

Introduction to Database Design

Introduction to Database Design

Key concepts, steps, and best practices for creating robust, maintainable databases.

1. What Is Database Design?

Aspect	Description
Definition	The process of organizing data elements and their relationships to support efficient storage, retrieval, and manipulation.
Goal	Produce a logical model that reflects real-world requirements and a physical implementation that performs well, scales, and maintains data integrity.
Scope	From high-level business requirements to detailed table structures, indexes, constraints, and security settings.

2. Core Concepts

Concept	Explanation	Typical Artifacts
Entity	Anything about which data is stored (e.g., <i>Customer</i> , <i>Order</i>).	Entity-Relationship (ER) diagram, table name
Attribute	A property of an entity (e.g., <i>Customer.Name</i> , <i>Order.Date</i>).	Column definition, data type, size
Relationship	Association between two or more entities (e.g., <i>Customer places Order</i>).	Foreign keys, relationship lines in ER diagram
Primary Key (PK)	Unique identifier for each row in a table.	CustomerID, OrderID
Foreign Key (FK)	Column(s) that reference a PK in another table, enforcing referential integrity.	Order.CustomerID ? Customer.CustomerID
Normalization	Decomposition of tables to eliminate redundancy and update anomalies.	Normal forms (1NF-5NF)
Denormalization	Intentional redundancy for performance (e.g., pre-aggregated totals).	Materialized views, summary tables

Introduction to Database Design

Concept	Explanation	Typical Artifacts
Integrity Constraints	Rules that keep data valid (PK, FK, UNIQUE, CHECK, NOT NULL).	Constraint definitions in DDL
Index	Data structure that speeds up search/retrieval.	B-tree, hash, composite indexes
Transaction	Unit of work that must be atomic, consistent, isolated, durable (ACID).	BEGIN ... COMMIT blocks

3. Design Process (Step-by-Step)

1. Gather Requirements

- Interview stakeholders.
- Identify business processes, reports, and data entry forms.
- Produce a *requirements specification* (entities, attributes, rules).

2. Conceptual Modeling

- Create an **Entity-Relationship Diagram (ERD)** or **UML class diagram**.
- Define entities, attributes, primary keys, and relationships (1:1, 1:M, M:N).

3. Logical Modeling

- Translate the ERD into relational tables.
- Apply **normalization** (usually up to 3NF or BCNF).
- Define PKs, FKs, and other constraints.
- Document data types and column sizes.

4. Physical Modeling

- Choose a DBMS (e.g., PostgreSQL, MySQL, Oracle, SQL Server).
- Refine data types to match the platform (e.g., VARCHAR, NUMERIC).
- Add **indexes**, **partitioning**, **tablespaces**, and **storage parameters**.
- Plan for **backup**, **recovery**, and **security** (roles, privileges).

5. Implementation

Introduction to Database Design

- Write DDL scripts (CREATE TABLE, CREATE INDEX, ALTER TABLE).
- Populate reference data (lookup tables).
- Set up constraints, triggers, and stored procedures as needed.

6. Testing & Validation

- Load sample data and run typical queries.