LAB 1

Title: Introduction to DDL and DML statements in SQL

Objective:

- To be familiar with concept of DDL and DML
- To use different DDL and DML statements to perform various operations on database

Theory:

SQL (Structured Query Language)

- SQL stands for Structured Query Language
- SQL let us to access and manipulate databases
- SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standardization (ISO) in 1987

DDL (Data Definition Language)

- ✓ DDL or Data Definition Language actually consists of the SQL commands that can be used to define the database schema.
- ✓ It simply deals with descriptions of the database schema and is used to create and modify the structure of database objects in the database.
- ✓ DDL is a set of SQL commands used to create, modify, and delete database structures but not data.

Examples of DDL commands:

CREATE: This command is used to create the database or its objects (like table,views, store procedure and triggers).

DROP: This command is used to delete objects from the database.

ALTER: This is used to alter the structure of the database.

TRUNCATE: This is used to remove all records from a table, including all spaces allocated for the records are removed.

DML (Data Manipulation Language)

The SQL commands that deals with the manipulation of data present in the database belong to DML.

Examples of DML:

INSERT – is used to insert data into a table.

UPDATE – is used to update existing data within a table.

DELETE – is used to delete records from a database table

SELECT- It is used to retrieve data from the database table.

Now let us discuss a detailed description of DDL and DML language and their usage with their syntax.

DDL

CREATE

To Create Database

- ✓ ***CREATE DATABASE DatabaseName;***
  eg: CREATE DATABASE employeedb;

To Create table

```
CREATE TABLE table_name (
    column1 datatype,
    column2 datatype,
    column3 datatype,
     ....
    columnn datatype
);
```

Example

```
CREATE TABLE employee_info
(eid int,
name varchar(50),
address varchar(50),
position varchar(50),
salary decimal(10,2)
);
```

DROP

To Remove Database

DROP DATABASE DatabaseName;

Eg. DROP DATABASE employeedb;

To Remove table

DROP TABLE table_name;

Eg; DROP TABLE employee_info;

TRUNCATE

To remove all rows from table

TRUNCATE TABLE table_name;

Eg. TRUNCATE TABLE employee_info;

ALTER

- The ALTER TABLE statement is used to add, delete, or modify columns in an existing table.
- It is also used to add and drop various constraints on an existing table.

To add Column in table

ALTER TABLE table_name
ADD column_name datatype;

Eg.

ALTER TABLE employee_info

ADD department varchar(30);

To remove column from table

ALTER TABLE table_name
DROP COLUMN column_name;

Eg. ALTER TABLE employee_info
DROP COLUMN address;

To rename column of table

```
ALTER TABLE table_name
CHANGE COLUMN old_name  new_name datatype;
```

Eg. ALTER TABLE employee_info
CHANGE COLUMN address location varchar(30);

(Note: This syntax is for MariaDB and any vary in different DBMS)

To modify data type of any column

ALTER TABLE table_name
MODIFY column_name datatype;

Eg.

ALTER TABLE employee_info

MODIFY location char(20);

DML

INSERT

To Insert data into table

Method-I

INSERT INTO table_name (column1, column2, column3, ...)

VALUES (value1, value2, value3, ...);

Eg.

INSERT INTO employee_info(eid,name,address)

VALUES(2,'ram','pokhara');

Method-II

INSERT INTO table_name

VALUES (value1, value2, value3, ...);

Eg.

INSERT INTO employee_info

VALUES(1,'Hari','kathmandu','manager',55625);

UPDATE

To update existing data within a table.

```
UPDATE table_name
SET column1 = value1, column2 = value2, ...
WHERE condition;
```

Eg.

UPDATE employee_info

SET salary=75000

WHERE position='manager';

DELETE

To delete records from a table

```
DELETE FROM table_name WHERE condition;
```

DELETE FROM employee_info where address='chitwan';

SELECT

To retrieve data from the database table.

```
SELECT column1, column2, ...
FROM table_name
WHERE condition;
```

Eg.

SELECT position,salary
from employee_info

where salary >30000;

Some useful commands

- ✓ To Select Database

  USE DatabaseName;

  Eg. use employee_db;

- ✓ **To Show Databases**

  show databases;

- ✓ **To see which database is selected**

  select database();

- ✓ **To see tables in a selected database**

  show tables;

Problem:

1. Create a database named **eemcDB**

2. Create table named **student_info** database named **eemcDB** with following columns and datatypes

Sid	Name	contact	Faculty	College_name
Int	Varchar(50)	char(10)	Varchar(50)	Varchar(50)

3.Now add column named address with datatype varchar(30)

4. Delete the column named contact

5.Rename column named address as location

6.Change data type of faculty to char(20)

7. Insert minimum 10 information of student into table named student_info

- ✓ Insert 1 information of student whose faculty is not known
- ✓ Insert 1 information of student whose college_name is not known

8. Update the information of student whose sid=3 by setting faculty =’civil’

9.Update the information of student whose name is ‘ram’ and location is ‘kathmandu’ by setting faculty=’computer’

10.Delete the information of student whose faculty is civil and location is pokhara

11. Display all the information of student from table named student_info

12.Display name and faculty of student whose location is Kathmandu

13.Display name and faculty of student whose location is pokhara and college_name is eemc

14.Delete all rows from table

15.Delete the table named student_info

16.Delete the database named eemcDB;

Note: ***Students are suggested insert information of student into table in such a way that above operations can be performed.***

Discussion: *(This portion is left for student)*

Conclusion: *(This portion is left for student)*

*******THE END*******