page (champ « tex_position_on_page ») et les pages de début et de fin dans la numérisation (« all_start_page_file » et « all_end_page_file »). Troisièmement, les relations entre énigmes doivent être établies après l'importation, quand le texte constitue la republication d'un texte (« tex_is_reproduction_of_tex »), la suite d'un autre texte (« tex_is_continuation_of_tex ») ou la réponse à un autre texte (« tex_is_response_to_tex »).

Reconnaissance textuelle et balisage sur la plateforme *Transkribus*

DONNÉES RELATIVES AUX TEXTES

Objectifs

La plateforme *Transkribus* (https://www.transkribus.org) permet de reconnaître le texte des livraisons du *Mercure* à partir d'une numérisation en format PDF, d'améliorer cette transcription et d'y intégrer certaines données en balisant les articles et les poésies concernés. La transcription balisée est exportée en format XML. Nous la retouchons et nous récupérons sous la forme d'un fichier CSV les données pertinentes, dans la perspective de les importer dans la base de données Drupal. Cette procédure fournit une reconnaissance textuelle souvent meilleure que celle des grandes bibliothèques numériques, accélère l'indexation des livraisons et permet au besoin de disposer d'une indexation provisoire, en attendant de compléter les données sur notre propre plateforme.

Données concernées

En balisant les transcriptions, nous générons les données suivantes (voir les ontologies) :
- Fait partie d'une livraison (tex_is_part_of_iss).
- Fait partie d'une section (tex_is_part_of_sec).
- Page de début dans la numérisation (all_start_page_file).
- Page de début (all_start_page).
- Page de fin dans la numérisation (all_end_page_file).
- Page de fin (all_end_page).
- Position sur la page (tex_position_on_page).
- Titre (tex_title).
- Titre d'après la table (tex_title_table).
- Incipit (tex_incipit).
- Signature (tex_signature).
- Genre (tex_author_gender).
- Collectivité (tex_author_collectivity).
- Langue (tex_language).
- Domaine (tex_domain).
- Vers et prose (tex_verse_prose).
- Genre paratextuel (tex_paratextual_genre).
- Type d'écrit journalistique (tex_type_of_journalistic_writing).
- Courrier des lecteurs (tex_readers_letters).
- Texte (tex_ocr).
- Source externe (tex_ocr_external_source).
- Fait partie d'un dossier (tex_is_part_of_fol).
- Remarque (all_remark).
- Remarque privée (all_remark_private).
- Validité (all_validity).

Utilisation

La procédure présuppose un compte payant sur *Transkribus* pour générer une reconnaissance textuelle sur de grands lots de pages et pour exporter la transcription en format XML-TEI [2].

[2] Nous pouvons aussi travailler à partir de la reconnaissance textuelle de *Google Books* ou de *Gallica*, par exemple, et utiliser ainsi *Transkribus* sans frais. Nous avons développé un programme qui permet d'extraire la

À partir de la reconnaissance textuelle générée par *Transkribus*, les données sont produites de différentes façons. Sur l'interface de *Transkribus*, nous plaçons des balises <tex_start> et <tex_end> qui marquent le début et la fin de chaque texte. Nous balisons ensuite des segments qui deviendront des données textuelles dans la base de données Drupal (texte enrichi avec balises HTML) : le titre, l'*incipit*, la signature et d'éventuelles remarques publiques ou privées. Le texte à proprement parler (tex_ocr) est généré d'après les balises <tex_start> et <tex_end>. Il inclut les segments retouchés, mais nous ne corrigeons pas le reste de la transcription (le corps du texte). Les pages de début et de fin dans le fichier PDF (all_start_page_file, all_end_page_file) sont captées automatiquement. En revanche, nous transcrivons manuellement et balisons le titre d'après la table pour permettre son importation en même temps que les autres données du texte. De la même façon, nous pouvons ajouter des remarques publiques ou privées sur le texte. Enfin, nous injectons des données dans les attributs des balises <tex_start> et <tex_end> : l'identifiant de la livraison à laquelle appartient le texte, l'identifiant de son éventuelle section, ses pages de début et de fin dans la livraison, sa position sur la page, son ou ses domaines, sa forme (vers ou prose), son ou ses genres paratextuels, son ou ses types d'écrit journalistique, sa langue, son appartenance au courrier des lecteurs, le ou les genres de ses auteurs, la mention d'auteur collectif et le ou les dossiers auxquels le texte se rattache.

Une fois importées dans la base Drupal, ces données fournissent un premier aperçu des sommaires et des fiches de textes. Celles-ci doivent ensuite être vérifiées et complétées manuellement, sur l'interface de Drupal, au moyen de notre formulaire de saisie habituel. Cette procédure ne tient pas compte des illustrations qui doivent être indexées séparément.

Procédure détaillée

1. Sur l'interface de Drupal, créer en premier lieu les fiches des sections de la ou des livraisons à indexer.
2. Sur l'interface de *Transkribus*, créer un projet et une collection.
3. Charger dans cette collection le fichier PDF de la ou des livraisons concernées.
4. Au niveau des collections, dans « Settings », accéder à « Tag settings » et créer les étiquettes suivantes : tex_start, tex_end, tex_title, tex_incipit, tex_signature, tex_title_table, all_remark et all_remark_private.
5. Modifier tex_start pour y ajouter les attributs suivants : tex_is_part_of_iss__target_id, tex_is_part_of_sec__target_id, all_start_page, tex_position_on_page, tex_domain, tex_verse_prose, tex_paratextual_genre, tex_type_of_journalistic_writing, tex_readers_letters, tex_author_gender, tex_author_collectivity, tex_language et tex_is_part_of_fol.
6. Modifier tex_end pour y ajouter les attributs suivants : all_end_page, tex_position_on_page, tex_domain, tex_verse_prose, tex_paratextual_genre, tex_type_of_journalistic_writing, tex_readers_letters, tex_author_gender, tex_author_collectivity, tex_language et tex_is_part_of_fol.
7. Ouvrir le PDF, sélectionner les pages à indexer et cliquer sur « Process with AI ».
8. Choisir « Text Recognition » et sélectionner un modèle de reconnaissance textuelle. Nous utilisons « Transkribus Print M1 ».
9. Ouvrir le PDF sur l'interface, sélectionner la première page à indexer et commencer le balisage.
    a. Ajouter n'importe quel caractère juste avant le début du texte pour le baliser avec tex_start. Dans les attributs de tex_start, donner les propriétés suivantes :

reconnaissance textuelle des PDF téléchargés et de l'importer dans *Transkribus*, mais cette procédure alternative n'est pas documentée ici.

i. tex_is_part_of_iss__target_id : l'identifiant Drupal de la livraison. Il suffit de renseigner cet identifiant dans le tout premier texte de la livraison à indexer.
ii. tex_is_part_of_sec__target_id : le cas échéant, l'identifiant Drupal de la section à laquelle appartient le texte.
iii. all_start_page : le numéro de page du début du texte dans la livraison, conformément aux ontologies.
iv. tex_position_on_page : la position sur la page, conformément aux ontologies.
v. tex_domain : le cas échéant, le label du domaine du texte, conformément au référentiel concerné (par exemple, « Sciences politiques » ou « Histoire naturelle »). Accepte plusieurs valeurs, séparées par une virgule.
vi. tex_verse_prose : préciser « Vers » ou « Vers et prose », le cas échéant. La valeur par défaut est « Prose ».
vii. tex_paratextual_genre : le cas échéant, le label du genre paratextuel, conformément au référentiel concerné (par exemple, « Approbation » ou « Errata »). Accepte plusieurs valeurs, séparées par une virgule.
viii. tex_type_of_journalistic_writing : le cas échéant, le label du type d'écrit journalistique, conformément au référentiel concerné (par exemple, « Article / Nouvelle littéraire » ou « Poésie »). Accepte plusieurs valeurs, séparées par une virgule.
ix. tex_readers_letters : indiquer « Oui » si le texte appartient au courrier des lecteurs. Valeur par défaut : « Non ».
x. tex_author_gender : si le genre de l'auteur est connu, indiquer « Homme » ou « Femme ». Accepte plusieurs valeurs, séparées par une virgule. Valeur par défaut : « Indéterminé ».
xi. tex_author_collectivity : si le texte est composé par une collectivité, indiquer « Oui ». Valeur par défaut : « Non ».
xii. tex_language : la langue du texte, conformément au référentiel concerné. Valeur par défaut : « Français ».
xiii. tex_is_part_of_fol : le cas échéant, l'identifiant Drupal du ou des dossiers auxquels nous souhaitons rattacher le texte.

b. Identifier un éventuel titre, baliser les passages à mettre en italique et en exposant, puis baliser tout le segment avec tex_title.
c. Répéter ces opérations pour l'*incipit*, avec la balise tex_incipit.
d. Répéter ces opérations pour une éventuelle signature, avec la balise tex_signature.
e. Ajouter n'importe quel caractère juste après la fin du texte pour le baliser avec tex_end. Dans les attributs de tex_end, donner les propriétés suivantes :
    i. all_end_page : le numéro de page de la fin du texte dans la livraison, conformément aux ontologies.
    ii. Renseigner au besoin les autres attributs, décrits ci-dessus, s'ils n'ont pas été renseignés parmi les attributs de tex_start, ou seulement de façon partielle. Par exemple, il est possible d'ajouter dans l'attribut tex_domain de la balise tex_end un second domaine.
f. Quelque part entre les balises tex_start et tex_end, transcrire l'éventuel titre d'après la table. Baliser les passages à mettre en italique et en exposant, puis baliser tout le segment avec tex_title_table.

g. Répéter cette opération pour une éventuelle remarque publique ou privée, en balisant le texte ajouté avec all_remark ou all_remark_private.

