



**CSCI 5408 – Data Management, Warehousing, Analytics**

**Assignment 2**

**Work done by,**

**Name: Guturu Rama Mohan Vishnu**

**Banner ID: B00871849**

**Email: [rm286720@dal.ca](mailto:rm286720@dal.ca)**

## **DECLARATION**

**I, Guturu Rama Mohan Vishnu, declare that in assignment 2 of CSCI 5408 course, writing queries is not done programmatically or using any online or offline tools. However, the webpages or the domain mentioned in this document are visited manually, and some useful information is gathered for education purpose only. Information, such as email, personal contact numbers, or names of people are not extracted. The course instructor or the Faculty of Computer Science cannot be held responsible for any misuse of the extracted data.**

## **Problem #2:** Research and Development to simulate a distributed DBMS

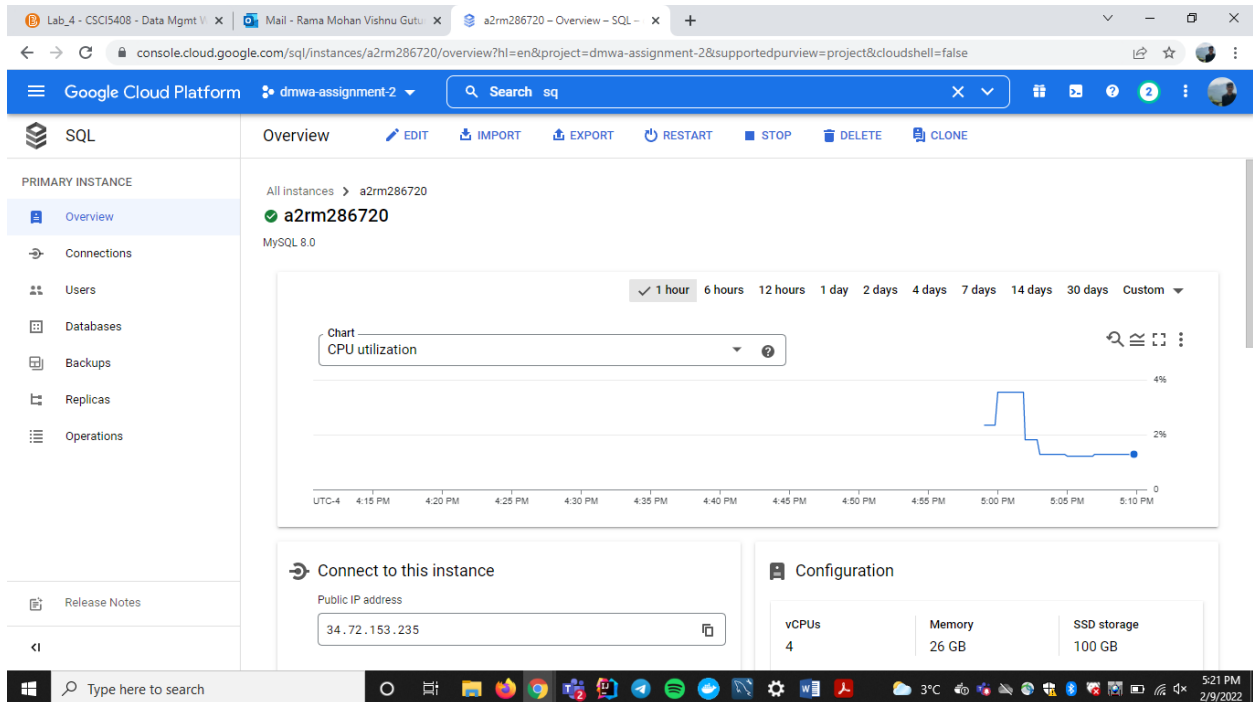
### **Solution:**

#### **Task 1 – Building database from given dataset**

The steps I have taken to clean up the .csv files are listed as below:

- First step I took to clean the datasets is to look for null values in it.
- The plan was to replace null values with '0' if it's a numerical column and replace with '' if it's a text/string column.
- I have gone through all the files but I couldn't find a single null value in all the 9 datasets. So, I have processed to next step.
- Next step is to look for empty cells in numeric columns to replace them with '0'. I have found few empty cells in few files in numerical columns. So as planned, I replaced them with '0'.
- The next step in data cleaning is to remove data duplicates. There are some files which have duplicate values. So, in order to not have any issues while creating primary and foreign keys, I deleted the rows which had the duplicates.
- Since we do not need to remove any columns from the csv files, none of the columns were deleted.

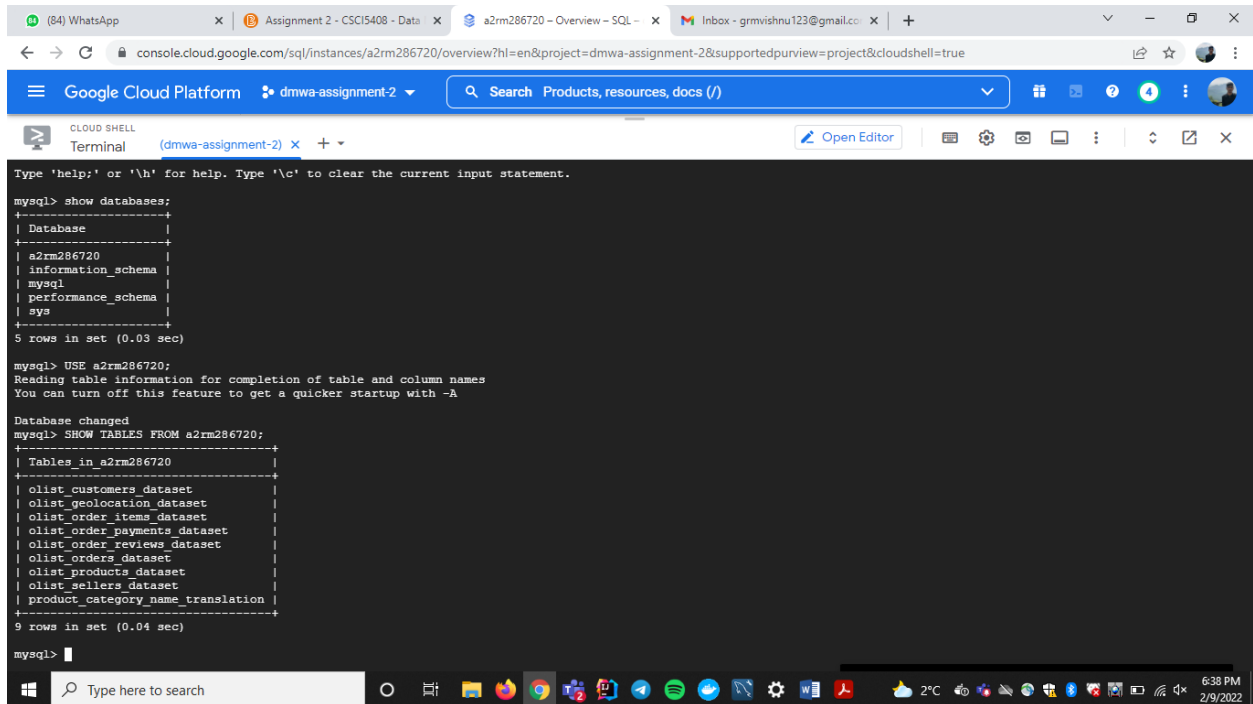
Proof that I have completed the required steps regarding creating a GCP MySQL Virtual Machine Instance and imported the tables in to the instance:



The screenshot shows the Google Cloud Platform console for a MySQL instance named **a2rm286720**. The instance is running MySQL 8.0. The left sidebar shows the navigation menu with options like Overview, Connections, Users, Databases, Backups, Replicas, and Operations. The main content area displays the 'Databases' section, which includes a 'CREATE DATABASE' button and a table listing the databases present in the instance.

Name	Collation	Character set	Type
a2rm286720	utf8_general_ci	utf8	User
information_schema	utf8_general_ci	utf8	System
mysql	utf8_general_ci	utf8	System
performance_schema	utf8mb4_0900_ai_ci	utf8mb4	System
sys	utf8mb4_0900_ai_ci	utf8mb4	System

Screenshot of running the MySQL command “SHOW TABLES FROM a2rm286720;” in GCP Virtual Machine instance:



