Department of Information and Technology

Gauhati University Institute of Science & Technology

Gauhati University, GNB Nagar,Ghy-14

LABORATORY REPORT

***Database Management Systems*(IT-311)**

Name: Bidipta Saikia

ROLL No: 190102006 **Semester:** Vth Semester

Branch: IT

INDEX

EXPERIMENT NO: 1
Introduction to MYSQL

AIM: To study about Mysql database

OBJECTIVES:

THEORY:
MySQL, is one of the most popular Open Source SQL database management systems. MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed, and supported by MySQL AB, which is a Swedish company.
MySQL is becoming so popular because of many good reasons:

MySQL is released under an open-source license. So you have nothing to pay to use it.

MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.

MySQL uses a standard form of the well-known SQL data language.

MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.

MySQL works very quickly and works well even with large data sets.

MySQL is very friendly to PHP, the most appreciated language for web development. MySQL supports large databases, up to 50 million rows or more in a table.
MySQL is customizable.

RESULT:

The mysql database is studied.

EXPERIMENT NO:2
DATA DEFINITION LANGUAGE (DDL) COMMANDS

AIM: Consider the database for an organisation. Write the queries for the following

(i) create the database

Ans: CREATE DATABASE organization5;
(ii) select the current database
Ans: USE organization5;

(iii)Create the following tables.
a. employee (emp_no, emp_name, DOB, address, doj, mobile_no,dept_no, salary).
b. department (dept_no, dept_name, location).
(iv)Include necessary constraints.

Ans:
a) CREATE TABLE employee (
emp_no INT PRIMARY KEY,
emp_name VARCHAR(20) NOT NULL,
DOB DATE,
address VARCHAR(50),
DOJ DATE,
mobile_no INT UNIQUE,
dept_no INT DEFAULT 1,
salary INT
);

b) CREATE TABLE department(
dept_no INT PRIMARY KEY auto_increment,
dept_name VARCHAR(30) NOT NULL,
location VARCHAR(30)
);
(v) List all the tables in the current database
Ans: SHOW TABLES;

```
MySQL  localhost:33060+ ssl  organization5  SQL > show tables;
+--------------------------+
| Tables_in_organization5  |
+--------------------------+
| department               |
| employee                 |
+--------------------------+
```

(vi)Display the structure of the employee table
Ans: DESCRIBE employee;

```
MySQL  localhost:33060+ ssl  organization5  SQL > desc employee;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| emp_no    | int         | NO   | PRI | NULL    |       |
| emp_name  | varchar(20) | NO   |     | NULL    |       |
| DOB       | date        | YES  |     | NULL    |       |
| address   | varchar(50) | YES  |     | NULL    |       |
| DOJ       | date        | YES  |     | NULL    |       |
| mobile_no | int         | YES  | UNI | NULL    |       |
| dept_no   | int         | YES  |     | 1       |       |
| salary    | int         | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
8 rows in set (0.0033 sec)
```

(vii) Add a new column Designation to the employee table

Ans: ALTER TABLE employee ADD Designation VARCHAR(20);
(viii) Drop the column location from Dept table
Ans: ALTER TABLE department DROP location;
(ix) Drop the tables
Ans: DROP table department;
DROP TABLE employee;
(x) Delete the database
Ans: DROP DATABASE organization5;

OBJECTIVES:
To understand DDL commands.

THEORY:

Database Queries:

Before creating any tables, MySQL requires you to create a database by executing the CREATE DATABASE command.

>Create a database

CREATE DATABASE <database name>

Delete a database DROP DATABASE <database name>

Select the database USE <database name>

List all databases SHOW databases;

Rename a database ALTER DATABASE <database name> RENAME <new database name>

Table Queries:

>To Create a table CREATE TABLE <tablename> (<fieldname>< fieldtype>(<fieldsize>) , …)
>List all tables in the current database SHOW tables;
>Show table format with column names and data types DESCRIBE <tablename>
>Modify the structure of table
ALTER TABLE <table name> <alter specifications>
ALTER TABLE <table name> DROP COLUMN <column name>
ALTER TABLE<table name> ADD COLUMN <column name> datatype>(<size>)
>Delete the table

DROP TABLE <table name>

Constraints:

>**Primary key** A PRIMARY KEY constraint for a table enforces the table to accept unique data for a specific column and this constraint create a unique index for accessing the table faster

>**UNIQUE** The UNIQUE constraint in Mysql does not allow to insert a duplicate value in a column.

>**NOT NULL** In Mysql NOT NULL constraint allows to specify that a column can not contain any NULL value.

>**FOREIGN KEY** A FOREIGN KEY in mysql creates a link between two tables by one specific column of both table. The specified column in one table must be a

>**PRIMARY KEY** and referred by the column of another table known as FOREIGN KEY.

>**CHECK** The CHECK constraint determines whether the value is valid or not from a logical expression.

>**DEFAULT** While inserting data into a table, if no value is supplied to a column, then the column gets the value set as DEFAULT

PROCEDURE:

(i) CREATE DATABASE command

(ii) USE DATABASE command

(iii) CREATE TABLE command

(iv) PRIMARY KEY,NOT NULL etc

(v) SHOW TABLES command

(vi) DESCRIBE TABLE command

(vii) ALTER TABLE command

(viii) ALTER TABLE command

(ix) DROP TABLE command

(x) DROP DATABASE command

RESULT:

The DDL commands have been executed successfully.

OUTPUT:

Problems:

1. Consider the database for a college and design an ER diagram. Write the query for the following.

(i) Create the tables:
Student (sid, sname, sex, dob,dno)

Department (dno, dname)

Faculty (F_id, fname, designation, salary,dno)

Course (cid, cname, credits,dno)

Register (sid,cid,sem)

Teaching (f_id,cid,sem)

Hostel(hid,hname,seats,)

(ii) Include the necessary constraints NOT NULL, DEFAULT, CHECK, and PRIMARY KEY, UNIQUE.

Ans.:

CREATE TABLE Student(sid INT PRIMARY KEY, sname VARCHAR(20) NOT NULL,sex VARCHAR(10), dob DATE, dno INT DEFAULT '1');

CREATE TABLE Department (dno INT PRIMARY KEY, dname VARCHAR(30) NOT NULL);

CREATE TABLE Faculty (F_id INT PRIMARY KEY, fname VARCHAR(20) NOT NULL,

designation VARCHAR(20) DEFAULT 'Professor', salary INT, dno INT DEFAULT '1');

CREATE TABLE Course (cid CHAR(10) PRIMARY KEY, cname VARCHAR(30) NOT NULL, credits INT, dno INT);

CREATE TABLE Register (sid INT PRIMARY KEY, cid CHAR(10) UNIQUE, sem INT NOT NULL, CHECK (sem<=8));

CREATE TABLE Teaching(f_id INT PRIMARY KEY, cid CHAR(10) UNIQUE, sem INT NOT NULL, CHECK (sem<=8));

CREATE TABLE Hostel(hid CHAR(10) PRIMARY KEY, hname VARCHAR(20) NOT NULL, seats INT DEFAULT '0');

(iii) Create a database college.

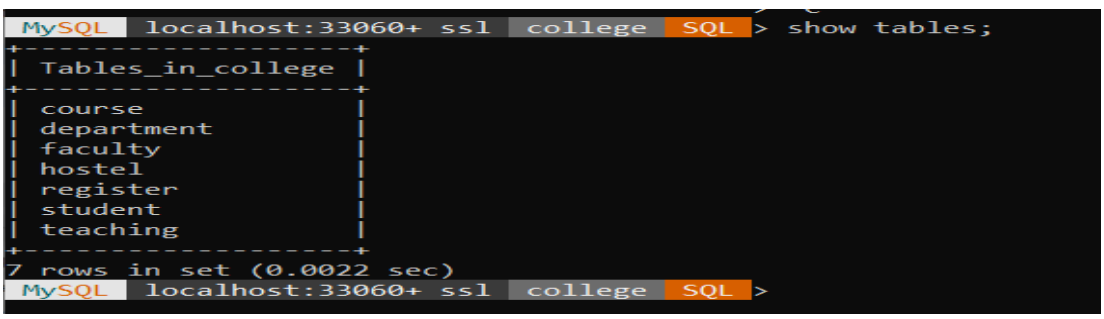
Ans.: CREATE DATABASE college;

(iv) Use college as the current database

Ans.: USE college;

(v) Display all the tables in college database

Ans.: SHOW TABLES;



(vi) Describe the structure of all tables.