10. Répéter ces opérations pour tous les textes de la ou des livraisons à indexer.
11. Retourner à la vue d'ensemble du fichier PDF, sélectionner toutes les pages concernées et cliquer sur « Export ».
12. Sélectionner le format « TEI Export », avec les paramètres suivants : « Export all tags in document », *stylesheet* « Standard » et *line breaks* « Line tags (<l>…<l>) ». Décocher les autres paramètres, notamment « Include zones ». Cliquer sur « Start export ». À noter : nous utilisons le format TEI uniquement comme format de transition, sans respecter le standard TEI. *Transkribus* va produire un fichier XML qui devrait ressembler à ceci (fichier « 1752,_03.xml ») :

```
<?xml version='1.0' encoding='UTF-8'?>
<TEI xmlns='http://www.tei-c.org/ns/1.0'>
    <teiHeader>
        <fileDesc>
            <titleStmt>
                <title type='main'>1752, 03</title>
            </titleStmt>
            <publicationStmt>
                <publisher>tranScriptorium</publisher>
            </publicationStmt>
            <sourceDesc>
                <bibl><publisher>TRP document creator:
timothee.lechot@gmail.com</publisher></bibl>
            </sourceDesc>
        </fileDesc>
    </teiHeader>
    <text>
        <body>
            <div>
                <pb n='7'/>
                <p>
                    <lg>
                        <l></l>
                        <l><tex_start tex_paratextual_genre='Avertissement'
all_start_page='s. p.' tex_is_part_of_sec__target_id='2567702'
tex_is_part_of_iss__target_id='1383566'>s</tex_start></l>
                        <l><tex_title>AVIS.</tex_title></l>
                        <l><tex_incipit><hi rend='italic:true;'>L&apos;ADRESSE du
Mercure est à M.</hi> MERIEN,</tex_incipit></l>
                        <l><hi rend='italic:true;'><tex_incipit>Commis au Mercure,
rue de l&apos;Echelle Saint Honoré,</tex_incipit></hi></l>
                        <l>à Hotel de la Roche-sur-Yon, pour remettre à</l>
                        <l>M. l&apos;Abbé Raynal.</l>
                        <l>Nous prions très-instamment ceux qui nous adresse-</l>
                        <l>ront des Paquets par la Poste, d&apos;en affranchir le
port,</l>
                        <l>pour nous épargner le déplaisir de les rebuter, &amp; à
eux</l>
                        <l>celui de ne pas voir paroître leurs Ouvrages.</l>
                        <l>Les Libraires des Provinces ou des Pays Etrangers,</l>
                        <l>qui souhaiteront avoir le Mercure de France de la pre-
</l>
                        <l>miere main, &amp; plus promptement, n&apos;auront
qu&apos;à</l>
                        <l>écrire à l&apos;adresse ci-dessus indiquée.</l>
```

```
                        <l>On l&apos;envoye aussi par la Poste, aux personnes de
Pro-</l>
                        <l>vince qui le desirent, les frais de la poste ne sont
pas</l>
                        <l>considerables.</l>
                        <l>On avertit aussi que ceux qui voudront qu&apos;on le
porte</l>
                        <l>chez eux à Paris chaque mois, n&apos;ont qu&apos;à
faire sçavoir</l>
                        <l>leurs intentions, leur nom &amp; leur demeure audii
sieur</l>
                        <l>Merien, Commis au Mercure ; on lour portera le
Mercure</l>
                        <l>très-exactement, moyennant 21 livres par an,
qu&apos;ils</l>
                        <l>payeront, sçavoir, 10 liv. 10 s. en recevant le
second</l>
                        <l>volume de Juin, &amp; 10 l. 10 s. en recevant le
second</l>
                        <l>volume de Décembre. On les supplie instamment de</l>
                        <l>d nner leurs ordres pour que ces payemens soient
faits</l>
                        <l>dans leurs tems.</l>
                        <l>On prie aussi les personnes de Province, à qui on</l>
                        <l>envoye le Mercure par la Poste, d&apos;être exactes a
faire</l>
                        <l>payer au Bureau du Mercure à la fin de chaque sémes-
</l>
                        <l>tre, sans cela on seroit hors d&apos;état de soutenir
les</l>
                        <l>avances considérables qu&apos;exige l&apos;impression
de cet</l>
                        <l>ouvrage.</l>
                        <l>On adresse la même priere auxe Libraires de
Province.</l>
                        <l>Les personnes qui voudront d&apos;autres Mercures
que</l>
                        <l>eux du mois courant, les trouveront chez la veuve</l>
                        <l>Pissot, Quai de Conti.</l>
                        <l>PRIX XXX. SOLS.<tex_end all_end_page='s.
p.'>f</tex_end></l>
                        <l></l>
                        <l></l>
                        <l>Digitized by Google</l>
                    </lg>
                </p>
                <pb n='8'/>
                <p>
                    <lg>
                        <l>—</l>
                        <l></l>
                        <l></l>
                        <l>MERCURE</l>
                        <l>DE FRANCE,</l>
                        <l>DÉDIÉ AU ROI.</l>
                        <l>MARS. 1752.</l>
                        <l></l>
                        <l>PIECES FUGITIVES,</l>
                        <l>en Vers &amp; en Prose.</l>
```

```
                        <l><tex_start all_start_page='3'
tex_is_part_of_sec__target_id='2567702' tex_verse_prose='Vers'
tex_domain='Sciences politiques' tex_type_of_journalistic_writing='Poésie'>—
</tex_start></l>
                        <l><tex_title>EPITRE</tex_title></l>
                        <l><hi rend='italic:true;'><tex_title>A Monsieur le
Marquis de Crillon, Député</tex_title></hi></l>
                        <l><hi rend='italic:true;'><tex_title>de la Ville
d&apos;Avignon pour complimenter</tex_title></hi></l>
                        <l><hi rend='italic:true;'><tex_title>le Roi sur la
Naissance de M. le Duc de</tex_title></hi></l>
                        <l><hi
rend='italic:true;'><tex_title>Bourgogne.</tex_title></hi> <tex_title_table>Epître
à Monsieur le Marquis de Crillon,</tex_title_table></l>
                        <l><tex_incipit>Heritier d&apos;un nom
Héroïque</tex_incipit></l>
                        <l>Dont vous relevez la splendeur</l>
                        <l></l>
                        <l>Dans huit jours, la chose est publique</l>
                        <l>Traîné dans un char magnifique</l>
                        <l>Vous irez brillant Orateur,</l>
                        <l>Sur le ton du Panégyrique</l>
                        <l>Etaler votre Rhétorique</l>
                        <l>A ij</l>
                        <l>Digitized by Google</l>
                    </lg>
                </p>
                <pb n='9'/>
                <p>
                    <lg>
                        <l>4 MERCURE DEFRANCE.</l>
                        <l>Dans le pays de la Grandeur:</l>
                        <l>Du Peuple le plus pacisique</l>
                        <l>..</l>
                        <l>O le plus brave Ambassadeur!</l>
                        <l>C&apos;est bien l&apos;entendre, Monseigneur;</l>
                        <l>Lorsque la paix vous fait défense</l>
                        <l>De signaler votre grand coeur</l>
                        <l>Votre ame ardente s&apos;en offense</l>

                        [...]

                        <l>Beaux-Arts,</l>
                        <l></l>
                        <l>Spectacles.</l>
                        <l>190</l>
                        <l>Nouvelles Etrangeres</l>
                        <l>200</l>
                        <l>France, nouvelles de la Cour, de Paris,</l>
                        <l>Description des Fêtes de Versailles.</l>
                        <l>207<tex_end all_end_page='216'>e</tex_end></l>
                        <l><tex_start tex_paratextual_genre='Emplacement
d&apos;une figure' all_start_page='216'>s</tex_start><tex_incipit><hi
rend='italic:true;'>La Chanson notée doit regarder la page</hi>
168</tex_incipit><tex_end all_end_page='216'>e</tex_end></l>
                        <l><tex_start tex_paratextual_genre='Colophon'
all_start_page='216'>s</tex_start><tex_incipit>De l&apos;Imprimerie de J.
BULLOT.</tex_incipit><tex_end all_end_page='216'>e</tex_end></l>
                        <l>Digitized by Google</l>
```

```
                    </lg>
                </p>
            </div>
        </body>
    </text>
</TEI>
```

