

Lesson 8: Database Design & Development

1. Database Fundamentals

Core Concepts

- **Data vs. Information:** Raw facts vs. processed, meaningful data.
- **Database:** An organized collection of structured information, or data, typically stored electronically.
- **DBMS (Database Management System):** Software used to create, manage, and interact with databases.

Database Models (Evolution)

- **Flat-file:** Simple text files. *Con: High data redundancy.*
- **Hierarchical:** Tree-like structure (one-to-many). *Con: Rigid, can't easily model many-to-many.*
- **Network:** Graph-like structure (many-to-many). *Pro: More flexible. Con: Very complex to manage.*
- **Relational:** Modern standard; data in related tables.
- **Object-Relational:** Combines Relational + Object-Oriented concepts.

2. Conceptual Design (ER Diagrams)

ERD Components (The Blueprint)

- **Entity:** A real-world object (e.g., Student, Course). *Symbol: Rectangle.*
- **Attribute:** A property of an entity (e.g., Student_Name). *Symbol: Oval.*
- **Key Attribute:** Underlined attribute that uniquely identifies an entity instance.
- **Relationship:** How entities are connected (e.g., Student enrolls in Course). *Symbol: Diamond.*

Relationship Cardinality

- **One-to-One (1:1):** e.g., A driver has one license.
- **One-to-Many (1:M):** e.g., A mother can have many children.
- **Many-to-Many (M:M):** e.g., A student can enroll in many courses, and a course can have many students.

EER (Extended ER) Diagrams

- Used for more complex designs involving concepts like **specialization** and **generalization**.

3. Logical Design & Normalization

Relational Model Keys

- **Terminology:** Relation (Table), Tuple (Row), Attribute (Column).
- **Candidate Key:** An attribute that can uniquely identify a row.
- **Primary Key (PK):** The *chosen* candidate key. Cannot be NULL.
- **Alternate Key:** A candidate key that was *not* chosen as the PK.
- **Foreign Key (FK):** A primary key from one table used in another to create a link.

Database Constraints

- **NOT NULL:** Ensures a column cannot have a NULL value.
- **UNIQUE:** Ensures all values in a column are different.
- **CHECK:** Ensures all values in a column satisfy a specific condition.

ER to Relational Mapping Rules

- **Entities:** Each entity becomes a **Table**. The entity's identifier becomes the table's Primary Key.
- **1:M Relationship:** Take the PK of the '1' side table and add it as a **Foreign Key** in the 'M' side table.
- **M:M Relationship:** Create a new table (a **Junction Table**) that contains the PKs of both original tables as Foreign Keys.

Normalization (Quality Control)

- **Purpose:** To reduce **data redundancy** and avoid **data anomalies** (errors during data modification).
 - **Insert Anomaly:** Cannot add a new record because some data is missing.
 - **Update Anomaly:** Have to update the same data in multiple places.
 - **Delete Anomaly:** Deleting one record unintentionally deletes other, unrelated information.
- **Functional Dependencies:**
 - **Partial:** A non-key attribute depends on only a *part* of a composite PK.
 - **Transitive:** A non-key attribute depends on *another non-key attribute*.
- **The Normal Forms (1NF, 2NF, 3NF):**
 - **1NF:** No repeating groups; all values are atomic.
 - **2NF:** In 1NF + No partial dependencies.
 - **3NF:** In 2NF + No transitive dependencies.

4. Essential SQL Commands

SQL Sub-languages & Commands

- **DDL (Data Definition Language):** Defines structure.
 - CREATE TABLE, ALTER TABLE, DROP TABLE
- **DML (Data Manipulation Language):** Manages data.
 - INSERT, UPDATE, DELETE, SELECT