ADVANCED DATABASE MANAGEMENT SYSTEMS (IT-408)

PRACTICAL FILE

University School of Information, Communication and Technology

Guru Gobind Singh Indraprastha University, Delhi

Submitted by: YASH ARYAN (06816403220)

Batch: B.Tech (CSE) 8th Semester

Submitted to: Dr. Priyanka Bhutani

Index

2	Retrieve the name and address of all employees who workfor the ‘Research’ department.	
3	For every project located in ‘Delhi’, list the project number,the controlling department number, and the department manager’s last name, address, and birthdate.	
4	Find the maximum salary, the minimum salary, and theaverage salary among employees who work for the ‘Research’ department.	
5	For each department retrieve the department number, thenumber of employees in the department, and their averagesalary.	
6	For each project on which more than two employees work,retrieve the project number, project name, and the number ofemployees who work on that project.	
7	Retrieve a list of employees and the projects each works in,ordered by the employee’s department and within eachdepartment ordered alphabetically by employee last name.	
8	Give all employee in the ‘Research’ department a 10% raisein salary.	
9	Add a column JOBTITLE to the EMPLOYEE table (assume data). Delete only those who are working as lecturer.	
10	Display only those employees whose dept_no is 30.	
11	Display dept_no from the table employee avoiding the duplicated values.	
12	Display the rows whose salary ranges from 15000 to 30000.	
13	Count the total records in the employee table.	
14	Find how many job titles are available in employee table.	
15	What is the difference between maximum and minimum salaries of employees in the organization?	
16	Issue a query to find all the employees who work in the same job as Arjun.	
17	Display all the dept numbers available with the dept and employee tables avoiding duplicates	
18	Display all the dept numbers available with the dept and employee tables.	
19	Display all the dept numbers available in employee and not in dept tables and vice versa.	
20	The organization wants to display only the details of the employees those who are ASP.	

PART A

Lab 1

Aim: Create a table STUDENT with the following fields and proper constraints:

FIELD	CONSTRAINTS
Enrollment No.	Not NULL, Primary Key
Name	Not NULL
Program	Not NULL
Address	No constraints
Phone Number	No constraints
Date of Birth	No constraints

Query:

```
mysql> CREATE DATABASE DATABASE_1;
Query OK, 1 row affected (0.01 sec)

mysql> USE DATABASE_1;
Database changed
mysql> CREATE TABLE STUDENT (
    -> Enrollment_No INT NOT NULL,
    -> Name VARCHAR(100) NOT NULL,
    -> Program VARCHAR(50) NOT NULL,
    -> Address VARCHAR(255),
    -> Phone_Number VARCHAR(15),
    -> Date_of_Birth DATE,
    -> PRIMARY KEY (Enrollment_No)
    -> );
Query OK, 0 rows affected (0.06 sec)
```

Lab 2

Aim: Insert Data into table STUDENT. (assume data)

Query:

```
mysql> INSERT INTO STUDENT (Enrollment_No, Name, Program, Address, Phone_Number, Date_of_Birth)
    -> VALUES (1, 'John Doe', 'Computer Science', '123 Main St', '555-1234', '2000-01-01');
Query OK, 1 row affected (0.04 sec)

mysql>
mysql> INSERT INTO STUDENT (Enrollment_No, Name, Program, Address, Phone_Number, Date_of_Birth)
    -> VALUES (2, 'Jane Smith', 'Electrical Engineering', '456 Elm St', '555-5678', '1999-02-02');
Query OK, 1 row affected (0.00 sec)

mysql>
mysql> INSERT INTO STUDENT (Enrollment_No, Name, Program, Address, Phone_Number, Date_of_Birth)
    -> VALUES (3, 'Alice Johnson', 'Mechanical Engineering', '789 Oak St', '555-9012', '2001-03-03');
Query OK, 1 row affected (0.00 sec)
```

Lab 3

Aim: Insert more than one record into STUDENT table using a single insert command.

