ASSIGNMENT NO: 2

Sumary: This assignment gives the proper understanding of the working of transaction block having various operations like SELECT and UPDATE in both kind of databases i.e. local and remote (GCP).

PROBLEM -1:

For Problem 1, I worked on the dataset or the link: (https://parks.novascotia.ca/)

Part -1:

Converting final ERD/EERD into a physical design and translating each entity into one table in physical design. (Steps shown below)

CREATE LOCAL DATABASE:

Part -2:

1. I created database in MySQL RDBMS: Local database as "a2_loc<b00899642>" .
2. I made 14 tables having 5 data rows for each table in "a2_loc<b00899642>" database.
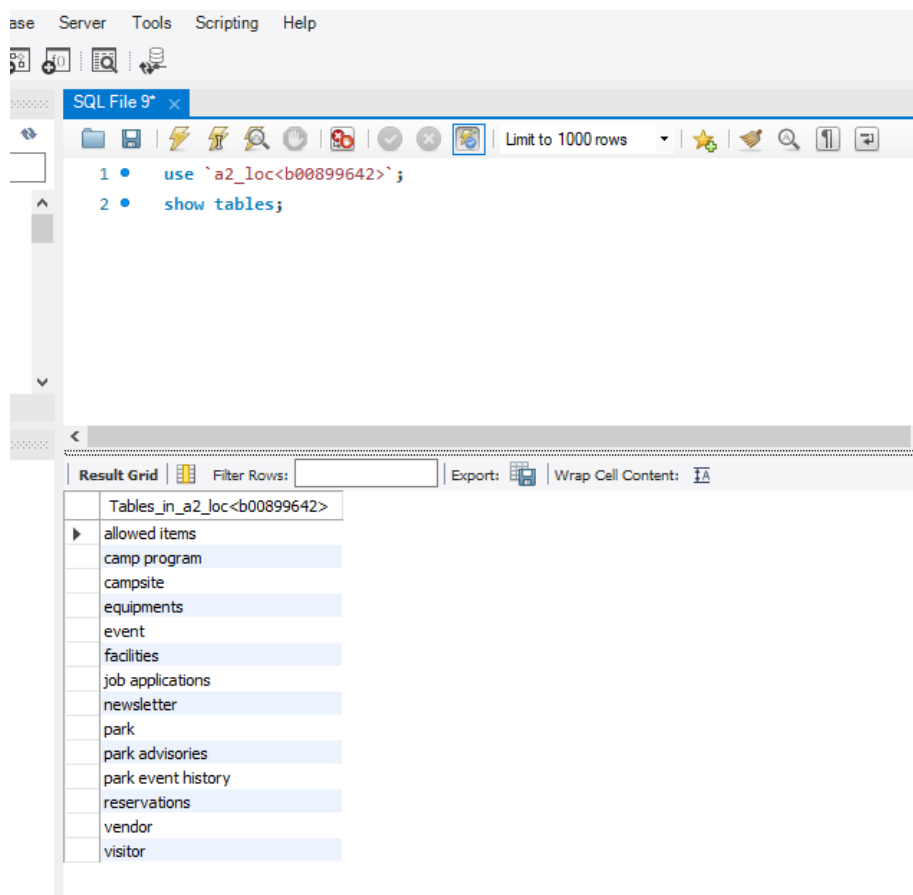
Part – 2.a):

1. Performed 1 SELECT and 1 UPDATE operation. The steps for this are as follows:

Wrote 1 transaction block with 1 SELECT and 1 UPDATE operation.

- 1 SELECT operation on table - Park:

- The below screenshot is of table - Vendor before 1 UPDATE operation:

- Below screenshot shows - 1 UPDATE operation on table - Vendor after 1 UPDATE query:
  In the column, V_name – "Robert" changed to "Oliver" having V_id = 650

Part- 2.b):

Executing the transaction and recording the execution time (in milliseconds). To get time in milliseconds, I used mysql> set profiling = 1; To show time in millisecondsmysql, I mentioned> show profiles;

Execution Time:

Query_ID	Duration	Query
33	0.00021150	START TRANSACTION
34	0.00034450	SELECT *FROM park LIMIT 0, 1000
35	0.00038275	SELECT *FROM vendor WHERE V_id = 650 LIMI...
36	0.00054000	UPDATE vendor SET V_name = "Oliver" WHERE ...
37	0.00013975	COMMIT
38	0.00041025	use `a2_loc<b00899642>`
39	0.00024250	SELECT DATABASE()
40	0.00067475	SET AUTOCOMMIT = 0
41	0.00024125	SET profiling = 1
42	0.00010025	SHOW WARNINGS
43	0.00028175	START TRANSACTION
44	0.00049700	SELECT *FROM park LIMIT 0, 1000
45	0.00050350	SELECT *FROM vendor WHERE V_id = 650 LIMI...
46	0.00103700	UPDATE vendor SET V_name = "Oliver" WHERE ...
47	0.00027850	COMMIT

Part - 2.c)

Added 1 more SELECT and 1 more UPDATE operation, and measured the execution time.

Query:

```
1  use `a2_loc<b00899642>`;
2  SET AUTOCOMMIT = 0;
3  SET profiling = 1;
4  START TRANSACTION;
5  SELECT *FROM park;
6  SELECT *FROM vendor WHERE V_id = 650;
7  UPDATE vendor SET V_name = "Oliver" WHERE V_id = 650;
8  SELECT *FROM reservations;
9  SELECT *FROM `job applications` WHERE JA_id = 1400;
10 UPDATE `job applications` SET Requirement = "Education Background" WHERE  JA_id = 1400;
11 COMMIT;
12 SHOW PROFILES;
13
```

- Added 1 SELECT Operation on Reservation Table:

Reservation_id	Reservation_fees	Park_name	Arrival_date	Departure_date	Party_size
211	$150	Dark Sky Preserve	12-1-22	21-2-22	21
311	$250	Boat Launch	3-3-22	17-3-22	30
411	$300	Mi'kmaw heritage	23-3-22	31-3-22	50
511	$200	Otentik	14-4-22	25-4-22	45
611	$350	Hiking Trails	30-5-22	10-6-22	35
NULL	NULL	NULL	NULL	NULL	NULL

- Added 1 UPDATE Operation on Job Applications Table:

Before UPDATE Operation, Job Application Table:

JA_id	Requirement	Deadline	JA_name
1100	Resume	12-6-22	Developer
1200	Skills	20-7-22	Software Engineer
1300	Work Experience	25-7-22	Cashier
1400	Volunteering	2-8-22	Park attendant
1500	Extra-curricular	10-8-22	Manager
NULL	NULL	NULL	NULL

- After UPDATE Operation, Job Application Table:

In the above table Job Applications, the column "requirement" having "Volunterring is changed to Education Background" on update where JA_id = 1400.

