INF2003: Database Systems
Lab 05: SQL Transaction

Objectives
To gain a practical understanding of transactions in SQL.
Deliverables
You are required to finish the lab tasks listed below and submit your report. All tasks shall be completed.

Format: **ONE (1) PDF** file.
Filename: <lab id>_<student id>_<name>_Lab5. pdf,
e.g., P1_2002008_your_name_Lab5.pdf.
Venue: INF2003@Dropbox in xSiTe.
Deadline: The following Thursday (11:59pm) after the lab is conducted.

Background Problem Statement
After the past labs, you should be experienced in using SQL in different environments, e.g., laptop, edge device, and Python. You should also be familiar with various SQL operations, especially those based on CRUD and serial. In this lab, we go deeper and investigate SQL transactions.

Please provide the SQL scripts and screenshots of the results for each question in the lab report.

Exercise:
Q1. Set up users and database and complete the following tasks:

- Create two database users (trans_user1 and trans_user2) at localhost with full privileges and list the users. (You may do this in the root account).

  SQL Statements:

```
CREATE USER ‘trans_user1’@localhost;
CREATE USER ‘trans_user2’@localhost;

GRANT ALL PRIVILEGES ON * . * TO ‘trans_user1’@localhost;
GRANT ALL PRIVILEGES ON * . * TO ‘trans_user2’@localhost;
```

- Create a new database named “TransTest”, create a new table named “mobile_credit” with column “hp_number” as integer and primary key, column “hp_name” as char, and column “hp_credit” as integer. Select and show all data in the new table.

  SQL Statements:

```
MariaDB [(none)]> CREATE DATABASE TransTest;
Query OK, 1 row affected (0.003 sec)

MariaDB [(none)]> USE TransTest;
Database changed
MariaDB [TransTest]> CREATE TABLE mobile_credit (
    -> hp_number INT PRIMARY KEY,
    -> hp_name CHAR(50),
    -> hp_credit INT);
Query OK, 0 rows affected (0.022 sec)

MariaDB [TransTest]> SELECT * FROM mobile_credit;
Empty set (0.002 sec)
```

Q2. Start a transaction:

- Start TWO (2) sessions for trans_user1 and trans_user2 respectively.
- In trans_user1's session, start a transaction, insert 2 tuples in the mobile_credit table, and show all data in the table.

    12345678, 'Lewis Hamilton', 200
    87654321, 'Max Verstappen', 200

SQL Statements:

```
USE transtest;

START transaction;

INSERT INTO mobile_credit (hp_number, hp_name, hp_credit)
VALUES (12345678, 'Lewis Hamilton', 200);
INSERT INTO mobile_credit (hp_number, hp_name, hp_credit)
VALUES (87654321, 'Max Verstappen', 200);
```

```
SELECT * FROM mobile_credit;
```

hp_number	hp_name	hp_credit
12345678	Lewis Hamilton	200
87654321	Max Verstappen	200
NULL	NULL	NULL

- In trans_user2's session, select and show all data in the table and explain why or why not the tuples you just inserted are visible to trans_user2. **Ans:**

hp_number	hp_name	hp_credit
NULL	NULL	NULL

The table is empty as trans_user1 has not committed the transaction

Q3. Run COMMIT:

- Go back to trans_user1's session and execute "COMMIT". Then in trans_user2's session, select and show all data in the table, and explain why or why not the tuples you just inserted are visible in this session.

Ans:

hp_number	hp_name	hp_credit
12345678	Lewis Hamilton	200
87654321	Max Verstappen	200
NULL	NULL	NULL

Q4. Run UPDATE:

- Go back to trans_user1's session and use UPDATE statements to reduce Lewis Hamilton's mobile credit by 100 and add Max Verstappen's hp_credit by 100. Then run COMMIT.

SQL Statements: By trans_user1

```
UPDATE mobile_credit SET hp_credit = hp_credit - 100 WHERE
hp_number = 12345678;
UPDATE mobile_credit SET hp_credit = hp_credit + 100 WHERE
hp_number
```

hp_number	hp_name	hp_credit
12345678	Lewis Hamilton	100
87654321	Max Verstappen	300
NULL	NULL	NULL

- In trans_user2's session, select and show all data in the table, and observe and indicate if the credits have been updated successfully.

SQL Statements: By trans_user2

hp_number	hp_name	hp_credit
12345678	Lewis Hamilton	100
87654321	Max Verstappen	300
NULL	NULL	NULL

Q5. Run ROLLBACK:

- Go back to trans_user1's session, add SAVEPOINT sp_1 and sp_2 into the existing code after the two UPDATE statements and add ROLLBACK statement to roll back to the 1st SAVEPOINT, sp_1. Then run COMMIT.

SQL Statements: By trans_user1

```
UPDATE mobile_credit SET hp_credit = hp_credit - 100 WHERE
hp_number = 12345678;
SAVEPOINT sp_1;
SELECT * FROM mobile_credit;
UPDATE mobile_credit SET hp_credit = hp_credit + 100 WHERE
hp_number = 87654321;
SAVEPOINT sp_2;
SELECT * FROM mobile_credit;
ROLLBACK to sp_1;
SELECT * FROM mobile_credit;
COMMIT;
```

hp_number	hp_name	hp_credit
12345678	Lewis Hamilton	0
87654321	Max Verstappen	300
NULL	NULL	NULL

- In trans_user2's session, select and show the data in the table. Observe and indicate if can access the data and why.

SQL Statements: By trans_user2

hp_number	hp_name	hp_credit
12345678	Lewis Hamilton	0
87654321	Max Verstappen	300

Ans:

Yes, trans_user2 is able to access the data as trans_user2 has committed the changes

Q6. (Optional) Run LOCK:

- In trans_user1's session, create an exclusive (WRITE) LOCK to "mobile_credit" table, add Lewis Hamilton's credit by 100 and select and show all data in the table.

SQL Statements: By trans_user1

```
START TRANSACTION;

LOCK TABLES mobile_credit WRITE;

UPDATE mobile_credit SET hp_credit = hp_credit + 100 WHERE
hp_number = 12345678;
SELECT * FROM mobile_credit;
COMMIT;
```

- Switch to trans_user2's session, try to select and show all data in the table. Observe and indicate if the query can be executed successfully. Some of you may be surprised that you can execute

it successfully. If so, you may close both of your sessions, and re-connect them. Observe again and describe why.

SQL Statements: By trans_user2

```
 SELECT * FROM mobile_credit;
 Error Code: 2013. Lost connection to MySQL server during
query
```

Ans:

Trans_user2 is blocked from reading the table until the lock is released as trans_user1 has locked the table to ensure exclusive write access

- After those tests, you may unlock the tables in trans_user1's session, and re-test in trans_user2's session, to see if anything changes.

SQL Statements: By trans_user1

SQL Statements: By trans_user2

Ans:

------ END ------