Query:

```
mysql> INSERT INTO STUDENT (Enrollment_No, Name, Program, Address, Phone_Number, Date_of_Birth)
    -> VALUES
    -> (4, 'Emily Brown', 'Civil Engineering', '321 Maple St', '555-2345', '2001-04-04'),
    -> (5, 'Michael Green', 'Biotechnology', '654 Pine St', '555-6789', '2002-05-05'),
    -> (6, 'Sarah Davis', 'Chemical Engineering', '987 Cedar St', '555-3456', '1998-06-06');
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

Lab 4

Aim: Create a table MARKS with the following fields and constraints :

FIELDS	CONSTRAINTS
Enrollment Number	Not NULL, foreign Key
Semester	Not NULL
Marks 1	b/w 0 to 100
Marks 2	b/w 0 to 100
Marks 3	b/w 0 to 100
Marks 4	b/w 0 to 100
Marks 5	b/w 0 to 100
Marks 6	b/w 0 to 100

Create composite Primary key with Enrollment Number and Semester.

Query:

```
mysql> CREATE TABLE MARKS (
    ->     Enrollment_Number INT NOT NULL,
    ->     Semester INT NOT NULL,
    ->     Marks_1 INT CHECK (Marks_1 BETWEEN 0 AND 100),
    ->     Marks_2 INT CHECK (Marks_2 BETWEEN 0 AND 100),
    ->     Marks_3 INT CHECK (Marks_3 BETWEEN 0 AND 100),
    ->     Marks_4 INT CHECK (Marks_4 BETWEEN 0 AND 100),
    ->     Marks_5 INT CHECK (Marks_5 BETWEEN 0 AND 100),
    ->     Marks_6 INT CHECK (Marks_6 BETWEEN 0 AND 100),
    ->     PRIMARY KEY (Enrollment_Number, Semester),
    ->     FOREIGN KEY (Enrollment_Number) REFERENCES STUDENT(Enrollment_No)
    -> );
Query OK, 0 rows affected (0.04 sec)
```

Lab 5

Aim: Insert data into MARKS table. (assume data)

Query:

```
mysql> INSERT INTO MARKS (Enrollment_Number, Semester, Marks_1, Marks_2, Marks_3, Marks_4, Marks_5, Marks_6)
    -> VALUES
    -> (1, 1, 85, 78, 90, 88, 76, 84),
    -> (2, 1, 82, 74, 85, 80, 79, 81),
    -> (3, 1, 88, 82, 87, 90, 85, 89);
Query OK, 3 rows affected (0.02 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

Lab 6

Aim: Add a column percentage to the MARKS table.

Query:

```
mysql> ALTER TABLE MARKS
    -> ADD Percentage DECIMAL(5,2); -- Assuming percentage can have up to 2 decimal places
Query OK, 0 rows affected (0.07 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

Lab 7

Aim: Using Enrollment number of a student, write a query to change the address and phone number of the student.

Query:

```
mysql> UPDATE STUDENT
    -> SET Address = '123 New Street', Phone_Number = '555-4321'
    -> WHERE Enrollment_No = 1;
Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

Lab 8

Aim: Create an Index on table STUDENT with the name and enrollment number as combination fields.

Query:

```
mysql> CREATE INDEX idx_student_name_enrollment_no ON STUDENT (Name, Enrollment_No);
Query OK, 0 rows affected (0.10 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

Lab 9

Aim: Write a query to :

a. Display names of all students as well as their DOB and Enrollment number in alphabetical order.

Query:

```
mysql> SELECT Name, Date_of_Birth, Enrollment_No
    -> FROM STUDENT
    -> ORDER BY Name ASC;
+---------------+---------------+---------------+
| Name          | Date_of_Birth | Enrollment_No |
+---------------+---------------+---------------+
| Alice Johnson | 2001-03-03    |             3 |
| Emily Brown   | 2001-04-04    |             4 |
| Jane Smith    | 1999-02-02    |             2 |
| John Doe      | 2000-01-01    |             1 |
| Michael Green | 2002-05-05    |             5 |
| Sarah Davis   | 1998-06-06    |             6 |
+---------------+---------------+---------------+
6 rows in set (0.01 sec)
```

b. Display names of all Students whose Name starts with ‘P’.

Query:

