**Problem 2: University Examination System**

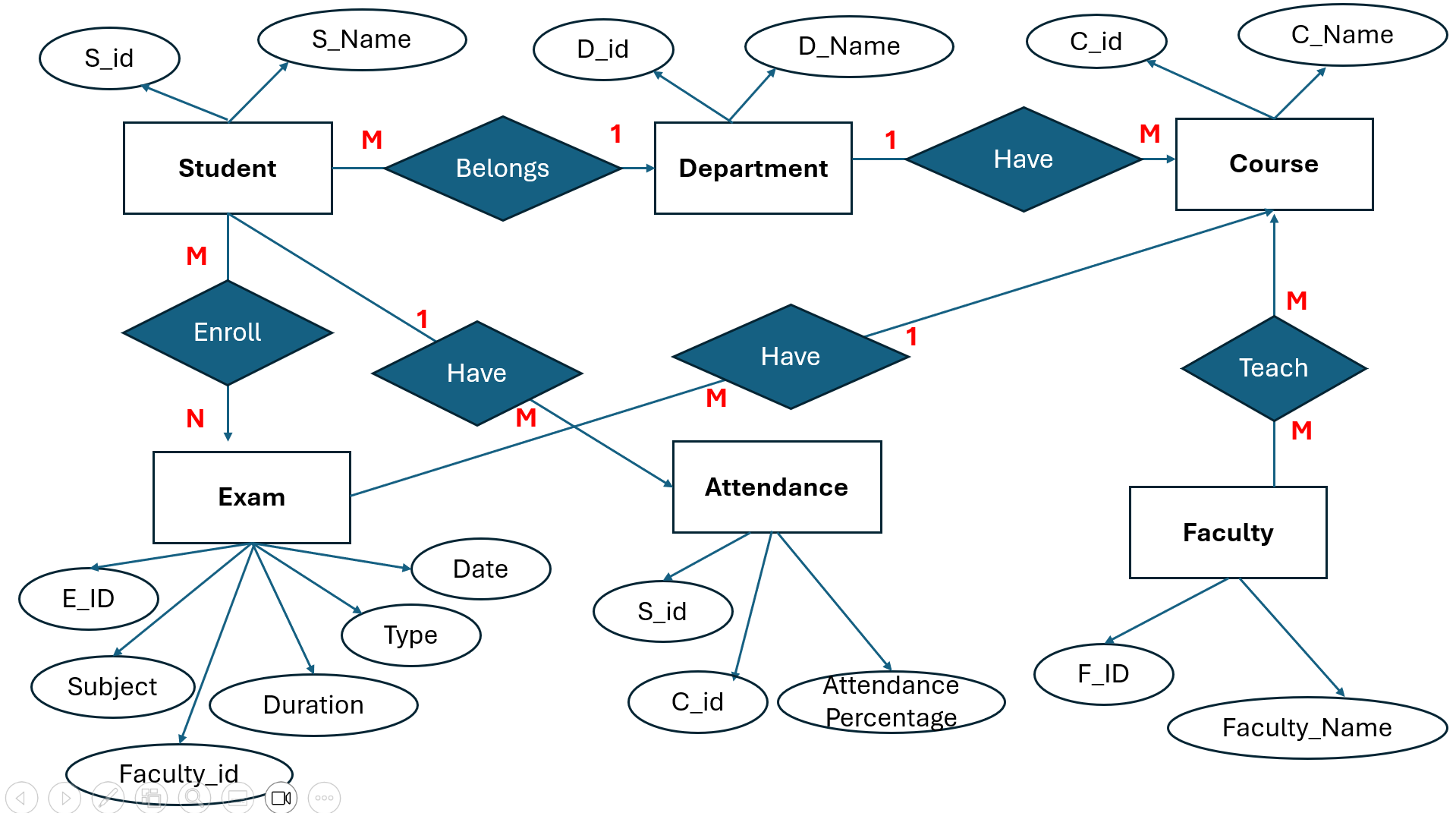
Design an Entity-Relationship schema for a university examination system that manages data about **exams**, **students**, **faculty members**, **courses**, and **departments**.