JA_id	Requirement	Deadline	JA_name
1100	Resume	12-6-22	Developer
1200	Skills	20-7-22	Software Engineer
1300	Work Experience	25-7-22	Cashier
1400	Education Background	2-8-22	Park attendant
1500	Extra-curricular	10-8-22	Manager
NULL	NULL	NULL	NULL

Execution Time with updated transaction:

Query_ID	Duration	Query
110	0.00030525	SELECT * FROM `a2_loc<b00899642>`.`job a...
111	0.00253725	SELECT * FROM `a2_loc<b00899642>`.`job a...
112	0.00363650	use `a2_loc<b00899642>`
113	0.00032700	SELECT DATABASE()
114	0.00631850	SET AUTOCOMMIT = 0
115	0.00022550	SET profiling = 1
116	0.00016950	SHOW WARNINGS
117	0.00098150	START TRANSACTION
118	0.00169300	SELECT *FROM park LIMIT 0, 1000
119	0.00264325	SELECT *FROM vendor WHERE V_id = 650 LIMI...
120	0.00236675	UPDATE vendor SET V_name = "Oliver" WHERE ...
121	0.00064150	SELECT *FROM reservations LIMIT 0, 1000
122	0.00086200	SELECT *FROM `job applications` WHERE JA_i...
123	0.00744675	UPDATE `job applications` SET Requirement = ...
124	0.00771250	COMMIT

CREATE REMOTE DATABASE:

Part – 3:

1. I created database in MySQL RDBMS (GCP): Remote database as "a2_rem<b00899642>" .
2. I made 14 tables having 5 data rows for each table in "a2_rem<b00899642>" database.

```
Query 1    newsletter    camp program    campsite    allowed items    SQL File 3* ×

    1 •    use `a2_rem<b00899642>`;
    2 •    show tables;
```

Tables_in_a2_rem<b00899642>
allowed items
camp program
campsite
equipments
event
facilities
job applications
newsletter
park
park advisories
park event history
reservations
vendor
visitor

Result 1 ×

Part – 3.a)

Write one transaction block with 1 SELECT and 1 UPDATE operation (your choice).

Query:

- 1 SELECT Operation on Park Table:

Park_id	Check_in	Check_out	Availability	City	Location
210	2:00 pm	3:00pm	Sunday	Halifax	Bedford
310	3:00pm	4:00pm	Monday	Truro	Mahone Bay
410	4:00pm	5:00pm	Tuesday	Yarmouth	Frost Park
510	5:00pm	6:00pm	Wednesday	Sydney	Breton Island
610	6:00pm	7:00pm	Thursday	Chester	Oak Island
NULL	NULL	NULL	NULL	NULL	NULL

- Vendor table before UPDATE operation:

V_id	V_name	Emailid	Address
610	Alex	alex@gmail.cam	Halifax
620	Greg	greg@gmail.com	Truro
630	Meet	meet@gmail.com	Sydney
640	Justin	justin@yahoo.com	Chester
650	Robert	robert@yahoo.com	Yarmouth
NULL	NULL	NULL	NULL

- Vendor table after UPDATE operation:
  In the column, V_name – "Robert" changed to "Oliver" having V_id = 650

V_id	V_name	Emailid	Address
610	Alex	alex@gmail.cam	Halifax
620	Greg	greg@gmail.com	Truro
630	Meet	meet@gmail.com	Sydney
640	Justin	justin@yahoo.com	Chester
650	Oliver	robert@yahoo.com	Yarmouth
NULL	NULL	NULL	NULL

Part – 3.b)

Execute the transaction and record the execution time (in milliseconds) To get time in milliseconds, you can use mysql>set profiling = 1; To show time in milliseconds, you can use mysql>show profiles;


```
1 •   use `a2_rem<b00899642>`;
2 •   SET AUTOCOMMIT = 0;
3 •   SET profiling = 1;
4 •   START TRANSACTION;
5 •   SELECT *FROM park;
6 •   SELECT *FROM vendor WHERE V_id = 650;
7 •   UPDATE vendor SET V_name = "Oliver" WHERE V_id = 650;
8 •   COMMIT;
9 •   SHOW PROFILES;
```

Execution Time:

Query_ID	Duration	Query
52	0.00038200	UPDATE `job applications` SET Requirement = …
53	0.00019650	COMMIT
54	0.00028125	use `a2_rem<b00899642>`
55	0.00017075	SELECT DATABASE()
56	0.00135425	show tables
57	0.00015400	use `a2_rem<b00899642>`
58	0.00020800	SELECT DATABASE()
59	0.00016550	SET AUTOCOMMIT = 0
60	0.00016875	SET profiling = 1
61	0.00014500	SHOW WARNINGS
62	0.00015550	START TRANSACTION
63	0.00070525	SELECT *FROM park LIMIT 0, 1000
64	0.00065975	SELECT *FROM vendor WHERE V_id = 650 LIMI…
65	0.00044950	UPDATE vendor SET V_name = "Oliver" WHERE …
66	0.00546525	COMMIT

Part - 3.c)

Added 1 more SELECT and 1 more UPDATE operation, and measured the execution time.

Query:


```
1  ●    use `a2_rem<b00899642>`;
2  ●    SET AUTOCOMMIT = 0;
3  ●    SET profiling = 1;
4  ●    START TRANSACTION;
5  ●    SELECT *FROM park;
6  ●    SELECT *FROM vendor WHERE V_id = 650;
7  ●    UPDATE vendor SET V_name = "Oliver" WHERE V_id = 650;
8  ●    SELECT *FROM reservations;
9  ●    SELECT *FROM `job applications` WHERE JA_id = 1400;
10 ●    UPDATE `job applications` SET Requirement = "Education Background" WHERE  JA_id = 1400;
11 ●    COMMIT;
12 ●    SHOW PROFILES;
13
```

Added 1 SELECT Operation on reservations table and 1 UPDATE operation on job applications table:

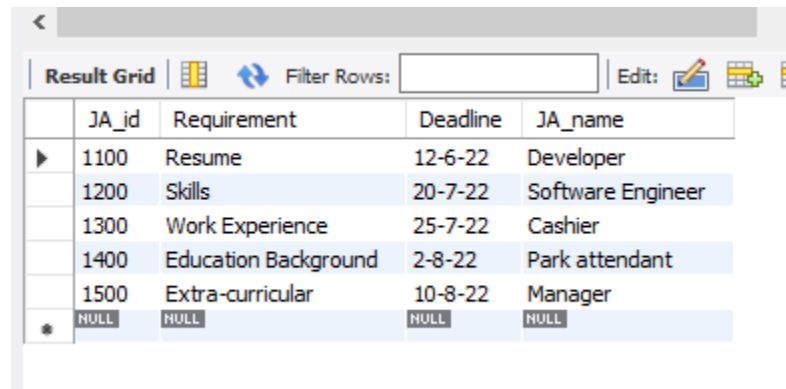- 1 SELECT operation on reservation table:

Reservation_id	Reservation_fees	Park_name	Arrival_date	Departure_date	Party_size
211	$150	Dark Sky Preserve	12-1-22	21-2-22	21
311	$250	Boat Launch	3-3-22	17-3-22	30
411	$300	Mi'kmaw heritage	23-3-22	31-3-22	50
511	$200	Otentik	14-4-22	25-4-22	45
611	$350	Hiking Trails	30-5-22	10-6-22	35
NULL	NULL	NULL	NULL	NULL	NULL

- Before 1 UPDATE Operation on Job Applications:

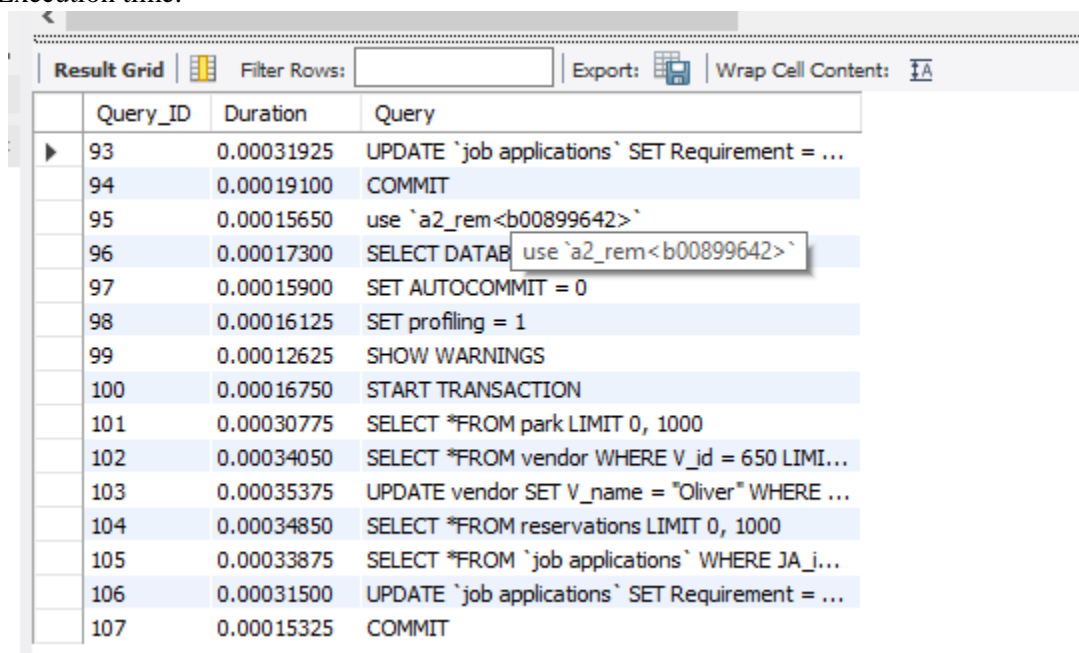| JA_id	Requirement	Deadline	JA_name
1100	Resume	12-6-22	Developer
1200	Skills	20-7-22	Software Engineer
1300	Work Experience	25-7-22	Cashier
1400	Volunteering	2-8-22	Park attendant
1500	Extra-curricular	10-8-22	Manager
NULL	NULL	NULL	NULL

- After 1 UPDATE Operation on Job Applications:

In the above table Job Applications, the column "requirement" having "Volunterring is changed to Education Background" on update where JA_id = 1400.

Execution time:

I noticed, the execution time for the remote database is faster than local because the processor could be faster. Network Latency is not that considerable.
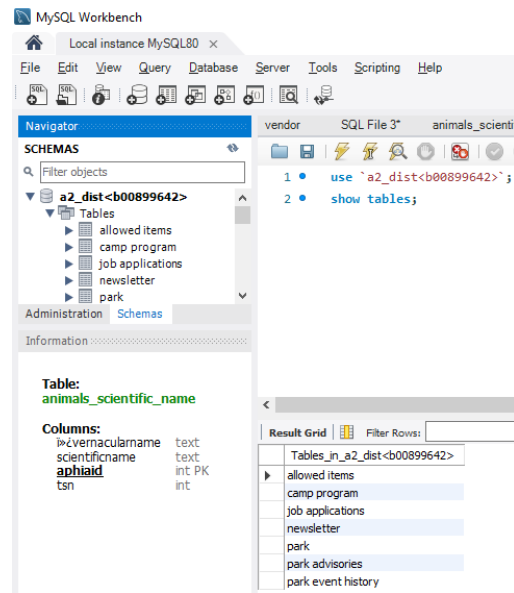
## DISTRIBUTED DATABASE:

Part – 4)

Created two databases with identical names , one in your remote MySQL RDBMS (GCP) as "a2_dist<b00899642>", and another in your local MySQL RDBMS as "a2_dist<b00899642>".

There were total 14 tables in Physical design. I distributed those tables equally. 7 tables in local RDBMS and 7 tables in remote location.

- • Tables in Local Database are as follows:

- • Tables in Remote Database are as follows:
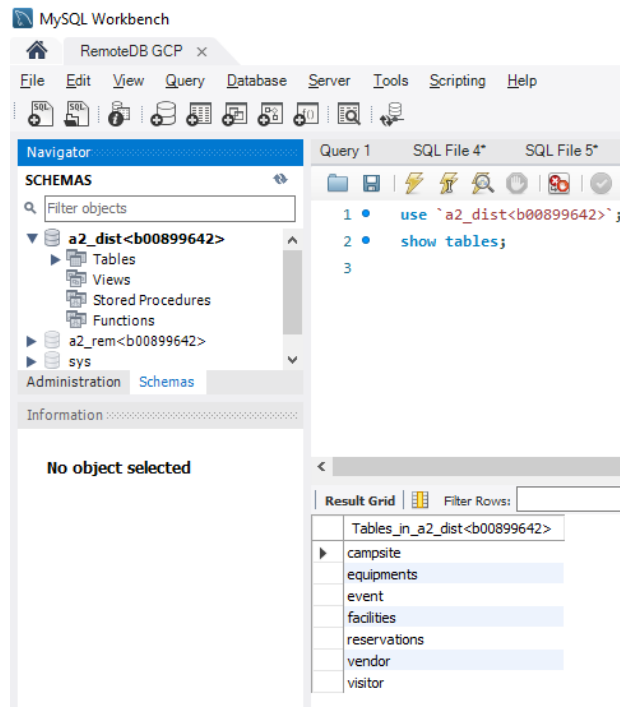
Part – 4.a):

Java Code to connect 2 databases:
I used hashmap to connect tables to their databases. I mapped each table to their respective connection i.e. local connection and remote connection.