Ans.:

Describe course,

```
MySQL localhost:33060+ ssl college SQL > describe course;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| cid     | char(10)    | NO   | PRI | NULL    |       |
| cname   | varchar(30) | NO   |     | NULL    |       |
| credits | int         | YES  |     | NULL    |       |
| dno     | int         | YES  |     | NULL    |       |
+---------+-------------+------+-----+---------+-------+
```

Describe department,

```
MySQL localhost:33060+ ssl college SQL > describe department;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| dno   | int         | NO   | PRI | NULL    |       |
| dname | varchar(30) | NO   |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
```

Describe faculty,

```
MySQL localhost:33060+ ssl college SQL > describe faculty;
+-------------+-------------+------+-----+-----------+-------+
| Field       | Type        | Null | Key | Default   | Extra |
+-------------+-------------+------+-----+-----------+-------+
| F_id        | int         | NO   | PRI | NULL      |       |
| fname       | varchar(20) | NO   |     | NULL      |       |
| designation | varchar(20) | YES  |     | Professor |       |
| salary      | int         | YES  |     | NULL      |       |
| dno         | int         | YES  |     | 1         |       |
+-------------+-------------+------+-----+-----------+-------+
```

Describe hostel,

```
MySQL localhost:33060+ ssl college SQL > describe hostel;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| hid   | char(10)    | NO   | PRI | NULL    |       |
| hname | varchar(20) | NO   |     | NULL    |       |
| seats | int         | YES  |     | 0       |       |
+-------+-------------+------+-----+---------+-------+
```

Describe register,

```
MySQL localhost:33060+ ssl college SQL > describe register;
+-------+----------+------+-----+---------+-------+
| Field | Type     | Null | Key | Default | Extra |
+-------+----------+------+-----+---------+-------+
| sid   | int      | NO   | PRI | NULL    |       |
| cid   | char(10) | YES  | UNI | NULL    |       |
| sem   | int      | NO   |     | NULL    |       |
+-------+----------+------+-----+---------+-------+
```

Describe student,

```
MySQL localhost:33060+ ssl college SQL > describe student;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| sid   | int         | NO   | PRI | NULL    |       |
| sname | varchar(20) | NO   |     | NULL    |       |
| sex   | varchar(10) | YES  |     | NULL    |       |
| dob   | date        | YES  |     | NULL    |       |
| dno   | int         | YES  |     | 1       |       |
+-------+-------------+------+-----+---------+-------+
```

Describe teaching,

```
MySQL  localhost:33060+ ssl  college  SQL > describe teaching;
+-------+----------+------+-----+---------+-------+
| Field | Type     | Null | Key | Default | Extra |
+-------+----------+------+-----+---------+-------+
| f_id  | int      | NO   | PRI | NULL    |       |
| cid   | char(10) | YES  | UNI | NULL    |       |
| sem   | int      | NO   |     | NULL    |       |
+-------+----------+------+-----+---------+-------+
```

(vii) Modify the student table to add a new field ‘grade’.

Ans. ALTER TABLE student ADD grade CHAR(2);

2. Consider the database for a banking enterprise. Write the queries for the below questions.

(i) Create the following tables

Table	Attributes
customer	cid,cname,loc,sex,dob
Bank_brn	bcode,bloc,bsate
Deposit	Dacno,dtype,ddate,damt
Loan	Lacno,ltype,ldate,lamt
Accounts_in	Bcode,cid
depositor	cid,dacno
borrower	cid,lacno

(ii). Include necessary constraints.

Ans:

CREATE TABLE customer (cid INT PRIMARY KEY, cname VARCHAR(20) NOT NULL, loc VARCHAR(30), sex VARCHAR(10), dob DATE);

CREATE TABLE Bank_brn (bcode VARCHAR(20) PRIMARY KEY, bloc VARCHAR(30), bstate VARCHAR(20));

CREATE TABLE Deposit (Dacno INT PRIMARY KEY, dtype VARCHAR(20), ddate DATE, damt DECIMAL(9,2) NOT NULL);

CREATE TABLE Loan (Lacno INT PRIMARY KEY, ltype VARCHAR(20), ldate DATE, lamt DECIMAL(10,2) NOT NULL);

CREATE TABLE Accounts_in (Bcode VARCHAR(20) PRIMARY KEY, cid INT NOT NULL);

CREATE TABLE depositor (cid INT PRIMARY KEY, dacno INT UNIQUE);

CREATE TABLE borrower (cid INT PRIMARY KEY, lacno INT UNIQUE);

(iii). Tables are created under the database ‘bank’.

Ans:

CREATE DATABASE bank;
USE bank;

(iv). Display all the tables in bank database.

Ans:

```
MySQL  localhost:33060+ ssl  banking  SQL > show tables;
+------------------+
| Tables_in_banking |
+------------------+
| accounts_in      |
| bank_brn         |
| borrower         |
| customer         |
| deposit          |
| depositor        |
| loan             |
+------------------+
```

(v). Describe the structure of all tables.

Ans:

Describe accounts_in,

```
MySQL  localhost:33060+ ssl  banking  SQL > describe accounts_in;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| Bcode | varchar(20) | NO   | PRI | NULL    |       |
| cid   | int         | NO   |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
```

Describe bank_brn,

```
MySQL  localhost:33060+ ssl  banking  SQL > describe bank_brn;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| bcode  | varchar(20) | NO   | PRI | NULL    |       |
| bloc   | varchar(30) | YES  |     | NULL    |       |
| bstate | varchar(20) | YES  |     | NULL    |       |
+--------+-------------+------+-----+---------+-------+
```

Describe borrower,

```
MySQL  localhost:33060+ ssl  banking  SQL > describe borrower;
+-------+------+------+-----+---------+-------+
| Field | Type | Null | Key | Default | Extra |
+-------+------+------+-----+---------+-------+
| cid   | int  | NO   | PRI | NULL    |       |
| lacno | int  | YES  | UNI | NULL    |       |
+-------+------+------+-----+---------+-------+
```

Describe customer,

```
MySQL  localhost:33060+ ssl  banking  SQL > describe customer;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| cid   | int         | NO   | PRI | NULL    |       |
| cname | varchar(20) | NO   |     | NULL    |       |
| loc   | varchar(30) | YES  |     | NULL    |       |
| sex   | varchar(10) | YES  |     | NULL    |       |
| dob   | date        | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
```

Describe deposit,

```
MySQL  localhost:33060+ ssl  banking  SQL > describe deposit;
+-------+--------------+------+-----+---------+-------+
| Field | Type         | Null | Key | Default | Extra |
+-------+--------------+------+-----+---------+-------+
| Dacno | int          | NO   | PRI | NULL    |       |
| dtype | varchar(20)  | YES  |     | NULL    |       |
| ddate | date         | YES  |     | NULL    |       |
| damt  | decimal(9,2) | NO   |     | NULL    |       |
+-------+--------------+------+-----+---------+-------+
```

Describe depositor,

```
MySQL  localhost:33060+ ssl  banking  SQL > describe depositor;
-------+------+------+-----+---------+-------+
 Field | Type | Null | Key | Default | Extra |
-------+------+------+-----+---------+-------+
 cid   | int  | NO   | PRI | NULL    |       |
 dacno | int  | YES  | UNI | NULL    |       |
-------+------+------+-----+---------+-------+
```

Describe loan,

```
MySQL  localhost:33060+ ssl  banking  SQL > describe loan;
-------+---------------+------+-----+---------+-------+
 Field | Type          | Null | Key | Default | Extra |
-------+---------------+------+-----+---------+-------+
 Lacno | int           | NO   | PRI | NULL    |       |
 ltype | varchar(20)   | YES  |     | NULL    |       |
 ldate | date          | YES  |     | NULL    |       |
 lamt  | decimal(10,2) | NO   |     | NULL    |       |
-------+---------------+------+-----+---------+-------+
```

(vi). Delete tables

Ans:

```
DROP TABLE Bank_brn;
DROP TABLE Deposit;
DROP TABLE customer;
DROP TABLE Loan;
DROP TABLE Accounts_in;
DROP TABLE depositor;
DROP TABLE borrower;
```

EXPERIMENT NO 3

DATA MANIPULATION LANGUAGE(DML)

AIM:

Consider the database for an organisation. Write the queries for the following.

(i) Add 5 rows in the employee and dept tables.

Ans:

```
CREATE TABLE Employee(
emp_no INT PRIMARY KEY,
emp_name VARCHAR(20) NOT NULL,
DOB DATE,
DOJ DATE,
address VARCHAR(30),
mobile_no INT UNIQUE,
D_no INT NOT NULL,
designation VARCHAR(30),
salary INT
);

INSERT INTO employee VALUES(1, "Bidipta",20060506,20210102, "Jorhat", 785662, 1, " Officer", 50000);
INSERT INTO employee
VALUES(2, "Manikangkan", 19990506, 20200805, "Golaghat", 866434, 3, "HR", 15000);
INSERT INTO employee
VALUES(3, "Bhupen", 19891105,20210805, " New Delhi", 643461, 2, "Officer", 25000);
INSERT INTO employee
VALUES(4, "Pulak", 19850805,20200905, " Mumbai", 664007, 3, "Girlscript Manager", 1000);
INSERT INTO employee
VALUES(5, "Gaurav", 19280609, 20150707, " Mumbai", 640565, 1, "CEO", 55000);

CREATE TABLE department(
dept_no INT PRIMARY KEY AUTO_INCREMENT,
dept_name VARCHAR(30) NOT NULL,
location VARCHAR(30)
);

INSERT INTO department(dept_name, location) VALUES("Human Resources", "Surya Nagar");
INSERT INTO department(dept_name, location) VALUES("Information Technology", "K.B. Road");
INSERT INTO department(dept_name, location) VALUES("Accounting and Finance", "Nahoroni Complex");
```

(ii) Display all the records from the above tables.