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**
* 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 = 101, D\_Name = "CSE"
  + D\_id = 102, D\_Name = "Electrical"

### **📚 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:  
  + E\_ID (Exam ID)
  + Subject (e.g., Java, ML, DBMS)
  + Date, Duration, 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 (

S\_id INT PRIMARY KEY,

S\_Name VARCHAR(100),

D\_id INT,

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 (

C\_id INT PRIMARY KEY,

C\_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)

);

### **🔁 Faculty Teaches Course**

CREATE TABLE Teaches (

F\_ID INT,

C\_id INT,

FOREIGN KEY (F\_ID) REFERENCES Faculty(F\_ID),

FOREIGN KEY (C\_id) REFERENCES Course(C\_id)

);

### **📝 Exam Table**

CREATE TABLE Exam (

Exam\_ID INT PRIMARY KEY,

Subject VARCHAR(100),

Type VARCHAR(50), -- 'Internal' or 'External'

Date DATE,

Duration INT, -- In minutes

Faculty\_id INT,

C\_id INT,

FOREIGN KEY (Faculty\_id) REFERENCES Faculty(F\_ID),

FOREIGN KEY (C\_id) REFERENCES Course(C\_id)

);

**📊 Attendance Table**

CREATE TABLE Attendance (

S\_id INT,

C\_id INT,

Attendance\_Percentage DECIMAL(5,2),

PRIMARY KEY (S\_id, C\_id),

FOREIGN KEY (S\_id) REFERENCES Student(S\_id),

FOREIGN KEY (C\_id) REFERENCES Course(C\_id)

);

### 

### **📌 OPTIONAL: Enrollment Table (Recommended)**

CREATE TABLE Enrollment (

S\_id INT,

C\_id INT,

PRIMARY KEY (S\_id, C\_id),

FOREIGN KEY (S\_id) REFERENCES Student(S\_id),

FOREIGN KEY (C\_id) REFERENCES Course(C\_id)

);

/\* ========== DATA INSERTION ========== \*/

-- Department

INSERT INTO Department (D\_id, D\_Name) VALUES

(1, 'Computer Science'),

(2, 'Mechanical Engineering'),

(3, 'Electrical Engineering');

-- Student

INSERT INTO Student (S\_id, S\_Name, D\_id) VALUES

(101, 'Ravi Kumar', 1),

(102, 'Anita Sharma', 2),

(103, 'Mohit Verma', 3);

-- Course

INSERT INTO Course (C\_id, C\_Name, D\_id) VALUES

(201, 'Data Structures', 1),

(202, 'Thermodynamics', 2),

(203, 'Circuit Theory', 3);

-- Faculty

INSERT INTO Faculty (F\_ID, Faculty\_Name) VALUES

(301, 'Dr. Ramesh'),

(302, 'Prof. Neha'),

(303, 'Dr. Singh');

-- Teaches

INSERT INTO Teaches (F\_ID, C\_id) VALUES

(301, 201),

(302, 202),

(303, 203);

-- Exam

INSERT INTO Exam (Exam\_ID, Subject, Type, Date, Duration, Faculty\_id, C\_id) VALUES

(401, 'Data Structures', 'Internal', '2025-06-20', 90, 301, 201),

(402, 'Thermodynamics', 'External', '2025-06-22', 120, 302, 202),

(403, 'Circuit Theory', 'Internal', '2025-06-24', 60, 303, 203);

-- Attendance

INSERT INTO Attendance (S\_id, C\_id, Attendance\_Percentage) VALUES

(101, 201, 85.50),

(102, 202, 78.25),

(103, 203, 92.00);

-- Enrollment

INSERT INTO Enrollment (S\_id, C\_id) VALUES

(101, 201),

(102, 202),

(103, 203);

