CSCI 5408

DATA MANAGEMENT AND WAREHOUSING

LAB ASSIGNMENT - 4

Banner ID: B00952865

GitLab Assignment Link: https://git.cs.dal.ca/apurohit/CSCI5408_F23_B00952865_AdityaMaheshbhai_Purohit/-/tree/main/Lab4

Table of Contents

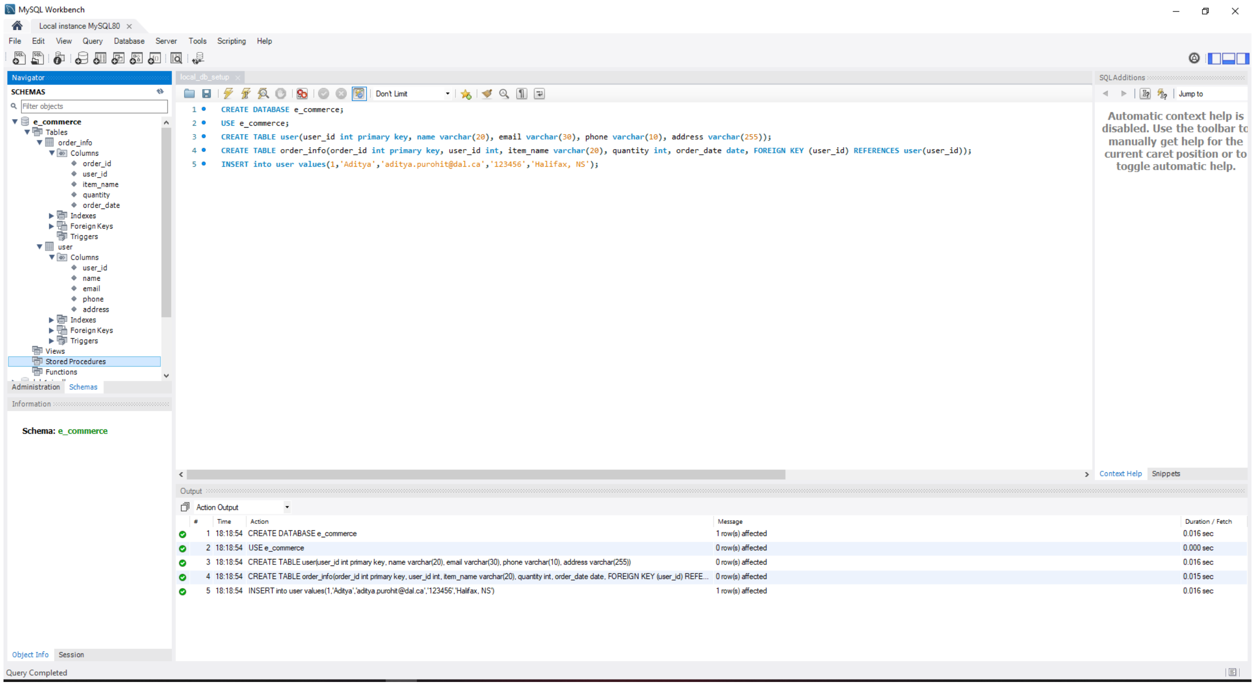
Local Instance Set-up..3

Remote Instance Set-up 4

Java Program with Profiling (Query Execution times):..6

References:...9

Local Instance Set-up



```
CREATE DATABASE e_commerce;
USE e_commerce;
CREATE TABLE user(user_id int primary key, name varchar(20), email varchar(30), phone varchar(10), address varchar(255));
CREATE TABLE order_info(order_id int primary key, user_id int, item_name varchar(20), quantity int, order_date date, FOREIGN KEY (user_id) REFERENCES user(user_id));
INSERT into user values(1,'Aditya','aditya.purohit@dal.ca','123456','Halifax, NS');
```