13. À partir de ce fichier XML, nous pouvons générer un fichier CSV au moyen du code suivant (fichier « csv_transkribus_text.php ») :

```php
<?php
//Texts: create a CSV file from a transcription by Transkribus.


//Choose an XML-TEI file.

$file = '1752,_03.xml'; //Example: 1752,_03.xml
$xml = file_get_contents($file);

//OCR source.

$ocr_source = NULL; //The source of the OCR (tex_ocr_external_source). Value
matching the source in the "Source de la reconnaissance textuelle" taxonomy of the
Drupal database. Example for "Transkribus (Transkribus Print M1)": 29826

$total_displayed = 0;

$output_csv = '';

//Global corrections.

$xml = str_replace('¬', '-', $xml);
$xml = preg_replace('/<l\b[^>]*>\s*<\/l>/', '', $xml);

preg_match_all('/<l\b[^>]*>.*?<\/l>/si', $xml, $matches);

foreach ($matches[0] as $line) {
    if (
        strpos($line, 'Digitize') !== false ||
        strpos($line, 'Googl') !== false
    ) {
        $xml = str_replace($line, '', $xml);
    }
}

//Style tags within structural tags.

$xml = preg_replace(
    '/<hi([^>]*)><tex_title>/i',
    '<tex_title><hi$1>',
    $xml
);

$xml = preg_replace(
    '/<\/tex_title><\/hi>/i',
    '</hi></tex_title>',
    $xml
);
$xml = preg_replace(
```

```
    '/<hi([^>]*)><tex_title_table>/i',
    '<tex_title_table><hi$1>',
    $xml
);

$xml = preg_replace(
    '/<\/tex_title_table><\/hi>/i',
    '</hi></tex_title_table>',
    $xml
);
$xml = preg_replace(
    '/<hi([^>]*)><tex_incipit>/i',
    '<tex_incipit><hi$1>',
    $xml
);

$xml = preg_replace(
    '/<\/tex_incipit><\/hi>/i',
    '</hi></tex_incipit>',
    $xml
);
$xml = preg_replace(
    '/<hi([^>]*)><tex_signature>/i',
    '<tex_signature><hi$1>',
    $xml
);

$xml = preg_replace(
    '/<\/tex_signature><\/hi>/i',
    '</hi></tex_signature>',
    $xml
);

function mergeFormattingTags($text) {

    //Two consecutive italic and superscript segments.
    $text = preg_replace(
        '#</sup></em>(\s*)<em><sup>#i',
        '$1',
        $text
    );

    //Two consecutive italic segments.
    $text = preg_replace(
        '#</em>(\s*)<em>#i',
        '$1',
        $text
    );

    //Two consecutive superscript segments.
    $text = preg_replace(
        '#</sup>(\s*)<sup>#i',
        '$1',
        $text
    );

    return $text;
}
```

```php
//Create the HTML tags.

$xml = preg_replace_callback(
    '/<hi\s+rend=[\'"]([^\'"]*)[\'"]>|<\/hi>/i',
    function ($matches) {

        static $stack = [];

        if ($matches[0] === '</hi>') {

            $close = array_pop($stack);

            return $close ?? '';
        }

        $rend = strtolower($matches[1]);

        if (
            strpos($rend, 'italic:true') !== false &&
            strpos($rend, 'superscript:true') !== false
        ) {
            $stack[] = '</sup></em>';
            return '<em><sup>';
        }

        if (strpos($rend, 'italic:true') !== false) {
            $stack[] = '</em>';
            return '<em>';
        }

        if (strpos($rend, 'superscript:true') !== false) {
            $stack[] = '</sup>';
            return '<sup>';
        }

        $stack[] = '';

        return '';
    },
    $xml
);

//Line breaks.

$xml = str_ireplace('<l>', '', $xml);
$xml = str_ireplace('</l>', '<br>', $xml);

//Complete segments.

preg_match_all(
    '/<tex_start\b[^>]*>.*?<\/tex_end>/si',
    $xml,
    $matches,
    PREG_OFFSET_CAPTURE
);

//Pages in the file.

preg_match_all(
```

```
    '/<pb\b[^>]*n=["\']([^"\']+)["\'][^>]*\/>/i',
    $xml,
    $pbs,
    PREG_OFFSET_CAPTURE
);

//Create text data from the XML-TEI file.

$tex_is_part_of_iss = null;

foreach ($matches[0] as $match) {

    $text = $match[0];

    $startPos = $match[1];
    $endPos = $startPos + strlen($text);

    //Pages in the file (all_start_page_file, all_end_page_file).

    $all_start_page_file = null;
    $all_end_page_file = null;

    foreach ($pbs[1] as $pb) {

        if ($pb[1] < $startPos) {
            $all_start_page_file = $pb[0];
        }

        if ($pb[1] < $endPos) {
            $all_end_page_file = $pb[0];
        } else {
            break;
        }
    }

    //Pages in the periodical (all_start_page, all_end_page).

    if (preg_match("/all_start_page='([^']*)'/", $text, $m)) {
        $all_start_page = $m[1];
    } else {
        $all_start_page = null;
    }

    if (preg_match("/all_end_page='([^']*)'/", $text, $m)) {
        $all_end_page = $m[1];
    } else {
        $all_end_page = null;
    }

    //Issue (tex_is_part_of_iss).

    if (
        preg_match(
            "/tex_is_part_of_iss__target_id='([^']+)'/",
            $text,
            $m
        )
    ) {
        $tex_is_part_of_iss = $m[1];
```

```
    }

    //Section (tex_is_part_of_sec).

    if (preg_match("/tex_is_part_of_sec__target_id='([^']*)'/", $text, $m)) {
        $tex_is_part_of_sec = $m[1];
    } else {
        $tex_is_part_of_sec = null;
    }

    //Position on the page (tex_position_on_page).

    if (preg_match("/tex_position_on_page='([^']*)'/", $text, $m)) {
        $tex_position_on_page = $m[1];
    } else {
        $tex_position_on_page = null;
    }

    //Domain (tex_domain).

    $domains = [];

    if (
        preg_match_all(
            '/tex_domain=["\']([^"\']+)["\']/',
            $text,
            $domainsMatches
        )
    ) {
        foreach ($domainsMatches[1] as $value) {

            $parts = array_map('trim', explode(',', $value));

            $domains = array_merge($domains, $parts);
        }
    }

    $domains = array_unique($domains);

    $domains_target_id_array = [
        mb_strtolower('Belles-lettres, théâtre') => 197,
        mb_strtolower('Belles-lettres') => 197,
        mb_strtolower('Théâtre') => 197,
        mb_strtolower('Système et histoire des connaissances humaines') => 183,
        mb_strtolower('Système des connaissances humaines') => 183,
        mb_strtolower('Histoire des connaissances humaines') => 183,
        mb_strtolower('Système') => 183,
        mb_strtolower('Connaissances humaines') => 183,
        mb_strtolower('Beaux-arts') => 196,
        mb_strtolower('Histoire') => 195,
        mb_strtolower('Mathématiques') => 194,
        mb_strtolower('Métiers, arts mécaniques') => 193,
        mb_strtolower('Métiers') => 193,
        mb_strtolower('Arts mécaniques') => 193,
        mb_strtolower('Histoire naturelle') => 192,
        mb_strtolower('Sciences de la guerre') => 191,
        mb_strtolower('Sciences politiques') => 190,
        mb_strtolower('Pédagogie') => 189,
        mb_strtolower('Philosophie') => 188,
```

```
        mb_strtolower('Médecine, chirurgie, pharmacie') => 187,
        mb_strtolower('Médecine') => 187,
        mb_strtolower('Chirurgie') => 187,
        mb_strtolower('Pharmacie') => 187,
        mb_strtolower('Jurisprudence, droit') => 186,
        mb_strtolower('Jurisprudence') => 186,
        mb_strtolower('Droit') => 186,
        mb_strtolower('Théologie, religion') => 185,
        mb_strtolower('Théologie') => 185,
        mb_strtolower('Religion') => 185,
        mb_strtolower('Philologie, connaissance des langues') => 184,
        mb_strtolower('Philologie') => 184,
        mb_strtolower('Connaissance des langues') => 184,
        mb_strtolower('Musique, danse') => 198,
        mb_strtolower('Musique') => 198,
        mb_strtolower('Danse') => 198,
    ]; //Matching the values of the "Domaine" taxonomy of the Drupal database.

    $domains_target_id_array_2 = [];

    foreach ($domains as $domain) {

        $key = mb_strtolower(trim($domain));

        if (isset($domains_target_id_array[$key])) {
            $domains_target_id_array_2[] = $domains_target_id_array[$key];
        }
    }

    $domains_target_id_array_2 = array_unique($domains_target_id_array_2);

    $tex_domain = implode(',', $domains_target_id_array_2);

    //Form (tex_verse_prose).

    $forms = [];

    if (
        preg_match_all(
            '/tex_verse_prose=["\']([^"\']+)["\']/',
            $text,
            $formsMatches
        )
    ) {
        foreach ($formsMatches[1] as $value) {

            $parts = array_map('trim', explode(',', $value));

            $forms = array_merge($forms, $parts);
        }
    }

    $forms = array_unique($forms);

    $forms_target_id_array = [
        mb_strtolower('Vers et prose') => 174, //Values matching "Vers et prose"
the Drupal database.
        mb_strtolower('Vers, prose') => 174,
        mb_strtolower('Prose, vers') => 174,
```

```
        mb_strtolower('Vers') => 172, //Values matching "Vers" the Drupal
database.
        mb_strtolower('Prose') => 173, //Values matching "Prose" the Drupal
database (default value).
    ];

    $forms_target_id_array_2 = [];

    foreach ($forms as $form) {

        $key = mb_strtolower(trim($form));

        if (isset($forms_target_id_array[$key])) {
            $forms_target_id_array_2[] = $forms_target_id_array[$key];
        }
    }

    $forms_target_id_array_2 = array_unique($forms_target_id_array_2);

    $tex_verse_prose = implode(',', $forms_target_id_array_2);

    if ($tex_verse_prose === '') {
        $tex_verse_prose = 173;
    }

    //Language (tex_language).

    $languages = [];

    if (
        preg_match_all(
            '/tex_language=["\'](["\']+)["\']/',
            $text,
            $languagesMatches
        )
    ) {
        foreach ($languagesMatches[1] as $value) {

            $parts = array_map('trim', explode(',', $value));

            $languages = array_merge($languages, $parts);
        }
    }

    $languages = array_unique($languages);

    $languages_target_id_array = [
        mb_strtolower('Français') => 149, //Default value.

        mb_strtolower('Latin') => 150,
        mb_strtolower('Grec') => 151,
        mb_strtolower('Hébreu') => 152,
        mb_strtolower('Allemand') => 153,
        mb_strtolower('Anglais') => 154,
        mb_strtolower('Espagnol') => 155,
        mb_strtolower('Italien') => 156,
        mb_strtolower('Occitan') => 157,
        mb_strtolower('Autre dialecte français') => 158,
        mb_strtolower('Autre dialecte') => 158,
```

```
        mb_strtolower('Dialecte') => 158,
        mb_strtolower('Autre langue') => 159,
        mb_strtolower('Autre') => 159,
    ]; //Values matching the "Langue" taxonomy in the Drupal database.

    $languages_target_id_array_2 = [];

    foreach ($languages as $language) {

        $key = mb_strtolower(trim($language));

        if (isset($languages_target_id_array[$key])) {
            $languages_target_id_array_2[] = $languages_target_id_array[$key];
        }
    }

    $tex_language = end($languages_target_id_array_2);

    if ($tex_language === false) {
        $tex_language = 149;
    }

    //Paratext (tex_paratextual_genre).

    $paratexts = [];

    if (
        preg_match_all(
            '/tex_paratextual_genre=["\']([^"\']+)["\']/',
            $text,
            $paratextsMatches
        )
    ) {
        foreach ($paratextsMatches[1] as $value) {

            $parts = array_map('trim', explode(',', $value));

            $paratexts = array_merge($paratexts, $parts);
        }
    }

    $paratexts = array_unique($paratexts);

