Problem 2:University Examination System

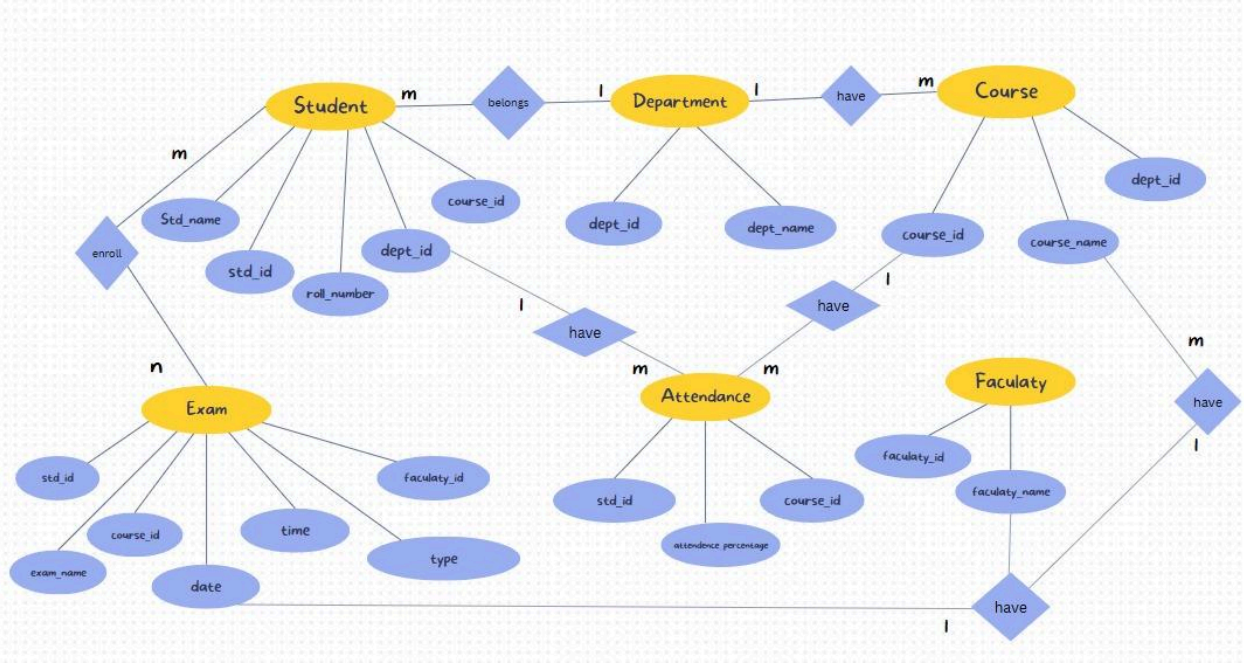
Design an Entity-Relationship schema for a university examination system that manages data about **exams**, **students**, **faculty members**, **courses**, and **departments**.

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.



🧑‍🎓 Student

- Each student (`S_id`, `S_Name`) belongs to **one department,** **roll_number,dept_id,course_id**.
- A student **enrolls in courses**
- A student **gives exams** and **has attendance** in each course

---

🏢 Department

- Each department (`D_id`, `D_Name`) offers multiple **courses**
- Examples:
    - D_id = 108, D_Name = "ECE"
    - D_id = 420, D_Name = "EEE"

📚 Course

- Each course (`C_id`, `C_Name`) belongs to **one department**
- A course is **taught by faculty**
- Students **attend** courses and give **exams** in them.

👨‍🏫 Faculty

- Each faculty (`F_ID`, `Faculty_Name`) can:
    - **Teach courses**
    - **Create exams** (examiner)

📝 Exam

- Each exam has:
    - `Std_ID,Exam_name` (Exam ID)
    - `Course_name` (e.g., Java, ML, DBMS)
    - `Date`, `Time`, `Type` (internal/external)
    - Created by a faculty (`Faculty_id`)
- Students attempt exams for courses they're enrolled in.

