Lecture 4: Introduction to Databases

CSS 200 - Intro to Information Systems

Module 1

- What is an information system?
- Where do we use information systems?
- What is the difference between Data, Information and Knowledge?

What is an information system?

- An information system is a combination of technology, people, and processes that work together to collect, store, manage, and share data. It helps organizations make decisions, solve problems, and improve efficiency by providing accurate and timely information.

Where do we use information systems?

- Information systems are used in various sectors like business, education, and healthcare to support daily operations and long-term planning. They include hardware, software, databases, and networks, all designed to process and distribute information to users who need it.

What is the difference between Data, Information and Knowledge?

- Data refers to raw, unorganized facts or figures that by themselves have no meaning. For example, numbers, dates, or a list of names are considered data.
- Information is what you get when data is processed, organized, or structured in a way that adds context and meaning. For instance, data about sales figures organized in a report becomes information that can be used to understand business performance.
- Knowledge goes a step further and is the understanding or insight gained from analyzing information. It involves interpreting information and applying it to make decisions or solve problems, such as using sales information to predict future trends or improve strategies.

Module 2

- Explain the role of Enterprise Architecture in IT Governance
- Networking Devices: Hub, Repeater, Switch, Router, Gateway

Understanding Enterprise Architecture in IT Governance

What is Enterprise Architecture (EA)?

- Think of EA as a framework for how an organization's IT (technology) and business processes work together. It helps visualize and organize the different components like systems, data, and processes.

What is IT Governance?

- IT Governance is like a set of rules and guidelines that ensure the organization's IT supports its goals. It helps make sure that technology is used wisely and responsibly.

How Does EA Help with IT Governance?

- Alignment with Business Goals: EA ensures that IT projects and initiatives align with what the business wants to achieve. It's like making sure everyone is moving toward the same goal.
- Standardization: EA helps create standard processes and systems across the organization. This consistency makes it easier to manage and reduces confusion.
- Risk Management: By providing a clear view of all IT components, EA helps identify potential risks (like security issues) and allows organizations to plan ahead to avoid them.
- Informed Decision-Making: EA gives leaders a comprehensive view of technology and business processes, enabling them to make better decisions about where to invest and how to improve.
- Performance Measurement: EA often includes metrics that help track how well IT is performing. This allows organizations to see what's working and what isn't.

How Does EA Help with IT Governance?

- Managing Change: As businesses evolve, EA provides guidance on how to introduce new technologies or processes smoothly, reducing disruption.
- Improved Communication: EA acts as a common language that helps different parts of the organization communicate better, making collaboration easier.
- Regulatory Compliance: EA helps organizations ensure they are following laws and regulations related to technology, making it easier to prove compliance when needed.
- Resource Optimization: By identifying overlapping technologies or processes, EA helps organizations use their resources more effectively, saving time and money.
- Long-term Planning: EA encourages looking ahead and planning for future technology needs, ensuring the organization remains adaptable and sustainable.