    $paratexts_target_id_array = [
        mb_strtolower('Autre élément paratextuel') => 171,
        mb_strtolower('Autre') => 171,
        mb_strtolower('Table des articles') => 170,
        mb_strtolower('Table des matières') => 170,
        mb_strtolower('Table') => 170,
        mb_strtolower('Privilège') => 169,
        mb_strtolower('Errata') => 168,
        mb_strtolower('Emplacement d\'une figure') => 167,
        mb_strtolower('Emplacement d&apos;une figure') => 167,
        mb_strtolower('Emplacement') => 167,
        mb_strtolower('Figure') => 167,
        mb_strtolower('Dédicace') => 166,
        mb_strtolower('Colophon') => 165,
        mb_strtolower('Calendrier') => 163,
        mb_strtolower('Avertissement / Avis des éditeurs') => 162,
```

```
        mb_strtolower('Avertissement, Avis des éditeurs') => 162,
        mb_strtolower('Avis des éditeurs / Avertissement') => 162,
        mb_strtolower('Avis des éditeurs, Avertissement') => 162,
        mb_strtolower('Avis des éditeurs') => 162,
        mb_strtolower('Avertissement') => 162,
        mb_strtolower('Approbation') => 161,
        mb_strtolower('Annonce / Avis commercial') => 160,
        mb_strtolower('Annonce, Avis commercial') => 160,
        mb_strtolower('Avis commercial / Annonce') => 160,
        mb_strtolower('Avis commercial, Annonce') => 160,
        mb_strtolower('Avis commercial') => 160,
        mb_strtolower('Annonce') => 160,
    ]; //Matching the values of the "Genre paratextuel" taxonomy of the Drupal
database.

    $paratexts_target_id_array_2 = [];

    foreach ($paratexts as $paratext) {

        $key = mb_strtolower(trim($paratext));

        if (isset($paratexts_target_id_array[$key])) {
            $paratexts_target_id_array_2[] = $paratexts_target_id_array[$key];
        }
    }

    $paratexts_target_id_array_2 = array_unique($paratexts_target_id_array_2);

    $tex_paratextual_genre = implode(',', $paratexts_target_id_array_2);

    //Type of journalistic writing (tex_type_of_journalistic_writing).

    $types = [];

    if (
        preg_match_all(
            '/tex_type_of_journalistic_writing=["\']([^"\']+)["\']/',
            $text,
            $typesMatches
        )
    ) {
        foreach ($typesMatches[1] as $value) {

            $parts = array_map('trim', explode(',', $value));

            $types = array_merge($types, $parts);
        }
    }

    $types = array_unique($types);

    $types_target_id_array = [
        mb_strtolower('Transition / Conclusion') => 22761,
        mb_strtolower('Transition, Conclusion') => 22761,
        mb_strtolower('Conclusion / Transition') => 22761,
        mb_strtolower('Conclusion') => 22761,
        mb_strtolower('Transition') => 22761,
        mb_strtolower('Morts / Naissances / Mariages') => 11527,
        mb_strtolower('Morts, Naissances, Mariages') => 11527,
```

```
        mb_strtolower('Naissances / Morts / Mariages') => 11527,
        mb_strtolower('Naissances, Morts, Mariages') => 11527,
        mb_strtolower('Morts / Naissances') => 11527,
        mb_strtolower('Morts, Naissances') => 11527,
        mb_strtolower('Naissances / Morts') => 11527,
        mb_strtolower('Naissances, Morts') => 11527,
        mb_strtolower('Mariages / Morts') => 11527,
        mb_strtolower('Mariages, Morts') => 11527,
        mb_strtolower('Morts / Mariages') => 11527,
        mb_strtolower('Morts, Mariages') => 11527,
        mb_strtolower('Morts') => 11527,
        mb_strtolower('Naissances') => 11527,
        mb_strtolower('Mariages') => 11527,
        mb_strtolower('Mort') => 11527,
        mb_strtolower('Naissance') => 11527,
        mb_strtolower('Mariage') => 11527,
        mb_strtolower('Autre type d\'écrit journalistique') => 182,
        mb_strtolower('Autre type d&apos;écrit journalistique') => 182,
        mb_strtolower('Autre type d\'écrit') => 182,
        mb_strtolower('Autre type d&apos;écrit') => 182,
        mb_strtolower('Autre type') => 182,
        mb_strtolower('Autre') => 182,
        mb_strtolower('Textes de loi') => 181,
        mb_strtolower('Texte de loi') => 181,
        mb_strtolower('Loi') => 181,
        mb_strtolower('Arrêts') => 181,
        mb_strtolower('Décrets') => 181,
        mb_strtolower('Relation / Nouvelle politique') => 180,
        mb_strtolower('Relation, Nouvelle politique') => 180,
        mb_strtolower('Nouvelle politique / Relation') => 180,
        mb_strtolower('Nouvelle politique') => 180,
        mb_strtolower('Relation') => 180,
        mb_strtolower('Prélude') => 179,
        mb_strtolower('Poésie') => 178,
        mb_strtolower('Jeu d\'esprit / Question') => 177,
        mb_strtolower('Jeu d&apos;esprit / Question') => 177,
        mb_strtolower('Jeu d\'esprit, Question') => 177,
        mb_strtolower('Jeu d&apos;esprit, Question') => 177,
        mb_strtolower('Question / Jeu d\'esprit') => 177,
        mb_strtolower('Question / Jeu d&apos;esprit') => 177,
        mb_strtolower('Question, Jeu d\'esprit') => 177,
        mb_strtolower('Question, Jeu d&apos;esprit') => 177,
        mb_strtolower('Jeu d\'esprit') => 177,
        mb_strtolower('Jeu d&apos;esprit') => 177,
        mb_strtolower('Question') => 177,
        mb_strtolower('Jeu') => 177,
        mb_strtolower('Fiction en prose') => 176,
        mb_strtolower('Fiction') => 176,
        mb_strtolower('Article / Nouvelle littéraire') => 175,
        mb_strtolower('Article, Nouvelle littéraire') => 175,
        mb_strtolower('Nouvelle littéraire / Article') => 175,
        mb_strtolower('Nouvelle littéraire, Article') => 175,
        mb_strtolower('Nouvelle littéraire') => 175,
        mb_strtolower('Article') => 175,
    ]; //Matching the values of the "Type d'écrit journalistique" taxonomy of the
Drupal database.

    $types_target_id_array_2 = [];
```

```
    foreach ($types as $type) {

        $key = mb_strtolower(trim($type));

        if (isset($types_target_id_array[$key])) {
            $types_target_id_array_2[] = $types_target_id_array[$key];
        }
    }

    $types_target_id_array_2 = array_unique($types_target_id_array_2);

    $tex_type_of_journalistic_writing = implode(',', $types_target_id_array_2);

    //Readers' letters (tex_readers_letters).

    $readers = [];

    if (
        preg_match_all(
            '/tex_readers_letters=["\']([^"\']+)["\']/',
            $text,
            $readersMatches
        )
    ) {
        foreach ($readersMatches[1] as $value) {

            $parts = array_map('trim', explode(',', $value));

            $readers = array_merge($readers, $parts);
        }
    }

    $readers = array_unique($readers);

    $readers_target_id_array = [
        mb_strtolower('Non') => '', //Default value.
        mb_strtolower('Oui') => 1,
    ];

    $readers_target_id_array_2 = [];

    foreach ($readers as $reader) {

        $key = mb_strtolower(trim($reader));

        if (isset($readers_target_id_array[$key])) {
            $readers_target_id_array_2[] = $readers_target_id_array[$key];
        }
    }

    $tex_readers_letters = end($readers_target_id_array_2);

    if ($tex_readers_letters === false) {
        $tex_readers_letters = '';
    }

    //Gender of the author (tex_author_gender).

    $genders = [];
```

```
    if (
        preg_match_all(
            '/tex_author_gender=["\']([^"\']+)["\']/',
            $text,
            $gendersMatches
        )
    ) {
        foreach ($gendersMatches[1] as $value) {

            $parts = array_map('trim', explode(',', $value));

            $genders = array_merge($genders, $parts);
        }
    }

    $genders = array_unique($genders);

    $genders_target_id_array = [
        mb_strtolower('Homme') => 146, //Values matching "Homme" the Drupal
database.
        mb_strtolower('Femme') => 147, //Values matching "Femme" the Drupal
database.
        mb_strtolower('Indéterminé') => 148, //Values matching "Indéterminé" the
Drupal database (default value).
    ];

    $genders_target_id_array_2 = [];

    foreach ($genders as $gender) {

        $key = mb_strtolower(trim($gender));

        if (isset($genders_target_id_array[$key])) {
            $genders_target_id_array_2[] = $genders_target_id_array[$key];
        }
    }

    $genders_target_id_array_2 = array_unique($genders_target_id_array_2);

    $tex_author_gender = implode(',', $genders_target_id_array_2);

    if ($tex_author_gender === '') {
        $tex_author_gender = 148;
    }

    //Collectivity (tex_author_collectivity).

    $collectivity = [];

    if (
        preg_match_all(
            '/tex_author_collectivity=["\']([^"\']+)["\']/',
            $text,
            $collectivityMatches
        )
    ) {
        foreach ($collectivityMatches[1] as $value) {
```

```
            $parts = array_map('trim', explode(',', $value));

            $collectivity = array_merge($collectivity, $parts);
        }
    }

    $collectivity = array_unique($collectivity);

    $collectivity_target_id_array = [
        mb_strtolower('Non') => '', //Default value.
        mb_strtolower('Oui') => 1,
    ];

    $collectivity_target_id_array_2 = [];

    foreach ($collectivity as $collectivity_2) {

        $key = mb_strtolower(trim($collectivity_2));

        if (isset($collectivity_target_id_array[$key])) {
            $collectivity_target_id_array_2[] =
$collectivity_target_id_array[$key];
        }
    }

    $tex_author_collectivity = end($collectivity_target_id_array_2);

    if ($tex_author_collectivity === false) {
        $tex_author_collectivity = '';
    }

    //Main Title (tex_title).

    $tex_title = '';

    if (
        preg_match_all(
            '/<tex_title>(.*?)<\/tex_title>/si',
            $text,
            $m,
            PREG_OFFSET_CAPTURE
        )
    ) {
        $titles = [];

        $previousEnd = null;

        foreach ($m[1] as $match) {

            $title = strip_tags($match[0], '<em><sup>');
            $offset = $match[1];

            $title = trim($title);

            if ($title === '') {
                continue;
            }

            if ($previousEnd !== null) {
```

```
                $between = substr(
                    $text,
                    $previousEnd,
                    $offset - $previousEnd
                );

                $between = trim(strip_tags($between));

                if (mb_strlen($between) >= 50) {
                    $titles[] = ' / ';
                } else {
                    $titles[] = ' ';
                }
            }

            $titles[] = $title;

            $previousEnd = $offset + strlen($match[0]);
        }

        $tex_title = implode('', $titles);

        $tex_title = preg_replace(
            '/\s+/',
            ' ',
            $tex_title
        );

        $tex_title = mergeFormattingTags($tex_title);

        $tex_title = html_entity_decode(
            $tex_title,
            ENT_QUOTES | ENT_XML1,
            'UTF-8'
        );
    }

    //Manage rich text field.

    if($tex_title != ''){

        $tex_title = preg_replace(
            '/(^|[\s(\[{])"(?=\S)/u',
            '$1« ',
            $tex_title
        );

