Tribhuvan University

Faculty of Humanities and Social Science

Prithvi Narayan Campus, Pokhara

Lab Report on DBMS (CACS255)

BCA 4th Sem

[2079 Batch]

Submitted By

Name: Deep Darshan Thapa

Roll No: 04

Batch: 2079

Submitted To

Devilal Timilsina

BCA Program, PNC

Index

S.N.	Objective	Date of submission	Remarks
1.			
2.			
3.			
4.			
5.			
6.			
7.			
8.			
9.			
10.			
11.			

LAB 1

Objective: To Design A Simple Database

THEORY

A database is a collection of structured data that can be accessed or stored in a computer system. It is usually managed through a Database Management System (DBMS), a software used to manage data. There are different types of databases, and relational databases, such as MySQL, are particularly popular for their ability to manage data through tables and relationships.

Demonstration

Purpose of creating database is defined first. “CREATE” command is used to create database.

```
Setting environment for using XAMPP for Windows.
user@DESKTOP-F7BSI9A c:\xampp
# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 8
Server version: 10.4.24-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create database dmslab;
Query OK, 1 row affected (0.004 sec)

MariaDB [(none)]> use dmslab;
Database changed
MariaDB [dmslab]>
```

Based on purpose the main entities and their attributes are identified

```
MariaDB [(none)]> use dmslab;
Database changed
MariaDB [dmslab]> create table students (stud_id int primary key,
    -> name varchar(50) not null,
    -> address varchar (70) not null,
    -> contact int (10) not null);
Query OK, 0 rows affected (0.028 sec)

MariaDB [dmslab]>
```

Data was added to each table using INSERT INTO statements

```
MariaDB [dmslab]> insert into students values  ('1', 'Deep','pokhara', '9876543210');
Query OK, 1 row affected, 1 warning (0.019 sec)

MariaDB [dmslab]>
```

To describe the structure of table “*Desc*” statement was used

```
MariaDB [dmslab]> desc students;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| stud_id | int(11)     | NO   | PRI | NULL    |       |
| name    | varchar(50) | NO   |     | NULL    |       |
| address | varchar(70) | NO   |     | NULL    |       |
| contact | int(10)     | NO   |     | NULL    |       |
+---------+-------------+------+-----+---------+-------+
4 rows in set (0.017 sec)
```

To display content from table “*select*” statement was used

```
MariaDB [dmslab]> select * from students;
+---------+------+---------+------------+
| stud_id | name | address | contact    |
+---------+------+---------+------------+
|       1 | Deep | pokhara | 2147483647 |
+---------+------+---------+------------+
1 row in set (0.002 sec)
```

To delete database “DROP DATABASE database_name; ” command is used

```
MariaDB [dmslab]> drop database dmslab;
Query OK, 1 row affected (0.015 sec)

MariaDB [(none)]>
```

Conclusion

In this lab the objective was to design a simple database. We successfully designed a simple database using basic operations.

LAB 2

Objective: To query and manipulate data in database

THEORY

Manipulating and querying data are essential aspects of working with relational databases. Data Manipulation Language (DML) in SQL includes commands to insert, update, delete, and retrieve data from tables, enabling users to manage and analyze data effectively.

Demonstration

Basic query

```
MariaDB [dmslab]> SELECT * FROM students;
+----+------+---------+------------+
| id | name | address | contact    |
+----+------+---------+------------+
|  1 | Deep | Pokhara | 2147483647 |
+----+------+---------+------------+
1 row in set (0.003 sec)
```

Conditional query

```
MariaDB [dmslab]> SELECT name, contact FROM students WHERE address = 'Pokhara';
+------+------------+
| name | contact    |
+------+------------+
| Deep | 2147483647 |
+------+------------+
1 row in set (0.003 sec)
```

To modify existing records

```
MariaDB [dmslab]> UPDATE students
    -> SET contact = 97650000000 WHERE id=1;
Query OK, 0 rows affected, 1 warning (0.006 sec)
Rows matched: 1  Changed: 0  Warnings: 1
```

To delete table

```
MariaDB [dmslab]> DELETE FROM students
    -> WHERE id = 1
    -> ;
Query OK, 1 row affected (0.004 sec)
```

Conclusion

In this lab the objective was to query and manipulate data in database. Querying and manipulating data are fundamental operations in database management, enabling efficient handling of large datasets. Mastery of these techniques is essential for effectively managing relational databases and driving data-driven decisions.