Ans:

```
SELECT * FROM employee;
```

```
MySQL  localhost:33060+ ssl  organisation  SQL > SELECT * FROM employee;
+--------+-------------+------------+------------+------------+-----------+------+--------------------+--------+
| emp_no | emp_name    | DOB        | DOJ        | address    | mobile_no | D_no | designation        | salary |
+--------+-------------+------------+------------+------------+-----------+------+--------------------+--------+
|      1 | Bidipta     | 2006-05-06 | 2021-01-02 | Jorhat     |    785662 |    1 |  Officer           |  50000 |
|      2 | Manikangkan | 1999-05-06 | 2020-08-05 | Golaghat   |    866434 |    3 | HR                 |  15000 |
|      3 | Bhupen      | 1989-11-05 | 2021-08-05 |  New Delhi |    643461 |    2 | Officer            |  25000 |
|      4 | Pulak       | 1985-08-05 | 2020-09-05 |  Mumbai    |    664007 |    3 | Girlscript Manager |   1500 |
|      5 | Gaurav      | 1928-06-09 | 2015-07-07 |  Mumbai    |    640565 |    1 | typist             |  55000 |
+--------+-------------+------------+------------+------------+-----------+------+--------------------+--------+
```

SELECT * FROM department;

```
MySQL  localhost:33060+ ssl  organisation  SQL > select * from department;
+---------+-------------------------+-------------------+
| dept_no | dept_name               | location          |
+---------+-------------------------+-------------------+
|       1 | Human Resources         | Surya Nagar       |
|       2 | Information Technology  | K.B. Road         |
|       3 | Accounting and Finance  | Nahoroni Complex  |
|       4 | Research & Development  | Lyon Estates      |
|       5 | Marketing               | FCI Road          |
+---------+-------------------------+-------------------+
```

(iii) Display the empno and name of all the employees from department no 2.

Ans:

SELECT emp_no, emp_name FROM employee where D_no = 2;

```
MySQL  localhost:33060+ ssl  organisation  SQL > select emp_no, emp_name from employee where D_no = 2;
+--------+----------+
| emp_no | emp_name |
+--------+----------+
|      3 | Bhupen   |
+--------+----------+
```

(iv) Display empno,name,designation,dept no and salary in the descending order of salary.

Ans:

SELECT emp_no, emp_name, designation, D_no, salary FROM employee ORDER BY salary DESC;

```
MySQL  localhost:33060+ ssl  organisation  SQL > select emp_no, emp_name, designation, D_no, salary from employee order
by salary desc;
+--------+------------+-------------------+------+--------+
| emp_no | emp_name   | designation       | D_no | salary |
+--------+------------+-------------------+------+--------+
|      5 | Gaurav     | typist            |    1 |  55000 |
|      1 | Bidipta    |  Officer          |    1 |  50000 |
|      3 | Bhupen     | Officer           |    2 |  25000 |
|      2 | Manikangkan | HR               |    3 |  15000 |
|      4 | Pulak      | Girlscript Manager |    3 |   1500 |
+--------+------------+-------------------+------+--------+
```

(v) Display the empno and name of all employees whose salary is between 2000 and 5000.

Ans:

SELECT emp_no, emp_name FROM employee WHERE salary >= 2000 AND salary <= 5000;

```
MySQL  localhost:33060+ ssl  organisation  SQL > select emp_no,emp_name from employee where salary>=2000 and salary<=5000;
Empty set (0.0010 sec)
MySQL  localhost:33060+ ssl  organisation  SQL >
```

(vi) Display all designations without duplicate values.

Ans:

SELECT DISTINCT designation FROM employee;

```
MySQL  localhost:33060+ ssl  organisation  SQL > select distinct designation from employee;
+--------------------+
| designation        |
+--------------------+
|  Officer           |
| HR                 |
| Officer            |
| Girlscript Manager |
| typist             |
+--------------------+
```

(vii) Display the dept name and total salary of employees of each department.

Ans:

SELECT dept_name, SUM(salary) AS "Total salary" FROM department D, employee E WHERE E.D_no = D.Dept_no GROUP BY Dept_no;

```
MySQL  localhost:33060+ ssl  organisation  SQL > SELECT dept_name, SUM(salary) AS "Total salary" FROM department D, emp
loyee E WHERE E.D_no = D.Dept_no GROUP BY Dept_no;
+-----------------------+--------------+
| dept_name             | Total salary |
+-----------------------+--------------+
| Human Resources       |       105000 |
| Information Technology |        25000 |
| Accounting and Finance |        16500 |
+-----------------------+--------------+
```

(viii) Change the salary of employees to 25000 whose designation is ‘Typist’.

Ans:

UPDATE employee SET salary = 25000 WHERE designation = 'Typist';

(ix) Change the mobile no of employee named ‘john’.

Ans:

UPDATE employee SET mobile_no = 9649825 WHERE emp_name = "John";

(x) Delete all employees whose salaries are equal to Rs.7000.

Ans:

DELETE FROM employee WHERE salary = 7000;

(xi) Select the department that has total salary paid for its employees more than 25000.

Ans:

SELECT Dept_name, SUM(salary) AS "Total salary" FROM department D, employee E WHERE D.Dept_no = E.D_no GROUP BY D_no HAVING Sum(salary) > 25000;

```
MySQL  localhost:33060+ ssl  organisation  SQL > SELECT Dept_name, SUM(salary) AS "Total salary" FROM department D, emp
loyee E WHERE D.Dept_no = E.D_no GROUP BY D_no HAVING Sum(salary) > 25000;
+-----------------+--------------+
| Dept_name       | Total salary |
+-----------------+--------------+
| Human Resources |        75000 |
+-----------------+--------------+
```

OBJECTIVES:

- To understand how to insert, update and delete data from within a table.
- To learn how to retrieve data from a table using the SELECT statement.

THEORY:

1. INSERT

INSERT INTO *tablename* VALUES (*value1, value2, ..., valuen*).

2. UPDATE

UPDATE <table> SET <field1> = <value1> AND <field2> = <value2> WHERE <conditions>

3. DELETE

DELETE FROM <table> WHERE <condition>

4. SELECT

a) Retrieve from all columns

SELECT * FROM <table>

b) Retrieve from selected columns

SELECT <column 1>, <column 2> FROM <table>

c) Retrieve unique values

SELECT DISTINCT <column name> FROM <table>

d) Retrieve data satisfying a given condition

SELECT <columns> FROM <tables> WHERE <condition>

PROCEDURE:

(i) Use insert command
(ii) Use Select command
(iii) Use Select command with where condition
(iv)Use Select command with order by clause
(v) Use Select command with operators
(vi)Use Select command with DISTINCT keyword
(vii) Use Select command with group by clause
(viii) Use Update command
(ix)Use Update command
(x) Use Delete command
(xi)Use select command with group by and having clause

OUTPUT:

Problems:

1. Consider the database for a college. Write the query for the following.
(i) Insert at least 5 tuples into each table.
Ans:

```
CREATE TABLE Student(
sid INT PRIMARY KEY,
sname VARCHAR(20) NOT NULL,
dob DATE,
address VARCHAR(20) NOT NULL,
gender VARCHAR(10),
mobile_no INT UNIQUE,
dept_no INT DEFAULT '1'
);

INSERT INTO Student VALUES(1, "Bidipta", 19940702, "Golaghat", "Male", 8767893, 2);
INSERT INTO Student VALUES(2, "Saikia",  19930804, "Adabari", "Male", 76236011, 1);
INSERT INTO Student VALUES(3, "Manikangkan", 19951104, "Hill Valley", "Female", 8976542, 2);
INSERT INTO Student VALUES(4, "Bhupen ",  19970212, "Nalbari", "Male", 7234561, 3);
INSERT INTO Student VALUES(5, "Pulak",  19960311, "Rangia", "Female", 9231456, 4);
INSERT INTO Student(sid,sname,dob,address,gender, mobile_no) VALUES(6, "Chris",  19950915, "Jalukbari", "Male",
7645321);
INSERT INTO Student VALUES(7, "Jaaz",  19990619, "Golaghat", "Male", 7543212, 5);
INSERT INTO Student VALUES(8, "Franklin",  19950115, "Nogaon", "Female", 86543298, 6);
INSERT INTO Student VALUES(9, "Marcus",  19980518, "Bokaghat", "Male", 78432765, 1);

CREATE TABLE Department (D_no INT NOT NULL, dept_name VARCHAR(30) NOT NULL);

INSERT INTO Department VALUES(1, "Computer");
INSERT INTO Department VALUES(2, "ECE");
INSERT INTO Department VALUES(3, "Biotechnology");
INSERT INTO Department VALUES(4, "Mechanical");
INSERT INTO Department VALUES(5, "Law");
INSERT INTO Department VALUES(6, "Literature");
```

```
CREATE TABLE Faculty (F_id INT PRIMARY KEY, fname VARCHAR(20) NOT NULL, designation VARCHAR(20)
DEFAULT 'Professor', salary INT, dept_no INT DEFAULT '1');

INSERT INTO Faculty VALUES(01, "ARJUN SAHA", "Professor", 28000, 2);
INSERT INTO Faculty VALUES(02, "RIMA DEV", "Assistant Professor", 18000, 1);
INSERT INTO Faculty(F_id, fname, salary, dept_no) VALUES(03, "RAJESH ROY", 16000, 2);
INSERT INTO Faculty(F_id, fname, salary) VALUES(04, "RAHUL DEV", 20000);
INSERT INTO Faculty VALUES(05, "PRIYA DUTTA", "Associate Professor", 30000, 3);
INSERT INTO Faculty VALUES(06, "DINESH JAIN", "Assistant Professor", 23000, 5);
INSERT INTO Faculty VALUES(07, "ROHAN DUBEY", "Professor", 23000, 6);
INSERT INTO Faculty VALUES(08, "BHASKAR SAIKIA", "Assistant Professor", 29000, 2);
```

(ii) List the details of students in the ascending order of date of birth

Ans:

SELECT * FROM Student ORDER BY dob;

```
MySQL  localhost:33060+ ssl  collegedb  SQL > SELECT * FROM Student ORDER BY dob;
+-----+-------------+------------+-------------+--------+-----------+----------+
| sid | sname       | dob        | address     | gender | mobile_no | dept_no  |
+-----+-------------+------------+-------------+--------+-----------+----------+
|   2 | Saikia      | 1993-08-04 | Adabari     | Male   |  76236011 |        1 |
|   1 | Bidipta     | 1994-07-02 | Golaghat    | Male   |   8767893 |        2 |
|   8 | Franklin    | 1995-01-15 | Nogaon      | Female |  86543298 |        6 |
|   6 | Chris       | 1995-09-15 | Jalukbari   | Male   |   7645321 |        1 |
|   3 | Manikangkan | 1995-11-04 | Hill Valley | Female |   8976542 |        2 |
|   5 | Pulak       | 1996-03-11 | Rangia      | Female |   9231456 |        4 |
|   4 | Bhupen      | 1997-02-12 | Nalbari     | Male   |   7234561 |        3 |
|   9 | Marcus      | 1998-05-18 | Bokaghat    | Male   |  78432765 |        1 |
|   7 | Jaaz        | 1999-06-19 | Golaghat    | Male   |   7543212 |        5 |
+-----+-------------+------------+-------------+--------+-----------+----------+
```

(iii) Display the details of students from computer department

Ans:

SELECT S.*, D.dept_name FROM Student S, department D WHERE S.dept_no = D.D_no GROUP BY sid HAVING Dept_name = "Computer";

```
MySQL  localhost:33060+ ssl  collegedb  SQL > SELECT S.*, D.dept_name FROM Student S, department D WHERE S.dept_no = D.
D_no GROUP BY sid HAVING Dept_name = "Computer";
+-----+--------+------------+----------+--------+-----------+---------+-----------+
| sid | sname  | dob        | address  | gender | mobile_no | dept_no | dept_name |
+-----+--------+------------+----------+--------+-----------+---------+-----------+
|   2 | Saikia | 1993-08-04 | Adabari  | Male   |  76236011 |       1 | Computer  |
|   6 | Chris  | 1995-09-15 | Jalukbari| Male   |   7645321 |       1 | Computer  |
|   9 | Marcus | 1998-05-18 | Bokaghat | Male   |  78432765 |       1 | Computer  |
+-----+--------+------------+----------+--------+-----------+---------+-----------+
```

(iv) List the faculties in the descending order of salary

Ans:

SELECT * FROM Faculty ORDER BY salary DESC;

```
MySQL  localhost:33060+ ssl  collegedb  SQL > SELECT * FROM Faculty ORDER BY salary DESC;
+------+----------------+---------------------+--------+---------+
| F_id | fname          | designation         | salary | dept_no |
+------+----------------+---------------------+--------+---------+
|    5 | PRIYA DUTTA    | Associate Professor |  30000 |       3 |
|    8 | BHASKAR SAIKIA | Assistant Professor |  29000 |       2 |
|    1 | ARJUN SAHA     | Professor           |  28000 |       2 |
|    6 | DINESH JAIN    | Assistant Professor |  23000 |       5 |
|    7 | ROHAN DUBEY    | Professor           |  23000 |       6 |
|    4 | RAHUL DEV      | Professor           |  20000 |       1 |
|    2 | RIMA DEV       | Assistant Professor |  18000 |       1 |
|    3 | RAJESH ROY     | Professor           |  16000 |       2 |
+------+----------------+---------------------+--------+---------+
```

(v) Display the total number of students in each department

Ans:

SELECT d_no, dept_name, COUNT(*) as 'Total No. of students' FROM department D, student S WHERE S.dept_no = D.D_no GROUP BY dept_name;

```
MySQL  localhost:33060+ ssl  collegedb  SQL > SELECT d_no,
nt D, student S WHERE S.dept_no = D.D_no GROUP BY dept_name
+------+---------------+-----------------------+
| d_no | dept_name     | Total No. of students |
+------+---------------+-----------------------+
|    2 | ECE           |                     2 |
|    1 | Computer      |                     3 |
|    3 | Biotechnology |                     1 |
|    4 | Mechanical    |                     1 |
|    5 | Law           |                     1 |
|    6 | Literature    |                     1 |
+------+---------------+-----------------------+
```

(vi) Display the total number of faculties in each department with salary greater than 25000.

Ans:

SELECT dept_no, COUNT(*) as "No. of Faculties with Salary > 25000" FROM faculty WHERE salary > 25000 GROUP BY dept_no;

```
MySQL  localhost:33060+ ssl  collegedb  SQL > SELECT dept_no, COUNT(*) as
aculty
                                        -> WHERE salary > 25000 GROUP B
+---------+-------------------------------------+
| dept_no | No. of Faculties with Salary > 25000 |
+---------+-------------------------------------+
|       2 |                                   2 |
|       3 |                                   1 |
+---------+-------------------------------------+
```

2. Consider the database for a banking enterprise. Write the queries for the below questions.

(i) Insert at least 5 tuples in each table

Ans:

```
CREATE TABLE
customer (c_id INT PRIMARY KEY,
c_name VARCHAR(20) NOT NULL,
ph_no INT ,
dob DATE,
gender VARCHAR(10),
C_city VARCHAR(20)
);

INSERT INTO customer VALUES(01, "Rita Sinha", 9786853, 19870209, "Female", "Kolkata");
INSERT INTO customer VALUES(02, "Manish Bora", 9786151, 19890511, "Male", "Mumbai");
INSERT INTO customer VALUES(03, "Franklin Marshal", 78964311, 19910612, "Male", "Chennai");
INSERT INTO customer VALUES(04, "Ritu Devi", 88765423, 19930418, "Female", "Mumbai");
INSERT INTO customer VALUES(05, "Priyam Goyal", 98435425, 19880520, "Female", "Guwahati");
INSERT INTO customer VALUES(06, "Rajdeep Das", 83567095, 19950223, "Male", "Kolkata");
INSERT INTO customer VALUES(07, "Shubham Anand", 9876532, 19900420, "Male", "Mumbai");
INSERT INTO customer VALUES(08, "Bishnu Samuel", 8355435, 19920325, "Male", "Kolkata");
INSERT INTO customer VALUES(09, "Anurag Sharma", 9876212, 19870402, "Male", "Delhi");
```

CREATE TABLE Bank_brn (b_code VARCHAR(20) PRIMARY KEY, b_loc VARCHAR(30), b_state VARCHAR(20));