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

#	Time	Action	Message	Duration / Fetch
1	18:18:54	CREATE DATABASE e_commerce	1 row(s) affected	0.016 sec
2	18:18:54	USE e_commerce	0 row(s) affected	0.000 sec
3	18:18:54	CREATE TABLE user(user_id int primary key, name varchar(20), email varchar(30), phone varchar(10), address varchar(255))	0 row(s) affected	0.016 sec
4	18:18:54	CREATE TABLE order_info(order_id int primary key, user_id int, item_name varchar(20), quantity int, order_date date, FOREIGN KEY (user_id) REFE...	0 row(s) affected	0.015 sec
5	18:18:54	INSERT into user values(1,'Aditya','aditya.purohit@dal.ca','123456','Halifax, NS')	1 row(s) affected	0.016 sec

Figure 1: Create E-commerce DB on local instance using set of queries [1].

DDL Queries:

CREATE DATABASE e_commerce;

USE e_commerce;

CREATE TABLE user(user_id int primary key, name varchar(20), email varchar(30), phone varchar(10), address varchar(255));

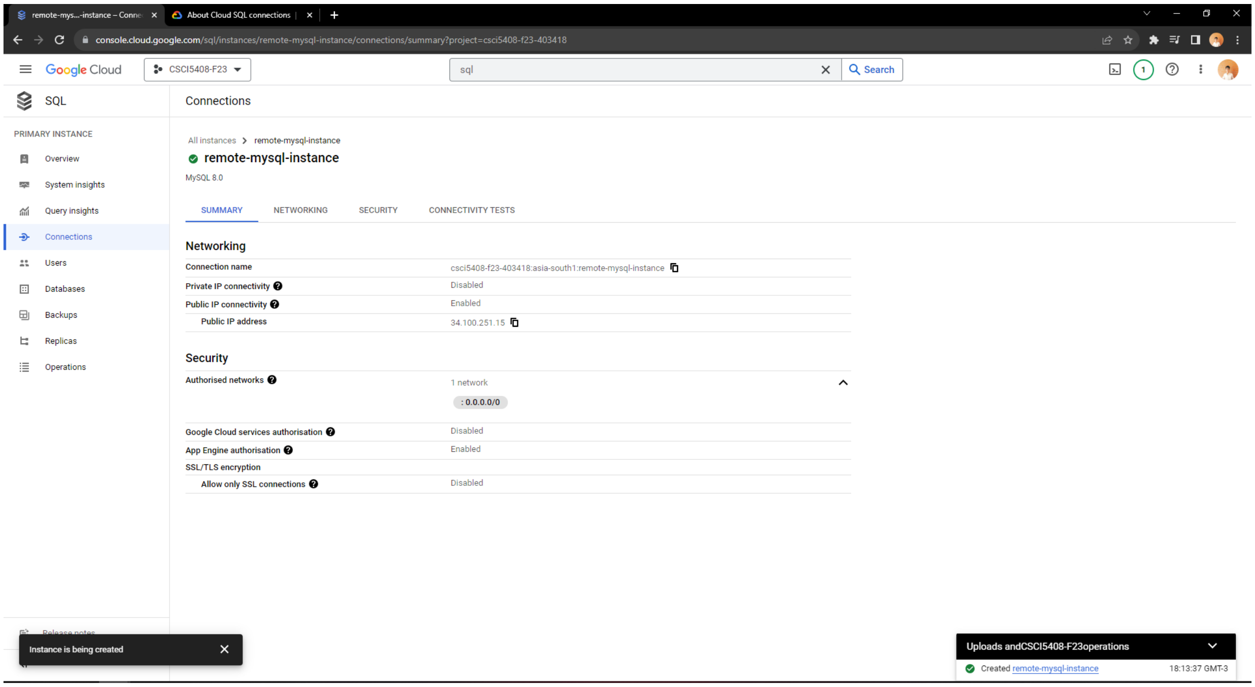
CREATE TABLE order_info(order_id int primary key, user_id int, item_name varchar(20), quantity int, order_date date, FOREIGN KEY (user_id) REFERENCES user(user_id));

INSERT into user values(1,'Aditya','aditya.purohit@dal.ca','123456','Halifax, NS');

Explanation:

I have created 2 Tables: user and order_info and inserted a user, so that its id can be used while order creation in java program later.

