Assignment 2 & 3

Name: Diptangshu Dey

RollNo: 20CS8018

Question:

University database:

A student is described by a unique Roll Number, Name Adress, and Semester. Each student enrolls himself in an Academic programme offered by a Department. Academic programmes have programme name(unique), duration, a programme code(unique) and a list of courses (both core and elective course) while the departments have department code (unique), department name (unique), HoD who is a Teacher and list of courses offered by it. Each teacher is described by employee code (unique), name, department and designation. A student registers some courses in a semester. A course is described by a unique course number, title of the course, credit allotted for the course and offering department. Database stores the grades obtained by different student in different courses registered by him/her in different semesters. Database also stores information about the courses offered by a department in a semester, the corresponding teacher(s) for each course.

Assignment 2:

Make appropriate tables for the above database and try to find out the following queries:

1. Find all the students' name, city, course allotted from the CSE department.
2. List the total number of Faculty in the CSE department.
3. List the available courses from the CSE department.
4. List the all students in a particular semester.
5. List the students who earned CGPA greater than or equal to 8.5

Assignment 3:

Make appropriate tables for the above database and try to find out the following queries:

1. How much subjects are registered by a student in each semester.
2. List the common students who are allotted the same courses of both the programme MCA and M.Tech.
3. List the total number of student enrolled in the subject DBMS.
4. Retrieve the semester of the student under DBMS subject.
5. Retrieve all the student name and arrange into ascending order.

Solution

Create Tables and insert data

```
CREATE TABLE `Academic_Programmes` (
  `Name` tinytext DEFAULT NULL,
  `Programme_Code` varchar(10) NOT NULL,
  `Duration` int(2) DEFAULT NULL,
  PRIMARY KEY (`Programme_Code`),
  UNIQUE KEY `Name` (`Name`) USING HASH
)

CREATE TABLE `Course` (
  `CourseNo` varchar(3) NOT NULL,
  `Title` tinytext DEFAULT NULL,
  `Credits` int(2) DEFAULT NULL,
  PRIMARY KEY (`CourseNo`)
)

CREATE TABLE `Courses_Offerred` (
  `CourseNo` varchar(3) DEFAULT NULL,
  `Department` varchar(3) DEFAULT NULL,
  `Teacher` varchar(3) DEFAULT NULL,
  `Programme` varchar(10) DEFAULT NULL
)

CREATE TABLE `Department` (
  `DepartmentCode` varchar(3) NOT NULL,
  `Name` tinytext DEFAULT NULL,
  PRIMARY KEY (`DepartmentCode`)
)


CREATE TABLE `Enrollment` (
  `RollNo` varchar(4) DEFAULT NULL,
  `ProgrammeCode` varchar(10) DEFAULT NULL,
  `RegistrationNo` varchar(10) NOT NULL,
  PRIMARY KEY (`RegistrationNo`)
)

CREATE TABLE `Grade` (
  `RollNo` varchar(4) DEFAULT NULL,
  `CourseNo` varchar(3) DEFAULT NULL,
  `Grade` float(2,2) DEFAULT NULL
)

CREATE TABLE `Registration` (
  `RollNo` varchar(4) DEFAULT NULL,
  `CourseNo` varchar(3) DEFAULT NULL,
  `Semester` int(2) DEFAULT NULL
)

CREATE TABLE `Student` (
  `RollNo` varchar(4) NOT NULL,
  `Name` tinytext DEFAULT NULL,
  `Address` varchar(40) DEFAULT NULL,
  `Semester` int(2) DEFAULT NULL,
```

```
  PRIMARY KEY (`RollNo`)
)

CREATE TABLE `Teacher` (
  `EmployeeCode` varchar(3) NOT NULL,
  `Name` tinytext DEFAULT NULL,
  `Department` varchar(3) DEFAULT NULL,
  `Designation` tinytext DEFAULT NULL,
  PRIMARY KEY (`EmployeeCode`)
)

INSERT INTO Academic_Programmes (Name, Programme_Code, Duration)
VALUES ('Bachelor of Technology', 'BTech', 4),
       ('Master of Technology', 'MTech', 2),
       ('Master of Computer Applications', 'MCA', 3);

INSERT INTO Course (CourseNo, Title, Credits)
VALUES ('CSC101', 'Introduction to Computer Science', 3),
       ('ENG101', 'English Composition', 3),
       ('MAT101', 'Calculus I', 4);

INSERT INTO Department (DepartmentCode, Name)
VALUES ('CSC', 'Computer Science'),
       ('ENG', 'English'),
       ('MAT', 'Mathematics');

INSERT INTO Teacher (EmployeeCode, Name, Department, Designation)
VALUES ('T001', 'John Doe', 'CSC', 'Associate Professor'),
       ('T002', 'Jane Smith', 'ENG', 'Assistant Professor'),
       ('T003', 'Mike Johnson', 'MAT', 'Professor');

INSERT INTO Courses_Offerred (CourseNo, Department, Teacher, Programme)
VALUES ('CSC101', 'CSC', 'T001', 'BTech'),
       ('ENG101', 'ENG', 'T002', 'MTech'),
       ('MAT101', 'MAT', 'T003', 'MCA');

INSERT INTO Student (RollNo, Name, Address, Semester)
VALUES ('S001', 'Alice Smith', '123 Main St', 1),
       ('S002', 'Bob Johnson', '456 Maple Ave', 2),
       ('S003', 'Carol Lee', '789 Oak St', 3);

INSERT INTO Enrollment (RollNo, ProgrammeCode, RegistrationNo)
VALUES ('S001', 'BTech', 'R001'),
       ('S002', 'MTech', 'R002'),
       ('S003', 'MCA', 'R003');

INSERT INTO Registration (RollNo, CourseNo, Semester)
VALUES ('S001', 'CSC101', 1),
       ('S002', 'ENG101', 2),
       ('S003', 'MAT101', 3);

INSERT INTO Grade (RollNo, CourseNo, Grade)
VALUES ('S001', 'CSC101', 3.5),
```

```
        ('S002', 'ENG101', 4.0),
        ('S003', 'MAT101', 3.7);
```