INSERT INTO Bank_brn VALUES('B001','MG Road', 'Maharashtra');
INSERT INTO Bank_brn VALUES('B002','Chitranjan Park','Tamil Nadu');
INSERT INTO Bank_brn VALUES('B003','Defence Market','Punjab');
INSERT INTO Bank_brn VALUES('B004','Derawal Nagal','Maharashtra');
INSERT INTO Bank_brn VALUES('B005','Ashok Nagar','Maharashtra');
INSERT INTO Bank_brn VALUES('B006','Dharapur','Assam');
INSERT INTO Bank_brn VALUES('B007','Raja Maidan Road','Assam');
INSERT INTO Bank_brn VALUES('B008','Rajendra Nagar','Assam');
INSERT INTO Bank_brn VALUES('B009','Sahupur','Bihar');

CREATE TABLE D_scheme(c_id INT PRIMARY KEY, dep_scheme VARCHAR(30) NOT NULL);

INSERT INTO D_scheme VALUES(1, "National Savings Certificate"),
(2, "Post Office Savings Account"),
(3, "Public Provident Fund"),
(4, "Post Office Time Deposit"),
(5, "Sukanya Smriddhi Yojana");

(ii) Display the branch details

Ans:

SELECT * FROM Bank_brn;

```
MySQL  localhost:33060+ ssl  bankingdb  SQL > SELECT * FROM Ba
+--------+------------------+-------------+
| b_code | b_loc            | b_state     |
+--------+------------------+-------------+
| B001   | MG Road          | Maharashtra |
| B002   | Chitranjan Park  | Tamil Nadu  |
| B003   | Defence Market   | Punjab      |
| B004   | Derawal Nagal    | Maharashtra |
| B005   | Ashok Nagar      | Maharashtra |
| B006   | Dharapur         | Assam       |
| B007   | Raja Maidan Road | Assam       |
| B008   | Rajendra Nagar   | Assam       |
| B009   | Sahupur          | Bihar       |
+--------+------------------+-------------+
```

(iii)List the customers of ‘Mumbai’ city

Ans:

SELECT c_id, c_name FROM customer WHERE c_city = "Mumbai";

```
MySQL  localhost:33060+ ssl  bankingdb  SQL > SEL
+------+---------------+
| c_id | c_name        |
+------+---------------+
|    2 | Manish Bora   |
|    4 | Ritu Devi     |
|    7 | Shubham Anand |
+------+---------------+
```

(iv) List the male customers of ‘Kolkata’city

Ans:
SELECT c_id, c_name FROM customer WHERE gender = "Male" AND c_city = 'Kolkata';

```
MySQL  localhost:33060+ ssl  bankingdb  SQL
+------+---------------+
| c_id | c_name        |
+------+---------------+
|    6 | Rajdeep Das   |
|    8 | Bishnu Samuel |
+------+---------------+
```

(v) List the state having more than one branch.

Ans:
SELECT b_state, COUNT(b_state) AS “Total no. of branches” FROM bank_brn GROUP BY b_state HAVING COUNT(b_state) > 1;

```
+------------+----------------------+
| b_state    | Total no. of branches |
+------------+----------------------+
| Maharashtra |                    3 |
| Assam      |                    3 |
+------------+----------------------+
2 rows in set (0.00 sec)
```

(vi) List the deposit schemes provided by the bank to the customers
Ans:
SELECT * FROM D_scheme;

```
MySQL  localhost:33060+ ssl  bankingdb  SQL >
+------+-------------------------------+
| c_id | dep_scheme                    |
+------+-------------------------------+
|    1 | National Savings Certificate  |
|    2 | Post Office Savings Account   |
|    3 | Public Provident Fund         |
|    4 | Post Office Time Deposit      |
|    5 | Sukanya Smriddhi Yojana       |
+------+-------------------------------+
```

(vii) Delete the entire content of any table
Ans:
DROP TABLE Bank_brn;

```
MySQL  localhost:33060+ ssl  bankingdb  SQL > DROP TABLE Bank_brn;
Query OK, 0 rows affected (0.0307 sec)
MySQL  localhost:33060+ ssl  bankingdb  SQL >
```

EXPERIMENT NO 4

Sub Queries and Joins

AIM:

Consider the database for the organization and Write the queries for the following

(i) Display the empno, name, and salaries for employees whose average salary is higher than the average salary of the organization.

(ii) Display the details of employees whose salary is equal to the minimum salary of organisation.

(iii) Display all the employees whose designation is same as that of 'Arun'

(iv) display the empno and name of employees who earn more than any Employee in dept 1.
(v) Display the empno, name, departments that the departments are same in both the emp and dept

(vi)Display the employee details by implementing left inner join

(vii) Display employee details by implementing a right outer join

OBJECTIVES:

To understand sub queries and join in Mysql.

THEORY:

NESTED QUERIES: A sub query is a query within a query. These sub queries can reside in the WHERE clause, the FROM clause, or the SELECT clause. The first query in the SQL statement is known as the outer query. The query inside the SQL statement is known as the inner query. The inner query is executed first. The output of an inner query is used as the input for the outer query. The entire SQL statement is sometimes referred to as a nested query.

JOINS: MySQL JOINS are used to retrieve data from multiple tables. A MySQL JOIN is performed whenever two or more tables are joined in a SQL statement.

There are different types of MySQL joins:

- MySQL INNER JOIN (or sometimes called simple join) MySQL LEFT
- OUTER JOIN (or sometimes called LEFT JOIN)
- MySQL RIGHT OUTER JOIN (or sometimes called RIGHT JOIN)

INNER JOIN (simple join):
MySQL INNER JOINS return all rows from multiple tables where the join condition is met.

Syntax:

Select columns from table1 Inner join table2

On table1.column=table2.column;

LEFT OUTER JOIN

Another type of join is called a MySQL LEFT OUTER JOIN. This type of join returns all rows from the LEFT-hand table specified in the ON condition and only those rows from the other table where the joined fields are equal.

Syntax:

Select columns from table left join table2

On table1.column=table2.column;

RIGHT OUTER JOIN

Another type of join is called a MySQL RIGHT OUTER JOIN. This type of join returns all rows from the RIGHT-hand table specified in the ON condition and only those rows from the other table where the joined fields are equal.

Syntax:

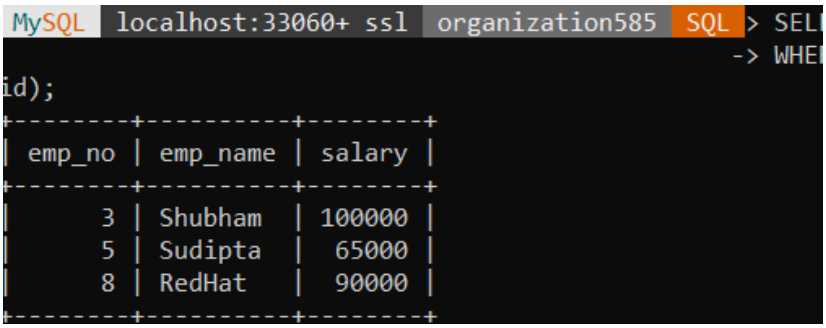
Select columns from table 1

Right join table2

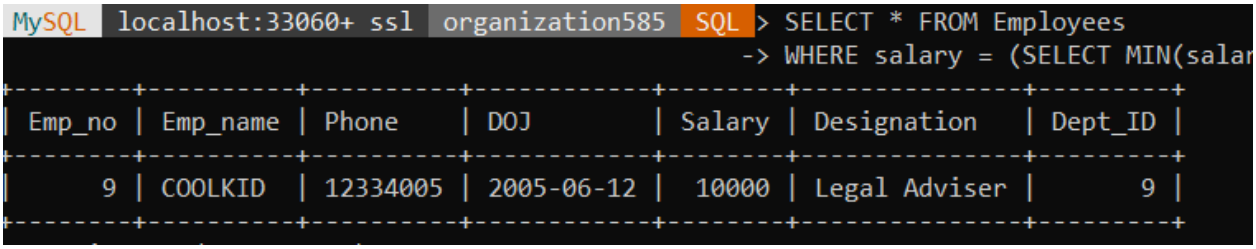
On table1.column=table2.column;

CODE AND OUTPUT:

(i). SELECT emp_no, emp_name, salary FROM Employees
WHERE salary > ALL(SELECT avg(salary) FROM Employees group by Dept_id);



(ii). SELECT * FROM Employees
WHERE salary = (SELECT MIN(salary) FROM employees);



(iii) SELECT emp_name FROM Employees WHERE Designation =
(SELECT Designation FROM Employees WHERE Emp_name = "Arun");

```
mysql> SELECT emp_name FROM Employees WHERE Designation =
    -> (SELECT Designation FROM Employees WHERE Emp_name = "Arun");
+-----------+
| emp_name  |
+-----------+
| Arun      |
| Riya      |
| Tom       |
+-----------+
3 rows in set (0.00 sec)
```

(iv).SELECT Emp_no, Emp_name FROM Employees WHERE
salary>(SELECT MAX(salary) FROM Employees WHERE Dept_id = 1);

```
+--------+----------+
| Emp_no | Emp_name |
+--------+----------+
|      3 | Shubham  |
+--------+----------+
1 row in set (0.0009 sec)
MySQL  localhost:33060+ ssl  organization585  SQL >
```

