What is Data?

Data is raw facts and figures that represent information. It can be numbers, text, images, or any form of input that computers process.

- **Types**: Structured (organized in tables, like spreadsheets) vs. Unstructured (free-form, like emails or videos).
- **Example**: my name "Raju" is text data; my age "25" is categorical data.

Data becomes valuable when organized and analyzed, leading us to databases.

What is a Database?

A database is a structured collection of data stored electronically for easy access, management, and updating. It's like a digital filing cabinet.

- **Purpose**: Stores data efficiently to avoid redundancy and ensure quick retrieval.
- **Example**: A phonebook app uses a database to store contacts.

Databases handle vast amounts of data, from personal apps to enterprise systems like those used by Indian banks (e.g., HDFC or SBI).

What is a Database Management System (DBMS)?

A DBMS is software that interacts with users, applications, and the database to capture, analyze, and manage data. It acts as a bridge.

- **Functions**: Create, read, update, delete (CRUD) operations; security; backups.
- **Types**: Hierarchical (tree-like), Network (graph-like), but most modern are Relational or Non-Relational.

DBMS ensures data integrity by preventing duplicate entries in a customer list.
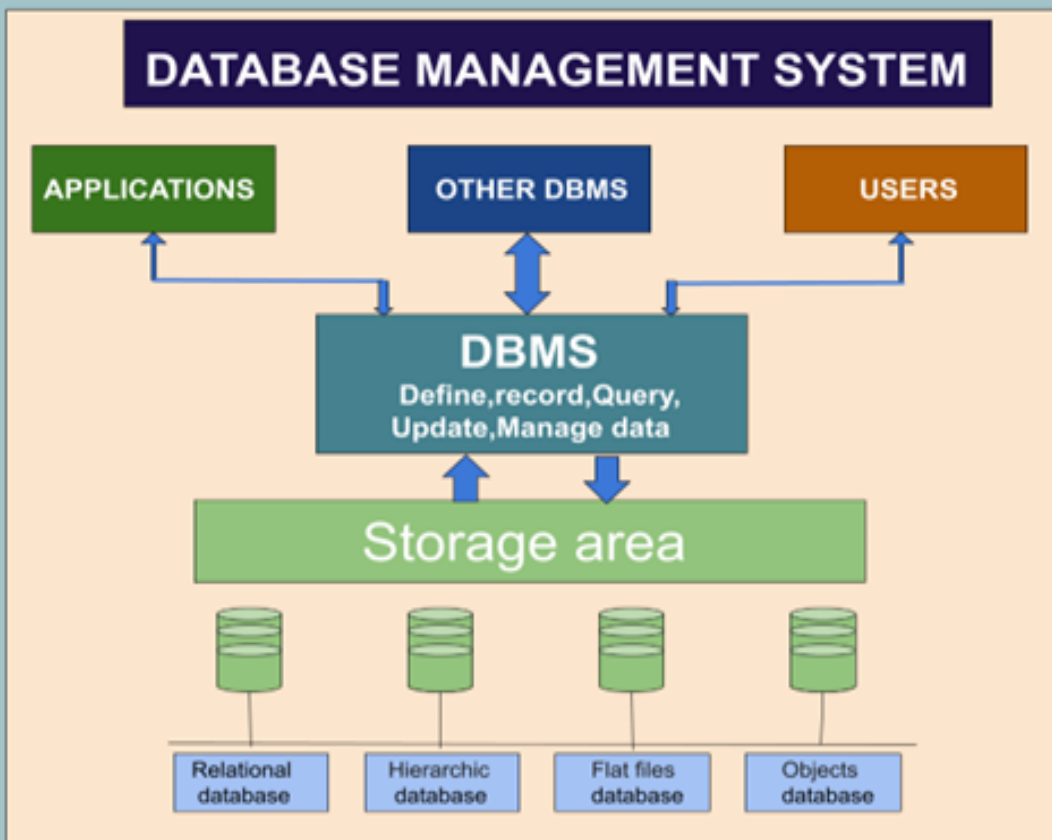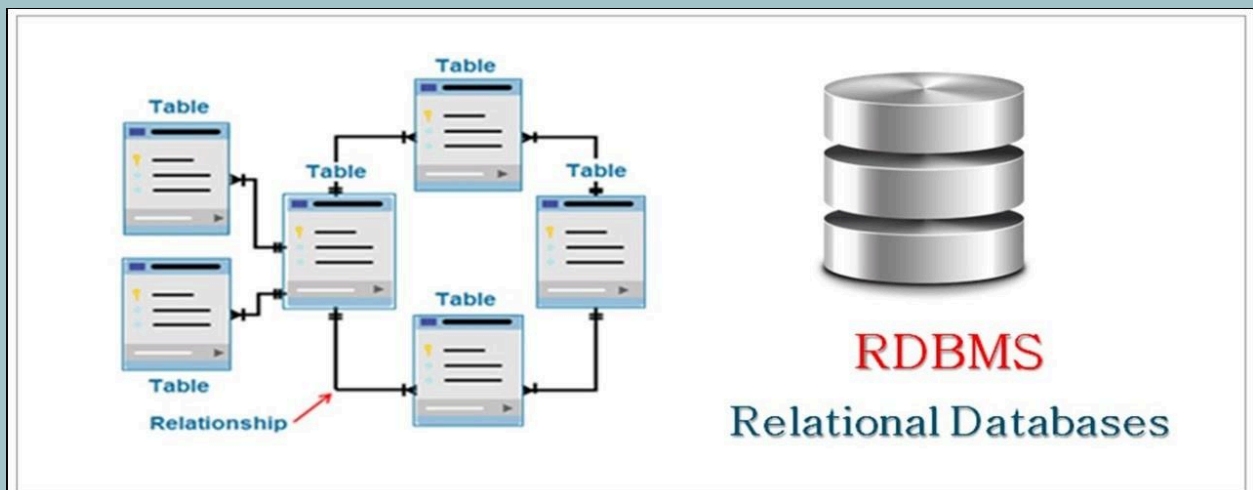
Relational DBMS (RDBMS)