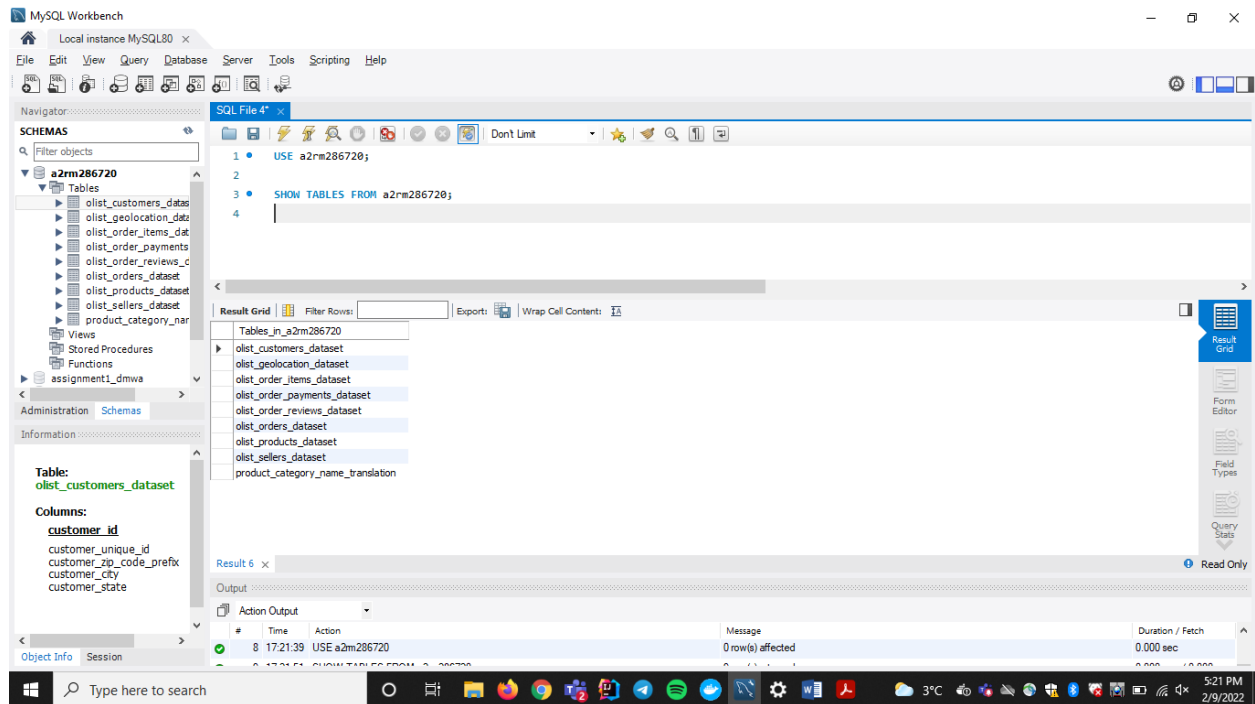
```
mysql> show databases;
+-----+
| Database |
+-----+
| a2rm286720 |
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.03 sec)

mysql> USE a2rm286720;
Reading table information for completion of table and column names
You can turn off this feature with -A

Database changed
mysql> SHOW TABLES FROM a2rm286720;
+-----+
| Tables_in_a2rm286720 |
+-----+
| olist_customers_dataset |
| olist_geolocation_dataset |
| olist_order_items_dataset |
| olist_order_payments_dataset |
| olist_order_reviews_dataset |
| olist_orders_dataset |
| olist_products_dataset |
| olist_sellers_dataset |
| product_category_name_translation |
+-----+
9 rows in set (0.04 sec)

mysql>
```

Screenshot of running the MySQL command “SHOW TABLES FROM a2rm286720;” in my local instance:



## **Task 2 – Perform Transaction in your local machine**

Queries for task 2 as required are:

```
USE a2rm286720;
```

```
SET autocommit = 0;
```

```
START TRANSACTION;
```

```
# select statement 1
```

```
SELECT order_id FROM olist_orders_dataset WHERE customer_id IN (SELECT customer_id  
FROM olist_customers_dataset WHERE customer_city = 'sao paulo' OR 'curitiba');
```

```
# insert statement 1
```

```
INSERT INTO olist_sellers_dataset (seller_id, seller_zip_code_prefix, seller_city, seller_state)  
VALUES ('shdvbjyshve87efe8ut7qwyd', '28', 'boleto', 'TS'),  
( 'ebfyew6gi7rtg3i4fr723gyevdki7w6t47', '4', 'voucher', 'AP');
```

```
# update statement 1
```

```
UPDATE olist_products_dataset JOIN product_category_name_translation SET  
olist_products_dataset.product_category_name =  
product_category_name_translation.product_category_name WHERE  
olist_products_dataset.product_category_name =  
product_category_name_translation.product_category_name_english;
```

```
# select statement 2
```

```
SELECT customer_id FROM olist_customers_dataset WHERE customer_zip_code_prefix IN  
(SELECT geolocation_zip_code_prefix FROM olist_geolocation_dataset WHERE  
geolocation_zip_code_prefix BETWEEN '1000' AND '1100');
```

```
# insert statement 2
```

```
INSERT INTO olist_products_dataset VALUES ('vefuyu4ryfswvyfuyfieuh783', 'market_place',  
'12', '1912', '9', '69', '73', '873', '25'), ('w78rwugfwurfg67fwyefv', 'cool_stuff', '16', '143', '3', '56',  
'34', '6772', '782');
```

```
# update statement 2
```

```
UPDATE olist_order_items_dataset AS i JOIN olist_orders_dataset AS o ON i.order_id =  
o.order_id JOIN olist_order_reviews_dataset AS r ON r.order_id = o.order_id SET i.price =  
i.price - (i.price/1) WHERE r.review_score = 1;
```

```
# select statement 3
```

```
SELECT product_id, product_category_name FROM olist_products_dataset WHERE  
product_id = (SELECT product_id FROM olist_order_items_dataset GROUP BY product_id  
ORDER BY sum(price) DESC LIMIT 1);
```

```
# delete statement 1
```

```
DELETE FROM olist_order_payments_dataset WHERE  
olist_order_payments_dataset.payment_type = 'not defined' AND  
olist_order_payments_dataset.payment_value = 0;
```

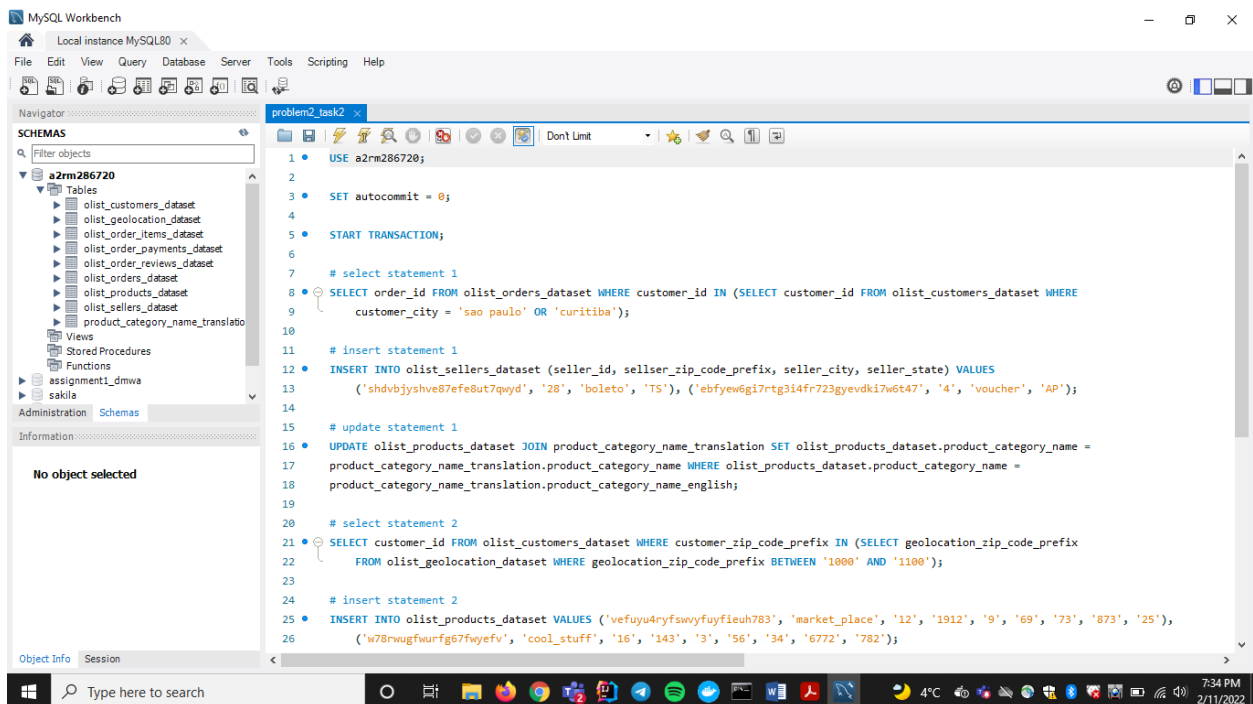
```
# delete statement 2
```

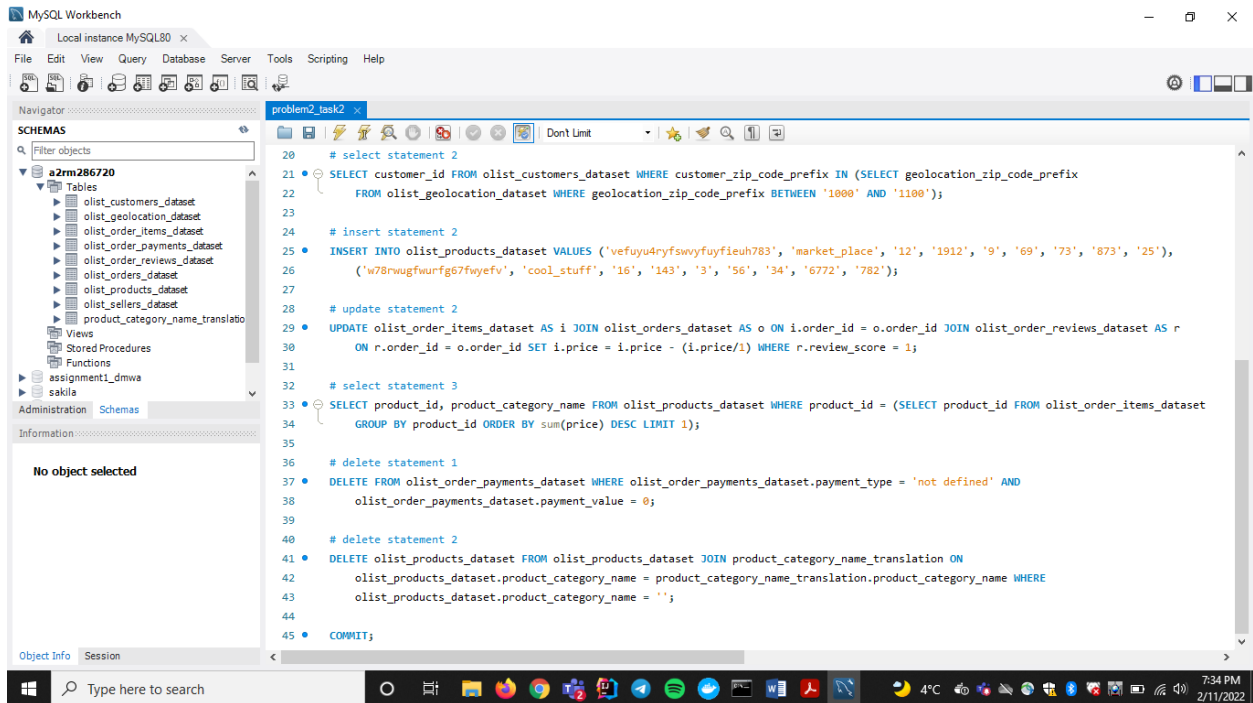


```
DELETE olist_products_dataset FROM olist_products_dataset JOIN
product_category_name_translation ON olist_products_dataset.product_category_name =
product_category_name_translation.product_category_name WHERE
olist_products_dataset.product_category_name = '';
```