Part – 4.b):

Java Code with 2 SELECT and 2 UPDATE operation in 1 Transaction block:
The transaction block here is the method transaction() in which all operation is performed

```java
                                                           1 usage
21     public void transaction() throws SQLException {
22         //Local Database - select operation on newsletter table
23         statement = hash_map.get("newsletter").createStatement();
24         hash_map.get("newsletter").setAutoCommit(false);
25         ResultSet result = statement.executeQuery(sql: "SELECT *FROM newsletter");
26         System.out.println("Newsletter Table:");
27         while (result.next()) {
28
29             System.out.println(result.getString(columnIndex: 1) + ", " + result.getString(columnIndex: 2)+ ", " +result.getString(columnIndex: 3));
30         }
31         hash_map.get("newsletter").commit();
32         hash_map.get("newsletter").rollback();
33         System.out.println();
34
35         //Local Database - select operation on park table
36         statement = hash_map.get("park").createStatement();
37         hash_map.get("park").setAutoCommit(false);
38         PreparedStatement stmt = hash_map.get("park").prepareStatement(sql: "SELECT *FROM park WHERE Park_id =?");
39         stmt.setInt(parameterIndex: 1, x: 610);
40         ResultSet result1 = statement.executeQuery(sql: "SELECT *FROM park");
41         System.out.println("Updated Park Table:");
42         while (result1.next()) {
43
44             System.out.println(result1.getInt(columnIndex: 1) + ", " + result1.getString(columnIndex: 2) + ", " + result1.getString(columnIndex: 3) + ", " + result1.getS
45         }
```

```java
44             System.out.println(result1.getInt(columnIndex: 1) + ", " + result1.getString(columnIndex: 2) + ", " + result1.getString(columnIndex: 3) + ", "
45         }
46         //Local Database - update operation on park table
47         statement = hash_map.get("park").createStatement();
48         PreparedStatement stmt1 = hash_map.get("park").prepareStatement(sql: "UPDATE park SET City=? WHERE Park_id =?");
49         stmt1.setString(parameterIndex: 1, x: "Mumbai");
50         stmt1.setInt(parameterIndex: 2, x: 610);
51         int row = stmt1.executeUpdate();
52         System.out.println("number of rows updated" + row);
53         System.out.println();
54         hash_map.get("park").commit();
55         hash_map.get("park").rollback();
56
57         //Remote Database - select operation on equipments table
58         statement = hash_map.get("equipments").createStatement();
59         hash_map.get("equipments").setAutoCommit(false);
60         ResultSet result2 = statement.executeQuery(sql: "SELECT *FROM equipments");
61         System.out.println("Equipments Table:");
62         while (result2.next()) {
63             System.out.println(result2.getInt(columnIndex: 1) + ", " + result2.getString(columnIndex: 2) + ", " + result2.getString(columnIndex: 3) + ", " + result2.get
64         }
65         System.out.println();
66         hash_map.get("equipments").commit();
67         hash_map.get("equipments").rollback();
68
```

```java
63             System.out.println(result2.getInt(columnIndex: 1) + ", " + result2.getString(columnIndex: 2) + ", " + result2.getString(columnIndex: 3) + ", "
64         }
65         System.out.println();
66         hash_map.get("equipments").commit();
67         hash_map.get("equipments").rollback();
68
69         //Remote Database - select operation on vendor table
70         statement = hash_map.get("vendor").createStatement();
71         hash_map.get("vendor").setAutoCommit(false);
72         PreparedStatement stmt2 = hash_map.get("vendor").prepareStatement(sql: "SELECT *FROM vendor WHERE V_id =?");
73         stmt2.setInt(parameterIndex: 1, x: 650);
74         ResultSet result3 = statement.executeQuery(sql: "SELECT *FROM vendor");
75         System.out.println("Updated Vendor Table:");
76         while (result3.next()) {
77             System.out.println(result3.getInt(columnIndex: 1) + ", " + result3.getString(columnIndex: 2) + ", " + result3.getString(columnIndex: 3) + ", " + result3.getS
78
79         }
80         //Remote Database - update operation on vendor table
81         statement = hash_map.get("vendor").createStatement();
82         PreparedStatement stmt3 = hash_map.get("vendor").prepareStatement(sql: "UPDATE vendor SET V_name=? WHERE V_id =?");
83         stmt3.setString(parameterIndex: 1, x: "Oliver");
84         stmt3.setInt(parameterIndex: 2, x: 650);
85         int row1 = stmt3.executeUpdate();
86         System.out.println("number of rows updated" + row1);
87         hash_map.get("vendor").commit();
88         hash_map.get("vendor").rollback();
89     }
```

Part – 4.c):

- **1 SELECT 1 UPDATE in local site:**
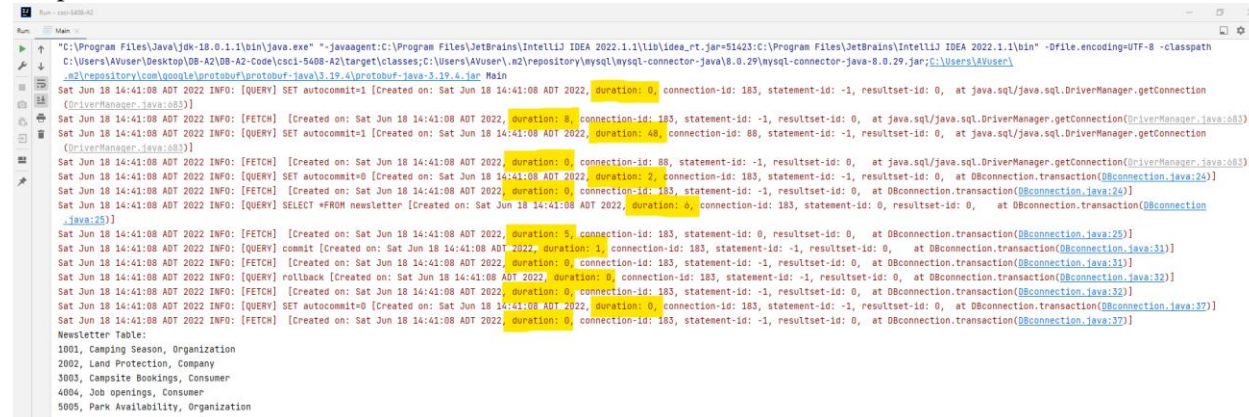 1 SELECT operation performed on "newsletter" table.

