Database Management Systems
Types of Databases

Vikas Thammanna Gowda

01/17/2025

1 Types of Databases Based on Data Model

1.1 Structured Data

- **Definition:** Structured data follows a pre-defined schema or format, such as rows and columns in a table. Each column corresponds to a specific attribute, and each row represents a record.
- **Example:** Relational databases (RDBMS) like Oracle or MySQL store structured data. Tables consist of keys (primary, foreign) to enforce relationships between entities.
- **Usage:** Applications where data must conform to rules for accurate computations, like payroll systems or banking applications.
- **Advantages:**
  - Simple to use and query using SQL.
  - Maintains strong consistency and data integrity.
- **Limitations:** Poor performance for unstructured or semi-structured data, requiring significant effort to adapt.

1.2 Semi-Structured Data

- **Definition:** Semi-structured data does not adhere to a strict schema but has some level of organization, often represented in formats like XML or JSON. Attributes can vary across records.
- **Example:** JSON data from APIs or CSV files with optional columns.
- **Usage:** Common in web-based applications or data integration scenarios where the data format evolves over time.
- **Advantages:**
  - Flexible and adaptable for rapidly changing data needs.
  - Easier to process than unstructured data.
- **Limitations:** More challenging to query and analyze compared to structured data.

1.3 Unstructured Data

- **Definition:** Unstructured data lacks any inherent structure or format, making it harder to process with traditional database tools. Examples include multimedia files, logs, and documents.
- **Example:** Images, audio, video, sensor logs, and emails.
- **Usage:** Suitable for big data platforms and data lakes, where raw data is stored before processing.
- **Advantages:**
  - Can accommodate any data type.
  - Essential for domains like AI/ML, which rely on vast amounts of unstructured data.
- **Limitations:** Requires advanced tools for storage, processing, and analysis, such as NoSQL databases or Hadoop.

2 Types of Databases Based on Data Organization

2.1 Relational Databases (RDBMS)

- **Description:** Organize data into tables with predefined relationships between them using primary and foreign keys. Supports ACID properties (Atomicity, Consistency, Isolation, Durability).
- **Examples:** Oracle, MySQL, SQL Server.
- **Usage:** Transaction-heavy systems like banking, ERP, and CRM systems.
- **Advantages:**
  - Strong consistency and reliable data integrity.
  - Powerful query capabilities via SQL.
- **Limitations:** Performance issues with very large datasets or distributed systems.

2.2 Non-Relational Databases (NoSQL)

- **Description:** Designed to handle diverse data models (key-value, document, graph, column-family) for unstructured or semi-structured data.
- **Examples:** MongoDB (document-oriented), DynamoDB (key-value store).
- **Usage:** Real-time analytics, IoT, social media applications.
- **Advantages:**
  - Highly scalable and flexible.
  - Supports distributed systems.
- **Limitations:** Lacks standard query language like SQL.

2.3 Hierarchical Databases

- **Description:** Data is organized in a tree-like structure with parent-child relationships. Each child node has a single parent.
- **Example:** IBM Information Management System (IMS).
- **Usage:** Early databases for file systems, simple ERP implementations.
- **Advantages:**

  - Fast access due to predefined hierarchy.
  - Simple to understand for hierarchical data.

- **Limitations:** Inflexible when relationships change or expand.

2.4 Network Databases

- **Description:** Extends hierarchical models to allow multiple parent-child relationships, represented as a graph.

- **Example:** Integrated Data Store (IDS).

- **Usage:** Complex scenarios such as network management or supply chain systems.

- **Advantages:**

  - Supports complex relationships.
  - High performance for many-to-many relationships.

- **Limitations:** More challenging to implement and maintain.

3 Types of Databases Based on Data Storage Architecture

3.1 Distributed Databases

- **Description:** Data is stored across multiple physical locations. Communication between locations happens via networks.

- **Types:**

  - Homogeneous: All locations use the same DBMS.
  - Heterogeneous: Different DBMSs across locations.

- **Examples:** Google Spanner, Apache Cassandra.

- **Usage:** Systems requiring high availability and disaster recovery, like global financial services.

- **Advantages:**

  - High fault tolerance.
  - Scalability for growing data needs.

- **Limitations:** Increased complexity in coordination and consistency.

3.2 Cloud Databases

- **Description:** Hosted and managed in the cloud, offering scalability and accessibility.

- **Examples:** Amazon RDS, Google BigQuery, Azure SQL Database.

- **Usage:** Applications with fluctuating workloads, such as e-commerce platforms or SaaS.

- **Advantages:**

  - Cost-effective (pay-as-you-go model).
  - Easy to scale.

- **Limitations:** Dependency on internet connectivity and cloud provider reliability.

3.3 In-Memory Databases

- **Description:** Store data in volatile memory instead of disks for faster access.
- **Examples:** Redis, Memcached.
- **Usage:** Real-time analytics, leaderboards, and caching.
- **Advantages:**
  - High-speed read/write operations.
  - Low latency for mission-critical applications.
- **Limitations:** Limited data size (dependent on memory) and potential data loss during power outages.

