CSCI 5408

Data Management, Warehousing and Analytics

Assignment 2

Problem - 1

Name : Sagarkumar Pankajbhai Vaghasia

CSID : vaghasia

Banner ID : #B00878629

GitLab Link :
https://git.cs.dal.ca/vaghasia/csci5408_s22_sagarkumar_vaghasia_b00878629

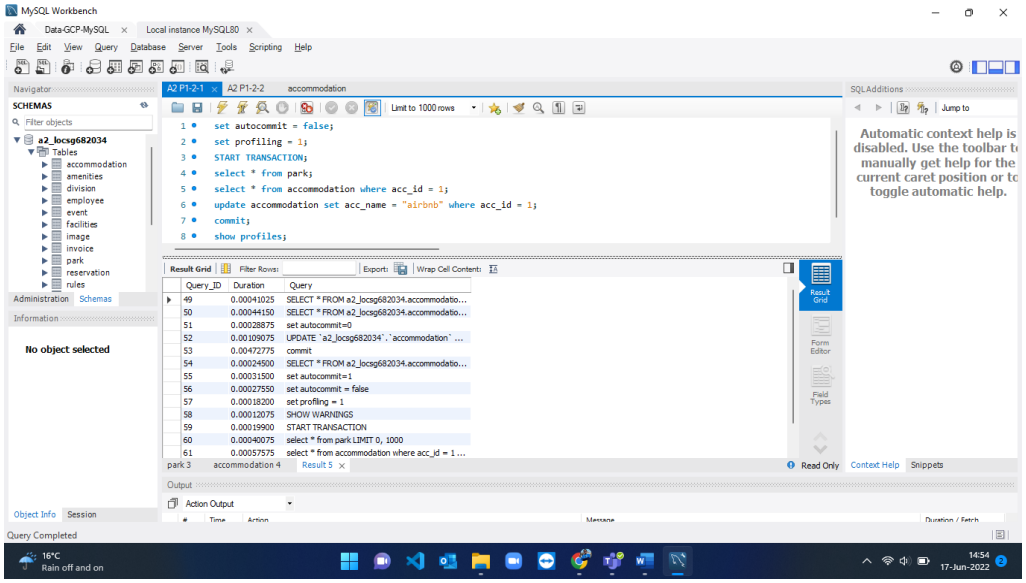
Problem #1

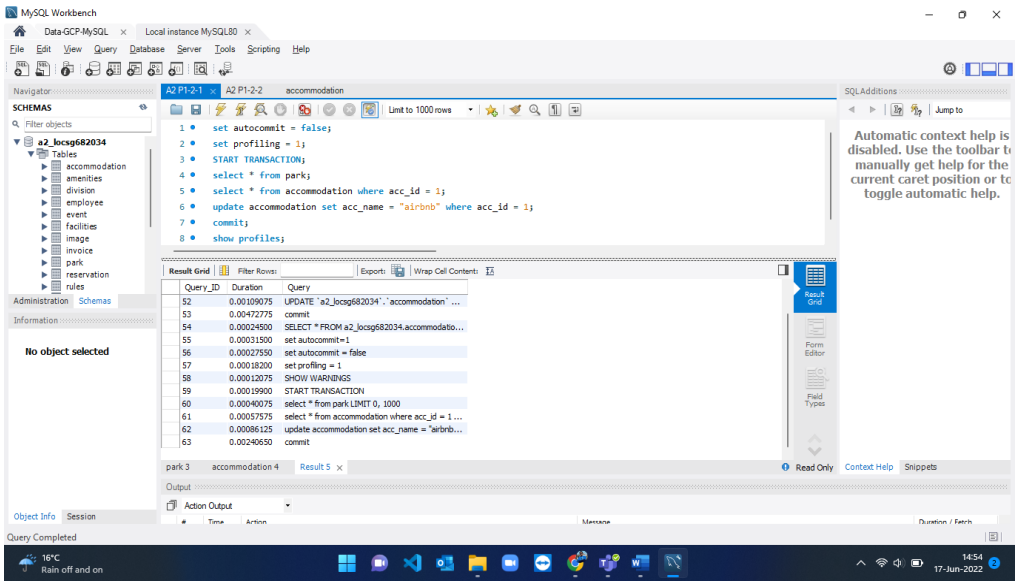
1) I have created all the tables from the previous EERD of Parks Nova Scotia. There were total 12 entities I created previously. So, I have created 12 tables according to the entities and inserted 4 to 5 rows of data in each tables. The screenshot for the list of tables is given below and the values which I inserted are submitted in dump.

