**Comprehensive DBMS Assignment: University Database System (UDS)**

Course: Database Management Systems (DBMS)

**Assignment 1**

**Overview:** In this assignment, you will design and model a University Database System (UDS) that will include entities such as Students, Faculty, Courses, Departments, Classrooms, and Enrollments. You will define relationships between these entities and specify their cardinalities. Finally, you will create an ER diagram to represent the system and document your design choices.

1. Design an ER diagram for a University Database System (UDS).
   * The system should model the following entities:
     + **Student**: StudentID (PK), Name, DOB, Email, Phone
     + **Faculty**: FacultyID(PK), Name, Email, DepartmentID(FK)
     + **Course**: CourseID(PK), CourseName, DepartmentID, Credits
     + **Department**: DepartmentID(PK), DepartmentName, Location
     + **Enrollment**: EnrollmentID(PK), StudentID, CourseID, EnrollmentDate
     + **Classroom**: ClassroomID(PK), ClassroomName, Building, Capacity

**Note:** Identify the keys and denote it accordingly on your own (PK-Primary Key, FK-Foreign Key)

1. Define the relationships between these entities.

You now need to define relationships between the entities. Think about how these entities interact in the real world.

* + - **Enrollment:** A student can enroll in multiple courses, and a course can have many students.
    - **Faculty-Course:** A faculty member can teach multiple courses, and a course can be taught by multiple faculty members.
    - **Department-Course:** A department offers multiple courses.
    - **Department-Faculty:** A faculty member belongs to a department.
    - **Classroom-Course:** A classroom can host multiple courses, and a course can be taught in multiple classrooms.

1. Specify the cardinality of each relationship (one-to-one, one-to-many, many-to-many).
2. Deliverables (How and what to submit):
   * + Submit the ER diagram as a graphical representation (use tools like Lucidchart, Draw.io, or MS Visio).
     + Include an explanation for each entity, relationship, and cardinality.

**For Example:**

* + - 1. **Entities**: Describe each entity and its attributes. For example, the student entity holds information about a student such as name, email, and phone number.
      2. **Relationships**: Describe the relationships between the entities:
* Who is involved in each relationship.
* Why the relationship is many-to-many, one-to-many,etc.
* Describe how these relationships reflect real-world university operations.
  + - 1. **Cardinalities**: Explain the cardinality (e.g., one-to-many, many-to-many) for each relationship and provide a justification for each.

NOTE:

1. The diagram should be clear and well-structured. All entities, attributes, and relationships must be present, and cardinalities should be correctly represented.
2. There should be a correct identification of relationship types (many-to-many, one-to-many, etc.).

Answers:

1.

Relations:

1. The department has multiple students and faculty members.

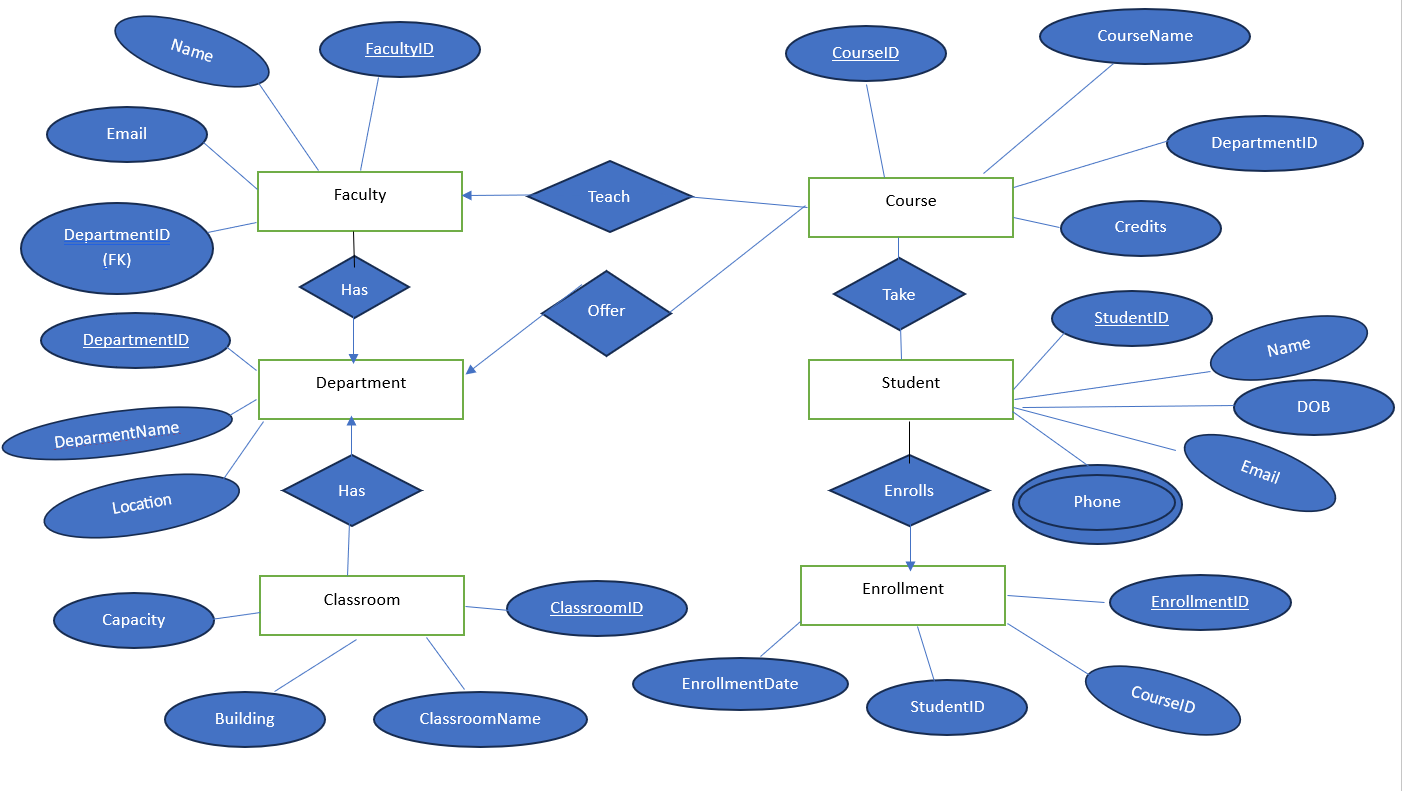
2. Each department can run multiple courses.