4 Types of Databases Based on Use Cases

4.1 Operational Databases

- **Description:** Focused on real-time, transactional workloads.
- **Examples:** MySQL, PostgreSQL.
- **Usage:** E-commerce, banking systems.
- **Advantages:** Optimized for high write throughput.
- **Limitations:** Poor for analytical queries.

4.2 Analytical Databases

- **Description:** Optimized for analytics, reporting, and decision-making.
- **Examples:** Snowflake, Teradata.
- **Usage:** Data warehouses and business intelligence.
- **Advantages:** Fast query performance for read-heavy workloads.
- **Limitations:** Not suitable for frequent updates.

4.3 Mobile Databases

- **Description:** Designed for mobile devices with offline synchronization.
- **Examples:** SQLite, Realm.
- **Usage:** Mobile apps like messaging and offline-first apps.
- **Advantages:** Supports offline functionality.
- **Limitations:** Resource constraints of mobile devices.

5 Multiple Choice Questions

1. What is a key characteristic of structured data?
   - (a) Follows no predefined schema
   - (b) Organized as rows and columns in a table
   - (c) Stored in binary formats
   - (d) Completely lacks consistency

2. Which of the following is an example of structured data?
   - (a) JSON from APIs
   - (b) Tables in Oracle Database
   - (c) Sensor logs
   - (d) Video files

3. What is a major limitation of structured data systems?
   - (a) Inflexibility in querying
   - (b) Poor performance with unstructured data
   - (c) Difficulty in creating relationships between entities
   - (d) Inability to use SQL

4. Which type of data often uses XML or JSON formats?
   - (a) Structured
   - (b) Semi-structured
   - (c) Unstructured
   - (d) Binary

5. Which domain relies heavily on unstructured data?
   - (a) E-commerce systems
   - (b) AI/ML systems
   - (c) Relational databases
   - (d) ERP systems

6. Which property is characteristic of relational databases?
   - (a) Flexible schema
   - (b) Stores key-value pairs
   - (c) Enforces ACID properties
   - (d) Lacks query language support

7. Which of these is NOT an example of a relational database?
   - (a) Oracle
   - (b) MySQL
   - (c) MongoDB
   - (d) SQL Server

8. What kind of database is ideal for real-time analytics and IoT applications?
   (a) Relational
   (b) Non-relational
   (c) Hierarchical
   (d) Network

9. Which database model organizes data in a tree-like structure?
   (a) Key-value
   (b) Network
   (c) Hierarchical
   (d) Columnar

10. In a network database, how is data organized?
   (a) As rows and columns
   (b) As key-value pairs
   (c) In a graph structure
   (d) As binary objects

11. What distinguishes distributed databases?
   (a) Data is stored in a single location
   (b) Data is stored across multiple physical locations
   (c) Data is stored in memory
   (d) Data is inaccessible offline

12. What is a key feature of homogeneous distributed databases?
   (a) Different DBMS at all locations
   (b) Same DBMS at all locations
   (c) Data stored only in memory
   (d) Use of key-value data models

13. Which of the following is an example of a distributed database?
   (a) Oracle
   (b) Google Spanner
   (c) SQLite
   (d) Realm

14. What is the primary advantage of cloud databases?
   (a) Low scalability
   (b) Requires on-premise infrastructure
   (c) Cost-effective and scalable
   (d) Cannot handle fluctuating workloads

15. Which type of database stores data in volatile memory?

    (a) Distributed
    (b) In-memory
    (c) Hierarchical
    (d) Analytical

16. Which database is best suited for real-time, transactional workloads?

    (a) Analytical
    (b) Operational
    (c) Columnar
    (d) Mobile

17. Which of the following is an example of an operational database?

    (a) Snowflake
    (b) Teradata
    (c) MySQL
    (d) Apache HBase

18. What is the main limitation of operational databases?

    (a) Inefficiency in handling transactional workloads
    (b) Poor for analytical queries
    (c) Lack of schema enforcement
    (d) Inability to handle real-time updates

19. What type of database is optimized for analytics and reporting?

    (a) Operational
    (b) Analytical
    (c) Mobile
    (d) Network

20. Which database is designed for mobile devices with offline synchronization capabilities?

    (a) SQLite
    (b) Redis
    (c) Snowflake
    (d) Teradata

21. What is the primary advantage of mobile databases?

    (a) Supports large-scale analytics
    (b) Ensures offline functionality
    (c) Highly scalable for distributed systems
    (d) Uses a predefined schema

22. Which database stores data as objects, aligning with object-oriented programming principles?

    (a) Columnar
    (b) Network
    (c) Object-oriented
    (d) Non-relational

23. Which of the following is an example of a columnar database?

    (a) Apache HBase
    (b) MongoDB
    (c) Realm
    (d) DynamoDB

24. What is the key limitation of columnar databases?

    (a) Inefficient for transactional workloads
    (b) Poor query performance for analytics
    (c) Cannot handle unstructured data
    (d) Lack of scalability

25. Which type of database combines data storage flexibility with hierarchical relationships?

    (a) Hierarchical
    (b) Network
    (c) Key-value
    (d) Relational