DALHOUSIE UNIVERSITY

Data Management, Warehousing and Analytics

Jaskaran Singh

MACS

B00948857

js356337@dal.ca

Lab Assignment 4

Gitlab Repository link:

https://git.cs.dal.ca/singh16/csci5408_s23_b00948857_jaskaran_singh.git

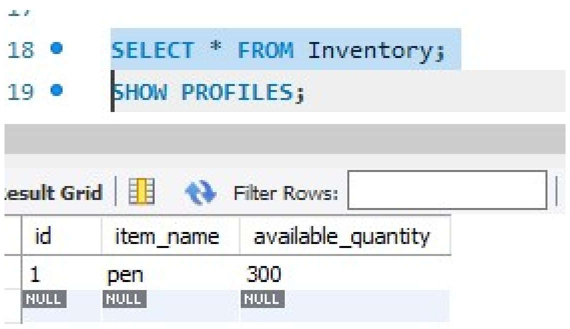
- A distributed MySQL DBMS system has been setup with two databases.

Local Database : **Order_Management**

- First one is on local machine with name ‘**Order_Management’**. It contains 2 tables named User and Order_info.
- User table has id as primary key.
- Order_info has order_id as primary key and user_id as foreign key referencing user table.
- SQL queries are present in ‘local_scripts’ file.

Remote Database : **Inventory_Management**

- Second one is on GCP with name ‘**Inventory_Management’**. It contains inventory table.
- Inventory table has id as primary key.
- Also, initially, 300 quantities of pens are inserted in inventory table.
- SQL queries are present in ‘remote_scripts’ file.

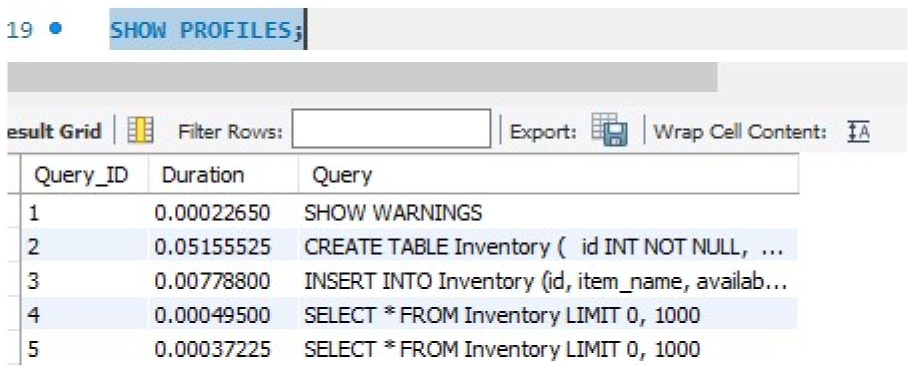


```
SELECT * FROM Inventory;
SHOW PROFILES;
```

id	item_name	available_quantity
1	pen	300
NULL	NULL	NULL

Figure 1: Record in Inventory table initially

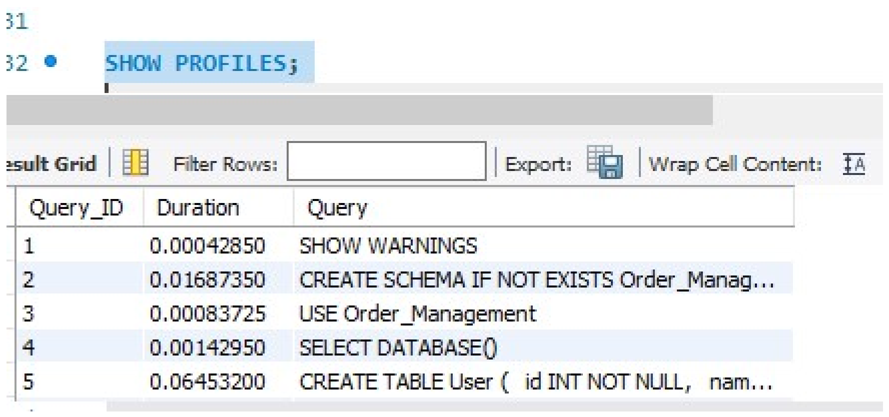
- Also, profiling has been set to 1 to measure performance on both the databases.



```
SHOW PROFILES;
```

Query_ID	Duration	Query
1	0.00022650	SHOW WARNINGS
2	0.05155525	CREATE TABLE Inventory (id INT NOT NULL, ...
3	0.00778800	INSERT INTO Inventory (id, item_name, availab...
4	0.00049500	SELECT * FROM Inventory LIMIT 0, 1000
5	0.00037225	SELECT * FROM Inventory LIMIT 0, 1000

Figure 2: Latency from remote database(GCP)



```
SHOW PROFILES;
```

Query_ID	Duration	Query
1	0.00042850	SHOW WARNINGS
2	0.01687350	CREATE SCHEMA IF NOT EXISTS Order_Manag...
3	0.00083725	USE Order_Management
4	0.00142950	SELECT DATABASE()
5	0.06453200	CREATE TABLE User (id INT NOT NULL, nam...

Figure 3 : Latency from local database

JAVA PROGRAM

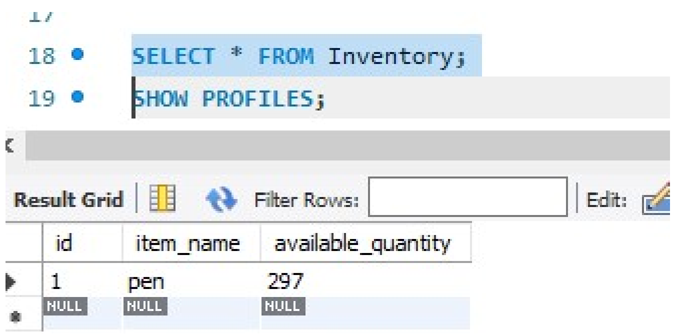
- Java program is present in 'ConnectionApplication' folder.
- A maven project has been created in Java to connect with both the databases(remote and local).
- 'mysql-connector-java' dependency has been used to connect Java program with MySQL database. It provides MySQL JDBC driver to connect with database.
- In this java program, firstly records are fetched from inventory table from GCP.
- DriverManager class from 'java.sql' package has been used to create a Connection instance with both local and remote databases.
- Firstly, records from inventory table has been fetched in the program.
- Then, an order has been created in the Order_info table for 3 quantities of pen and accordingly, updated quantites are then stored in remote database.

Working of JAVA PROGRAM:

1. **Establishing database connections:**
    - The program establishes connections to both the local and remote databases using the provided connection details (URL, username, and password).
    - The DriverManager.getConnection() method is used to establish the connections.
2. **Fetching item details from the remote database:**

- A SELECT query is executed on the Inventory table in the remote database to fetch item details (item name and available quantity).
- The query result is retrieved as a ResultSet object.

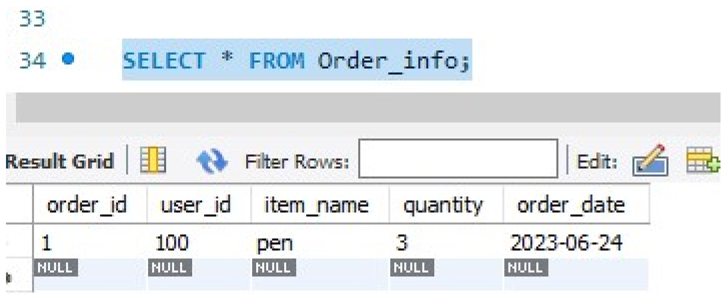
3. **Processing fetched item details:**
    - Using a loop, the program iterates through the rows in the ResultSet to access each item's details (item name and available quantity).
    - Within the loop, you can perform any necessary operations with the fetched data (e.g., displaying the details or performing calculations).
4. **Creating an order in the local database:**
    - For each item, an INSERT query is prepared to insert the order information (order ID, user ID, item name, quantity, and order date) into the Order_info table in the local database.
    - The prepared statement is executed using the executeUpdate() method to perform the insertion.
5. **Updating the quantity in the remote database:**
    - An UPDATE query is prepared to update the available_quantity column in the Inventory table for the corresponding item.
    - The prepared statement is executed using the executeUpdate() method to update the quantity in the remote database.
6. **Closing resources and connections:**
    - After the processing is complete, the program closes database connections to release resources.



```
SELECT * FROM Inventory;
SHOW PROFILES;
```

id	item_name	available_quantity
1	pen	297
NULL	NULL	NULL

Figure 4: Record in Inventory table (GCP) after order of 3 inventory created



```
SELECT * FROM Order_info;
```

order_id	user_id	item_name	quantity	order_date
1	100	pen	3	2023-06-24
NULL	NULL	NULL	NULL	NULL

Figure 5: Record in Order_info table after order of 3 inventory created

References:

- Java Database Connectivity with MySQL, Java T Point,

https://www.javatpoint.com/example-to-connect-to-the-mysql-database