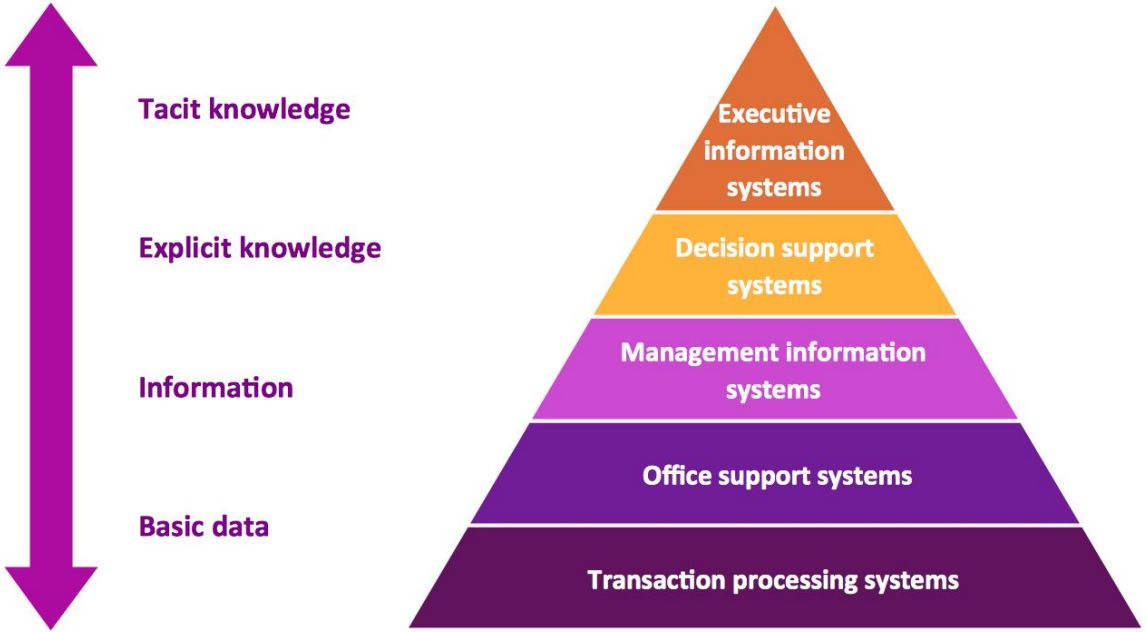
Networking Devices

- Network Definition: A network is a group of connected devices that share data and resources. These networks can vary in scale, from small home setups to global enterprise systems.
- Hub: Broadcasts data to all connected devices. It lacks intelligence, sending data to everyone instead of the intended recipient.
- Switch: Intelligently forwards data only to the intended device within a local network, reducing congestion and allowing full-duplex communication.
- Router: Connects different networks and determines the best path for data between them using IP addresses. Essential for internet connectivity.
- Repeater: Amplifies and retransmits weak signals to extend network range. Operates at the physical layer.
- Gateway: Acts as a translator between different networks, enabling communication by converting protocols.

Module 3

- Explain the role and objectives of Customer Relationship Management (CRM) and Supply Relationship Management (SRM).
- Transaction Processing Systems (TPS)
- Office Automation Systems (OAS)
- Management Information Systems (MIS)
- Decision Support Systems (DSS)
- Executive Information Systems (EIS)

Types of Information Systems Overview



Hierarchy of Information Systems: From Data to Knowledge

- Transaction Processing Systems (TPS): These systems handle basic data, primarily concerned with the day-to-day transactions of an organization. They are foundational, dealing with large volumes of operational data like sales, inventory, and payroll.
- Office Support Systems (OSS): These systems help with the daily operations within an office environment, such as document management, communication (e.g., email), and basic collaboration tools.
- Management Information Systems (MIS): At this level, systems are used to convert raw data from transaction systems into more structured information. MIS provides middle management with reports and summaries, supporting routine decision-making.
- Decision Support Systems (DSS): These systems are used for more complex decision-making, offering tools for data analysis, forecasting, and simulation. DSS helps in processing explicit knowledge, giving managers insights to make informed decisions on non-routine matters.
- Executive Information Systems (EIS): At the top of the hierarchy, these systems are designed for top-level executives. They focus on summarizing and presenting key performance indicators and strategic information, often dealing with tacit knowledge (unwritten, intuitive knowledge) that guides high-level decision-making.

Summary

System	Purpose	Users	Key Features	Example
TPS	Handle routine, high-volume transactions	Operational staff (clerks, cashiers)	Structured, repetitive, real-time processing	POS systems, payroll systems
OAS	Automate routine office tasks	Clerical staff, knowledge workers	Productivity software (word processing, emails, etc.)	Microsoft Office suite
MIS	Provide reports for decision-making	Middle management	Summarized reports from structured data	Sales management systems
DSS	Support decision-making with data analysis	Managers, analysts	Analytical tools, "what-if" analysis, simulations	Forecasting, investment systems
EIS	Provide top-level information for executives	Executives, senior managers	High-level summaries, real-time dashboards	Executive dashboards

What will we cover today?