        $tex_title = preg_replace(
            '/(?<=\S)"(?=[\s.,;:!?)\]}]|$)/u',
            ' »',
            $tex_title
        );

        $tex_title = str_replace("’", "'", $tex_title);
        $tex_title = '{ format: full_html, value: "<p>'.$tex_title.'</p>" }';
    }

    //Title from the table (tex_title_table).
```

```
    $tex_title_table = null;

    if (
        preg_match(
            '/<tex_title_table>(.*?)<\/tex_title_table>/si',
            $text,
            $m
        )
    ) {
        $tex_title_table = trim($m[1]);

        $text = preg_replace(
            '/<tex_title_table>.*?<\/tex_title_table>/si',
            '',
            $text,
            1
        );
    }

    $tex_title_table = html_entity_decode(
        $tex_title_table,
        ENT_QUOTES | ENT_XML1,
        'UTF-8'
    );

    //Manage rich text field.

    if($tex_title_table != ''){

        $tex_title_table = preg_replace(
            '/(^|[\s(\[{])"(?=\S)/u',
            '$1« ',
            $tex_title_table
        );

        $tex_title_table = preg_replace(
            '/(?<=\S)"(?=[\s.,;:!?)\]}]|$)/u',
            ' »',
            $tex_title_table
        );

        $tex_title_table = str_replace("’", "'", $tex_title_table);
        $tex_title_table = '{ format: full_html, value:
"<p>'.$tex_title_table.'</p>" }';
    }

    //Incipit (tex_incipit).

    $tex_incipit = '';

    if (
        preg_match_all(
            '/<tex_incipit>(.*?)<\/tex_incipit>/si',
            $text,
            $m,
            PREG_OFFSET_CAPTURE
        )
    ) {
```

```
        $incipits = [];

        $previousEnd = null;

        foreach ($m[1] as $match) {

            $incipit = strip_tags($match[0], '<em><sup>');
            $offset = $match[1];

            $incipit = trim($incipit);

            if ($incipit === '') {
                continue;
            }

            if ($previousEnd !== null) {

                $between = substr(
                    $text,
                    $previousEnd,
                    $offset - $previousEnd
                );

                $between = trim(strip_tags($between));

                if (mb_strlen($between) >= 50) {
                    $incipits[] = ' / ';
                } else {
                    $incipits[] = ' ';
                }
            }

            $incipits[] = $incipit;

            $previousEnd = $offset + strlen($match[0]);
        }

        $tex_incipit = implode('', $incipits);

        $tex_incipit = preg_replace(
            '/\s+/',
            ' ',
            $tex_incipit
        );

        $tex_incipit = mergeFormattingTags($tex_incipit);

        $tex_incipit = html_entity_decode(
            $tex_incipit,
            ENT_QUOTES | ENT_XML1,
            'UTF-8'
        );
    }

    //Manage rich text field.

    if($tex_incipit != ''){

        $tex_incipit = preg_replace(
```

```
            '/(^|[\s(\[{])"(?=\S)/u',
            '$1« ',
            $tex_incipit
        );

        $tex_incipit = preg_replace(
            '/(?<=\S)"(?=[\s.,;:!?)\]}]|$)/u',
            ' »',
            $tex_incipit
        );

        $tex_incipit = str_replace("’", "'", $tex_incipit);
        $tex_incipit = '{ format: full_html, value: "<p>'.$tex_incipit.'</p>" }';
    }

    //Signature (tex_signature).

    $tex_signature = '';

    if (
        preg_match_all(
            '/<tex_signature>(.*?)<\/tex_signature>/si',
            $text,
            $m,
            PREG_OFFSET_CAPTURE
        )
    ) {
        $signatures = [];

        $previousEnd = null;

        foreach ($m[1] as $match) {

            $signature = strip_tags($match[0], '<em><sup>');
            $offset = $match[1];

            $signature = trim($signature);

            if ($signature === '') {
                continue;
            }

            if ($previousEnd !== null) {

                $between = substr(
                    $text,
                    $previousEnd,
                    $offset - $previousEnd
                );

                $between = trim(strip_tags($between));

                if (mb_strlen($between) >= 50) {
                    $signatures[] = ' / ';
                } else {
                    $signatures[] = ' ';
                }
            }
```

```
            $signatures[] = $signature;

            $previousEnd = $offset + strlen($match[0]);
        }

        $tex_signature = implode('', $signatures);

        $tex_signature = preg_replace(
            '/\s+/',
            ' ',
            $tex_signature
        );

        $tex_signature = mergeFormattingTags($tex_signature);

        $tex_signature = html_entity_decode(
            $tex_signature,
            ENT_QUOTES | ENT_XML1,
            'UTF-8'
        );
    }

    //Manage rich text field.

    if($tex_signature != ''){

        $tex_signature = preg_replace(
            '/(^|[\s(\[{])"(?=\S)/u',
            '$1« ',
            $tex_signature
        );

        $tex_signature = preg_replace(
            '/(?<=\S)"(?=[\s.,;:!?)\]}]|$)/u',
            ' »',
            $tex_signature
        );

        $tex_signature = str_replace("’", "'", $tex_signature);
        $tex_signature = '{ format: full_html, value: "<p>'.$tex_signature.'</p>"
}';
    }

    //Folder (tex_is_part_of_fol).

    $folders = [];

    if (
        preg_match_all(
            '/tex_is_part_of_fol=["\']([^"\']+)["\']/',
            $text,
            $folders_matches
        )
    ) {
        foreach ($folders_matches[1] as $value) {
            $parts = array_map('trim', explode(',', $value));

            foreach ($parts as $part) {
                if ($part !== '' && ctype_digit($part)) {
```

```
                $folders[] = $part;
            }
        }
    }
}

$folders = array_values(array_unique($folders));

$tex_is_part_of_fol = implode(',', $folders);

//Public remark (all_remark).

$all_remark = null;

if (
    preg_match(
        '/<all_remark>(.*?)<\/all_remark>/si',
        $text,
        $m
    )
) {
    $all_remark = trim($m[1]);

    $text = preg_replace(
        '/<all_remark>.*?<\/all_remark>/si',
        '',
        $text,
        1
    );
}

$all_remark = html_entity_decode(
    $all_remark,
    ENT_QUOTES | ENT_XML1,
    'UTF-8'
);

//Manage rich text field.

if($all_remark != ''){

    $all_remark = preg_replace(
        '/(^|[\s(\[{])"(?=\S)/u',
        '$1« ',
        $all_remark
    );

    $all_remark = preg_replace(
        '/(?<=\S)"(?=[\s.,;:!?)\]}]|$)/u',
        ' »',
        $all_remark
    );

    $all_remark = str_replace("’", "'", $all_remark);
    $all_remark = '{ format: full_html, value: "<p>'.$all_remark.'</p>" }';
}

//Private remark (all_remark_private).
```

```
    $all_remark_private = null;

    if (
        preg_match(
            '/<all_remark_private>(.*?)<\/all_remark_private>/si',
            $text,
            $m
        )
    ) {
        $all_remark_private = trim($m[1]);

        $text = preg_replace(
            '/<all_remark_private>.*?<\/all_remark_private>/si',
            '',
            $text,
            1
        );
    }

    $all_remark_private = html_entity_decode(
        $all_remark_private,
        ENT_QUOTES | ENT_XML1,
        'UTF-8'
    );

    //Manage rich text field.

    if($all_remark_private != ''){

        $all_remark_private = preg_replace(
            '/(^|[\s(\[{])"(?=\S)/u',
            '$1« ',
            $all_remark_private
        );

        $all_remark_private = preg_replace(
            '/(?<=\S)"(?=[\s.,;:!?)\]}]|$)/u',
            ' »',
            $all_remark_private
        );

        $all_remark_private = str_replace("’", "'", $all_remark_private);
        $all_remark_private = '{ format: full_html, value:
"<p>'.$all_remark_private.'</p>" }';
    }

    //OCR (text_ocr).

    $tex_ocr = $text;

    $tex_ocr = preg_replace(
        '/<tex_start\b[^>]*>.*?<\/tex_start>/si',
        '',
        $tex_ocr
    );

    $tex_ocr = preg_replace(
        '/<tex_end\b[^>]*>.*?<\/tex_end>/si',
        '',
```

```
        $tex_ocr
    );

    $tex_ocr = preg_replace(
        '/<tex_title_table\b[^>]*>.*?<\/tex_title_table>/si',
        '',
        $tex_ocr
    );

    $tex_ocr = preg_replace(
        '/<all_remark\b[^>]*>.*?<\/all_remark>/si',
        '',
        $tex_ocr
    );

    $tex_ocr = preg_replace(
        '/<all_remark_private\b[^>]*>.*?<\/all_remark_private>/si',
        '',
        $tex_ocr
    );

    $tex_ocr = preg_replace(
        '/<\/?(tex_title|tex_incipit|tex_signature)\b[^>]*>/i',
        '',
        $tex_ocr
    );

    $tex_ocr = preg_replace(
        '/^[ \t]*(<br\s*\/?>)[ \t]*$\R?/mi',
        '',
        $tex_ocr
    );

    $tex_ocr = preg_replace(
        '/<\/lg>.*?<lg\b[^>]*>/si',
        '',
        $tex_ocr
    );

    $tex_ocr = preg_replace(
        '/<\/?lg\b[^>]*>/i',
        '',
        $tex_ocr
    );

    $tex_ocr = preg_replace(
        '/(<br\s*\/?>)\s+/i',
        '$1',
        $tex_ocr
    );

    $tex_ocr = preg_replace(
        '/<br\s*\/?>/i',
        "\n",
        $tex_ocr
    );

    $tex_ocr = strip_tags($tex_ocr);
```

```
    $tex_ocr = html_entity_decode(
        $tex_ocr,
        ENT_QUOTES | ENT_XML1,
        'UTF-8'
    );

    $tex_ocr = str_replace("’", "'", $tex_ocr);
    $tex_ocr = trim($tex_ocr);

    //Validity (all_validity).

    $all_validity = 30541; //Value matching "Indexation préliminaire" in the
Drupal database.

    //Source of the OCR (tex_ocr_external_source) (see at the top of the file).

    $tex_ocr_external_source = $ocr_source;

    //Combine the fields in CSV format.

    $row = [
        $tex_is_part_of_iss,
        $tex_is_part_of_sec,
        $all_start_page_file,
        $all_start_page,
        $all_end_page_file,
        $all_end_page,
        $tex_position_on_page,
        $tex_title,
        $tex_title_table,
        $tex_incipit,
        $tex_signature,
        $tex_author_gender,
        $tex_author_collectivity,
        $tex_language,
        $tex_domain,
        $tex_verse_prose,
        $tex_paratextual_genre,
        $tex_type_of_journalistic_writing,
        $tex_readers_letters,
        $tex_ocr,
        $tex_ocr_external_source,
        $tex_is_part_of_fol,
        $all_remark,
        $all_remark_private,
        $all_validity
    ];

    $fp = fopen('php://memory', 'r+');
    fputcsv($fp, $row);
    rewind($fp);
    $output_csv .= stream_get_contents($fp);
    fclose($fp);

    $total_displayed ++;

}

//Echo the number of results.
```

```
echo 'Fichier XML : '.$file.'<br>';
echo 'Nombre de textes : '.$total_displayed.'<br><br>';

//Output in CSV format.

echo '<textarea rows="40" cols="180">';
echo
'tex_is_part_of_iss,tex_is_part_of_sec,all_start_page_file,all_start_page,all_end_
page_file,all_end_page,tex_position_on_page,tex_title,tex_title_table,tex_incipit,
tex_signature,tex_author_gender,tex_author_collectivity,tex_language,tex_domain,te
x_verse_prose,tex_paratextual_genre,tex_type_of_journalistic_writing,tex_readers_l
etters,tex_ocr,tex_ocr_external_source,tex_is_part_of_fol,all_remark,all_remark_pr
ivate,all_validity';
echo "\n";
echo $output_csv;
echo '</textarea>';

?>
```