Remote Instance Set-up



remote-mysql-instance

MySQL 8.0

Networking

Connection name: csci5408-f23-403418:asia-south1:remote-mysql-instance

Private IP connectivity: Disabled

Public IP connectivity: Enabled

Public IP address: 34.100.251.15

Security

Authorised networks: 1 network

: 0.0.0.0/0

Google Cloud services authorisation: Disabled

App Engine authorisation: Enabled

SSL/TLS encryption

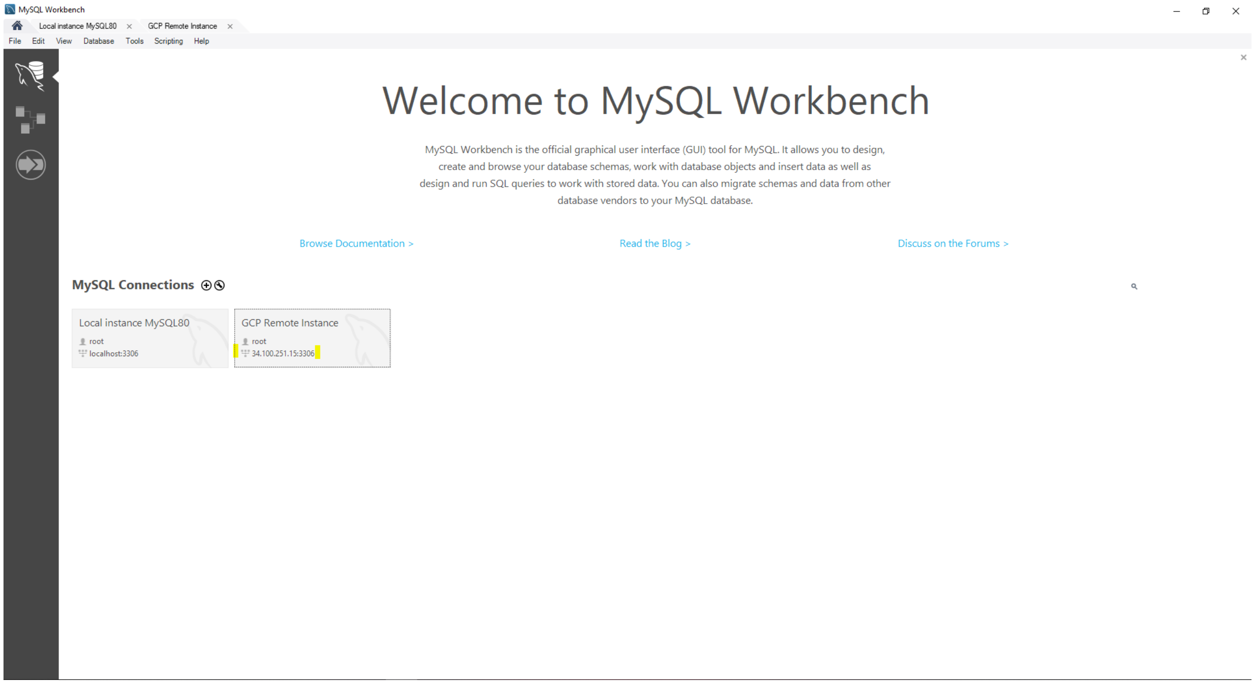
Allow only SSL connections: Disabled

Instance is being created

Uploads andCSCI5408-F23operations

Created remote-mysql-instance 18:13:37 GMT-3

Figure 2: Instance created on GCP [2]