2) I have created one local database and named it a2_locsg682034 where I have put the above-mentioned 12 tables and their values. Then, I wrote a transaction block having 1 SELECT and 1 UPDATE statement.

```
set autocommit = false;
set profiling = 1;
START TRANSACTION;
select * from park;
select * from accommodation where acc_id = 1;
update accommodation set acc_name = "airbnb" where acc_id = 1;
commit;
show profiles;
```

I have recorded the execution time. The transaction having 1 SELECT AND 1 UPDATE and its output along with the recorded time is given below:

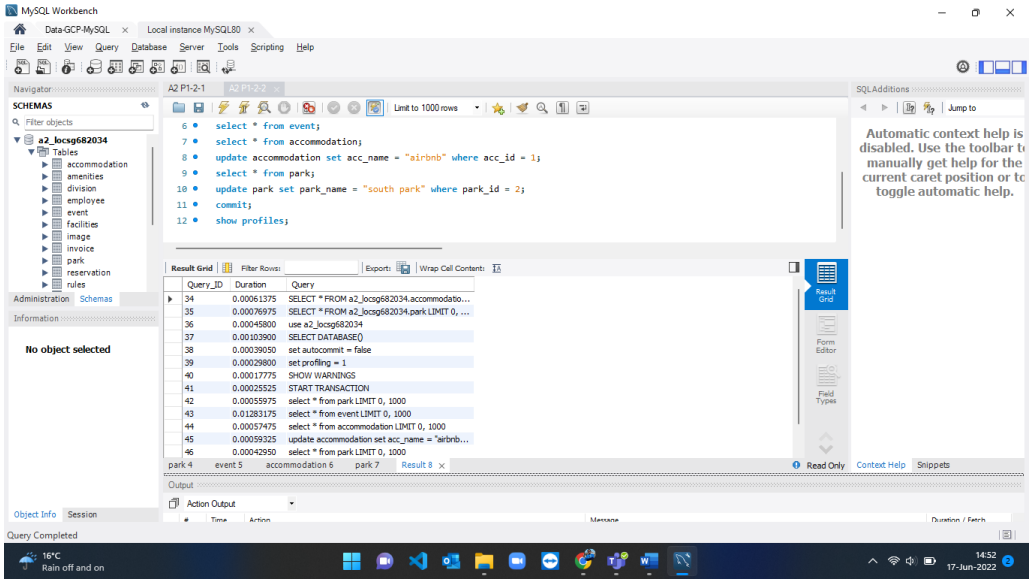