3. Each classroom can seat multiple students.

4. Multiple courses can be taught in one classroom.

5. Multiple students can enroll to a department as well as to the course.

6. A faculty member can teach only one course.



2.

1. Enrollment: Many to many relationships.

The Student and Course entities share a many-to-many relationship. A student can enroll in multiple courses, and a course can have multiple students enrolled.

For example, if a department offers eight courses, a student may be required to select six out of them. This means that multiple students will be enrolled in each course, and each student will be taking multiple courses.

To properly model this in an ER diagram, an associative entity (e.g., Enrollment) should be introduced to represent this many-to-many relationship. This entity can include attributes such as EnrollmentDate, Grade, and Semester to track students' progress.

2. Faculty – course: Many to Many relationships.

The Faculty and Course entities share a many-to-many relationship. A faculty member can teach multiple courses, and a course can be taught by multiple faculty members.

Some universities have rules limiting faculty members to teaching only two courses, while in other cases, a faculty member may need to teach multiple courses. Additionally, a single course may require multiple faculty members for different sections, specializations, or scheduling flexibility.

To properly model this in an ER diagram, an associative entity (e.g., Faculty\_Course) should be introduced to represent the many-to-many relationship. This entity can include attributes such as Semester, Role (Primary/Assistant Instructor), or Teaching Hours.

3. Department – Course: One to many relationships.

The Department and Course entities share a one-to-many relationship. A department can offer multiple courses, but each course is associated with only one department.

For example, the Statistics department can offer various statistics-related courses, but no other department can offer these specific courses. Similarly, courses offered by the Mathematics or Physics departments are exclusive to those departments.

In an ER diagram, this relationship is represented by a foreign key (DepartmentID) in the Course table, ensuring that each course belongs to a single department.

4. Department – Faculty: One to many relationships.

The Department and Faculty entities share a one-to-many relationship. A department can have multiple faculty members, but each faculty member belongs to only one department.

For example, faculty members from the Statistics department cannot simultaneously be part of the Mathematics or Physics department. This ensures that each faculty member is exclusively associated with a single department.

In an ER diagram, this relationship is represented by linking the Faculty entity to the Department entity using a foreign key (DepartmentID) in the Faculty table, enforcing the one-to-many constraint.

5. Classroom Course: Many to Many relationships.

Classrooms and courses share a many-to-many relationship. A course can be taught in multiple classrooms based on availability, and a classroom can host multiple courses at different times.

Additionally, each department manages multiple classrooms, and courses offered by that department can be conducted in any of these classrooms as per the schedule and availability.

To model this in an ER diagram, an associative entity (such as Course\_Schedule) can be introduced to resolve the many-to-many relationship between Courses and Classrooms, incorporating attributes like TimeSlot and Instructor.