

ETL with MySQL, PostgreSQL, and Bash

Assignment overview

You need to keep data synchronized between different databases/data warehouses as a part of your daily routine. One task that is routinely performed is the sync up of the transactional database and staging data warehouse. Automating this sync-up will save you time and standardize your process.

The Data Warehouse gets information from several sources including the transactional (OLTP) database. Transactional data from the OLTP database (in this case MySQL) needs to be propagated to the warehouse on a frequent basis. This data movement can be updated using ETL processes.

In this project, you will set up an ETL process using a shell script to extract new transactional data for each day from the MySQL database and load it into the staging data warehouse in PostgreSQL. You will set up a Cron Job to schedule these tasks.

Later you will perform the transformation on the table in the staging warehouse to create a dimension table and a fact table.

You will then export these tables as CSV files to the production warehouse.

You will begin preparing the lab environment by starting the **MySQL** server. You will then create a sales database in MySQL and load two csv files **sales_old_data.csv** which has an old timestamp and the **sales_new_data.csv** with the current timestamp incrementally in **sales_data table**. You will be executing a setup shell script which will automate the process of creating the database, table and loading the table.

Next, you will start your staging warehouse **PostgreSQL** server. You will then execute another setup script to create tables in PostgresSQL.

The final task requires you to automate the extraction of data from the sales database present in the transactional database (MySQL) and load it in the Staging Warehouse (PostgreSQL) using a **cron job**. You will download the shell script from the link provided in the lab and use it as a template. After performing each task, you will need to take a screenshot of the command you used and its output, and name the screenshot.