Revision: Data & Database Concepts

- Key Concepts and Models
- Optional: Name / Date

What is Data?

- Raw facts, figures, or observations
- Can be numbers, text, images, or audio
- Example: Temperature readings, student names

Types of Data

- Structured: Tables, spreadsheets, Databases
- Unstructured: Emails, videos, social media posts
- Semi-structured: Flexible schema (JSON, XML, NoSQL)

Importance of Data

- Supports decision-making
- Basis for analytics and reporting
- Enables predictive modeling

What is a Database?

- A database is an organized collection of data that is stored electronically and managed in a way that makes it easy to **store, retrieve, update, and share** information. Instead of keeping data scattered in files or spreadsheets, a database brings everything together in a structured way.
- Supports querying, updating, and reporting
- Example: Library catalog, banking system

Characteristics of a Database

- Persistent storage
- Data integrity
- Accessibility and sharing
- Security

Types of Databases

- Relational (SQL)
- NoSQL (Document, Key-Value, Graph)
- Distributed / Cloud databases

Real-life Applications

- Banking: Customer accounts, transactions
- E-commerce: Products, orders, customers
- Healthcare: Patient records, treatment data

What is DBMS?

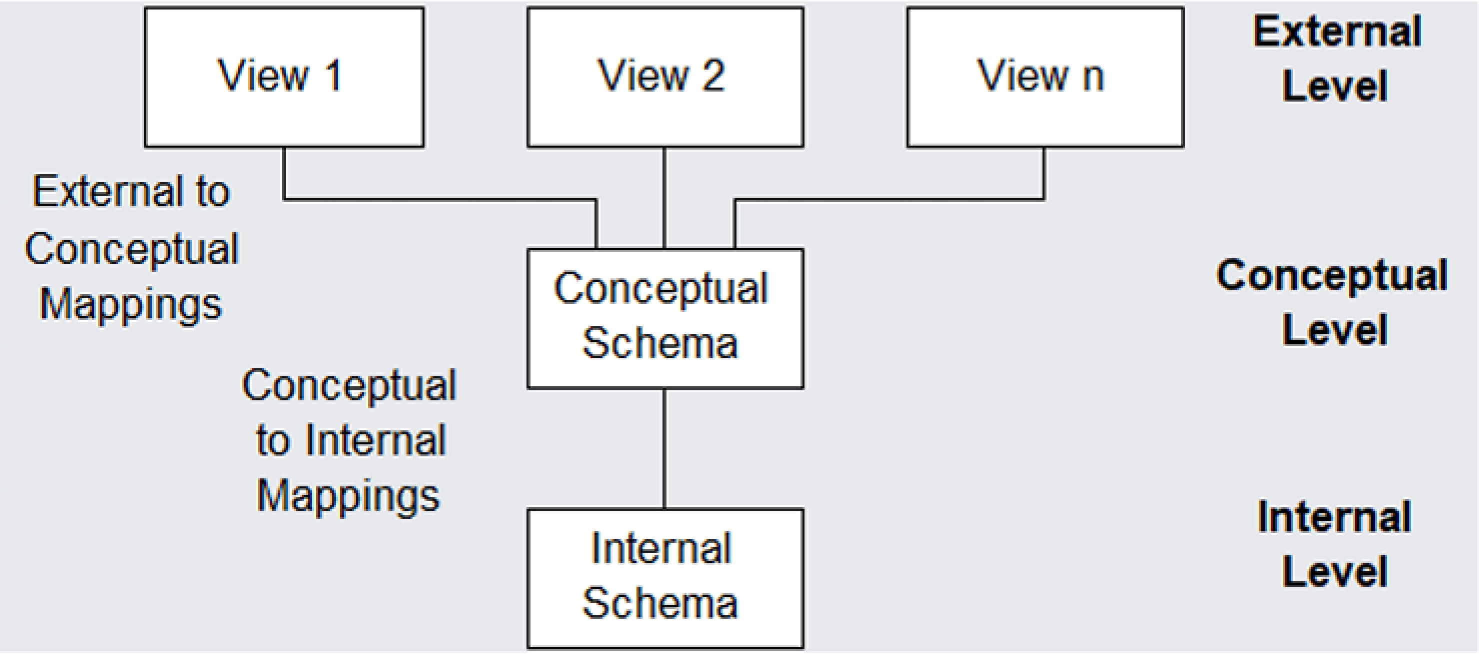
- Software that interacts with databases and users
- Provides data management, security, and consistency

Functions of DBMS

- Store, retrieve, update data
- Security and access control
- Backup and recovery
- Maintain data integrity

DBMS Architecture

- 3-level Architecture



Advantages of DBMS

- Efficient data management
- Multiple user access
- Reduced redundancy
- Data security

Examples of DBMS

- MySQL
- PostgreSQL
- Oracle
- MongoDB

DBMS vs File System

- Redundancy: Low vs High
- Security: High vs Low
- Data Integrity: Enforced vs Manual

Components of DBMS

- DB Engine
- Query Processor
- Metadata
- Transaction Management

Relational Database Concept

- Stores data in tables (relations)
- Supports SQL queries
- Ensures data integrity

Tables, Rows, and Columns

- Table = Relation
- Row = Tuple
- Column = Attribute
- Example Table: Student(ID, Name, Age)

Keys

- Primary Key: Unique identifier
- Foreign Key: Links tables
- Candidate Key: Potential unique identifiers

Relationships in Relational DB

- One-to-One
- One-to-Many
- Many-to-Many

Functions of Relational Database

- Store and retrieve data efficiently
- Maintain consistency
- Enforce constraints

Constraints

- Entity Constraint: Primary key must be unique
- Referential Constraint: Foreign key must exist in parent table
- Domain Constraint: Attribute must have valid values

Indexing and Views

- Index: Speeds up queries
- View: Logical subset of data for user
- Example SQL: `CREATE VIEW ActiveCustomers AS SELECT Name, Email FROM Customers WHERE Status = 'Active';`

What is Data View?