RDBMS organizes data into tables with rows (records) and columns (attributes), linked by relationships (e.g., primary/foreign keys).

- **Key Features**: Uses SQL for queries; enforces ACID properties (Atomicity, Consistency, Isolation, Durability) for reliable transactions.
- **Examples**: Based on 2026 data from DB-Engines and surveys, popular RDBMS include:
    - Oracle: Leads for enterprises, used in banking and ERP.
    - PostgreSQL: Open-source, top for developers; great for web apps.
    - MySQL: Widely used for web (e.g., e-commerce); but its popularity dipped in 2026 due to alternatives.
    - Microsoft SQL Server: Strong in Windows ecosystems.
    - Others: MariaDB, SQLite (for mobile/lightweight).

RDBMS shines in scenarios needing structured data, like financial records.

Non-Relational DBMS (Non-RDBMS or NoSQL)

NoSQL databases handle unstructured or semi-structured data, scaling horizontally for big data.

- **Key Features**: Flexible schemas; types include Document (JSON-like), Key-Value, Column-Family, Graph.
- **Examples**: From 2026 rankings:
    - MongoDB: Document-based, top NoSQL; used by Uber/Lyft for real-time data.
    - Apache Cassandra: Wide-column, scalable for distributed systems.
    - Redis: Key-value, fast for caching.
    - Amazon DynamoDB: Cloud-native, serverless.
    - Others: Neo4j (Graph), Elasticsearch (Search).

NoSQL is ideal for apps like social media (e.g., Twitter/X) handling massive, varied data.