```
mysql> SELECT Name
    -> FROM STUDENT
    -> WHERE Name LIKE 'P%';
Empty set (0.01 sec)
```

c. Display names of all students whose name has the letter ‘A’ anywhere in the name.

Query:

```
mysql> SELECT Name
    -> FROM STUDENT
    -> WHERE Name LIKE '%A%';
+---------------+
| Name          |
+---------------+
| Alice Johnson |
| Jane Smith    |
| Michael Green |
| Sarah Davis   |
+---------------+
4 rows in set (0.00 sec)
```

d. Display name of the student, DOB and age of the student as of today.

Query:

```
mysql> SELECT
    ->     Name,
    ->     Date_of_Birth,
    ->     TIMESTAMPDIFF(YEAR, Date_of_Birth, CURDATE()) AS Age
    -> FROM
    ->     STUDENT;
+---------------+---------------+------+
| Name          | Date_of_Birth | Age  |
+---------------+---------------+------+
| John Doe      | 2000-01-01    |   24 |
| Jane Smith    | 1999-02-02    |   25 |
| Alice Johnson | 2001-03-03    |   23 |
| Emily Brown   | 2001-04-04    |   23 |
| Michael Green | 2002-05-05    |   22 |
| Sarah Davis   | 1998-06-06    |   25 |
+---------------+---------------+------+
6 rows in set (0.00 sec)
```

e. Display the names of all the students born in September.

```
mysql> SELECT Name
    -> FROM STUDENT
    -> WHERE MONTH(Date_of_Birth) = 9;
Empty set (0.00 sec)
```

f. Display only the names and year of birth of all the students.

```
mysql> SELECT Name, YEAR(Date_of_Birth) AS Year_of_Birth
    -> FROM STUDENT;
+---------------+---------------+
| Name          | Year_of_Birth |
+---------------+---------------+
| John Doe      |          2000 |
| Jane Smith    |          1999 |
| Alice Johnson |          2001 |
| Emily Brown   |          2001 |
| Michael Green |          2002 |
| Sarah Davis   |          1998 |
+---------------+---------------+
6 rows in set (0.00 sec)
```

Lab 10

Aim: Write a query to create a VIEW with both tables STUDENT and MARKS which contains the Name, Enrollment Number, semester along with percentage of marks.

Query:

```
mysql> CREATE VIEW Student_Marks_View AS
    -> SELECT s.Name, m.Enrollment_Number, m.Semester,
    ->     ((m.Marks_1 + m.Marks_2 + m.Marks_3 + m.Marks_4 + m.Marks_5 + m.Marks_6) / 6) AS Percentage
    -> FROM STUDENT s
    -> JOIN MARKS m ON s.Enrollment_No = m.Enrollment_Number;
Query OK, 0 rows affected (0.03 sec)
```

Lab 11

Aim: Write a query using the above VIEW to

a. Display name, enrollment number and percentage of marks for all students who have scored more than 70% marks.

Query:

```
mysql> SELECT Name, Enrollment_Number, Percentage
    -> FROM Student_Marks_View
    -> WHERE Percentage > 70;
+---------------+-------------------+------------+
| Name          | Enrollment_Number | Percentage |
+---------------+-------------------+------------+
| John Doe      |                 1 |    83.5000 |
| Jane Smith    |                 2 |    80.1667 |
| Alice Johnson |                 3 |    86.8333 |
+---------------+-------------------+------------+
3 rows in set (0.01 sec)
```

b. Display the name and enrollment number of the student who has received the highest marks in the examination.

Query:

```
mysql> SELECT Name, Enrollment_Number
    -> FROM Student_Marks_View
    -> WHERE Percentage = (
    ->     SELECT MAX(Percentage)
    ->     FROM Student_Marks_View
    -> );
+---------------+-------------------+
| Name          | Enrollment_Number |
+---------------+-------------------+
| Alice Johnson |                 3 |
+---------------+-------------------+
1 row in set (0.01 sec)
```

c. Replace the above view by altering its definition and adding DOB of student as well.

Query:

```
mysql> ALTER VIEW Student_Marks_View AS
    -> SELECT s.Name, s.Date_of_Birth, m.Enrollment_Number, m.Semester,
    ->     ((m.Marks_1 + m.Marks_2 + m.Marks_3 + m.Marks_4 + m.Marks_5 + m.Marks_6) / 6) AS Percentage
    -> FROM STUDENT s
    -> JOIN MARKS m ON s.Enrollment_No = m.Enrollment_Number;
Query OK, 0 rows affected (0.02 sec)
```

Lab 12

Aim: Create a table CLASS with same structure as STUDENT and import all the data from STUDENT into this table and then:

a. Modify the size of the address column.

Query:

```
mysql> CREATE TABLE CLASS LIKE STUDENT;
Query OK, 0 rows affected (0.09 sec)

mysql> INSERT INTO CLASS SELECT * FROM STUDENT;
Query OK, 6 rows affected (0.02 sec)
Records: 6  Duplicates: 0  Warnings: 0

mysql> ALTER TABLE CLASS MODIFY Address VARCHAR(255);
Query OK, 0 rows affected (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

b. Truncate the CLASS table.

Query:

```
mysql> TRUNCATE TABLE CLASS;
Query OK, 0 rows affected (0.14 sec)
```

c. Drop the table CLASS.

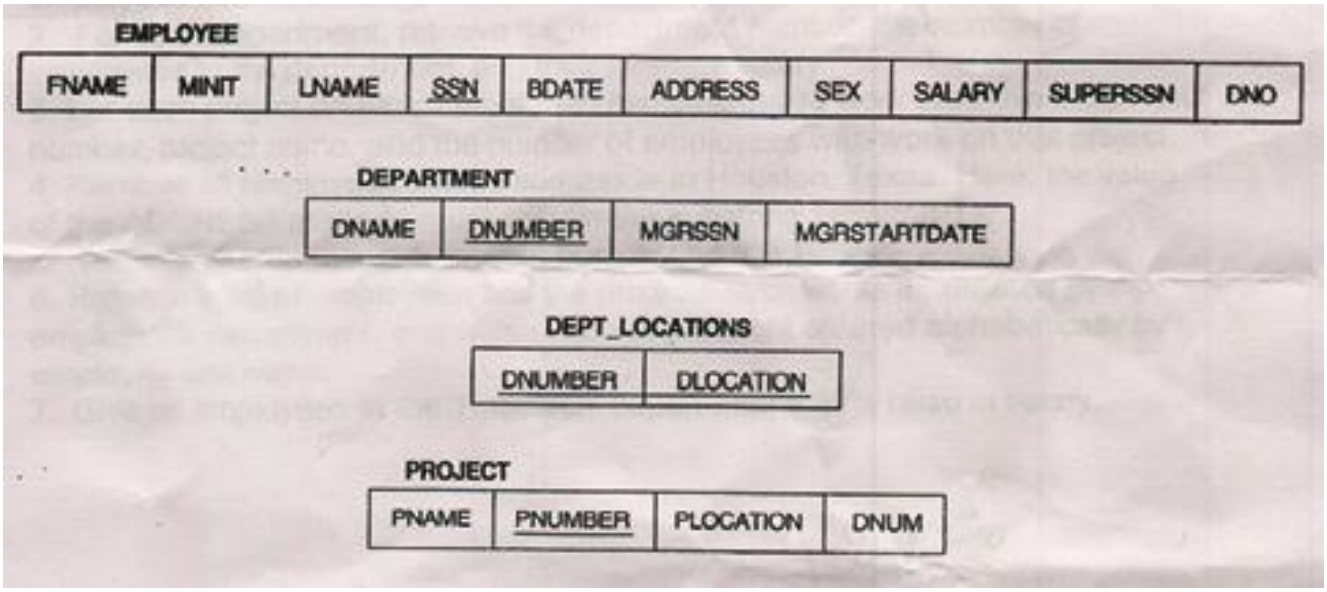
Query:

```
mysql> DROP TABLE CLASS;
Query OK, 0 rows affected (0.04 sec)
```

