



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

AY: 2024-25

Class:	SE	Semester:	IV
Course Code:	CSL402	Course Name:	Database Management System Lab

Name of Student:	Shruti Gauchandra
Roll No. :	16
Experiment No.:	7
Title of the Experiment:	Perform DCL and TCL Commands.
Date of Performance:	13/02/25
Date of Submission:	20/02/25

Evaluation

Performance Indicator	Max. Marks	Marks Obtained
Performance	5	
Understanding	5	
Journal work and timely submission	10	
Total	20	

Performance Indicator	Exceed Expectations (EE)	Meet Expectations (ME)	Below Expectations (BE)
Performance	4-5	2-3	1
Understanding	4-5	2-3	1
Journal work and timely submission	8-10	5-8	1-4

Checked by

Name of Faculty : Ms. Neha Raut

Signature :

Date:



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

Experiment No 7

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



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

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:

2. SAVEPOINT SAVEPOINT_NAME;

Implementation:

Code:

BEGIN;

-- Insert new customers

INSERT INTO Customer (Customer_ID, F_name, L_Name, Email_id, Mobile_no, DOB, Gender)

VALUES

(11, 'David', 'Clark', 'davidc@example.com', '9012345678', '1990-12-05', 'Male'),

(12, 'Emma', 'Taylor', 'emmat@example.com', '9123456781', '1993-07-22', 'Female');



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

-- Insert new theatres

```
INSERT INTO Theatre (t_id, t_name, t_location, t_pincode)
```

```
VALUES
```

```
(11, 'IMAX Cinemas', 'Downtown, Los Angeles', '90001'),
```

```
(12, 'Regal Cinema', 'Times Square, New York', '10036');
```

```
SAVEPOINT before_tickets;
```

-- Insert new ticket bookings

```
INSERT INTO Ticket (Ticket_ID, Customer_ID, t_id, Movie_Name, Ticket_Price, Show_Date)
```

```
VALUES
```

```
(6, 11, 11, 'Dune Part Two', 320.00, '2024-04-06'),
```

```
(7, 12, 12, 'Oppenheimer', 280.00, '2024-04-07');
```

```
SELECT * FROM Ticket;
```

-- Rollback if there is an issue with ticket bookings

```
ROLLBACK TO before_tickets;
```

```
SELECT * FROM Ticket;
```

-- Release the savepoint

```
RELEASE SAVEPOINT before_tickets;
```

```
COMMIT;
```

Output:

	Ticket_ID	Customer_ID	t_id	Movie_Name	Ticket_Price	Show_Date
▶	1	1	1	Inception	250.00	2024-04-01
	2	2	3	Avengers	300.00	2024-04-02
	3	3	5	Interstellar	275.00	2024-04-03
	4	4	7	Titanic	200.00	2024-04-04
	5	5	9	Joker	350.00	2024-04-05
*	NULL	NULL	NULL	NULL	NULL	NULL



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

Conclusion:

Performing DCL (Data Control Language) and TCL (Transaction Control Language) commands is vital for ensuring data security, integrity, and consistency within a database system. DCL commands, such as GRANT and REVOKE, are used to manage user permissions, allowing administrators to control who can access or modify data, thereby enhancing overall security. On the other hand, TCL commands like COMMIT, ROLLBACK, and SAVEPOINT help manage transactions by grouping operations into a single unit of work, ensuring that either all changes are saved or none at all in the event of an error.

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

Ans. Rollback issues in MySQL arise due to various factors such as auto-commit mode, non-transactional storage engines, missing savepoints, stored procedure errors, and foreign key constraints. By default, MySQL operates in auto-commit mode, meaning every statement is immediately committed, preventing rollback from working unless auto-commit is disabled or transactions are explicitly started using START TRANSACTION . Additionally, MyISAM tables do not support transactions, so switching to the InnoDB storage engine ensures rollback functionality. Savepoints play a crucial role in rolling back specific parts of a transaction rather than the entire operation, allowing better control over database modifications. In stored procedures, rollback failures can occur if errors are not handled properly, which can be resolved by using DECLARE EXIT HANDLER to catch exceptions and ensure rollback is executed. Foreign key constraints can also cause rollback failures if a referenced row is deleted, requiring temporary disabling of constraints before performing operations. By addressing these issues through proper transaction management, storage engine selection, and error handling, rollback in MySQL can be effectively implemented without failures.

B) Explain how to create a user in sql.

Ans. To create a user in SQL, the CREATE USER statement is used, specifying the username, host, and password for authentication. For example, in MySQL, a user can be created with CREATE USER 'user1'@'localhost' IDENTIFIED BY 'securepassword'; , which allows the user to connect only from the local machine. After creating the user, privileges must be granted using the GRANT command, such as GRANT ALL PRIVILEGES ON database_name.* TO 'user1'@'localhost'; , followed by FLUSH PRIVILEGES; to apply the changes. To verify created users, the SELECT user, host FROM mysql.user; query can be used. Proper user creation and privilege management ensure secure database access and controlled permissions for different users.