Code:

```java
//Local Database - select operation on newsletter table
statement = hash_map.get("newsletter").createStatement();
hash_map.get("newsletter").setAutoCommit(false);
ResultSet result = statement.executeQuery( sql: "SELECT *FROM newsletter");
System.out.println("Newsletter Table:");
while (result.next()) {

    System.out.println(result.getString( columnIndex: 1) + ", " + result.getString( columnIndex: 2)+ ", " +result.getString( columnIndex: 3));
}
hash_map.get("newsletter").commit();
hash_map.get("newsletter").rollback();
System.out.println();
```

Output:

1 UPDATE operation performed on "Park" table.
Code:

```
//Local Database - select operation on park table
statement = hash_map.get("park").createStatement();
hash_map.get("park").setAutoCommit(false);
PreparedStatement stmt = hash_map.get("park").prepareStatement( sql: "SELECT *FROM park WHERE Park_id =?");
stmt.setInt( parameterIndex: 1, x: 610);
ResultSet result1 = statement.executeQuery( sql: "SELECT *FROM park");
System.out.println("Updated Park Table:");
while (result1.next()) {

    System.out.println(result1.getInt( columnIndex: 1) + ", " + result1.getString( columnIndex: 2) + ", " + result1.getString( columnIndex: 3) + ", " + result1.getS
}
//Local Database - update operation on park table
statement = hash_map.get("park").createStatement();
PreparedStatement stmt1 = hash_map.get("park").prepareStatement( sql: "UPDATE park SET City=? WHERE Park_id =?");
stmt1.setString( parameterIndex: 1, x: "Mumbai");
stmt1.setInt( parameterIndex: 2, x: 610);
int row = stmt1.executeUpdate();
System.out.println("number of rows updated" + row);
System.out.println();
hash_map.get("park").commit();
hash_map.get("park").rollback();
```

Output:

```
Updated Park Table:
210, 2:00 pm, 3:00pm, Sunday, Halifax, Bedford
310, 3:00pm, 4:00pm, Monday, Truro, Mahone Bay
410, 4:00pm, 5:00pm, Tuesday, Yarmouth, Frost Park
510, 5:00pm, 6:00pm, Wednesday, Sydney, Breton Island
610, 6:00pm, 7:00pm, Thursday, Mumbai, Oak Island
Sat Jun 18 15:05:26 ADT 2022 INFO: [QUERY] SELECT *FROM park [Created on: Sat Jun 18 15:05:26 ADT 2022, duration: 2, connection-id: 185, statement-id: 0, resultset-id: 0, at DBconnection.transaction(DBconnection.java:40)]
Sat Jun 18 15:05:26 ADT 2022 INFO: [FETCH]  [Created on: Sat Jun 18 15:05:26 ADT 2022, duration: 0, connection-id: 185, statement-id: 0, resultset-id: 0,   at DBconnection.transaction(DBconnection.java:40)]
Sat Jun 18 15:05:26 ADT 2022 INFO: [QUERY] UPDATE park SET City='Mumbai' WHERE Park_id =610 [Created on: Sat Jun 18 15:05:26 ADT 2022, duration: 2, connection-id: 185, statement-id: 0, resultset-id: 0,   at DBconnection
.transaction(DBconnection.java:51)]
Sat Jun 18 15:05:26 ADT 2022 INFO: [FETCH]  [Created on: Sat Jun 18 15:05:26 ADT 2022, duration: 0, connection-id: 185, statement-id: 0, resultset-id: 0,   at DBconnection.transaction(DBconnection.java:51)]
Sat Jun 18 15:05:26 ADT 2022 INFO: [QUERY] commit [Created on: Sat Jun 18 15:05:26 ADT 2022, duration: 1, connection-id: 185, statement-id: -1, resultset-id: 0,   at DBconnection.transaction(DBconnection.java:54)]
Sat Jun 18 15:05:26 ADT 2022 INFO: [FETCH]  [Created on: Sat Jun 18 15:05:26 ADT 2022, duration: 0, connection-id: 185, statement-id: -1, resultset-id: 0,  at DBconnection.transaction(DBconnection.java:54)]
Sat Jun 18 15:05:26 ADT 2022 INFO: [QUERY] rollback [Created on: Sat Jun 18 15:05:26 ADT 2022, duration: 1, connection-id: 185, statement-id: -1, resultset-id: 0,  at DBconnection.transaction(DBconnection.java:55)]
Sat Jun 18 15:05:26 ADT 2022 INFO: [FETCH]  [Created on: Sat Jun 18 15:05:26 ADT 2022, duration: 0, connection-id: 185, statement-id: -1, resultset-id: 0,  at DBconnection.transaction(DBconnection.java:55)]
number of rows updated1

Sat Jun 18 15:05:26 ADT 2022 INFO: [QUERY] SET autocommit=0 [Created on: Sat Jun 18 15:05:26 ADT 2022, duration: 47, connection-id: 410, statement-id: -1, resultset-id: 0, at DBconnection.transaction(DBconnection.java:59)]
Sat Jun 18 15:05:26 ADT 2022 INFO: [FETCH]  [Created on: Sat Jun 18 15:05:26 ADT 2022, duration: 0, connection-id: 410, statement-id: -1, resultset-id: 0,  at DBconnection.transaction(DBconnection.java:59)]
```

Before running code, before update:

Park_id	Check_in	Check_out	Availability	City	Location
210	2:00 pm	3:00pm	Sunday	Halifax	Bedford
310	3:00pm	4:00pm	Monday	Truro	Mahone Bay
410	4:00pm	5:00pm	Tuesday	Yarmouth	Frost Park
510	5:00pm	6:00pm	Wednesday	Sydney	Breton Island
610	6:00pm	7:00pm	Thursday	Dartmouth	Oak Island
NULL	NULL	NULL	NULL	NULL	NULL

After running code:

Update reflected in MySQL local database:
In City column, the "Dartmouth" is updated to "Mumbai" where Park_id = 610.

- 1 SELECT 1 UPDATE in remote site:
  1 SELECT operation performed on "Equipments" table.

