Database Management Systems
Activity: One to Many Relationships

Vikas Thammanna Gowda

03/04/2025

What is UML (Unified Modeling Language)?

UML is a standardized way to visually model the structure and behavior of systems. While it is widely used in software development, UML is also useful in database design to create conceptual, logical, and physical models of your database. In the context of databases, UML diagrams, specifically class diagrams, serve as a substitute or complement to traditional ERDs (Entity-Relationship Diagrams) for representing data models.

Think of UML as a tool to represent how data is structured and related in your database. It provides a blueprint for understanding and designing tables, attributes, keys, and relationships.

Purpose of UML in Database Design

When designing a relational database, you need to:

1. **Identify the entities** (tables in a database).
2. **Define their attributes** (columns in the table).
3. **Specify the relationships** between these entities (e.g., one-to-many, many-to-many).
4. **Apply constraints** such as primary keys, foreign keys, and cardinality rules.

In UML, an entity (table) is modeled as a **class**, which corresponds to a table in the database. Each class has:

- **A name:** Represents the table name.
- **Attributes:** Represents the columns in the table.
- **Constraints:** Identifies primary keys, foreign keys, and unique keys.

Example: A `Customer` table represented as a UML class:

Customer
`customer_id {PK}`
`first_name`
`last_name`
`email {U}`
`phone_number`

Here:

- `{PK}` denotes the **primary key** (`customer_id` uniquely identifies each customer).
- `{U}` denotes a **unique constraint** (`email` must be unique for each customer).

Enhanced Project Management System

Our organization is launching a new project management tool to efficiently track and manage all aspects of our projects. The system must handle various types of information related to each project, including core project details, individual tasks, important milestones, and the team members involved. The goal is to build a scalable, maintainable solution that supports robust data integrity and easy updates.

Objectives

- The system must store key project data such as the project name, a detailed description, start and end dates, and the current status (e.g., Planned, In Progress, Completed).
- For every project, there will be multiple tasks. Each task should have a name, a detailed description, a due date, and a status (e.g., Not Started, In Progress, Completed).
- Tasks must be linked to their corresponding project so that when a project is modified or deleted, the related tasks are also properly updated or removed.
- Projects will have key milestones indicating important checkpoints or achievements. Each milestone should include a name, a detailed description, a target or achieved date, and a status (e.g., Pending, Achieved).
- Milestones must be associated with the appropriate project, ensuring that changes in project information cascade correctly to milestone data.
- The system should record information about individuals assigned to projects. This includes their names, contact email addresses, and roles (e.g., Developer, Manager, Designer).
- Each team member is linked to a specific project, and their records should update automatically if project details change.

UML Diagram:

Create a UML diagram for the Project Management System. Clearly indicate the primary keys, foreign keys, and unique constraints.

03/18/2025: Activity: Many to Many Relationship

One team member can work in multiple projects

Create a UML diagram for the Project Management System. Clearly indicate the primary keys, foreign keys, and unique constraints.