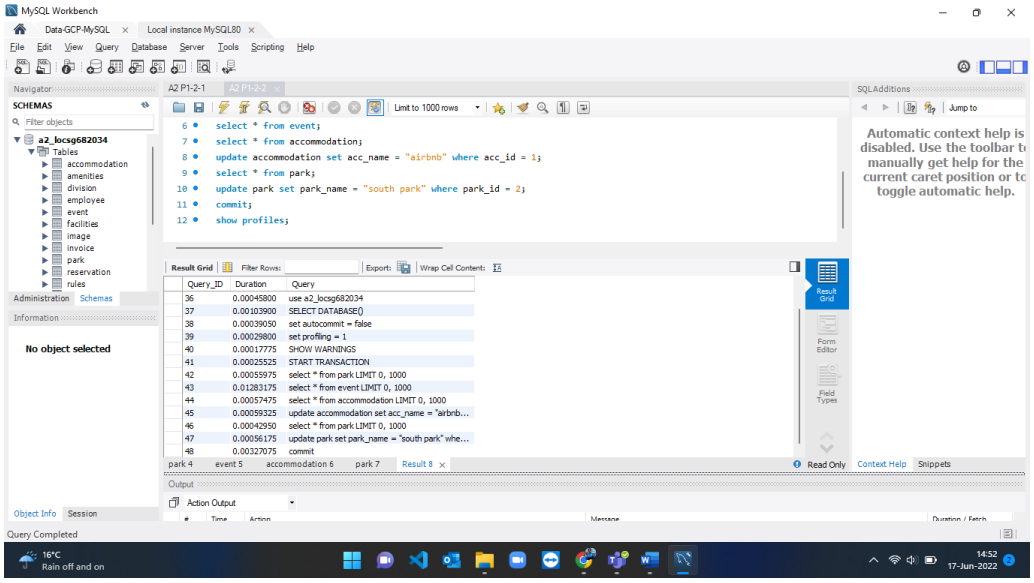


After That, I added one more SELECT and UPDATE statement in the transaction.

```
set autocommit = false;
set profiling = 1;
START TRANSACTION;
select * from park;
select * from event;
select * from accommodation;
update accommodation set acc_name = "airbnb" where acc_id = 1;
select * from park;
update park set park_name = "south park" where park_id = 2;
commit;
show profiles;
```

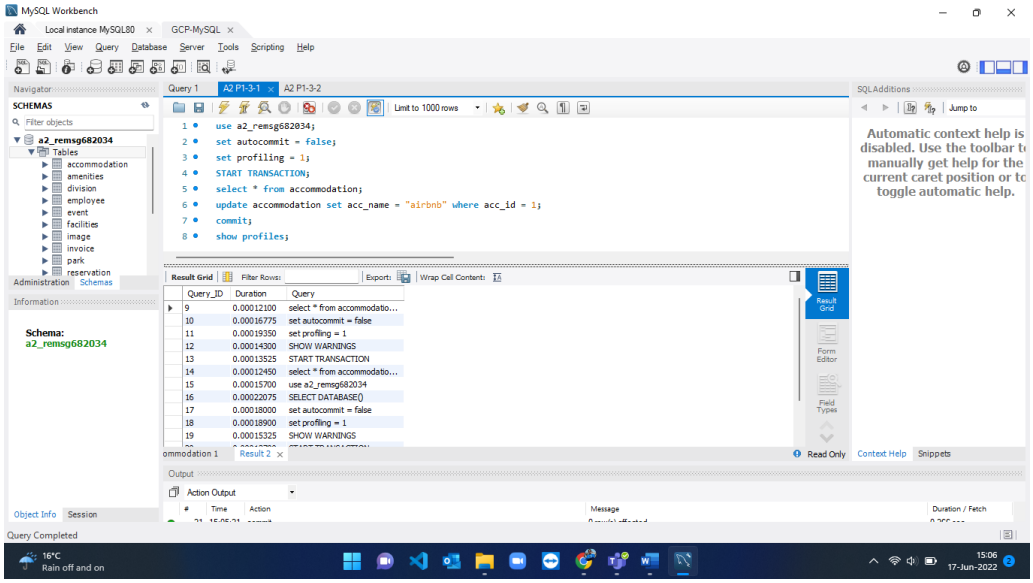
I have recorded the execution time. The transaction having 2 SELECT AND 2 UPDATE and its output along with the recorded time is given below:

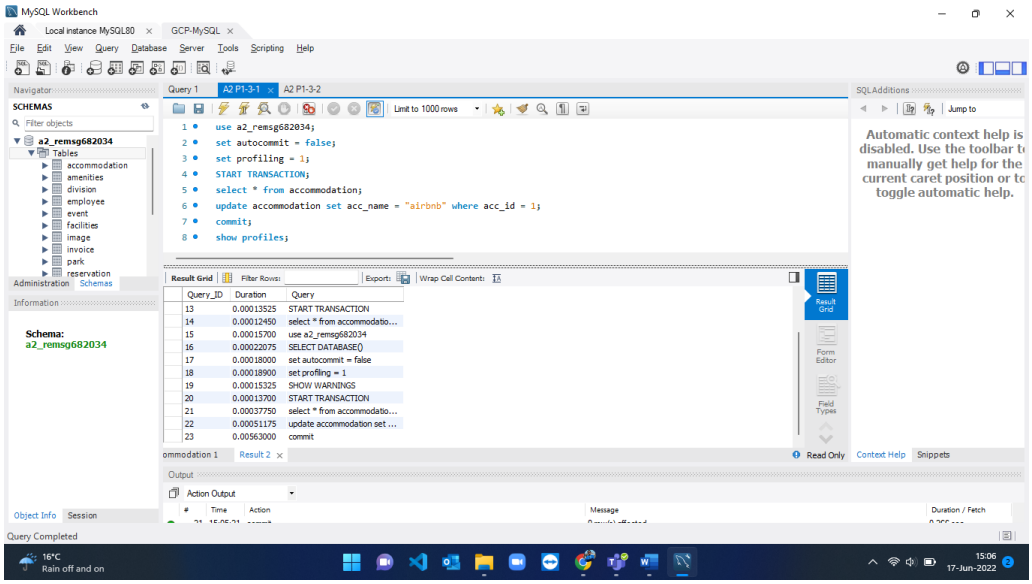




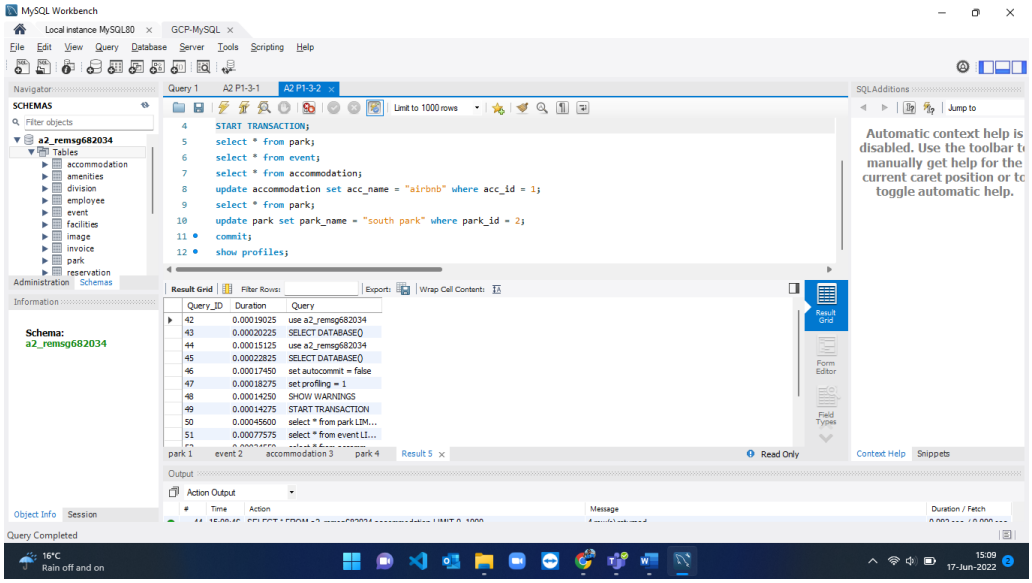
Sr No.	Transaction	Time (seconds)
1	1 SELECT 1 UPDATE	0.012541
2	2 SELECT 2 UPDATE	0.022824

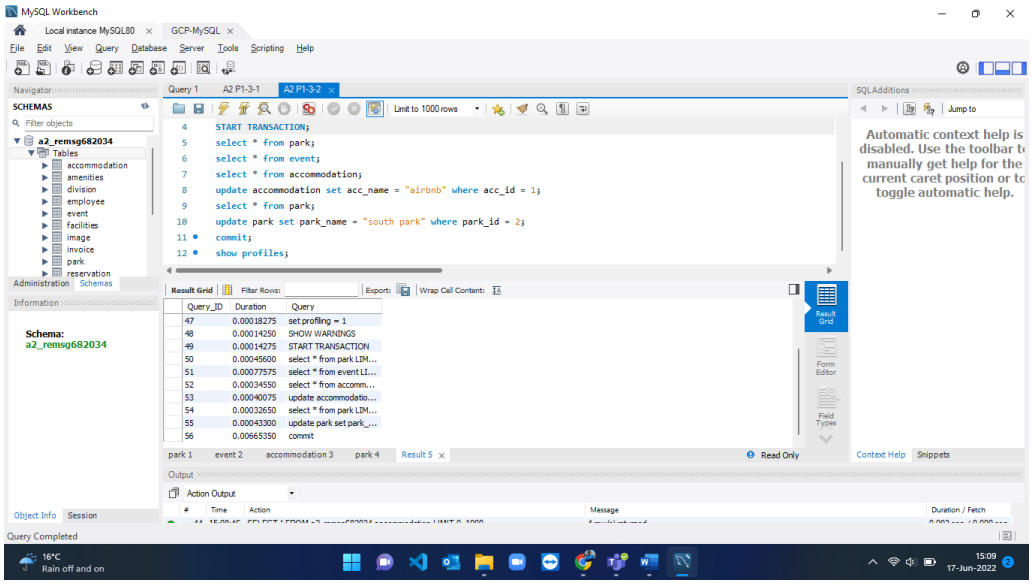
3) I have created one remote database by using GCP and named it a2_remsg682034 where I have put the 12 tables and their values. Then, I wrote the same transaction block as in 2) having 1 SELECT and 1 UPDATE statement and I recorded the execution time. The remote transaction and its output along with the recorded time is given below :





Then, I added one SELECT and UPDATE statement in the transaction and recorded the time taken by it.