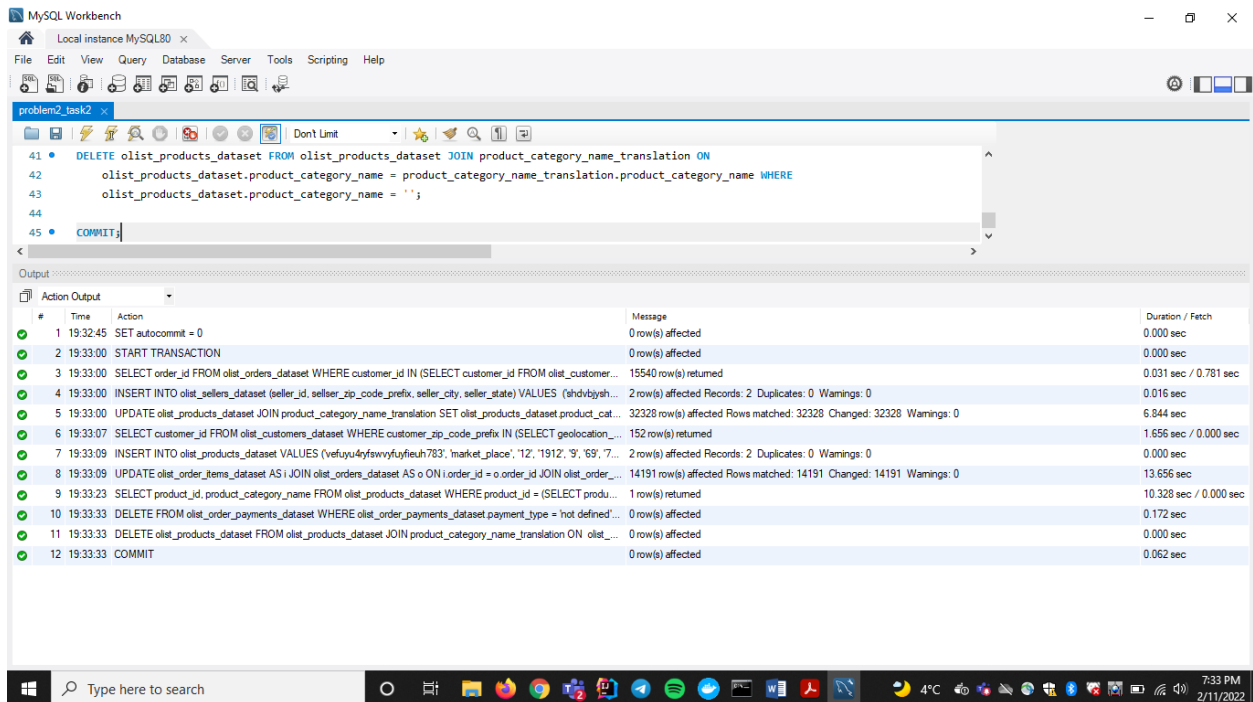
```
COMMIT;
```

I have also added the screenshots of it as proof in this document:





Proof that the transaction was successful without any issues and also to show the transaction time/queries time:



### **Task 3 – Perform Transaction in the Remote machine**

Queries for task 3 as required are:

```
USE a2rm286720;
```

```
SET autocommit = 0;
```

```
START TRANSACTION;
```

```
# insert statement 1
```

```
INSERT INTO olist_customers_dataset (customer_id, customer_unique_id,  
customer_zip_code_prefix, customer_city, customer_state) VALUES  
( 'yugrfo6i7fgqufyhgr36fre8', '3uygfi45f7evu6e74fr8u7we', '23453', 'halifax', 'NS'),  
( 'eyrgfu5quyfu3y4rfvge6w483hrey', '437gf34gr79giwf84', '785', 'mumbai', 'Karnataka'),  
( 'fyg7e5t3874witrwufhgrw4urhof83', 'i34gt7824tr93igrf723i', '1912', 'hyderabad' , 'telangana'),  
( 'ugfo3746f73twrg7fi2qgef', 'wu4tgr78wegtf7834gfu7wki7gf', '2119', 'hyderabad', 'telangana');
```

```
# insert statement 2
```

```
INSERT INTO product_category_name_translation (product_category_name,  
product_category_name_english) VALUES ('lait', 'milk'), ('yaourt', 'yogurt'), ('fromage',  
'cheese'), ('beurre', 'butter');
```

```
# select statement 1
```

```
SELECT * FROM olist_products_dataset INNER JOIN (SELECT product_id FROM  
olist_order_items_dataset ORDER BY (order_item_id*price) DESC LIMIT 1) AS A ON  
A.product_id = olist_products_dataset.product_id;
```

```
# select statement 2
```

```
SELECT seller_id FROM olist_sellers_dataset WHERE sellser_zip_code_prefix IN (SELECT
geolocation_zip_code_prefix FROM olist_geolocation_dataset WHERE
geolocation_zip_code_prefix BETWEEN '1000' AND '1100');
```

# select statement 3

```
SELECT olist_sellers_dataset.seller_id, olist_sellers_dataset.sellser_zip_code_prefix,
olist_sellers_dataset.seller_city, olist_sellers_dataset.seller_state FROM olist_sellers_dataset
JOIN olist_order_items_dataset ON olist_sellers_dataset.seller_id =
olist_order_items_dataset.seller_id JOIN olist_orders_dataset ON
olist_order_items_dataset.order_id = olist_orders_dataset.order_id JOIN (SELECT order_id
FROM olist_order_payments_dataset ORDER BY payment_installments DESC LIMIT 1) AS A
ON A.order_id = olist_orders_dataset.order_id;
```

# update statement 1

```
UPDATE olist_sellers_dataset JOIN olist_geolocation_dataset ON
olist_sellers_dataset.sellser_zip_code_prefix =
olist_geolocation_dataset.geolocation_zip_code_prefix SET
olist_sellers_dataset.sellser_zip_code_prefix = 143 AND olist_sellers_dataset.seller_city =
'markapur' AND olist_geolocation_dataset.geolocation_zip_code_prefix = 143 AND
olist_geolocation_dataset.geolocation_state = 'AP' WHERE olist_sellers_dataset.seller_state =
'PR';
```

# update statement 2

```
UPDATE olist_order_reviews_dataset JOIN olist_orders_dataset ON
olist_order_reviews_dataset.order_id = olist_orders_dataset.order_id JOIN
olist_customers_dataset ON olist_orders_dataset.customer_id =
olist_customers_dataset.customer_id SET
```

```
olist_order_reviews_dataset.review_comment_message = 'I am satisfied' WHERE  
olist_customers_dataset.customer_unique_id = 'e90a1b194724309bbaa6228c398d1748';
```

```
# delete statement 1
```

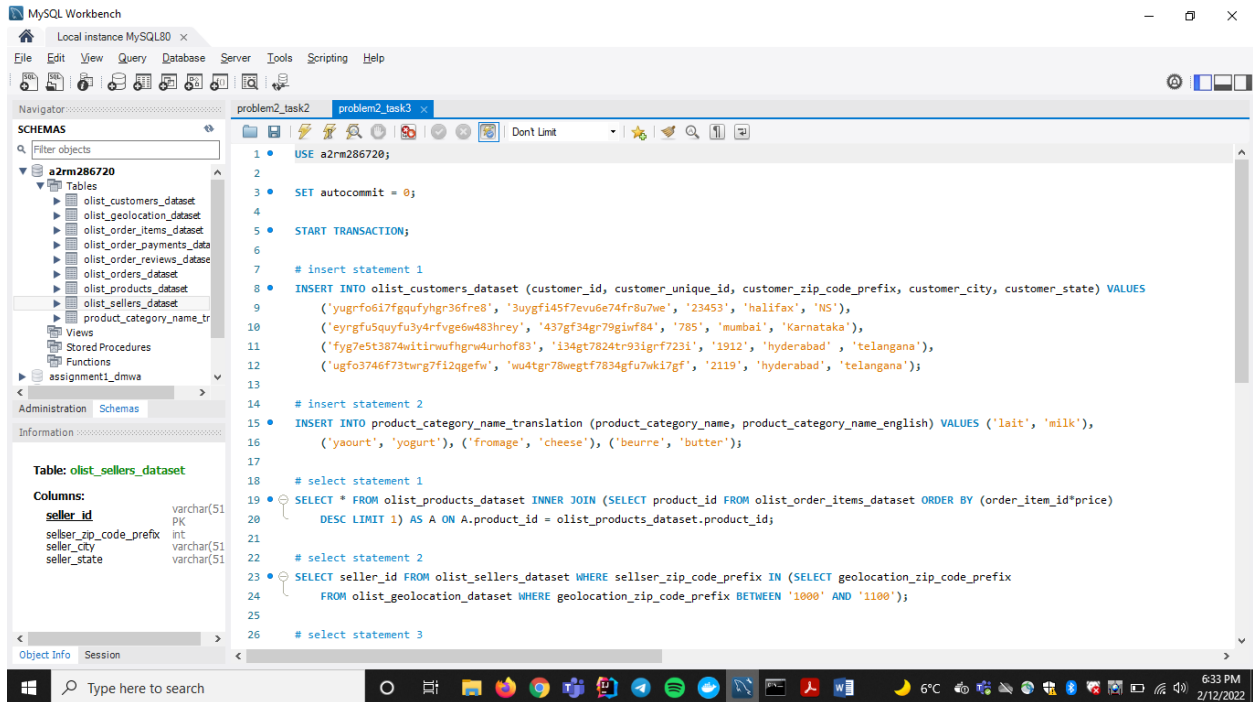
```
DELETE olist_products_dataset FROM olist_products_dataset INNER JOIN  
olist_order_items_dataset ON olist_products_dataset.product_id =  
olist_order_items_dataset.product_id INNER JOIN olist_sellers_dataset ON  
olist_order_items_dataset.seller_id = olist_sellers_dataset.seller_id WHERE  
olist_sellers_dataset.seller_id = '1025f0e2d44d7041d6cf58b6550e0bfa';
```

```
# delete statement 2
```

```
DELETE olist_customers_dataset FROM olist_customers_dataset INNER JOIN  
olist_orders_dataset ON olist_customers_dataset.customer_id = olist_orders_dataset.customer_id  
INNER JOIN olist_order_payments_dataset ON olist_orders_dataset.order_id =  
olist_order_payments_dataset.order_id WHERE  
olist_order_payments_dataset.payment_installments = 24;
```

```
COMMIT;
```

