Hands-on Lab: Automating ETL Using Shell Scripts

Estimated time needed: **45** minutes

Scenario

You are a data engineer at an e-commerce company. 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.

In this project, you will set up an ETL process using **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 **fact** table.

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

Objectives

In this assignment, you will:

- Create a **shell script** to:
    - Extract the data from the **MySQL** database and load it into the **PostgreSQL** Staging warehouse
    - Transform the data and load in **DimDate** and **FactSales** table
    - Export the tables as CSV files for loading into the **production** warehouse.
    - Schedule a **cron job** to automate these tasks

Important Notice about this lab environment

Please be aware that sessions for this lab environment are not persistent. Every time you connect to this lab, it will create a new environment. Any data you may have saved in an earlier session will get lost. Please plan to complete this lab in a single session to avoid losing your data.

This Skills Network(SN) Labs Cloud IDE provides a hands-on environment for the course and project-related labs. It utilizes **Theia**, an open-source IDE (Integrated Development Environment) platform that runs on a desktop or the cloud. To complete this lab, you will use the Cloud IDE based on **Theia** running in a Docker container.

Software used in this project

In this project, you will use MySQL for data extraction and PostgreSQL Database for staging,transformation and loading. These relational database management systems (RDBMS) are designed to store, manipulate, and retrieve data efficiently.

Note - Screenshots

Throughout this lab, you will be prompted to take screenshots and save them on your device. You will need these screenshots to answer graded quiz questions or upload them as your submission for peer review at the end of this course. You can use various free screengrabbing tools or your operating system's shortcut keys to do this (for example, **Alt+PrintScreen** in Windows and **Shift+Command+3** for Mac).

Prepare the lab environment

Before you start the assignment:

1. Download the following CSV files from the URL:

   https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB0321EN-SkillsNetwork/ETL/sales_olddata.csv

   https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB0321EN-SkillsNetwork/ETL/sales_newdata.csv

2. Start MySQL server using the below button (or) from the SN tool box.

Open MySQL Page in IDE

3. Download the setup script from the link:

https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/geRwyvvpVRM-4QsI1HBmbw/setupmysqldb.sh

4. Go to **File->Open->Project** in the menu and double-click on the setupmysqldb.sh file to open it.
5. Add the password of your MySQL database in the file wherever it is saying <Replace with your mysqlserver password> and then save the file by selecting **File > Save**.

   Please make a note of the password as you will need it in the subsequent step.
6. Next, run the command `bash setupmysqldb.sh` in the **terminal** to execute the script

