

# Database Management System(DBMS)

## Unit 1: Introduction to DBMS

- **Syllabus Topics:** Data, database, DBMS, characteristics, users, architecture, data abstraction, independence.
- **Detailed Explanations:**
  - **Data Independence:** This is the ability to modify the schema at one level without affecting the schema at the next higher level.
    - *Logical Data Independence:* Changing the logical schema (e.g., adding a new column to a table) without changing the external view or application programs.
    - *Physical Data Independence:* Changing the physical storage (e.g., switching from hard drive to SSD or changing hashing algorithms) without affecting the logical schema.
  - **Three-Schema Architecture:**
    - *Internal Level:* How data is physically stored (bytes, blocks).
    - *Conceptual (Logical) Level:* Describes *what* data is stored (tables, relationships).
    - *External (View) Level:* How the user sees the data (forms, reports).
- **Video Resource:** [Data Independence in DBMS \(Hindi\)](#)

## Unit 2: Entity–Relationship (ER) Modeling

- **Syllabus Topics:** Entities, attributes, relationships, ER diagrams, keys, mapping, generalization, specialization, aggregation.
- **Detailed Explanations:**
  - **Generalization vs. Specialization:**
    - *Generalization (Bottom-Up):* Combining lower-level entities with common attributes into a higher-level entity (e.g., "Car" and "Truck"  $\rightarrow$  "Vehicle").
    - *Specialization (Top-Down):* Breaking a higher-level entity into specific lower-level entities based on distinct characteristics (e.g., "Person"  $\rightarrow$  "Student" and "Teacher").
  - **Aggregation:** Used when a relationship itself needs to be treated as an entity to participate in another relationship. It solves the limitation of binary relationships.
  - **Keys:**
    - *Candidate Key:* A minimal set of attributes that uniquely identifies a record.
    - *Primary Key:* The specific candidate key chosen to identify records.
- **Video Resource:** [Generalization, Specialization & Aggregation \(Hindi\)](#)

## Unit 3: Relational Model

- **Syllabus Topics:** Relational algebra, relational calculus, integrity constraints, schemas.
- **Detailed Explanations:**
  - **Relational Algebra (Procedural):** A set of operations used to manipulate relations.
    - *Selection ( $\sigma$ ):* Filters rows based on a condition (Horizontal slicing).
    - *Projection ( $\pi$ ):* Filters columns (Vertical slicing).
    - *Cartesian Product ( $\times$ ):* Combines every row of one table with every row of another.
  - **Integrity Constraints:** Rules to maintain data quality.
    - *Entity Integrity:* Primary key cannot be NULL.
    - *Referential Integrity:* Foreign key must match a Primary Key in the parent table or be NULL.
- **Video Resource:** [Relational Algebra Complete Explanation \(Hindi\)](#)

