Assignment No: 4

Aim:

Design any database with at least 3 entities and relationships between them. Apply DCL and DDL commands. Draw suitable ER/EER diagram for the system.

Objective:

- To understand the different issues involved in the design and implementation of a databasesystem
- To understand and use Data Definition Language and Data Control Language to write query for adatabase

Theory:

DATA DEFINITION LANGUAGE (DDL): The Data Definition Language (DDL) is used to create and destroy databases and database objects. These commands will primarily be used by database administrators during the setup and removal phases of a database project. **Some commands of DDL are:**

- CREATE – to create table (objects) in thedatabase
- ALTER – alters the structure of thedatabase
- DROP – delete table from thedatabase
- TRUNCATE – remove all records from a table, including all spaces allocated for the records areremoved
- RENAME – rename atable

1. CREATE:

(a) CREATE DATABASE: You can create a MySQL database by using MySQLCommand
Syntax:

CREATE DATABASE database_name;

Example:

Let's take an example to create a database name "employees"

CREATE DATABASE employees;

We can check the created database by the following query:

SHOW DATABASES;

(b) USE DATABASE: Used to select a particulardatabase.

Syntax:

USE database_name;

Example: Let's take an example to use a database name "customers".

USE customers;

(c) DROP DATABASE: You can drop/delete/remove a MySQL database easily with the MySQL command. You should be careful while deleting any database because you will lose your all the data available in yourdatabase.

Syntax:

DROP DATABASE database_name;

Example: Let's take an example to drop a database name "employees"

DROP DATABASE employees;

(d) CREATE TABLE: This is used to create a new relation(table)

The MySQL CREATE TABLE command is used to create a new table into the database.

Syntax:

Following is a generic syntax for creating a MySQL table in the database.

CREATE TABLE table_name (column_namecolumn_type...);

Example:

Here, we will create a table named "student" in the database "mydatabase".

```
CREATE TABLE cus_tbl(
  roll_no INT NOT NULL ,
  fnameVARCHAR(100) NOT NULL,
  surname VARCHAR(100) NOT NULL,
  PRIMARY KEY ( roll_no )
```

See the created table: Use the following command to see the table already created:

SHOW tables;

See the table structure: Use the following command to see the table already

created: DESCRIBE table_name;

2. ALTER:

MySQL ALTER statement is used when you want to change the name of your table or any table field. It is also used to add or delete an existing column in a table.

The ALTER statement is always used with "ADD", "DROP" and "MODIFY" commands according to the situation.

(a) ALTER TABLE ...ADD...: This is used to add some extra fields intoexisting

relation.

Syntax: ALTER TABLE relation_name ADD (new field_1 data_type(size), new field_2 data_type(size),..);

Example: ALTER TABLE student ADD (Address CHAR(10));

(b) ALTER TABLE...MODIFY...: This is used to change the width as well as data type of fields of existingrelations.

Syntax: ALTER TABLE relation_name MODIFY (field_1 newdata_type(Size), field_2 newdata_type(Size),... field_newdata_type(Size));

Example: ALTER TABLE student MODIFY(fname VARCHAR(10),class VARCHAR(5));

c) ALTERTABLE..DROP..... This is used to remove any field of existingrelations.

Syntax: ALTER TABLE relation_name DROP COLUMN (field_name);

Example: ALTER TABLE student DROP column (sname);

d)ALTER TABLE..RENAME...: This is used to change the name of fields in existingrelations.

Syntax: ALTER TABLE relation_name RENAME COLUMN (OLD field_name) to (NEW field_name);

Example: ALTER TABLE student RENAME COLUMN sname to stu_name;

3. RENAME: It is used to modify the name of the existing databaseobject.

Syntax: RENAME TABLE old_relation_name TO new_relation_name;

Example: RENAME TABLE studentd TO studentd1;

4. TRUNCATE andDROP

Difference between Truncate &Drop:-

RUNCATE: This command will remove the data permanently. But structure will not be removed.
DROP: This command will delete the table data and structure permanently.