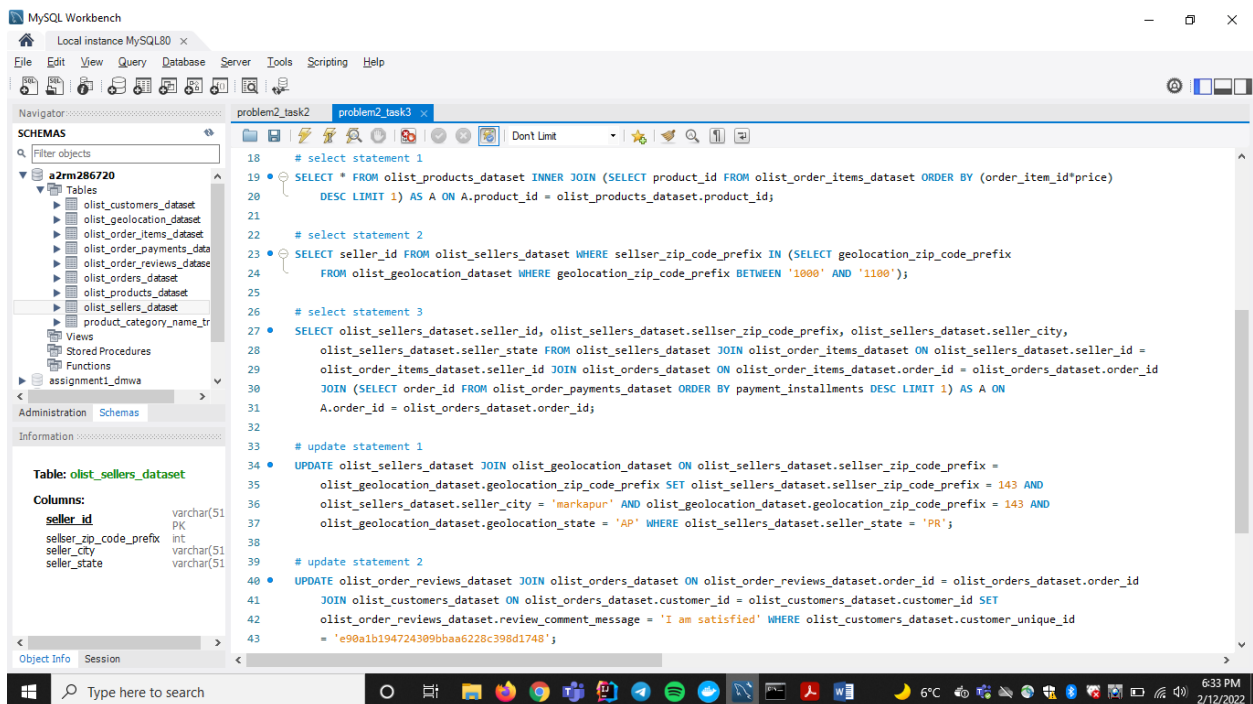
I have also added the screenshots of it as proof in this document:



```
1 • USE a2rm286720;
2
3 • SET autocommit = 0;
4
5 • START TRANSACTION;
6
7 # insert statement 1
8 • INSERT INTO olist_customers_dataset (customer_id, customer_unique_id, customer_zip_code_prefix, customer_city, customer_state) VALUES
9   ('yugrfo617fgqufyhgr36fre8', '3uygfi45f7evu6e74fr8u7we', '23453', 'halifax', 'NS'),
10  ('eyrgfu5quyfu3y4rfvge6w483hrey', '437gf34gr79giw84', '785', 'mumbai', 'Karnataka'),
11  ('fyg7e5t3874wltirwufhgrw4urhof83', 'i34gt7824tr93lgrf723l', '1912', 'hyderabad', 'telangana'),
12  ('ugfo3746f73twrg7f12qgef', 'wu4tgr78wgtf7834gfu7wk17gf', '2119', 'hyderabad', 'telangana');
13
14 # insert statement 2
15 • INSERT INTO product_category_name_translation (product_category_name, product_category_name_english) VALUES ('lait', 'milk'),
16  ('yaourt', 'yogurt'), ('fromage', 'cheese'), ('beurre', 'butter');
17
18 # select statement 1
19 • SELECT * FROM olist_products_dataset INNER JOIN (SELECT product_id FROM olist_order_items_dataset ORDER BY (order_item_id*price)
20   DESC LIMIT 1) AS A ON A.product_id = olist_products_dataset.product_id;
21
22 # select statement 2
23 • SELECT seller_id FROM olist_sellers_dataset WHERE seller_zip_code_prefix IN (SELECT geolocation_zip_code_prefix
24   FROM olist_geolocation_dataset WHERE geolocation_zip_code_prefix BETWEEN '1000' AND '1100');
25
26 # select statement 3
```

Table: olist\_sellers\_dataset

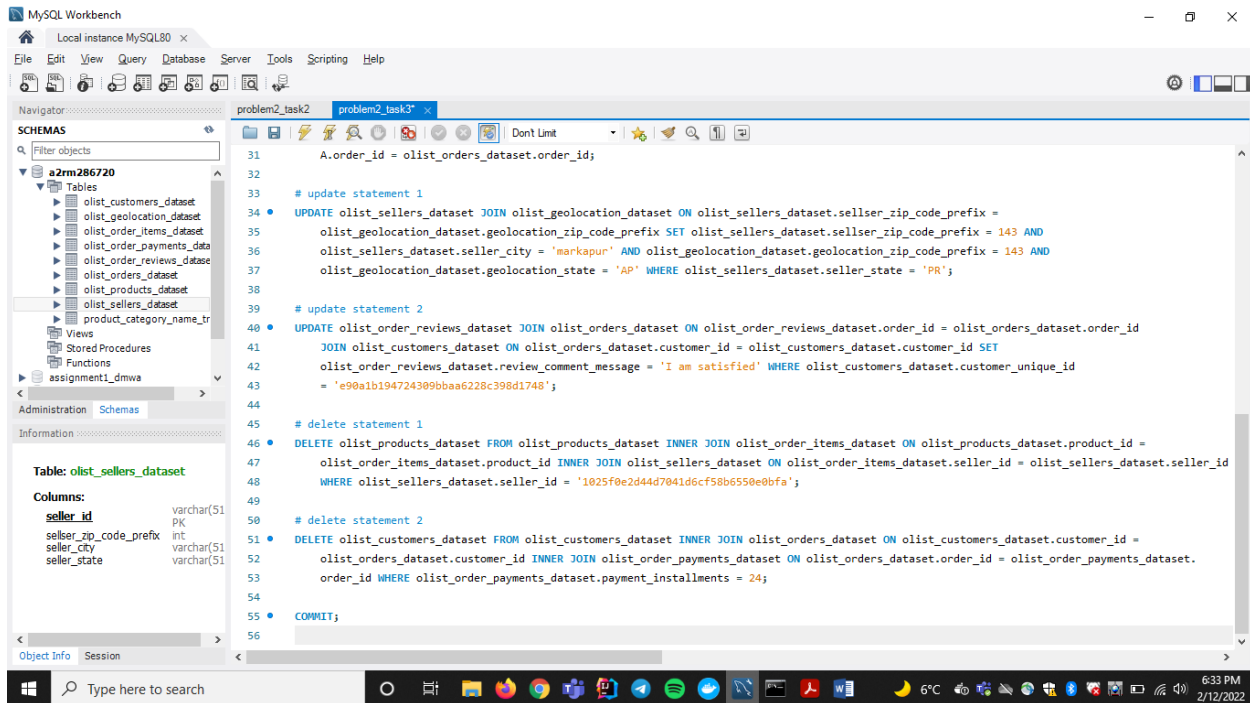
Columns:	
seller_id	varchar(51) PK
seller_zip_code_prefix	int
seller_city	varchar(51)
seller_state	varchar(51)