PART B

Lab 1

Aim: Create a database with the tables listed below. Choose appropriate data types for each attribute. Add appropriate tuples to the table. Retrieve the birthdate and address of the employee whose name is 'John B.Smith'.



Query:

```
mysql> CREATE DATABASE DATABASE_2;
Query OK, 1 row affected (0.00 sec)
```

```
mysql> USE DATABASE_2;
Database changed
```

```
mysql> -- Create EMPLOYEE table
mysql> CREATE TABLE EMPLOYEE (
    ->      FNAME VARCHAR(50),
    ->      MINIT CHAR(1),
    ->      LNAME VARCHAR(50),
    ->      SSN CHAR(9) PRIMARY KEY,
    ->      BDATE DATE,
    ->      ADDRESS VARCHAR(100),
    ->      SEX CHAR(1),
    ->      SALARY DECIMAL(10, 2),
    ->      SUPERSSN CHAR(9),
    ->      DNO INT
    -> );
Query OK, 0 rows affected (0.03 sec)
```

```
mysql> -- Create DEPT table
mysql> CREATE TABLE DEPT (
    ->      DNAME VARCHAR(50),
    ->      DNUMBER INT PRIMARY KEY,
    ->      MGRSSN CHAR(9),
    ->      MORSTARTDATE DATE
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql>
```

```
mysql> -- Create DEPT_LOC table
mysql> CREATE TABLE DEPT_LOC (
    ->     DNUMBER INT PRIMARY KEY,
    ->     DLOCATION VARCHAR(100)
    -> );
Query OK, 0 rows affected (0.03 sec)
```

```
mysql> -- Create PROJECT table
mysql> CREATE TABLE PROJECT (
    ->     PNAME VARCHAR(100),
    ->     PNUMBER INT PRIMARY KEY,
    ->     PLOCATION VARCHAR(100),
    ->     DNUM INT
    -> );
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> -- Add tuples to EMPLOYEE table
mysql> INSERT INTO EMPLOYEE (FNAME, MINIT, LNAME, SSN, BDATE, ADDRESS, SEX, SALARY, SUPERSSN, DNO) VALUES
    -> ('John', 'B', 'Smith', '123456789', '1980-05-15', '123 Main St, Anytown, USA', 'M', 50000.00, '987654321', 5),
    -> ('Alice', 'M', 'Johnson', '234567890', '1975-10-20', '456 Oak St, Anycity, USA', 'F', 60000.00, '888665555', 4),
    -> ('Robert', 'K', 'Williams', '345678901', '1982-12-30', '789 Elm St, Anysville, USA', 'M', 55000.00, '333445555', 5),
    -> ('James', 'E', 'Jones', '456789012', '1979-07-25', '321 Pine St, Anytown, USA', 'M', 70000.00, NULL, 4);
Query OK, 4 rows affected (0.01 sec)
Records: 4  Duplicates: 0  Warnings: 0

mysql>
mysql> -- Add tuples to DEPT table
mysql> INSERT INTO DEPT (DNAME, DNUMBER, MGRSSN, MORSTARTDATE) VALUES
    -> ('Research', 5, '333445555', '2005-01-01'),
    -> ('Human Resources', 4, '888665555', '2003-05-10');
Query OK, 2 rows affected (0.01 sec)
Records: 2  Duplicates: 0  Warnings: 0

mysql>
mysql> -- Add tuples to DEPT_LOC table
mysql> INSERT INTO DEPT_LOC (DNUMBER, DLOCATION) VALUES
    -> (4, 'Building A'),
    -> (5, 'Building B');
Query OK, 2 rows affected (0.01 sec)
Records: 2  Duplicates: 0  Warnings: 0

mysql>
mysql> -- Add tuples to PROJECT table
mysql> INSERT INTO PROJECT (PNAME, PNUMBER, PLOCATION, DNUM) VALUES
    -> ('Research Project 1', 1, 'Anytown', 5),
    -> ('HR Project 1', 2, 'Anycity', 4);
Query OK, 2 rows affected (0.01 sec)
Records: 2  Duplicates: 0  Warnings: 0
```

```
mysql> SELECT BDATE, ADDRESS
    -> FROM EMPLOYEE
    -> WHERE FNAME = 'John' AND LNAME = 'Smith' AND MINIT = 'B';
+------------+----------------------------+
| BDATE      | ADDRESS                    |
+------------+----------------------------+
| 1980-05-15 | 123 Main St, Anytown, USA |
+------------+----------------------------+
1 row in set (0.00 sec)
```