📊 Attendance

- Attendance is recorded per **student per course**
- Contains: `S_id`, `C_id`, and `Attendance_Percentage`

📋 3. SQL TABLE CREATION STATEMENTS

🧑‍🎓 Student Table

```
CREATE TABLE Student (
    Std_id INT PRIMARY KEY,
    Std_Name VARCHAR(100),
    Roll_number int,
    D_id INT,
    Course_id int
    FOREIGN KEY (course_id) REFERENCES Course(Course_id),
    FOREIGN KEY (D_id) REFERENCES Department(D_id)
);
```

🏢 Department Table

```
CREATE TABLE Department (
    D_id INT PRIMARY KEY,
    D_Name VARCHAR(100)
);
```

📚 Course Table

```
CREATE TABLE Course (
    Course_id INT PRIMARY KEY,
    Course_Name VARCHAR(100),
    D_id INT,
    FOREIGN KEY (D_id) REFERENCES Department(D_id)
);
```

👨‍🏫 Faculty Table

```
CREATE TABLE Faculty (
    F_ID INT PRIMARY KEY,
    Faculty_Name VARCHAR(100)
     );
```

📝 Exam Table

```
CREATE TABLE Exam (
    Exam_name VARCHAR(50),
    Date DATE,
    Time INT, -- in minutes
    Type VARCHAR(50), -- internal,external,online,offline
    Std_id INT,
    Course_id INT,
    F_id INT,
    FOREIGN KEY (Std_id) REFERENCES Student(Std_id),
    FOREIGN KEY (Course_id) REFERENCES Course(Course_id), -- Now valid
```

```
    FOREIGN KEY (F_id) REFERENCES Faculty(F_id)
);
```

📊 Attendance Table

```
CREATE TABLE Attendance (
    Std_id INT,
    Course_id INT,
    Attendance_Percentage DECIMAL(5,2),
    PRIMARY KEY (S_id, C_id),
    FOREIGN KEY (Std_id) REFERENCES Student(Std_id),
    FOREIGN KEY (Course_id) REFERENCES Course(Course_id)
);
```