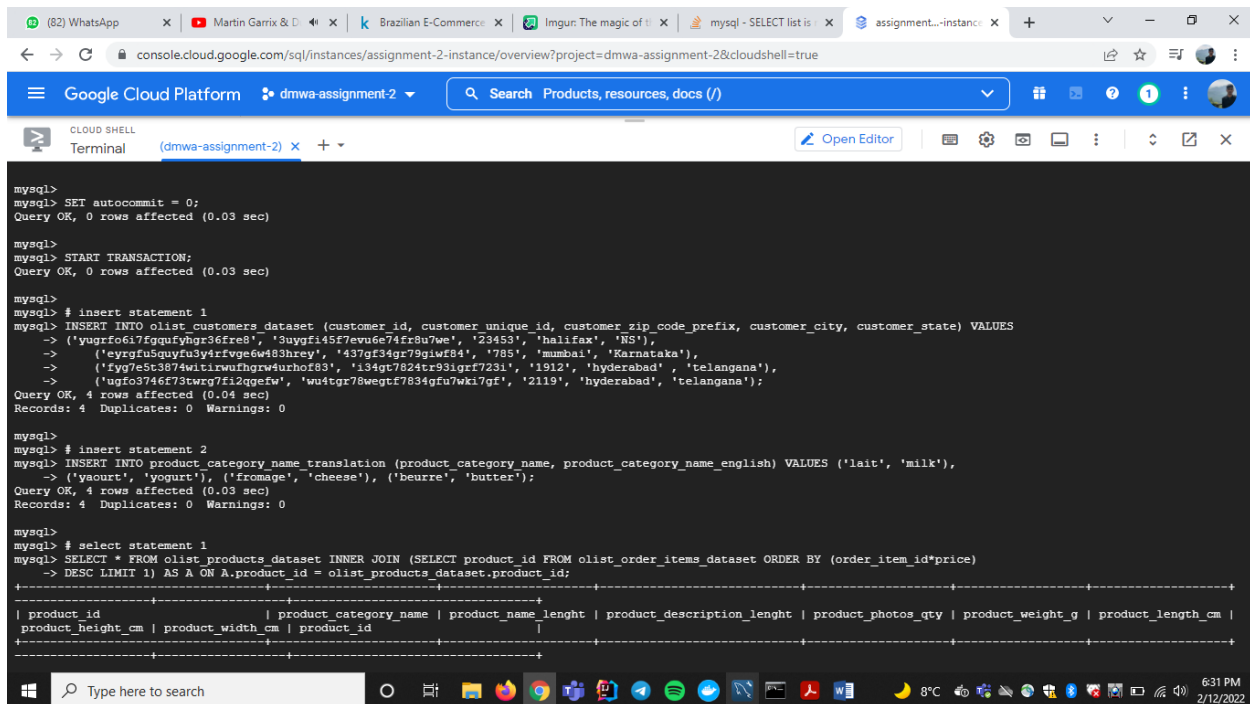
```
18 # select statement 1
19 • SELECT * FROM olist_products_dataset INNER JOIN (SELECT product_id FROM olist_order_items_dataset ORDER BY (order_item_id*price)
20   DESC LIMIT 1) AS A ON A.product_id = olist_products_dataset.product_id;
21
22 # select statement 2
23 • SELECT seller_id FROM olist_sellers_dataset WHERE seller_zip_code_prefix IN (SELECT geolocation_zip_code_prefix
24   FROM olist_geolocation_dataset WHERE geolocation_zip_code_prefix BETWEEN '1000' AND '1100');
25
26 # select statement 3
27 • SELECT olist_sellers_dataset.seller_id, olist_sellers_dataset.seller_zip_code_prefix, olist_sellers_dataset.seller_city,
28   olist_sellers_dataset.seller_state FROM olist_sellers_dataset JOIN olist_order_items_dataset ON olist_sellers_dataset.seller_id =
29   olist_order_items_dataset.seller_id JOIN olist_orders_dataset ON olist_order_items_dataset.order_id = olist_orders_dataset.order_id
30   JOIN (SELECT order_id FROM olist_order_payments_dataset ORDER BY payment_installments DESC LIMIT 1) AS A ON
31   A.order_id = olist_orders_dataset.order_id;
32
33 # update statement 1
34 • UPDATE olist_sellers_dataset JOIN olist_geolocation_dataset ON olist_sellers_dataset.seller_zip_code_prefix =
35   olist_geolocation_dataset.geolocation_zip_code_prefix SET olist_sellers_dataset.seller_zip_code_prefix = 143 AND
36   olist_sellers_dataset.seller_city = 'markapur' AND olist_geolocation_dataset.geolocation_zip_code_prefix = 143 AND
37   olist_geolocation_dataset.geolocation_state = 'AP' WHERE olist_sellers_dataset.seller_state = 'PR';
38
39 # update statement 2
40 • UPDATE olist_order_reviews_dataset JOIN olist_orders_dataset ON olist_order_reviews_dataset.order_id = olist_orders_dataset.order_id
41   JOIN olist_customers_dataset ON olist_orders_dataset.customer_id = olist_customers_dataset.customer_id SET
42   olist_order_reviews_dataset.review_comment_message = 'I am satisfied' WHERE olist_customers_dataset.customer_unique_id
43   = 'e90ab1b194724309bbaa6228c398d1748';
```

Table: olist\_sellers\_dataset

Columns:	
seller_id	varchar(51) PK
seller_zip_code_prefix	int
seller_city	varchar(51)
seller_state	varchar(51)



Proof that the transaction was successful without any issues and also to show the transaction time/queries time:



```
(82) WhatsApp x Martin Garrix & D x Brazilian E-Commerce x Imgur: The magic of i x mysql - SELECT list is x assignment...-instance x +
console.cloud.google.com/sql/instances/assignment-2-instance/overview?project=dmwa-assignment-2&cloudshell=true
Google Cloud Platform dmwa-assignment-2 Search Products, resources, docs (/)
CLOUD SHELL Terminal (dmwa-assignment-2) x + Open Editor

| product_id | product_category_name | product_name_lenght | product_description_lenght | product_photos_qty | product_weight_g | product_length_cm |
|-----|-----|-----|-----|-----|-----|-----|
| 489ae2aa008f021502940f251d4cce7f | utilidades domesticas | 31 | 875 | 2 | 30000 | 60 |
| 61 | 33 | 489ae2aa008f021502940f251d4cce7f |

1 row in set (0.08 sec)

mysql>
mysql> # select statement 2
mysql> SELECT seller_id FROM olist_sellers_dataset WHERE sellser_zip_code_prefix IN (SELECT geolocation_zip_code_prefix
-> FROM olist_geolocation_dataset WHERE geolocation_zip_code_prefix BETWEEN '1000' AND '1100');
+-----+
| seller_id |
+-----+
| 09bad886111255c5b5030314fc7f1a4a |
| 14a08204d03bb6b6bde8029f801ae0eb |
| 1d503743d2526f030cc2c99540ec009c |
| 2d34636510ba8550490722f8f0ba2e4 |
| 5d378b73ab7dd6f0418d743e5dcb0bd1 |
| 630008e0f062605a415d694489f6f82e |
| 6902157e16728322d61a1321fa79e58 |
| 6da1992f915d77be95d7fa48b36904af |
| 7040e82f899a04d1b434b795a43b4617 |
| 784ba75dd9d20200c4ced3d7a77141a |
| 82921991f8b57b04560bb3bb40d49 |
| 8602a61d680a10a82cceeada0d99ea3d |
| 9d39404483343dd5c7ae254494d8388 |
| b6c6854d4d92a5f6f46be869da3f1a1a |
| c1dde11f12d05c478f5de2d7319ad3b2 |
| c84592044b180dec206770c38603814b |
| d594982f2877af63ace38ea1fca27c76 |
| dd53f1bb78714a40e7954c3be6d745 |
| e0487761face83d64fcada2408959a36 |

mysql>
```

```
(82) WhatsApp x Martin Garrix & D x Brazilian E-Commerce x Imgur: The magic of i x mysql - SELECT list is x assignment...-instance x +
console.cloud.google.com/sql/instances/assignment-2-instance/overview?project=dmwa-assignment-2&cloudshell=true
Google Cloud Platform dmwa-assignment-2 Search Products, resources, docs (/)
CLOUD SHELL Terminal (dmwa-assignment-2) x + Open Editor

| e0487761face83d64fcada2408959a36 |
| e5cbe890e679490127e9a390b46bd20 |
| f049a72cf58fd31b11f8919cade515e7 |
| f4aba7c0bca51484c30ab7bdc34bcd41 |
+-----+
22 rows in set (0.41 sec)

mysql>
mysql> # select statement 3
mysql> SELECT olist_sellers_dataset.seller_id, olist_sellers_dataset.sellser_zip_code_prefix, olist_sellers_dataset.seller_city,
-> olist_sellers_dataset.seller_state FROM olist_sellers_dataset JOIN olist_order_items_dataset ON olist_sellers_dataset.seller_id =
-> olist_order_items_dataset.seller_id JOIN olist_orders_dataset ON olist_order_items_dataset.order_id = olist_orders_dataset.order_id
-> JOIN (SELECT order_id FROM olist_order_payments_dataset ORDER BY payment_installments DESC LIMIT 1) AS A ON
-> A.order_id = olist_orders_dataset.order_id;
+-----+-----+-----+-----+
| seller_id | sellser_zip_code_prefix | seller_city | seller_state |
+-----+-----+-----+-----+
| 1df3c1969d7b9473ff13abc | 4160 | sao paulo | SP |
| 796ee4304cadf1ad3050fb7 | 21840 | rio de janeiro | RJ |
| fd57276679edd00e07089e5 | 87015 | maringa | PR |
+-----+-----+-----+-----+
3 rows in set (0.07 sec)

mysql>
mysql> # update statement 1
mysql> UPDATE olist_sellers_dataset JOIN olist_geolocation_dataset ON olist_sellers_dataset.seller_zip_code_prefix =
-> olist_geolocation_dataset.geolocation_zip_code_prefix SET olist_sellers_dataset.seller_zip_code_prefix = 143 AND
-> olist_sellers_dataset.seller_city = 'markapur' AND olist_geolocation_dataset.geolocation_zip_code_prefix = 143 AND
-> olist_geolocation_dataset.geolocation_state = 'AP' WHERE olist_sellers_dataset.seller_state = 'PR';
Query OK, 0 rows affected (0.03 sec)
Rows matched: 0 Changed: 0 Warnings: 0

mysql>
mysql> # update statement 2
mysql> UPDATE olist_order_reviews_dataset JOIN olist_orders_dataset ON olist_order_reviews_dataset.order_id = olist_orders_dataset.order_id
-> JOIN olist_customers_dataset ON olist_orders_dataset.customer_id = olist_customers_dataset.customer_id SET
```



