DBMS Case Study Assignment – University Management System

Introduction:

A small university wants to maintain a database to manage students, courses, instructors, and enrollments. The database should efficiently track student enrollments, course details, and instructor assignments while ensuring data integrity and reducing redundancy. The assignment is structured to follow the natural flow of database design: **starting from conceptual design, moving to logical design, then refining the schema through normalization, and finally implementing PL/SQL features and views**. Each task builds on the previous one, ensuring a clear interlink between conceptual modeling, schema creation, optimization, and advanced database operations.

Each Task detailed Explanation:

Task 1: Conceptual Database Design – ER Modeling (University Management System)

- Draw an ER diagram representing the university scenario.
- Include the entities: Student, Course, Instructor, and Enrollment.
- Define attributes for each entity and identify primary keys.
- Show relationships between entities with cardinalities (for example, "Student enrolls in Course").
- Map the ER diagram to a list of relational tables, showing which entities and relationships will become tables in the logical design.

Task 2: Logical Database Design – Schema Creation

- Convert the ER diagram from Task 1 into relational tables.
- Define primary keys, foreign keys, and data types for each table.

Task 3: Normalization

- Analyze the following table for redundancy and anomalies
- Identify Functional Dependencies (FDs) and candidate keys.
- Decompose the table into 3NF to remove redundancy and maintain integrity.
- Create the resulting normalized tables and insert sample data into them.

Task 4: PL/SQL and Views

- Create a trigger to prevent entering a grade above 10 in the Enrollment table.
- Create a view to display Student Name, Course Name, Instructor, and Grade.
- Write a query to update a grade using the view.