

MYSQL Assignment No:1

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 database system
- To understand and use Data Definition Language and Data Control Language to write query for a database

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 the database
- ALTER – alters the structure of the database
- DROP – delete table from the database
- TRUNCATE – remove all records from a table, including all spaces allocated for the records are removed
- RENAME – rename a table

1. CREATE:

(a)CREATE DATABASE: You can create a MySQL database by using MySQL Command
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 particular database.

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 your database.

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_name column_type...);

Example:

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

```
CREATE TABLE cus_tbl(  
  
    roll_no INT NOT NULL ,  
  
    fname VARCHAR(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 into existing

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 existing relations.

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) ALTER TABLE..DROP...: This is used to remove any field of existing relations.

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 existing relations.

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 database object.

Syntax: RENAME TABLE old_relation_name TO new_relation_name;

Example: RENAME TABLE studentd TO studentd1;

4. TRUNCATE and DROP

Difference between Truncate & Drop:-

TRUNCATE: 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 TABLE student;

Data Control Language(DCL) : This is 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 to database.
- **Revoke :** Take back permissions from user.

Syntax: GRANT privilege_name ON object_name
TO {user_name };

Example : GRANT CREATE TABLE TO user1;

REVOKE privilege_name
ON object_name
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_id varchar(10) Primary Key, C_name Varchar(15) Not NULL, City varchar(10).

Table Name 2: **BRANCH**

Fields:

Branch_id Varchar(5) Primary Key, bname Varchar (15), City varchar(10).

Table Name 3: **DEPOSIT**

Fields:

Acc_no varchar(10) Primary Key, Cust_id Varchar(10) Not NULL, Amount
int, Branch_id Varchar(5), Open_date date.

Table Name 4: **BORROW**

Fields:

Loan_no Varchar(5) Primary Key, Cust_id Varchar (10), Branch_id varchar(5),
Amount int.

Perform the following command/operation on the above table:

- 1) Create a Database
- 2) Show Database
- 3) Use Database
- 4) Drop Database
- 5) Create tables and Describe that Tables
- 6) Alter Command
 - i) Add column address to Customer table
 - ii) Modify any column
 - iii) Rename column address to new_address
 - iv) Drop column address from Customer table
 - v) Rename table Branch to Branch1
- 6) Perform DCL Commands Grant and Revoke on Customer table
- 7) Truncate table
- 8) Drop table

Conclusion:-

MYSQL Assignment No:2

Aim :

Design and implement a database and apply at least 10 different DML queries for the following task. For a given input string display only those records which match the given pattern or a phrase in the search string. Make use of wild characters and LIKE operator for the same. Make use of Boolean and arithmetic operators wherever necessary.

Objective :

- To understand the different issues involved in the design and implementation of a database system
- To understand and use Data Manipulation Language to query to manage a database

Theory :

DATA MANIPULATION LANGUAGE (DML): The Data Manipulation Language (DML) is used to retrieve, insert and modify database information. These commands will be used by all database users during the routine operation of the database. Let's take a brief look at the basic DML commands:

- SELECT – retrieve data from the a database
- INSERT – insert data into a table
- UPDATE – updates existing data within a table
- DELETE – deletes all records from a table

1. INSERT INTO: This is used to add records into a relation. These are three type of INSERT INTO queries which are as

a) Inserting a single record

Syntax: INSERT INTO < relation/table name> (**field_1,field_2.....field_n**)VALUES
(data_1,data_2,.....data_n);

Example: INSERT INTO student(sno,sname,address)VALUES
(1,'Ravi','M.Tech','Palakol');

b) To insert multiple record

Here, we are going to insert record in the "cus_tbl" table of "customers" database. INSERT INTO student