Lab 2

Aim: Retrieve the name and address of all employees who workfor the ‘Research’ department.

Query:

```
mysql> SELECT EMPLOYEE.FNAME, EMPLOYEE.LNAME, EMPLOYEE.ADDRESS
    -> FROM EMPLOYEE
    -> JOIN DEPT ON EMPLOYEE.DNO = DEPT.DNUMBER
    -> WHERE DEPT.DNAME = 'Research';
+--------+----------+----------------------------+
| FNAME  | LNAME    | ADDRESS                    |
+--------+----------+----------------------------+
| John   | Smith    | 123 Main St, Anytown, USA  |
| Robert | Williams | 789 Elm St, Anysville, USA |
+--------+----------+----------------------------+
2 rows in set (0.00 sec)
```

Lab 3

Aim: For every project located in ‘Delhi’, list the project number,the controlling department number, and the department manager’s last name, address, and birthdate.

Query:

```
mysql> SELECT
    ->     PROJECT.PNUMBER AS ProjectNumber,
    ->     PROJECT.DNUM AS ControllingDeptNumber,
    ->     EMPLOYEE.LNAME AS ManagerLastName,
    ->     EMPLOYEE.ADDRESS AS ManagerAddress,
    ->     EMPLOYEE.BDATE AS ManagerBirthdate
    -> FROM
    ->     PROJECT
    -> JOIN
    ->     DEPT ON PROJECT.DNUM = DEPT.DNUMBER
    -> JOIN
    ->     EMPLOYEE ON DEPT.MGRSSN = EMPLOYEE.SSN
    -> WHERE
    ->     PROJECT.PLOCATION = 'Delhi';
Empty set (0.00 sec)
```

Lab 4

Aim: Find the maximum salary, the minimum salary, and the average salary among employees who work for the ‘Research’ department.

Query:

```
mysql> SELECT
    ->     MAX(EMPLOYEE.SALARY) AS MaxSalary,
    ->     MIN(EMPLOYEE.SALARY) AS MinSalary,
    ->     AVG(EMPLOYEE.SALARY) AS AvgSalary
    -> FROM
    ->     EMPLOYEE
    -> JOIN
    ->     DEPT ON EMPLOYEE.DNO = DEPT.DNUMBER
    -> WHERE
    ->     DEPT.DNAME = 'Research';
+-----------+-----------+--------------+
| MaxSalary | MinSalary | AvgSalary    |
+-----------+-----------+--------------+
|  55000.00 |  50000.00 | 52500.000000 |
+-----------+-----------+--------------+
1 row in set (0.00 sec)
```

Lab 5

Aim: For each department retrieve the department number, the number of employees in the department, and their average salary.

