

# Assignment 1: Design and Implementation of Data Tables and their Conceptual Designs

## 1. Project Objective

The main goal of this assignment is to design, implement, and evaluate a robust database-driven software system that bridges theoretical database concepts with real-world system development. Develop a well-structured relational schema, enforce integrity and constraints, implement core functionalities, and apply conceptual modelling using UML and ER diagrams. The assignment aims to strengthen understanding of data management, normalization, transaction integrity, and systematic database design while ensuring scalability, correctness, and practical applicability.

## Core Technical Pipeline:

- **Phase 1: Requirement Analysis & System Specification**  
Identifying project scope, functionalities, and domain constraints.
- **Phase 2: Conceptual & Logical Modelling**  
Designing UML diagrams and transforming them into ER models with keys and relationships.
- **Phase 3: Schema Implementation & Integrity Enforcement**  
Creating normalized SQL tables and enforcing PK, FK, and constraint rules.
- **Phase 4: Data Validation, Testing & Documentation**  
Populating data, verifying correctness, and submitting diagrams and SQL dump.

## Deadline

15 February, 2026

---

**Instructor:** Dr. Yogesh K. Meena

January 28, 2026 • Semester II (2025 - 2026) • CS 432 – Databases(Course Project/Assignment 1)  
© 2026 Indian Institute of Technology, Gandhinagar. All rights reserved.

## 1. Assignment Overview

This assignment consists of **Two modules (Module A and Module B)**. Together, these modules focus on designing, implementing, and visualising a robust database system for the selected course project.

## 2. Module A: Database Design and Implementation

This module aims to design and implement a comprehensive database system that adheres to strict design constraints and real-world business rules. The system will demonstrate robust data management and integrity, providing a strong foundation for further development in the project domain.

### Constraints and Requirements

Each project must strictly adhere to the following constraints:

- **Member Table:** Include a Member table with attributes such as MemberID (PK), Name, Image, Age, Email, Contact Number, and other domain-relevant attributes.
- **Minimum Functionalities:** The system must support at least **five core functionalities**.
- **Minimum Entities:** The project should have **at least five distinct entities**.
- **Minimum Tables:** The database must contain **at least ten tables**.
- **Primary and Foreign Keys:** Each table must have a Primary Key or a uniquely identifiable candidate key, and relationships between tables must utilise Foreign Keys.
- **Functional Coverage:** Tables and queries must clearly support the main functionalities of the chosen project domain.
- **Real-Life Data:** Example data (synthetic data generated by project members, with at least 10-20 rows per table) should be meaningful and reflect real-world use cases.
- **Referential Integrity:** The database must satisfy referential integrity during update and delete operations.
- **NOT NULL Columns:** Each table must contain at least **three NOT NULL columns**.
- **Unique Row Identification:** Each row in any table should be uniquely identifiable (using a single attribute or a combination of attributes).
- **Logical Constraints:** Each project must follow all logical constraints relevant to the domain (e.g., exit time must be later than entry time).

### Evaluation Criteria (Module A)

All the above constraints must be satisfied for evaluation. Any violation will directly affect the score.

### 3. Module B: Conceptual Design (ER and UML Diagrams)

This module focuses on conceptual modelling of the database system using **UML** and **ER diagrams**. The diagrams must accurately reflect the schema implemented in Module A and provide a clear visual explanation of entities, relationships, and constraints.

#### Prerequisites

- Online UML/ER tools (e.g., Visio, draw.io, Lucidchart)
- A clear understanding of the database schema implemented in Module A.
- Reference:
  - UML: Chapter 10 – Ramez Elmasri and Shamkant B. Navathe
  - ER: Chapter 7 – Elmasri & Navathe; Chapter 6 – Silberschatz, Korth, Sudarshan

#### UML Diagram Creation

Start with a UML class diagram for systematic database design.

- Identify classes (entities), attributes, and methods (if applicable).
- Clearly define relationships (associations, generalisation, aggregation, or composition).
- Represent multiplicity (1:1, 1:M, M: M) properly.
- Follow standard UML conventions.

**Note:** Present as many UML diagrams as possible for your project.

#### Why UML First?

A UML class diagram provides a high-level conceptual design, helping to identify objects and their relationships before mapping them into database entities, ensuring a well-structured schema.

#### ER Diagram Creation

Based on your UML diagrams, convert them into an ER diagram, adhering to database schema constraints from Module A.

- Represent entities as rectangles and relationships as diamonds.
- Clearly mark Primary Keys (underlined) and Foreign Keys.
- Clearly define one-to-one, one-to-many, or many-to-many relationships.
- Use appropriate notation for clarity and accuracy.
- Ensure the ER diagram correctly maps to the SQL database schema from Module A.

#### How ER is Derived from UML

- Convert UML classes → ER entities
- Convert UML associations → ER relationships
- Define keys (primary & foreign) based on UML attributes.
- Adjust multiplicity (e.g., one-to-many in UML to a foreign key in ER).

### Evaluation Criteria (Module B)

- UML diagram completeness and correctness
- ER diagram accuracy and notation
- Alignment with Module A database schema
- Clarity and depth of explanation
- Justification of relationship types with examples
- Clear team member contributions

## 4. Submission Guidelines (Module A + B)

- **Module A:** Submit SQL dump file as per instructions in [Track1\\_Assignment1\\_Module\\_A\\_instructions](#).
- **Module B:** Submit a **single PDF report** containing:
  - UML diagrams
  - ER diagram
  - UML-to-ER transition explanation
  - Relationship justifications
  - Additional constraints
  - Team member names and contributions

### Important Notes

- Diagrams must strictly adhere to Module A constraints.
- Institute Honour Code applies: <https://iitgn.ac.in/students/honourcode>