

Assignment 4

TITLE

To perform:

- a) Altering a Table
- b) Dropping/Truncating/Renaming Tables
- c) Backing up and Restoring a Database

PROBLEM STATEMENT

Perform SQL operations for altering the structure of tables, removing table data or entire tables, renaming existing tables, and backing up and restoring databases.

This lab will help in understanding structural modifications of database schema and essential data recovery practices.

OBJECTIVES

- To learn how to alter the structure of a table (add, modify, delete columns).
- To learn how to delete table data using TRUNCATE and DROP commands.
- To learn to rename a table.
- To understand the process of backing up and restoring a database.
- To differentiate between DROP, DELETE, and TRUNCATE operations.

PLATFORM REQUIRED

- **Operating System:** Windows or Linux
- **Software/Tools:** MySQL, Oracle, or any SQL environment (MySQL Workbench, XAMPP, Command Line)

THEORY

Altering a Table

Used to modify the structure of an existing table.

Syntax:

-- Add a new column

```
ALTER TABLE table_name ADD column_name datatype;
```

-- Modify a column

```
ALTER TABLE table_name MODIFY column_name new_datatype;
```

-- Drop a column

```
ALTER TABLE table_name DROP COLUMN column_name;
```

Dropping a Table

Completely removes a table and its data from the database.

Syntax:

```
DROP TABLE table_name;
```

Truncating a Table

Deletes all rows from a table but retains the structure for future use.

Syntax:

```
TRUNCATE TABLE table_name;
```

Renaming a Table

Used to change the name of an existing table.

Syntax:

```
RENAME TABLE old_table_name TO new_table_name;
```

Backing up a Database

Export the database to a backup file using tools like mysqldump.

Command-line Syntax:

```
mysqldump -u username -p database_name > backup_file.sql
```

Restoring a Database

Import the backup file back into MySQL.

Command-line Syntax:

```
mysql -u username -p database_name < backup_file.sql
```

SCENARIO EXAMPLE

Consider the Student table:

Student_ID Name Address Phone

1 Amit Pune 9876543210

2 Priya Mumbai 9123456780

Example Commands:

- Add an Email column:

ALTER TABLE Student ADD Email VARCHAR(50);

- Modify Phone column:

ALTER TABLE Student MODIFY Phone BIGINT;

- Drop Address column:

ALTER TABLE Student DROP COLUMN Address;

- Rename the table:

RENAME TABLE Student TO Student_Info;

- Truncate all rows:

TRUNCATE TABLE Student_Info;

- Drop the table:

DROP TABLE Student_Info;

- Backup the database:

mysqldump -u root -p college_db > college_db_backup.sql

- Restore the database:

mysql -u root -p college_db < college_db_backup.sql

STEP-BY-STEP ALGORITHM

1. Open your SQL environment and connect to your database server.
2. Select the working database using:
USE database_name;
3. Alter the table using ALTER TABLE to add, modify, or drop a column.
4. Use RENAME TABLE to rename an existing table.
5. Use TRUNCATE TABLE to delete all rows but retain structure.
6. Use DROP TABLE to permanently delete the table.
7. To back up the database, use mysqldump on the command line.
8. To restore, use mysql command with the backup file.
9. Use SELECT * FROM table_name; to verify changes at every step.

QUESTIONS FOR PRACTICE

1. Write a command to add a column DOB (date of birth) to the Student table.
2. How do you permanently remove the Teacher table from the database?

3. What is the difference between DROP and TRUNCATE?
4. Write a command to rename the table Library to Library_Records.
5. What is the syntax to back up a database using mysqldump?
6. What command would you use to restore a database from a .sql file?
7. Explain why using TRUNCATE is faster than DELETE for large tables.

CONCLUSION

In this lab, you learned essential SQL commands used for altering the structure of tables, removing or renaming them, and backing up/restoring entire databases. These operations form the core of schema evolution and data safety practices in DBMS. Understanding the differences between TRUNCATE, DROP, and DELETE is vital for managing data accurately. Additionally, mastering backup and restore ensures your ability to protect and recover critical data efficiently, a fundamental requirement in real-world database systems.