Code:

```java
//Remote Database - select operation on equipments table
statement = hash_map.get("equipments").createStatement();
hash_map.get("equipments").setAutoCommit(false);
ResultSet result2 = statement.executeQuery( sql: "SELECT *FROM equipments");
System.out.println("Equipments Table:");
while (result2.next()) {
    System.out.println(result2.getInt( columnIndex: 1) + ", " + result2.getString( columnIndex: 2) + ", " + result2.getString( columnIndex: 3) + ", " + result2.get
}
System.out.println();
hash_map.get("equipments").commit();
hash_map.get("equipments").rollback();
```

Output:

1 UPDATE operation performed on "Vendor" table.

Code:

```java
//Remote Database - select operation on vendor table
statement = hash_map.get("vendor").createStatement();
hash_map.get("vendor").setAutoCommit(false);
PreparedStatement stmt2 = hash_map.get("vendor").prepareStatement( sql: "SELECT *FROM vendor WHERE V_id =?");
stmt2.setInt( parameterIndex: 1,  x: 650);
ResultSet result3 = statement.executeQuery( sql: "SELECT *FROM vendor");
System.out.println("Updated Vendor Table:");
while (result3.next()) {
    System.out.println(result3.getInt( columnIndex: 1) + ", " + result3.getString( columnIndex: 2) + ", " + result3.getString( columnIndex: 3) + ", " + result3.getS

}
//Remote Database - update operation on vendor table
statement = hash_map.get("vendor").createStatement();
PreparedStatement stmt3 = hash_map.get("vendor").prepareStatement( sql: "UPDATE vendor SET V_name=? WHERE V_id =?");
stmt3.setString( parameterIndex: 1,  x: "Oliver");
stmt3.setInt( parameterIndex: 2,  x: 650);
int row1 = stmt3.executeUpdate();
System.out.println("number of rows updated" + row1);
hash_map.get("vendor").commit();
hash_map.get("vendor").rollback();
```

Output:

```
Sat Jun 18 15:05:27 ADT 2022 INFO: [QUERY] SELECT *FROM vendor [Created on: Sat Jun 18 15:05:27 ADT 2022, duration: 50, connection-id: 410, statement-id: 0, resultset-id: 0,   at DBconnection.transaction(DBconnection.java:74)]
Sat Jun 18 15:05:27 ADT 2022 INFO: [FETCH]  [Created on: Sat Jun 18 15:05:27 ADT 2022, duration: 0, connection-id: 410, statement-id: 0, resultset-id: 0,   at DBconnection.transaction(DBconnection.java:74)]
Updated Vendor Table:
610, Alex, alex@gmail.cam, Halifax
620, Greg, greg@gmail.com, Truro
630, Meet, meet@gmail.com, Sydney
640, Justin, justin@yahoo.com, Chester
650, Oliver, robert@yahoo.com, Yarmouth
Sat Jun 18 15:05:27 ADT 2022 INFO: [QUERY] UPDATE vendor SET V_name='Oliver' WHERE V_id =650 [Created on: Sat Jun 18 15:05:27 ADT 2022, duration: 49, connection-id: 410, statement-id: 0, resultset-id: 0, at DBconnection
.transaction(DBconnection.java:85)]
Sat Jun 18 15:05:27 ADT 2022 INFO: [FETCH]  [Created on: Sat Jun 18 15:05:27 ADT 2022, duration: 0, connection-id: 410, statement-id: 0, resultset-id: 0,   at DBconnection.transaction(DBconnection.java:85)]
number of rows updated1
Sat Jun 18 15:05:27 ADT 2022 INFO: [QUERY] commit [Created on: Sat Jun 18 15:05:27 ADT 2022, duration: 49, connection-id: 410, statement-id: -1, resultset-id: 0,   at DBconnection.transaction(DBconnection.java:87)]
Sat Jun 18 15:05:27 ADT 2022 INFO: [FETCH]  [Created on: Sat Jun 18 15:05:27 ADT 2022, duration: 0, connection-id: 410, statement-id: -1, resultset-id: 0,   at DBconnection.transaction(DBconnection.java:87)]
Sat Jun 18 15:05:27 ADT 2022 INFO: [QUERY] rollback [Created on: Sat Jun 18 15:05:27 ADT 2022, duration: 55, connection-id: 410, statement-id: -1, resultset-id: 0, at DBconnection.transaction(DBconnection.java:88)]
Sat Jun 18 15:05:27 ADT 2022 INFO: [FETCH]  [Created on: Sat Jun 18 15:05:27 ADT 2022, duration: 0, connection-id: 410, statement-id: -1, resultset-id: 0,   at DBconnection.transaction(DBconnection.java:88)]

Process finished with exit code 0
```

Before running code, before update:
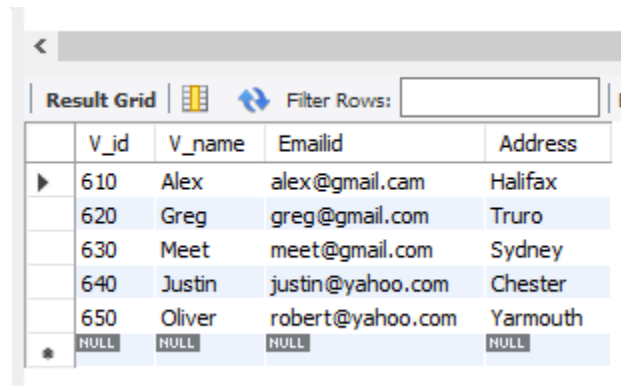
V_id	V_name	Emailid	Address
610	Alex	alex@gmail.cam	Halifax
620	Greg	greg@gmail.com	Truro
630	Meet	meet@gmail.com	Sydney
640	Justin	justin@yahoo.com	Chester
650	Robert	robert@yahoo.com	Yarmouth
NULL	NULL	NULL	NULL

Update reflected in Remote database:
In the column, V_name – "Robert" changed to "Oliver" having V_id = 650

Part – 4.d): To get time in milliseconds, I used profileSQL to trace queries and their execution/fetch times.

