University Examination System

Design an Entity-Relationship schema for a university examination system that manages data about exams, students, faculty members, courses, and departments.
Each department has a unique name and is headed by a faculty member. A department can offer multiple courses, and each course has a unique course code, title, and is coordinated by a faculty member.
Faculty members have an employee ID, name, and designation. They can teach multiple courses, coordinate specific courses, and also serve as heads of departments. A faculty member may handle multiple roles at once.
Students have a roll number and name, and each student belongs to one department. A student can enroll in multiple courses offered by that department. For each enrolled course, a student has an attendance percentage recorded.
Exams are created by faculty members. Each exam has a title, subject name (which is assumed to be the same as the course name), duration, date, type (internal or external), and is always linked to a specific course. Students may appear in multiple exams related to their courses, and for each exam, a student may have multiple attempts, with marks and attempt dates recorded for each.
All relationships between students, courses, faculty, and exams must reflect these associations clearly — such as student-course enrollment, faculty-course teaching, course-department mapping, and exam-course ownership.

🧾 SQL Table Creation Statements:

1) Department Table

```
CREATE TABLE Department (
    dept_name VARCHAR(100) PRIMARY KEY,
    head_faculty_id INT
);
```

2) Faculty Member Table

```
CREATE TABLE Faculty (
    emp_id INT PRIMARY KEY,
    name VARCHAR(100),
    designation VARCHAR(100)
);
```

3) Course Table

```
CREATE TABLE Course (
    course_code VARCHAR(10) PRIMARY KEY,
    title VARCHAR(100),
    dept_name VARCHAR(100),
    coordinator_id INT,
    FOREIGN KEY (dept_name) REFERENCES Department(dept_name),
    FOREIGN KEY (coordinator_id) REFERENCES Faculty(emp_id)
);
```

4) Faculty-Course Mapping Table (teaching assignments)

```
CREATE TABLE FacultyCourse (
    emp_id INT,
    course_code VARCHAR(10),
    PRIMARY KEY (emp_id, course_code),
    FOREIGN KEY (emp_id) REFERENCES Faculty(emp_id),
    FOREIGN KEY (course_code) REFERENCES Course(course_code)    );
```

5) Student Table

```
CREATE TABLE Student (
    roll_number INT PRIMARY KEY,
    name VARCHAR(100),
    dept_name VARCHAR(100),
    FOREIGN KEY (dept_name) REFERENCES Department(dept_name)
);
```

6) Student-Course Enrollment with Attendance

```
CREATE TABLE StudentCourse (
    roll_number INT,
    course_code VARCHAR(10),
    attendance_percentage DECIMAL(5,2),
    PRIMARY KEY (roll_number, course_code),
    FOREIGN KEY (roll_number) REFERENCES Student(roll_number),
    FOREIGN KEY (course_code) REFERENCES Course(course_code)
);
```

7) Exam Table

```
CREATE TABLE Exam (
    exam_id INT PRIMARY KEY,
    title VARCHAR(100),
    subject_name VARCHAR(100),
    duration INT,
    exam_date DATE,
    exam_type VARCHAR(20), -- Internal/External
    course_code VARCHAR(10),
    created_by INT,
    FOREIGN KEY (course_code) REFERENCES Course(course_code),
    FOREIGN KEY (created_by) REFERENCES Faculty(emp_id)
);
```

8) Exam Attempts Table (Students attempting exams)

```
CREATE TABLE ExamAttempt (
    attempt_id INT PRIMARY KEY,
    exam_id INT,
    roll_number INT,
    marks_obtained DECIMAL(5,2),
    attempt_number INT,
    FOREIGN KEY (exam_id) REFERENCES Exam(exam_id),
    FOREIGN KEY (roll_number) REFERENCES Student(roll_number)
);
```

🧾Values of Each Table :-

1) Department Table

```
 INSERT INTO
  Department (dept_name, head_faculty_id)
VALUES
  ('Computer Science', 101),
  ('Mechanical Engineering', 102),
  ('Electrical Engineering', 103);
```

2) Faculty Member Table

```
INSERT INTO Faculty (emp_id, name, designation) VALUES
(101, 'Dr. Asha Mehta', 'Professor'),
(102, 'Dr. Rajiv Nair', 'Associate Professor'),
(103, 'Dr. Meena Verma', 'Assistant Professor');
```

3) Course Table

```
INSERT INTO Course (course_code, title, dept_name, coordinator_id) VALUES
('CS101', 'Data Structures', 'Computer Science', 101),
('ME201', 'Thermodynamics', 'Mechanical Engineering', 102),
('EE301', 'Circuit Analysis', 'Electrical Engineering', 103);
```

4) Faculty-Course Mapping Table (teaching assignments)

```
INSERT INTO FacultyCourse (emp_id, course_code) VALUES
(101, 'CS101'),
(102, 'ME201'),
(103, 'EE301');
```

5) Student Table

```
INSERT INTO Student (roll_number, name, dept_name) VALUES
(1001, 'Anjali Sharma', 'Computer Science'),
(1002, 'Ravi Kumar', 'Mechanical Engineering'),
(1003, 'Pooja Singh', 'Electrical Engineering');
```

6) Student-Course Enrollment with Attendance

```
INSERT INTO Student_Course (roll_number, course_code,
attendance_percentage) VALUES
(1001, 'CS101', 92.5),
(1002, 'ME201', 85.0),
(1003, 'EE301', 78.3);
```

7) Exam Table

INSERT INTO Exam (exam_id, title, subject_name, duration, exam_date, exam_type, course_code, created_by) VALUES
(1, 'Mid Sem', 'Data Structures', 90, '2025-06-10', 'Internal', 'CS101', 101),
(2, 'End Sem', 'Thermodynamics', 120, '2025-06-15', 'External', 'ME201', 102),
(3, 'Quiz 1', 'Circuit Analysis', 60, '2025-06-20', 'Internal', 'EE301', 103);

8) Exam Attempts Table (Students attempting exams)

INSERT INTO Exam_Attempt (attempt_id, exam_id, roll_number, marks_obtained, attempt_number) VALUES
(1, 1, 1001, 88.0, 1),
(2, 2, 1002, 72.5, 1),
(3, 3, 1003, 67.0, 1);

SYSTEM GENERATED ER DIAGRAM