Syntax: TRUNCATE TABLE <Table

name>***Example*** TRUNCATE TABLE student;

Syntax: DROP TABLE <Table name>***Example***

DROP TABLEstudent;

Data Control Language(DCL) : Thisis used to control privilege in Database. To perform any operation in the database, such as for creating tables, sequences or views we need privileges.

DCL defines two commands,

- **Grant :**Gives user access privileges todatabase.
- **Revoke :**Take back permissions fromuser.

Syntax: GRANT privilege_nameON object_name TO {user_name };

Example:GRANT CREATE TABLE TO user1;

REVOKE privilege_name
ON object_nam
FROM {user_name };

Example:REVOKE CREATE TABLE FROM user1;

LAB PRACTICE ASSIGNMENT:

Consider the following table structures for this assignment: Table Name 1: **CUSTOMER**

Fields:

Cust_idvarchar(10) Primary Key,C_nameVarchar(15) Not NULL,City varchar(10).

Table Name 2: **BRANCH**

Fields:

Branch_idVarchar(5) Primary Key, bnameVarchar (15), City varchar(10).

Table Name 3: **DEPOSIT**

Fields:

Acc_novarchar(10) Primary Key,Cust_idVarchar(10) Not NULL, Amount int,Branch_idVarchar(5), Open_date date.

Table Name 4: **BORROW**

Fields:

Loan_noVarchar(5) Primary Key, Cust_idVarchar (10), Branch_idvarchar(5), Amountint.

Perform the following command/operation on the above table:

1) Create aDatabase
2) ShowDatabase
3) UseDatabase
4) DropDatabase
5) Create tables and Describe thatTables
6) AlterCommand
   i) Add column address to Customertable
   ii) Modify anycolumn
   iii) Rename column address tonew_address
   iv) Drop column address from Customertable
   v) Rename table Branch to Branch1
6) Perform DCL Commands Grant and Revoke on Customertable
7) Truncatetable
8) Droptable

Note:

1) For truncate and drop command create any othertable
2) In write-up, write the description and uses of all commands with syntax in the given format
3) Printout should also consists of command execution in the given orderonly.

Conclusion:-

We have studied and created a database with at least 3 entities and relationships between them and applied DCL and DDLcommands.

OUTPUT -

CREATE DATABASE

MySQL > CREATE DATABASE COLLEGE;

Query OK, 1 row affected (0.003 sec)

SHOW DATABASES

MySQL > SHOW DATABASES;

```
+--------------------+
| Database           |
+--------------------+
| college            |
| information_schema |
| mysql              |
| performance_schema |
| phpmyadmin         |
| test               |
+--------------------+
```

6 rows in set (0.002 sec)

USE DATABASE

MySQL > USE COLLEGE

Database changed

MySQL>

CREATE CUSTOMER TABLE

MySQL> CREATE TABLE CUSTOMER (ID INT PRIMARY KEY, NAME VARCHAR(15) NOT NULL, CITY VARCHAR(10));

Query OK, 0 rows affected (0.534 sec)

DESCRIBE

MySQL> DESC CUSTOMER;

```
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| ID    | int(11)     | NO   | PRI | NULL    |       |
| NAME  | varchar(15) | NO   |     | NULL    |       |
| CITY  | varchar(10) | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
```

3 rows in set (0.017 sec)

CREATETABLE BRANCH

MySQL> CREATE TABLE BRANCH (ID INT PRIMARY KEY, BNAME VARCHAR(15), CITY VARCHAR(10));

Query OK, 0 rows affected (0.247 sec)

DESCRIBE

MySQL> DESC BRANCH;

```
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| ID    | int(11)     | NO   | PRI | NULL    |       |
| BNAME | varchar(15) | YES  |     | NULL    |       |
| CITY  | varchar(10) | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
```

3 rows in set (0.009 sec)