Query:

```
mysql> SELECT
    ->     EMPLOYEE.DNO AS DepartmentNumber,
    ->     COUNT(*) AS NumberOfEmployees,
    ->     AVG(EMPLOYEE.SALARY) AS AverageSalary
    -> FROM
    ->     EMPLOYEE
    -> GROUP BY
    ->     EMPLOYEE.DNO;
+------------------+--------------------+---------------+
| DepartmentNumber | NumberOfEmployees | AverageSalary |
+------------------+--------------------+---------------+
|                5 |                  2 |  52500.000000 |
|                4 |                  2 |  65000.000000 |
+------------------+--------------------+---------------+
2 rows in set (0.01 sec)
```

Lab 6

Aim: For each project on which more than two employees work, retrieve the project number, project name, and the number of employees who work on that project.

Query:

```
mysql> SELECT
    ->     PROJECT.PNUMBER AS ProjectNumber,
    ->     PROJECT.PNAME AS ProjectName,
    ->     COUNT(*) AS NumberOfEmployees
    -> FROM
    ->     PROJECT
    -> JOIN
    ->     EMPLOYEE ON PROJECT.PNUMBER = EMPLOYEE.DNO
    -> GROUP BY
    ->     PROJECT.PNUMBER, PROJECT.PNAME
    -> HAVING
    ->     COUNT(*) > 2;
Empty set (0.00 sec)
```

Lab 7

Aim: Retrieve a list of employees and the projects each works in,ordered by the employee's department and within eachdepartment ordered alphabetically by employee last name.

Query:

```
mysql> SELECT
    ->     EMPLOYEE.FNAME AS FirstName,
    ->     EMPLOYEE.LNAME AS LastName,
    ->     PROJECT.PNAME AS ProjectName,
    ->     DEPT.DNAME AS DepartmentName
    -> FROM
    ->     EMPLOYEE
    -> JOIN
    ->     PROJECT ON EMPLOYEE.DNO = PROJECT.DNUM
    -> JOIN
    ->     DEPT ON EMPLOYEE.DNO = DEPT.DNUMBER
    -> ORDER BY
    ->     DEPT.DNAME,
    ->     EMPLOYEE.LNAME;
+-----------+----------+--------------------+-----------------+
| FirstName | LastName | ProjectName        | DepartmentName  |
+-----------+----------+--------------------+-----------------+
| Alice     | Johnson  | HR Project 1       | Human Resources |
| James     | Jones    | HR Project 1       | Human Resources |
| John      | Smith    | Research Project 1 | Research        |
| Robert    | Williams | Research Project 1 | Research        |
+-----------+----------+--------------------+-----------------+
4 rows in set (0.00 sec)
```

Lab 8

Aim: Give all employee in the ‘Research’ department a 10% raisein salary.

Query:

```
mysql> UPDATE EMPLOYEE
    -> SET SALARY = SALARY * 1.1
    -> WHERE DNO IN (
    ->     SELECT DNUMBER
    ->     FROM DEPT
    ->     WHERE DNAME = 'Research'
    -> );
Query OK, 2 rows affected (0.01 sec)
Rows matched: 2  Changed: 2  Warnings: 0
```

Lab 9

Aim: Add a column JOBTITLE to the EMPLOYEE table (assume data). Delete only those who are working as lecturer.

Query:

```
mysql> ALTER TABLE EMPLOYEE
    -> ADD JOBTITLE VARCHAR(100);
Query OK, 0 rows affected (0.06 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> DELETE FROM EMPLOYEE
    -> WHERE JOBTITLE = 'lecturer';
Query OK, 0 rows affected (0.00 sec)

mysql>
```

Lab 10

Aim: Display only those employees whose dept_no is 30.

Query:

```
mysql> SELECT *
    -> FROM EMPLOYEE
    -> WHERE DNO = 30;
Empty set (0.00 sec)
```

Lab 11

Aim: Display dept_no from the table employee avoiding the duplicated values.

Query:

```
mysql> SELECT DISTINCT DNO
    -> FROM EMPLOYEE;
+-------+
| DNO   |
+-------+
|     3 |
|     5 |
|     2 |
|     4 |
|     1 |
+-------+
5 rows in set (0.00 sec)
```

Lab 12

Aim: Display the rows whose salary ranges from 15000 to 30000.