- MLO 1: Explain basic database terminology.
- MLO 2: Identify personal database application.

What is a Database?

- Definition: A database is a structured collection of interrelated data that supports efficient data retrieval, insertion, and deletion. It organizes data into tables, views, schemas, and reports.
- Example: A university database stores information about students, faculty, and staff, enabling efficient access and management of this data.

What is a Database Management System (DBMS)?

- A Database Management System (DBMS) is a software system that is designed to manage and organize data in a structured manner. It allows users to create, modify, and query a database. DBMS provides an environment to store and retrieve data in convenient and efficient manner.

Database vs. Database Management System (DBMS)

- A Database and a Database Management System (DBMS) are closely related terms, but they serve different purposes:
- A database is a structured set of data. The data can be structured or unstructured and stored in various formats like tables, documents, and key-value pairs. It could be anything from a simple shopping list to a picture gallery or the vast amount of information in a corporate network.
- A Database Management System (DBMS) is software used to interact with a database. It provides an interface for users or applications to manipulate data, making the handling of large amounts of data more efficient and less error-prone. A DBMS oversees core administrative tasks such as data storage, retrieval, security, and query processing.

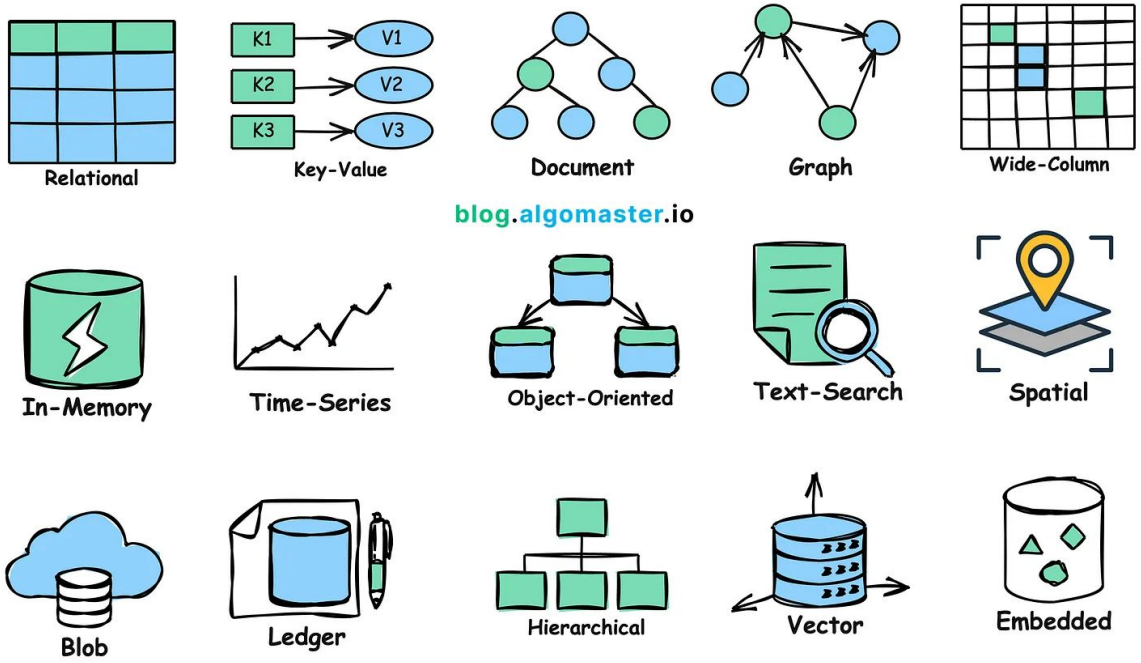
Database Management Systems: Use Cases and Examples

- Economics and Finance: Economic and financial firms store and manage information about customer transactions, stock market trades, bonds, mortgages, and other financial products.
- Healthcare: Healthcare organizations (hospitals, physician practices, etc.) store patient records and medical history that are used for health-data management.
- Government: Government agencies store and manage public records, regulatory data, and administrative information.
- Manufacturing: Manufacturing companies manage production schedules, inventory, and quality control data.
- Research and Academia: Groups such as universities and research institutions store and manage student information, research data, publications, and more.
- Retail: Retail businesses manage inventory, customer transactions, and supply-chain data.
- Software: Software companies store, manage, and migrate large volumes of data that is generated by the applications that they develop.