- Logical representation of data for users
- Abstracts physical storage

Types of Data Views

- Logical View: How data appears to users
- Physical View: How data is stored

Creating Views

- SQL Example: CREATE VIEW ActiveOrders AS SELECT * FROM Orders WHERE Status='Active';

Advantages of Views

- Simplifies queries
- Provides security
- Customizes user access

Data Architecture Definition

- Structure, storage, flow, and management of data

Layers of Data Architecture

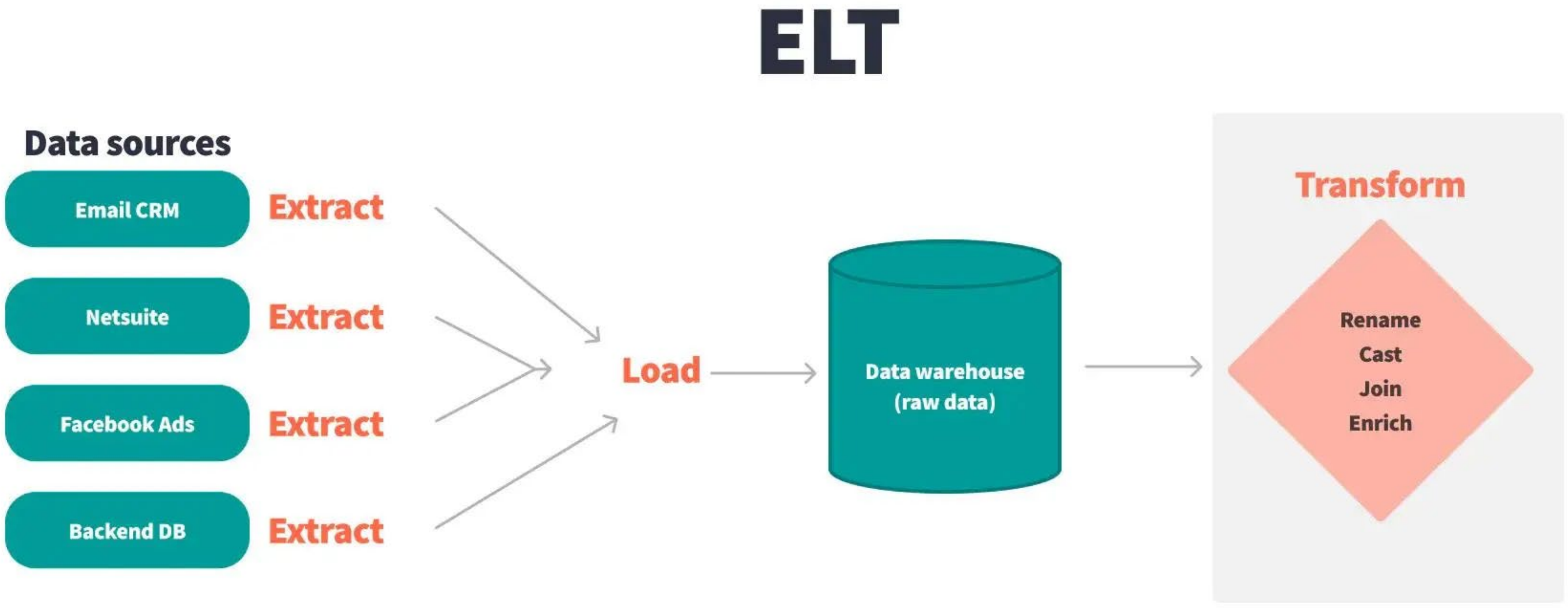
- 1. Data Sources
- 2. Storage (Databases, Warehouses)
- 3. Processing (ETL)
- 4. Presentation (Reports, Dashboards)

Data Warehouse

- Central repository for analytics
- Optimized for query performance

ETL Process

- Extract → Transform → Load



Data Flow Example

- Source Systems → Data Warehouse → Analytics → Dashboard

Importance of Good Architecture

- Efficiency
- Data consistency
- Scalability

ER Model Definition

- Visual representation of entities and their relationships

Components of ER Model

- Entity: Object (Student)
- Attribute: Property (Name, ID)
- Relationship: Connects entities

Types of Attributes

- Simple, Composite, Derived, Multi-valued

Types of Relationships

- 1:1, 1:N, N:M
- Example: Student enrolls in Course

ER Diagram Notation

- Rectangle = Entity
- Oval = Attribute
- Diamond = Relationship

What is Normalization?

- Organizing data to reduce redundancy and anomalies

Problems in Unnormalized Data

- Redundancy
- Update anomalies
- Deletion anomalies

First Normal Form (1NF)

- No repeating groups
- Atomic values

Second Normal Form (2NF)

- Remove partial dependencies

Third Normal Form (3NF)

- Remove transitive dependencies

Advantages of Normalization

- Data integrity
- Efficient storage
- Avoid redundancy