Welcome to MySQL Workbench

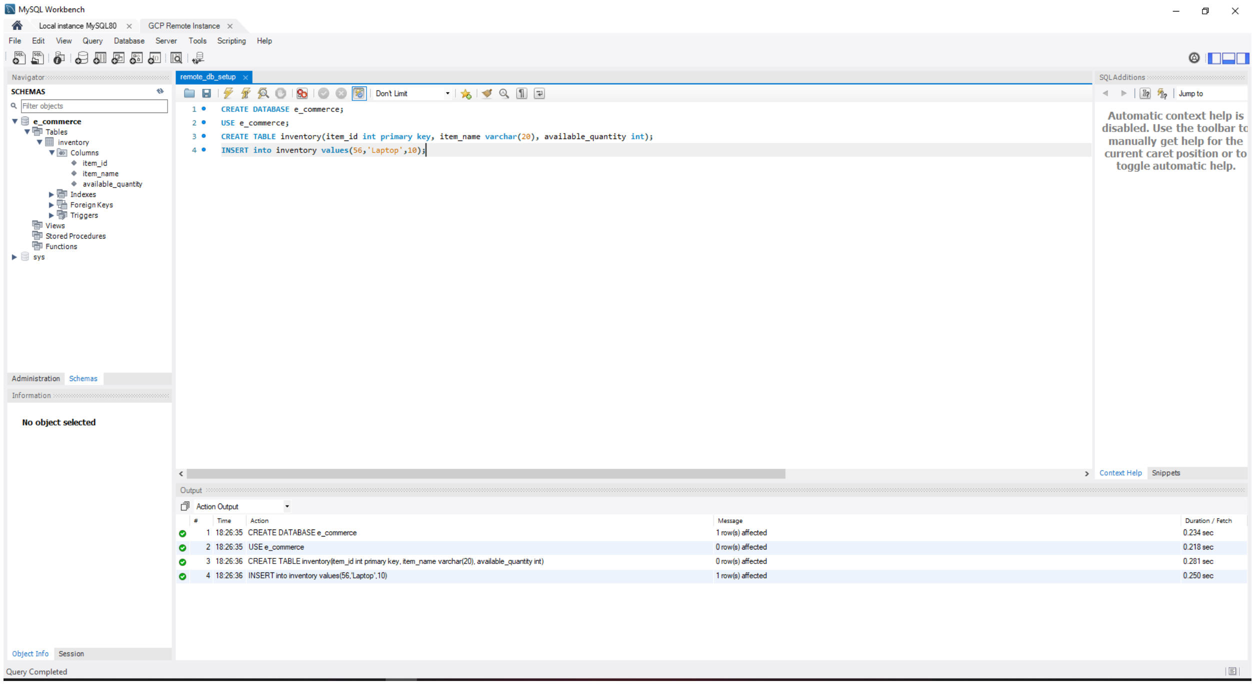
MySQL Workbench is the official graphical user interface (GUI) tool for MySQL. It allows you to design, create and browse your database schemas, work with database objects and insert data as well as design and run SQL queries to work with stored data. You can also migrate schemas and data from other database vendors to your MySQL database.

MySQL Connections

Local instance MySQL80
root
localhost:3306

GCP Remote Instance
root
34.100.251.15:3306

Figure 3: Connected MySQL Workbench with GCP MySQL instance [1]



```
CREATE DATABASE e_commerce;
USE e_commerce;
CREATE TABLE inventory(item_id int primary key, item_name varchar(20), available_quantity int);
INSERT into inventory values(56,'Laptop',10);
```

```
#  Time      Action                                                                               Message             Duration / Fetch
1  18:26:35  CREATE DATABASE e_commerce                                                           1 row(s) affected   0.234 sec
2  18:26:35  USE e_commerce                                                                       0 row(s) affected   0.218 sec
3  18:26:36  CREATE TABLE inventory(item_id int primary key, item_name varchar(20), available_quantity int)  0 row(s) affected   0.281 sec
4  18:26:36  INSERT into inventory values(56,'Laptop',10)                                         1 row(s) affected   0.250 sec
```

Figure 4: Set-up remote instance with inventory table [1]

DDL Queries:

CREATE DATABASE e_commerce;

USE e_commerce;

CREATE TABLE inventory(item_id int primary key, item_name varchar(20), available_quantity int);