Benefits of Database Management Systems

- Data Consistency: Data-consistency rules in the DBMS ensure accuracy and uniformity of your data across applications.
- Data Availability: Your data is available continuously, and authorized users can access it even during system failures or disruptions.
- Data-Process Automation: A DBMS enables you to automate various processes, including query execution, transaction management, task scheduling, data archiving, and data backup and recovery.
- Data Security: Data integrity and security are ensured by using constraints on values and by providing access controls (user permissions and access levels) to regulate data access.
- Data Sharing: Multiple users and applications can use the same data simultaneously.
- Data Organization and Management: DBMS tools for organization, indexing, and management enhance overall system performance and reduce storage costs.

Different Types Of Databases



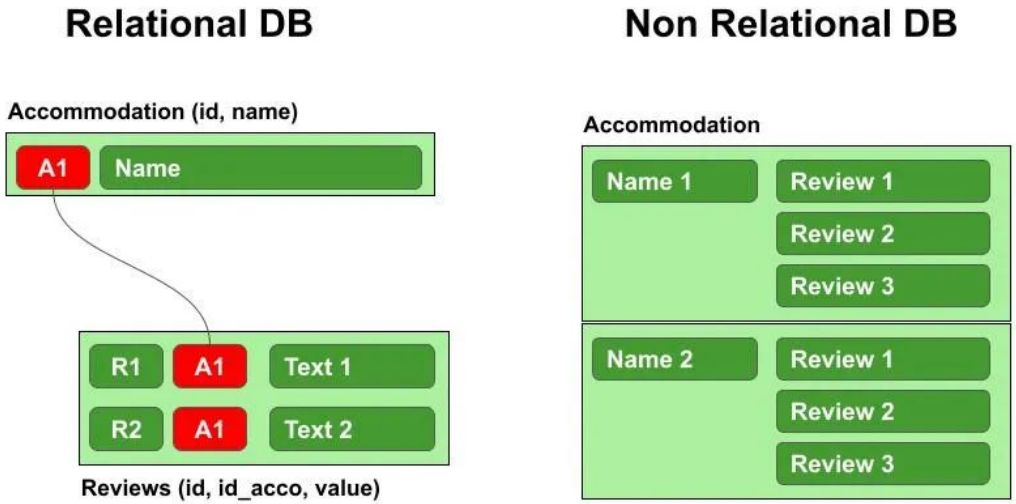
Types Of Databases

- Relational Databases (RDBMS)
- Key-Value Store
- Document Databases
- Graph Databases
- Object-Oriented Databases
- Hierarchical Databases

Relational Databases (RDBMS)

- Relational databases structure data into one or more tables of rows and columns, with a unique key identifying each row. Rows in a table can be linked to rows in other tables through foreign keys, establishing a relationship between them.
- They use Structured Query Language (SQL) for defining, manipulating, and querying data, making them highly versatile and widely used in various applications.

