# Project 2: University Examination System

## ◆ Project Title:

### ER Schema Design for University Examination System

## ◆ Objective:

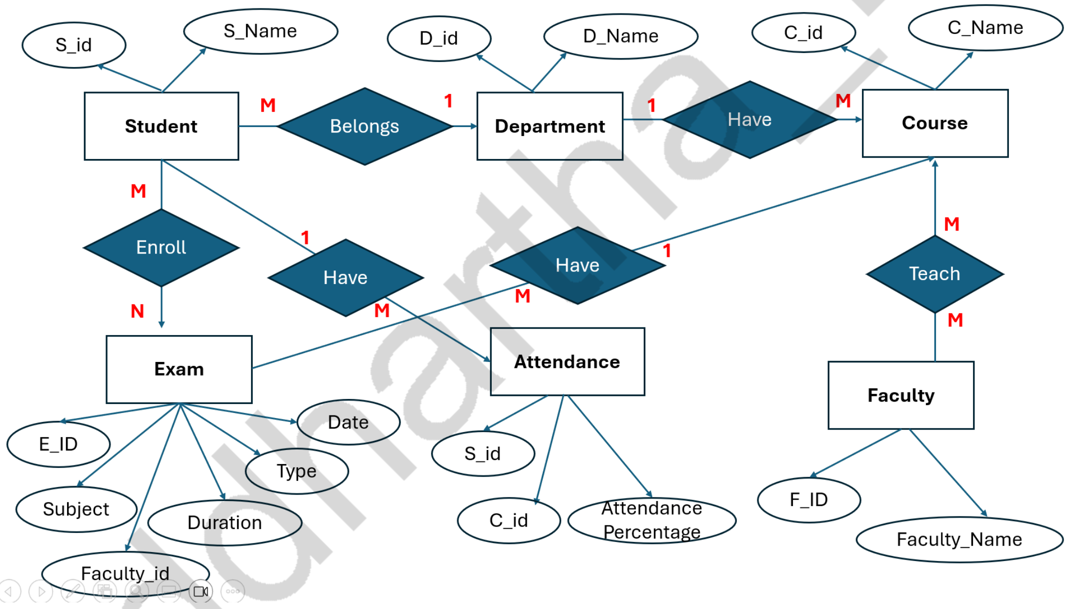
To develop an ER schema and corresponding relational database for managing university exams, including data about students, faculty members, departments, courses, and exams.

## ◆ Key Requirements:

- **Departments**
  - Attributes: Dept_Name (unique), Head_ID (Faculty)
  - Each department is headed by one faculty member.
  - A department can offer multiple courses.
- **Faculty**
  - Attributes: Emp_ID, Name, Designation
  - A faculty member may teach multiple courses, coordinate courses, and head a department simultaneously.
- **Courses**
  - Attributes: Course_Code (unique), Title, Coordinator_ID (Faculty), Dept_Name (Foreign Key)
  - Each course belongs to one department and is coordinated by one faculty member.
- **Students**
  - Attributes: Roll_No, Name, Dept_Name (Foreign Key)

    - A student belongs to one department and can enroll in multiple courses of that department.

- **Enrollments**
    - For each course a student is enrolled in, their attendance percentage is recorded.
- **Exams**
    - Attributes: Exam_ID, Title, Course_Code, Duration, Date, Type (internal/external), Created_By (Faculty)
    - Each exam is linked to a course and created by a faculty member.
- **Exam_Attempts**
    - Records each student's attempt in an exam with date and marks.



## 🧑‍🎓 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`

---

## 🏢 Department Table

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

---

## 📋 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));
```

---

### 📚 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 ========== */

### 1. Department Table

```
Department
INSERT INTO Department (D_id, D_Name) VALUES
(1, 'Computer Science'),
(2, 'Mechanical Engineering'),
(3, 'Electrical Engineering');
```

---

### 🔹 2. Faculty Table

```
Faculty
INSERT INTO Faculty (F_ID, Faculty_Name) VALUES
(301, 'Dr. Ramesh'),
(302, 'Prof. Neha'),
(303, 'Dr. Singh');
```

---

### 🔹 3. Student Table

```
Student
INSERT INTO Student (S_id, S_Name, D_id) VALUES
(101, 'Ravi Kumar', 1),
(102, 'Anita Sharma', 2),
(103, 'Mohit Verma', 3);
```

---

### 🔹 4. Course Table

```
-- Course
INSERT INTO Course (C_id, C_Name, D_id) VALUES
(201, 'Data Structures', 1),
(202, 'Thermodynamics', 2),
(203, 'Circuit Theory', 3);
```

---

### 🔹 5. Teaches Table

```
Teaches
INSERT INTO Teaches (F_ID, C_id) VALUES
(301, 201),
(302, 202),
(303, 203);
```

---

### 🔹 6. Exam Table