LAB 3

Objective: To Retrieve Data From Different Tables Using Joins

Theory

Data retrieval in MySQL is primarily done using the SELECT statement.Data retrieval from different tables using joins in MySQL allows you to combine rows from two or more tables based on a related column between them.

Demonstration

To demonstrate joins tables are created

```
MariaDB [dmslab]> INSERT INTO Employees (EmployeeID, Name, DepartmentID) VALUES
    -> (1, 'Deep', 101),
    -> (2, 'Deepa', 102),
    -> (3, 'Deepak', 103),
    -> (4, 'Manoj', NULL),
    -> (5, 'Ram', 104);
Query OK, 5 rows affected (0.008 sec)
Records: 5  Duplicates: 0  Warnings: 0

MariaDB [dmslab]>
```

```
MariaDB [dmslab]> CREATE TABLE Departments (
    ->     DepartmentID INT PRIMARY KEY,
    ->     DepartmentName VARCHAR(50) );
Query OK, 0 rows affected (0.017 sec)

MariaDB [dmslab]> INSERT INTO Departments (DepartmentID, DepartmentName) VALUES
    -> (101, 'Human Resources'),
    -> (102, 'Finance'),
    -> (103, 'Engineering'),
    -> (105, 'Sales');
Query OK, 4 rows affected (0.007 sec)
Records: 4  Duplicates: 0  Warnings: 0
```

Here are the main types of joins:

1. Inner Join

```
MariaDB [dmslab]>
MariaDB [dmslab]> SELECT Employees.Name AS EmployeeName, Departments.DepartmentName
    -> FROM Employees
    -> INNER JOIN Departments ON Employees.DepartmentID = Departments.DepartmentID;
+--------------+-----------------+
| EmployeeName | DepartmentName  |
+--------------+-----------------+
| Deep         | Human Resources |
| Deepa        | Finance         |
| Deepak       | Engineering     |
+--------------+-----------------+
3 rows in set (0.007 sec)
```

2. Left Join

```
MariaDB [dmslab]> Select Employees.Name AS EmployeeName, Departments.DepartmentName
    -> from Employees
    -> left join Departments on Employees.DepartmentID = Departments.DepartmentID;
+--------------+-----------------+
| EmployeeName | DepartmentName  |
+--------------+-----------------+
| Deep         | Human Resources |
| Deepa        | Finance         |
| Deepak       | Engineering     |
| Manoj        | NULL            |
| Ram          | NULL            |
+--------------+-----------------+
5 rows in set (0.003 sec)
```

3. Right Join

```
MariaDB [dmslab]> select Employees.Name AS EmployeeName, Departments.DepartmentName
    -> from Employees
    -> RIGHT JOIN Departments ON Employees.DepartmentID = Departments.DepartmentID;
+--------------+-----------------+
| EmployeeName | DepartmentName  |
+--------------+-----------------+
| Deep         | Human Resources |
| Deepa        | Finance         |
| Deepak       | Engineering     |
| NULL         | Sales           |
+--------------+-----------------+
4 rows in set (0.000 sec)

MariaDB [dmslab]>
```

4. Cross Join

```
MariaDB [dmslab]> SELECT Employees.Name AS EmployeeName, Departments.DepartmentName
    -> FROM Employees
    -> CROSS JOIN Departments;
+--------------+-----------------+
| EmployeeName | DepartmentName  |
+--------------+-----------------+
| Deep         | Human Resources |
| Deep         | Finance         |
| Deep         | Engineering     |
| Deep         | Sales           |
| Deepa        | Human Resources |
| Deepa        | Finance         |
| Deepa        | Engineering     |
| Deepa        | Sales           |
| Deepak       | Human Resources |
| Deepak       | Finance         |
| Deepak       | Engineering     |
| Deepak       | Sales           |
| Manoj        | Human Resources |
| Manoj        | Finance         |
| Manoj        | Engineering     |
| Manoj        | Sales           |
| Ram          | Human Resources |
| Ram          | Finance         |
| Ram          | Engineering     |
| Ram          | Sales           |
+--------------+-----------------+
20 rows in set (0.000 sec)
```

Conclusion

Using SQL joins, we can effectively retrieve and combine related data from multiple tables. Each type of join serves specific use cases, enhancing the flexibility and power of relational databases for complex queries.