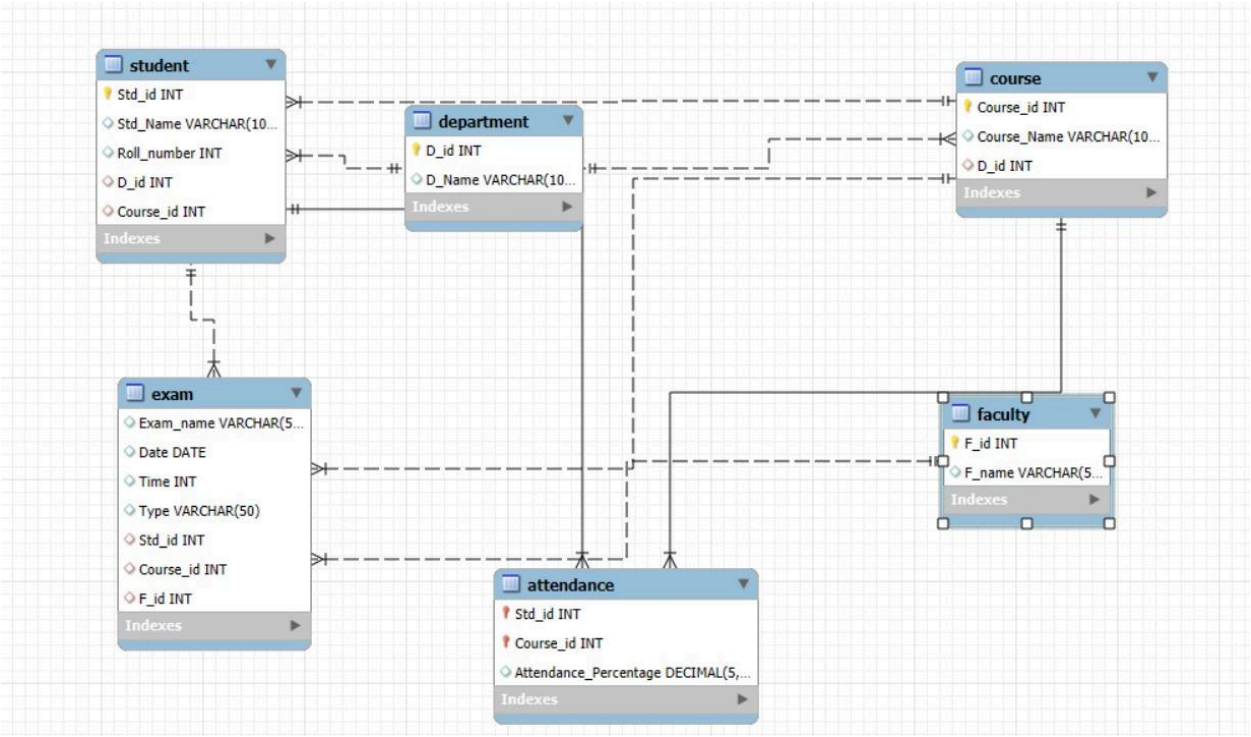
/* ========== Insert the values ========== */

```
-- Insert into Student
INSERT INTO Student (Std_id, Std_Name, Roll_number, D_id, Course_id)
VALUES
(5001, 'Alice Johnson', 101, 1, 101),
(5002, 'Bob Smith', 102, 1, 102),
(5003, 'Charlie Brown', 201, 2, 201),
(5004, 'David Wilson', 202, 2, 202),
(5005, 'Eva Davis', 301, 3, 301),
(5006, 'Frank Miller', 401, 4, 401),
(5007, 'Grace Taylor', 501, 5, 501);
-- Insert into Department
INSERT INTO Department (D_id, D_Name) VALUES
(1, 'Computer Science'),
(2, 'Electrical Engineering'),
(3, 'Mechanical Engineering'),
(4, 'Mathematics'),
(5, 'Physics');

-- Insert into Course
INSERT INTO Course (Course_id, Course_Name, D_id) VALUES
(101, 'Database Systems', 1),
(102, 'Algorithms', 1),
(201, 'Circuit Theory', 2),
(202, 'Power Systems', 2),
(301, 'Thermodynamics', 3),
```

```
(401, 'Calculus', 4),
(501, 'Quantum Mechanics', 5);
-- Insert into Faculty
INSERT INTO Faculty (F_ID, Faculty_Name) VALUES
(1001, 'Dr. Smith'),
(1002, 'Prof. Johnson'),
(1003, 'Dr. Williams'),
(1004, 'Prof. Brown'),
(1005, 'Dr. Davis'),
;
INSERT INTO Exam (Exam_name, Date, Time, Type, Std_id, Course_id, F_id)
VALUES
('Midterm DB', '2023-10-15', 90, 'internal', 5001, 101, 1001),
('Final Algo', '2023-12-10', 120, 'external', 5002,102, 1002),
('Circuit Lab', '2023-11-20', 60, 'online', 5003,201, 1003),
('Power Quiz', '2023-10-30', 45, 'internal', 5004,202, 1004),
('Thermo Final', '2023-12-15', 180, 'external', 5005,204, 1005),
;


INSERT INTO Attendance (Std_id, Course_id, Attendance_Percentage) VALUES
(5001, 101, 95.50),
(5002, 102, 88.75),
(5003, 201, 92.00),
(5004, 202, 85.25),
(5005, 301, 90.50),
(5006, 401, 97.00),
(5007, 501, 89.75);
```


student
Std_id INT
Std_Name VARCHAR(10...
Roll_number INT
D_id INT
Course_id INT
Indexes
department
D_id INT
D_Name VARCHAR(10...
Indexes
course
Course_id INT
Course_Name VARCHAR(10...
D_id INT
Indexes
exam
Exam_name VARCHAR(5...
Date DATE
Time INT
Type VARCHAR(50)
Std_id INT
Course_id INT
F_id INT
Indexes
faculty
F_id INT
F_name VARCHAR(5...
Indexes
attendance
Std_id INT
Course_id INT
Attendance_Percentage DECIMAL(5,...
Indexes