```
(cus_id, cus_firstname, cus_surname)
VALUES
(5, 'Ajeet', 'Maurya'),
(6, 'Deepika', 'Chopra'),
(7, 'Vimal', 'Jaiswal');
table(column1,column2...)
VLUES (value1,
```

2. SELECT: This is used to Retrieve data from one or more tables.

a) **SELECT FROM:** To display all fields for all records.

Syntax : SELECT * FROM relation_name;

Example : SQL> select * from dept;

DEPTNO	DNAME	LOC
-----	-----	-----
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

b) **SELECT - FROM -WHERE:** This query is used to display a selected set of fields for a selected set of records of a relation.

Syntax: SELECT a set of fields FROM relation_name WHERE condition;

Example: SQL> select * FROM dept WHERE deptno<=20;

DEPTNO	DNAME	LOC
-----	-----	-----
10	ACCOUNTING	NEW YORK
	RESEARC	
20	H	DALLAS

c) SELECT - FROM -WHERE- LIKE

The LIKE operator is used in a WHERE clause to search for a specified pattern in a column.

There are two wildcards used in conjunction with the LIKE operator:

- % - The percent sign represents zero, one, or multiple characters
- _ - The underscore represents a single character

Syntax: SELECT column1, column2, ...
FROM table_name
WHERE columnN LIKE pattern;

Example:

WHERE CustomerName LIKE 'a%';

SELECT * FROM Customers

LIKE Operator	Description
WHERE CustomerName LIKE 'a%'	Finds any values that starts with "a"
WHERE CustomerName LIKE '%a'	Finds any values that ends with "a"
WHERE CustomerName LIKE '%or%'	Finds any values that have "or" in any position
WHERE CustomerName LIKE '_r%'	Finds any values that have "r" in the second position
WHERE CustomerName LIKE 'a_ _%'	Finds any values that starts with "a" and are at least 3 characters in length
WHERE ContactName LIKE 'a%o'	Finds any values that starts with "a" and ends with "o"

d) SELECT - DISTINCT

The SELECT DISTINCT statement is used to return only distinct (different) values. Inside a table, a column often contains many duplicate values; and sometimes you only want to list the

different (distinct) values. The SELECT DISTINCT statement is used to return only distinct (different) values.

Syntax: SELECT DISTINCT *column1, column2, ...*
FROM *table_name*;

Example: SELECT COUNT(DISTINCT Country) FROM Customers;

e) SELECT - BETWEEN

The BETWEEN operator selects values within a given range. The values can be numbers, text, or dates.

The BETWEEN operator is inclusive: begin and end values are included.

Syntax: SELECT *column_name(s)*
FROM *table_name*
WHERE *column_name* BETWEEN *value1* AND *value2*;

Example: SELECT * FROM Products
WHERE Price BETWEEN 10 AND 20;

f) WHERE with - AND LOGICAL Operator

The WHERE clause when used together with the AND logical operator, is only executed if ALL filter criteria specified are met.

```
SELECT * FROM `movies` WHERE `category_id` = 2 AND `year_released` = 2008;
```

g) WHERE with - OR LOGICAL Operator

The WHERE clause when used together with the OR operator, is only executed if any or the entire specified filter criteria is met.

The following script gets all the movies in either category 1 or category 2

```
SELECT * FROM `movies` WHERE `category_id` = 1 OR `category_id` = 2;
```

h) WHERE with - Arithmetic Operator

Operator	Description	Example
----------	-------------	---------

=	Checks if the values of the two operands are equal or not, if yes, then the condition becomes true.	(A = B) is not true.
!=	Checks if the values of the two operands are equal or not, if the values are not equal then the condition becomes true.	(A != B) is true.
>	Checks if the value of the left operand is greater than the value of the right operand, if yes, then the condition becomes true.	(A > B) is not true.
<	Checks if the value of the left operand is less than the value of the right operand, if yes then the condition becomes true.	(A < B) is true.
>=	Checks if the value of the left operand is greater than or equal to the value of the right operand, if yes, then the condition becomes true.	(A >= B) is not true.
<=	Checks if the value of the left operand is less than or equal to the value of the right operand, if yes, then the condition becomes true.	(A <= B) is true.

Example: SELECT agent_code, agent_name,
working_area, (commission*2)
FROM agents
WHERE (commission*2)>0.25;

3. UPDATE-SET-WHERE: This is used to update the content of a record in a relation.

Syntax: UPDATE relation name SET Field_name1=data,field_name2=data,

WHERE field_name=data;

Example: UPDATE student SET sname = 'kumar' WHERE sno=1;

4. DELETE-FROM: This is used to delete all the records of a relation but it will retain the structure of that relation.

a) **DELETE-FROM:** This is used to delete all the records of relation.

Syntax: DELETE FROM relation_name;

Example: DELETE FROM std;

b) **DELETE -FROM-WHERE:** This is used to delete a selected record from a relation.

Syntax: DELETE FROM relation_name WHERE condition;

Example: DELETE FROM student WHERE sno = 2;

LAB PRACTICE ASSIGNMENT:

Consider the following table structure for this assignment:

CUSTOMER(Cust_id, C_name, City)

BRANCH(Branch_id, bname, City)

DEPOSIT(Acc_no , Cust_id, Amount, Branch_id, Open_date)

BORROW(Loan_no, Cust_id, Branch_id, Amount)

Perform the following queries on the above table:

1. Insert minimum 10 rows on each table and display that data.
2. List Cust_id along with customer name.
3. List Cust_id of customers having amount greater than 10000.
4. List account date of customer 'Anil'.
5. List Cust_id of customers who have opened account after 01/01/2016.
6. List account no. and Cust_id of customers having amount between 40,000 and 80,000.
7. List customer name starting with 'S'.
8. List customer from depositor starting with '_a%'.
9. List customer name, account no and amount from deposit having exactly 5 characters in name.
10. List Cust_id, Loan no and Loan amount of borrowers.
11. List cust_id and C_name of depositors.
12. List all the customers who are depositors but not borrowers.
13. List all the customers who are both depositors and borrowers.
14. List all the customers along with their amount who are either borrowers.
15. List the cities of depositor having branch 'Cherthala'.
16. Update 10% interest to all depositors.
17. Update 10% to all depositors living in 'Ernakulam'.
18. Change living city of the 'Aroor' branch borrowers to Aroor.

19. Delete branches having deposit from Kollam.
20. Delete depositors of branches having number of customers between 1 and 3.
21. Delete depositors having deposit less than Rs500.

Conclusion:-