(82) WhatsApp x Martin Garrix & D... x Brazilian E-Commerce x Imgun The magic of t... x mysql - SELECT list is... x assignment...-instance x +

console.cloud.google.com/sql/instances/assignment-2-instance/overview?project=dmwa-assignment-2&cloudshell=true

Google Cloud Platform dmwa-assignment-2 Search Products, resources, docs (/)

CLOUD SHELL Terminal (dmwa-assignment-2) + - Open Editor

```
mysql> # update statement 1
mysql> UPDATE olist_sellers_dataset JOIN olist_geolocation_dataset ON olist_sellers_dataset.seller_zip_code_prefix =
-> olist_geolocation_dataset.geolocation_zip_code_prefix SET olist_sellers_dataset.seller_zip_code_prefix = 143 AND
-> olist_sellers_dataset.seller_city = 'Marapuru' AND olist_geolocation_dataset.geolocation_zip_code_prefix = 143 AND
-> olist_geolocation_dataset.geolocation_state = 'AP' WHERE olist_sellers_dataset.seller_state = 'PR';
Query OK, 0 rows affected (0.03 sec)
Rows matched: 0 Changed: 0 Warnings: 0

mysql>
mysql> # update statement 2
mysql> UPDATE olist_order_reviews_dataset JOIN olist_orders_dataset ON olist_order_reviews_dataset.order_id = olist_orders_dataset.order_id
-> JOIN olist_customers_dataset ON olist_orders_dataset.customer_id = olist_customers_dataset.customer_id SET
-> olist_order_reviews_dataset.review_comment_message = 'I am satisfied' WHERE olist_customers_dataset.customer_unique_id
-> = 'e90alb194724309bba6228c398d1748';
Query OK, 1 row affected (0.09 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql>
mysql> # delete statement 1
mysql> DELETE olist_products_dataset FROM olist_products_dataset INNER JOIN olist_order_items_dataset ON olist_products_dataset.product_id =
-> olist_order_items_dataset.product_id INNER JOIN olist_sellers_dataset ON olist_order_items_dataset.seller_id = olist_sellers_dataset.seller_id
-> WHERE olist_sellers_dataset.seller_id = '1025f0e2d4d7041d6cf58b6550e0bfa';
Query OK, 154 rows affected (0.09 sec)

mysql>
mysql> # delete statement 2
mysql> DELETE olist_customers_dataset FROM olist_customers_dataset INNER JOIN olist_orders_dataset ON olist_customers_dataset.customer_id =
-> olist_orders_dataset.customer_id INNER JOIN olist_order_payments_dataset ON olist_orders_dataset.order_id = olist_order_payments_dataset.
-> order_id WHERE olist_order_payments_dataset.payment_installments = 24;
Query OK, 18 rows affected (0.09 sec)

mysql>
mysql> COMMIT;
Query OK, 0 rows affected (0.04 sec)

mysql>
```

Type here to search 8°C 6:31 PM 2/12/2022

## **Task 4 – Perform Distributed Transaction**

Queries for task 4 as required are:

```
USE a2rm286720;
```

```
SET autocommit = 0;
```

```
START TRANSACTION;
```

```
# insert statement 1
```

```
INSERT INTO olist_geolocation_dataset (geolocation_zip_code_prefix, geolocation_lat,
geolocation_lng, geolocation_city, geolocation_state) VALUES ('10101', '-27.23737', '-
46.64738', 'hyderabad', 'TS'), ('10102', '-87.23634', '-76.4847', 'bangalore', 'delhi'), ('10103', '-
25.72528', '-26.926529', 'pune', 'maharashtra');
```

```
# insert statement 2
```

```
INSERT INTO olist_order_reviews_dataset (review_id, order_id, review_score,
review_comment_title, review_comment_message, review_creation_date,
review_answer_timestamp) VALUES ('8948949dfbviur79283',
'009905140e9f8cc35d5be897937381db', '3', '', 'not bad and not good', '2/13/2022 0:00',
'2/14/2022 17:27'), ('ebf736gifwge7843', '00a379dfab816a83741012b71b264098', '4', 'review',
'good only', '12/19/2020 8:00', '5/9/2021 23:00');
```

```
# select statement 1
```

```
SELECT olist_customers_dataset.customer_id, olist_customers_dataset.customer_city,
olist_customers_dataset.customer_state FROM olist_customers_dataset JOIN
olist_orders_dataset ON olist_customers_dataset.customer_id = olist_orders_dataset.customer_id
WHERE olist_orders_dataset.customer_id = (SELECT customer_id FROM olist_orders_dataset
```

```
WHERE order_id = (SELECT order_id FROM olist_order_payments_dataset GROUP BY  
order_id ORDER BY sum(payment_value) DESC LIMIT 1));
```

```
# select statement 2
```

```
SELECT MAX(olist_order_payments_dataset.payment_value) FROM  
olist_order_payments_dataset WHERE order_id IN (SELECT order_id FROM  
olist_orders_dataset JOIN olist_customers_dataset ON olist_orders_dataset.customer_id =  
olist_customers_dataset.customer_id WHERE olist_orders_dataset.customer_id =  
'0004164d20a9e969af783496f3408652');
```

```
# select statement 3
```

```
SELECT olist_orders_dataset.customer_id FROM olist_orders_dataset JOIN  
olist_order_reviews_dataset ON olist_orders_dataset.order_id =  
olist_order_reviews_dataset.order_id WHERE olist_orders_dataset.order_id IN (SELECT  
olist_order_reviews_dataset.order_id FROM olist_order_reviews_dataset WHERE review_score  
= 1);
```

```
# update statement 1
```

```
UPDATE olist_order_items_dataset JOIN olist_products_dataset ON  
olist_order_items_dataset.product_id = olist_products_dataset.product_id SET  
olist_order_items_dataset.price = (olist_order_items_dataset.price * 1.1) WHERE  
olist_order_items_dataset.product_id IN (SELECT product_id FROM olist_products_dataset  
WHERE product_weight_g > 10000 AND product_length_cm > 100);
```

```
# update statement 2
```

```
UPDATE olist_sellers_dataset JOIN olist_order_items_dataset ON olist_sellers_dataset.seller_id  
= olist_order_items_dataset.seller_id SET olist_sellers_dataset.sellser_zip_code_prefix = 190699
```

```
WHERE olist_sellers_dataset.seller_id IN (SELECT seller_id FROM olist_order_items_dataset  
WHERE order_item_id >= 3);
```

```
# delete statement 1
```

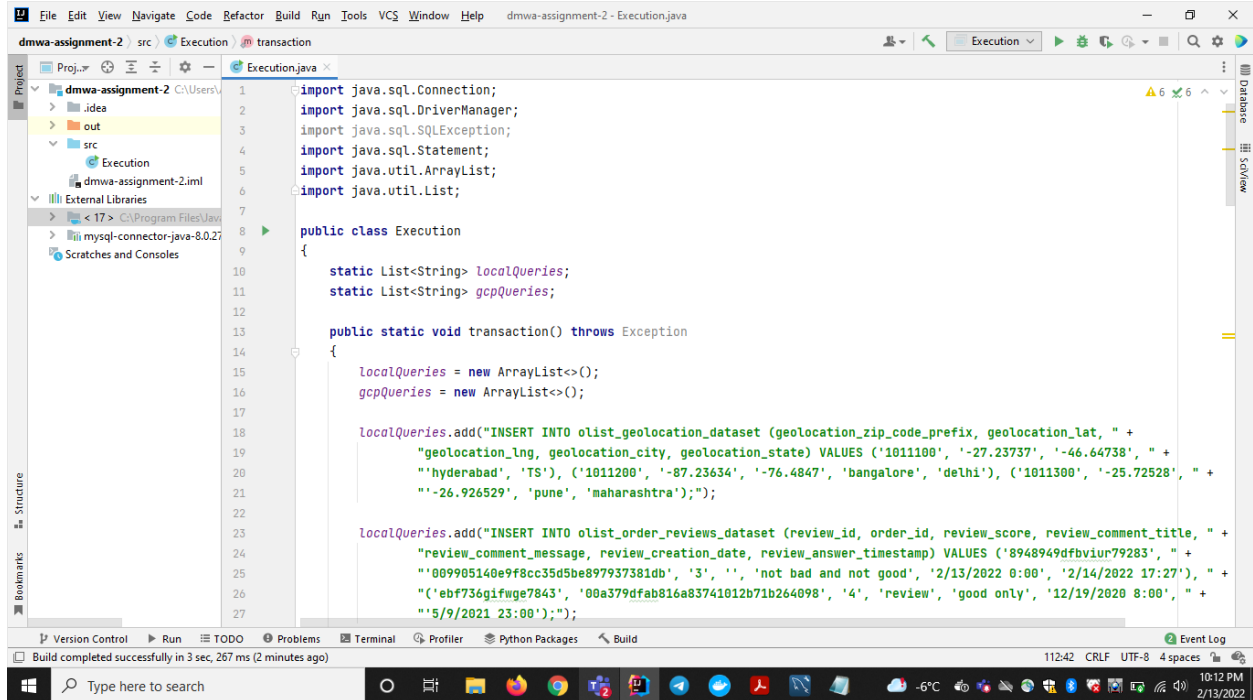
```
DELETE FROM product_category_name_translation WHERE  
product_category_name_translation.product_category_name IN (SELECT  
product_category_name FROM olist_products_dataset WHERE product_description_lenght >  
1000);
```

```
# delete statement 2
```

```
DELETE olist_products_dataset FROM olist_products_dataset WHERE  
olist_products_dataset.product_id IN (SELECT product_id FROM olist_order_items_dataset  
WHERE seller_id IN (SELECT seller_id FROM olist_sellers_dataset WHERE  
sellser_zip_code_prefix = 14093));
```