```
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);
```

---

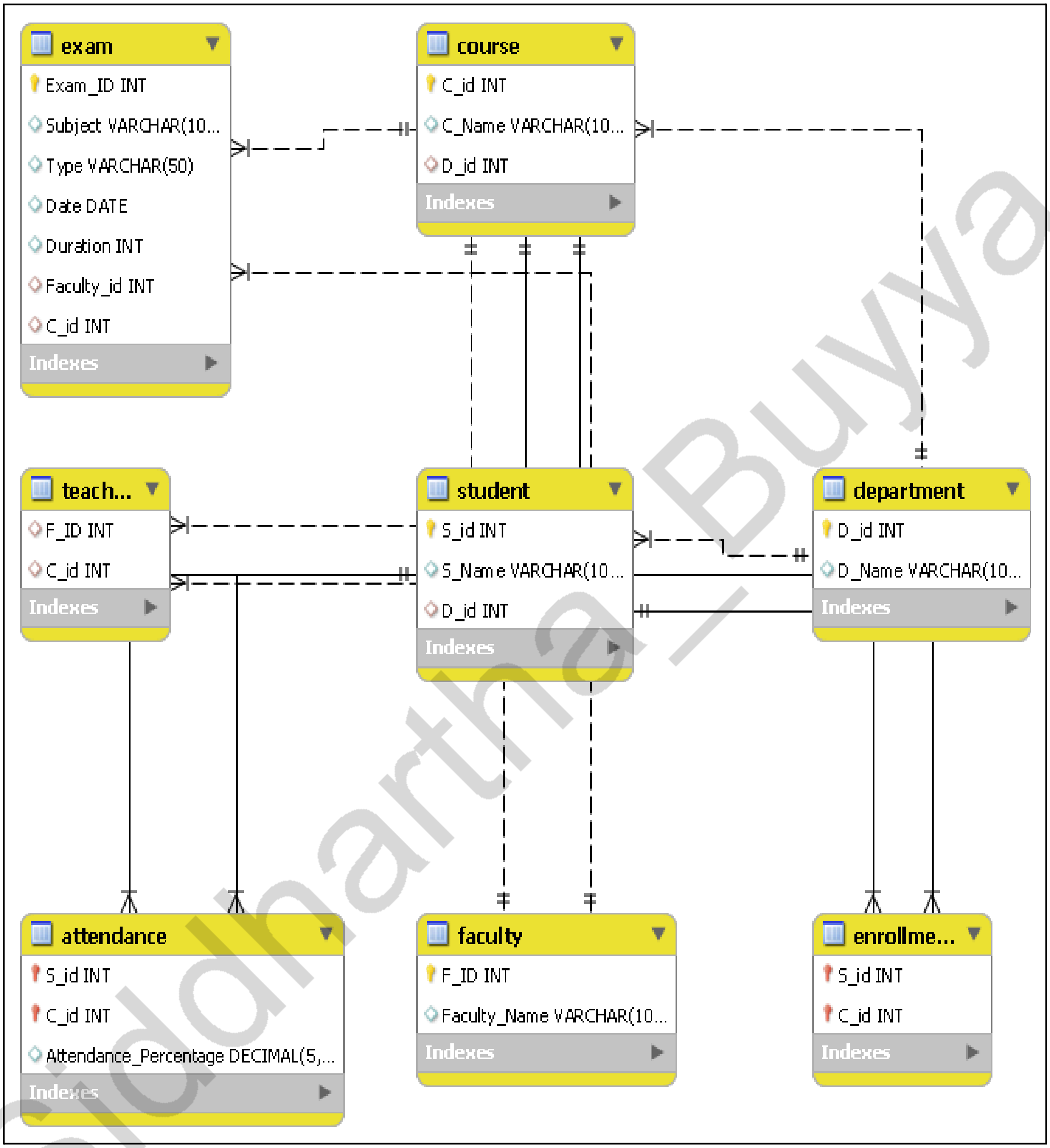
### 🔹 7. Enrollment Table

```
Enrollment
INSERT INTO Enrollment (S_id, C_id) VALUES
(101, 201),
(102, 202),
(103, 203);
```

---

### 🔹 8. Attendance Table

```
 Attendance
INSERT INTO Attendance (S_id, C_id, Attendance_Percentage) VALUES
(101, 201, 85.50),
(102, 202, 78.25),
(103, 203, 92.00);
```


exam
Exam_ID INT
Subject VARCHAR(10...
Type VARCHAR(50)
Date DATE
Duration INT
Faculty_id INT
C_id INT
Indexes
course
C_id INT
C_Name VARCHAR(10...
D_id INT
Indexes
teach...
F_ID INT
C_id INT
Indexes
student
S_id INT
S_Name VARCHAR(10...
D_id INT
Indexes
department
D_id INT
D_Name VARCHAR(10...
Indexes
attendance
S_id INT
C_id INT
Attendance_Percentage DECIMAL(5,...
Indexes
faculty
F_ID INT
Faculty_Name VARCHAR(10...
Indexes
enrollme...
S_id INT
C_id INT
Indexes