Hands on Lab - ETL

Estimated time needed: **30** minutes.

About This SN Labs Cloud IDE

This Skills Network Labs Cloud IDE provides a hands-on environment for course and project related labs. It utilizes Theia, an open-source IDE (Integrated Development Environment) platform, that can be run on desktop or on the cloud. To complete this lab, we will be using the Cloud IDE based on Theia and MySQL database running in a Docker container. You will also need an instance of DB2 running in IBM Cloud.

Important Notice about this lab environment

Please be aware that sessions for this lab environment are not persistent. A new environment is created for you every time you connect to this lab. Any data you may have saved in an earlier session will get lost. To avoid losing your data, please plan to complete these labs in a single session.

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 staging data warehouse and production data warehouse. Automating this sync up will save you a lot of time and standardize your process. You will be given a set of python scripts to start with. You will use/modify them to perform the incremental data load from MySQL server which acts as a staging warehouse to the IBM DB2 which is a production data warehouse. This script will be scheduled by the data engineers to sync up the data between the staging and production data warehouse.

Objectives

In this assignment you will write a python program that will:

- Connect to IBM DB2 data warehouse and identify the last row on it.
- Connect to MySQL staging data warehouse and find all rows later than the last row on the datawarehouse.
- Insert the new data in the MySQL staging data warehouse into the IBM DB2 production data warehouse.

Tools / Software

- MySQL Server
- IBM DB2 database running on IBM Cloud

Note - Screenshots

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

Prepare the lab environment

Before you start the assignment:

Step 1: Start MySQL server

Step 2: Create a database named `sales`

Step 3: Download the file below

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

Step 4: Import the data in the file `sales.sql` into the `sales` database.

Step 5: Verify that you can access your cloud instance of IBM DB2 server.

Step 6: Download the mysqlconnect.py python programs from link below.

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

Step 7: mysqlconnect.py has the sample code to help you understand how to connect to MySQL using Python.

Step 8: Modify mysqlconnect.py suitably and make sure you are able to connect to the MySQL server instance on the Theia environment.

Step 9: Download the db2connect.py python programs from link below.

https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB0321EN-SkillsNetwork/ETL/db2connect.py

Step 10: db2connect.py has the sample code to help you understand how to connect to the cloud instance of IBM DB2 using Python.

Step 11: Modify db2connect.py suitably and make sure you are able to connect to your cloud instance of IBM DB2 from the Theia environment.

Step 12: Download the file below

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

Step 13: Load sales.csv into a table named `sales_data` on your cloud instance of IBM DB2 database.

Optional step:

By default, the price and timestamp columns have nullable records. If you do not wish them to be null, please run the below script in DB2:

```
ALTER TABLE <table name>
ALTER COLUMN timestamp SET DATA TYPE TIMESTAMP;

ALTER TABLE <table name>
ALTER COLUMN timestamp SET NOT NULL;

ALTER TABLE <table name>
ALTER COLUMN timestamp SET DEFAULT CURRENT_TIMESTAMP;

ALTER TABLE <table name>
ALTER COLUMN price SET NOT NULL;

ALTER TABLE <table name>
ALTER COLUMN price SET DEFAULT 0;
```

Step 14: https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB0321EN-SkillsNetwork/ETL/automation.py

You will be using automation.py as a scafolding program to execute the tasks in this assignment

Note: After running the automation.py file, if you get any reason code 7 error in the terminal, run the below command in Db2 and then rerun the file.

```
call sysproc.admin_cmd('reorg table <schema name>.<table name>')
```

Exercise 1 - Automate loading of incremental data into the data warehouse

One of the routine tasks that is carried out around a data warehouse is the extraction of daily new data from the operational database and loading it into the data warehouse. In this exercise you will automate the extraction of incremental data, and loading it into the data warehouse.

Task 1 - Implement the function get_last_rowid()

In the program automation.py implement the function get_last_rowid()

This function must connect to the DB2 data warehouse and return the last rowid.

Take a screenshot of the python code clearly showing the implementation of the function get_last_rowid().

Name the screenshot `get_last_rowid.jpg`. (Images can be saved with either the .jpg or .png extension.)

Task 2 - Implement the function get_latest_records()

In the program automation.py implement the function get_latest_records()

This function must connect to the MySQL database and return all records later than the given last_rowid.

Take a screenshot of the python code clearly showing the implementation of the function get_latest_records().

Name the screenshot `get_latest_records.jpg`. (Images can be saved with either the .jpg or .png extension.)

Task 3 - Implement the function insert_records()

In the program automation.py implement the function insert_records()

This function must connect to the DB2 data warehouse and insert all the given records.

Take a screenshot of the python code clearly showing the implementation of the function insert_records().

Name the screenshot `insert_records.jpg`. (Images can be saved with either the .jpg or .png extension.)

Task 4 - Test the data synchronization

Run the program automation.py and test if the synchronization is happening as expected.

Take a screenshot of the program output .

Name the screenshot `synchronization.jpg`. (Images can be saved with either the .jpg or .png extension.)

End of the assignment.

Authors

Ramesh Sannareddy

Other Contributors

Rav Ahuja

Change Log

Date (YYYY-MM-DD)	Version	Changed By	Change Description
2021-13-12	0.1	Ramesh Sannareddy	Created initial version

Date (YYYY-MM-DD)	Version	Changed By	Change Description
2022-09-29	0.2	Appalabhaktula Hema	Updated code and instructions