Experiment No.7 Perform DCL and TCL commands
Date of Performance:
Date of Submission:

Aim :- Write a query to implement Data Control Language(DCL) and Transaction Control Language(TCL) commands

Objective :- To learn DCL commands like Grant and Revoke privileges to the user and TCL commands to commit the transactions and recover it using rollback and save points.

Theory:

Data Control Language:

DCL commands are used to grant and take back authority from any database user.

- Grant
- Revoke

a. Grant: It is used to give user access privileges to a database.

Example

```
1. GRANT SELECT, UPDATE ON MY_TABLE TO SOME_USER,
   ANOTHER_USER;
```

b. Revoke: It is used to take back permissions from the user.

Example

```
1. REVOKE SELECT, UPDATE ON MY_TABLE FROM USER1, USER2;
```

Transaction Control Language

TCL commands can only use with DML commands like INSERT, DELETE and UPDATE only.

These operations are automatically committed in the database that's why they cannot be used while creating tables or dropping them.

Here are some commands that come under TCL:

- COMMIT
- ROLLBACK
- SAVEPOINT

a. Commit: Commit command is used to save all the transactions to the database.

Syntax:

```
1. COMMIT;
```

Example:

```
1. DELETE FROM CUSTOMERS
2. WHERE AGE = 25;
3. COMMIT;
```

b. Rollback: Rollback command is used to undo transactions that have not already been saved to the database.

Syntax:

```
1. ROLLBACK;
```

Example:

```
1. DELETE FROM CUSTOMERS
2. WHERE AGE = 25;
3. ROLLBACK;
```

c. SAVEPOINT: It is used to roll the transaction back to a certain point without rolling back the entire transaction.

Syntax:

```
3.  SAVEPOINT SAVEPOINT_NAME;
```

Implementation:

Code:

DCL Commands:

farm_database

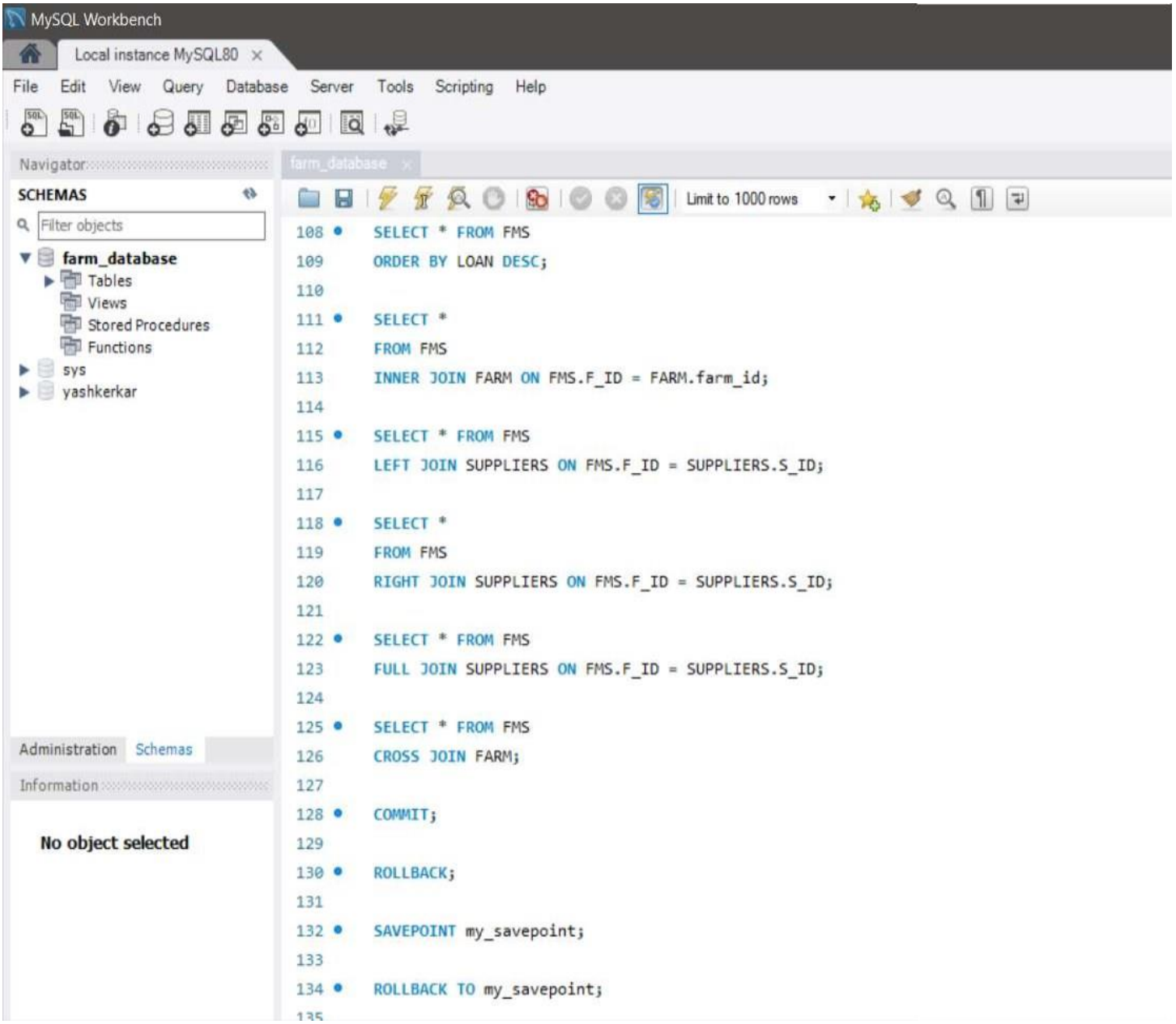
Limit to 1000 rows

```
128
129 •  COMMIT;
130
131 •  ROLLBACK;
132
133 •  SAVEPOINT my_savepoint;
134
135 •  ROLLBACK TO my_savepoint;
136
137 •  CREATE USER 'ankit'@'localhost';
138
139 •  DROP USER 'ankit'@'localhost';
140
141 •  GRANT SELECT, INSERT, UPDATE, DELETE ON FMS TO 'ankit'@'localhost';
142
143 •  GRANT SELECT, INSERT, UPDATE, DELETE ON FMS TO 'ankit'@'localhost';
144
145
146
```