Part – 4.e)
Output along with the time of execution:

```
"C:\Program Files\Java\jdk-18.0.1.1\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2022.1.1\lib\idea_rt.jar=51423:C:\Program Files\JetBrains\IntelliJ IDEA 2022.1.1\bin" -Dfile.encoding=UTF-8 -classpath
C:\Users\AVuser\Desktop\DB-A2\DB-A2-Code\csci-5408-A2\target\classes;C:\Users\AVuser\.m2\repository\mysql\mysql-connector-java\8.0.29\mysql-connector-java-8.0.29.jar;C:\Users\AVuser\
.m2\repository\com\google\protobuf\protobuf-java\3.19.4\protobuf-java-3.19.4.jar Main
Sat Jun 18 14:41:08 ADT 2022 INFO: [QUERY] SET autocommit=1 [Created on: Sat Jun 18 14:41:08 ADT 2022, duration: 0, connection-id: 183, statement-id: -1, resultset-id: 0,  at java.sql/java.sql.DriverManager.getConnection
(DriverManager.java:683)]
Sat Jun 18 14:41:08 ADT 2022 INFO: [FETCH]  [Created on: Sat Jun 18 14:41:08 ADT 2022, duration: 8, connection-id: 183, statement-id: -1, resultset-id: 0,  at java.sql/java.sql.DriverManager.getConnection(DriverManager.java:683)]
Sat Jun 18 14:41:08 ADT 2022 INFO: [QUERY] SET autocommit=1 [Created on: Sat Jun 18 14:41:08 ADT 2022, duration: 48, connection-id: 88, statement-id: -1, resultset-id: 0,  at java.sql/java.sql.DriverManager.getConnection
(DriverManager.java:683)]
Sat Jun 18 14:41:08 ADT 2022 INFO: [FETCH]  [Created on: Sat Jun 18 14:41:08 ADT 2022, duration: 0, connection-id: 88, statement-id: -1, resultset-id: 0,  at java.sql/java.sql.DriverManager.getConnection(DriverManager.java:683)]
Sat Jun 18 14:41:08 ADT 2022 INFO: [QUERY] SET autocommit=0 [Created on: Sat Jun 18 14:41:08 ADT 2022, duration: 2, connection-id: 183, statement-id: -1, resultset-id: 0,  at DBconnection.transaction(DBconnection.java:24)]
Sat Jun 18 14:41:08 ADT 2022 INFO: [FETCH]  [Created on: Sat Jun 18 14:41:08 ADT 2022, duration: 0, connection-id: 183, statement-id: -1, resultset-id: 0,  at DBconnection.transaction(DBconnection.java:24)]
Sat Jun 18 14:41:08 ADT 2022 INFO: [QUERY] SELECT *FROM newsletter [Created on: Sat Jun 18 14:41:08 ADT 2022, duration: 0, connection-id: 183, statement-id: 0, resultset-id: 0,  at DBconnection.transaction(DBconnection
.java:25)]
Sat Jun 18 14:41:08 ADT 2022 INFO: [FETCH]  [Created on: Sat Jun 18 14:41:08 ADT 2022, duration: 5, connection-id: 183, statement-id: 0, resultset-id: 0,  at DBconnection.transaction(DBconnection.java:25)]
Sat Jun 18 14:41:08 ADT 2022 INFO: [QUERY] commit [Created on: Sat Jun 18 14:41:08 ADT 2022, duration: 1, connection-id: 183, statement-id: -1, resultset-id: 0,  at DBconnection.transaction(DBconnection.java:31)]
Sat Jun 18 14:41:08 ADT 2022 INFO: [FETCH]  [Created on: Sat Jun 18 14:41:08 ADT 2022, duration: 0, connection-id: 183, statement-id: -1, resultset-id: 0,  at DBconnection.transaction(DBconnection.java:31)]
Sat Jun 18 14:41:08 ADT 2022 INFO: [QUERY] rollback [Created on: Sat Jun 18 14:41:08 ADT 2022, duration: 0, connection-id: 183, statement-id: -1, resultset-id: 0,  at DBconnection.transaction(DBconnection.java:32)]
Sat Jun 18 14:41:08 ADT 2022 INFO: [FETCH]  [Created on: Sat Jun 18 14:41:08 ADT 2022, duration: 0, connection-id: 183, statement-id: -1, resultset-id: 0,  at DBconnection.transaction(DBconnection.java:32)]
Sat Jun 18 14:41:08 ADT 2022 INFO: [QUERY] SET autocommit=0 [Created on: Sat Jun 18 14:41:08 ADT 2022, duration: 0, connection-id: 183, statement-id: -1, resultset-id: 0,  at DBconnection.transaction(DBconnection.java:37)]
Sat Jun 18 14:41:08 ADT 2022 INFO: [FETCH]  [Created on: Sat Jun 18 14:41:08 ADT 2022, duration: 0, connection-id: 183, statement-id: -1, resultset-id: 0,  at DBconnection.transaction(DBconnection.java:37)]
Newsletter Table:
1001, Camping Season, Organization
2002, Land Protection, Company
3003, Campsite Bookings, Consumer
4004, Job openings, Consumer
5005, Park Availability, Organization

Sat Jun 18 14:41:08 ADT 2022 INFO: [QUERY] SELECT *FROM park [Created on: Sat Jun 18 14:41:08 ADT 2022, duration: 2, connection-id: 183, statement-id: 0, resultset-id: 0,  at DBconnection.transaction(DBconnection.java:40)]
Sat Jun 18 14:41:08 ADT 2022 INFO: [FETCH]  [Created on: Sat Jun 18 14:41:08 ADT 2022, duration: 0, connection-id: 183, statement-id: 0, resultset-id: 0,  at DBconnection.transaction(DBconnection.java:40)]

Updated Park Table:
210, 2:00 pm, 3:00pm, Sunday, Halifax, Bedford
310, 3:00pm, 4:00pm, Monday, Truro, Mahone Bay
410, 4:00pm, 5:00pm, Tuesday, Yarmouth, Frost Park
510, 5:00pm, 6:00pm, Wednesday, Sydney, Breton Island
610, 6:00pm, 7:00pm, Thursday, Mumbai, Oak Island
number of rows updated1
```

```
Sat Jun 18 14:41:08 ADT 2022 INFO: [QUERY] UPDATE park SET City='Mumbai' WHERE Park_id =610 [Created on: Sat Jun 18 14:41:08 ADT 2022, duration: 2, connection-id: 183, statement-id: 0, resultset-id: 0,  at DBconnection
.transaction(DBconnection.java:51)]
Sat Jun 18 14:41:08 ADT 2022 INFO: [FETCH]  [Created on: Sat Jun 18 14:41:08 ADT 2022, duration: 0, connection-id: 183, statement-id: 0, resultset-id: 0,  at DBconnection.transaction(DBconnection.java:51)]
Sat Jun 18 14:41:08 ADT 2022 INFO: [QUERY] commit [Created on: Sat Jun 18 14:41:08 ADT 2022, duration: 1, connection-id: 183, statement-id: -1, resultset-id: 0,  at DBconnection.transaction(DBconnection.java:54)]