14. Copier les données depuis un navigateur web et coller l'ensemble dans un nouveau fichier CSV vide.
15. Importer les données dans *Drupal*, grâce à l'outil d'importation du module Webform.

Résumés produits par l'intelligence artificielle

DONNÉES RELATIVES AUX TEXTES

Objectifs

Sur la base de la reconnaissance optique de caractères des textes du *Mercure*, l'intelligence artificielle générative produit des résumés que nous importons dans notre base de données. Ces résumés offrent un aperçu des contenus textuels du *Mercure*.

Données concernées

Dans le champ tex_ocr, nous avons inséré la reconnaissance optique de caractères de chaque texte, en copiant celle des fichiers PDF des livraisons, téléchargés auprès de différentes bibliothèques numériques. Le contenu du champ tex_ocr permet, avec l'aide de l'intelligence artificielle générative, de produire des résumés qui sont ensuite importés dans le champ tex_sumary. La longueur du résumé dépend de la longueur du texte source (voir la procédure détaillée ci-dessous). En théorie, un résumé ne dépasse jamais 300 mots.

Utilisation

Tous les textes sont résumés, sauf ceux qui comptent moins de 500 caractères, espaces comprises. Les résumés ne sont pas retouchés par les collaborateurs. Ils sont destinés à l'affichage sur notre site, à l'exception des résumés des jeux littéraires (énigmes, logogriphes et charades) qui s'avèrent souvent insatisfaisants. L'affichage des résumés inclut un avertissement sur les erreurs qu'ils sont susceptibles de contenir. Nous précisons aussi le modèle de langage utilisé, à savoir Mistral. Tous les résumés sont susceptibles d'être remplacés quand nous modifions le contenu du champ tex_ocr ou quand le modèle de langage évolue.

Pour produire les résumés, nous recourons à l'aide d'un partenaire externe, Ivo Durisch de la société Lineasoft (Riva San Vitale, Suisse). Ce partenaire décrit son travail ci-dessous, dans l'annexe « Traitement des données et usage du modèle Mistral ».

Les textes ne peuvent être résumés qu'après avoir saisi leur reconnaissance optique de caractères dans la base de données. À intervalles réguliers, nous générons les résumés des textes indexés (ou modifiés) récemment, par lots de quelques centaines ou milliers de textes, avant d'importer ces nouveaux résumés dans la base de données.

Procédure détaillée

1. Nous extrayons le contenu du champ tex_ocr en format JSON, au moyen du module Views Data Export de *Drupal*. Pour chaque texte, nous extrayons également l'identifiant du contenu (nid), l'identifiant du formulaire Webform soumis (all_webform_submission_id) et l'identifiant unique universel du formulaire Webform soumis (uuid) qui facilitera plus tard l'importation des résumés. Le fichier obtenu se présente comme suit :


```
[{"nid":"2549136","all_webform_submission_id":"2549149","uuid":"\n  \n    \n
\n      \n          36cd6eb9-e19f-4e25-ad67-e3148d2a39fc\n\n    \n  \n
","tex_ocr":"Le Mardy 22. de ce mois , l\u0026#039;Acad\u00e9mie\u003Cbr
\/\u003E\r\nRoyale des Belles- Letrres tint son As\u0002sembl\u00e9e publique. M.
l\u0026#039;Abb\u00e9 Bignon y\u003Cbr \/\u003E\r\npr\u00e9sida.\u003Cbr
\/\u003E\r\nM. l\u0026#039;Abb\u00e9 Canaye ouvrit la s\u00e9ance par\u003Cbr
\/\u003E\r\nla lecture d\u0026#039;une Dissertation ; sur la Per\u0002sonne , les
Sentimens , et les Ouvrages\u003Cbr \/\u003E\r\nd\u0026#039;Anaximandre ,
Philosophe Grec , Disci\u0002ple de Thales.\u003Cbr \/\u003E\r\nM.
l\u0026#039;Abb\u00e9 Sallier lut pour M. Racine\u003Cbr \/\u003E\r\nabsent , un
parallele de l\u0026#039;Andromaque\u003Cbr \/\u003E\r\nd\u0026#039;Eurypide , et
de celle de M. Racine ,\u003Cbr \/\u003E\r\npere.\u003Cbr \/\u003E\r\nM.\u003Cbr
\/\u003E\r\nAVRIL. 1732. 827\u003Cbr \/\u003E\r\nM. de Foncemagne lut un Discours
,\u003Cbr \/\u003E\r\ndans lequel il examine et refute le senti\u0002ment du Comte
de Boulinvilliers , au\u003Cbr \/\u003E\r\n\u00b4sujet de l\u0026#039;Election des
Generaux de l\u0026#039;Ar\u0002m\u00e9e , que cet Auteur donne \u00e0 la
Nation\u003Cbr \/\u003E\r\nFran\u00e7oise pendant la premiere Race , et\u003Cbr
\/\u003E\r\nqu\u0026#039;il croyoit \u00eatre une chose essentielle\u0002ment
separ\u00e9e de la Royaut\u00e9.\u003Cbr \/\u003E\r\nM. l\u0026#039;Abb\u00e9
Sallier, lut une Ode de Pin\u0002dare , traduite en Fran\u00e7ois , avec des
Re\u0002marques historiques et critiques , suivant\u003Cbr \/\u003E\r\nla Methode
du feu Abb\u00e9 Massieu.\u003Cbr \/\u003E\r\nM.Bonami, termina la S\u00e9ance par
l\u0026#039;his\u0002toire de la guerre que Cesar so\u00fbtint dans\u003Cbr
\/\u003E\r\nAlexandrie contre Ptolem\u00e9e et les Egyp\u0002tiens , apr\u00e8s la
mort de Pomp\u00e9e , et la\u003Cbr \/\u003E\r\nBataille de Pharsale ,
conform\u00e9ment aux\u003Cbr \/\u003E\r\nPlans de la ville d\u0026#039;Alexandrie
, ancienne\u003Cbr \/\u003E\r\net moderne , et de ses deux Ports ,
\u0026amp;c.\u003Cbr \/\u003E\r\ndont il avoit donn\u00e9 l\u0026#039;explication
dans une\u003Cbr \/\u003E\r\nDissertation lue \u00e0
l\u0026#039;Acad\u00e9mie.\u003Cbr \/\u003E\r\nLe Mecredi 23,
l\u0026#039;Acad\u00e9mie Royale des\u003Cbr \/\u003E\r\nSciences , tint aussi son
Assembl\u00e9e publi\u0002que d\u0026#039;apr\u00e8s P\u00e2ques.\u003Cbr
\/\u003E\r\nM. de Fontenelle ouvrit l\u00e1 sc\u00e9ance par\u003Cbr
\/\u003E\r\nPEloge de M. de Maisons , Pr\u00e9sident\u003Cbr \/\u003E\r\ndu
Parlement , et honoraire de cette\u003Cbr \/\u003E\r\nAcad\u00e9mie , mort
l\u0026#039;ann\u00e9e derniere.\u003Cbr \/\u003E\r\nM.d\u0026#039;Ons-embrai lut
apr\u00e8s la description\u003Cbr \/\u003E\r\net la maniere de construire un
Instrument\u003Cbr \/\u003E\r\npropre\u003Cbr \/\u003E\r\n868 MERCURE DE
FRANCE\u003Cbr \/\u003E\r\npropre\u00e0 faire \u00e9xecuter les Airs
```

```
deMusique\u003Cbr \/\u003E\r\ndans le vrai mouvement qu\u0026#039;ils ont
\u00e9t\u00e9\u003Cbr \/\u003E\r\ncomposez.\u003Cbr \/\u003E\r\n,\u003Cbr
\/\u003E\r\nM. de Reaumur lut ensuite un M\u00e9moi\u0002re d\u0026#039;Histoire
naturelle , sur l\u0026#039;insecte ap\u0002pell\u00e9 Teigne des arbres.\u003Cbr
\/\u003E\r\nApr\u00e8s cel\u00e0 M. Dufay lut un M\u00e9moire\u003Cbr
\/\u003E\r\nsur la maniere de colorer le marbre..\u003Cbr \/\u003E\r\nM. Buache
donna la Description d\u0026#039;un s\u003Cbr \/\u003E\r\nInstrument propre \u00e0
observer la varia\u0002tion de l\u0026#039;aiguille aimant\u00e9e , et en
m\u00eame\u003Cbr \/\u003E\r\ntems sa d\u00e9clinaison .\u003Cbr \/\u003E\r\nM.
Duhamel finit la sc\u00e9ance par la lec\u0002ture d\u0026#039;un M\u00e9moire sur
la formation d\u0026#039;un\u003Cbr \/\u003E\r\nsel , \u0026amp;c. Nous donnerons
un Extrait de\u003Cbr \/\u003E\r\nces
M\u00e9moires."}][{"nid":"2553354","all_webform_submission_id":"2553367","uuid":"\
n  \n    \n      \n      \n          f0f1fd10-08ff-4842-a896-bc93cf216145\n\n
\n  \n          ","tex_ocr":"ENIGM E.\u003Cbr \/\u003E\r\nE ne fuis point Iris ,
je ne fais point l\u0026#039;Aurore ,\u003Cbr \/\u003E\r\nEt
j\u0026#039;\u00e9tale \u017fouvent les plus vives couleurs .\u003Cbr
\/\u003E\r\nJe ne fuis Z\u00e9phire , ni Flore ,\u003Cbr \/\u003E\r\nEtj\u00e9
fais \u017fur mes pas \u00e9clore mille fleurs ;\u003Cbr \/\u003E\r\nMoins riche
moins brillante on me voit au Village\u003Cbr \/\u003E\r\nRendre \u00e0
l\u0026#039;humble Pafteur fes utiles pr\u00e9fents ;\u003Cbr \/\u003E\r\nMais je
ne puis fufire \u00e0 mes dons bienfai\u017fans ,\u003Cbr \/\u003E\r\nEt je me
ruine en voyage.\u003Cbr \/\u003E\r\nEnfin pour des fils \u00e9trangers ,\u003Cbr
\/\u003E\r\nJ\u0026#039;abandonne le feul que dans mon fein je porte.\u003Cbr
\/\u003E\r\nMais il trame avec eux une ligue fi forte ,\u003Cbr
\/\u003E\r\nQu\u0026#039;apr\u00e8s l\u0026#039;avoir conduit \u00e0 travers les
dangers ,\u003Cbr \/\u003E\r\nIl me chaffe d\u0026#039;entr\u0026#039;eux ,
\u0026amp; qu\u0026#039;il faut que je forte\u003Cbr \/\u003E\r\nDa lieu m\u00eame
embelli par mes foins
pallagers."}][{"nid":"2553355","all_webform_submission_id":"2553368","uuid":"\n
\n    \n      \n      \n          c19e5ac4-4d87-4491-a204-1364ba77acc2\n\n    \n
\n          ","tex_ocr":"LOGOGRYPHE.\u003Cbr \/\u003E\r\nRienn\u0026#039;eft plus
vieux, rien n\u0026#039;e\u017ft \u017fi beau que moi.\u003Cbr \/\u003E\r\nDes
lettres de mon nom efface la troifi\u00e9me ;\u003Cbr \/\u003E\r\nVieux ou jeune ,
je fuis d\u0026#039;une laideur extr\u00eame .\u003Cbr \/\u003E\r\nRetranche la
feconde ; \u00e0 chaque inftant , chez toi,\u003Cbr
\/\u003E\r\nJ\u0026#039;augmente en d\u00e9pit de toi-m\u00eame.\u003Cbr
\/\u003E\r\nTon embarras me fait piti\u00e9 .\u003Cbr \/\u003E\r\nTu ne
m\u0026#039;as jamais vu , tu ne peux me conno\u00eetres\u003Cbr \/\u003E\r\nMais
reconnois au moins ma premiere moiti\u00e9 :\u003Cbr \/\u003E\r\nTu l\u0026#039;as
v\u00fb mourir \u0026amp; rena\u00eetre ,"}]
[...]
[{"nid":"2565636","all_webform_submission_id":"2565651","uuid":"\n  \n    \n
\n      \n          bc4d9930-ee94-4919-af85-b54f85d22975\n\n    \n  \n
","tex_ocr":"Avis pour placer la McdaiS\u00e2,"}]
```