Query:

```
mysql> SELECT *
    -> FROM EMPLOYEE
    -> WHERE SALARY BETWEEN 15000 AND 30000;
Empty set (0.00 sec)
```

Lab 13

Aim: Count the total records in the employee table.

Query:

```
mysql> SELECT COUNT(*) AS TotalRecords
    -> FROM EMPLOYEE;
+--------------+
| TotalRecords |
+--------------+
|            8 |
+--------------+
1 row in set (0.01 sec)
```

Lab 14

Aim: Find how many job titles are available in employee table.

Query:

```
mysql> SELECT COUNT(DISTINCT DNAME) AS NumberOfDepartments
    -> FROM DEPT
    -> WHERE DNUMBER IN (SELECT DNO FROM EMPLOYEE);
+----------------------+
| NumberOfDepartments  |
+----------------------+
|                    2 |
+----------------------+
1 row in set (0.00 sec)
```

Lab 15

Aim: What is the difference between maximum and minimum salaries of employees in the organization?

Query:

```
mysql> SELECT MAX(SALARY) - MIN(SALARY) AS SalaryDifference
    -> FROM EMPLOYEE;
+------------------+
| SalaryDifference |
+------------------+
|         15000.00 |
+------------------+
1 row in set (0.00 sec)
```

Lab 16

Aim: Issue a query to find all the employees who work in the same job as Arjun.

Query:

```
mysql> SELECT *
    -> FROM EMPLOYEE
    -> WHERE DNO = (
    ->       SELECT DNO
    ->       FROM EMPLOYEE
    ->       WHERE FNAME = 'Arjun'
    -> );
Empty set (0.00 sec)
```

Lab 17

Aim: Display all the dept numbers available with the dept and employee tables avoiding duplicates

Query:

```
mysql> SELECT DNUMBER
    -> FROM DEPT
    -> UNION
    -> SELECT DNO AS DNUMBER
    -> FROM EMPLOYEE;
+----------+
| DNUMBER  |
+----------+
|        4 |
|        5 |
|        3 |
|        2 |
|        1 |
+----------+
5 rows in set (0.00 sec)
```

Lab 18

Aim: Display all the dept numbers available with the dept and employee tables.

Query:

```
mysql> SELECT DNUMBER FROM DEPT
    -> UNION
    -> SELECT DNO FROM EMPLOYEE;
+---------+
| DNUMBER |
+---------+
|       4 |
|       5 |
|       3 |
|       2 |
|       1 |
+---------+
5 rows in set (0.00 sec)
```

Lab 19

Aim: Display all the dept numbers available in employee and not in dept tables and vice versa.

Query:

```
mysql> -- Department numbers available in EMPLOYEE table but not in DEPT table
mysql> SELECT DISTINCT E.DNO AS DeptNumberInEmployee
    -> FROM EMPLOYEE E
    -> LEFT JOIN DEPT D ON E.DNO = D.DNUMBER
    -> WHERE D.DNUMBER IS NULL;
+----------------------+
| DeptNumberInEmployee |
+----------------------+
|                    3 |
|                    2 |
|                    1 |
+----------------------+
3 rows in set (0.00 sec)

mysql>
mysql> -- Department numbers available in DEPT table but not in EMPLOYEE table
mysql> SELECT DISTINCT D.DNUMBER AS DeptNumberInDept
    -> FROM DEPT D
    -> LEFT JOIN EMPLOYEE E ON D.DNUMBER = E.DNO
    -> WHERE E.DNO IS NULL;
Empty set (0.00 sec)
```

Lab 20

Aim: The organization wants to display only the details of the employees those who are ASP.

Query:

```
mysql> SELECT *
    -> FROM EMPLOYEE
    -> WHERE DNO IN (
    ->       SELECT DNUMBER
    ->       FROM DEPT
    ->       WHERE DNAME = 'ASP'
    -> );
Empty set (0.00 sec)
```