CREATE DEPOSIT TABLE

MySQL> CREATE TABLE DEPOSIT (ACC_NUMBER VARCHAR(5) PRIMARY KEY, CUST_ID VARCHAR(10) NOT NULL, AMOUNT INT, BRANCH_ID VARCHAR(5), OPEN_DATE DATE);

Query OK, 0 rows affected (0.535 sec)

DESCRIBE:

MySQL> DESC DEPOSIT;

```
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| ACC_NUMBER | varchar(5)  | NO   | PRI | NULL    |       |
| CUST_ID    | varchar(10) | NO   |     | NULL    |       |
| AMOUNT     | int(11)     | YES  |     | NULL    |       |
| BRANCH_ID  | varchar(5)  | YES  |     | NULL    |       |
| OPEN_DATE  | date        | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
```

5 rows in set (0.017 sec)

CREATE TABLE BORROW

MySQL> CREATE TABLE BORROW (LOAD_NO VARCHAR(5) PRIMARY KEY, CUST_ID VARCHAR(10), BRANCH_ID VARCHAR(5), AMOUNT INT);

Query OK, 0 rows affected (0.282 sec)

DESCRIBE:

MySQL> DESC BORROW;

```
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| LOAD_NO   | varchar(5)  | NO   | PRI | NULL    |       |
| CUST_ID   | varchar(10) | YES  |     | NULL    |       |
| BRANCH_ID | varchar(5)  | YES  |     | NULL    |       |
| AMOUNT    | int(11)     | YES  |     | NULL    |       |
```

```
+-----------+-------------+------+-----+---------+-------+
```

4 rows in set (0.017 sec)

Alert command -

Add address column in customer

```
MySQL> ALTER TABLE CUSTOMER ADD ADDRESS VARCHAR(30);
```

Query OK, 0 rows affected (0.088 sec)

Records: 0 Duplicates: 0 Warnings: 0

```
MySQL> DESC CUSTOMER;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| ID      | int(11)     | NO   | PRI | NULL    |       |
| NAME    | varchar(15) | NO   |     | NULL    |       |
| CITY    | varchar(10) | YES  |     | NULL    |       |
| ADDRESS | varchar(30) | YES  |     | NULL    |       |
+---------+-------------+------+-----+---------+-------+
```

4 rows in set (0.009 sec)

Modify column:

```
MySQL> ALTER TABLE BRANCH CHANGE BNAME BRANCH_NAME VARCHAR(10);
```

Query OK, 0 rows affected (0.902 sec)

Records: 0 Duplicates: 0 Warnings: 0

Modify col name

```
MySQL> ALTER TABLE CUSTOMER CHANGE ADDRESS NEW_ADDRESS VARCHAR(30);
```

Query OK, 0 rows affected (0.111 sec)

Records: 0 Duplicates: 0 Warnings: 0

DROP ADDRESS

```
MySQL> ALTER TABLE CUSTOMER DROP NEW_ADDRESS;
Query OK, 0 rows affected (0.571 sec)
Records: 0  Duplicates: 0  Warnings: 0

MySQL> DESC CUSTOMER;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| ID    | int(11)     | NO   | PRI | NULL    |       |
| NAME  | varchar(15) | NO   |     | NULL    |       |
| CITY  | varchar(10) | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
3 rows in set (0.010 sec)
```

RENAME BRANCH

```
MySQL> ALTER TABLE BRANCH RENAME TO BRANCH1;
Query OK, 0 rows affected (0.294 sec)
```

DCL ON TABLE

GRANT

```
MySQL> GRANT SELECT ON CUSTOMER TO USER;
Query OK, 0 rows affected (0.013 sec)
```

REVOKE

```
MySQL> REVOKE SELECT ON CUSTOMER FROM USER;
Query OK, 0 rows affected (0.002 sec)
```

TRUNCATE

```
MySQL> TRUNCATE CUSTOMER;
Query OK, 0 rows affected (0.509 sec)
```

DROP TABLE

MySQL> DROP TABLE CUSTOMER;

Query OK, 0 rows affected (0.229 sec)