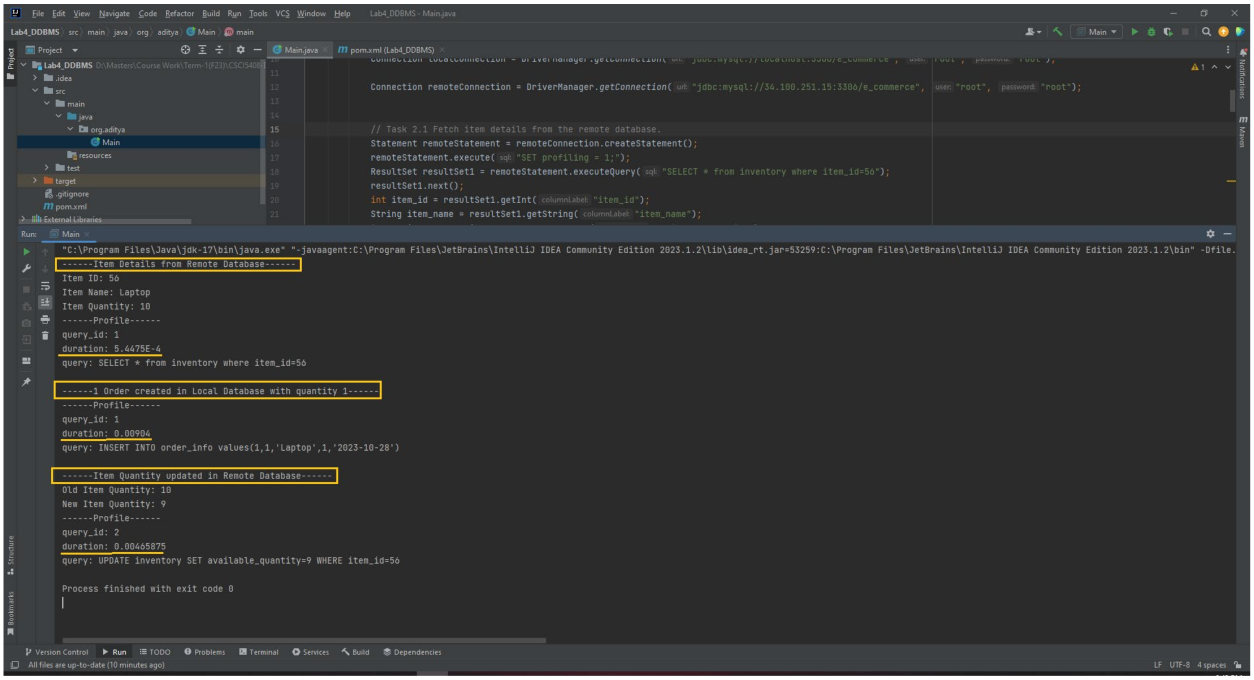
INSERT into inventory values(56,'Laptop',10);

Explanation:

I have created a remote MySQL instance and then using the credentials and public IP, I have connected MySQL Workbench with the instance. After that, I ran a set of queries to create an e_commerce database, inventory table and added Laptop item in the table, so that it can be later used my java program.

Java Program with Profiling (Query Execution times):

The entire java code can be found here:
https://git.cs.dal.ca/apurohit/CSCI5408_F23_B00952865_AdityaMaheshbhai_Purohit/-/tree/main/Lab4



```java
        Connection remoteConnection = DriverManager.getConnection("jdbc:mysql://34.100.251.15:3306/e_commerce", "root", "root");


        // Task 2.1 Fetch item details from the remote database.
        Statement remoteStatement = remoteConnection.createStatement();
        remoteStatement.execute("SET profiling = 1;");
        ResultSet resultSet1 = remoteStatement.executeQuery("SELECT * from inventory where item_id=56");
        resultSet1.next();
        int item_id = resultSet1.getInt("item_id");
        String item_name = resultSet1.getString("item_name");
```

```
"C:\Program Files\Java\jdk-17\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2023.1.2\lib\idea_rt.jar=53259:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2023.1.2\bin" -Dfile.
------Item Details from Remote Database------
Item ID: 56
Item Name: Laptop
Item Quantity: 10
------Profile------
query_id: 1
duration: 5.4475E-4
query: SELECT * from inventory where item_id=56

------1 Order created in Local Database with quantity 1------
------Profile------
query_id: 1
duration: 0.00904
query: INSERT INTO order_info values(1,1,'Laptop',1,'2023-10-28')

------Item Quantity updated in Remote Database------
Old Item Quantity: 10
New Item Quantity: 9
------Profile------
query_id: 2
duration: 0.00465875
query: UPDATE inventory SET available_quantity=9 WHERE item_id=56

Process finished with exit code 0
```

Figure 5: Java Program execution with 3 tasks and query execution time for each [3] [4] [5] [6].

