

Introduction to Database Systems

Introduction to Database Systems

Compact, easy-to-read study notes

1. What Is a Database?

Term	Definition
Database	An organized collection of related data stored electronically for fast retrieval, update, and management.
Database Management System (DBMS)	Software that creates, maintains, and controls access to databases. It handles storage, security, concurrency, and recovery.
Data vs. Information	<i>Data</i> = raw facts (e.g., numbers, text). <i>Information</i> = processed, meaningful data (e.g., a sales report).

2. Why Use a DBMS?

Need	How a DBMS Helps
Data Redundancy	Eliminates duplicate copies ? saves space, reduces inconsistencies.
Data Integrity	Enforces rules (primary keys, foreign keys, constraints).
Concurrent Access	Multiple users can read/write simultaneously with locking & transaction control.
Security	Fine-grained permissions, authentication, encryption.
Backup & Recovery	Automated snapshots, logs, point-in-time restore.
Query Capability	Powerful, declarative language (SQL) for ad-hoc queries.

3. Data Models – The Foundations

Model	Core Idea	Typical Use
Hierarchical	Tree-like parent-child relationships.	Legacy mainframes, XML.
Network	Graph with many-to-many links (records can have multiple parents).	Complex engineering data.

Introduction to Database Systems

Model	Core Idea	Typical Use
Relational	Data in tables (relations) with rows (tuples) and columns (attributes).	Most commercial/academic systems.
Object?Oriented	Objects with attributes & methods; supports inheritance.	CAD, multimedia, complex simulations.
Document / NoSQL	Semi?structured documents (JSON, BSON).	Web apps, big data, rapid prototyping.
Key?Value / Column ?Family	Simple key ? value mapping; column families store sparse data.	Caching, time?series, analytics.

The relational model dominates because of its mathematical foundation (set theory, relational algebra) and mature tooling.

4. Core Relational Concepts

Concept	Description	Example
Table (Relation)	Two?dimensional set of tuples.	Customers table.
Tuple (Row)	Single record in a table.	(101, 'Alice', 'NY').
Attribute (Column)	Named data field.	CustomerID, Name.
Primary Key	Unique identifier for each tuple.	CustomerID.
Foreign Key	Reference to a primary key in another table; enforces referential integrity.	Orders.CustomerID ? Customers.CustomerID.
Normalization	Decomposition of tables to eliminate redundancy & update anomalies.	1NF, 2NF, 3NF, BCNF.
SQL	Structured Query Language – the standard declarative language for relational DBMSs.	SELECT * FROM Customers WHERE State='NY';

5. The ACID Properties (Transaction Management)

Property	Meaning
----------	---------

Introduction to Database Systems

Atomicity A transaction is “all?or?nothing”. If any part fails, the whole transaction rolls back.

Consistency After commit, the database must satisfy all integrity constraints.

Isolation Concurrent transactions appear as if executed