1. **Design any database with at least 3 entities and relationships between them. Draw suitable ER/EER diagram for the system.**

**Answer :**

## 📘 ****1. Entity: Students****

Represents the students who can enroll in courses.

### 🧾 Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Data Type** | **Description** |
| student\_id | INT (PK) | Unique ID for each student |
| name | VARCHAR(100) | Full name of the student |
| email | VARCHAR(100) | Unique email address |
| age | INT | Age of the student |

## 📘 ****2. Entity: Courses****

Represents the courses that students can enroll in.

### 🧾 Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Data Type** | **Description** |
| course\_id | INT (PK) | Unique ID for each course |
| course\_name | VARCHAR(100) | Name of the course |
| description | TEXT | Description of the course content |

## 📘 ****3. Entity: Enrollments**** (Associative Entity)

Represents the many-to-many relationship between Students and Courses.

### 🧾 Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Data Type** | **Description** |
| enrollment\_id | INT (PK) | Unique ID for each enrollment record |
| student\_id | INT (FK) | References Students.student\_id |
| course\_id | INT (FK) | References Courses.course\_id |
| enrollment\_date | DATE | Date when the student enrolled |

🔹 **Step 3: Create and Use the Database**

CREATE DATABASE CoursePlatform;

USE CoursePlatform;

🔹 **Step 4: Create Tables**

CREATE TABLE Students (

student\_id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(100) NOT NULL,

email VARCHAR(100) UNIQUE NOT NULL,

age INT

);

CREATE TABLE Courses (

course\_id INT AUTO\_INCREMENT PRIMARY KEY,

course\_name VARCHAR(100) NOT NULL,

description TEXT

);

CREATE TABLE Enrollments (

enrollment\_id INT AUTO\_INCREMENT PRIMARY KEY,

student\_id INT NOT NULL,

course\_id INT NOT NULL,

enrollment\_date DATE,

FOREIGN KEY (student\_id) REFERENCES Students(student\_id) ON DELETE CASCADE,

FOREIGN KEY (course\_id) REFERENCES Courses(course\_id) ON DELETE CASCADE

);

### 🔹 ****Step 5: Insert Sample Data****

#### 📥 Add 4 Indian Students

INSERT INTO Students (name, email, age) VALUES

('Aarav Sharma', 'aarav.sharma@example.com', 20),

('Ananya Singh', 'ananya.singh@example.com', 22),

('Rohan Patel', 'rohan.patel@example.com', 21),

('Priya Verma', 'priya.verma@example.com', 23);

**📥 Add 3 Courses**

INSERT INTO Courses (course\_name, description) VALUES

('Database Management Systems', 'Learn about relational databases, SQL, and normalization.'),

('Web Development', 'Covers HTML, CSS, JavaScript, and backend development.'),

('Data Structures', 'Covers various data structures and their applications.');

**📥 Add Enrollments**

INSERT INTO Enrollments (student\_id, course\_id, enrollment\_date) VALUES

(1, 1, '2024-09-01'), -- Aarav in DBMS

(1, 2, '2024-09-05'), -- Aarav in Web Dev

(2, 1, '2024-09-03'), -- Ananya in DBMS

(3, 3, '2024-09-02'), -- Rohan in DS

(4, 2, '2024-09-04'); -- Priya in Web Dev

### 🔹 ****Step 6: Show Data (Output Results)****

#### 📋 View Students:

SELECT \* FROM Students;

**📋 View Courses:**

SELECT \* FROM Courses;

**📋 View Enrollments with Student Names and Course Names:**

SELECT e.enrollment\_id, s.name AS student\_name, c.course\_name, e.enrollment\_date

FROM Enrollments e

JOIN Students s ON e.student\_id = s.student\_id

JOIN Courses c ON e.course\_id = c.course\_id;