**COURSE TITLE** : **DATA BASE MANAGEMENT SYSTEM LAB**

**COURSE CODE** : **3139**

**COURSE CATEGORY** : **B**

**PERIODS/WEEK** : **4**

**PERIODS/SEMESTER** : **60**

**CREDITS** : **2**

**Software Requirement: MySql**

**Course General Outcomes:**

| Sl. | G.O | On completion of this course the student will be able : |
|---|---|---|
| 1 | 1 | Practice different DDL,DML and DCL commands |
| | 2 | Practice queries |
| 2 | 1 | Define views |
| | 2 | Create cursors and triggers |
| 3 | 1 | Apply connections from programming languages like JAVA and PHP |
| | 2 | Implement various embedded languages |
| 4 | 1 | To Understand Normalisation and Transactions |

**Specific Outcomes:**

1.1 Create one or two tables. Bring in all possible constraints like NOT NULL,

   DEFAULT, CHECK, PRIMARY KEY, UNIQUE etc.

1.2 Enter considerable number of tuples of all sorts to the tables.

1.3 Write various SELECT clauses to generate different types of listing of data.

   (Use WHERE, HAVING, DISTINCT, GROUP BY and ORDER BY clauses and subqueries)

**Core Problem (Phase 1):** Analyse a real problem situation/application and represent it using an ER-Diagram. Create tables as per ER Diagram and insert relevant data.


2.1 Create various views of the table.

2.2 Delete unwanted tuples from tables.

2.3 Modify the schema of one or more tables to incorporate a change.

2.4 Develop queries for inner and outer joins.


3.1 Create cursors and triggers for manipulating tables.

3.2 Write functions and/or procedure for various simple tasks.

3.3 Develop a simple application in Java/PHP to make a connection to the

   database and access the data.

**Core Problem (Phase 2):**

Check whether the tables created are normalized.

Develop functions/stored procedure for identified tasks.

Develop the miniature form of application that connects to the database using JDBC/ODBC.


4.1 Design and develop a form to enter data into tables using JAVA/PHP.

4.2 Develop a Java/PHP program to access the table and generate a report.

**Core Problem (Phase 3):**

Design and develop table data entry form(s).

Develop the final application from the miniature form.

**Sample Core Problem:** The diagram shown below represents the academic functioning of a college. There are five entities viz. Department, Faculty, Student, Course, and Hostel. All the five entities have their own attributes.DNumber, FacultyID, StudentID, CourseID, and HostelID are the key attributes of Department, Faculty, Student, Course and Hostel respectively. The entities are related to each other and the respective relationships are explained below:

**A college has many departments.** A department would have students as well as faculty. The one to many relationship between department and students, and, department and faculty states that a department belongs to many students and it employs many faculty members. Looking at these relationships in a reverse direction conveys that a student as well as a faculty belongs to a single department and thus establishes one to one relationship.

**A student can register himself into various courses**; similarly a course can be studied by many students. A student lives in a single hostel but a hostel accommodates many students. A department offers many courses but a particular course belongs to a particular department. A faculty teaches many courses but a particular course is taught by a single faculty only.
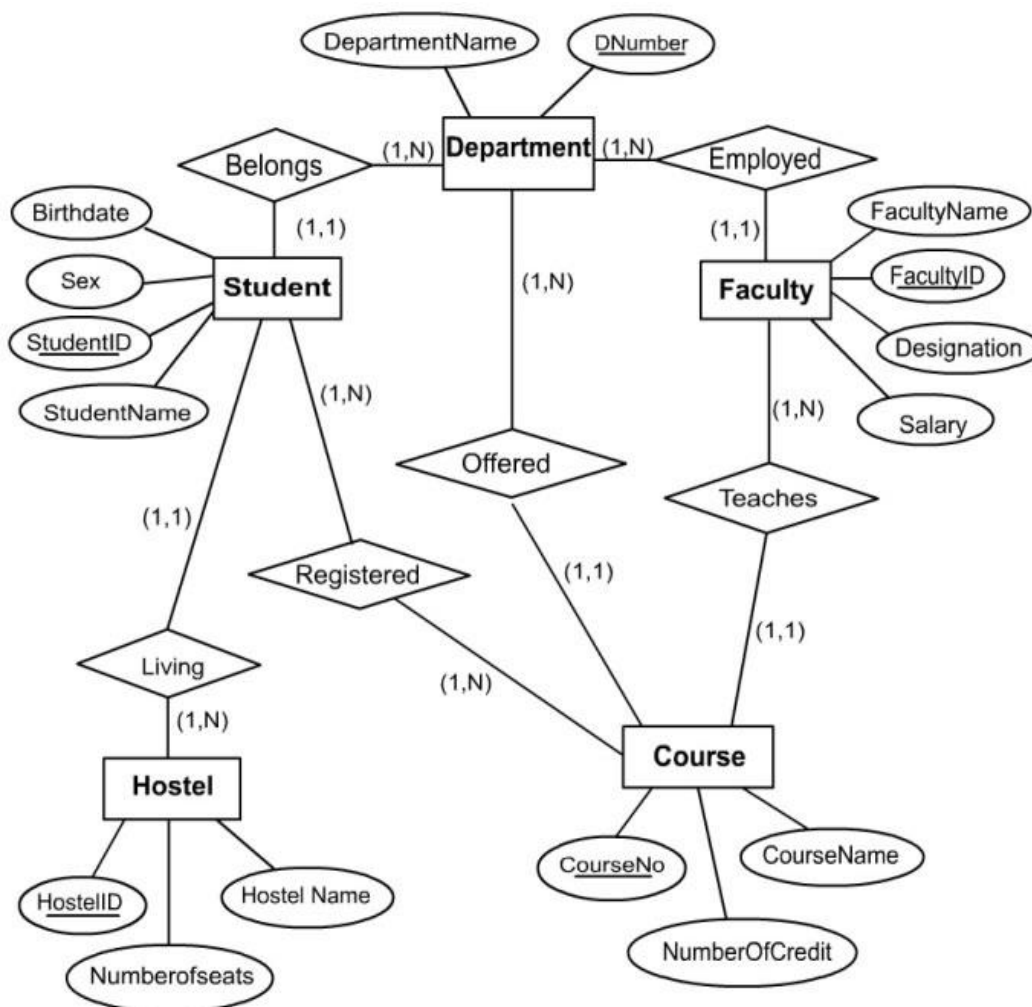
Phase 1:

**Create five tables: 1) Department, 2) Faculty, 3) Student, 4) Course and 5) Hostel. Include the necessary constraints.**

**Insert at least 10 tuples to each table.**

**Generate the following results:**

  i)     **List out the ID, Name and Date of Birth of students in the ascending order of Date of Birth**
  ii)    **List out the ID, Name and Date of Birth of students registered for a specific course.**
  iii)   **List out the ID, Name and Date of Birth of students registered for a specific course, staying in a specific Hostel.**
  iv)    **Prepare department wise student list with ID, Name and Date of Birth of students and also find number of students in each department.**


Phase 2:

**Modify the 'Student' table to add a new field 'Grade'.**

**List the names of faculties who teach for a specific course using sub-query.**


Phase 3:

**Create a cursor to modify the salary of 'Professors' belonging to all departments by 150%.**


Phase 4:

**Develop a JAVA/PHP application that allows addition, deletion and modification of all table data and generate reports.**