**Question 1:** **Based on the past labs, create two database users (trans\_user1 and**

**trans\_user2) at localhost with full privileges. (You may do this in**

**the root account). Then, create a new test database with your name**

**“Trans<name>”, e.g., TransWei.**

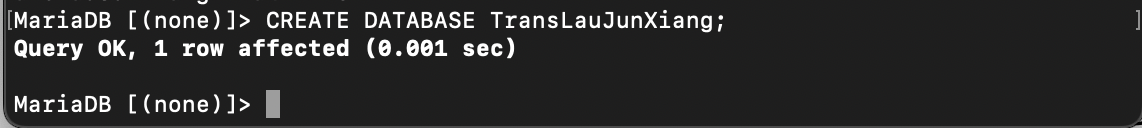
Command used:

1. CREATE USER 'trans\_user1'IDENTIFIED BY 'P@ssw0rd';
2. CREATE USER 'trans\_user2'IDENTIFIED BY 'P@ssw0rd';  
   Text

   Description automatically generated

Text

Description automatically generated

1. CREATE DATABASE TransLauJunXiang; 

**Question 2:** **Start TWO (2) sessions, each for one transaction user. (You may close**

**the root user’s session.) The 2 sessions run concurrently.**

Command used:

1. Text

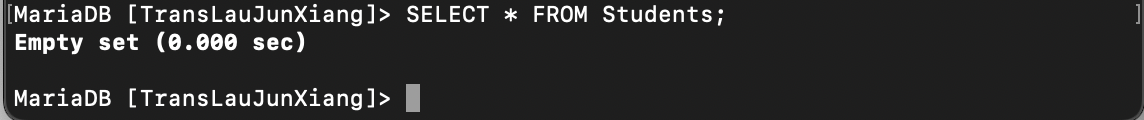
   Description automatically generated

**Question 3:** **In trans\_user1’s session, create a new table “Students” in the**

**database, with your familiar settings, e.g., column sid as integer and primary key, column sname as char, and column grade also as char. Select and show all data in the new table, and check if it is empty.**

Command used:

1. CREATE TABLE Students (sid INT PRIMARY KEY, sname CHAR(30), grade CHAR(3));  
   Text

   Description automatically generated
2. Through the use of the SQL command of SELECT statement, it has returned “Empty set” that indicates the table is indeed empty. Since the table was just created, there should also not be any data.  
   

**Question 4: In trans\_user2’s session, specify the same database. Select and show all the tables in the database. Observe and describe why or why not the table “Students’ is visible in this session.**

Command used:

1. Students table is visible from using trans\_user2. It is because both users have been granted all privileges and it is not constrained to anything.  
   Text

   Description automatically generated

**Question 5: Go back to trans\_user1’s session. Start a transaction. This is a key**

**step. In the transaction, randomly insert 3 tuples into the table.**

**Below are 3 sample tuples.**

**2104444, 'Lewis Hamilton', 'NA'**

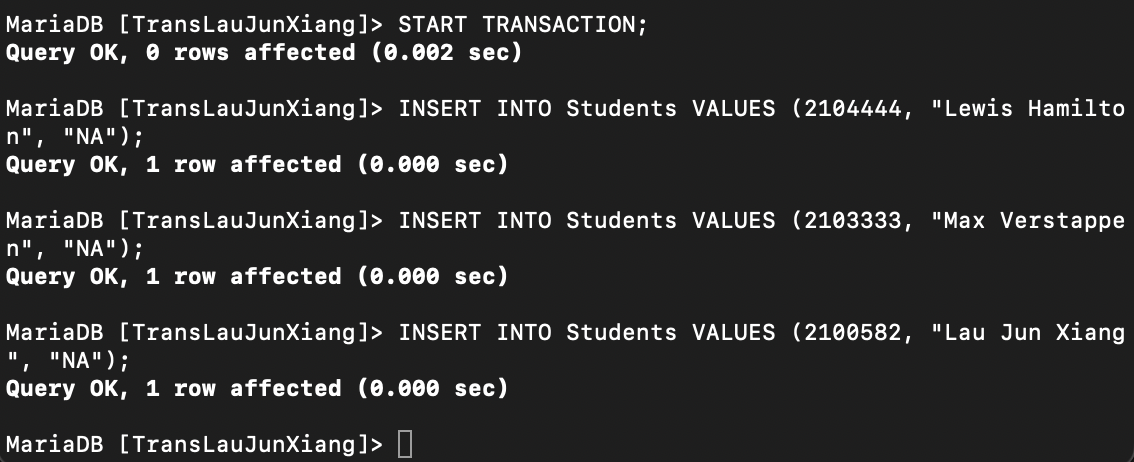
**2103333, 'Max Verstappen', 'NA'**

**<your student id>, <your name>, 'NA' Select and return the data in the table. You may not be surprised to see the tuples are already in the table.**