Explanation:

I created 2 connection objects in java, 1 for local and another for remote. After that I ran a select query to get the item details from inventory, then ran an insert query in the local database's order_info table only if the order quantity is less or equals the available stock. Lastly, ran the update query on the remote database inventory table to update the quantity.

Query Execution Timings:

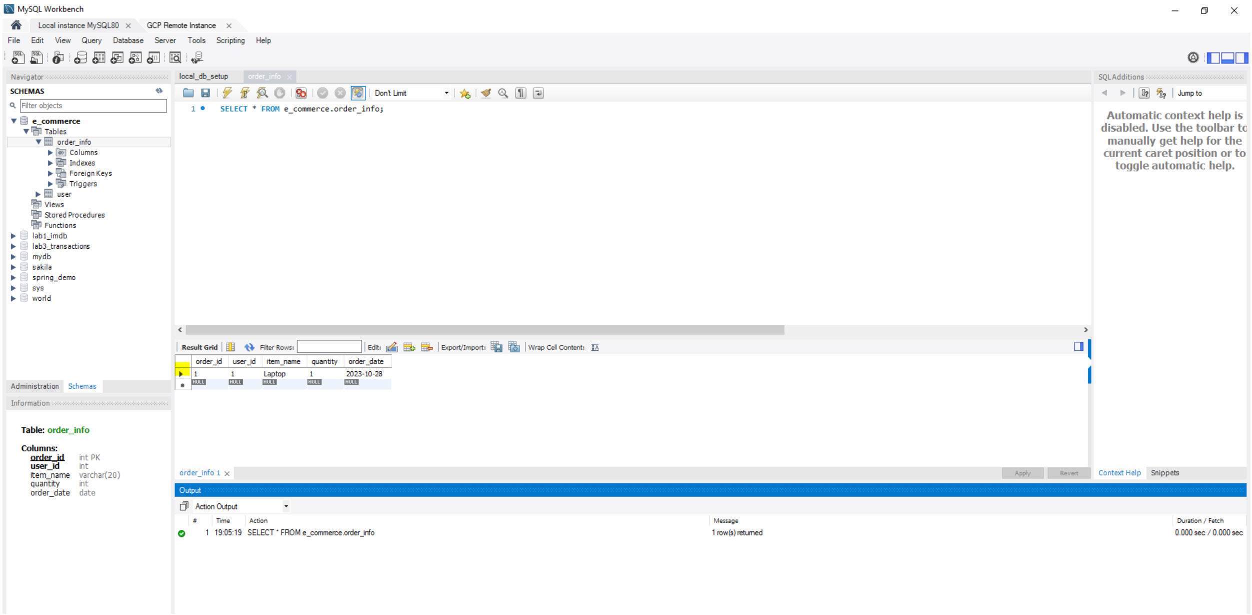
Remote Select Query: approx. 0.0005s (lowest)

Local Insert Query: approx. 0.009s (highest)

Remote Update Query: approx. 0.004s (2nd highest)

This query execution time differences are mainly due to the **complexity of each query**. As in Insert query we are writing a whole row, it takes highest time. Then, the update query also makes a write (over-write) operation but only on 1 value (i.e. available_quantity of only 1 item). Select is the most fast query here as it has to only perform read operation and there are no column filters or where clauses. **Note:** Query Execution time doesn't depend on the location of database (be it remote or local). Data Retrieval time is what differs for a same query, due to location and network speed.