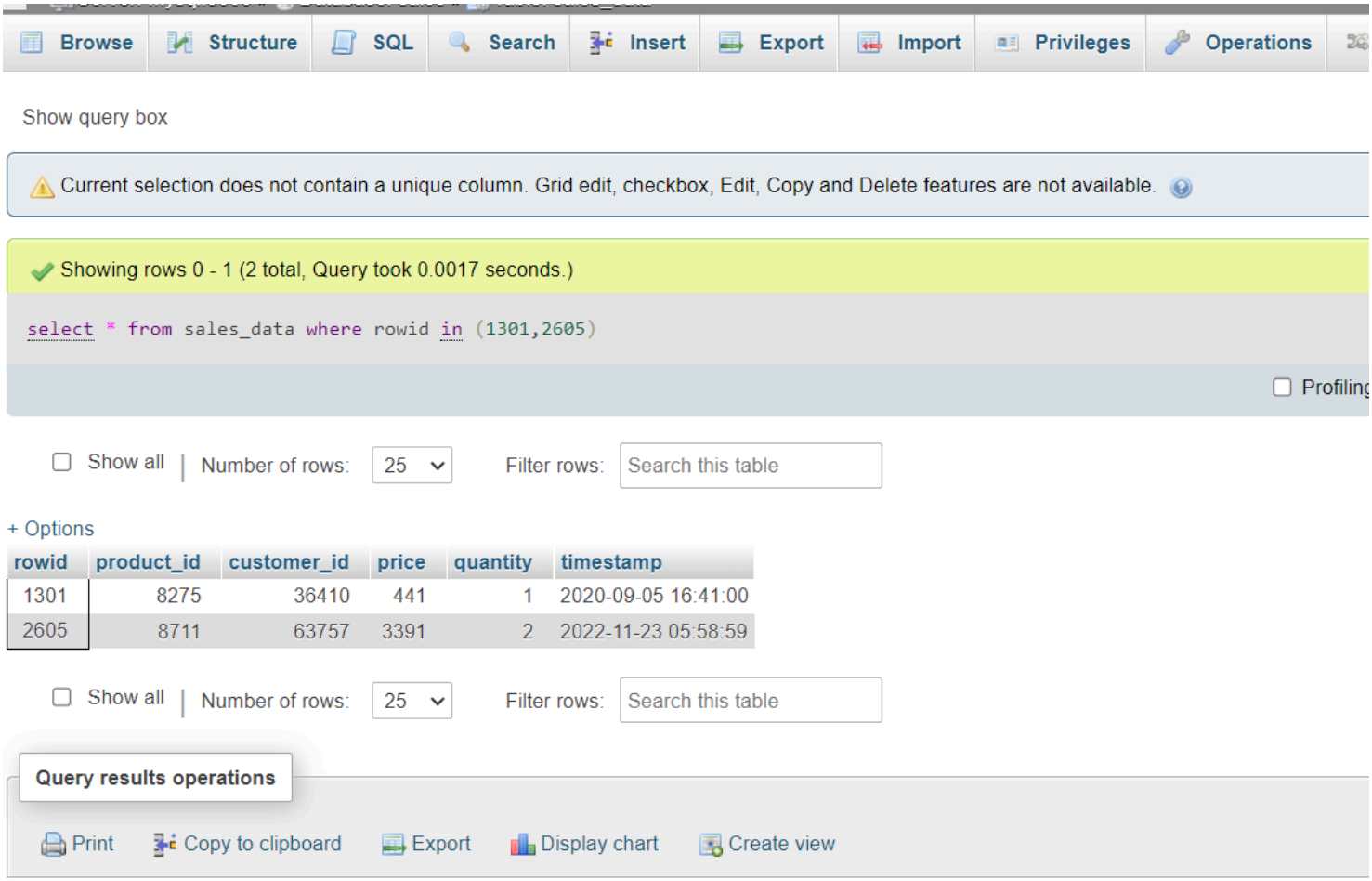
```
theia@theiadocker-lakshmih:/home/project$ bash setupmysqldb.sh
mysql: [Warning] Using a password on the command line interface can be insecu
re.
mysql: [Warning] Using a password on the command line interface can be insecu
re.
mysql: [Warning] Using a password on the command line interface can be insecu
re.
```

Please ignore the **password insecure** warning in the lab environment and proceed with the assignment.

Execute the query in MySQL UI to check whether the table **sales_data** is loaded with data.

```
select * from sales_data where rowid in (1301,2605)
```

Check whether you have got the output as displayed below:



7. Start PostgreSQL server

8. Download the setup script from the link:

https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/d0sRdGtbBILO5qeFBYsC8w/setuppostgresqldb.sh

