1. Instructions to execute the standardize FAERS data and generate safety signals ETL process

LTS Computing LLC

<http://www.ltscomputingllc.com/>

The source code (shell scripts and SQL scripts) developed by LTS Computing LLC to execute this process is available as open source under the Apache 2.0 license on [github](https://github.com/ltscomputingllc/faersdbstats)

* 1. System Prerequisites

1. Linux. This process was developed and executed on Ubuntu LTS 14.04.
2. OHDSI Usagi mapping tool
3. PostgreSQL database. This process was developed and executed on PostgreSQL 9.3.
4. Pgadmin III PostgreSQL SQL client. Use this client to run the SQL scripts.
   1. Reference Data Prerequisites

* Load the FDA Orange book file. NDA reference data file into a PostgreSQL table called “nda” to look-up ingredients by NDA number
* Download the data file from the [FDA Orange book website](http://www.fda.gov/Drugs/InformationOnDrugs/ucm129689.htm)
  + Run the following SQL script to load the data file:
    - load\_nda\_table.sql
* Load the country\_codes reference table by running the script:
  + load\_country\_code\_table.sql
* Load the EU drug name/active ingredient reference table by running the script:
  + load\_eu\_drug\_name\_active\_ingredient.sql
* Load the latest available OHDSI CDMV5 Vocabulary tables into a separate PostgreSQL schema (e.g. cdmv5). The vocabulary data files and the PostgreSQL table load scripts are available from the [OHDSI Athena website](http://www.ohdsi.org/web/athena/) On the Athena website vocabulary download page keep the pre-selected vocabularies and also select the MedDRA vocabulary (review the EULA link for MedDRA).
* Create the meddra / snomed mapping table by running the script:
  + create\_meddra\_snomed\_mapping\_table.sql
    1. Download current FAERS data files from the FDA website

Run the current FAERS data download shell script:

* download\_current\_files\_from\_faers.sh
  + 1. Combine the current data individual files into combined files to load

Run the following current data shell scripts:

* create\_current\_all\_demo\_data\_files\_with\_filename\_column.sh
* create\_current\_all\_drug\_data\_files\_with\_filename\_column.sh
* create\_current\_all\_indi\_data\_file\_with\_filename\_column.sh
* create\_current\_all\_outc\_data\_file\_with\_filename\_column.sh
* create\_current\_all\_reac\_data\_files\_with\_filename\_column.sh
* create\_current\_all\_rpsr\_data\_file\_with\_filename\_column.sh
* create\_current\_all\_ther\_data\_file\_with\_filename\_column.sh
  + 1. Create current data staging tables and Load data from each file

Run the following SQL scripts:

* load\_current\_demo\_table.sql
* load\_current\_drug\_table.sql
* load\_current\_indi\_table.sql
* load\_current\_outc\_table.sql
* load\_current\_reac\_table.sql
* load\_current\_rpsr\_table.sql
* load\_current\_ther\_table.sql
  + 1. Download legacy LAERS data files from the FDA website

Run the legacy FAERS data download shell script:

* download\_legacy\_files\_from\_faers.sh
  + 1. Combine the legacy data individual files into combined files to load

Run the following legacy data shell scripts:

* create\_legacy\_all\_demo\_data\_files\_with\_filename\_column.sh
* create\_legacy\_all\_drug\_data\_files\_with\_filename\_column.sh
* create\_legacy\_all\_indi\_data\_file\_with\_filename\_column.sh
* create\_legacy\_all\_outc\_data\_file\_with\_filename\_column.sh
* create\_legacy\_all\_reac\_data\_file\_with\_filename\_column.sh
* create\_legacy\_all\_rpsr\_data\_file\_with\_filename\_column.sh
* create\_legacy\_all\_ther\_data\_file\_with\_filename\_column.sh
  + 1. Create legacy data staging tables and Load data from each file

Run the following SQL scripts:

* load\_legacy\_demo\_table.sql
* load\_legacy\_drug\_table.sql
* load\_legacy\_indi\_table.sql
* load\_legacy\_outc\_table.sql
* load\_legacy\_reac\_table.sql
* load\_legacy\_rpsr\_table.sql
* load\_legacy\_ther\_table.sql
  1. De-duplicate cases

The aim of this section of the process is to remove duplicate cases.

* + 1. Single value imputation of demographic keys followed by removal of duplicate cases first based on demographic key fields and then based on case id.
* Run the **derive\_unique\_all\_case.sql** script SQL script
  1. Map drug names to RxNorm

The aim of this section of the process is to create a combined\_drug\_mapping table with the original drug name and the associated RxNorm concept\_id.