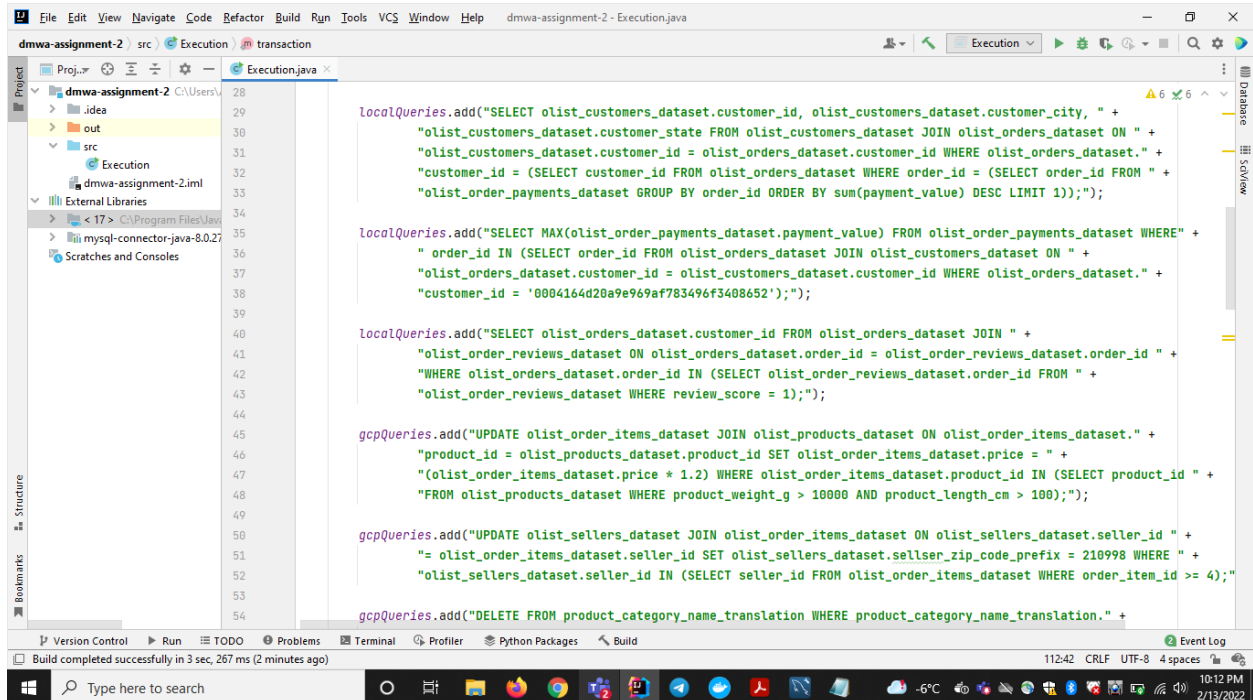
```
COMMIT;
```

I have also added the screenshots of the code as proof in this document:



This screenshot shows the first part of the Java code in 'Execution.java'. The code includes imports for SQL-related classes and a public class 'Execution' with a 'transaction()' method. The method initializes two ArrayLists, 'localQueries' and 'gcpQueries', and adds several SQL queries to them. The queries include INSERT statements for 'olist\_geolocation\_dataset' and 'olist\_order\_reviews\_dataset'.

```
1 import java.sql.Connection;
2 import java.sql.DriverManager;
3 import java.sql.SQLException;
4 import java.sql.Statement;
5 import java.util.ArrayList;
6 import java.util.List;
7
8 public class Execution
9 {
10     static List<String> localQueries;
11     static List<String> gcpQueries;
12
13     public static void transaction() throws Exception
14     {
15         localQueries = new ArrayList<>();
16         gcpQueries = new ArrayList<>();
17
18         localQueries.add("INSERT INTO olist_geolocation_dataset (geolocation_zip_code_prefix, geolocation_lat, " +
19             "geolocation_lng, geolocation_city, geolocation_state) VALUES ('1011100', '-27.23737', '-46.64738', " +
20             "'hyderabad', 'TS'), ('1011200', '-87.23634', '-76.4847', 'bangalore', 'delhi'), ('1011300', '-25.72528', " +
21             "'-26.926529', 'pune', 'maharashtra');");
22
23         localQueries.add("INSERT INTO olist_order_reviews_dataset (review_id, order_id, review_score, review_comment_title, " +
24             "review_comment_message, review_creation_date, review_answer_timestamp) VALUES ('8948949dfbvir79283', " +
25             "'009905140e9f8cc35d5be897937381db', '3', '', 'not bad and not good', '2/13/2022 0:00', '2/14/2022 17:27'), " +
26             "('ebf736g1fwge7843', '00a379dfab816a83741012b71b264098', '4', 'review', 'good only', '12/19/2020 8:00', " +
27             "'5/9/2021 23:00');");
```