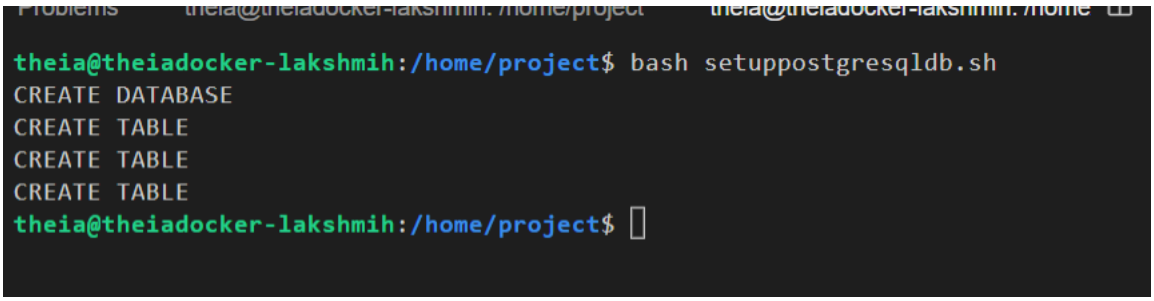
9. Go to **File->Open->Project** and double-click on the setuppostgresqldb.sh file to open.

Once the file is open, replace –host=localhost with **–host=postgres** on lines 3 and 5. Save the file by selecting **File > Save**.

10. Run the below command in the terminal by replacing <your password> with your postgres password that can be found under the connection information tab.

export PGPASSWORD=<your password>

11. Next, run the command `bash setuppostgresqldb.sh` in the **terminal** to execute the script.



Exercise 1

In further sections, you have to use the bash command to execute the shell script.

Extract the data from the MySQL database and load it into the PostgreSQL Staging warehouse

- Download the shell script named **ETL.sh** from the following location using `wget` command.

  https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/RMRYUo2Sxc6-kSjlq4gCzQ/ETL.sh

- Follow the instructions in the shell script to extract the data from the **transactional database(MySQL)** and load it in the **staging warehouse(PostgresSQL)**. ***(Here you are extracting new data)***.

  *Take a screenshot of the command used and name it **extract_load_data.png**. (Images can be saved with either the .jpg or .png extension.)*

- Check only data for last 24 hours is loaded in the sales_data table of the **Staging Data Warehouse** by executing the following query in PostgreSQL UI.

```
select count(*) from sales_data;
```

Exercise 2

Transform the data and load in DimDate and FactSales table

Task 1: Update the shell script `ETL.sh` to add a command to load the **DimDate** table.

The values of this table needs to be populated from the **sales_data** table by performing suitable **transformations**.

Execute the query below in PostgreSQL UI to check whether the table is populated with data.

```
select * from DimDate;
```

*Take a screenshot of the command used to perform the transformation and name as **DimDate.png** for DimDate table. (Images can be saved with either the .jpg or .png extension.)*

Task 2: Update the shell script `ETL.sh` to add a command to load the **FactSales** table.

Execute the query below in PostgreSQL UI to check whether the table is populated with data.

```
select * from FactSales;
```

The values of the table need to be populated from the **sales_data** table by performing suitable **transformations**.

*Take a screenshot of the command used to perform the transformation and name as **FactSales.png** for Factsales table. (Images can be saved with either the .jpg or .png extension.)*

Exercise 3

Export the tables as CSV files for loading into the production warehouse.

Task 1: Update the shell script `ETL.sh` to add a command to export **DimDate** and **FactSales** as **DimDate.csv** and **FactSales.csv**

*Take a screenshot of the contents of the exported csv files and name them as **exportDimDate.png** for the DimDate table and **exportFactSales.png** for the FactSales table. (Images can be saved with either the .jpg or .png*

Exercise 4

Schedule a cron job to automate these tasks

Task 1: Start the crontab.

Task 2: Write a command in the crontab editor to automate the tasks in the ETL.sh file every 24 hours.

*Take a screenshot of the command used and name it **schedule_job.jpg**. (Images can be saved with either the .jpg or .png extension.)*

Task 3: Start the cron service.

*Take a screenshot of the command used and name it **cron_start.jpg**. (Images can be saved with either the .jpg or .png extension.)*

To check whether cron is working properly you can schedule it to 30 minutes or 5 minutes.

Once the CSV files are extracted, remove the temporary tables DimDate and FactSales from the Staging warehouse.

End of assignment

Authors

Lakshmi Holla

Appalabhaktula Hema

Other Contributor(s)

Lavanya T S