Tables:

```
MariaDB [Assignment_2_3]> select * from Academic_Programmes;
+--------------------------------+----------------+----------+
| Name                           | Programme_Code | Duration |
+--------------------------------+----------------+----------+
| Bachelor of Technology         | BTech          |        4 |
| Master of Computer Applications | MCA            |        3 |
| Master of Technology           | MTech          |        2 |
+--------------------------------+----------------+----------+
3 rows in set (0.001 sec)

MariaDB [Assignment_2_3]> select * from Course;
+----------+---------------------------------+---------+
| CourseNo | Title                           | Credits |
+----------+---------------------------------+---------+
| CSC101   | Introduction to Computer Science |       3 |
| ENG101   | English Composition             |       3 |
| MAT101   | Calculus I                      |       4 |
+----------+---------------------------------+---------+
3 rows in set (0.000 sec)

MariaDB [Assignment_2_3]> select * from Courses_Offerred;
+----------+------------+---------+-----------+
| CourseNo | Department | Teacher | Programme |
+----------+------------+---------+-----------+
| CSC101   | CSC        | T001    | BTech     |
| ENG101   | ENG        | T002    | MTech     |
| MAT101   | MAT        | T003    | MCA       |
+----------+------------+---------+-----------+
3 rows in set (0.000 sec)

MariaDB [Assignment_2_3]> select * from Department;
+----------------+------------------+
| DepartmentCode | Name             |
+----------------+------------------+
| CSC            | Computer Science |
| ENG            | English          |
| MAT            | Mathematics      |
+----------------+------------------+
3 rows in set (0.000 sec)

MariaDB [Assignment_2_3]> select * from Enrollment;
+--------+---------------+----------------+
| RollNo | ProgrammeCode | RegistrationNo |
+--------+---------------+----------------+
| S001   | BTech         | R001           |
| S002   | MTech         | R002           |
| S003   | MCA           | R003           |
+--------+---------------+----------------+
3 rows in set (0.000 sec)
```

```
MariaDB [Assignment_2_3]> select * from Grade;
+--------+----------+-------+
| RollNo | CourseNo | Grade |
+--------+----------+-------+
| S001   | CSC101   |   3.5 |
| S002   | ENG101   |     4 |
| S003   | MAT101   |   3.7 |
+--------+----------+-------+
3 rows in set (0.000 sec)

MariaDB [Assignment_2_3]> select * from Registration;
+--------+----------+----------+
| RollNo | CourseNo | Semester |
+--------+----------+----------+
| S001   | CSC101   |        1 |
| S002   | ENG101   |        2 |
| S003   | MAT101   |        3 |
+--------+----------+----------+
3 rows in set (0.000 sec)

MariaDB [Assignment_2_3]> select * from Student;
+--------+-------------+---------------+----------+
| RollNo | Name        | Address       | Semester |
+--------+-------------+---------------+----------+
| S001   | Alice Smith | 123 Main St   |        1 |
| S002   | Bob Johnson | 456 Maple Ave |        2 |
| S003   | Carol Lee   | 789 Oak St    |        3 |
+--------+-------------+---------------+----------+
3 rows in set (0.000 sec)

MariaDB [Assignment_2_3]> select * from Teacher;
+--------------+-------------+------------+----------------------+
| EmployeeCode | Name        | Department | Designation          |
+--------------+-------------+------------+----------------------+
| T001         | John Doe    | CSC        | Associate Professor  |
| T002         | Jane Smith  | ENG        | Assistant Professor  |
| T003         | Mike Johnson | MAT       | Professor            |
+--------------+-------------+------------+----------------------+
3 rows in set (0.000 sec)
```