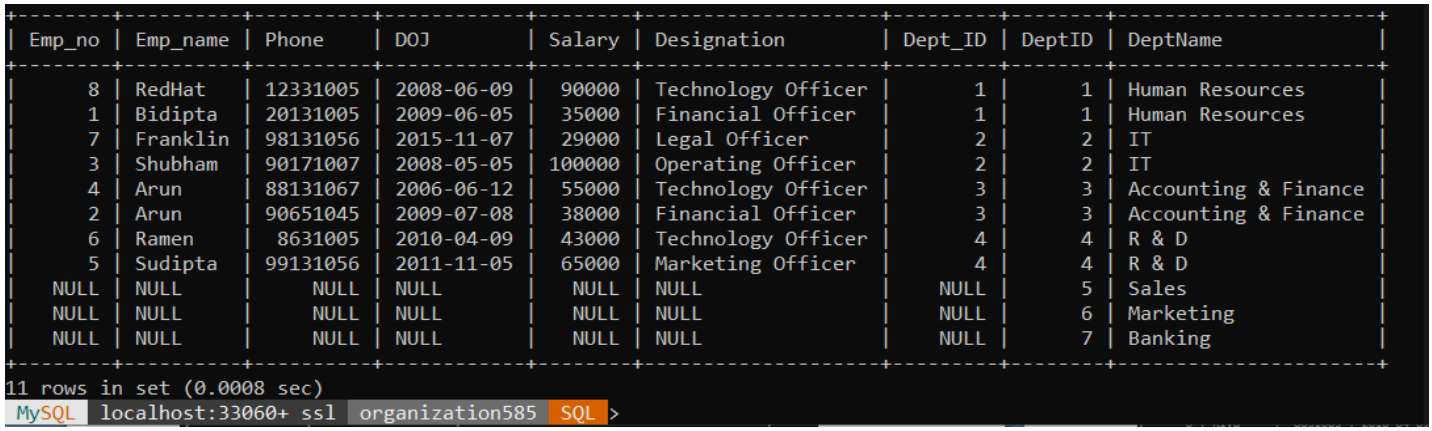
(v).SELECT emp_no, Emp_name, DeptName
FROM Employees
INNER JOIN Dept ON Dept_ID = DeptID;

```
                                                    -> IN
+--------+----------+----------------------+
| emp_no | Emp_name | DeptName             |
+--------+----------+----------------------+
|      1 | Bidipta  | Human Resources      |
|      2 | Arun     | Accounting & Finance |
|      3 | Shubham  | IT                   |
|      4 | Arun     | Accounting & Finance |
|      5 | Sudipta  | R & D                |
|      6 | Ramen    | R & D                |
|      7 | Franklin | IT                   |
|      8 | RedHat   | Human Resources      |
+--------+----------+----------------------+
8 rows in set (0.0018 sec)
MySQL  localhost:33060+ ssl  organization585  SQL >
```

(vi).SELECT * FROM Employees
LEFT JOIN dept
ON Employees.dept_ID = Dept.DeptID;

```
+--------+----------+----------+------------+--------+--------------------+---------+--------+----------------------+
| Emp_no | Emp_name | Phone    | DOJ        | Salary | Designation        | Dept_ID | DeptID | DeptName             |
+--------+----------+----------+------------+--------+--------------------+---------+--------+----------------------+
|      1 | Bidipta  | 20131005 | 2009-06-05 |  35000 | Financial Officer  |       1 |      1 | Human Resources      |
|      2 | Arun     | 90651045 | 2009-07-08 |  38000 | Financial Officer  |       3 |      3 | Accounting & Finance |
|      3 | Shubham  | 90171007 | 2008-05-05 | 100000 | Operating Officer  |       2 |      2 | IT                   |
|      4 | Arun     | 88131067 | 2006-06-12 |  55000 | Technology Officer |       3 |      3 | Accounting & Finance |
|      5 | Sudipta  | 99131056 | 2011-11-05 |  65000 | Marketing Officer  |       4 |      4 | R & D                |
|      6 | Ramen    |  8631005 | 2010-04-09 |  43000 | Technology Officer |       4 |      4 | R & D                |
|      7 | Franklin | 98131056 | 2015-11-07 |  29000 | Legal Officer      |       2 |      2 | IT                   |
|      8 | RedHat   | 12331005 | 2008-06-09 |  90000 | Technology Officer |       1 |      1 | Human Resources      |
|      9 | COOLKID  | 12334005 | 2005-06-12 |  10000 | Legal Adviser      |       9 |   NULL | NULL                 |
+--------+----------+----------+------------+--------+--------------------+---------+--------+----------------------+
9 rows in set (0.0009 sec)
MySQL  localhost:33060+ ssl  organization585  SQL >
```

(vii)SELECT * FROM Employees
RIGHT OUTER JOIN dept
ON Employees.dept_ID = Dept.DeptID;



Emp_no	Emp_name	Phone	DOJ	Salary	Designation	Dept_ID	DeptID	DeptName
8	RedHat	12331005	2008-06-09	90000	Technology Officer	1	1	Human Resources
1	Bidipta	20131005	2009-06-05	35000	Financial Officer	1	1	Human Resources
7	Franklin	98131056	2015-11-07	29000	Legal Officer	2	2	IT
3	Shubham	90171007	2008-05-05	100000	Operating Officer	2	2	IT
4	Arun	88131067	2006-06-12	55000	Technology Officer	3	3	Accounting & Finance
2	Arun	90651045	2009-07-08	38000	Financial Officer	3	3	Accounting & Finance
6	Ramen	8631005	2010-04-09	43000	Technology Officer	4	4	R & D
5	Sudipta	99131056	2011-11-05	65000	Marketing Officer	4	4	R & D
NULL	NULL	NULL	NULL	NULL	NULL	NULL	5	Sales
NULL	NULL	NULL	NULL	NULL	NULL	NULL	6	Marketing
NULL	NULL	NULL	NULL	NULL	NULL	NULL	7	Banking

11 rows in set (0.0008 sec)
MySQL localhost:33060+ ssl organization585 SQL >

RESULT:

The nested queries and joins are executed successfully.

Programs

1. Consider the database for a banking enterprise. Write the queries for the below questions.

(i) List the deposit account number and amount in which the deposit scheme having maximum deposit is opened

(ii) List the account number and amount of that savings bank deposit scheme in which minimum amount is deposited.

(iii) List the customers having accounts in ‘Chennai’ branch

(iv) List the customers having more than one account

(v) List the customers having same name but different account numbers.

(vi) List the customer name that is having maximum deposit account in bank

(vii) List the customer who has borrowed highest amount of home loan

(viii) Display the customer details by implementing left inner join

(ix)Display the customer details by implementing a right outer join

Ans:

(i) SELECT Dacno, damt FROM deposit WHERE damt = (SELECT MAX(damt) FROM deposit);

```
mysql> SELECT Dacno, damt FROM deposit WHERE damt = (SELECT MAX(damt) FROM deposit);
+-------+----------+
| Dacno | damt     |
+-------+----------+
|   203 | 28000.00 |
+-------+----------+
1 row in set (0.01 sec)
```

(ii) SELECT Dacno, damt FROM deposit WHERE
damt = (SELECT MIN(damt) FROM deposit WHERE dtype = "Savings");

```
mysql> SELECT Dacno, damt FROM deposit WHERE
    -> damt = (SELECT MIN(damt) FROM deposit WHERE dtype = "Savings");
+-------+----------+
| Dacno | damt     |
+-------+----------+
|   205 | 12000.00 |
+-------+----------+
1 row in set (0.00 sec)
```

(iii) SELECT c_id, c_name, ph_no FROM customer
INNER JOIN Accounts_in ON customer.c_id = Accounts_in.cid
INNER JOIN Bank_brn ON Accounts_in.Bcode = Bank_brn.bcode
WHERE Bank_brn.bloc = "Chennai";

```
mysql> SELECT c_id, c_name, ph_no FROM customer
    -> INNER JOIN Accounts_in ON customer.c_id = Accounts_in.cid
    -> INNER JOIN Bank_brn ON Accounts_in.Bcode = Bank_brn.bcode
    -> WHERE Bank_brn.bloc = "Chennai";
+------+--------+--------+
| c_id | c_name | ph_no  |
+------+--------+--------+
|    2 | Tom    | 944031 |
+------+--------+--------+
1 row in set (0.00 sec)
```

