

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 or PostgreSQL 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 project you will write a python program that will:

- Connect to IBM DB2 or PostgreSQL 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 or PostgreSQL production data warehouse.

Software Required

- MySQL Server
- IBM DB2 or PostgreSQL

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.

Note: Before executing `mysqlconnect.py` note that you install the connector using the command `python3 -m pip install mysql-connector-python==8.0.31`

In order to complete the tasks below, you have the option to complete them on either a DB2 database (Option A) or on PostgreSQL (Option B).

If you choose **Option A**, please follow tasks 9-12. If you choose **Option B**, please follow tasks 13-16.

Option A: If you choose DB2 as the data warehouse:

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

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

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

Note: Before executing `db2connect.py` note that you install the connector using the command `pip install --force-reinstall ibm_db==3.1.0 ibm_db_sa==0.3.3`

Step 10: Modify `db2connect.py` suitably and make sure you are able to connect to your cloud instance of IBM DB2 from the Theia environment.

Step 11: Download the file below

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

Step 12: Load `sales.csv` into a table named `sales_data` on your cloud instance of IBM DB2 database.

OR

Option B: If you choose PostgreSQL as the data warehouse:

Step 13: Download the `postgresqlconnect.py` python program from the link below.

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

`postgresqlconnect.py` has the sample code to help you understand how to connect to the PostgreSQL data warehouse using Python.

Note: Before executing `postgresqlconnect.py` note that you install the connector using the command `python3 -m pip install psycopg2`

Step 14: Modify `postgresqlconnect.py` suitably and make sure you are able to connect to PostgreSQL from the Theia environment.

Step 15: Download the file below

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

Step 16: Create a table called `sales_data` using the columns `rowid`, `product_id`, `customer_id`, `price`, `quantity` `timestamp`. Load `sales.csv` into the table `sales_data` on your PostgreSQL database.

Step 17: Download the `automation.py` from the following URL

<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 scaffolding program to execute the tasks in this assignment

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.

In order to complete Tasks 1 and 3 below, you have an option to complete the tasks on a DB2 database (Option A), or on PostgreSQL (Option B).

Task 1 - Implement the function `get_last_rowid()`

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

Option A: If you choose DB2 as the data warehouse:

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

Option B: If you choose PostgreSQL as the data warehouse:

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

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.

Task 3 - Implement the function insert_records()

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

Option A: If you choose DB2 as the data warehouse:

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

Option B: If you choose PostgreSQL as the data warehouse:

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

Task 4 - Test the data synchronization

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

End of the assignment.

Date (YYYY-MM-DD)	Version	Changed By	Change Description
2023-06-29	0.7	Lakshmi Holla	Updated PostgreSql
2023-05-11	0.6	Eric Hao & Vladislav Boyko	Updated Page Frames
2023-05-10	0.5	Eric Hao & Vladislav Boyko	Updated Page Frames
2023-05-10	0.4	Eric Hao & Vladislav Boyko	Updated Page Frames
2021-13-12	0.1	Ramesh Sannareddy	Created initial version
2022-09-29	0.2	Appalabhaktula Hema	Updated code and instructions
2023-05-04	0.3	Benny Li	Republished

© IBM Corporation 2023. All rights reserved.