# load_contactscategories_jsonl_to_mysql.py - a script to load website contacts & categories data into MySQL

*WhoisXML API, Inc. 2019.*

## Table of Contents


1. About the script
    1.1. A usable demonstration script
    1.2. Cross-platform
2. Prerequisites
    2.1. Python libraries
    2.2. MySQL settings
3. How to use
    3.1. Setting up a mysql database
4. Loading data with the script
5. Limitations
6. Performance notes


version 0.0.1

## 1. About the script

This script is intended for subscribers of Whois XML's Website Contacts & Categories database download product (http://website-contacts-database.whoisxmlapi.com) to help loading downloaded jsonl files into a MySQL database.

### *1.1. A usable demonstration script*

The script is intended to be a programming example of how to accomplish this task and also to be useful as it is. For this reason the script is not perfectly robust, there are no checks of the validity of the input files and the exception handling is also not very detailed, to avoid a complex and less readable code. If used properly, however, it is can be used efficiently.

### *1.2. Cross-platform*

The script has to work on any system where Python and the necessary libraries are available. It has been tested on Ubuntu Linux and Microsoft Windows, but on other platforms such as Mac OS X it should work, too.

# 2. Prerequisites

## *2.1. Python libraries*

The loader script is written in Series 3 Python; it was tested with Python 3.6.7. on Linux and Python 3.7.2 on Windows. It uses the following libraries:

- Pandas (https://pandas.pydata.org/) , a data analysis library, in order to efficiently load chunks of jsonl files.
- MySQL connector: the library to access MySQL databases, provided by Oracle.

While Pandas can be simply installed with the package manager ("pip install pandas"), the vanilla MySQL connector is available from its download web-page (dev.mysql.com/downloads/connector/python/) . On some systems you may install both with the package manager of your OS (e.g. with "apt" on Debian-flavor Linuxes, including Ubuntu and Mint).

## *2.2. MySQL settings*

As the data can contain Unicode characters to be stored on 4 bytes, while MySQL uses a 3-byte encoding by default, it is recommended to enable the 4-byte Unicode system-wide, by adding the lines

```
character-set-server = utf8mb4
collation-server = utf8mb4_unicode_ci
```

to the configuration file of the MySQL server (e.g. /etc/mysql/mysql.conf.d/mysqld.cnf on Ubuntu systems), and restarting the service. Please consult the documentation of MySQL if your prefer not to modify this setting system-wide.

# 3. How to use

We demonstrate the use of the script using the demonstration data available from the website. There art two kinds of datasets available:

**Domain names only**

These contain domain names, country codes and categories to which the given site belongs to. A site can belong to multiple categories.

**Domain names and contact information**

these contain full information including e-mails, postal addresses, social network links, etc. as described in the product specs:
https://website-contacts-database.whoisxmlapi.com/specifications

The script supports both types of files. In this demonstration we assume that the two sample files: categories_database_sample.jsonl (domain names only) and contacts_database_sample.jsonl (full data) are to be loaded into a MySQL database. In a production environment the desired files should be used instead.

The script automatically detects and decompresses gzip compressed files, so if you have downloaded a large file in this format, there is no need to uncompress it in advance.

The following description has been prepared on a Ubuntu Linux system. The script works on Windows, too, provided that Python, MySQL and the necessary packages are installed. In the Windows command-line the "./" before the script's name is not needed.