This screenshot shows the second part of the Java code in 'Execution.java'. The code continues the 'transaction()' method by adding more SQL queries to the 'localQueries' and 'gcpQueries' ArrayLists. The queries include SELECT statements for customer information, order payments, and order reviews, as well as UPDATE and DELETE statements for order items and product categories.

```
28
29 localQueries.add("SELECT olist_customers_dataset.customer_id, olist_customers_dataset.customer_city, " +
30     "olist_customers_dataset.customer_state FROM olist_customers_dataset JOIN olist_orders_dataset ON " +
31     "olist_customers_dataset.customer_id = olist_orders_dataset.customer_id WHERE olist_orders_dataset." +
32     "customer_id = (SELECT customer_id FROM olist_orders_dataset WHERE order_id = (SELECT order_id FROM " +
33     "olist_order_payments_dataset GROUP BY order_id ORDER BY sum(payment_value) DESC LIMIT 1));");
34
35 localQueries.add("SELECT MAX(olist_order_payments_dataset.payment_value) FROM olist_order_payments_dataset WHERE " +
36     "order_id IN (SELECT order_id FROM olist_orders_dataset JOIN olist_customers_dataset ON " +
37     "olist_orders_dataset.customer_id = olist_customers_dataset.customer_id WHERE olist_orders_dataset." +
38     "customer_id = '0004164d20a9e969af783496f3408652');");
39
40 localQueries.add("SELECT olist_orders_dataset.customer_id FROM olist_orders_dataset JOIN " +
41     "olist_order_reviews_dataset ON olist_orders_dataset.order_id = olist_order_reviews_dataset.order_id " +
42     "WHERE olist_orders_dataset.order_id IN (SELECT olist_order_reviews_dataset.order_id FROM " +
43     "olist_order_reviews_dataset WHERE review_score = 1);");
44
45 gcpQueries.add("UPDATE olist_order_items_dataset JOIN olist_products_dataset ON olist_order_items_dataset." +
46     "product_id = olist_products_dataset.product_id SET olist_order_items_dataset.price = " +
47     "(olist_order_items_dataset.price * 1.2) WHERE olist_order_items_dataset.product_id IN (SELECT product_id " +
48     "FROM olist_products_dataset WHERE product_weight_g > 10000 AND product_length_cm > 100);");
49
50 gcpQueries.add("UPDATE olist_sellers_dataset JOIN olist_order_items_dataset ON olist_sellers_dataset.seller_id " +
51     "= olist_order_items_dataset.seller_id SET olist_sellers_dataset.sellser_zip_code_prefix = 210998 WHERE " +
52     "olist_sellers_dataset.seller_id IN (SELECT seller_id FROM olist_order_items_dataset WHERE order_item_id >= 4);");
53
54 gcpQueries.add("DELETE FROM product_category_name_translation WHERE product_category_name_translation." +
```

```
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help dmwa-assignment-2 - Execution.java

dmwa-assignment-2 | src | Execution | transaction | Execution.java | Database |
Project | dmwa-assignment-2 | .idea | out | src | Execution | dmwa-assignment-2.iml | External Libraries | mysql-connector-java-8.0.27 |
Build completed successfully in 3 sec, 267 ms (2 minutes ago) | 112:42 CRLF UTF-8 4 spaces | 10:12 PM 2/13/2022

52 "olist_sellers_dataset.seller_id IN (SELECT seller_id FROM olist_order_items_dataset WHERE order_id
53
54 gcpQueries.add("DELETE FROM product_category_name_translation WHERE product_category_name_translation." +
55 "product_category_name IN (SELECT product_category_name FROM olist_products_dataset WHERE " +
56 "product_description_length > 9000);");
57
58 gcpQueries.add("DELETE olist_products_dataset FROM olist_products_dataset WHERE olist_products_dataset.product_id " +
59 "IN (SELECT product_id FROM olist_order_items_dataset WHERE seller_id IN (SELECT seller_id FROM " +
60 "olist_sellers_dataset WHERE seller_zip_code_prefix = 14091));");
61
62 String transactionStart = "START TRANSACTION;";
63 String transactionEnd = "COMMIT;";
64 long startTime = 0;
65 long endTime = 0;
66
67 try
68 {
69     Connection connect1;
70     Connection connect2;
71
72     Class.forName("com.mysql.cj.jdbc.Driver");
73     connect1 = DriverManager.getConnection("jdbc:mysql://localhost:3306/a2rm286720", user: "root", password: "Wishhh@20
74     Statement stat1 = connect1.createStatement();
75     connect1.setAutoCommit(false);
76     startTime = System.currentTimeMillis();
77
78     stat1.execute(transactionStart);
```

```
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help dmwa-assignment-2 - Execution.java

dmwa-assignment-2 | src | Execution | transaction | Execution.java | Database |
Project | dmwa-assignment-2 | .idea | out | src | Execution | dmwa-assignment-2.iml | External Libraries | mysql-connector-java-8.0.27 |
Build completed successfully in 3 sec, 267 ms (2 minutes ago) | 112:42 CRLF UTF-8 4 spaces | 10:12 PM 2/13/2022

76 startTime = System.currentTimeMillis();
77
78 stat1.execute(transactionStart);
79 System.out.println("Transaction started in local instance.");
80 stat1.execute(localQueries.get(0));
81 System.out.println("First insert query executed.");
82 stat1.execute(localQueries.get(1));
83 System.out.println("Second insert query executed.");
84 stat1.execute(localQueries.get(2));
85 System.out.println("First select query executed.");
86 stat1.execute(localQueries.get(3));
87 System.out.println("Second select query executed.");
88 stat1.execute(localQueries.get(4));
89 System.out.println("Third select query executed.");
90
91 connect2 = DriverManager.getConnection("jdbc:mysql://34.72.255.252:3306/a2rm286720", user: "root", password: "Wishhh
92 Statement stat2 = connect2.createStatement();
93 connect2.setAutoCommit(false);
94
95 stat2.execute(transactionStart);
96 System.out.println("Transaction started in VM instance.");
97 stat2.execute(gcpQueries.get(0));
98 System.out.println("First update query executed.");
99 stat2.execute(gcpQueries.get(1));
100 System.out.println("Second update query executed.");
101 stat2.execute(gcpQueries.get(2));
102 System.out.println("First delete query executed.");
```

```
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help dmwa-assignment-2 - Execution.java

dmwa-assignment-2 src > Execution > transaction

Project
  dmwa-assignment-2
  > idea
  > out
  > src
  > Execution
  > dmwa-assignment-2.iml
  > External Libraries
  > < 17 > C:\Program Files\Java
  > mysql-connector-java-8.0.27
  > Scratches and Consoles

Execution.java
100 System.out.println("Second update query executed.");
101 stat2.execute(gcpQueries.get(2));
102 System.out.println("First delete query executed.");
103 stat2.execute(gcpQueries.get(3));
104 System.out.println("Second delete query executed.");
105
106 stat1.execute(transactionEnd);
107 System.out.println("Transaction committed in local instance.");
108 stat2.execute(transactionEnd);
109 System.out.println("Transaction committed in VM instance.");
110 endTime = System.currentTimeMillis();
111
112 connect1.setAutoCommit(true);
113 connect2.setAutoCommit(true);
114
115 connect1.close();
116 connect2.close();
117
118 catch (Exception e)
119 {
120     e.printStackTrace();
121 }
122
123 System.out.println("Transaction start time is: " + startTime);
124 System.out.println("Transaction end time is: " + endTime);
125 System.out.println("The total transaction execution time is: " + ((endTime - startTime) / 1000) + " seconds");
126
127
128
129
130
131
132
133

Build completed successfully in 3 sec, 267 ms (2 minutes ago)
112:42 CRLF UTF-8 4 spaces 10:12 PM 2/13/2022
```

```
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help dmwa-assignment-2 - Execution.java

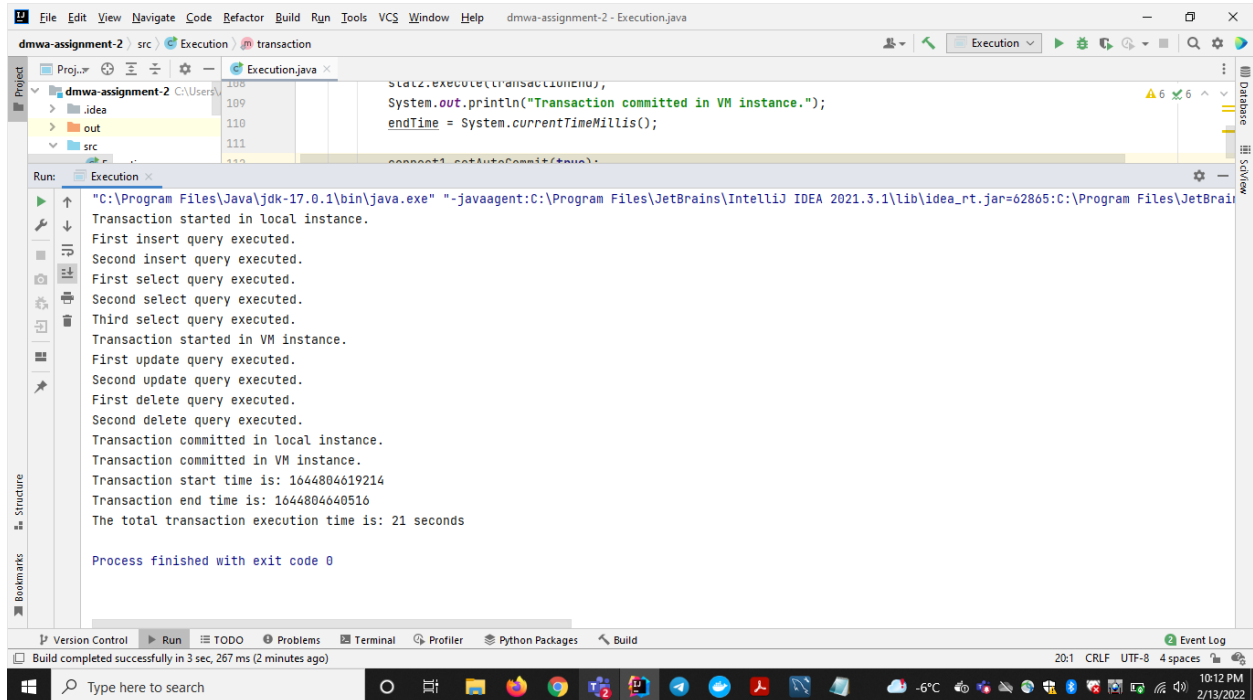
dmwa-assignment-2 src > Execution > transaction

Project
  dmwa-assignment-2
  > idea
  > out
  > src
  > Execution
  > dmwa-assignment-2.iml
  > External Libraries
  > < 17 > C:\Program Files\Java
  > mysql-connector-java-8.0.27
  > Scratches and Consoles

Execution.java
112 connect1.setAutoCommit(true);
113 connect2.setAutoCommit(true);
114
115 connect1.close();
116 connect2.close();
117
118 catch (Exception e)
119 {
120     e.printStackTrace();
121 }
122
123 System.out.println("Transaction start time is: " + startTime);
124 System.out.println("Transaction end time is: " + endTime);
125 System.out.println("The total transaction execution time is: " + ((endTime - startTime) / 1000) + " seconds");
126
127
128 public static void main(String[] args) throws Exception
129 {
130     transaction();
131 }
132
133

Build completed successfully in 3 sec, 267 ms (2 minutes ago)
112:42 CRLF UTF-8 4 spaces 10:12 PM 2/13/2022
```

Proof that the transaction was successful without any issues and also to show the transaction time/queries time:



The screenshot displays the IntelliJ IDEA IDE interface. The top toolbar shows the 'Run' button (a green play icon) with a dropdown menu set to 'Execution'. The 'Run' button has been clicked, and the 'Run' tab is active, showing the execution output. The output log contains the following text:

```
"C:\Program Files\Java\jdk-17.0.1\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2021.3.1\lib\idea_rt.jar=62865:C:\Program Files\JetBrains\IntelliJ IDEA 2021.3.1\bin" -Didea.config.path=C:\Program Files\JetBrains\IntelliJ IDEA 2021.3.1\config -Didea.system.path=C:\Program Files\JetBrains\IntelliJ IDEA 2021.3.1\system -Didea.version=2021.3.1 -jar dmwa-assignment-2.jar
Transaction started in local instance.
First insert query executed.
Second insert query executed.
First select query executed.
Second select query executed.
Third select query executed.
Transaction started in VM instance.
First update query executed.
Second update query executed.
First delete query executed.
Second delete query executed.
Transaction committed in local instance.
Transaction committed in VM instance.
Transaction start time is: 1644804619214
Transaction end time is: 1644804640516
The total transaction execution time is: 21 seconds

Process finished with exit code 0
```

The bottom status bar indicates 'Build completed successfully in 3 sec, 267 ms (2 minutes ago)'. The system tray at the bottom right shows the date and time as '10:12 PM 2/13/2022'.



### Worksheet

Tasks	Execution Time	Transaction Query	Your Observations
Task 2	33.546 seconds	Mentioned above in the document	Technically, the update statements took longer than the rest of the statements. Insert statements were the fastest to get executed. Delete statements are the second fastest whereas the select statements were also fast enough except one statement, where it took almost 10 sec.
Task 3	1.03 seconds	Mentioned above in the document	Each and every statement that was executed on VM instance happened real real quick. The execution time difference between the local instance and VM instance is unbelievable. No statement on VM instance took more than 0.5 seconds to get executed and show the result.
Task 4	21.302 seconds	Mentioned above in the document	Even though this was a distributed transaction, the connections to the databases and switching the connections

			<p>happened fast enough.</p> <p>Execution of the queries also happened in a good time, not fast not slow. So I can say that even distributed transactions happen fast enough.</p>
--	--	--	---

My gitlab assignment 2 repo url : <https://git.cs.dal.ca/rguturu/csci-5408-w2022-b00871849-gutururamamohanvishnu/-/tree/main/assignment-2>