Status of Tables after java program execution:



```
SELECT * FROM e_commerce.order_info;
```

order_id	user_id	item_name	quantity	order_date
1	1	Laptop	1	2023-10-28
NULL	NULL	NULL	NULL	NULL

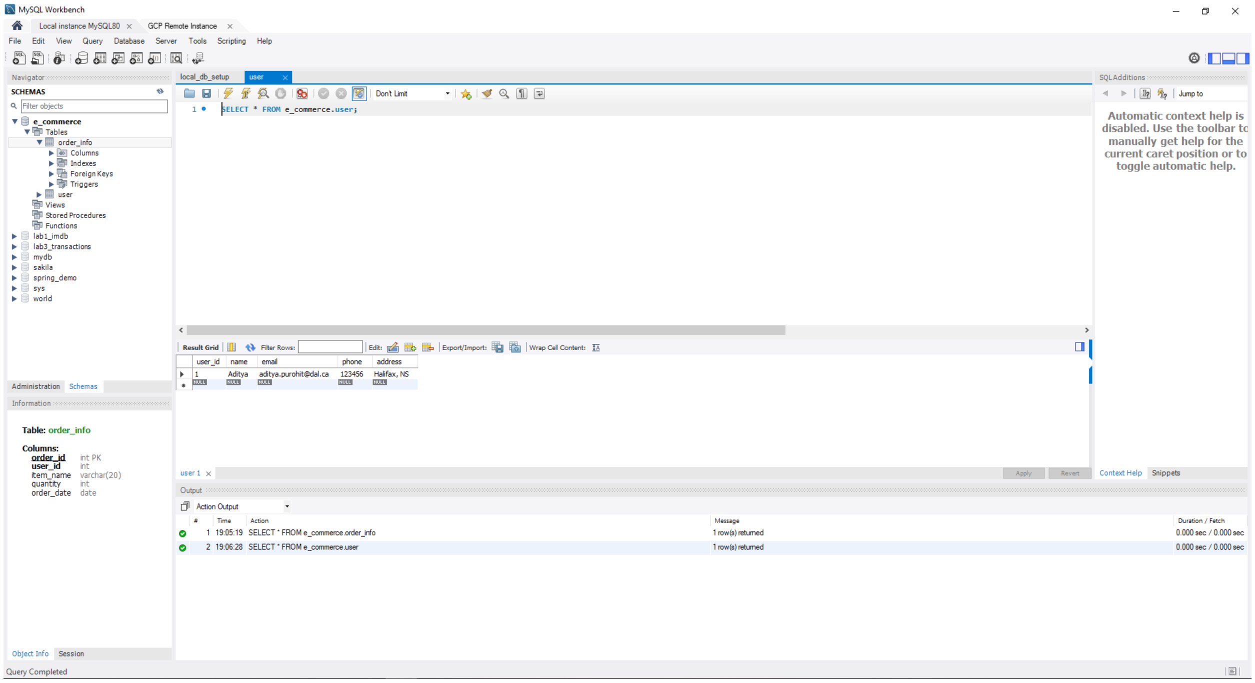
```
#  Time      Action                                   Message              Duration / Fetch
1  19:05:19  SELECT * FROM e_commerce.order_info      1 row(s) returned    0.000 sec / 0.000 sec
```

Table: order_info

Columns:

order_id	int PK
user_id	int
item_name	varchar(20)
quantity	int
order_date	date

Figure 6: order_info table in local instance [1].