2. Nous transmettons ce fichier à notre partenaire qui effectue plusieurs opérations (voir ci-dessous l'annexe « Traitement des données et usage du modèle Mistral »).
    a. Division du fichier JSON pour disposer d'un fichier par texte.
    b. Nettoyage léger de la reconnaissance optique de caractères.
    c. Division des textes en segments.
    d. Production de résumés partiels, au moyen du modèle de langage Mistral utilisé localement. Les prompts sont les suivants :

       SYSTEM_PROMPT_SUMMARY : « Vous êtes un assistant spécialisé dans le résumé de textes historiques. Votre tâche consiste à produire un résumé fidèle, objectif et factuel du texte fourni.

- Résumez en un maximum de 50/150/300 mots [en fonction de la longueur du texte source].
- Conservez uniquement les informations présentes dans le texte.
- Adoptez un ton formel, neutre et précis.
- N'ajoutez aucune interprétation ou contexte extérieur.
- N'utilisez aucun balisage (par exemple ```). »

DEFAULT_USER_REQUEST : « Résumez le document, en mettant clairement en évidence les points essentiels, en utilisant exclusivement les informations contenues dans le texte. »

e. À partir des résumés partiels, l'intelligence artificielle recompose pour chaque texte un résumé complet et cohérent.

3. Le partenaire nous transmet alors les résumés en format CSV. Le fichier se présente comme suit :

```
"sid","uuid","tex_summary"
"2561537","e9dc940d-cfb7-4409-b012-370c6aaf4103","En mai 1757, des événements
militaires et diplomatiques marquent l'Allemagne. À Vienne, le 18 mai, les détails
de la bataille près de Prague du 6 mai restent flous. Les troupes du Prince
Charles de Lorraine à Prague totalisent 45 000 hommes, tandis que l'armée du Feld-
Maréchal Comte de Daun, entre Kollin et Kuttenberg, en compte plus de 50 000 et
doit être renforcée par le corps du Général Nadafty et 20 bataillons. À
Ratisbonne, le 22 mai, le Baron de Mackau transmet une déclaration du roi de
France à la Diète de l'Empire, datée du 26 avril. À Amberg, 800 hommes des troupes
prussiennes, avec des canons et des mortiers, arrivent le 18 mai, entraînant une
rencontre entre un lieutenant-colonel prussien et des représentants de la régence.
Par ailleurs, des troupes légères du roi de Prusse pénètrent dans le Haut-
Palatinat, imposant des contributions à certains bourgs des États de l'Électeur
Palatin."
"2555116","357a89bf-587a-4deb-96c9-1faf84631636","Le texte présente deux énigmes :
'La Dame Solitaire' et 'Clione, Nymphe enjoüée'. La première énigme décrit une
entité née dans le feu et pouvant mourir dans l'eau. Elle subsiste malgré la perte
de couleur, chaleur et les éléments. Elle existe même après la fin du monde."
"2565548","f4447c41-3098-4bf6-af48-a61b5087b6db","L'auteur s'adresse aux juges et
protecteurs des talents, reconnaissant leur rôle dans l'encouragement des
artistes. Il mentionne avoir présenté des ballets héroïques en 1758, tels que
'Télémaque dans l'Ile de Calypso', 'le Sultan généreux' et 'la Dispute des Faunes
& des Bergers'. Aujourd'hui, il présente 'Ulysse dans l'Ile de Circé', un genre de
ballet peu connu à Paris, nécessitant une exposition, une intrigue, des situations
et un dénouement. L'auteur exprime ses efforts pour réunir ces éléments essentiels
et sa gratitude pour les suffrages reçus, tout en reconnaissant que ses œuvres
sont loin de la perfection. Il souligne la difficulté croissante de la composition
et de l'exécution de ces ballets, nécessitant de l'âme, du sentiment et de la
pratique. Il appelle à l'indulgence pour les danseurs et danseuses, habitués à de
petits divertissements, mais désireux de s'améliorer. L'auteur sollicite le
soutien du public pour ces artistes et exprime son espoir de succès grâce à son
travail assidu et son dévouement à satisfaire les volontés du public."
"2565554","d3175abf-9f9a-476a-ac2f-0ce89776cf61","Le texte décrit le voyage et la
mission diplomatique du Chevalier de Chaumont, ambassadeur de France au Siam.
Parti de Brest le 3 mars de l'année précédente avec deux navires, il arriva à
Versailles le 24 juin, après avoir parcouru 12 000 lieues en moins de 15 mois et
demi. Son itinéraire inclut le passage du Tropique du Cancer, du Cap de Bonne
Espérance, ainsi que des escales à Batavia et au Siam. À son arrivée, le roi de
Siam avait préparé des maisons et des mandarins pour l'accueillir et lui offrit
une audience solennelle avec des honneurs exceptionnels, permettant à Chaumont et
```

```
à ses gentilshommes de rester assis tandis que la cour était prosternée. Le roi de
Siam manifesta une grande admiration pour le roi de France et ses actions,
demandant des nouvelles de la famille royale française et offrant des présents
somptueux. Chaumont visita également des places fortes et permit à un ingénieur de
vérifier les fortifications. Il était accompagné de six jésuites, experts en
mathématiques et astronomie, destinés à la Chine pour y établir le christianisme.
Impressionné par leurs connaissances, le roi de Siam offrit de leur construire une
église et une maison. Le roi de Siam, veuf, a une fille, la princesse Reine, qui
règne souverainement sur trois grandes provinces. Chaumont reçut la première
dignité du royaume, devenant Oya, et fut dispensé de certaines prosternations. Il
refusa une partie des meubles du palais mais offrit des présents, dont un miroir à
la princesse Reine. Il quitta le Siam avec l'admiration de la cour et du peuple,
accompagné par le favori du roi jusqu'à son embarquement. Les ambassadeurs
siamois, au nombre de trois, l'accompagnèrent en France avec une suite de
cinquante personnes, apportant des présents pour le roi de France et la maison
royale. Chaumont arriva à Brest le 19 juin, marquant une grande diligence navale."
[...]
"2553521","9f3c5a42-fdd4-4933-9646-aa37c491fa7d","Le texte est un poème ou une
chanson morale daté de juin 1761, écrit par M. Charpentier et édité par Tournelle.
Il s'adresse à une personne chère, évoquant une distance jalouse qui a ébranlé la
confiance, mais sans éteindre les sentiments tendres et sincères du locuteur. Ce
dernier invite la personne à découvrir ces sentiments à travers six lettres
mystérieuses. Le poème mentionne une préface de bonne humeur et de gaieté, un
animal redoutable qui déroute les passants, et un mot de la bouche adorable de la
personne aimée. La deuxième partie du texte aborde le destin et le passage du
temps. Le locuteur parle de son abandon du Dieu du vin pour Silvia et de sa
crainte envers les Muses, bien qu'il se demande s'il est plus sage pour autant. Il
mentionne une promesse attendue et offre de la musique et une liqueur, concluant
par une épithète de son cœur, qu'il affirme être sans tache."
```