Queries:

1.1. Find all the students' name, city, course allotted from the CSE department.

```sql
SELECT s.Name, s.Address, c.Title
FROM Student s
INNER JOIN Registration r ON s.RollNo = r.RollNo
INNER JOIN Courses_Offerred co ON r.CourseNo = co.CourseNo
INNER JOIN Course c ON r.CourseNo = c.CourseNo
WHERE co.Department = 'CSC';
```

```
MariaDB [Assignment_2_3]> SELECT s.Name, s.Address, c.Title
    -> FROM Student s
    -> INNER JOIN Registration r ON s.RollNo = r.RollNo
    -> INNER JOIN Courses_Offerred co ON r.CourseNo = co.CourseNo
    -> INNER JOIN Course c ON r.CourseNo = c.CourseNo
    -> WHERE co.Department = 'CSC';
+-------------+-------------+---------------------------------+
| Name        | Address     | Title                           |
+-------------+-------------+---------------------------------+
| Alice Smith | 123 Main St | Introduction to Computer Science |
+-------------+-------------+---------------------------------+
1 row in set (0.001 sec)
```

1.2. List the total number of Faculty in the CSE department.

```sql
SELECT COUNT(*) as TotalFaculty
FROM Teacher t
INNER JOIN Department d ON t.Department = d.DepartmentCode
WHERE d.DepartmentCode = 'CSC';
```

```
MariaDB [Assignment_2_3]> SELECT COUNT(*) as TotalFaculty
    -> FROM Teacher t
    -> INNER JOIN Department d ON t.Department = d.DepartmentCode
    -> WHERE d.DepartmentCode = 'CSC';
+--------------+
| TotalFaculty |
+--------------+
|            1 |
+--------------+
1 row in set (0.000 sec)
```

1.3. List the available courses from the CSE department.

```sql
SELECT c.Title, c.Credits
FROM Courses_Offerred co
INNER JOIN Course c ON co.CourseNo = c.CourseNo
INNER JOIN Department d ON co.Department = d.DepartmentCode
WHERE d.DepartmentCode = 'CSC';
```

```
MariaDB [Assignment_2_3]> SELECT c.Title, c.Credits
    -> FROM Courses_Offerred co
    -> INNER JOIN Course c ON co.CourseNo = c.CourseNo
    -> INNER JOIN Department d ON co.Department = d.DepartmentCode
    -> WHERE d.DepartmentCode = 'CSC';
+---------------------------------+---------+
| Title                           | Credits |
+---------------------------------+---------+
| Introduction to Computer Science |       3 |
+---------------------------------+---------+
1 row in set (0.001 sec)
```

1.4. List the all students in a particular semester.

```sql
SELECT Name
FROM Student
WHERE Semester = 2;
```

```
MariaDB [Assignment_2_3]> SELECT Name
    -> FROM Student
    -> WHERE Semester = 2;
+-------------+
| Name        |
+-------------+
| Bob Johnson |
+-------------+
1 row in set (0.000 sec)
```