Differences between Relational-DBMS and Non-Relational DBMS
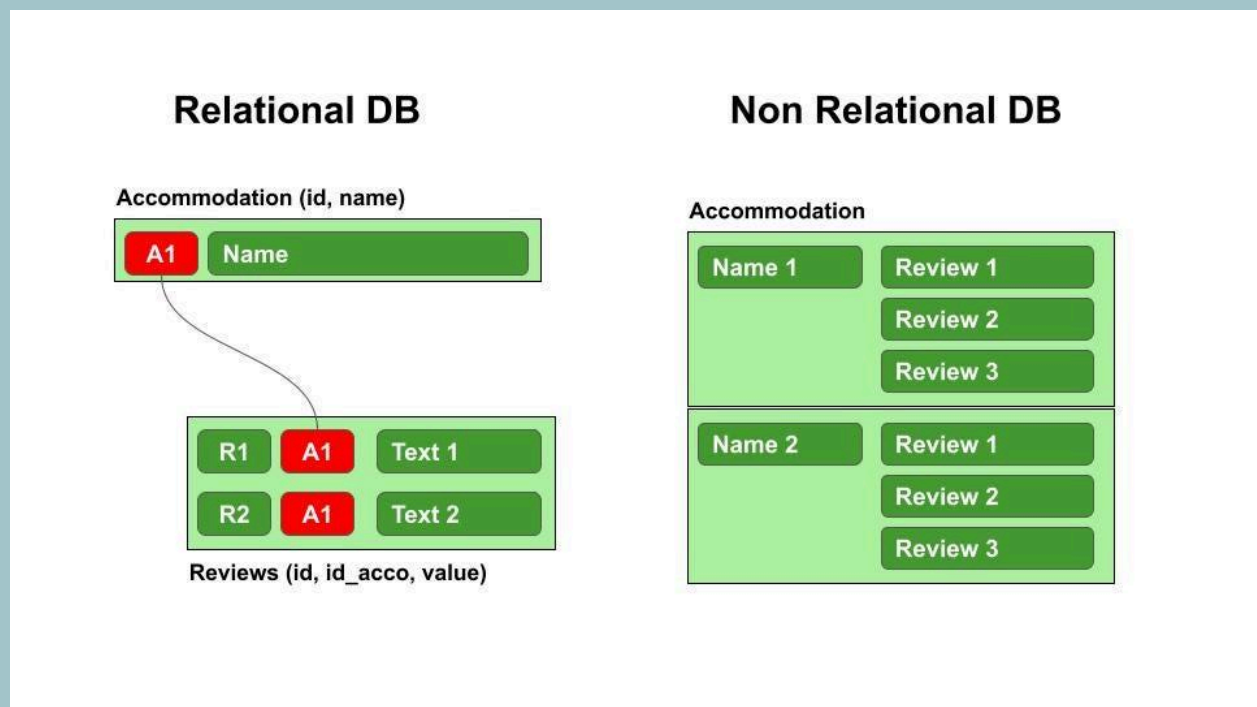
Aspect	RDBMS (SQL)	Non-RDBMS (NoSQL)
Data Structure	Tables with fixed schema	Flexible (documents, key-value, etc.)
Query Language	SQL (standardized)	Varies (e.g., MongoDB Query Language)
Scalability	Vertical (bigger servers)	Horizontal (add nodes)
ACID Compliance	Strong (transactions)	Often BASE (eventual consistency)
Use Cases	Structured data, finance	Big data, real-time apps
Examples	Oracle, PostgreSQL	MongoDB, Cassandra

Introduction to SQL

SQL is a standard language for managing and manipulating relational databases. Pronounced "sequel" or "S-Q-L."

- **Components**:
    - DDL (Data Definition Language): CREATE, ALTER, DROP tables.
    - DML (Data Manipulation Language): INSERT, UPDATE, DELETE, SELECT.
    - DCL (Data Control Language): GRANT, REVOKE permissions.
- **Why SQL?**: Human-readable, standardized (ANSI/ISO since 1986).

SQL queries data like: SELECT * FROM users WHERE age > 18;

History of SQL and Databases

- **Past**: Databases emerged in the 1960s (e.g., hierarchical like IMS). In 1970, Edgar F. Codd proposed the relational model. IBM's Donald Chamberlin and Raymond Boyce created SEQUEL (later SQL) in 1974 for System R. Oracle released the first commercial SQL RDBMS in 1979. NoSQL arose in the 2000s for web-scale data (e.g., Google BigTable inspired Cassandra).
- **Present**: SQL dominates structured data; NoSQL for big/unstructured. Tools like PostgreSQL and MongoDB are prevalent in 2026. In India, companies like Infosys use Oracle for enterprise, while startups prefer PostgreSQL/MongoDB.
- **Future**: SQL evolves with AI integration (e.g., vector search in PostgreSQL). Databases become AI-native, unified (handling SQL/NoSQL), and serverless. By 2030, expect "agent-ready" data with governance focus.

Uses of SQL

- Querying data (e.g., reports).
- Managing transactions (e.g., banking withdrawals).
- Data analysis (e.g., business intelligence).
- Web/apps (e.g., user logins). In Hyderabad's tech hubs, SQL powers e-commerce like Flipkart or analytics in startups.

How SQL Works

1. **Connect**: Application connects to DBMS.
2. **Parse Query**: DBMS checks syntax.
3. **Optimize**: Chooses an efficient execution plan.
4. **Execute**: Retrieves/manipulates data.
5. **Return Results**: Sends back to user/app.

.