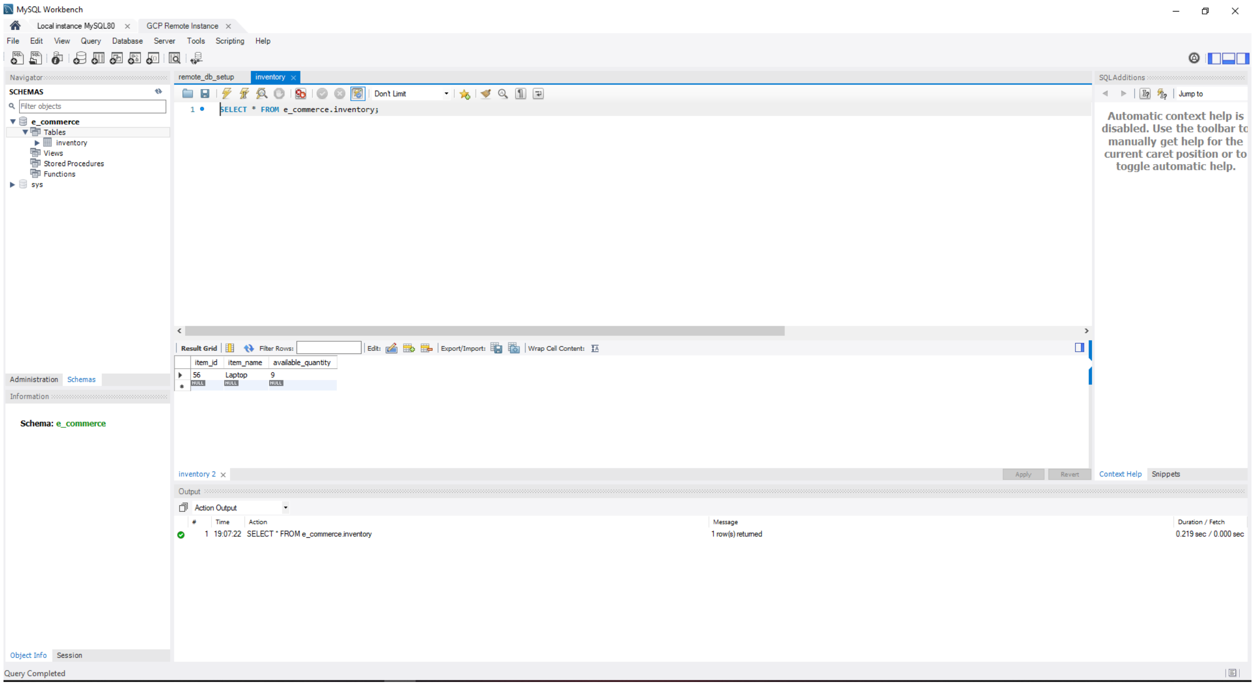


```
SELECT * FROM e_commerce.user;
```

user_id	name	email	phone	address
1	Aditya	aditya.purohit@dal.ca	123456	Halifax, NS
NULL	NULL	NULL	NULL	NULL

```
#  Time      Action                                   Message              Duration / Fetch
1  19:05:19  SELECT * FROM e_commerce.order_info      1 row(s) returned    0.000 sec / 0.000 sec
2  19:06:28  SELECT * FROM e_commerce.user            1 row(s) returned    0.000 sec / 0.000 sec
```

Figure 7: user table in local instance [1].



Schema: e_commerce

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

```sql
SELECT * FROM e_commerce.inventory;
```

item_id	item_name	available_quantity
56	Laptop	9
NULL	NULL	NULL

```
#  Time      Action                                Message           Duration / Fetch
1  19:07:22  SELECT * FROM e_commerce.inventory    1 row(s) returned 0.219 sec / 0.000 sec
```

Figure 8: inventory table in remote instance [1].

Explanation:

It can be seen that the order_info table has a new entry and the inventory table has been updated, as the available_quantity is reduced by 1. The user table remains as it is because the java program doesn't touch that table in any way.

References:

[1] "MySQL Workbench," MySQL, https://www.mysql.com/products/workbench/ (accessed Oct. 28, 2023).

[2] "Google Cloud SQL Instances", Google Cloud Platform, https://console.cloud.google.com/sql/instances/ (accessed Oct. 28, 2023).

[3] "Java | Oracle," Java.com, https://www.java.com/en/ (accessed Oct. 28, 2023).

[4] "IntelliJ IDEA – the leading Java and Kotlin Ide," JetBrains, https://www.jetbrains.com/idea/ (accessed Oct. 28, 2023).

[5] "MySQL Connector Java » 8.0.33 - maven repository," MVN Repository, https://mvnrepository.com/artifact/mysql/mysql-connector-java/8.0.33 (accessed Oct. 28, 2023).

[6] B. Porter, J. van Zyl, and O. Lamy, "Welcome to Apache Maven," Maven, https://maven.apache.org/ (accessed Oct. 28, 2023).