* Run the **map\_all\_drug\_name\_to\_rxnorm.sql** script SQL script
  + 1. a) Map current data drug name to RxNorm - Regular Expression
* The regex mapping section of the SQL script uses regular expression mapping of current data drug table drugname to the cdmv5.concept\_table concept\_name and generates the drug\_regex\_mapping table:
* Note. This will also create the combined\_drug\_mapping table with all unique current data drugnames, setting the update\_type column to 'regex - <regex type>' and the concept\_id to the matching (RxNorm) concept\_id in the drug\_regex\_mapping table.
  + 1. b) Map current data drug name to RxNorm – Active Ingredient
* The Active Ingredient number mapping section of the SQL script generates the drug\_ai\_mapping table
* Note. This will also set the combined\_drug\_mapping table update\_type column to 'drug active ingredients' and the concept\_id to the matching (RxNorm) concept\_id in the drug\_regex\_mapping table.
  + 1. c) Map current data drug name to RxNorm – NDA Number
* The NDA number mapping section of the SQL script generates the drug\_nda\_mapping table
* Note. This will also set the combined\_drug\_mapping table update\_type column to 'drug nda\_num ingredients' and the concept\_id to the matching (RxNorm) concept\_id in the drug\_ai\_mapping table.
  + 1. d) Map current data drug name to RxNorm - USAGI
* Run the create\_drug\_usagi\_mapping\_table.sql SQL script to create the drug\_usagi\_mapping table.
* Run the create\_usagi\_import\_table.sql SQL script to create the usagi\_import table.
* Run the generate\_drug\_export\_for\_usagi.sql SQL script to generate the set of unmapped codes.
* Export the set of unmapped codes to a file using pgadmin client export functionality.
* Load the file into the USAGI tool for manual mapping.
* Open the USAGI tool. Follow the instructions on the [USAGI OHDSI wiki page](http://www.ohdsi.org/web/wiki/doku.php?id=documentation:software:usagi) to initialize and load the data file into USAGI. Note. As part of USAGI setup you will need the CDMV5 vocabulary data files from the Athena website.
* Manually map some % of the drug names to RxNorm concept\_ids in USAGI prioritized by descending frequency of occurrence and export the results to a file.
* Export the mapped codes from USAGI to a file.
* Load the USAGI mapped codes file into the usagi\_import table using pgadmin client import functionality
* Run the load\_drug\_usagi\_mapping.sql SQL script to populate the drug\_usagi\_mapping table
* Note. The USAGI section of the **map\_all\_drug\_name\_to\_rxnorm.sql** script SQL script will set the combined\_drug\_mapping table update\_type column to 'usagi' and the concept\_id to the matching (RxNorm) concept\_id in the drug\_usagi\_mapping table.
  1. Standardize RxNorm concepts to standard RxNorm concepts of Ingredient or Clinical Drug Form

The aim of this section of the process is to create a standard\_drug table with the original drug name and the associated standard RxNorm concept\_id.

* Run the **standardize\_combined\_drug\_mapping.sql** script
  1. Derive standard outcome category (FAERS outcome)

Derive the high level FAERS outcome categories (death, disability etc) for the current data standard\_drug table

* Run the d**erive\_standard\_case\_outcome\_category.sql** script
  1. Derive standard drug indication from MedDRA preferred term to MedDRA concept\_id

Convert the FAERS indication MedDRA preferred terms into MedDRA concept ids (limited to the cases in the unique\_all\_case table) in a new table called standard\_case\_indication.

* Run the **derive\_standard\_case\_indication.sql** script
  1. Derive standard case outcome (FAERS reaction) from MedDRA preferred term to MedDRA concept\_id

Convert the FAERS reactions (adverse event outcomes) MedDRA preferred terms into MedDRA concept ids (limited to the cases in the unique\_all\_case table) in a new table called standard\_case\_outcome.

* Run the **derive\_standard\_case\_outcome.sql** script
  1. Derive standard drug outcome (FAERS reaction) counts

Create drug/outcome combination counts (counts for pairs of drug RxNorm concept\_id, outcome (reaction) MedDRA concept\_id) and store the counts in a new table called standard\_drug\_outcome\_count

* Run the **derive\_standard\_drug\_outcome\_count.sql** script
  1. Derive standard drug outcome (FAERS reaction) 2x2 contingency tables

Compute the 2x2 contingency table for each drug/outcome pari in a new table called standard\_drug\_outcome\_contingency\_table

* Run the **derive\_standard\_drug\_outcome\_contingency\_table.sql** script (Note, the run-time is around 6 hours)
  1. Derive standard drug outcome (FAERS reaction) statistics

Compute the PRR, ROR and associated 95% CI limit statistics in a new table called standard\_drug\_outcome\_statistics

* Run the **derive\_standard\_drug\_outcome\_statistics.sql** script.

Map MedDRA codes to SNOMED codes

Map drug indications and adverse event outcomes (FAERS reactions) and associated drug/outcome pair counts from MedDRA preferred terms to SNOMED-CT concepts

* Run the **map\_meddra\_to\_snomed.sql** script.

Create standard drug outcome drill down table

Create standard\_drug\_outcome\_drilldown table for use in joins to get all cases for a drug/outcome pair count

* Run the **derive\_standard\_drug\_outcome\_drilldown.sql** script.

***Copy standard output tables and source data tables***

Copy the standard output tables and the source data tables to the "standard" and "source" schemas. The final output from the whole process consists of the tables in the “standard” and “source” schemas. All the tables in the “standard” schema are prefixed with “standard\_” for consistency (they are the ‘standardized tables’). The “source” schema tables contain the original source data.

* Run the **copy\_tables\_to\_other\_schemas.sql** script.