4. Il s’agit maintenant d’ajouter les nouveaux résumés dans la base de données *Drupal*, en modifiant les fiches de textes concernées et en écrasant au besoin les résumés préexistants. Nous importons donc le fichier CSV avec l’outil d’importation du module Webform.
5. Grâce à l’identifiant unique universel, l’importation modifie tous les formulaires Webform concernés, sans en créer de nouveaux. Cependant, le module Webform Content Creator qui devrait mettre à jour les contenus correspondants aux formulaires ne fonctionne qu’imparfaitement. Dans ces circonstances, il faut accéder *via* l’interface d’administration de Webform à la liste des formulaires soumis et réenregistrer manuellement les formulaires qui viennent d’être modifiés. Il suffit d’ouvrir le formulaire et de l’enregistrer de nouveau, sans toucher aux données qu’il contient.

ANNEXE : TRAITEMENT DES DONNÉES ET USAGE DU MODÈLE MISTRAL

Par Ivo Durisch (Lineasoft, Riva San Vitale, Suisse), 21 décembre 2025.

Introduction

Ce projet vise à enrichir une base de données publique de textes historiques par la production de résumés automatiques issus de données OCR. L’objectif est de faciliter l’accès aux contenus, d’améliorer leur exploitabilité scientifique et de permettre une navigation efficace au sein de corpus volumineux, tout en respectant strictement le contenu des sources originales.

Les textes traités présentent plusieurs contraintes méthodologiques. Le volume de données est élevé et hétérogène, tant en nombre de documents qu’en longueur, certains textes pouvant atteindre plusieurs centaines de pages. La qualité de l’OCR est variable selon les documents.

En outre, les résumés produits doivent répondre à une exigence centrale de neutralité et de fidélité au texte, sans ajout d'interprétation ou de contexte externe.

Afin de répondre à ces contraintes, le projet repose sur un traitement entièrement local des données, sans recours à des services cloud. Ce choix garantit la maîtrise des données et la reproductibilité des résultats.

Notions générales

Cette section présente les notions techniques essentielles à la compréhension du dispositif mis en place, en privilégiant une approche descriptive et non spéculative.

Qu'est-ce qu'un LLM (Large Language Model) ?

Un *Large Language Model* (LLM) est un modèle statistique entraîné sur de grands corpus textuels, capable de produire, transformer et résumer du texte à partir d'un contenu fourni en entrée. Il ne possède pas de compréhension propre des textes et n'effectue aucune interprétation. Son fonctionnement dépend étroitement de la qualité des données d'entrée et des instructions qui lui sont fournies, ce qui rend nécessaire l'usage de prompts stricts et contrôlés.

Les LLM sont particulièrement adaptés aux tâches de résumé automatique, dans la mesure où ils permettent une synthèse lexicale fondée exclusivement sur les informations présentes dans le texte source, tout en conservant les éléments factuels essentiels.

Qu'est-ce qu'un agent logiciel ?

Dans ce projet, le modèle de langage est intégré au sein d'un agent logiciel, entendu comme un programme qui organise et enchaîne plusieurs étapes de traitement. L'agent gère notamment la segmentation des textes, les contrôles intermédiaires et la recomposition finale des résumés.

Le modèle Mistral

Le modèle de langage utilisé est un modèle *open-weight* exécutable localement. Il offre de bonnes performances pour les tâches de résumé tout en permettant un contrôle complet du processus de traitement.

Ce choix répond à des exigences de souveraineté des données et de maîtrise du pipeline.

Données d'entrée

Cette section décrit la nature des données sources utilisées dans le projet ainsi que les transformations appliquées afin de fournir une entrée stable et exploitable au dispositif de résumé automatique.

Format des données sources

Les données d'entrée sont fournies sous la forme de fichiers structurés au format JSON. Chaque fichier contient une liste de documents correspondant à des textes historiques individuels, accompagnés de métadonnées et de champs textuels issus de la reconnaissance optique de caractères (OCR). Le texte OCR constitue la matière première du traitement et présente une grande variabilité en matière de longueur, de structure et de qualité.

Plusieurs problèmes typiques liés à l'OCR ont été identifiés. Les textes peuvent contenir des artefacts de reconnaissance, tels que des caractères incorrects ou obsolètes, des balises ou

fragments de balisage résiduels, ainsi que des coupures de mots ou des espacements irréguliers. La longueur des documents est également très hétérogène, allant de textes courts à des documents de plusieurs centaines de pages. Ces caractéristiques rendent nécessaire une phase de préparation des données avant tout traitement automatique ultérieur.

Transformation des données

Avant leur soumission au modèle de langage, les données JSON sont transformées en documents textuels individuels. Chaque enregistrement est extrait de la structure JSON et converti en un format textuel autonome, permettant un traitement unitaire et une traçabilité claire entre les textes sources et les résultats produits.

Une phase de nettoyage léger est ensuite appliquée au texte OCR. Cette étape vise à supprimer les artefacts évidents susceptibles de perturber le traitement automatique, tels que certaines balises résiduelles, des entités encodées ou des irrégularités typographiques fréquentes dans les corpus anciens. Le nettoyage est volontairement conservateur. L'objectif n'est pas de normaliser linguistiquement les sources, mais de stabiliser leur forme afin de garantir une entrée cohérente et lisible.

À l'issue de cette transformation, chaque document est associé à un identifiant unique et stocké dans un format structuré compatible avec les étapes ultérieures du pipeline, notamment la génération des résumés et leur importation dans la base de données publique. Cette procédure assure une correspondance univoque entre les données d'entrée, les traitements appliqués et les fichiers de sortie, contribuant ainsi à la traçabilité et à la reproductibilité du dispositif.

Pipeline de résumé

La production des résumés repose sur un pipeline structuré en plusieurs étapes successives, combinant des traitements déterministes et l'intervention contrôlée d'un modèle de langage génératif. Cette organisation permet de traiter des textes de longueur variable tout en garantissant la fidélité au contenu source et la stabilité des résultats.

Segmentation des textes

Les modèles de langage présentent des limites de contexte qui empêchent le traitement direct de textes très longs. Afin de surmonter cette contrainte, les documents sont préalablement découpés en segments de taille contrôlée. La segmentation est réalisée par des règles déterministes, indépendantes du modèle génératif, et respecte autant que possible la structure logique des textes.

Cette étape prépare les données pour l'intervention ultérieure du modèle de langage, sans introduire de transformation sémantique.

Résumés partiels

Le modèle de langage génératif intervient à ce stade pour la première fois. Chaque segment textuel est soumis individuellement au modèle, accompagné d'instructions explicites définissant le cadre du résumé. Le modèle est utilisé exclusivement comme outil de synthèse lexicale : il produit un résumé partiel fondé uniquement sur le contenu du segment fourni en entrée.

Les contraintes imposées au modèle portent notamment sur la longueur maximale du résumé, la neutralité du ton et l'interdiction de toute interprétation ou ajout d'information

externe. Le rôle du modèle génératif est ainsi strictement encadré et limité à la condensation contrôlée de l'information textuelle.

Recomposition du résumé final

Le modèle de langage génératif intervient une seconde fois lors de la recomposition du résumé final. Les résumés partiels produits précédemment sont agrégés et soumis au modèle afin d'obtenir une synthèse globale cohérente du document initial.

Lorsque la taille cumulée des résumés intermédiaires dépasse les limites autorisées, la recomposition est effectuée de manière itérative, par regroupements successifs. À chaque étape, le modèle génératif est utilisé pour condenser l'information sans en modifier le contenu factuel. Un contrôle final de la longueur et de la cohérence est appliqué afin de garantir un résumé unique, proportionné et fidèle au texte source.

Formats de sortie et intégration

Les résumés finaux sont générés dans un format structuré conçu pour être directement compatible avec les procédures d'importation de la base de données. Chaque résumé est associé de manière univoque à l'identifiant du texte source correspondant, garantissant une correspondance 1:1 entre les documents originaux et les contenus produits. Des vérifications systématiques sont appliquées avant l'intégration, portant notamment sur l'encodage des caractères, la présence des champs requis et le respect des contraintes de longueur, afin d'assurer la cohérence, la stabilité et l'intégrité des données au sein du système d'information.

Limites et précautions

Les résumés produits dans le cadre de ce projet sont strictement non interprétatifs et se limitent à une synthèse factuelle des informations présentes dans les textes sources. Leur qualité dépend en partie de celle des données OCR en entrée ; toutefois, l'expérience acquise montre que les modèles de langage utilisés sont capables de compenser de nombreuses lacunes liées à une reconnaissance imparfaite, notamment en reconstruisant des mots ou segments manquants sur la base de régularités linguistiques et contextuelles propres aux textes issus de l'OCR.