1.5. List the students who earned CGPA greater than or equal to 8.5

```sql
SELECT s.Name
FROM Student s
INNER JOIN Grade g ON s.RollNo = g.RollNo
GROUP BY s.RollNo
HAVING AVG(g.Grade) >= 8.5;
```

```
MariaDB [Assignment_2_3]> SELECT s.Name
    -> FROM Student s
    -> INNER JOIN Grade g ON s.RollNo = g.RollNo
    -> GROUP BY s.RollNo
    -> HAVING AVG(g.Grade) >= 8.5;
Empty set (0.001 sec)
```

2.1. How much subjects are registered by a student in each semester.

```sql
SELECT s.RollNo, s.Name, r.Semester, COUNT(r.CourseNo) AS NumCourses
FROM Student s
INNER JOIN Registration r ON s.RollNo = r.RollNo
GROUP BY s.RollNo, r.Semester;
```

```
MariaDB [Assignment_2_3]> SELECT s.RollNo, s.Name, r.Semester, COUNT(r.CourseNo) AS NumCourses
    -> FROM Student s
    -> INNER JOIN Registration r ON s.RollNo = r.RollNo
    -> GROUP BY s.RollNo, r.Semester;
+--------+-------------+----------+------------+
| RollNo | Name        | Semester | NumCourses |
+--------+-------------+----------+------------+
| S001   | Alice Smith |        1 |          1 |
| S002   | Bob Johnson |        2 |          1 |
| S003   | Carol Lee   |        3 |          1 |
+--------+-------------+----------+------------+
3 rows in set (0.000 sec)
```

2.2. List the common students who are allotted the same courses of both the programme MCA and M.Tech.

```sql
SELECT s.RollNo, s.Name, c.CourseNo
FROM Student s
INNER JOIN Enrollment e ON s.RollNo = e.RollNo
INNER JOIN Registration r ON r.RollNo = s.RollNo
INNER JOIN Course c ON c.CourseNo = r.CourseNo
WHERE e.ProgrammeCode IN ('MCA', 'MTech')
GROUP BY s.RollNo, c.CourseNo
HAVING COUNT(DISTINCT e.ProgrammeCode) = 2;
```

```
MariaDB [Assignment_2_3]> SELECT s.RollNo, s.Name, c.CourseNo
    -> FROM Student s
    -> INNER JOIN Enrollment e ON s.RollNo = e.RollNo
    -> INNER JOIN Registration r ON r.RollNo = s.RollNo
    -> INNER JOIN Course c ON c.CourseNo = r.CourseNo
    -> WHERE e.ProgrammeCode IN ('MCA', 'MTech')
    -> GROUP BY s.RollNo, c.CourseNo
    -> HAVING COUNT(DISTINCT e.ProgrammeCode) = 2;
Empty set (0.001 sec)
```

2.3. List the total number of student enrolled in the subject DBMS.

```sql
SELECT COUNT(DISTINCT RollNo) AS Total_Students
FROM Registration
WHERE CourseNo = 'DBMS';
```

```
MariaDB [Assignment_2_3]> SELECT COUNT(DISTINCT RollNo) AS Total_Students
    -> FROM Registration
    -> WHERE CourseNo = 'DBMS';
+----------------+
| Total_Students |
+----------------+
|              0 |
+----------------+
1 row in set (0.001 sec)
```

2.4. Retrieve the semester of the student under DBMS subject.

```sql
SELECT DISTINCT s.Semester
FROM Student s
INNER JOIN Registration r ON s.RollNo = r.RollNo
WHERE r.CourseNo = 'DBMS';
```

```
MariaDB [Assignment_2_3]> SELECT DISTINCT s.Semester
    -> FROM Student s
    -> INNER JOIN Registration r ON s.RollNo = r.RollNo
    -> WHERE r.CourseNo = 'DBMS';
Empty set (0.000 sec)
```

2.5. Retrieve all the student name and arrange into ascending order.

```sql
SELECT Name
FROM Student
ORDER BY Name ASC;
```

```
MariaDB [Assignment_2_3]> SELECT Name
    -> FROM Student
    -> ORDER BY Name ASC;
+-------------+
| Name        |
+-------------+
| Alice Smith |
| Bob Johnson |
| Carol Lee   |
+-------------+
3 rows in set (0.000 sec)
```