(iv) SELECT c_id, c_name, ph_no FROM customer
INNER JOIN Accounts_in ON Customer.c_id = accounts_in.cid
GROUP BY Accounts_in.cid HAVING COUNT(accounts_in.cid)>1;

```
mysql> SELECT  c_id, c_name, ph_no FROM customer
    -> INNER JOIN Accounts_in ON Customer.c_id = accounts_in.cid
    -> GROUP BY Accounts_in.cid HAVING COUNT(accounts_in.cid)>1;
+------+--------+--------+
| c_id | c_name | ph_no  |
+------+--------+--------+
|    3 | Bhupen | 943351 |
+------+--------+--------+
1 row in set (0.00 sec)
```

(v) SELECT * from customer where c_name in (select c_name from customer group by c_name having count(*) > 1);

(vi) SELECT c_name FROM customer JOIN depositor
ON customer.c_id = depositor.cid JOIN deposit ON depositor.dacno = deposit.Dacno
WHERE deposit.damt = (SELECT MAX(damt) FROM deposit);

```
mysql> SELECT c_name FROM customer JOIN depositor
    -> ON customer.c_id = depositor.cid JOIN deposit ON depositor.dacno = deposit.Dacno
    -> WHERE deposit.damt = (SELECT MAX(damt) FROM deposit);
+--------+
| c_name |
+--------+
| Bhupen |
+--------+
1 row in set (0.00 sec)
```

(vii) SELECT c_name FROM customer JOIN borrower
ON customer.c_id=borrower.cid JOIN loan
ON loan.Lacno=borrower.lacno WHERE loan.lamt=(SELECT max(lamt)
FROM loan GROUP BY ltype HAVING ltype="Home");

```
mysql> SELECT c_name FROM customer JOIN borrower
    -> ON customer.c_id=borrower.cid JOIN loan
    -> ON loan.Lacno=borrower.lacno WHERE loan.lamt=(SELECT max(lamt)
    -> FROMloan GROUP BY ltype HAVING ltype="Home");
+---------+
| c_name  |
+---------+
| Priety  |
+---------+
1 row in set (0.01 sec)
```

(viii) SELECT * FROM customer LEFT JOIN accounts_in ON
accounts_in.cid=customer.c_id ORDER BY customer.c_id;

```
mysql> SELECT * FROM customer LEFT JOIN accounts_in ON
    -> accounts_in.cid=customer.c_id ORDER BY customer.c_id;
+------+--------+--------+------------+--------+----------+-------+------+
| c_id | c_name | ph_no  | dob        | gender | C_city   | Bcode | cid  |
+------+--------+--------+------------+--------+----------+-------+------+
|    1 | Rajiv  | 941032 | 1980-03-12 | Male   | Pune     | 101   |    1 |
|    2 | Tom    | 944031 | 1982-07-12 | Male   | Mumbai   | 102   |    2 |
|    3 | Bhupen | 943351 | 1983-06-15 | Male   | Guwahati | 106   |    3 |
|    3 | Bhupen | 943351 | 1983-06-15 | Male   | Guwahati | 103   |    3 |
|    4 | Priety | 942225 | 1990-04-15 | Female | Mumbai   | 104   |    4 |
|    5 | Jane   | 882342 | 1987-08-19 | Female | Chennai  | 105   |    5 |
|    6 | Harry  | 951037 | 1984-05-09 | Male   | Jorhat   | NULL  | NULL |
|    7 | Rajiv  | 953122 | 1989-03-19 | Male   | Kolkata  | NULL  | NULL |
+------+--------+--------+------------+--------+----------+-------+------+
8 rows in set (0.00 sec)
```

(ix) SELECT * FROM customer RIGHT JOIN accounts_in ON
accounts_in.cid=customer.c_id ORDER BY customer.c_id;

```
sql> SELECT * FROM customer RIGHT JOIN accounts_in ON
  -> accounts_in.cid=customer.c_id ORDER BY customer.c_id;
-----+--------+--------+------------+--------+----------+-------+-----+
c_id | c_name | ph_no  | dob        | gender | C_city   | Bcode | cid |
-----+--------+--------+------------+--------+----------+-------+-----+
   1 | Rajiv  | 941032 | 1980-03-12 | Male   | Pune     | 101   |   1 |
   2 | Tom    | 944031 | 1982-07-12 | Male   | Mumbai   | 102   |   2 |
   3 | Bhupen | 943351 | 1983-06-15 | Male   | Guwahati | 103   |   3 |
   3 | Bhupen | 943351 | 1983-06-15 | Male   | Guwahati | 106   |   3 |
   4 | Priety | 942225 | 1990-04-15 | Female | Mumbai   | 104   |   4 |
   5 | Jane   | 882342 | 1987-08-19 | Female | Chennai  | 105   |   5 |
-----+--------+--------+------------+--------+----------+-------+-----+
rows in set (0.00 sec)
```

2. Consider the database for a college. Write the queries for the below questions.

(i) List out the ID, Name and Date of Birth of students registered for a specific course.
(ii) List out the ID, Name and Date of Birth of students registered for a specific course, staying in a specific Hostel.
(iii) List the names of faculties who teach for a specific course.
(iv)Display the student details by implementing left inner join
(v) Display the student details by implementing a right outer join

Ans:

(i) select student.sid, student.sname, dob from student join
register on register.s_id=student.sid join courses
on courses.C_id=register.cid where courses.C_name="B.Sc";

```
+-----+----------------+------------+
| sid | sname          | dob        |
+-----+----------------+------------+
|   4 | Shahin Rahman  | 1994-11-19 |
|   7 | Rani Bharadwaj | 1996-10-05 |
+-----+----------------+------------+
2 rows in set (0.00 sec)
```

(ii) SELECT student.sid,student.sname,dob FROM student JOIN register ON
register.s_id=student.sid JOIN courses ON courses.C_id=register.cid
JOIN Hostel_details ON Hostel_details.sid=register.s_id JOIN hostel ON
Hostel_details.h_id=hostel.h_no WHERE courses.C_Name="BCA" AND hostel.H_name="Hostel A";

```
+-----+----------+------------+
| sid | sname    | dob        |
+-----+----------+------------+
|   3 | Max Diaz | 1997-11-02 |
+-----+----------+------------+
1 row in set (0.00 sec)
```

(iii) SELECT F_Name FROM faculty JOIN teaching ON teaching.F_id=faculty.F_id
JOIN courses ON courses.C_Id=teaching.course_id WHERE C_name="MBA";

```
+----------------+
| F_Name         |
+----------------+
| Ishan Kashyap  |
| Maria Crerar   |
+----------------+
2 rows in set (0.00 sec)
```

(iv) SELECT student.*, department.dname, hostel_details.h_id, courses.* FROM student

LEFT JOIN department ON student.dno=department.dno LEFT JOIN hostel_details ON student.sid=hostel_details.sid LEFT JOIN register ON student.sid=register.s_id LEFT JOIN courses ON register.cid=courses.c_id;

sid	sname	phone	dob	dno	dname	h_id	C_Id	C_name
1	Akash Verma	8865409	1996-03-06	1	Information Technology	2	11	B.Tech
2	Karan Shergill	78432163	1996-06-02	2	ECE	1	13	LLB
3	Max Diaz	78432155	1997-11-02	3	Law	1	15	BCA
4	Shahin Rahman	78432543	1994-11-19	3	Law	3	12	B.Sc
5	Arjun Singh	78432423	1996-08-11	4	Mechanical	2	14	MBA
6	Mark Cuban	78432177	1996-03-15	5	Literature	2	11	B.Tech
7	Rani Bharadwaj	7843222	1996-10-05	3	Law	4	12	B.Sc
8	Mark D'silva	9985321	1995-10-10	1	Information Technology	NULL	11	B.Tech

(v) SELECT student.*, department.dname, hostel_details.h_id, courses.*
FROM student RIGHT JOIN department ON student.dno=department.dno
RIGHT JOIN hostel_details ON student.sid=hostel_details.sid
RIGHT JOIN register ON student.sid=register.s_id
RIGHT JOIN courses ON register.cid=courses.c_id;

sid	sname	phone	dob	dno	dname	h_id	C_Id	C_name
1	Akash Verma	8865409	1996-03-06	1	Information Technology	2	11	B.Tech
6	Mark Cuban	78432177	1996-03-15	5	Literature	2	11	B.Tech
NULL	NULL	NULL	NULL	NULL	NULL	NULL	11	B.Tech
4	Shahin Rahman	78432543	1994-11-19	3	Law	3	12	B.Sc
7	Rani Bharadwaj	7843222	1996-10-05	3	Law	4	12	B.Sc
2	Karan Shergill	78432163	1996-06-02	2	ECE	1	13	LLB
5	Arjun Singh	78432423	1996-08-11	4	Mechanical	2	14	MBA
3	Max Diaz	78432155	1997-11-02	3	Law	1	15	BCA
NULL	NULL	NULL	NULL	NULL	NULL	NULL	16	B.Pharm
NULL	NULL	NULL	NULL	NULL	NULL	NULL	17	M.Tech

EXPERIMENT NO 5

VIEWS

AIM: Write the queries for the following.