Sat Jun 18 14:41:08 ADT 2022 INFO: [FETCH]  [Created on: Sat Jun 18 14:41:08 ADT 2022, duration: 0, connection-id: 183, statement-id: -1, resultset-id: 0,  at DBconnection.transaction(DBconnection.java:54)]
Sat Jun 18 14:41:08 ADT 2022 INFO: [QUERY] rollback [Created on: Sat Jun 18 14:41:08 ADT 2022, duration: 1, connection-id: 183, statement-id: -1, resultset-id: 0,  at DBconnection.transaction(DBconnection.java:55)]
Sat Jun 18 14:41:08 ADT 2022 INFO: [FETCH]  [Created on: Sat Jun 18 14:41:08 ADT 2022, duration: 0, connection-id: 183, statement-id: -1, resultset-id: 0,  at DBconnection.transaction(DBconnection.java:55)]
Sat Jun 18 14:41:08 ADT 2022 INFO: [QUERY] SET autocommit=0 [Created on: Sat Jun 18 14:41:08 ADT 2022, duration: 49, connection-id: 88, statement-id: -1, resultset-id: 0,  at DBconnection.transaction(DBconnection.java:59)]
Sat Jun 18 14:41:08 ADT 2022 INFO: [FETCH]  [Created on: Sat Jun 18 14:41:08 ADT 2022, duration: 0, connection-id: 88, statement-id: -1, resultset-id: 0,  at DBconnection.transaction(DBconnection.java:59)]
Sat Jun 18 14:41:08 ADT 2022 INFO: [QUERY] SELECT *FROM equipments [Created on: Sat Jun 18 14:41:08 ADT 2022, duration: 49, connection-id: 88, statement-id: 0, resultset-id: 0,  at DBconnection.transaction(DBconnection
.java:60)]
Sat Jun 18 14:41:08 ADT 2022 INFO: [FETCH]  [Created on: Sat Jun 18 14:41:08 ADT 2022, duration: 0, connection-id: 88, statement-id: 0, resultset-id: 0,  at DBconnection.transaction(DBconnection.java:60)]
Equipments Table:
12,  Sleeping bags, mandatory, 2
13,  Mess kits , optional, 3
14, Medications , mandatory, 5
15, water shoes, mandatory, 7
16, Tent, optional, 1

Sat Jun 18 14:41:08 ADT 2022 INFO: [QUERY] commit [Created on: Sat Jun 18 14:41:08 ADT 2022, duration: 48, connection-id: 88, statement-id: -1, resultset-id: 0,  at DBconnection.transaction(DBconnection.java:66)]
Sat Jun 18 14:41:08 ADT 2022 INFO: [FETCH]  [Created on: Sat Jun 18 14:41:08 ADT 2022, duration: 0, connection-id: 88, statement-id: -1, resultset-id: 0,  at DBconnection.transaction(DBconnection.java:66)]
Sat Jun 18 14:41:08 ADT 2022 INFO: [QUERY] rollback [Created on: Sat Jun 18 14:41:08 ADT 2022, duration: 52, connection-id: 88, statement-id: -1, resultset-id: 0,  at DBconnection.transaction(DBconnection.java:67)]
Sat Jun 18 14:41:08 ADT 2022 INFO: [FETCH]  [Created on: Sat Jun 18 14:41:08 ADT 2022, duration: 0, connection-id: 88, statement-id: -1, resultset-id: 0,  at DBconnection.transaction(DBconnection.java:67)]
Sat Jun 18 14:41:08 ADT 2022 INFO: [QUERY] SET autocommit=0 [Created on: Sat Jun 18 14:41:08 ADT 2022, duration: 46, connection-id: 88, statement-id: -1, resultset-id: 0,  at DBconnection.transaction(DBconnection.java:71)]
Sat Jun 18 14:41:08 ADT 2022 INFO: [FETCH]  [Created on: Sat Jun 18 14:41:08 ADT 2022, duration: 0, connection-id: 88, statement-id: -1, resultset-id: 0,  at DBconnection.transaction(DBconnection.java:71)]
Sat Jun 18 14:41:08 ADT 2022 INFO: [QUERY] SELECT *FROM vendor [Created on: Sat Jun 18 14:41:08 ADT 2022, duration: 48, connection-id: 88, statement-id: 0, resultset-id: 0,  at DBconnection.transaction(DBconnection.java:74)]
Sat Jun 18 14:41:08 ADT 2022 INFO: [FETCH]  [Created on: Sat Jun 18 14:41:08 ADT 2022, duration: 0, connection-id: 88, statement-id: 0, resultset-id: 0,  at DBconnection.transaction(DBconnection.java:74)]
Updated Vendor Table:
610, Alex, alex@gmail.cam, Halifax
620, Greg, greg@gmail.com, Truro
630, Meet, meet@gmail.com, Sydney
640, Justin, justin@yahoo.com, Chester
650, Oliver, robert@yahoo.com, Yarmouth
Sat Jun 18 14:41:09 ADT 2022 INFO: [QUERY] UPDATE vendor SET V_name='Oliver' WHERE V_id =650 [Created on: Sat Jun 18 14:41:09 ADT 2022, duration: 45, connection-id: 88, statement-id: 0, resultset-id: 0,  at DBconnection
.transaction(DBconnection.java:85)]
Sat Jun 18 14:41:09 ADT 2022 INFO: [FETCH]  [Created on: Sat Jun 18 14:41:09 ADT 2022, duration: 0, connection-id: 88, statement-id: 0, resultset-id: 0,  at DBconnection.transaction(DBconnection.java:85)]
number of rows updated1
Sat Jun 18 14:41:09 ADT 2022 INFO: [QUERY] commit [Created on: Sat Jun 18 14:41:09 ADT 2022, duration: 51, connection-id: 88, statement-id: -1, resultset-id: 0,  at DBconnection.transaction(DBconnection.java:87)]
Sat Jun 18 14:41:09 ADT 2022 INFO: [FETCH]  [Created on: Sat Jun 18 14:41:09 ADT 2022, duration: 0, connection-id: 88, statement-id: -1, resultset-id: 0,  at DBconnection.transaction(DBconnection.java:87)]
Sat Jun 18 14:41:09 ADT 2022 INFO: [QUERY] rollback [Created on: Sat Jun 18 14:41:09 ADT 2022, duration: 43, connection-id: 88, statement-id: -1, resultset-id: 0,  at DBconnection.transaction(DBconnection.java:88)]
Sat Jun 18 14:41:09 ADT 2022 INFO: [FETCH]  [Created on: Sat Jun 18 14:41:09 ADT 2022, duration: 0, connection-id: 88, statement-id: -1, resultset-id: 0,  at DBconnection.transaction(DBconnection.java:88)]

Process finished with exit code 0
```

References:

1. https://dev.mysql.com/doc/connector-j/8.0/en/connector-j-connp-props-debugging-profiling.html#cj-conn-prop_profileSQL
2. https://docs.oracle.com/javase/tutorial/jdbc/basics/connecting.html#:~:text=A%20database%20connection%20URL%20is,is%20specified%20by%20your%20DBMS.
3. https://dal.brightspace.com/d2l/le/content/221749/viewContent/3040910/View

Git Link:

https://git.cs.dal.ca/umatiya/csci5408_s22_faiza_umatiya_b00899642.git