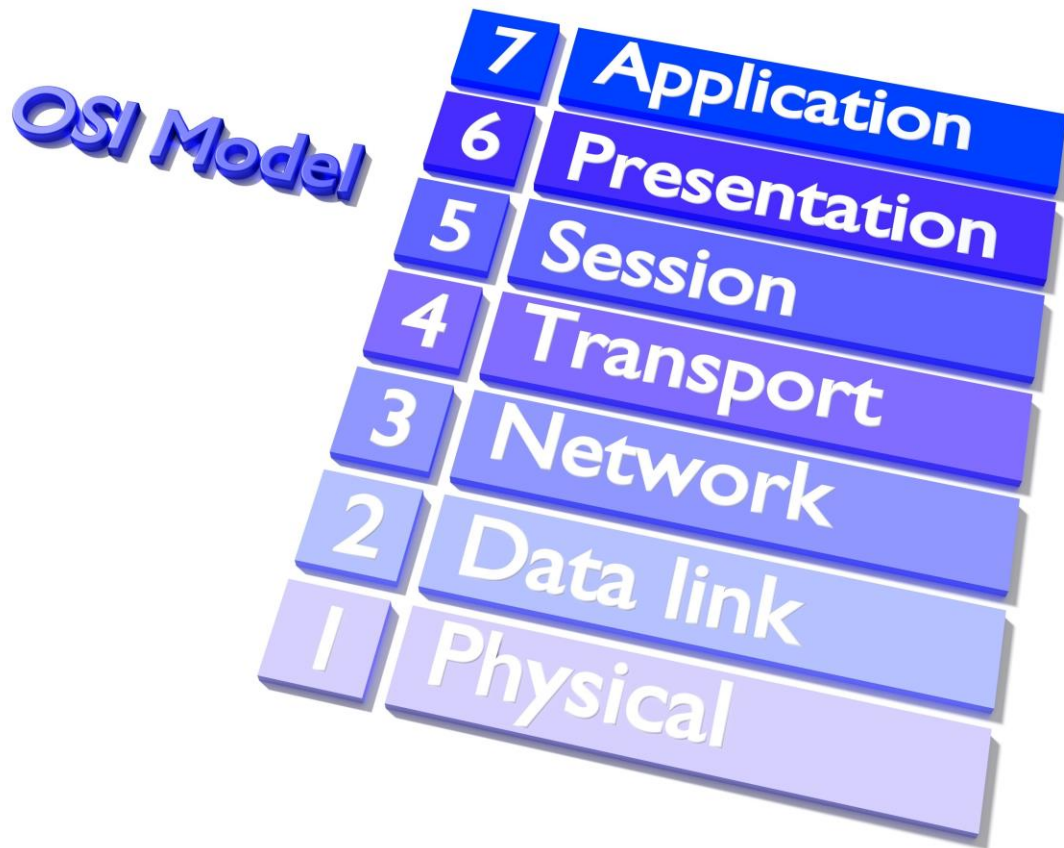
## Unit 4: SQL & Query Processing

- **Syllabus Topics:** DDL, DML, joins, nested queries, indexes, views, triggers.
- **Detailed Explanations:**
  - **Joins:**
    - *Inner Join:* Returns only matching rows from both tables.
    - *Left Outer Join:* Returns all rows from the left table and matching rows from the right (NULL if no match).
    - *Full Outer Join:* Returns all rows when there is a match in either left or right table.
  - **Triggers:** Automatic procedures that execute in response to certain events (INSERT, UPDATE, DELETE) on a table. Used for audit trails or enforcing complex business rules.
  - **Stored Procedures:** Pre-compiled collections of SQL statements stored in the database for reuse and better performance.
- **Video Resource:** [SQL Joins Explained with Examples \(Hindi\)](#)

## Unit 5: Normalization & Database Design

- **Syllabus Topics:** Functional dependencies, 1NF, 2NF, 3NF, BCNF, decomposition.
- **Detailed Explanations:**
  - **Functional Dependency (FD):** A relationship where attribute  $A$  determines attribute  $B$  ( $A \rightarrow B$ ).
  - **Normal Forms:**
    - *1NF:* Eliminate repeating groups; ensure atomicity.

- **2NF:** Eliminate **Partial Dependency** (All non-key attributes must depend on the *full* primary key).
- **3NF:** Eliminate **Transitive Dependency** (Non-key attributes should not depend on other non-key attributes).
- **BCNF:** A stricter version of 3NF where for every dependency  $X \rightarrow Y$ ,  $X$  must be a Super Key.



Shutterstock

- **Video Resource:** [Normalization 1NF, 2NF, 3NF, BCNF \(Hindi\)](#)

## Unit 6: Transaction Management

- **Syllabus Topics:** ACID properties, schedules, locks, 2-phase locking, serializability.
- **Detailed Explanations:**
  - **ACID Properties:**
    - **Atomicity:** All operations of a transaction happen, or none do (All-or-nothing).
    - **Consistency:** Database moves from one valid state to another.
    - **Isolation:** Transactions occur independently without interference.
    - **Durability:** Committed changes are permanent, even after a system crash.

- **Serializability:** Ensuring that the outcome of concurrent transactions is the same as if they were executed one after another (serially).
- **2-Phase Locking (2PL):** A protocol to ensure serializability by having a "Growing Phase" (acquiring locks) and a "Shrinking Phase" (releasing locks).
- **Video Resource:** [ACID Properties in DBMS \(Hindi\)](#)

## Unit 7: File Organization & Emerging Technologies

- **Syllabus Topics:** Indexing (B/B+ trees), hashing, NoSQL, distributed DBs.
- **Detailed Explanations:**
  - **B+ Trees:** An advanced self-balancing tree used for indexing. Unlike B-Trees, B+ Trees store data pointers *only* at the leaf nodes, while internal nodes only store keys for navigation. This allows for faster range queries and efficient disk usage.
  - **Hashing:** Mapping a search key directly to a disk address using a hash function.
    - *Static Hashing:* Fixed number of buckets.
    - *Dynamic Hashing:* Buckets grow or shrink as needed (Extendible Hashing).
- **Video Resource:** [B Tree and B+ Tree Indexing \(Hindi\)](#)