(i) Create a view emp from employee such that it contains only emp_no and emp_name and department.

(ii) Create a view dept from department with only dept_no and location.

(iii) Create a view that contains the details of employees who are managers only.

(iv) drop the views.

OBJECTIVES:

To understand views in Mysql

THEORY:

A view is the tailored presentation of data contained in one or more table and can also be said as restricted view to the data in the tables. A view is a “virtual table” or a “stored query” which takes the output of a query and treats it as a table. The table upon which a view is created is called as base table. A view is a logical table based on a table or another view. A view contains no data of its own but is like a window through which data from tables can be viewed or changed. The tables on which a view is based are called base tables. The view is stored as a SELECT statement in the data dictionary.

Advantages of a view:

a. Additional level of table security.

b. Hides data complexity.

c. Simplifies the usage by combining multiple tables into a single table

Creating and dropping view:

Syntax:

```
Create or replace view  view_name   AS   SELECT
      column_name(s) FROM table_name WHERE condition;


Drop view <view name>;
```

PROCEDURE:

1) Create the employee table

2) Create the view

3) display the content of view

Result:

Thus the views are created successfully.

Problems:

1. Create and drop views on student table.

Ans:

(i) *View that contains only Student ID, Student Name & Date of Birth:*

CREATE VIEW view_1 AS SELECT sid, sname, dob FROM student;

```
MySQL  localhost:33060+ ssl  collegedb  SQL > select * from view_1;
+-----+-------------+------------+
| sid | sname       | dob        |
+-----+-------------+------------+
|   1 | Bidipta     | 1994-07-02 |
|   2 | Saikia      | 1993-08-04 |
|   3 | Manikangkan | 1995-11-04 |
|   4 | Bhupen      | 1997-02-12 |
|   5 | Pulak       | 1996-03-11 |
|   6 | Chris       | 1995-09-15 |
|   7 | Jaaz        | 1999-06-19 |
|   8 | Franklin    | 1995-01-15 |
|   9 | Marcus      | 1998-05-18 |
+-----+-------------+------------+
```

ii) *View that contains only student name, address and Mobile no.:*

CREATE VIEW view_2 AS SELECT sname, address, mobile_no FROM student;

```
MySQL  localhost:33060+ ssl  collegedb  SQL > select * from view_2;
+-------------+-------------+-----------+
| sname       | address     | mobile_no |
+-------------+-------------+-----------+
| Bidipta     | Golaghat    |   8767893 |
| Saikia      | Adabari     |  76236011 |
| Manikangkan | Hill Valley |   8976542 |
| Bhupen      | Nalbari     |   7234561 |
| Pulak       | Rangia      |   9231456 |
| Chris       | Jalukbari   |   7645321 |
| Jaaz        | Golaghat    |   7543212 |
| Franklin    | Nogaon      |  86543298 |
| Marcus      | Bokaghat    |  78432765 |
+-------------+-------------+-----------+
```

(iii) Dropping the views:

DROP VIEW view_1;

DROP VIEW view_2;

2. Create and drop views on customer and deposit table.

Ans:

(i) View that contains only Customer ID & Customer Name, from Customer table:

CREATE VIEW view_1 AS SELECT C_id, C_name FROM customer;

```
mysql> SELECT * FROM view_1;
+------+---------------+
| C_id | C_name        |
+------+---------------+
|    1 | Rakesh Sinha  |
|    2 | Nisha Bora    |
|    3 | John Marshal  |
|    4 | Neha Devi     |
|    5 | Swati Goyal   |
|    6 | Rishav Das    |
|    7 | Shiv Anand    |
|    8 | David Samuel  |
|    9 | Naina Sharma  |
+------+---------------+
```

(ii) View that contains Name & Date of Birth of those customers who are from Mumbai :
CREATE VIEW view_2 AS SELECT C_name, dob FROM customer WHERE c_city = "Mumbai";

```
mysql> SELECT * FROM view_2;
+------------+------------+
| C_name     | dob        |
+------------+------------+
| Nisha Bora | 1989-05-11 |
| Neha Devi  | 1993-04-18 |
| Shiv Anand | 1990-04-20 |
+------------+------------+
3 rows in set (0.00 sec)
```

(iii) View that Conatins only Account No & Deposit Type from Deposit Table:
CREATE VIEW dep1 AS SELECT Dacno, dtype FROM Deposit;

```
mysql> SELECT * FROM dep1;
+--------+--------------------+
| Dacno  | dtype              |
+--------+--------------------+
| 346721 | Current Deposit    |
| 432872 | Recurring Deposit  |
| 512342 | Fixed Deposit      |
| 612259 | Tax Saving Deposit |
+--------+--------------------+
4 rows in set (0.01 sec)
```

(iv) View that Conatins only Account No & Deposit Amount of those customers who have deposit amount greater than 20000, from Deposit Table:

CREATE VIEW dep2 AS SELECT Dacno, damt FROM Deposit WHERE damt >20000;

```
mysql> SELECT * FROM dep2;
+--------+----------+
| Dacno  | damt     |
+--------+----------+
| 346721 | 28000.00 |
| 612259 | 25000.00 |
+--------+----------+
2 rows in set (0.00 sec)
```

(v) DROP VIEW view_1;

DROP VIEW view_1;

DROP VIEW dep1;

DROP VIEW dep2;

EXPERIMENT NO 6

PROCEDURE

AIM: Write a procedure which increases the salary of an employee. It accepts an employee number and salary increase amount. It uses the employee number to find the current salary from the EMPLOYEE table and update the salary.

OBJECTIVES: To understand procedure in Mysql

THEORY

PROCEDURE:

In MySQL, a procedure is a stored program that you can pass parameters into. It does not return a value like a function does.

Syntax

Create procedure procedure name (parameter data type, parameter data type...)
Begin
Declaration section
Executable _section
End;

Procedure name

The name to assign to this procedure in MySQL.

Parameter

When creating a procedure, there are three types of parameters that can be declared:

1. IN - The parameter can be referenced by the procedure. The value of the parameter cannot be overwritten by the procedure.
2. OUT - The parameter cannot be referenced by the procedure, but the value of the parameter can be overwritten by the procedure.

3. IN OUT - The parameter can be referenced by the procedure and the value of the parameter can be overwritten by the procedure.

Declaration section

The place in the procedure where you declare local variables.

Executable section

The place in the procedure where you enter the code for the procedure.

PROCEDURE:

i.)Write the procedure with empno and increment name as parameter.
ii.)Use update command to increment salary.

CODE:

```
USE company;
DELIMITER //
CREATE procedure update_salary(IN emp_id int, IN increment int)
BEGIN
	UPDATE employee SET salary = salary + increment WHERE emp_no = emp_id;
END //
DELIMITER ;
CALL update_salary(3, 2000);
```

OUTPUT:

Salary of emp_no 3 is changed to 12000 from 10000;

```
mysql> SELECT emp_no, emp_name, salary FROM employee;
+--------+----------+----------+
| emp_no | emp_name | salary   |
+--------+----------+----------+
|      1 | Rahul    | 35000.00 |
|      2 | Priya    | 32000.00 |
|      3 | Sam      | 12000.00 |
|      4 | Jenny    | 50000.00 |
|      5 | Tom      | 38000.00 |
+--------+----------+----------+
5 rows in set (0.00 sec)
```

PROBLEMS:

1. Write a procedure which accept the account number of a customer and retrieve the balance.

Ans:

```
CREATE DATABASE bankdb;
USE bankdb;
CREATE TABLE balance_details (Acc_no INT PRIMARY KEY, Balance DECIMAL(9,2) NOT NULL);
INSERT INTO balance_details VALUES( 11, 30000), (12, 35000), (13, 40000), (15, 25000), (16, 44000);
DELIMITER //
CREATE PROCEDURE get_balance (IN ac_no INT)
BEGIN
	SELECT * FROM balance_details WHERE acc_no = ac_no;
END //
DELIMITER ;
CALL get_balance(11);
```

Output:

```
MySQL  localhost:33060+ ssl  bankdb  SQL > CALL get_balance(11);
+--------+----------+
| Acc_no | Balance  |
+--------+----------+
|     11 | 30000.00 |
+--------+----------+
1 row in set (0.0020 sec)
```

2. Write a procedure which accepts the student number and displays the department in which he belongs to.

Ans:

```
use college;
DELIMITER //
CREATE PROCEDURE get_dept (IN s_no INT)
BEGIN
SELECT student.dno, department.dname FROM department INNER JOIN student ON student.dno =
department.dno WHERE sid = s_no;
END //
DELIMITER ;

CALL get_dept (1);
```

OUTPUT:

```
MySQL  localhost:33060+ ssl  college  SQL > call get_dept (1);
+-----+-------+
| dno | dname |
+-----+-------+
|   2 | ECE   |
+-----+-------+
```