Output:

33	10:00:19	CREATE USER 'ankit'@'localhost'	0 row(s) affected
34	10:00:24	GRANT SELECT, INSERT, UPDATE, DELETE ON FMS TO 'ankit'@'localhost'	0 row(s) affected
35	10:00:33	GRANT SELECT, INSERT, UPDATE, DELETE ON FMS TO 'ankit'@'localhost'	0 row(s) affected

TCL Commands



Output:

✓	22	09:43:58	COMMIT	0 row(s) affected
✓	23	09:44:05	ROLLBACK	0 row(s) affected
✓	24	09:44:10	SAVEPOINT my_savepoint	0 row(s) affected

Conclusion:

In this practical session, we explored the creation of a database schema for managing farm-related data. We defined tables for farmers, farms, suppliers, and tools using SQL commands. Additionally, we inserted sample data into these tables to simulate a real-world scenario.

Throughout the session, we performed various SQL queries to retrieve and manipulate data, including filtering, sorting, and joining data from multiple tables. We also utilized Transaction Control Language (TCL) commands to manage transactions, ensuring data integrity and consistency within the database.

Furthermore, we demonstrated the use of Data Control Language (DCL) commands to control access privileges within the database system. By granting and revoking privileges to specific users or roles, we managed data security effectively.

1. Explain about issues faced during rollback in mysql and how it got resolved.

Ans. **Deadlocks:** Deadlocks occur when two or more transactions are waiting for each other to release locks on resources that the other transactions need. This situation can lead to a deadlock, where none of the transactions can proceed. To resolve deadlocks, MySQL automatically detects them and chooses one transaction to be the victim, rolling it back to break the deadlock and allow the other transactions to proceed.

Lock Contention: Lock contention happens when multiple transactions are trying to access the same resources concurrently, leading to contention for locks. This contention can cause delays and performance issues. To mitigate lock contention, it's essential to design transactions and queries in a way that minimizes the need for locking, such as using appropriate isolation levels and optimizing queries.

Resource Exhaustion: During rollback, MySQL may encounter resource exhaustion issues, such as running out of memory or disk space. This situation can occur when rolling back a large transaction that has consumed significant resources. To address resource exhaustion, you can increase system resources, optimize transactions to reduce their resource consumption, or break down large transactions into smaller ones.

Partial Rollbacks: In some cases, a rollback operation may fail to complete fully due to errors or interruptions. This can leave the database in an inconsistent state, with some changes rolled back and others not. To handle partial rollbacks, you can use savepoints to mark intermediate points within transactions, allowing you to roll back to a specific savepoint rather than the beginning of the transaction.

Data Integrity: Rollback operations should maintain data integrity, ensuring that the database remains in a consistent state after the rollback. However, if the rollback process encounters errors or inconsistencies, it may fail to restore the database to its original state. To ensure data integrity, it's crucial to perform thorough testing and validation of rollback operations and implement appropriate error handling mechanisms.

2. Explain how to create a user in sql.

Ans. Identify the Database Management System (DBMS): SQL is a standard language for interacting with databases, but different DBMSs have their specific syntax for creating users. Common database systems include MySQL, PostgreSQL, SQL Server, Oracle, etc. Make sure you're familiar with the syntax specific to the DBMS you're using.

Connect to the Database: Before creating a user, ensure that you are connected to the database where you want to create the user. You'll typically use a database management tool or command-line interface to execute SQL commands.

Execute the Appropriate SQL Command: Use the appropriate SQL command to create a user. Below are examples of how to create a user in different database systems:

MYSQL:

```
CREATE USER 'username'@'hostname' IDENTIFIED BY 'password';
```