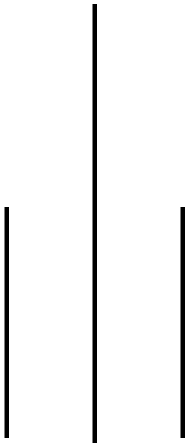
Lab Report
of

DBMS
Subject Code: CSC265



Submitted To

SOCH COLLEGE OF IT

(AFFILIATED TO TRIBHUVAN UNIVERSITY)

Ranipauwa, Pokhara – 11

Submitted by:

Nishan Parajuli

University Registration number:5-2-1179-10-2023

College Roll Number: 10

Program: Bachelor of Science in Computer Science and Information Technology (B.Sc.CSIT)

Semester: 4th semester

List of Exercises

S/N	Title of Experiment	Date	Remarks
1	**TO UNDERSTAND AND PRACTICE THE USE OF DATA DEFINITION LANGUAGE (DDL) COMMANDS.**	**JUNE 29**	
2	**: TO LEARN AND APPLY DATA MANIPULATION LANGUAGE (DML) COMMANDS.**	**JULY 9**	
3	**TO UNDERSTAND AND IMPLEMENT PRIMARY KEYS AND FOREIGN KEYS.**	**JULY 10**	
4	**: TO PRACTICE SQL SET OPERATIONS (UNION, INTERSECT, EXCEPT) AND THE USE OF THE ORDER BY CLAUSE.**	**JULY 11**	
5	**TO UNDERSTAND AND IMPLEMENT SQL CONCEPTS OF VIEWS, AGGREGATE FUNCTIONS, GROUPING,SORTING, SUBQUERIES, AND TABLE UPDATES AND JOINS.**	**AUGUST 28**	
6	**TO PRACTICE SQL QUERIES, JOINS, VIEWS, SUBQUERIES, AGGREGATE FUNCTIONS, AND GROUPING FOR EFFICIENT DATA RETRIEVAL AND MANIPULATION.**	**AUGUST 28**	
7	**TO UNDERSTAND AND APPLY DATABASE NORMALIZATION (1NF, 2NF, 3NF, BCNF) AND DEMONSTRATE THE USE OF SQL JOIN OPERATIONS ON NORMALIZED TABLES.**	**AUGUST 29**	
8	**TO LEARN HOW TO GRANT AND REVOKE PRIVILEGES FOR SPECIFIC OPERATIONS (LIKE SELECT, INSERT, UPDATE, DELETE) ON A PARTICULAR TABLE.**	**AUGUST 31**	
9	**TO UNDERSTAND HOW TO IMPLEMENT DATA INTEGRITY CONSTRAINTS USING ASSERTIONS AND TRIGGERS IN SQL.**	**AUGUST 31**	

Lab:1

Objective:--To understand and practice the use of Data Definition Language (DDL) commands.

Theory:

Data Definition Language (DDL)

DDL is a set of SQL commands used to define and manage the structure of database objects such as tables, schemas, indexes, and views. These commands deal with the creation, alteration, and deletion of database structures, but they do not manipulate the actual data stored in them. Execution of DDL commands results in changes to the database schema and is automatically committed.

Common DDL Commands:

- **CREATE** – Used to create new database objects like tables, views, or indexes.
- **ALTER** – Used to modify an existing database object, such as adding or dropping a column.
- **DROP** – Used to delete existing objects from the database permanently.
- **TRUNCATE** – Used to remove all rows from a table, while keeping its structure.
- **RENAME** – Used to change the name of a database object.

Task:

1. Create a database named 'college'.

->> CREATE DATABASE college;

2. Create a table 'course' with columns: course_id (INT), course_name (VARCHAR(50)), duration .

->> USE college;

CREATE TABLE course (course_id INT, course_name VARCHAR(50), duration INT);

3. Add a new column 'fees' (DECIMAL) to 'course'.
   ->> ALTER TABLE course ADD fees DECIMAL(10,2);

4. Modify the 'course_name' to accept up to 100 characters.
   ->> ALTER TABLE course MODIFY course_name VARCHAR(100);

5. Add a comment to 'duration' column as 'in months'.
   ->> ALTER TABLE course MODIFY duration INT COMMENT 'in months' ;

6. Show full details of the table using SHOW FULL COLUMNS.
   ->> SHOW FULL COLUMNS FROM course;

7. Create a table 'faculty' and then drop it.
   ->> CREATE TABLE faculty (id INT, faculty_name VARCHAR(50));
   DROP TABLE faculty;

8. Rename 'course' table to 'college_courses'.
   ->> RENAME TABLE course TO college_courses;

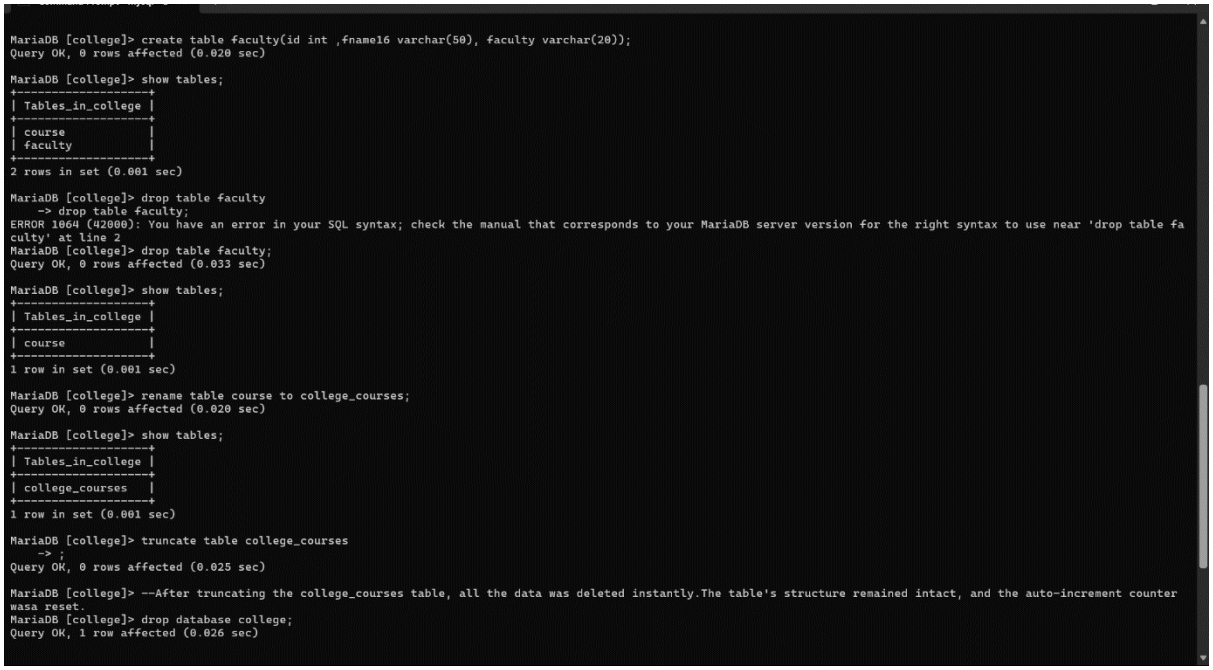
9. Truncate 'college_courses' table and describe what happened.
   ->> TRUNCATE TABLE college_courses;

10. Drop the database 'college'.

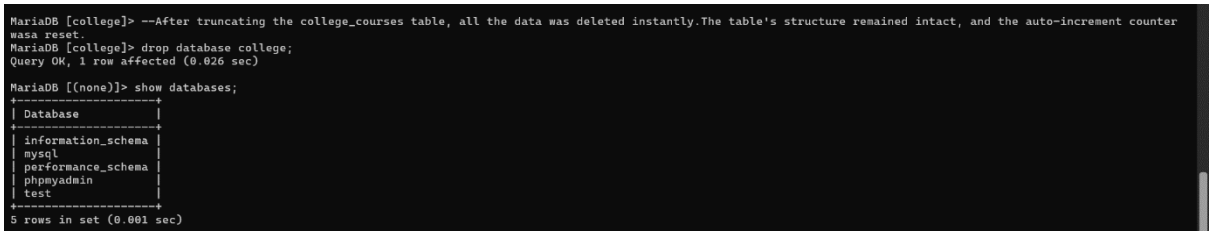
    ->> DROP DATABASE college;

```
Microsoft Windows [Version 10.0.22631.5549]
(c) Microsoft Corporation. All rights reserved.

C:\Users\HP>mysql -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 8
Server version: 10.4.32-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 college;
Query OK, 1 row affected (0.002 sec)

MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| college            |
| information_schema |
| mysql              |
| performance_schema |
| phpmyadmin         |
| test               |
+--------------------+
6 rows in set (0.040 sec)

MariaDB [(none)]> use college;
Database changed
MariaDB [college]> create table course(course_id int, course_fname16 varchar(50), duration int);
Query OK, 0 rows affected (0.013 sec)

MariaDB [college]> describe course;
+----------------+-------------+------+-----+---------+-------+
| Field          | Type        | Null | Key | Default | Extra |
+----------------+-------------+------+-----+---------+-------+
| course_id      | int(11)     | YES  |     | NULL    |       |
| course_fname16 | varchar(50) | YES  |     | NULL    |       |
| duration       | int(11)     | YES  |     | NULL    |       |
+----------------+-------------+------+-----+---------+-------+
3 rows in set (0.024 sec)

MariaDB [college]> alter table course add fees decimal;
Query OK, 0 rows affected (0.010 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [college]> alter table course modify fname VARCHAR(100);
```

```
MariaDB [college]> alter table course modify fname VARCHAR(100);
ERROR 1054 (42S22): Unknown column 'fname' in 'course'
MariaDB [college]> alter table course modify fname16 VARCHAR(100);
ERROR 1054 (42S22): Unknown column 'fname16' in 'course'
MariaDB [college]> describe course;
+----------------+---------------+------+-----+---------+-------+
| Field          | Type          | Null | Key | Default | Extra |
+----------------+---------------+------+-----+---------+-------+
| course_id      | int(11)       | YES  |     | NULL    |       |
| course_fname16 | varchar(50)   | YES  |     | NULL    |       |
| duration       | int(11)       | YES  |     | NULL    |       |
| fees           | decimal(10,0) | YES  |     | NULL    |       |
+----------------+---------------+------+-----+---------+-------+
4 rows in set (0.022 sec)

MariaDB [college]> alter table course modify course_fname16 varchar(100);
Query OK, 0 rows affected (0.044 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [college]> describe course;
+----------------+---------------+------+-----+---------+-------+
| Field          | Type          | Null | Key | Default | Extra |
+----------------+---------------+------+-----+---------+-------+
| course_id      | int(11)       | YES  |     | NULL    |       |
| course_fname16 | varchar(100)  | YES  |     | NULL    |       |
| duration       | int(11)       | YES  |     | NULL    |       |
| fees           | decimal(10,0) | YES  |     | NULL    |       |
+----------------+---------------+------+-----+---------+-------+
4 rows in set (0.020 sec)

MariaDB [college]> alter table course modify duration int comment 'in months';
Query OK, 0 rows affected (0.021 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [college]> show full columns from course;
+----------------+---------------+--------------------+------+-----+---------+-------+---------------------------------+-----------+
| Field          | Type          | Collation          | Null | Key | Default | Extra | Privileges                      | Comment   |
+----------------+---------------+--------------------+------+-----+---------+-------+---------------------------------+-----------+
| course_id      | int(11)       | NULL               | YES  |     | NULL    |       | select,insert,update,references |           |
| course_fname16 | varchar(100)  | utf8mb4_general_ci | YES  |     | NULL    |       | select,insert,update,references |           |
| duration       | int(11)       | NULL               | YES  |     | NULL    |       | select,insert,update,references | in months |
| fees           | decimal(10,0) | NULL               | YES  |     | NULL    |       | select,insert,update,references |           |
+----------------+---------------+--------------------+------+-----+---------+-------+---------------------------------+-----------+
4 rows in set (0.013 sec)

MariaDB [college]> create table faculty(id int ,fname16 varchar(50), faculty varchar(20));
Query OK, 0 rows affected (0.020 sec)
```



```
MariaDB [college]> create table faculty(id int ,fname16 varchar(50), faculty varchar(20));
Query OK, 0 rows affected (0.020 sec)

MariaDB [college]> show tables;
+-------------------+
| Tables_in_college |
+-------------------+
| course            |
| faculty           |
+-------------------+
2 rows in set (0.001 sec)

MariaDB [college]> drop table faculty
    -> drop table faculty;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near 'drop table fa
culty' at line 2
MariaDB [college]> drop table faculty;
Query OK, 0 rows affected (0.033 sec)

MariaDB [college]> show tables;
+-------------------+
| Tables_in_college |
+-------------------+
| course            |
+-------------------+
1 row in set (0.001 sec)

MariaDB [college]> rename table course to college_courses;
Query OK, 0 rows affected (0.020 sec)

MariaDB [college]> show tables;
+-------------------+
| Tables_in_college |
+-------------------+
| college_courses   |
+-------------------+
1 row in set (0.001 sec)

MariaDB [college]> truncate table college_courses
    -> ;
Query OK, 0 rows affected (0.025 sec)

MariaDB [college]> --After truncating the college_courses table, all the data was deleted instantly.The table's structure remained intact, and the auto-increment counter
wasa reset.
MariaDB [college]> drop database college;
Query OK, 1 row affected (0.026 sec)
```



```
MariaDB [college]> --After truncating the college_courses table, all the data was deleted instantly.The table's structure remained intact, and the auto-increment counter
wasa reset.
MariaDB [college]> drop database college;
Query OK, 1 row affected (0.026 sec)

MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| phpmyadmin         |
| test               |
+--------------------+
5 rows in set (0.001 sec)
```

Conclusion:

In this lab, we practiced different DDL commands such as CREATE, ALTER, DROP, and TRUNCATE. These commands helped us understand how to define, modify, and manage the structure of a database, which is essential for organizing data effectively.

Lab:2

Objective: To learn and apply Data Manipulation Language (DML) commands.

Theory:

Data Manipulation Language (DML)

DML is a set of SQL commands used to manage and manipulate the data stored in database objects like tables. Unlike DDL, which defines structure, DML deals with inserting, updating, deleting, and retrieving data. These commands allow users to work with the actual records inside the database.

Common DML Commands:

- **INSERT** – Adds new records into a table.
- **UPDATE** – Modifies existing records in a table.
- **DELETE** – Removes records from a table.
- **SELECT** – Retrieves data from one or more tables.

Task:

1: Create a database and tables.

```
--> CREATE DATABASE collegeDB;

    USE collegeDB;

CREATE TABLE student (   s_id INT PRIMARY KEY,   s_name VARCHAR(50),  age INT,

     city VARCHAR(30));

CREATE TABLE course ( course_id INT PRIMARY KEY, course_name VARCHAR(50) duration INT,

          fees DECIMAL(10,2));

CREATE TABLE faculty (f_id INT PRIMARY KEY, f_name VARCHAR(50),  department VARCHAR(30),

          salary DECIMAL(10,2));
```

2: Insert records into the tables.

```
->>     INSERT INTO student VALUES (101,'Adam',19,'Pokhara') (102,'Alex',20,'Kathmandu');

        INSERT INTO student VALUES (103,'Chris',18,'Butwal');

        INSERT INTO course VALUES (1,'Database Systems',6,12000.50);

        INSERT INTO course VALUES (2,'Computer Networks',5,10000.00);

        INSERT INTO course VALUES (3,'Operating Systems',4,8000.00);

        INSERT INTO faculty VALUES (1,'Dr. Smith','Computer Science',55000.00);
```

INSERT INTO faculty VALUES (2,'Dr. John','Information Tech',48000.00);

INSERT INTO faculty VALUES (3,'Dr. Anita','Mathematics',50000.00);

3: Retrieve records using SELECT

->> SELECT * FROM student;

SELECT s_name, city FROM student;

SELECT * FROM student WHERE age > 18;

SELECT * FROM course ORDER BY fees DESC;

SELECT AVG(fees) AS average_fees FROM course;

4: Update records.

-- Update with condition

UPDATE student SET age = 21 WHERE s_id = 102;

-- Update multiple columns

UPDATE faculty SET salary = 60000, department = 'AI' WHERE f_id = 1;

-- Update all rows (without WHERE)

UPDATE course SET duration = 12;

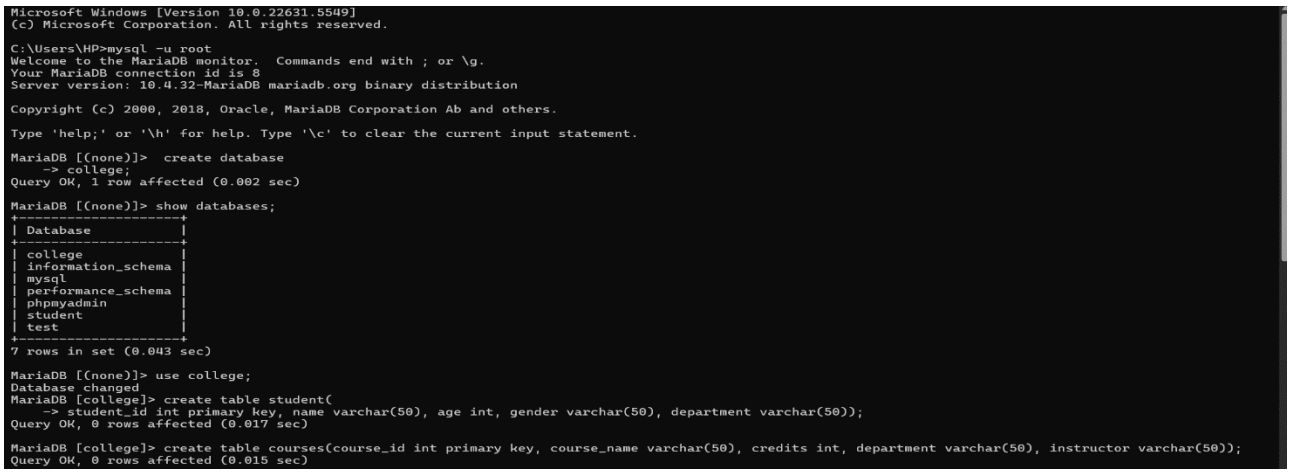
5: Delete records.

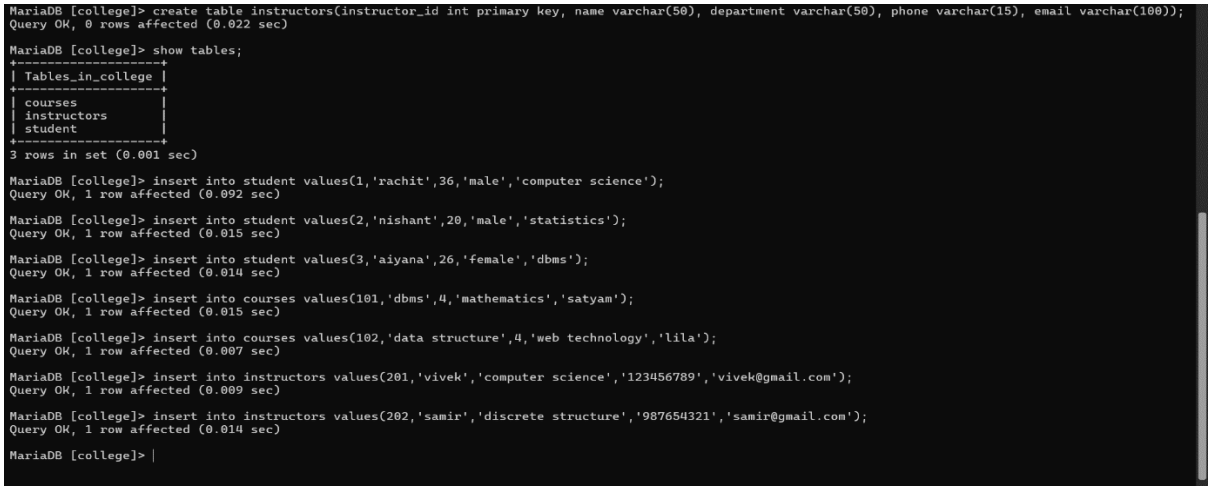
-- Delete with condition

DELETE FROM student WHERE s_id = 103;

-- Delete all rows (but keep table)

DELETE FROM course;





```
MariaDB [college]> create table instructors(instructor_id int primary key, name varchar(50), department varchar(50), phone varchar(15), email varchar(100));
Query OK, 0 rows affected (0.022 sec)

MariaDB [college]> show tables;
+------------------+
| Tables_in_college |
+------------------+
| courses          |
| instructors      |
| student          |
+------------------+
3 rows in set (0.001 sec)

MariaDB [college]> insert into student values(1,'rachit',36,'male','computer science');
Query OK, 1 row affected (0.092 sec)

MariaDB [college]> insert into student values(2,'nishant',20,'male','statistics');
Query OK, 1 row affected (0.015 sec)

MariaDB [college]> insert into student values(3,'aiyana',26,'female','dbms');
Query OK, 1 row affected (0.014 sec)

MariaDB [college]> insert into courses values(101,'dbms',4,'mathematics','satyam');
Query OK, 1 row affected (0.015 sec)

MariaDB [college]> insert into courses values(102,'data structure',4,'web technology','lila');
Query OK, 1 row affected (0.007 sec)

MariaDB [college]> insert into instructors values(201,'vivek','computer science','123456789','vivek@gmail.com');
Query OK, 1 row affected (0.009 sec)

MariaDB [college]> insert into instructors values(202,'samir','discrete structure','987654321','samir@gmail.com');
Query OK, 1 row affected (0.014 sec)

MariaDB [college]>
```

```
MariaDB [college]> select * from student;
+------------+---------+------+--------+------------------+
| student_id | name    | age  | gender | department       |
+------------+---------+------+--------+------------------+
|          1 | rachit  |   36 | male   | computer science |
|          2 | nishant |   20 | male   | statistics       |
|          3 | aiyana  |   26 | female | dbms             |
+------------+---------+------+--------+------------------+
3 rows in set (0.014 sec)

MariaDB [college]> select * from student where department = 'computer science';
+------------+--------+------+--------+------------------+
| student_id | name   | age  | gender | department       |
+------------+--------+------+--------+------------------+
|          1 | rachit |   36 | male   | computer science |
+------------+--------+------+--------+------------------+
1 row in set (0.001 sec)

MariaDB [college]> select * from courses order by credits desc;
+-----------+----------------+---------+-----------------+-------------+
| course_id | course_name    | credits | department      | instructor  |
+-----------+----------------+---------+-----------------+-------------+
|       101 | dbms           |       4 | mathematics     | satyam      |
|       102 | data structure |       4 | web technology  | lila        |
+-----------+----------------+---------+-----------------+-------------+
2 rows in set (0.009 sec)
```

```
MariaDB [college]> select name, age from student where age>20;
+--------+------+
| name   | age  |
+--------+------+
| rachit |   36 |
| aiyana |   26 |
+--------+------+
2 rows in set (0.001 sec)

MariaDB [college]> update student set department = 'ai' where student_id = 3;
Query OK, 1 row affected (0.006 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [college]> update student set age = age + 1;
Query OK, 3 rows affected (0.007 sec)
Rows matched: 3  Changed: 3  Warnings: 0

MariaDB [college]> delete from student where student_id = 2;
Query OK, 1 row affected (0.015 sec)

MariaDB [college]> delete from instructors;
Query OK, 2 rows affected (0.008 sec)
```

Conclusion:

In this lab, we practiced DML commands such as INSERT, UPDATE, DELETE, and SELECT. These commands helped us understand how to manipulate and retrieve data from database tables effectively.

Lab:3

Objective:- To understand and implement Primary Keys and Foreign Keys.

Theory:

- **Primary Key**: A column (or combination of columns) that uniquely identifies each row in a table. It does not allow NULL values or duplicates.
- **Foreign Key**: A column that establishes a relationship between two tables by referencing the primary key of another table. It enforces referential integrity.
- **NOT NULL**: Ensures that a column cannot contain NULL values.
- **Constraints**: Rules applied to columns to enforce data validity and consistency.

Cascade Actions:

- **ON DELETE CASCADE** → Deleting a parent record automatically deletes related child records.
- **ON UPDATE CASCADE** → Updating a primary key automatically updates the referencing foreign key in child tables.

Tasks:

1.Table creation with constraints.

```
CREATE TABLE Departments ( DeptID INT PRIMARY KEY,DeptName VARCHAR(100));

CREATE TABLE Employees (  EmpID INT PRIMARY KEY,  Name VARCHAR(100) NOT NULL,

  DeptID INT,  FOREIGN KEY (DeptID) REFERENCES Departments(DeptID));
```

2.Insert operation

```
-- Insert valid departments

  INSERT INTO Departments VALUES

  (1, 'HR'),                                          expected error at invalid insert

 (2, 'IT'),(3, 'Finance');

---insert valid department

  INSERT INTO Employees VALUES

 (101, 'Alice', 1),(102, 'Bob', 2);

 -- Invalid insert (DeptID 99 does not exist)

  INSERT INTO Employees VALUES (103, 'Charlie', 99);
```

ERROR 1452 (23000): Cannot add or update a child row:

a foreign key constraint fails (`collegeDB`.`Assignments`, CONSTRAINT `Assignments_ibfk_1` FOREIGN KEY (`EmpID`) REFERENCES `Employees` (`EmpID`))

3.**Select distinct.**

```
-- Show unique department IDs from Employees
SELECT DISTINCT DeptID FROM Employees;

-- Show unique department names from Departments

SELECT DISTINCT DeptName FROM Departments
```

4. Altering Constraints

```
-- Drop existing foreign key
ALTER TABLE Employees DROP FOREIGN KEY employees_ibfk_1;

-- Add constraint with CASCADE options

ALTER TABLE Employees
ADD CONSTRAINT fk_dept
FOREIGN KEY (DeptID) REFERENCES Departments(DeptID)
ON DELETE CASCADE,ON UPDATE CASCADE;
```

→ Test Cascade Behavior:

```
DELETE FROM Departments WHERE DeptID = 1;

UPDATE Departments SET DeptID = 4 WHERE DeptID = 2;
```

5.**Projects and assignment tables.**

```
-- Projects table
CREATE TABLE Projects (  ProjectID INT PRIMARY KEY,
   DeptID INT,FOREIGN KEY (DeptID) REFERENCES Departments(DeptID));
-- Assignments table (Composite PK)
CREATE TABLE Assignments (
   EmpID INT, ProjectID INT,HoursWorked INT, PRIMARY KEY (EmpID, ProjectID),
   FOREIGN KEY (EmpID) REFERENCES Employees(EmpID),
   FOREIGN KEY (ProjectID) REFERENCES Projects(ProjectID));
--Insert into Assignments
        -- Valid assignment
INSERT INTO Assignments VALUES (102, 1, 10);

        -- Invalid assignment (EmpID does not exist)
INSERT INTO Assignments VALUES (999, 1, 5);
```

➔ Expected Error:

ERROR 1452 (23000): Cannot add or update a child row:
a foreign key constraint fails (`collegeDB`.`Assignments`, CONSTRAINT `Assignments_ibfk_1` FOREIGN KEY (`EmpID`) REFERENCES `Employees` (`EmpID`))

```
Microsoft Windows [Version 10.0.22631.5549]
(c) Microsoft Corporation. All rights reserved.

C:\Users\HP>mysql -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 8
Server version: 10.4.32-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)]> show databases;
+--------------------+
| Database           |
+--------------------+
| college            |
| information_schema |
| mysql              |
| performance_schema |
| phpmyadmin         |
| student            |
| test               |
+--------------------+
7 rows in set (0.002 sec)

MariaDB [(none)]> create database lab3;
Query OK, 1 row affected (0.002 sec)

MariaDB [(none)]> use lab3;
Database changed
MariaDB [lab3]> CREATE TABLE Departments (
    ->  DeptID INT PRIMARY KEY,
    ->  DeptName VARCHAR(100)
    -> );
Query OK, 0 rows affected (0.025 sec)

MariaDB [lab3]> CREATE TABLE Employees (
    ->  EmpID INT PRIMARY KEY,
    ->  Name VARCHAR(100) NOT NULL,
    ->  DeptID INT,
    ->  FOREIGN KEY (DeptID) REFERENCES Departments(DeptID)
    -> );
Query OK, 0 rows affected (0.056 sec)
```

```
MariaDB [lab3]> INSERT INTO Departments VALUES
   -> (1, 'HR'),
   -> (2, 'IT'),
   -> (3, 'Finance');
Query OK, 3 rows affected (0.016 sec)
Records: 3  Duplicates: 0  Warnings: 0

MariaDB [lab3]> INSERT INTO Employees VALUES
   -> (101, 'Alice', 1),
   -> (102, 'Bob', 2);
Query OK, 2 rows affected (0.005 sec)
Records: 2  Duplicates: 0  Warnings: 0

MariaDB [lab3]> INSERT INTO Employees VALUES (103, 'Charlie', 99);
ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails (`lab3`.`employees`, CONSTRAINT `employees_ibfk_1` FOREIGN KEY (`DeptID
`) REFERENCES `departments` (`DeptID`))
MariaDB [lab3]> show Employees;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near
'Employees' at line 1
MariaDB [lab3]> show Employees
   -> ;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near
'Employees' at line 1
MariaDB [lab3]> describe Employees;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| EmpID  | int(11)     | NO   | PRI | NULL    |       |
| Name   | varchar(100) | NO   |     | NULL    |       |
| DeptID | int(11)     | YES  | MUL | NULL    |       |
+--------+-------------+------+-----+---------+-------+
3 rows in set (0.030 sec)

MariaDB [lab3]> describe departments;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| DeptID   | int(11)     | NO   | PRI | NULL    |       |
| DeptName | varchar(100) | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
2 rows in set (0.028 sec)
```

```
MariaDB [lab3]> select distinct Name from Employees;
+-------+
| Name  |
+-------+
| Alice |
| Bob   |
+-------+
2 rows in set (0.001 sec)

MariaDB [lab3]> describe employees;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| EmpID  | int(11)     | NO   | PRI | NULL    |       |
| Name   | varchar(100)| NO   |     | NULL    |       |
| DeptID | int(11)     | YES  | MUL | NULL    |       |
+--------+-------------+------+-----+---------+-------+
3 rows in set (0.034 sec)

MariaDB [lab3]> select * from employees;
+-------+-------+--------+
| EmpID | Name  | DeptID |
+-------+-------+--------+
|   101 | Alice |      1 |
|   102 | Bob   |      2 |
|   103 | Alice |      3 |
+-------+-------+--------+
3 rows in set (0.001 sec)

MariaDB [lab3]> ALTER TABLE Employees DROP FOREIGN KEY employees_ibfk_1
Query OK, 0 rows affected (0.021 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [lab3]> ALTER TABLE Employees
    -> ADD CONSTRAINT fk_dept
    -> FOREIGN KEY (DeptID) REFERENCES Departments(DeptID)
    -> ON DELETE CASCADE
    -> ON UPDATE CASCADE;
Query OK, 3 rows affected (0.083 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

```
MariaDB [lab3]> DELETE FROM Departments WHERE DeptID = 1;
Query OK, 1 row affected (0.008 sec)

MariaDB [lab3]> select * from employees;
+-------+-------+--------+
| EmpID | Name  | DeptID |
+-------+-------+--------+
|   102 | Bob   |      2 |
|   103 | Alice |      3 |
+-------+-------+--------+
2 rows in set (0.000 sec)

MariaDB [lab3]> UPDATE Departments SET DeptID = 4 WHERE DeptID = 2;
Query OK, 1 row affected (0.006 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [lab3]> select * from employees;
+-------+-------+--------+
| EmpID | Name  | DeptID |
+-------+-------+--------+
|   102 | Bob   |      4 |
|   103 | Alice |      3 |
+-------+-------+--------+
2 rows in set (0.001 sec)

MariaDB [lab3]> CREATE TABLE Projects (
    ->  ProjectID INT PRIMARY KEY,
    ->  DeptID INT,
    ->  FOREIGN KEY (DeptID) REFERENCES Departments(DeptID)
    -> );
Query OK, 0 rows affected (0.047 sec)

MariaDB [lab3]> CREATE TABLE Assignments (
    ->  EmpID INT,
    ->  ProjectID INT,
    ->  HoursWorked INT,
    ->  PRIMARY KEY (EmpID, ProjectID),
    ->  FOREIGN KEY (EmpID) REFERENCES Employees(EmpID),
    ->  FOREIGN KEY (ProjectID) REFERENCES Projects(ProjectID)
    -> );
Query OK, 0 rows affected (0.046 sec)
```

```
MariaDB [lab3]> INSERT INTO Assignments VALUES (102, 1, 10);
ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails (`lab3`.`assignments`, CONSTRAINT `assignments_ibfk_2` FOREIGN KEY (`Pr
ojectID`) REFERENCES `projects` (`ProjectID`))
MariaDB [lab3]> show assignments;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near
'assignments' at line 1
MariaDB [lab3]> describe assignments;
+-------------+---------+------+-----+---------+-------+
| Field       | Type    | Null | Key | Default | Extra |
+-------------+---------+------+-----+---------+-------+
| EmpID       | int(11) | NO   | PRI | NULL    |       |
| ProjectID   | int(11) | NO   | PRI | NULL    |       |
| HoursWorked | int(11) | YES  |     | NULL    |       |
+-------------+---------+------+-----+---------+-------+
3 rows in set (0.029 sec)
```


```
MariaDB [lab3]> INSERT INTO Assignments VALUES (102, 1, 10);
ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails (`lab3`.`assignments`, CONSTRAINT `assignments_ibfk_2` FOREIGN KEY (`Pr
ojectID`) REFERENCES `projects` (`ProjectID`))
MariaDB [lab3]> show assignments;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near
'assignments' at line 1
MariaDB [lab3]> describe assignments;
+-------------+---------+------+-----+---------+-------+
| Field       | Type    | Null | Key | Default | Extra |
+-------------+---------+------+-----+---------+-------+
| EmpID       | int(11) | NO   | PRI | NULL    |       |
| ProjectID   | int(11) | NO   | PRI | NULL    |       |
| HoursWorked | int(11) | YES  |     | NULL    |       |
+-------------+---------+------+-----+---------+-------+
3 rows in set (0.029 sec)
```


Conclusion:

In this lab, we practiced the use of keys with NOT NULL constraints to maintain uniqueness and integrity of data. We also implemented ON DELETE CASCADE and ON UPDATE CASCADE to handle changes in parent-child table relationships automatically, ensuring referential integrity across tables.

Lab:4

Objective: To practice SQL set operations (UNION, INTERSECT, EXCEPT) and the use of the Order by clause.

Theory:

UPDATE

- The **UPDATE** command is a DML statement used to modify existing records in a table. It allows changing one or more column values based on a condition using the WHERE clause. Without WHERE, all rows are updated.

INTERSECT

- The **INTERSECT** operator is used in SQL to return the common rows from two SELECT queries. It only retrieves distinct records that appear in the result sets of both queries.

EXCEPT

- The **EXCEPT** operator returns all distinct rows from the first SELECT query that are not present in the second SELECT query. It is useful for finding differences between two result sets.

Task:

Table 1: depositor(account)

cusname	accno
John	101
Alice	102
Bob	103
John	104
Emily	105
John	106

Table 2: borrower(loan)

cusname	loanno
John	201
Alice	202
Derek	203
John	204
John	205

Table 3: loan

loanno	amount	branchname
201	3000	Pokhara
202	2000	Pokhara
203	3000	Kathmandu
204	1000	Pokhara
205	3000	Biratnagar

Q1: Find all customers who either have an account or a loan

```
SELECT cusname FROM depositor UNION SELECT cusname FROM borrower;
```

Q2: Display all customers including duplicate.

```
SELECT cusname FROM depositor UNION ALL SELECT cusname FROM borrower;
```

Q3: Find customers who have both a loan and an account.

```
SELECT DISTINCT d.cusname FROM depositor d JOIN borrower b ON d.cusname = b.cusname;
```

Q4: Include repeated names based on number of common entries.

```
SELECT d.cusname FROM depositor d JOIN borrower b ON d.cusname = b.cusname;
```

Q5: Find customers who have an account but no loan.

SELECT DISTINCT d.cusname FROM depositor dWHERE d.cusname NOT IN (SELECT b.cusname FROM borrower b);

Q6: Reflect duplicates for customers with more accounts than loans.

SELECT d.cusname FROM depositor d LEFT JOIN borrower b ON d.cusname = b.cusname

WHERE b.cusname IS NULL;

Q7: Retrieve customers from borrower at branch 'Pokhara', sorted alphabetically.

SELECT b.cusname FROM borrower b JOIN loan l ON b.loanno = l.loanno

WHERE l.branchname = 'Pokhara' ,ORDER BY b.cusname ASC;

Q8: List all loans, sorted by amount (DESC), and for ties sort by loanno (ASC).

SELECT * FROM loan ORDER BY amount DESC, loanno ASC;

```
Microsoft Windows [Version 10.0.22631.5624]
(c) Microsoft Corporation. All rights reserved.

C:\Users\HP>mysql -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 8
Server version: 10.4.32-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 samir
    -> ;
Query OK, 1 row affected (0.002 sec)

MariaDB [(none)]> use samir;
Database changed
MariaDB [samir]> CREATE TABLE depositor (
    ->     cusname VARCHAR(50),
    ->     accno INT
    -> );
Query OK, 0 rows affected (0.014 sec)

MariaDB [samir]>
MariaDB [samir]> INSERT INTO depositor (cusname, accno) VALUES
    -> ('John', 101),
    -> ('Alice', 102),
    -> ('Bob', 103),
    -> ('John', 104),
    -> ('Emily', 105),
    -> ('John', 106);
Query OK, 6 rows affected (0.055 sec)
Records: 6  Duplicates: 0  Warnings: 0

MariaDB [samir]> CREATE TABLE borrower (
    ->     cusname VARCHAR(50),
    ->     loanno INT
    -> );
Query OK, 0 rows affected (0.018 sec)

MariaDB [samir]>
MariaDB [samir]> INSERT INTO borrower (cusname, loanno) VALUES
    -> ('John', 201),
```

```
MariaDB [samir]> INSERT INTO borrower (cusname, loanno) VALUES
    -> ('John', 201),
    -> ('Alice', 202),
    -> ('Derek', 203),
    -> ('John', 204),
    -> ('John', 205);
Query OK, 5 rows affected (0.003 sec)
Records: 5  Duplicates: 0  Warnings: 0

MariaDB [samir]> CREATE TABLE loan (
    ->     loanno INT,
    ->     amount INT,
    ->     branchname VARCHAR(50)
    -> );
Query OK, 0 rows affected (0.025 sec)

MariaDB [samir]>
MariaDB [samir]> INSERT INTO loan (loanno, amount, branchname) VALUES
    -> (201, 3000, 'Pokhara'),
    -> (202, 2000, 'Pokhara'),
    -> (203, 3000, 'Kathmandu'),
    -> (204, 1000, 'Pokhara'),
    -> (205, 3000, 'Biratnagar');
Query OK, 5 rows affected (0.004 sec)
Records: 5  Duplicates: 0  Warnings: 0

MariaDB [samir]> SELECT cusname FROM depositor
    -> UNION
    -> SELECT cusname FROM borrower;
+---------+
| cusname |
+---------+
| John    |
| Alice   |
| Bob     |
| Emily   |
| Derek   |
+---------+
5 rows in set (0.001 sec)

MariaDB [samir]> SELECT cusname FROM depositor
    -> UNION ALL
    -> SELECT cusname FROM borrower;
+---------+
```

```
    -> SELECT cusname FROM borrower;
+---------+
| cusname |
+---------+
| John    |
| Alice   |
| Bob     |
| John    |
| Emily   |
| John    |
| John    |
| Alice   |
| Derek   |
| John    |
| John    |
+---------+
11 rows in set (0.001 sec)

MariaDB [samir]> SELECT cusname FROM depositor
    -> INTERSECT
    -> SELECT cusname FROM borrower;
+---------+
| cusname |
+---------+
| John    |
| Alice   |
+---------+
2 rows in set (0.001 sec)

MariaDB [samir]> SELECT cusname FROM depositor
    -> INTERSECT ALL
    -> SELECT cusname FROM borrower;
ERROR 1064 (42000): You have an error in your SQL syntax; c
'ALL
SELECT cusname FROM borrower' at line 2
MariaDB [samir]> SELECT cusname FROM depositor
    -> INTERSECT ALL
    -> SELECT cusname FROM borrower;
ERROR 1064 (42000): You have an error in your SQL syntax; c
'ALL
SELECT cusname FROM borrower' at line 2
MariaDB [samir]> SELECT d.cusname
    -> FROM depositor d
    -> JOIN borrower b ON d.cusname = b.cusname;
```

```
+---------+
| cusname |
+---------+
| John    |
| John    |
| John    |
| Alice   |
| John    |
| John    |
| John    |
| John    |
| John    |
| John    |
+---------+
10 rows in set (0.011 sec)

MariaDB [samir]> SELECT cusname FROM depositor
    -> EXCEPT
    -> SELECT cusname FROM borrower;
+---------+
| cusname |
+---------+
| Bob     |
| Emily   |
+---------+
2 rows in set (0.001 sec)

MariaDB [samir]> SELECT cusname FROM depositor
    -> EXCEPT ALL
    -> SELECT cusname FROM borrower;
ERROR 1064 (42000): You have an error in your SQL s
'ALL
SELECT cusname FROM borrower' at line 2
MariaDB [samir]> SELECT b.cusname
    -> FROM borrower b
    -> JOIN loan l ON b.loanno = l.loanno
    -> WHERE l.branchname = 'Pokhara'
    -> ORDER BY b.cusname ASC;
+---------+
| cusname |
+---------+
| Alice   |
| John    |
| John    |
```

```
+---------+
3 rows in set (0.006 sec)

MariaDB [samir]> SELECT * FROM loan
    -> ORDER BY amount DESC, loanno ASC;
+--------+--------+------------+
| loanno | amount | branchname |
+--------+--------+------------+
|    201 |   3000 | Pokhara    |
|    203 |   3000 | Kathmandu  |
|    205 |   3000 | Biratnagar |
|    202 |   2000 | Pokhara    |
|    204 |   1000 | Pokhara    |
+--------+--------+------------+
5 rows in set (0.001 sec)

MariaDB [samir]> SELECT d.cusname
    -> FROM depositor d
    -> LEFT JOIN borrower b ON d.cusname = b.cusname
    -> WHERE d.cusname NOT IN (
    ->     SELECT cusname FROM borrower
    -> )
    -> UNION ALL
    -> SELECT d.cusname
    -> FROM depositor d
    -> JOIN borrower b ON d.cusname = b.cusname
    -> GROUP BY d.cusname
    -> HAVING COUNT(d.cusname) > COUNT(b.cusname);
+---------+
| cusname |
+---------+
| Bob     |
| Emily   |
+---------+
2 rows in set (0.002 sec)

MariaDB [samir]> --except all isn't supported by my sql
MariaDB [samir]> --turn around
MariaDB [samir]> SELECT d.cusname
    -> FROM depositor d
    -> LEFT JOIN borrower b ON d.cusname = b.cusname
    -> WHERE d.cusname NOT IN (
    ->     SELECT cusname FROM borrower
    -> )
```

```
+---------+
| cusname |
+---------+
| Bob     |
| Emily   |
+---------+
2 rows in set (0.002 sec)

MariaDB [samir]> --except all isn't supported by my sql
MariaDB [samir]> --turn around
MariaDB [samir]> SELECT d.cusname
    -> FROM depositor d
    -> LEFT JOIN borrower b ON d.cusname = b.cusname
    -> WHERE d.cusname NOT IN (
    ->     SELECT cusname FROM borrower
    -> )
    -> UNION ALL
    -> SELECT d.cusname
    -> FROM depositor d
    -> JOIN borrower b ON d.cusname = b.cusname
    -> GROUP BY d.cusname
    -> HAVING COUNT(d.cusname) > COUNT(b.cusname);
+---------+
| cusname |
+---------+
| Bob     |
| Emily   |
+---------+
2 rows in set (0.002 sec)

MariaDB [samir]>
```

Lab:5

Objective:-To understand and implement SQL concepts of views, aggregate functions, grouping,sorting, subqueries, and table updates and joins.

Theory:

- **Views**
- A **View** is a virtual table created from the result of a SQL query. It does not store data physically but provides a simplified way to access and manipulate complex queries.
- Aggregate Functions
- **Aggregate functions** perform calculations on a set of values and return a single value. Examples include SUM(), AVG(), COUNT(), MIN(), and MAX().
- Grouping

- The **GROUP BY** clause groups rows based on one or more columns so that aggregate functions can be applied to each group of data.
- Sorting

- The **ORDER BY** clause is used to sort the result set in ascending (ASC) or descending (DESC) order based on one or more columns.
- Subqueries

- A **Subquery** is a query nested inside another SQL query. It can be used in SELECT, INSERT, UPDATE, or DELETE statements to provide results dynamically.
- Table Updates

- The **UPDATE** command modifies existing records in a table, either for all rows or specific rows using a WHERE condition.
- Joins
- A **Join** combines rows from two or more tables based on a related column. Common types are **INNER JOIN**, **LEFT JOIN**, **RIGHT JOIN**, and **FULL JOIN**

Task:

1.**Create Table and Insert Data**.

```
CREATE TABLE customers (ID INT PRIMARY KEY,  NAME VARCHAR(50), AGE INT,

ADDRESS VARCHAR(50),SALARY DECIMAL(10,2));

INSERT INTO customers VALUES

(1, 'Ramesh',  32, 'Ahmedabad', 2000.00),(2, 'Khilan',  25, 'Delhi',    1500.00),

(3, 'Kaushik',  23, 'Kota',     2000.00),(4, 'Chaitali', 25, 'Mumbai',   6500.00),

(5, 'Hardik',  27, 'Bhopal',   8500.00),(6, 'Komal',   22, 'MP',       4500.00),

        (7, 'Muffy',   24, 'Indore',  10000.00);
```

2. Create View.

CREATE VIEW customers_view AS SELECT NAME, AGE FROM customers;

3. Select All from View.

SELECT * FROM customers_view;

4. Select Names of People Aged 25 from View.

SELECT NAME FROM customers_view WHERE AGE = 25;

5. Drop View.

DROP VIEW customers_view;

6. Aggregate function.

SELECT COUNT(*) FROM customers WHERE AGE = 25; SELECT COUNT(*) FROM customers;

SELECT MIN(AGE) FROM customers; SELECT MAX(AGE) FROM customers;

SELECT AVG(SALARY) FROM customers;

7. Update Table Data (Change Names)

UPDATE customers SET NAME = 'Ramesh' WHERE ID = 2; UPDATE customers SET NAME = 'Kaushik' WHERE ID = 4;

8.Group by sum.

SELECT NAME, SUM(SALARY) FROM customers GROUP BY NAME;

9. Sort Records.

SELECT * FROM customers ORDER BY NAME DESC, AGE ASC;

10. Select with IN.

SELECT * FROM customers WHERE ADDRESS IN ('Kota', 'Mumbai', 'Indore');

11. Subquery (Salary > 4500)

SELECT * FROM customers WHERE ID IN (SELECT ID FROM customers WHERE SALARY > 4500);

12. Update with Subquery.

UPDATE customersSET SALARY = SALARY * 0.25 WHERE AGE IN (SELECT AGE FROM (SELECT AGE FROM customers WHERE AGE > 27) AS t);

Joins :

1. inner join.

SELECT t1.OrderID, t0.ProductID, t0.Name, t0.UnitPrice, t1.Quantity, t1.Price FROM tblProduct AS t0 INNER JOIN tblOrder AS t1 ON t0.ProductID = t1.ProductID ORDER BY t1.OrderID;

2. LEFT OUTER JOIN.

SELECT t1.OrderID AS OrderID, t0.ProductID, t0.Name, t0.UnitPrice, t1.Quantity AS Quantity, t1.Price AS Price FROM tblProduct AS t0 LEFT OUTER JOIN tblOrder AS t1 ON t0.ProductID = t1.ProductIDORDER BY t0.ProductID;

3. RIGHT OUTER JOIN.

SELECT t1.OrderID AS OrderID, t0.ProductID, t0.Name, t0.UnitPrice, t1.Quantity AS Quantity, t1.Price AS Price FROM tblProduct AS t0 RIGHT OUTER JOIN tblOrder AS t1 ON t0.ProductID = t1.ProductIDORDER BY t0.ProductID;

4. CROSS JOIN.

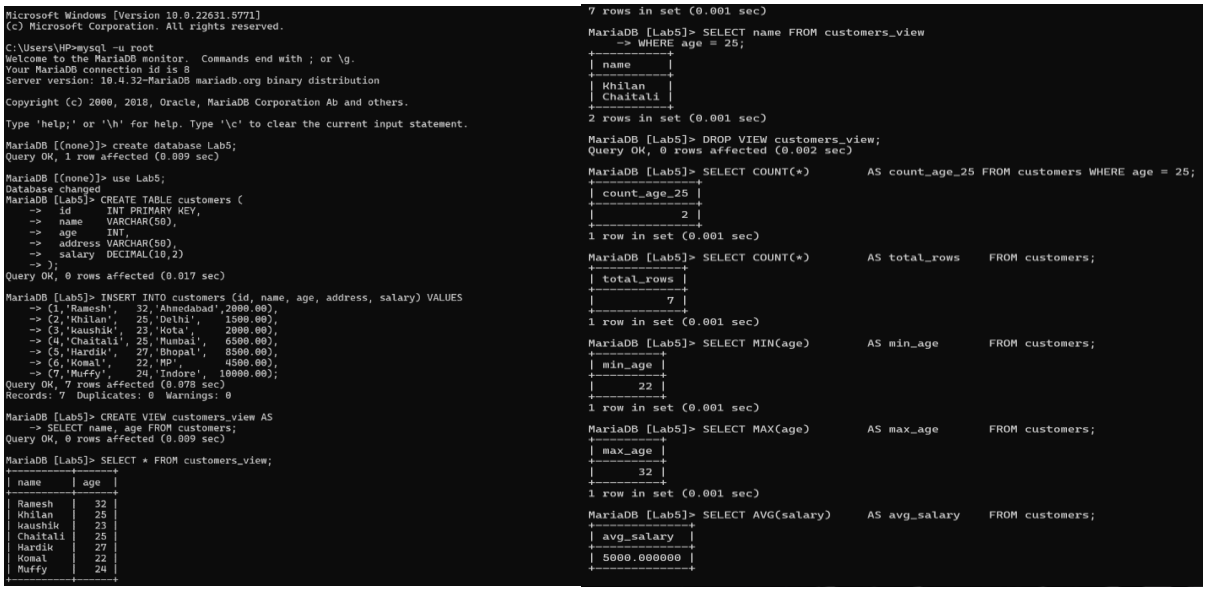
SELECT t1.OrderID, t0.ProductID, t0.Name, t0.UnitPrice, t1.Quantity, t1.Price FROM tblProduct AS t0, tblOrder AS t1 ORDER BY t0.ProductID;

5. INNER JOIN with 3 tables.

SELECT t1.OrderID, t0.ProductID, t0.Name, t0.UnitPrice, t1.Quantity, t1.Price, t2.Name AS Customer FROM tblProduct AS t0 INNER JOIN tblOrder AS t1 ON t0.ProductID = t1.ProductID INNER JOIN tblCustomer AS t2 ON t1.CustomerID = t2.CustID ORDER BY t1.OrderID;

6. INNER JOIN on multiple conditions.

SELECT t1.OrderID, t0.ProductID, t0.Name, t0.UnitPrice, t1.Quantity, t1.Price, t2.Name AS Customer FROM tblProduct AS t0 INNER JOIN tblOrder AS t1 ON t0.ProductID = t1.ProductID INNER JOIN tblCustomer AS t2 ON t1.CustomerID = t2.CustID AND t1.ContactNo = t2.ContactNo ORDER BY t1.OrderID;



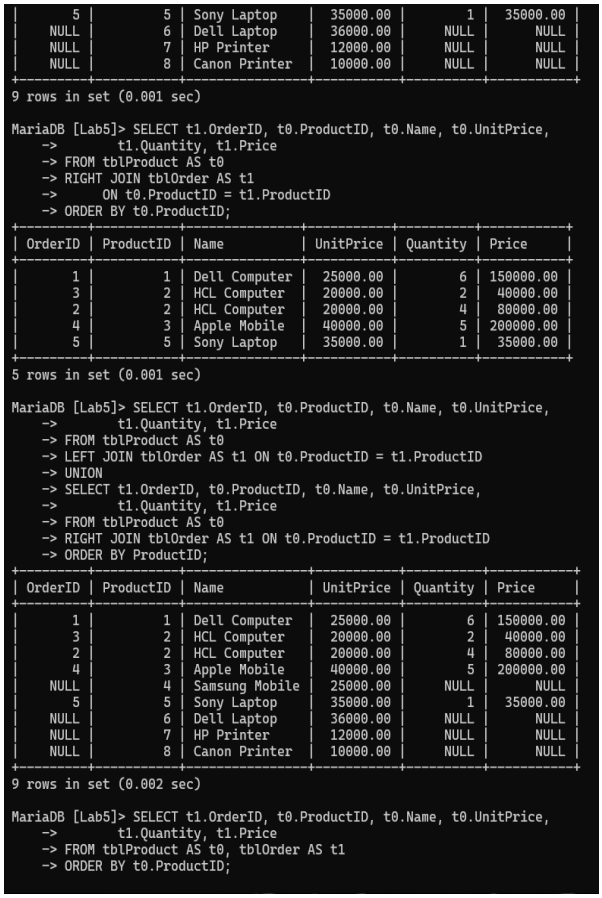
```
Microsoft Windows [Version 10.0.22631.5771]
(c) Microsoft Corporation. All rights reserved.

C:\Users\HP>mysql -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 8
Server version: 10.4.32-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 Lab5;
Query OK, 1 row affected (0.009 sec)

MariaDB [(none)]> use Lab5;
Database changed
MariaDB [Lab5]> CREATE TABLE customers (
    ->   id      INT PRIMARY KEY,
    ->   name    VARCHAR(50),
    ->   age     INT,
    ->   address VARCHAR(50),
    ->   salary  DECIMAL(10,2)
    -> );
Query OK, 0 rows affected (0.017 sec)

MariaDB [Lab5]> INSERT INTO customers (id, name, age, address, salary) VALUES
    -> (1,'Ramesh',   32,'Ahmedabad',2000.00),
    -> (2,'Khilan',   25,'Delhi',    1500.00),
    -> (3,'kaushik',  23,'Kota',     2000.00),
    -> (4,'Chaitali', 25,'Mumbai',   6500.00),
    -> (5,'Hardik',   27,'Bhopal',   8500.00),
    -> (6,'Komal',    22,'MP',       4500.00),
    -> (7,'Muffy',    24,'Indore',  10000.00);
Query OK, 7 rows affected (0.078 sec)
Records: 7  Duplicates: 0  Warnings: 0

MariaDB [Lab5]> CREATE VIEW customers_view AS
    -> SELECT name, age FROM customers;
Query OK, 0 rows affected (0.009 sec)

MariaDB [Lab5]> SELECT * FROM customers_view;
+----------+------+
| name     | age  |
+----------+------+
| Ramesh   |   32 |
| Khilan   |   25 |
| kaushik  |   23 |
| Chaitali |   25 |
| Hardik   |   27 |
| Komal    |   22 |
| Muffy    |   24 |
+----------+------+
7 rows in set (0.001 sec)

MariaDB [Lab5]> SELECT name FROM customers_view
    -> WHERE age = 25;
+----------+
| name     |
+----------+
| Khilan   |
| Chaitali |
+----------+
2 rows in set (0.001 sec)

MariaDB [Lab5]> DROP VIEW customers_view;
Query OK, 0 rows affected (0.002 sec)

MariaDB [Lab5]> SELECT COUNT(*)        AS count_age_25 FROM customers WHERE age = 25;
+--------------+
| count_age_25 |
+--------------+
|            2 |
+--------------+
1 row in set (0.001 sec)

MariaDB [Lab5]> SELECT COUNT(*)        AS total_rows    FROM customers;
+--------------+
| total_rows   |
+--------------+
|            7 |
+--------------+
1 row in set (0.001 sec)

MariaDB [Lab5]> SELECT MIN(age)        AS min_age       FROM customers;
+----------+
| min_age  |
+----------+
|       22 |
+----------+
1 row in set (0.001 sec)

MariaDB [Lab5]> SELECT MAX(age)        AS max_age       FROM customers;
+----------+
| max_age  |
+----------+
|       32 |
+----------+
1 row in set (0.001 sec)

MariaDB [Lab5]> SELECT AVG(salary)     AS avg_salary    FROM customers;
+--------------+
| avg_salary   |
+--------------+
| 5000.000000  |
+--------------+
```

```
+--------------+
1 row in set (0.000 sec)

MariaDB [Lab5]> SELECT SUM(salary)     AS total_salary  FROM customers;
+--------------+
| total_salary |
+--------------+
|     35000.00 |
+--------------+
1 row in set (0.001 sec)

MariaDB [Lab5]> UPDATE customers SET name = 'Ramesh' WHERE id = 2;
Query OK, 1 row affected (0.007 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [Lab5]> UPDATE customers SET name = 'kaushik' WHERE id = 4;
Query OK, 1 row affected (0.010 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [Lab5]> SELECT * FROM customers ORDER BY id;
+----+---------+------+-----------+----------+
| id | name    | age  | address   | salary   |
+----+---------+------+-----------+----------+
|  1 | Ramesh  |   32 | Ahmedabad |  2000.00 |
|  2 | Ramesh  |   25 | Delhi     |  1500.00 |
|  3 | kaushik |   23 | Kota      |  2000.00 |
|  4 | kaushik |   25 | Mumbai    |  6500.00 |
|  5 | Hardik  |   27 | Bhopal    |  8500.00 |
|  6 | Komal   |   22 | MP        |  4500.00 |
|  7 | Muffy   |   24 | Indore    | 10000.00 |
+----+---------+------+-----------+----------+
7 rows in set (0.001 sec)

MariaDB [Lab5]> SELECT name, SUM(salary) AS total_salary
    -> FROM customers
    -> GROUP BY name;
+---------+--------------+
| name    | total_salary |
+---------+--------------+
| Hardik  |      8500.00 |
| kaushik |      8500.00 |
| Komal   |      4500.00 |
| Muffy   |     10000.00 |
| Ramesh  |      3500.00 |
+---------+--------------+
5 rows in set (0.001 sec)

MariaDB [Lab5]> SELECT *
    -> FROM customers
    -> ORDER BY name DESC, age ASC;
+----+---------+------+-----------+----------+
| id | name    | age  | address   | salary   |
+----+---------+------+-----------+----------+
```

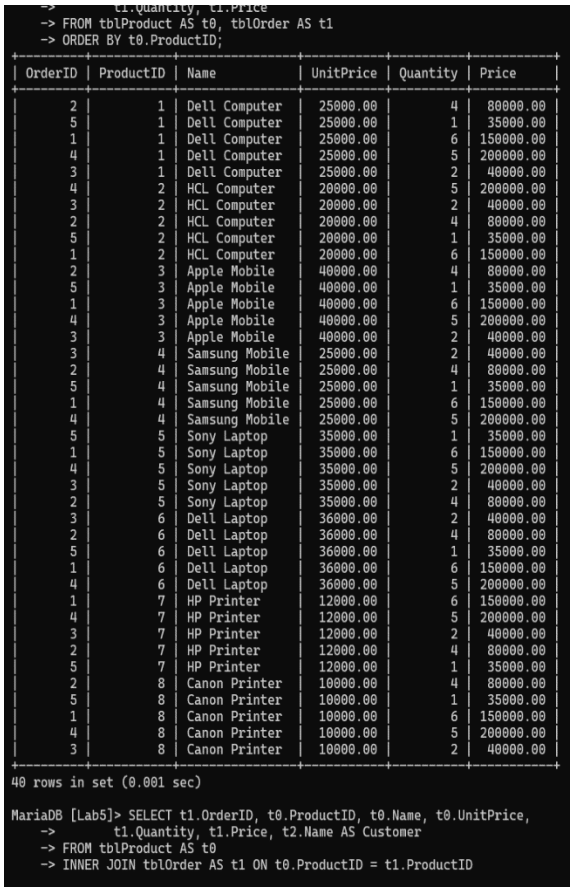
```
+----+---------+------+-----------+----------+
| id | name    | age  | address   | salary   |
+----+---------+------+-----------+----------+
|  2 | Ramesh  |   25 | Delhi     |  1500.00 |
|  1 | Ramesh  |   32 | Ahmedabad |  2000.00 |
|  7 | Muffy   |   24 | Indore    | 10000.00 |
|  6 | Komal   |   22 | MP        |  4500.00 |
|  3 | kaushik |   23 | Kota      |  2000.00 |
|  4 | kaushik |   25 | Mumbai    |  6500.00 |
|  5 | Hardik  |   27 | Bhopal    |  8500.00 |
+----+---------+------+-----------+----------+
7 rows in set (0.001 sec)

MariaDB [Lab5]> SELECT *
    -> FROM customers
    -> WHERE LOWER(address) IN ('kota','mumbai','indore');
+----+---------+------+---------+----------+
| id | name    | age  | address | salary   |
+----+---------+------+---------+----------+
|  3 | kaushik |   23 | Kota    |  2000.00 |
|  4 | kaushik |   25 | Mumbai  |  6500.00 |
|  7 | Muffy   |   24 | Indore  | 10000.00 |
+----+---------+------+---------+----------+
3 rows in set (0.002 sec)

MariaDB [Lab5]> SELECT *
    -> FROM customers
    -> WHERE id IN (SELECT id FROM customers WHERE salary > 4500);
+----+---------+------+---------+----------+
| id | name    | age  | address | salary   |
+----+---------+------+---------+----------+
|  4 | kaushik |   25 | Mumbai  |  6500.00 |
|  5 | Hardik  |   27 | Bhopal  |  8500.00 |
|  7 | Muffy   |   24 | Indore  | 10000.00 |
+----+---------+------+---------+----------+
3 rows in set (0.001 sec)

MariaDB [Lab5]> UPDATE customers
    -> SET salary = salary * 0.25
    -> WHERE age IN (SELECT age FROM customers WHERE age > 27);
Query OK, 1 row affected (0.007 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [Lab5]> SELECT * FROM customers ORDER BY id;
+----+---------+------+-----------+----------+
| id | name    | age  | address   | salary   |
+----+---------+------+-----------+----------+
|  1 | Ramesh  |   32 | Ahmedabad |   500.00 |
|  2 | Ramesh  |   25 | Delhi     |  1500.00 |
|  3 | kaushik |   23 | Kota      |  2000.00 |
|  4 | kaushik |   25 | Mumbai    |  6500.00 |
|  5 | Hardik  |   27 | Bhopal    |  8500.00 |
|  6 | Komal   |   22 | MP        |  4500.00 |
```

```
|  6 | Komal   |   22 | MP        |  4500.00 |
|  7 | Muffy   |   24 | Indore    | 10000.00 |
+----+---------+------+-----------+----------+
7 rows in set (0.001 sec)

MariaDB [Lab5]> DROP TABLE IF EXISTS tblOrder;
Query OK, 0 rows affected, 1 warning (0.002 sec)

MariaDB [Lab5]> DROP TABLE IF EXISTS tblProduct;
Query OK, 0 rows affected, 1 warning (0.000 sec)

MariaDB [Lab5]> DROP TABLE IF EXISTS tblCustomer;
Query OK, 0 rows affected, 1 warning (0.001 sec)

MariaDB [Lab5]> CREATE TABLE tblCustomer (
    ->     CustID INT PRIMARY KEY,
    ->     Name VARCHAR(50),
    ->     Address VARCHAR(100),
    ->     ContactNo VARCHAR(20)
    -> );
Query OK, 0 rows affected (0.022 sec)

MariaDB [Lab5]> INSERT INTO tblCustomer (CustID, Name, Address, ContactNo) VALUES
    -> (1, 'Sam', 'New Delhi', '9555555555'),
    -> (2, 'Rahul', 'Gurgaon', '9666666666'),
    -> (3, 'Hans', 'Noida', '9444444444'),
    -> (4, 'Jeetu', 'Delhi', '9333333333'),
    -> (5, 'Ankit', 'Noida', '9222222222');
Query OK, 5 rows affected (0.009 sec)
Records: 5  Duplicates: 0  Warnings: 0

MariaDB [Lab5]> CREATE TABLE tblProduct (
    ->     ProductID INT PRIMARY KEY,
    ->     Name VARCHAR(100),
    ->     UnitPrice DECIMAL(10,2),
    ->     CatID INT,
    ->     EntryDate DATETIME,
    ->     ExpiryDate DATETIME
    -> );
Query OK, 0 rows affected (0.027 sec)

MariaDB [Lab5]> INSERT INTO tblProduct (ProductID, Name, UnitPrice, CatID, EntryDate, ExpiryDate) VALUES
    -> (1, 'Dell Computer', 25000, 1, '2012-10-16 23:05:05.550', '2012-10-16 23:05:05.550'),
    -> (2, 'HCL Computer', 20000, 1, '2012-10-16 23:05:46.990', '2012-10-16 23:05:46.990'),
    -> (3, 'Apple Mobile', 40000, 3, '2012-10-16 23:06:11.283', '2012-10-16 23:06:11.283'),
    -> (4, 'Samsung Mobile', 25000, 3, '2012-10-16 23:06:28.727', '2012-10-16 23:06:28.727'),
    -> (5, 'Sony Laptop', 35000, 2, '2012-10-16 23:07:05.212', '2012-10-16 23:07:05.212'),
    -> (6, 'Dell Laptop', 36000, 2, '2012-10-16 23:07:38.201', '2012-10-16 23:07:38.201'),
    -> (7, 'HP Printer', 12000, 4, '2012-10-16 23:07:45.010', '2012-10-16 23:07:45.010'),
    -> (8, 'Canon Printer', 10000, 4, '2012-10-16 23:07:54.213', '2012-10-16 23:07:54.213');
Query OK, 8 rows affected (0.016 sec)
Records: 8  Duplicates: 0  Warnings: 0
```

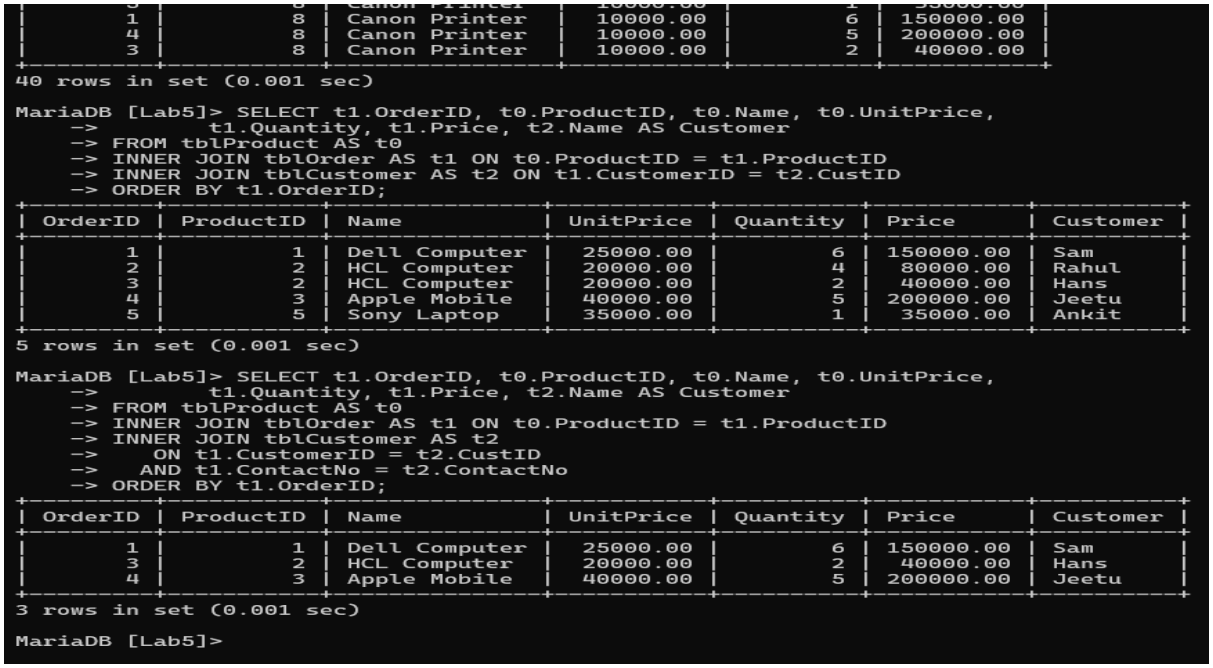
```
Records: 8  Duplicates: 0  Warnings: 0

MariaDB [Lab5]> CREATE TABLE tblOrder (
    ->     OrderID INT PRIMARY KEY,
    ->     ProductID INT,
    ->     Quantity INT,
    ->     Price DECIMAL(12,2),
    ->     CustomerID INT,
    ->     ContactNo VARCHAR(20),
    ->     FOREIGN KEY (ProductID) REFERENCES tblProduct(ProductID),
    ->     FOREIGN KEY (CustomerID) REFERENCES tblCustomer(CustID)
    -> );
Query OK, 0 rows affected (0.048 sec)

MariaDB [Lab5]> INSERT INTO tblOrder (OrderID, ProductID, Quantity, Price, CustomerID, ContactNo) VALUES
    -> (1, 1, 6, 150000, 1, '9555555555'),
    -> (2, 2, 4, 80000, 2, NULL),
    -> (3, 2, 2, 40000, 3, '9444444444'),
    -> (4, 3, 5, 200000, 4, '9333333333'),
    -> (5, 5, 1, 35000, 5, '9666666666');
Query OK, 5 rows affected (0.016 sec)
Records: 5  Duplicates: 0  Warnings: 0

MariaDB [Lab5]> SELECT t1.OrderID, t0.ProductID, t0.Name, t0.UnitPrice,
    ->       t1.Quantity, t1.Price
    -> FROM tblProduct AS t0
    -> INNER JOIN tblOrder AS t1
    ->     ON t0.ProductID = t1.ProductID
    -> ORDER BY t1.OrderID;
+---------+-----------+---------------+-----------+----------+----------+
| OrderID | ProductID | Name          | UnitPrice | Quantity | Price    |
+---------+-----------+---------------+-----------+----------+----------+
|       1 |         1 | Dell Computer |  25000.00 |        6 | 150000.00 |
|       2 |         2 | HCL Computer  |  20000.00 |        4 |  80000.00 |
|       3 |         2 | HCL Computer  |  20000.00 |        2 |  40000.00 |
|       4 |         3 | Apple Mobile  |  40000.00 |        5 | 200000.00 |
|       5 |         5 | Sony Laptop   |  35000.00 |        1 |  35000.00 |
+---------+-----------+---------------+-----------+----------+----------+
5 rows in set (0.001 sec)

MariaDB [Lab5]> SELECT t1.OrderID, t0.ProductID, t0.Name, t0.UnitPrice,
    ->       t1.Quantity, t1.Price
    -> FROM tblProduct AS t0
    -> LEFT JOIN tblOrder AS t1
    ->     ON t0.ProductID = t1.ProductID
    -> ORDER BY t0.ProductID;
+---------+-----------+---------------+-----------+----------+----------+
| OrderID | ProductID | Name          | UnitPrice | Quantity | Price    |
+---------+-----------+---------------+-----------+----------+----------+
|       1 |         1 | Dell Computer |  25000.00 |        6 | 150000.00 |
|       3 |         2 | HCL Computer  |  20000.00 |        2 |  40000.00 |
|       2 |         2 | HCL Computer  |  20000.00 |        4 |  80000.00 |
|       4 |         3 | Apple Mobile  |  40000.00 |        5 | 200000.00 |
```



```
|       5 |         5 | Sony Laptop     |  35000.00 |        1 |  35000.00 |
|    NULL |         6 | Dell Laptop     |  36000.00 |     NULL |      NULL |
|    NULL |         7 | HP Printer      |  12000.00 |     NULL |      NULL |
|    NULL |         8 | Canon Printer   |  10000.00 |     NULL |      NULL |
+---------+-----------+-----------------+-----------+----------+-----------+
9 rows in set (0.001 sec)

MariaDB [Lab5]> SELECT t1.OrderID, t0.ProductID, t0.Name, t0.UnitPrice,
    ->        t1.Quantity, t1.Price
    -> FROM tblProduct AS t0
    -> RIGHT JOIN tblOrder AS t1
    ->      ON t0.ProductID = t1.ProductID
    -> ORDER BY t0.ProductID;
+---------+-----------+-----------------+-----------+----------+-----------+
| OrderID | ProductID | Name            | UnitPrice | Quantity | Price     |
+---------+-----------+-----------------+-----------+----------+-----------+
|       1 |         1 | Dell Computer   |  25000.00 |        6 | 150000.00 |
|       3 |         2 | HCL Computer    |  20000.00 |        2 |  40000.00 |
|       2 |         2 | HCL Computer    |  20000.00 |        4 |  80000.00 |
|       4 |         3 | Apple Mobile    |  40000.00 |        5 | 200000.00 |
|       5 |         5 | Sony Laptop     |  35000.00 |        1 |  35000.00 |
+---------+-----------+-----------------+-----------+----------+-----------+
5 rows in set (0.001 sec)

MariaDB [Lab5]> SELECT t1.OrderID, t0.ProductID, t0.Name, t0.UnitPrice,
    ->        t1.Quantity, t1.Price
    -> FROM tblProduct AS t0
    -> LEFT JOIN tblOrder AS t1 ON t0.ProductID = t1.ProductID
    -> UNION
    -> SELECT t1.OrderID, t0.ProductID, t0.Name, t0.UnitPrice,
    ->        t1.Quantity, t1.Price
    -> FROM tblProduct AS t0
    -> RIGHT JOIN tblOrder AS t1 ON t0.ProductID = t1.ProductID
    -> ORDER BY ProductID;
+---------+-----------+-----------------+-----------+----------+-----------+
| OrderID | ProductID | Name            | UnitPrice | Quantity | Price     |
+---------+-----------+-----------------+-----------+----------+-----------+
|       1 |         1 | Dell Computer   |  25000.00 |        6 | 150000.00 |
|       3 |         2 | HCL Computer    |  20000.00 |        2 |  40000.00 |
|       2 |         2 | HCL Computer    |  20000.00 |        4 |  80000.00 |
|       4 |         3 | Apple Mobile    |  40000.00 |        5 | 200000.00 |
|    NULL |         4 | Samsung Mobile  |  25000.00 |     NULL |      NULL |
|       5 |         5 | Sony Laptop     |  35000.00 |        1 |  35000.00 |
|    NULL |         6 | Dell Laptop     |  36000.00 |     NULL |      NULL |
|    NULL |         7 | HP Printer      |  12000.00 |     NULL |      NULL |
|    NULL |         8 | Canon Printer   |  10000.00 |     NULL |      NULL |
+---------+-----------+-----------------+-----------+----------+-----------+
9 rows in set (0.002 sec)

MariaDB [Lab5]> SELECT t1.OrderID, t0.ProductID, t0.Name, t0.UnitPrice,
    ->        t1.Quantity, t1.Price
    -> FROM tblProduct AS t0, tblOrder AS t1
    -> ORDER BY t0.ProductID;
```



```
    ->        t1.Quantity, t1.Price
    -> FROM tblProduct AS t0, tblOrder AS t1
    -> ORDER BY t0.ProductID;
+---------+-----------+-----------------+-----------+----------+-----------+
| OrderID | ProductID | Name            | UnitPrice | Quantity | Price     |
+---------+-----------+-----------------+-----------+----------+-----------+
|       2 |         1 | Dell Computer   |  25000.00 |        4 |  80000.00 |
|       5 |         1 | Dell Computer   |  25000.00 |        1 |  35000.00 |
|       1 |         1 | Dell Computer   |  25000.00 |        6 | 150000.00 |
|       4 |         1 | Dell Computer   |  25000.00 |        5 | 200000.00 |
|       3 |         1 | Dell Computer   |  25000.00 |        2 |  40000.00 |
|       4 |         2 | HCL Computer    |  20000.00 |        5 | 200000.00 |
|       3 |         2 | HCL Computer    |  20000.00 |        2 |  40000.00 |
|       2 |         2 | HCL Computer    |  20000.00 |        4 |  80000.00 |
|       5 |         2 | HCL Computer    |  20000.00 |        1 |  35000.00 |
|       1 |         2 | HCL Computer    |  20000.00 |        6 | 150000.00 |
|       2 |         3 | Apple Mobile    |  40000.00 |        4 |  80000.00 |
|       5 |         3 | Apple Mobile    |  40000.00 |        1 |  35000.00 |
|       1 |         3 | Apple Mobile    |  40000.00 |        6 | 150000.00 |
|       4 |         3 | Apple Mobile    |  40000.00 |        5 | 200000.00 |
|       3 |         3 | Apple Mobile    |  40000.00 |        2 |  40000.00 |
|       3 |         4 | Samsung Mobile  |  25000.00 |        2 |  40000.00 |
|       2 |         4 | Samsung Mobile  |  25000.00 |        4 |  80000.00 |
|       5 |         4 | Samsung Mobile  |  25000.00 |        1 |  35000.00 |
|       1 |         4 | Samsung Mobile  |  25000.00 |        6 | 150000.00 |
|       4 |         4 | Samsung Mobile  |  25000.00 |        5 | 200000.00 |
|       5 |         5 | Sony Laptop     |  35000.00 |        1 |  35000.00 |
|       1 |         5 | Sony Laptop     |  35000.00 |        6 | 150000.00 |
|       4 |         5 | Sony Laptop     |  35000.00 |        5 | 200000.00 |
|       3 |         5 | Sony Laptop     |  35000.00 |        2 |  40000.00 |
|       2 |         5 | Sony Laptop     |  35000.00 |        4 |  80000.00 |
|       3 |         6 | Dell Laptop     |  36000.00 |        2 |  40000.00 |
|       2 |         6 | Dell Laptop     |  36000.00 |        4 |  80000.00 |
|       5 |         6 | Dell Laptop     |  36000.00 |        1 |  35000.00 |
|       1 |         6 | Dell Laptop     |  36000.00 |        6 | 150000.00 |
|       4 |         6 | Dell Laptop     |  36000.00 |        5 | 200000.00 |
|       1 |         7 | HP Printer      |  12000.00 |        6 | 150000.00 |
|       4 |         7 | HP Printer      |  12000.00 |        5 | 200000.00 |
|       3 |         7 | HP Printer      |  12000.00 |        2 |  40000.00 |
|       2 |         7 | HP Printer      |  12000.00 |        4 |  80000.00 |
|       5 |         7 | HP Printer      |  12000.00 |        1 |  35000.00 |
|       2 |         8 | Canon Printer   |  10000.00 |        4 |  80000.00 |
|       5 |         8 | Canon Printer   |  10000.00 |        1 |  35000.00 |
|       1 |         8 | Canon Printer   |  10000.00 |        6 | 150000.00 |
|       4 |         8 | Canon Printer   |  10000.00 |        5 | 200000.00 |
|       3 |         8 | Canon Printer   |  10000.00 |        2 |  40000.00 |
+---------+-----------+-----------------+-----------+----------+-----------+
40 rows in set (0.001 sec)

MariaDB [Lab5]> SELECT t1.OrderID, t0.ProductID, t0.Name, t0.UnitPrice,
    ->        t1.Quantity, t1.Price, t2.Name AS Customer
    -> FROM tblProduct AS t0
    -> INNER JOIN tblOrder AS t1 ON t0.ProductID = t1.ProductID
    -> INNER JOIN tblCustomer AS t2 ON t1.CustomerID = t2.CustID
    -> ORDER BY t1.OrderID;
```



```
|       1 |         8 | Canon Printer   |  10000.00 |        6 | 150000.00 |
|       4 |         8 | Canon Printer   |  10000.00 |        5 | 200000.00 |
|       3 |         8 | Canon Printer   |  10000.00 |        2 |  40000.00 |
+---------+-----------+-----------------+-----------+----------+-----------+
40 rows in set (0.001 sec)

MariaDB [Lab5]> SELECT t1.OrderID, t0.ProductID, t0.Name, t0.UnitPrice,
    ->        t1.Quantity, t1.Price, t2.Name AS Customer
    -> FROM tblProduct AS t0
    -> INNER JOIN tblOrder AS t1 ON t0.ProductID = t1.ProductID
    -> INNER JOIN tblCustomer AS t2 ON t1.CustomerID = t2.CustID
    -> ORDER BY t1.OrderID;
+---------+-----------+-----------------+-----------+----------+-----------+----------+
| OrderID | ProductID | Name            | UnitPrice | Quantity | Price     | Customer |
+---------+-----------+-----------------+-----------+----------+-----------+----------+
|       1 |         1 | Dell Computer   |  25000.00 |        6 | 150000.00 | Sam      |
|       2 |         2 | HCL Computer    |  20000.00 |        4 |  80000.00 | Rahul    |
|       3 |         2 | HCL Computer    |  20000.00 |        2 |  40000.00 | Hans     |
|       4 |         3 | Apple Mobile    |  40000.00 |        5 | 200000.00 | Jeetu    |
|       5 |         5 | Sony Laptop     |  35000.00 |        1 |  35000.00 | Ankit    |
+---------+-----------+-----------------+-----------+----------+-----------+----------+
5 rows in set (0.001 sec)

MariaDB [Lab5]> SELECT t1.OrderID, t0.ProductID, t0.Name, t0.UnitPrice,
    ->        t1.Quantity, t1.Price, t2.Name AS Customer
    -> FROM tblProduct AS t0
    -> INNER JOIN tblOrder AS t1 ON t0.ProductID = t1.ProductID
    -> INNER JOIN tblCustomer AS t2
    ->     ON t1.CustomerID = t2.CustID
    ->    AND t1.ContactNo = t2.ContactNo
    -> ORDER BY t1.OrderID;
+---------+-----------+-----------------+-----------+----------+-----------+----------+
| OrderID | ProductID | Name            | UnitPrice | Quantity | Price     | Customer |
+---------+-----------+-----------------+-----------+----------+-----------+----------+
|       1 |         1 | Dell Computer   |  25000.00 |        6 | 150000.00 | Sam      |
|       3 |         2 | HCL Computer    |  20000.00 |        2 |  40000.00 | Hans     |
|       4 |         3 | Apple Mobile    |  40000.00 |        5 | 200000.00 | Jeetu    |
+---------+-----------+-----------------+-----------+----------+-----------+----------+
3 rows in set (0.001 sec)

MariaDB [Lab5]>
```

Conclusion:

In this lab, we practiced using views, aggregate functions, grouping, sorting, subqueries, table updates, and joins. These concepts helped us understand how to organize, analyze, and manipulate data efficiently in relational databases.

Lab:6

Objective:To practice SQL queries, joins, views, subqueries, aggregate functions, and grouping for efficient data retrieval and manipulation.

Task:

1. Create Employee Table

CREATE TABLE employee (emp_id INT PRIMARY KEY, fname VARCHAR(50), lname VARCHAR(50), salary DECIMAL(10,2), joindate DATE, dept VARCHAR(50));

2. Insert John Smith

INSERT INTO employee VALUES (1, 'John', 'Smith', 60000, '2025-01-01', 'Computer');

3. Select First_Name, Last_Name as Surname

SELECT fname, lname AS Surname FROM employee;

4. Select Unique Departments

SELECT DISTINCT dept FROM employee;

5. Select All Employee Details Ordered by First Name Desc

SELECT * FROM employee ORDER BY fname DESC;

6. Employees Named “John” or “Roy”

SELECT * FROM employee WHERE fname IN ('John', 'Roy');

7. Employees with Salary Between 50000 and 80000

SELECT * FROM employee WHERE salary BETWEEN 50000 AND 80000;

8. Employee Incentives from Computer Department

SELECT e.fname, e.lname, i.incentive_date, i.incentive_amt FROM employee e INNER JOIN incentives i ON e.emp_id = i.emp_id WHERE e.dept = 'Computer';

9. Retrieve Rows with Custno Between 98 and 100

SELECT * FROM customer WHERE custno BETWEEN 98 AND 100;

10. Retrieve Rows Where City is PKR, KTM, or BTW

SELECT * FROM customer WHERE city IN ('PKR','KTM','BTW');

11. Create and Select View

CREATE VIEW testview AS SELECT fname, lname, rollno, address FROM employee; SELECT * FROM testview;

12. Aggregate Functions on Salary Table

SELECT COUNT(*) FROM salary; SELECT AVG(basic) FROM salary; SELECT MAX(basic) FROM salary;

SELECT MIN(basic) FROM salary; SELECT SUM(basic) FROM salary;

13. Second Maximum Salary

SELECT MAX(basic) FROM salary WHERE basic NOT IN (SELECT MAX(basic) FROM salary);

14. Employee Names and Departments Ordered

SELECT e.ename, d.dname FROM emp e INNER JOIN dept d ON e.deptno = d.deptno ORDER BY d.dname;

15. Update 10% Salary for Computer Dept

UPDATE employee SET salary = salary * 1.10 WHERE deptno = 2;

16. Employees Earning Less Than Average Salary

SELECT * FROM employee WHERE salary < (SELECT AVG(basic) FROM salary);

17. Employees Earning Less Than 10000

SELECT * FROM employee WHERE empno IN (SELECT empno FROM salary WHERE basic < 10

Grouping:

```
MariaDB [lab6db]> CREATE TABLE Store_Information (
    ->     Store_Name VARCHAR(50),
    ->     Sales DECIMAL(10,2),
    ->     Txn_Date DATE
    -> );
Query OK, 0 rows affected (0.012 sec)

MariaDB [lab6db]> INSERT INTO Store_Information (Store_Name, Sales, Txn_Date) VALUES
    -> ('Los Angeles', 1500, '1999-01-05'),
    -> ('San Diego', 250, '1999-01-07'),
    -> ('Los Angeles', 300, '1999-01-08'),
    -> ('Boston', 700, '1999-01-08');
Query OK, 4 rows affected (0.003 sec)
Records: 4  Duplicates: 0  Warnings: 0

MariaDB [lab6db]> CREATE TABLE Geography (
    ->     Region_Name VARCHAR(50),
    ->     Store_Name VARCHAR(50)
    -> );
Query OK, 0 rows affected (0.019 sec)

MariaDB [lab6db]> INSERT INTO Geography (Region_Name, Store_Name) VALUES
    -> ('East', 'Boston'),
    -> ('East', 'New York'),
    -> ('West', 'Los Angeles'),
    -> ('West', 'San Diego');
Query OK, 4 rows affected (0.011 sec)
Records: 4  Duplicates: 0  Warnings: 0

MariaDB [lab6db]> SELECT g.Region_Name AS REGION, SUM(s.Sales) AS SALES
    -> FROM Geography g
    -> JOIN Store_Information s ON g.Store_Name = s.Store_Name
    -> GROUP BY g.Region_Name;
+--------+---------+
| REGION | SALES   |
+--------+---------+
| East   |  700.00 |
| West   | 2050.00 |
+--------+---------+
2 rows in set (0.001 sec)
```

```
MariaDB [lab6db]> SELECT g.Store_Name AS STORE, SUM(s.Sales) AS SALES
    -> FROM Geography g
    -> INNER JOIN Store_Information s ON g.Store_Name = s.Store_Name
    -> GROUP BY g.Store_Name;
+-------------+---------+
| STORE       | SALES   |
+-------------+---------+
| Boston      |  700.00 |
| Los Angeles | 1800.00 |
| San Diego   |  250.00 |
+-------------+---------+
3 rows in set (0.001 sec)

MariaDB [lab6db]> SELECT g.Store_Name AS STORE, SUM(s.Sales) AS SALES
    -> FROM Geography g
    -> LEFT JOIN Store_Information s ON g.Store_Name = s.Store_Name
    -> GROUP BY g.Store_Name;
+-------------+---------+
| STORE       | SALES   |
+-------------+---------+
| Boston      |  700.00 |
| Los Angeles | 1800.00 |
| New York    |    NULL |
| San Diego   |  250.00 |
+-------------+---------+
4 rows in set (0.002 sec)

MariaDB [lab6db]> -- Sum sales by region
MariaDB [lab6db]> SELECT g.Region_Name AS REGION, SUM(s.Sales) AS SALES
    -> FROM Geography g
    -> JOIN Store_Information s ON g.Store_Name = s.Store_Name
    -> GROUP BY g.Region_Name;
+--------+---------+
| REGION | SALES   |
+--------+---------+
| East   |  700.00 |
| West   | 2050.00 |
+--------+---------+
2 rows in set (0.001 sec)
```

Join:

```
MariaDB [lab6db]>
MariaDB [lab6db]> -- Inner join sum by store
MariaDB [lab6db]> SELECT g.Store_Name AS STORE, SUM(s.Sales) AS SALES
    -> FROM Geography g
    -> INNER JOIN Store_Information s ON g.Store_Name = s.Store_Name
    -> GROUP BY g.Store_Name;
+-------------+---------+
| STORE       | SALES   |
+-------------+---------+
| Boston      |  700.00 |
| Los Angeles | 1800.00 |
| San Diego   |  250.00 |
+-------------+---------+
3 rows in set (0.001 sec)

MariaDB [lab6db]>
MariaDB [lab6db]> -- Left join example
MariaDB [lab6db]> SELECT g.Store_Name AS STORE, SUM(s.Sales) AS SALES
    -> FROM Geography g
    -> LEFT JOIN Store_Information s ON g.Store_Name = s.Store_Name
    -> GROUP BY g.Store_Name;
+-------------+---------+
| STORE       | SALES   |
+-------------+---------+
| Boston      |  700.00 |
| Los Angeles | 1800.00 |
| New York    |    NULL |
| San Diego   |  250.00 |
+-------------+---------+
4 rows in set (0.001 sec)

MariaDB [lab6db]>
```

Conclusion:

we practiced SQL queries, views, joins, subqueries, and aggregate functions. We learned to retrieve, filter, and group data efficiently, and observed how different joins and constraints affect the results. The lab enhanced our understanding of relational database operations and data analysis.

Lab:7

Objective: To understand and apply database normalization (1NF, 2NF, 3NF, BCNF) and demonstrate the use of SQL JOIN operations on normalized tables.

Theory:

Normalization is the process of organizing data to reduce redundancy and improve integrity.

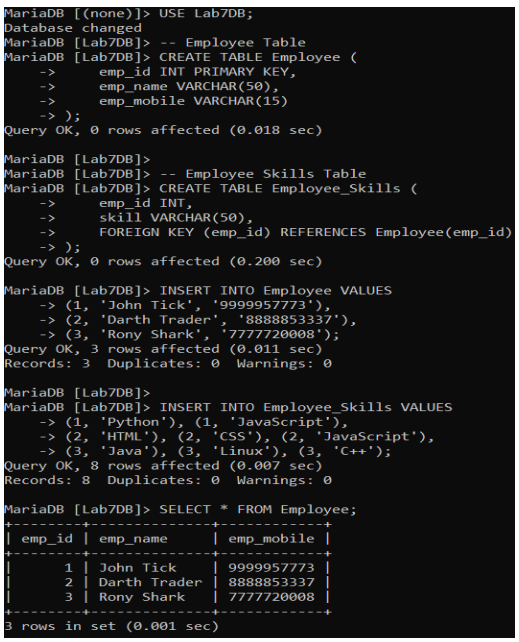
- **1NF (First Normal Form):** Eliminate repeating groups and ensure atomic values.
- **2NF (Second Normal Form):** Remove partial dependencies (non-prime attributes depending on part of a candidate key).
- **3NF (Third Normal Form):** Remove transitive dependencies (non-prime attributes depending on other non-prime attributes).
- **BCNF (Boyce-Codd Normal Form):** Every determinant must be a candidate key.

Joins are used to combine data from multiple tables:

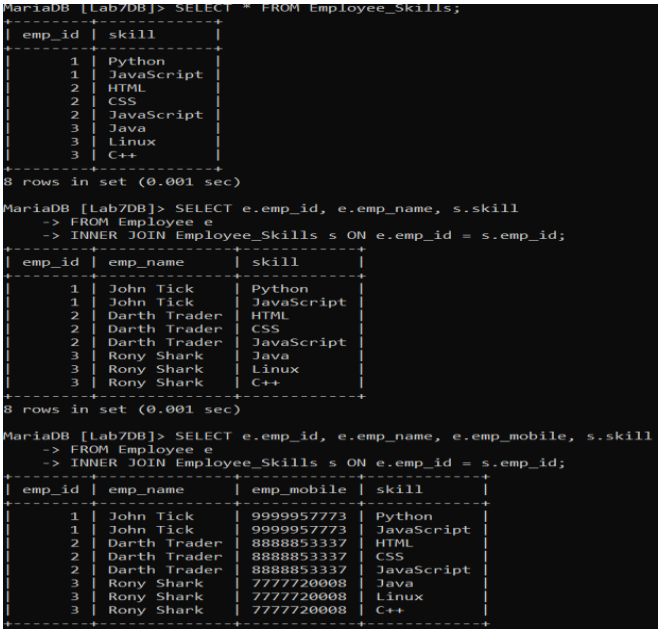
- **INNER JOIN:** Returns only matching rows.
- **LEFT JOIN:** Returns all rows from the left table and matching rows from the right.
- **RIGHT JOIN:** Returns all rows from the right table and matching rows from the left.
- **FULL OUTER JOIN:** Returns all rows when there is a match in either table.
- **SELF JOIN:** Joins a table with itself.

Tasks :

Q1. First Normal Form (1NF)



```
MariaDB [(none)]> USE Lab7DB;
Database changed
MariaDB [Lab7DB]> -- Employee Table
MariaDB [Lab7DB]> CREATE TABLE Employee (
    ->     emp_id INT PRIMARY KEY,
    ->     emp_name VARCHAR(50),
    ->     emp_mobile VARCHAR(15)
    -> );
Query OK, 0 rows affected (0.018 sec)

MariaDB [Lab7DB]>
MariaDB [Lab7DB]> -- Employee Skills Table
MariaDB [Lab7DB]> CREATE TABLE Employee_Skills (
    ->     emp_id INT,
    ->     skill VARCHAR(50),
    ->     FOREIGN KEY (emp_id) REFERENCES Employee(emp_id)
    -> );
Query OK, 0 rows affected (0.200 sec)

MariaDB [Lab7DB]> INSERT INTO Employee VALUES
    -> (1, 'John Tick', '9999957773'),
    -> (2, 'Darth Trader', '8888853337'),
    -> (3, 'Rony Shark', '7777720008');
Query OK, 3 rows affected (0.011 sec)
Records: 3  Duplicates: 0  Warnings: 0

MariaDB [Lab7DB]>
MariaDB [Lab7DB]> INSERT INTO Employee_Skills VALUES
    -> (1, 'Python'), (1, 'JavaScript'),
    -> (2, 'HTML'), (2, 'CSS'), (2, 'JavaScript'),
    -> (3, 'Java'), (3, 'Linux'), (3, 'C++');
Query OK, 8 rows affected (0.007 sec)
Records: 8  Duplicates: 0  Warnings: 0

MariaDB [Lab7DB]> SELECT * FROM Employee;
+--------+-------------+------------+
| emp_id | emp_name    | emp_mobile |
+--------+-------------+------------+
|      1 | John Tick   | 9999957773 |
|      2 | Darth Trader | 8888853337 |
|      3 | Rony Shark  | 7777720008 |
+--------+-------------+------------+
3 rows in set (0.001 sec)
```



```
MariaDB [Lab7DB]> SELECT * FROM Employee_Skills;
+--------+------------+
| emp_id | skill      |
+--------+------------+
|      1 | Python     |
|      1 | JavaScript |
|      2 | HTML       |
|      2 | CSS        |
|      2 | JavaScript |
|      3 | Java       |
|      3 | Linux      |
|      3 | C++        |
+--------+------------+
8 rows in set (0.001 sec)

MariaDB [Lab7DB]> SELECT e.emp_id, e.emp_name, s.skill
    -> FROM Employee e
    -> INNER JOIN Employee_Skills s ON e.emp_id = s.emp_id;
+--------+-------------+------------+
| emp_id | emp_name    | skill      |
+--------+-------------+------------+
|      1 | John Tick   | Python     |
|      1 | John Tick   | JavaScript |
|      2 | Darth Trader | HTML       |
|      2 | Darth Trader | CSS        |
|      2 | Darth Trader | JavaScript |
|      3 | Rony Shark  | Java       |
|      3 | Rony Shark  | Linux      |
|      3 | Rony Shark  | C++        |
+--------+-------------+------------+
8 rows in set (0.001 sec)

MariaDB [Lab7DB]> SELECT e.emp_id, e.emp_name, e.emp_mobile, s.skill
    -> FROM Employee e
    -> INNER JOIN Employee_Skills s ON e.emp_id = s.emp_id;
+--------+-------------+------------+------------+
| emp_id | emp_name    | emp_mobile | skill      |
+--------+-------------+------------+------------+
|      1 | John Tick   | 9999957773 | Python     |
|      1 | John Tick   | 9999957773 | JavaScript |
|      2 | Darth Trader | 8888853337 | HTML       |
|      2 | Darth Trader | 8888853337 | CSS        |
|      2 | Darth Trader | 8888853337 | JavaScript |
|      3 | Rony Shark  | 7777720008 | Java       |
|      3 | Rony Shark  | 7777720008 | Linux      |
|      3 | Rony Shark  | 7777720008 | C++        |
+--------+-------------+------------+------------+
```

Q2. Second Normal Form (2NF)

```
MariaDB [Lab7DB]> -- Students table
MariaDB [Lab7DB]> CREATE TABLE Students (
    ->     student_id INT PRIMARY KEY,
    ->     student_name VARCHAR(50)
    -> );
Query OK, 0 rows affected (0.018 sec)

MariaDB [Lab7DB]>
MariaDB [Lab7DB]> -- Subjects table
MariaDB [Lab7DB]> CREATE TABLE Subjects (
    ->     subject_id INT PRIMARY KEY,
    ->     subject_name VARCHAR(50),
    ->     teacher_name VARCHAR(50)
    -> );
Query OK, 0 rows affected (0.011 sec)

MariaDB [Lab7DB]>
MariaDB [Lab7DB]> -- Marks table
MariaDB [Lab7DB]> CREATE TABLE Marks (
    ->     student_id INT,
    ->     subject_id INT,
    ->     marks INT,
    ->     FOREIGN KEY (student_id) REFERENCES Students(student_id),
    ->     FOREIGN KEY (subject_id) REFERENCES Subjects(subject_id)
    -> );
Query OK, 0 rows affected (0.015 sec)

MariaDB [Lab7DB]> SELECT * FROM Students;
Empty set (0.002 sec)

MariaDB [Lab7DB]> SELECT * FROM Subjects;
Empty set (0.002 sec)

MariaDB [Lab7DB]> INSERT INTO Students (student_id, student_name) VALUES
    -> (1, 'Akon'),
    -> (2, 'Bkon');
Query OK, 2 rows affected (0.008 sec)
Records: 2  Duplicates: 0  Warnings: 0

MariaDB [Lab7DB]> SELECT * FROM Students;
+------------+--------------+
| student_id | student_name |
+------------+--------------+
|          1 | Akon         |
|          2 | Bkon         |
+------------+--------------+
```

```
MariaDB [Lab7DB]> INSERT INTO Subjects (subject_id, subject_name, teacher_name) VALUES
    -> (1, 'C Language', 'Miss. Sita'),
    -> (2, 'DSA', 'Mr. Dilip'),
    -> (3, 'Operating System', 'Mr. Lata');
Query OK, 3 rows affected (0.004 sec)
Records: 3  Duplicates: 0  Warnings: 0

MariaDB [Lab7DB]> SELECT * FROM Subjects;
+------------+------------------+--------------+
| subject_id | subject_name     | teacher_name |
+------------+------------------+--------------+
|          1 | C Language       | Miss. Sita   |
|          2 | DSA              | Mr. Dilip    |
|          3 | Operating System | Mr. Lata     |
+------------+------------------+--------------+
3 rows in set (0.001 sec)

MariaDB [Lab7DB]> INSERT INTO Marks (student_id, subject_id, marks) VALUES
    -> (1, 1, 70),
    -> (1, 2, 82),
    -> (2, 3, 65);
Query OK, 3 rows affected (0.010 sec)
Records: 3  Duplicates: 0  Warnings: 0

MariaDB [Lab7DB]> SELECT * FROM Marks;
+------------+------------+-------+
| student_id | subject_id | marks |
+------------+------------+-------+
|          1 |          1 |    70 |
|          1 |          2 |    82 |
|          2 |          3 |    65 |
+------------+------------+-------+
3 rows in set (0.001 sec)

MariaDB [Lab7DB]> SELECT st.student_name, sub.subject_name, m.marks, sub.teacher_name
    -> FROM Marks m
    -> INNER JOIN Students st ON m.student_id = st.student_id
    -> INNER JOIN Subjects sub ON m.subject_id = sub.subject_id;
+--------------+------------------+-------+--------------+
| student_name | subject_name     | marks | teacher_name |
+--------------+------------------+-------+--------------+
| Akon         | C Language       |    70 | Miss. Sita   |
| Akon         | DSA              |    82 | Mr. Dilip    |
| Bkon         | Operating System |    65 | Mr. Lata     |
+--------------+------------------+-------+--------------+
3 rows in set (0.001 sec)
```

Q3. **Third Normal Form (3NF).**

```
MariaDB [Lab7DB]> -- Employees table
MariaDB [Lab7DB]> CREATE TABLE Employees (
    ->     emp_id INT PRIMARY KEY,
    ->     emp_name VARCHAR(50),
    ->     emp_zip INT
    -> );
Query OK, 0 rows affected (0.022 sec)

MariaDB [Lab7DB]>
MariaDB [Lab7DB]> -- Location table
MariaDB [Lab7DB]> CREATE TABLE Location (
    ->     emp_zip INT PRIMARY KEY,
    ->     emp_state VARCHAR(50),
    ->     emp_city VARCHAR(50),
    ->     emp_district VARCHAR(50)
    -> );
Query OK, 0 rows affected (0.013 sec)
```

```
MariaDB [Lab7DB]> INSERT INTO Employees (emp_id, emp_name, emp_zip) VALUES
    -> (1, 'Aiyana', 28002),
    -> (2, 'Sita', 28008),
    -> (3, 'Radha', 28007),
    -> (4, 'Saru', 28007),
    -> (5, 'Romi', 28008);
Query OK, 5 rows affected (0.009 sec)
Records: 5  Duplicates: 0  Warnings: 0

MariaDB [Lab7DB]> SELECT * FROM Employees;
+--------+----------+---------+
| emp_id | emp_name | emp_zip |
+--------+----------+---------+
|      1 | Aiyana   |   28002 |
|      2 | Sita     |   28008 |
|      3 | Radha    |   28007 |
|      4 | Saru     |   28007 |
|      5 | Romi     |   28008 |
+--------+----------+---------+
5 rows in set (0.001 sec)

MariaDB [Lab7DB]> INSERT INTO Location (emp_zip, emp_state, emp_city, emp_district) VALUES
    -> (28002, 'State1', 'Pokhara', NULL),
    -> (28007, 'State3', 'Ktm', 'Kathmandu'),
    -> (28008, 'State2', 'Kaski', 'Janakpur');
Query OK, 3 rows affected (0.013 sec)
Records: 3  Duplicates: 0  Warnings: 0

MariaDB [Lab7DB]> SELECT * FROM Location;
+---------+-----------+---------+--------------+
| emp_zip | emp_state | emp_city | emp_district |
+---------+-----------+---------+--------------+
|   28002 | State1    | Pokhara | NULL         |
|   28007 | State3    | Ktm     | Kathmandu    |
|   28008 | State2    | Kaski   | Janakpur     |
+---------+-----------+---------+--------------+
3 rows in set (0.002 sec)
```

```
MariaDB [Lab7DB]> SELECT e.emp_id, e.emp_name, l.emp_city, l.emp_district, l.emp_state
    -> FROM Employees e
    -> LEFT JOIN Location l ON e.emp_zip = l.emp_zip;
+--------+----------+---------+--------------+-----------+
| emp_id | emp_name | emp_city | emp_district | emp_state |
+--------+----------+---------+--------------+-----------+
|      1 | Aiyana   | Pokhara | NULL         | State1    |
|      2 | Sita     | Kaski   | Janakpur     | State2    |
|      3 | Radha    | Ktm     | Kathmandu    | State3    |
|      4 | Saru     | Ktm     | Kathmandu    | State3    |
|      5 | Romi     | Kaski   | Janakpur     | State2    |
+--------+----------+---------+--------------+-----------+
5 rows in set (0.001 sec)

MariaDB [Lab7DB]>
```

5. Boyce-Codd Normal Form (BCNF / 3.5NF)

```
MariaDB [Lab7DB]> -- Students Table
MariaDB [Lab7DB]> CREATE TABLE BCNF_Students (
    ->     student_id INT PRIMARY KEY
    -> );
Query OK, 0 rows affected (0.030 sec)

MariaDB [Lab7DB]>
MariaDB [Lab7DB]> -- Subjects Table
MariaDB [Lab7DB]> CREATE TABLE BCNF_Subjects (
    ->     subject VARCHAR(50) PRIMARY KEY,
    ->     professor VARCHAR(50)
    -> );
Query OK, 0 rows affected (0.012 sec)

MariaDB [Lab7DB]>
MariaDB [Lab7DB]> -- Student_Subject Table
MariaDB [Lab7DB]> CREATE TABLE Student_Subject (
    ->     student_id INT,
    ->     subject VARCHAR(50),
    ->     FOREIGN KEY (student_id) REFERENCES BCNF_Students(student_id),
    ->     FOREIGN KEY (subject) REFERENCES BCNF_Subjects(subject)
    -> );
Query OK, 0 rows affected (0.082 sec)

MariaDB [Lab7DB]> INSERT INTO BCNF_Students (student_id) VALUES
    -> (101),(102),(103),(104);
Query OK, 4 rows affected (0.010 sec)
Records: 4  Duplicates: 0  Warnings: 0

MariaDB [Lab7DB]>
MariaDB [Lab7DB]> SELECT * FROM BCNF_Students;
+------------+
| student_id |
+------------+
|        101 |
|        102 |
|        103 |
|        104 |
+------------+
4 rows in set (0.009 sec)
```

```
MariaDB [Lab7DB]> INSERT INTO BCNF_Subjects (subject, professor) VALUES
    -> ('Java','P.Java'),
    -> ('C++','P.Cpp'),
    -> ('C#','P.Chash');
Query OK, 3 rows affected (0.009 sec)
Records: 3  Duplicates: 0  Warnings: 0

MariaDB [Lab7DB]>
MariaDB [Lab7DB]> SELECT * FROM BCNF_Subjects;
+----------+------------+
| subject  | professor  |
+----------+------------+
| C#       | P.Chash    |
| C++      | P.Cpp      |
| Java     | P.Java     |
+----------+------------+
3 rows in set (0.003 sec)

MariaDB [Lab7DB]> INSERT INTO Student_Subject (student_id, subject) VALUES
    -> (101,'Java'),
    -> (101,'C++'),
    -> (102,'Java'),
    -> (103,'C#'),
    -> (104,'Java');
Query OK, 5 rows affected (0.010 sec)
Records: 5  Duplicates: 0  Warnings: 0

MariaDB [Lab7DB]>
MariaDB [Lab7DB]> SELECT * FROM Student_Subject;
+-------------+---------+
| student_id  | subject |
+-------------+---------+
|         101 | Java    |
|         101 | C++     |
|         102 | Java    |
|         103 | C#      |
|         104 | Java    |
+-------------+---------+
5 rows in set (0.012 sec)
```

Join Practice on Normalized Tables.

```
MariaDB [Lab7DB]> SELECT st.student_id, su.subject, su.professor
    -> FROM Student_Subject stsu
    -> INNER JOIN BCNF_Students st ON stsu.student_id = st.student_id
    -> INNER JOIN BCNF_Subjects su ON stsu.subject = su.subject;
+------------+---------+-----------+
| student_id | subject | professor |
+------------+---------+-----------+
|        103 | C#      | P.Chash   |
|        101 | C++     | P.Cpp     |
|        101 | Java    | P.Java    |
|        102 | Java    | P.Java    |
|        104 | Java    | P.Java    |
+------------+---------+-----------+
5 rows in set (0.002 sec)

MariaDB [Lab7DB]> SELECT e.emp_name, l.emp_city, l.emp_district
    -> FROM Employees e
    -> LEFT JOIN Location l ON e.emp_zip = l.emp_zip;
+----------+----------+--------------+
| emp_name | emp_city | emp_district |
+----------+----------+--------------+
| Aiyana   | Pokhara  | NULL         |
| Sita     | Kaski    | Janakpur     |
| Radha    | Ktm      | Kathmandu    |
| Saru     | Ktm      | Kathmandu    |
| Romi     | Kaski    | Janakpur     |
+----------+----------+--------------+
5 rows in set (0.001 sec)

MariaDB [Lab7DB]> SELECT su.professor, stsu.student_id
    -> FROM Student_Subject stsu
    -> RIGHT JOIN BCNF_Subjects su ON stsu.subject = su.subject;
+-----------+------------+
| professor | student_id |
+-----------+------------+
| P.Java    |        101 |
| P.Cpp     |        101 |
| P.Java    |        102 |
| P.Chash   |        103 |
| P.Java    |        104 |
+-----------+------------+
5 rows in set (0.001 sec)
```

```
MariaDB [Lab7DB]> SELECT e.emp_name, l.emp_zip
    -> FROM Employees e
    -> LEFT JOIN Location l ON e.emp_zip = l.emp_zip
    -> UNION
    -> SELECT e.emp_name, l.emp_zip
    -> FROM Employees e
    -> RIGHT JOIN Location l ON e.emp_zip = l.emp_zip;
+----------+---------+
| emp_name | emp_zip |
+----------+---------+
| Aiyana   |   28002 |
| Sita     |   28008 |
| Radha    |   28007 |
| Saru     |   28007 |
| Romi     |   28008 |
+----------+---------+
5 rows in set (0.002 sec)

MariaDB [Lab7DB]> SELECT e1.emp_name AS Employee1, e2.emp_name AS Employee2, l.emp_city
    -> FROM Employees e1
    -> JOIN Employees e2 ON e1.emp_id <> e2.emp_id
    -> JOIN Location l ON e1.emp_zip = l.emp_zip AND e2.emp_zip = l.emp_zip;
+-----------+-----------+----------+
| Employee1 | Employee2 | emp_city |
+-----------+-----------+----------+
| Romi      | Sita      | Kaski    |
| Saru      | Radha     | Ktm      |
| Radha     | Saru      | Ktm      |
| Sita      | Romi      | Kaski    |
+-----------+-----------+----------+
4 rows in set (0.001 sec)

MariaDB [Lab7DB]>
```

Lab:8

Objective:To learn how to grant and revoke privileges for specific operations (like SELECT, INSERT, UPDATE, DELETE) on a particular table.

Theory:

GRANT command is used by the DBA to provide specific privileges to users on databases or tables.

REVOKE command is used to take back previously granted privileges.

Privileges can be at different levels (global, database, table, column).

Task:

Task:

1. Create a new database CompanyDB and table Employees with 3 sample records.

CREATE DATABASE CompanyDB; USE CompanyDB;

CREATE TABLE Employees (emp_id INT PRIMARY KEY, emp_name VARCHAR(50), salary DECIMAL(10,2));

INSERT INTO Employees VALUES (1, 'John Smith', 50000), (2, 'Ava Brown', 60000), (3, 'Liam Johnson', 55000);

2. Create two users: manager1 and staff1 with passwords.

CREATE USER 'manager1'@'localhost' IDENTIFIED BY 'Manager@123';

CREATE USER 'staff1'@'localhost' IDENTIFIED BY 'Staff@123';

3. Grant manager1 the privileges to SELECT and UPDATE the Employees table.

GRANT SELECT, UPDATE ON CompanyDB.Employees TO 'manager1'@'localhost';

4: Grant staff1 only SELECT privilege

GRANT SELECT ON CompanyDB.Employees TO 'staff1'@'localhost';

5: Revoke UPDATE Privilege from manager1.

REVOKE UPDATE ON CompanyDB.Employees FROM 'manager1'@'localhost';

6: Verify Again.

SHOW GRANTS FOR 'manager1'@'localhost';

Output (after revoke):

GRANT SELECT ON `CompanyDB`.`Employees` TO 'manager1'@'localhost';

7. Check privileges:

```
MariaDB [CompanyDB]> SHOW GRANTS FOR 'manager1'@'localhost';
+-----------------------------------------------------------------------------------------------------------------------+
| Grants for manager1@localhost                                                                                         |
+-----------------------------------------------------------------------------------------------------------------------+
| GRANT USAGE ON *.* TO `manager1`@`localhost` IDENTIFIED BY PASSWORD '*AF5F2434A1869B6197E49152E1DD2CFE5DFA1415' |
| GRANT SELECT, UPDATE ON `companydb`.`employees` TO `manager1`@`localhost`                                             |
+-----------------------------------------------------------------------------------------------------------------------+
2 rows in set (0.001 sec)

MariaDB [CompanyDB]> SHOW GRANTS FOR 'staff1'@'localhost';
+-----------------------------------------------------------------------------------------------------------------------+
| Grants for staff1@localhost                                                                                           |
+-----------------------------------------------------------------------------------------------------------------------+
| GRANT USAGE ON *.* TO `staff1`@`localhost` IDENTIFIED BY PASSWORD '*2206BC87090355B5AF4C61CDDCB64CEAA993280A' |
| GRANT SELECT ON `companydb`.`employees` TO `staff1`@`localhost`                                                       |
+-----------------------------------------------------------------------------------------------------------------------+
2 rows in set (0.000 sec)

MariaDB [CompanyDB]> REVOKE UPDATE ON CompanyDB.Employees FROM 'manager1'@'localhost';
Query OK, 0 rows affected (0.007 sec)

MariaDB [CompanyDB]> SHOW GRANTS FOR 'manager1'@'localhost';
+-----------------------------------------------------------------------------------------------------------------------+
| Grants for manager1@localhost                                                                                         |
+-----------------------------------------------------------------------------------------------------------------------+
| GRANT USAGE ON *.* TO `manager1`@`localhost` IDENTIFIED BY PASSWORD '*AF5F2434A1869B6197E49152E1DD2CFE5DFA1415' |
| GRANT SELECT ON `companydb`.`employees` TO `manager1`@`localhost`                                                     |
+-----------------------------------------------------------------------------------------------------------------------+
2 rows in set (0.001 sec)

MariaDB [CompanyDB]>
```

Conclusion:

We successfully demonstrated how to grant and revoke user privileges in MySQL/MariaDB.

Lab:9

Objective: To understand how to implement data integrity constraints using Assertions and Triggers in SQL.

Theory:

Assertions: Assertions are constraints that enforce business rules on the data in a database.They ensure that certain conditions are always true (e.g., balance ≥ 1000).

Triggers: A trigger is a special stored procedure that automatically executes in response to certain events on a table (INSERT, UPDATE, DELETE).Triggers can enforce data integrity, log changes, or implement complex business rules.Types of triggers: BEFORE or AFTER an event.Example: After updating an account balance, a trigger can automatically insert a record into a transactions log table.

Task for assertions.

1. Create the Accounts table and insert at least 3 sample records.

CREATE DATABASE BankDB; USE BankDB;

CREATE TABLE Accounts (acc_no INT PRIMARY KEY, acc_holder VARCHAR(50), balance DECIMAL(10,2) CHECK (balance >= 1000)); output:

acc_no	acc_holder	balance
101	John Smith	5000.00
102	Ava Brown	3000.00
103	Liam Johnson	1500.00

INSERT INTO Accounts VALUES(101, 'John Smith', 5000),

(102, 'Ava Brown', 3000),(103, 'Liam Johnson', 1500);

2. Write an Assertion to ensure that no account balance goes below 1000.

ALTER TABLE Accounts ADD CONSTRAINT balance_check CHECK (balance >= 1000);

3. Demonstrate by inserting/updating a record that violates the assertion.

INSERT INTO Accounts VALUES (104, 'Emma White', 500);

UPDATE Accounts SET balance = 500 WHERE acc_no = 101;

```
MariaDB [BankDB]> -- This will FAIL
MariaDB [BankDB]> INSERT INTO Accounts VALUES (104, 'Emma White', 500);
ERROR 4025 (23000): CONSTRAINT `accounts.balance` failed for `bankdb`.`accounts`
MariaDB [BankDB]>
MariaDB [BankDB]> -- This will also FAIL
MariaDB [BankDB]> UPDATE Accounts SET balance = 500 WHERE acc_no = 101;
ERROR 4025 (23000): CONSTRAINT `accounts.balance` failed for `bankdb`.`accounts`
MariaDB [BankDB]>
```

Task for triggers:

4. Create the Transactions table.

CREATE TABLE Transactions (t_id INT AUTO_INCREMENT PRIMARY KEY, acc_no INT,operation VARCHAR(20),amount DECIMAL(10,2), trans_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP);

5. Write an AFTER UPDATE trigger on Accounts so that whenever the balance is updated, a record is automatically inserted into the Transactions table.

```
DELIMITER $$

CREATE TRIGGER after_account_update ,AFTER UPDATE ON Accounts FOR EACH ROW

BEGIN

   DECLARE action VARCHAR(20);

  IF NEW.balance > OLD.balance THEN

      SET action = 'Deposit';

   ELSE

      SET action = 'Withdrawal';

   END IF;

INSERT INTO Transactions (acc_no, operation, amount)

   VALUES (NEW.acc_no, action, ABS(NEW.balance - OLD.balance));

END$$

DELIMITER ;
```

```
+------+--------+------------+---------+---------------------+
| t_id | acc_no | operation  | amount  | trans_date          |
+------+--------+------------+---------+---------------------+
|    1 |    101 | Deposit    | 2000.00 | 2025-08-31 14:07:17 |
|    2 |    102 | Withdrawal | 1000.00 | 2025-08-31 14:07:27 |
+------+--------+------------+---------+---------------------+
2 rows in set (0.001 sec)
```

6. Test the trigger by performing deposit (balance increase) and withdrawal (balance decrease) operations on Accounts.

```
UPDATE Accounts SET balance = balance + 2000 WHERE acc_no = 101;

UPDATE Accounts SET balance = balance - 1000 WHERE acc_no = 102;
```

acc_no	acc_holder	balance
101	John Smith	9000.00
102	Ava Brown	1000.00
103	Liam Johnson	1500.00

7: Verify Trigger Effect.

SELECT * FROM Transactions;

t_id	acc_no	operation	amount	trans_date
1	101	Deposit	2000.00	2025-08-31 14:07:17
2	102	Withdrawal	1000.00	2025-08-31 14:07:27
3	101	Deposit	2000.00	2025-08-31 14:09:49
4	102	Withdrawal	1000.00	2025-08-31 14:09:49

Conclusion:

Assertions/Check constraints enforce rules such as minimum balance, preventing invalid data entry.

Triggers automatically log operations, like deposits and withdrawals, without manual intervention.

Together, they ensure data integrity, consistency, and reliability in the database.