Sr No.	Transaction	Time (seconds)
1	1 SELECT 1 UPDATE	0.008441
2	2 SELECT 2 UPDATE	0.010806

Transaction	Time taken in local Database	Time taken in remote database (GCP)
1 (1 SELECT & 1 UPDATE)	0.012541	0.008441
2 (2 SELECT & 2 UPDATE)	0.022824	0.010806

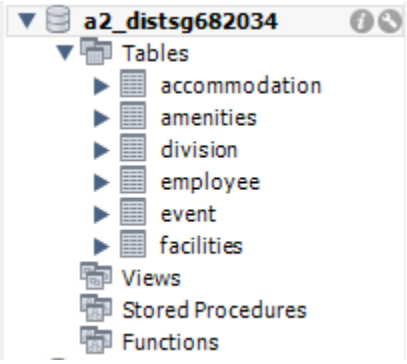
Observation and Analysis :

I have observed that in the local database the transaction having 1 SELECT and 1 UPDATE took less time compared to the transaction having 2 SELECT and 2 UPDATE.

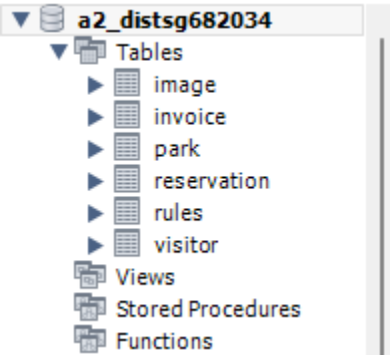
For the remote database (GCP), I have executed the same transactions as of the local database. Similar trend is observed where the transaction having 1 SELECT and 1 UPDATE took less time than the transaction having 2 SELECT and 2 UPDATE.

4) I have created 2 databases having the same name in local and remote(GCP) : a2_distsg682034. I have a total of 12 tables. So, I put 6 tables from that in the local database and 6 in the remote database. Then, I wrote java code to connect both the databases and I wrote a transaction block having 1 SELECT and 1 UPDATE executed on a local database and 1 SELECT and 1 UPDATE executed on a remote database.

Tables in a2_distsg682034 (local database) :



Tables in a2_distsg682034 (remote database) :



I have used HashMap to create a Global Data Catalog. Here, I have passed two arguments in the HashMap one is the table name and second is the connection. By using this way we can identify which table is located in which partition.

```
2 usages
public class GDC
{
    14 usages
    public HashMap<String, Connection> dataDictionary;

    1 usage
    public GDC(Connection local, Connection remote)
    {
        this.dataDictionary = new HashMap<String, Connection>();

        this.dataDictionary.put("accommodation",local);
        this.dataDictionary.put("amenities",local);
        this.dataDictionary.put("division",local);
        this.dataDictionary.put("employee",local);
        this.dataDictionary.put("event",local);
        this.dataDictionary.put("facilities",local);

        this.dataDictionary.put("image",remote);
        this.dataDictionary.put("invoice",remote);
        this.dataDictionary.put("park",remote);
        this.dataDictionary.put("reservation",remote);
        this.dataDictionary.put("rules",remote);
        this.dataDictionary.put("visitor",remote);
    }
```

I have written a transaction block having 1 SELECT and 1 UPDATE statement for the table(accommodation) in the local database and 1 SELECT and 1 UPDATE statement for the table(park) in the remote database.

```
public String statement1 = "set profiling = 1;";
1 usage
public String statement2 = "select * from accommodation;";
1 usage
public String statement3 = "update accommodation set acc_name = 'my_acc' where acc_id = 1;";
1 usage
public String statement4 = "select * from park;";
1 usage
public String statement5 = "update park set park_name = 'my_park' where park_id = 2;";
```

After execution of the whole transaction block which performs operations on both local and remote database, I measured the time taken by both and the output screenshot for the time taken is attached below: