Database Systems Textbook

Introduction to Databases

A database is an organized collection of structured information or data, typically stored electronically in a computer system. A database management system (DBMS) is software that interacts with end users, applications, and the database itself to capture and analyze data. Databases are essential for storing, retrieving, and managing large amounts of information efficiently and securely.

Relational Databases

Relational databases organize data into tables (relations) consisting of rows and columns. Each row represents a record, and each column represents an attribute of that record. Tables can be related to each other through foreign keys, enabling complex queries across multiple tables. Structured Query Language (SQL) is used to create, read, update, and delete data. Relational databases ensure data integrity through ACID properties: Atomicity, Consistency, Isolation, and Durability.

SQL is the standard language for relational database management. It includes Data Definition Language (DDL) commands for creating and modifying database structures, Data Manipulation Language (DML) commands for querying and modifying data, and Data Control Language (DCL) commands for managing access permissions. Popular relational databases include MySQL, PostgreSQL, Oracle Database, Microsoft SQL Server, and SQLite.

NoSQL Databases

NoSQL databases provide flexible schema designs and are optimized for specific data models and access patterns. They typically sacrifice some ACID guarantees for improved scalability and performance. Document databases like MongoDB store data in JSON-like documents. Key-value stores like Redis provide simple key-value pair storage with extremely fast access. Column-family stores like Cassandra organize data by columns rather than rows. Graph databases like Neo4j are optimized for storing and querying relationships between entities.

Database Design Principles

Effective database design involves normalization to reduce redundancy and improve data integrity. The process includes identifying entities, defining relationships, choosing appropriate data types, and creating indexes for query optimization. Normalization forms (1NF, 2NF, 3NF, BCNF) provide guidelines for organizing data efficiently. However, denormalization may be appropriate in some cases to improve read performance at the cost of some redundancy.