Another simple MySQL script to build the GeoNames database, with an efficient structure and indexes, based on the exportable dumps and on the postal codes dump . Please find the scripts and the documentation on GitHub
.

A user: a need

Download and unzip dumps according to your needs.

Main information: cities, feature codes and hierarchy

- download.geonames.org/export/dump/cities1000.zip
- download.geonames.org/export/dump/cities5000.zip
- download.geonames.org/export/dump/cities15000.zip

All GeoNames information should be relevant or not for you. By using cities tables (instead of allCountries table), you will upload only feature classes that are at least a small village (to a large metropole as a capital) with information on the population. Data between this three tables are redondante (cities in cities5000 are also in cities1000).

First 10 feature classes in the cities1000 table

```
   SELECT       a.fclasscode, COUNT(a.geonameid) AS NbGeoNamesId, b.name  FROM
geo_01cities1000
AS
a
INNER JOIN
geo_featurecodes
   AS b ON a.fclasscode = b.code  GROUP BY a.fclasscode  ORDER BY NbGeoNamesId DE
SC

LIMIT
0
,
10
;
```

P.PPL	69 408	populated place
P.PPLA3	27 195	seat of a third-order administrative division
P.PPLA4	26 600	seat of a fourth-order administrative division
P.PPLA2	16 160	seat of a second-order administrative division
P.PPLA	3 483	seat of a first-order administrative division
P.PPLX	2 336	section of populated place
P.PPLC	242	capital of a political entity
P.PPLL	239	populated locality
P.PPLQ	19	abandoned populated place
P.PPLG	14	seat of government of a political entity

- download.geonames.org/export/dump/featureCodes_en.txt
- download.geonames.org/export/dump/countryInfo.txt
- download.geonames.org/export/dump/admin1CodesASCII.txt
- download.geonames.org/export/dump/admin2Codes.txt
- download.geonames.org/export/dump/hierarchy.zip

To create the data model and upload the data (without allcountries table), use the first sql scriptGeoNames_01_create_db
. Pay attention:

- you will have warnings for the elevation column in the three tables for cities due to missing values. As we are using NOT NULL as default, the column elevation will receive '0' instead of having a missing value. You may change this behaviour.
- we are using CHARSET=utf8 COLLATE utf8_unicode_ci. You may consider using utf8mb4 (especially for alternateNames table or alternatenames columns).
- The data engine used here is MYISAM. Change this according to your Data management system.

Unsing zip/postal codes

- download.geonames.org/export/zip/allCountries.zip

Note: you will find all the regular expressions to collect zip codes in the table countryinfo in the two colomns:

- postalCodeRegex : list of all the detailed regular expressions by countries (if exists) to collect postal / zip codes

- postalCodeFormat : basic patterns of the postal / zip codes

Set up all feature classes

For those who want all information, you can download, unzip, and build a table with all geonames feature classes. This table have the same structure than cities1000, cities5000 and cities15000, and share some of information that is also listed in these three tables.

- download.geonames.org/export/dump/allCountries.zip

First 10 feature classes in the allcountries table

```
SELECT a.fclasscode, COUNT(a.geonameid) AS NbGeoNamesId, b.name FROM
geo_allcountries
AS
a
INNER JOIN
geo_featurecodes
AS
b
ON a.fclasscode = b.code GROUP BY a.fclasscode ORDER BY NbGeoNamesId DESC
LIMIT
0
,
10
;
```

P.PPL	4 100 754	populated place
H.STM	862 859	stream
T.MT	391 873	mountain
T.HLL	367 942	hill
S.FRM	322 320	farm
S.SCH	278 728	school
H.LK	266 213	lake
S.CH	246 672	church
S.HTL	241 482	hotel
H.STMI	200 196	intermittent stream

To add the allcountries table to the data model, use the second sql GeoNames_02_allcountries .

Pay attention, allcountries table:

- store more than 2.5 go of information (with indexes)
- all feature classes are in it (for some uses, it can add a lot of false positives)
- you will have warnings for the elevation column due to missing values. As we are using default NOT NULL, the column elevation will receive '0' instead of having a missing value. You may want to change this behaviour.
- we are using CHARSET=utf8 COLLATE utf8_unicode_ci. You may consider using utf8mb4 (especially for alternateNames table or alternatenames columns).
- The data engine used here is MYISAM. Change this according to your Data management system.

Added variables

The SQL scripts will add one column (and an index) to the raw GeoNames data model: fclasscode, to make you able to directly link the three cities tables and the allcountries table with the fclasscode table, without having to concatenat fclass and fcode. In other words to find easily feature classes labels for these tables.

In the allcountries table with all GeoNames Id, some feature codes are missing (in the featurecodes classification) or are not enough detailed (information is available only at the class level).

```
SELECT a.fclasscode, COUNT(a.geonameid) AS NbGeoNamesId, b.name FROM
geo_allcountries
AS
a
LEFT JOIN
geo_featurecodes
AS
b
ON
a
.
fclasscode
=
```

```
b
.
code

WHERE

b
.
code
IS
NULL

GROUP BY
a
.
fclasscode
;
```

S.	89591
.	5076
H.	497
U.	194
L.	31
P.	30
R.	12
T.	11
S.ARCHV	10
A.	4
S.TRAM	3

For the classes and the feature codes listed above (except for S.ARCHV and S.TRAM, assuming it may change between versions), the fclasscode is filed by an empty value '':

```
   UPDATE `geo_allcountries`   SET   `fclasscode` = ''   WHERE      `fclasscode` NOT
REGEXP                                                                    '[A-
Z]{1}.[A-Z]{1,5}
'
;
```

Excluded dumps

- alternateNames.zip (>100mo) alternative names for all the populated places (from allCountries.zip)