### 3.1. Setting up a mysql database

We create a user and a database for the purpose: as a root user we do

```
create user websitecc identified by 'websitecc';
create database websitecc;
grant all on websitecc.* to websitecc;
```

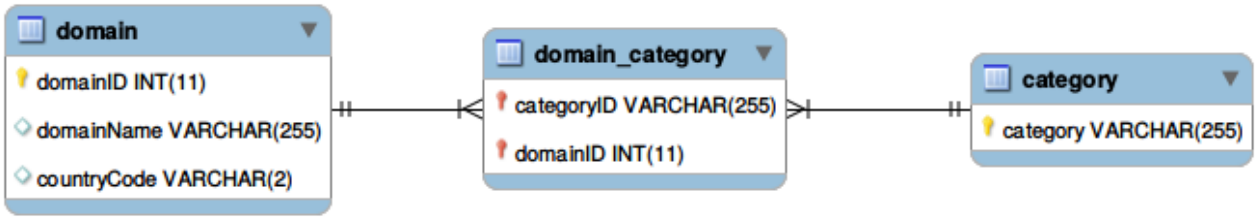
Next we create the schema for loading data. For this reason we run the appropriate ddl file supplied for this script:

```
mysql --user=websitecc --password=websitecc --database=websitecc <
website_categories.ddl
```

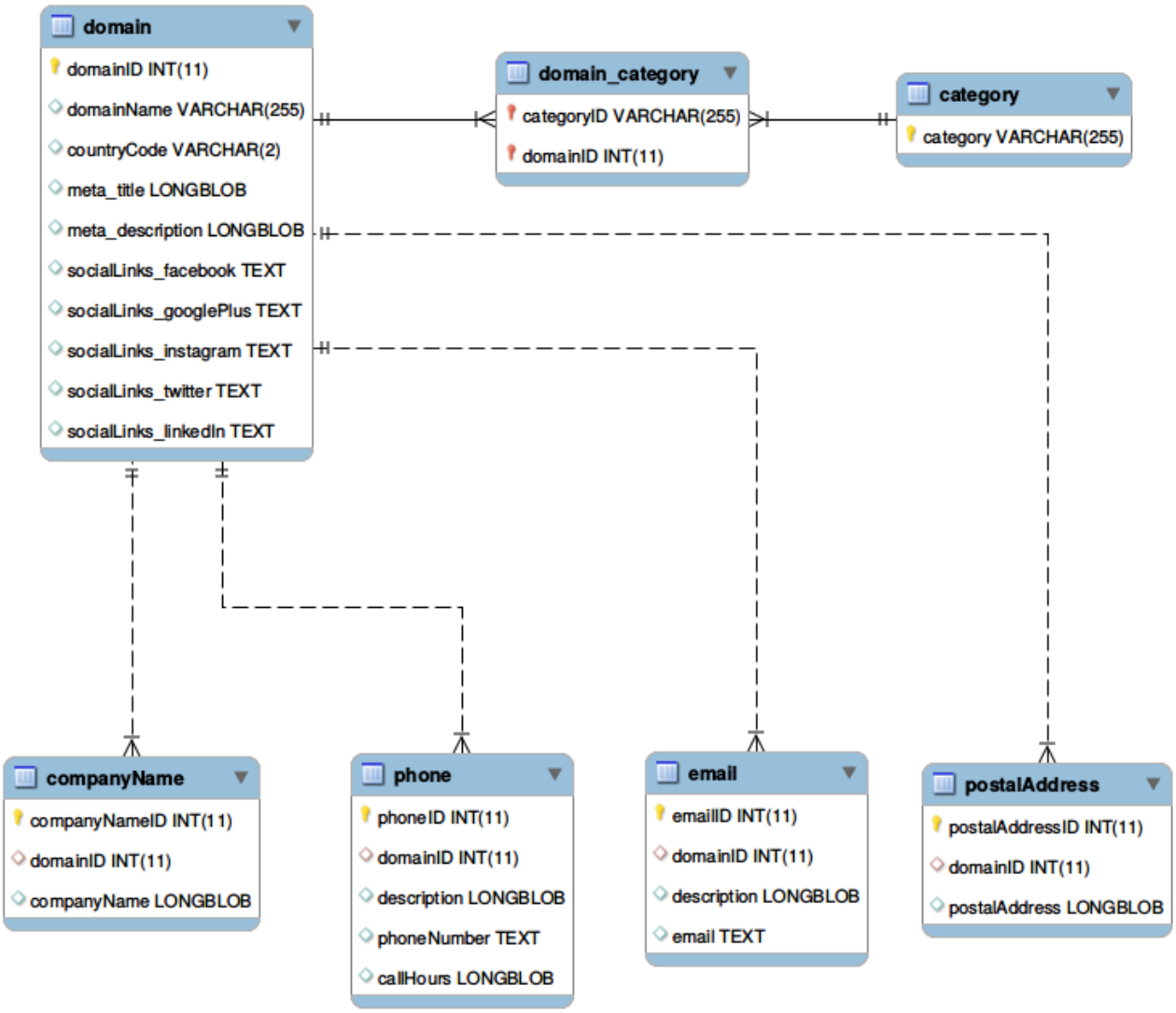
for domain-names only data or

```
mysql --user=websitecc --password=websitecc --database=websitecc <
website_contacts_categories.ddl
```

for complete data. This will create a simple schema with an n-m connection in the domains only case:




and a similar schema with a more detailed "domain" table having more children when contact data are also included:




(Note: we use TEXT fields because of the unpredictable lengths and LONGBLOBs because of the unpredictable character sets. This should be taken into account when putting indices on these fields.)

## 4. Loading data with the script

Once the database has been prepared as described, and the data have been downloaded, they can be loaded into the database with the script. The script is self-documenting, it supports the –help option:

```
./contactscategories_jsonl_to_mysql.py --help
```

gives information about the syntax and a full list of currently supported options.

A typical way of loading domain names only is

```
./load_contactscategories_jsonl_to_mysql.py \
--jsonl-file categories_database_sample.jsonl --domain-names-only  \
--mysql-password websitecc --mysql-database websitecc \
--mysql-user websitecc
```

whereas loading full data can be done with

```
./load_contactscategories_jsonl_to_mysql.py \
```

```
--jsonl-file contacts_database_sample.jsonl  \
--mysql-password websitecc --mysql-database websitecc \
--mysql-user websitecc
```

Notes:

- The –domain-names-only option can be used with files having full information and/or schemata with full information. In this case the contact information shall be ignored. The other way around, trying to load domain names only files without this option will result in error messages and malfunction.
- The script loads lines of the files in chunks. Commits occur after each chunk; this is a typical approach in relational data population. The size of chunks can be tuned with the –chunksize option, and the number of chunks to be loaded can be limited by the –nchunks option.
- Foreign key checks are turned off at the beginning and turned on at the end for better performance.
- The categories are inserted and updated dynamically.
- If you encounter "Memory error", decreasing the chunk size can help. This error occurs when you have a huge file to load.

# 5. Limitations

The script is mainly for demonstration, however it can be used in practice. It has, however, the following limitations.

- The schema is hard-coded; the table names should be the as in the provided ddl-s. It can be easily customized by rewriting the script.
- The script can only populate databases, apart from the categories it does not verify whether a record already exists.
- It reads records with children line by line; thus e.g. e-mails belonging to multiple records will be duplicated. It violates normalization. In the future there will be a mode of the script to overcome this, but it will be optional due to performance reasons: in many cases it is more efficient to load the data as they are and take account the unnormalized nature.
- The script does not yet support multi-threaded operation.

# 6. Performance notes

During the testing with a large file we had the following experience.

The test was run on Ubuntu 18.04.1 LTS, mysqld Ver 5.7.24-0ubuntu0.18.04.1 for Linux on x86_64 (Ubuntu), Python 3.6.7, pandas 0.23.4, mysql.connector 2.1.6; on a with Intel(R) Core(TM) i7-7700 CPU @ 3.60GHz, 4 gigabytes of RAM, running in a Virtualbox environment hosted on the same version of Linux, on a Dell Precision 3620 Mini Tower workstation.

Loading 12921323 records of contacts and categories from a gzipped jsonl file of size 951M took about 3 hours.

We also remark that under Windows 10 on the same (virtual) hardware we encountered "Memory error", so probably a Windows system needs more memory for this task.