Command used:

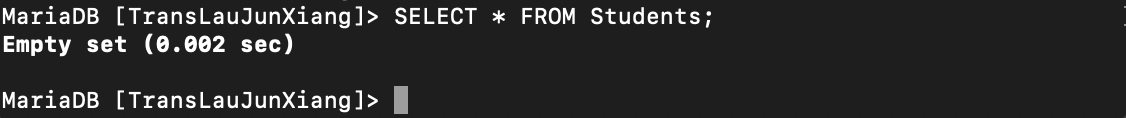
1. START TRANSACTION;  
   INSERT INTO Students VALUES (2104444, "Lewis Hamilton", "NA");

INSERT INTO Students VALUES (2103333, "Max Verstappen", "NA");

INSERT INTO Students VALUES (2100582, "Lau Jun Xiang", "NA");  


**Question 6: Switch back to trans\_user2’s session. Specify the database and show the tuples in the table “Students”. Observe and describe why or why not the tuples you just inserted are visible in this session.**

Command used:

1. Since the transaction was started by trans\_user1 but not committed, trans\_user2 will not be able to see.
2. SELECT \* FROM Students --- From trans\_user2  
   

**Question 7: Go back to trans\_user1’s session. Execute “COMMIT”. Then in**

**trans\_user2’s session, show the tuples in the table “Students” again.**

**For one more time, observe and describe why or why not the tuples**

**you just inserted are visible in this session.**

Command used:

1. COMMIT; SQL command have been used on trans\_user1 and SELECT \* FROM Students have been used on trans\_user2.
2. Since the COMMIT has been executed by trans\_user1, trans\_user2 will then be able to see the newly inserted data.  
   Graphical user interface, text

   Description automatically generated

**Question 8: In either session, start a transaction. In the transaction, first**

**update the grade of Lewis Hamilton to ‘A’ and create a SAVEPOINT as**

**sp\_lh. Then, update the grade of yourself to ‘A’ as well and create**

**another SAVEPOINT as sp\_me. After the two update operations, execute**

**ROLLBACK in the transaction. Select and return the data in the table.**

**Observe and indicate if the grades have been updated successfully.**

Command used:

1. START TRANSACTION;  
   UPDATE Students SET grade = "A" WHERE sname = "Lewis Hamilton";  
   SAVEPOINT sp\_lh;  
   UPDATE Students SET grade = "A" WHERE sname = "Lau Jun Xiang";  
   SAVEPOINT sp\_me;  
   ROLLBACK;  
   SELECT \* FROM Students;  
     
   Text

   Description automatically generated

**Question 9: Based on the above code, modify the ROLLBACK statement to roll back**

**to the 1st SAVEPOINT, sp\_lh. Select and return the data in the table.**

**Observe and indicate if any grades have been updated.**

**grades have been updated successfully.**

Command used:

1. START TRANSACTION;  
   UPDATE Students SET grade = "A" WHERE sname = "Lewis Hamilton";  
   SAVEPOINT sp\_lh;  
   UPDATE Students SET grade = "A" WHERE sname = "Lau Jun Xiang";  
   SAVEPOINT sp\_me;  
   ROLLBACK to sp\_lh;  
     
   As seen above, it rollback to sp\_lh and my grade is still NA. This is because there was a snapshot at the point of time and I used the command to rollback before my update made.  
   Text

   Description automatically generated

**Question 10: Based on the above code, add one more statement right above the**

**ROLLBACK one. The statement is to release the SAVEPOINT sp\_lh.**

**Execute the new query. Observe if the execution is successful.grades have been updated successfully.**

Command used:

1. START TRANSACTION;  
   UPDATE Students SET grade = "A" WHERE sname = "Lewis Hamilton";  
   SAVEPOINT sp\_lh;  
   UPDATE Students SET grade = "A" WHERE sname = "Lau Jun Xiang";  
   SAVEPOINT sp\_me;  
   RELEASE SAVEPOINT sp\_lh;  
   ROLLBACK;  
   SELECT \* FROM Students;  
   Text

   Description automatically generated

**Question 10: Based on the above code, add one more statement right above the**

**ROLLBACK one. The statement is to release the SAVEPOINT sp\_lh.**

**Execute the new query. Observe if the execution is successful.grades have been updated successfully.**

Command used:

1. 