COMTRADE Bulk Download and Loading Local DB Scripts

Ref: IHC/01/5209/2020

Luis Puerto

15/06/2020

This a small collection of scripts to download and load data from UN COMTRADE database bulk API to a local MySQLd database hosted on an EFI's local machine for its posterior cleaning.

Since the cleaning procedure is going to take place mainly on the BioUnit WorkStation inside of the EFI Headquarter premised, the scripts are configured to work on that machine. They can be be easily reconfigured to work on another machine if necessary.

Prerequisites

Token and access to UN COMTRADE

You need to have access to the UN COMTRADE bulk API either through your IP or using a token, since this API is a premium access API and it's not publicly available. You can get a token on this site if you are accessing to the Internet using one of the EFI's IP allowed to access the premium API by COMTRADE. If you are on the EFI headquarters' premises you are good to go, but if you are under a VPN, or any other location things can be different and a token is recommended to be used.

Software

The scripts are bash scripts so they have to be run on a bash terminal or on a compatible one. They were developed on Ubuntu 20.04. These scripts relay on:

- wget: to download the files and interact with the API.
- zip: to unzip the files.

Regarding the MySQL, it has to be installed on the system beforehand and properly configured. On the host machine MySQL 8 was installed and it the one that has been used with these scripts. Other previous versions most probably will work, but it haven't been tested.

Installation

To download the scripts, you just need to perform a git clone running on the terminal something like this:

```
git clone git@github.com:EuropeanForestInstitute/comtrade-scripts.git
```

Remember that you have to have set a proper ssh access on the host machine and configure it on your GitHub's account that are inside EFI's GitHub organization, since this scripts are kept private.

Sourcing the scripts and the functions inside them

The scripts have the ability to locate themselves on the system so they don't need to be sourced unless you need to use the functions alone. Scripts can be source running something like this on terminal:

```
$ source /path/to/the/script/you/want/to/source.sh
```

Then, you can use the functions inside that script on that terminal session. If you close the terminal or open another terminal session you need to source the script again.

For example, if you just want to download the yearly data and nothing else in the current host machine you can proceed as follows on the current setup:

```
$ source /home/smartforest/Code/tradeflows/comtrade-scripts/downloaddata.sh
$ bulkdl_y
```

Configuration

config.sh

You can configure some variables that are used in the scripts editing the config file `config.sh` that is located in the root of this project. You can configure:

- **Token**, if you have one.
- **Download path** where the data downloaded is going to be stored. Currently on the `~/Downloads` folder of the `smartforest` user on the host machine.
- Location of the **cleaning script**.
- **Years** of data to be download from UN COMTRADE, for the yearly and monthly scripts.

- Name of the target **database** and the name of **tables**.*

*These variables are given by the Trade Flows project and should not be changed unless they are also changed on the R code.

MySQL configuration

MySQL has to have configured the proper username and password beforehand. These access credentials are also given by the Trade Flows project. It's recommended to grand all privileges `GRANT ALL ON tradeflows.* TO 'R'@'localhost';` on the target database, but perhaps this could be assessed and take a more granular access to privileges, since the only actions that need to be performed by that username are:

- Create a database
- Create tables
- Truncate tables
- Drop tables
- Rename tables
- Write tables
- Purge binary logs

To get database credentials to work properly and safely from bash –where these scripts are going to be run– the following command has to be run beforehand, so the access password to the database is stored safely in a config file locally. The command is going to prompt you to input the password for that username.

```
$ mysql_config_editor set --login-path=local --host=localhost --user=username --password
```

Making the scripts executable

To be able to run the scripts, specially the automation script `dlrldata.sh` they need to be executable files. To archive that, for example the following command can be run on the terminal on the root of the project.

```
$ chmod +x dlrldata.sh
```

Functions

The functions are dummy simple functions with not options or variables to be set.

To download data

The functions related to download the data and handling it before it's loaded to the database are located on `downloaddata.sh`. This is a brief description of those functions.

`bulkdl_ctd_y`: It will download all yearly data files available between the given years on the config file and on the configured download folder. The download folder it will be created if not present. If there is a file with the same name as a file that is going to be downloaded the new file will be renamed.

`bulkdl_ctd_m`: It will download all monthly data files available between the given years on the config file and on the configured download folder. The download folder it will be created if not present. If there is a file with the same name as a file that is going to be downloaded the new file will be renamed.

`unzip_ctd_y`: It will unzip all the downloaded yearly zip files and it will delete them if the unzipping is successful.

`unzip_ctd_m`: It will unzip all the downloaded monthly zip files and it will delete them if the unzipping is successful.

`del_ctd_y`: It will delete the yearly download folder if present. This function it's useful to run it before running the downloading function so there are not duplication of files on that folder.

`del_ctd_m`: It will delete the monthly download folder if present. This function it's useful to run it before running the downloading function so you don't end up with duplicated data in form of renamed files.

To load data into the database

`c_tradeflows_db`: Creates the target database.

`c_comtrade_tb`: Creates the target table to load the full COMTRADE database.

`l_comtrade_yearly_data`: Loads the COMTRADE data to the `comtrade` table.

`dl_comtrade_tb`: Delete the COMTRADE table if it's present.

`db_clean_logs`: Delete the binary logs of the database that sometimes cause the use of a huge amount of disk space.

`dl_raw_flow_yearly_tb`: Delete the `raw_flow_yearly` table if there is present one.

`c_raw_flow_yearly_tb`: Creates a subset of data with the forestry chapters on a table named `raw_flow_yearly`. This table is going to be the source of data of the cleaning process.

`a_raw_flow_yearly_tb`: Archives the table `raw_flow_yearly` if there is one appending to the name of the table the current date and time on the form `_a_YYYYMMDDHHMM`.

`c_validated_flow_yearly_tb`: Creates the target table for the cleaning process with name `validated_flow_yearly`.

Table Schemata The functions that interact with the database and create tables apply a series of table schemata —in other words, which columns are created and the data type for each of them— that are given by the data itself and the cleaning process. They can't be changed unless they are also changed in the Trade Flows project, specially the names and the data type of some of the columns since they are checked if they are present during the cleaning process. If they are not, the cleaning process will not go through.

Automation file

The file `dlrldata.sh` is mean to run all the needed functions in sequence. It will also run the cleaning process if a path to the cleaning R script is provided. This script will perform the following actions through functions:

1. Delete the download folder for the yearly data in case there were any previously downloaded file.
2. Download all the files for the yearly data for the configured years.
3. Unzip all the downloaded files and delete them if the unzipping is successful.

4. Create the `tradeflows` database if not present
5. Delete the `comtrade` table if present.
6. Clean the binary logs.
7. Create the `comtrade` table.
8. Load the COMTRADE yearly data to the `comtrade` table.
9. Archive the `raw_flow_yearly` table if present.
10. Create a new `raw_flow_yearly` table with the new loaded data.
11. Create the `validated_flow_yearly` table if not present.
12. Trigger the cleaning process if the path to the script is set.

Cleaning process The cleaning process can also be added to the automation if the patch to the cleaning script is set on the config file.

A cleaning log will also be generated and storage in `/home/smartforest/Code/tradeflows/cleaninglogs/`

Scheduling

The automation script can be schedule using `cron`, which is a time-based job scheduler utility present in most in Unix-like systems like Linux. Since the downloading and loading process are time consuming —around 8 hours for the downloading and almost a whole day for the loading— it's recommended to schedule this procedure on a Friday in the evening so all the new data will be available on Monday morning.