Relational Databases (RDBMS)



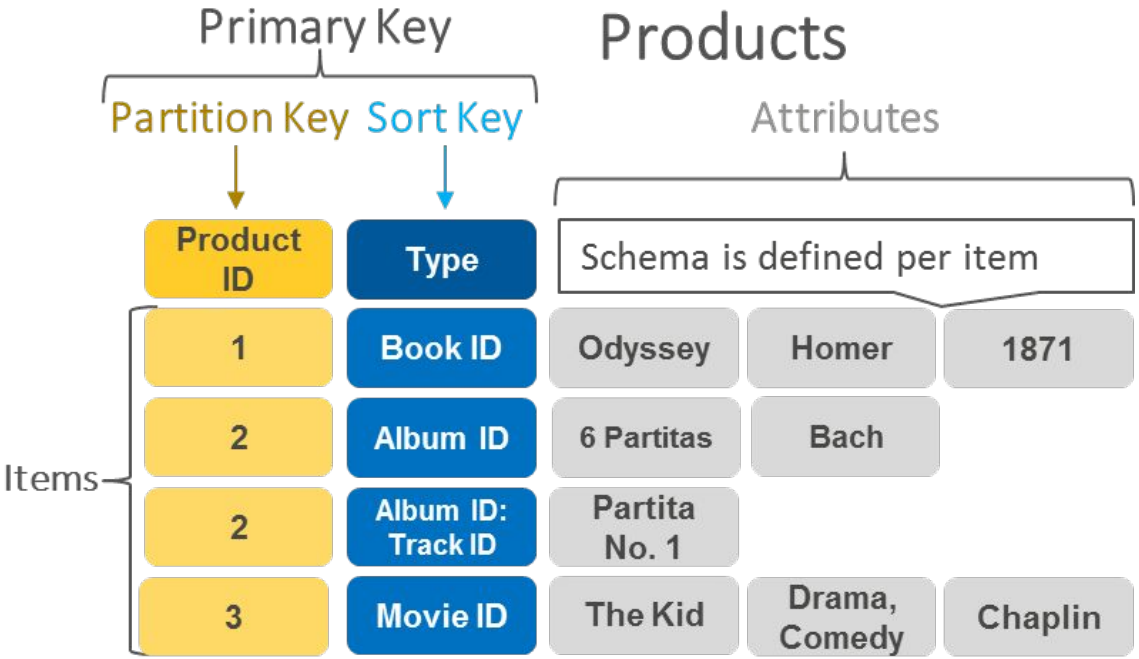
Relational Databases: Common Use-cases

- Enterprise Applications: For managing customer data, inventory, employee records, and financial transactions, where data integrity and relationships are critical.
- E-commerce Platforms: Handling product catalogs, customer orders, and payment transactions, requiring complex queries and transaction processing.
- Banking and Financial Services: Managing accounts, transactions, and user data, where the ACID properties ensure the reliability and consistency of financial operations.
- Examples: MySQL, PostgreSQL, Oracle Database.

Key-Value Store

- Key-value stores are NoSQL databases that store data as key-value pairs providing fast retrieval of values based on unique keys.
- They are primarily used when the data model is based on key-value pairs and requires high scalability, availability and throughput.
- However, they may not be the best fit for applications that require complex querying, data relationships, or strong consistency guarantees.

Key-Value Store



Key-Value Store: Common Use-cases

- Session Storage: Storing and managing user session information such as user preferences, shopping carts or authentication tokens in web applications.
- Caching: Implementing caching mechanisms to improve the performance of web applications by storing frequently accessed data in memory for rapid retrieval.
- Real-time data processing: Key-value stores can quickly store and retrieve data for real-time analytics, event processing, or message queues.
- Examples: Redis, DynamoDB.

Document Databases

- Document databases, a subset of the broader NoSQL family, are designed to store, manage, and retrieve document-oriented information.
- These databases handle data in a semi-structured format, typically JSON or XML, allowing for a more flexible schema than traditional relational databases.
- Document databases are particularly useful in scenarios where the data model evolves frequently, and where fast read and write performance is critical.
- However, they may not be the best fit for highly structured data or complex transactions spanning multiple documents or collections.

Document Databases

```
{
  "_id": "5cf0029caff5056591b0ce7d",
  "firstname": "Jane",
  "lastname": "Wu",
  "address": {
    "street": "1 Circle Rd",
    "city": "Los Angeles",
    "state": "CA",
    "zip": "90404"
  }
  "hobbies": ["surfing", "coding"]
}
```

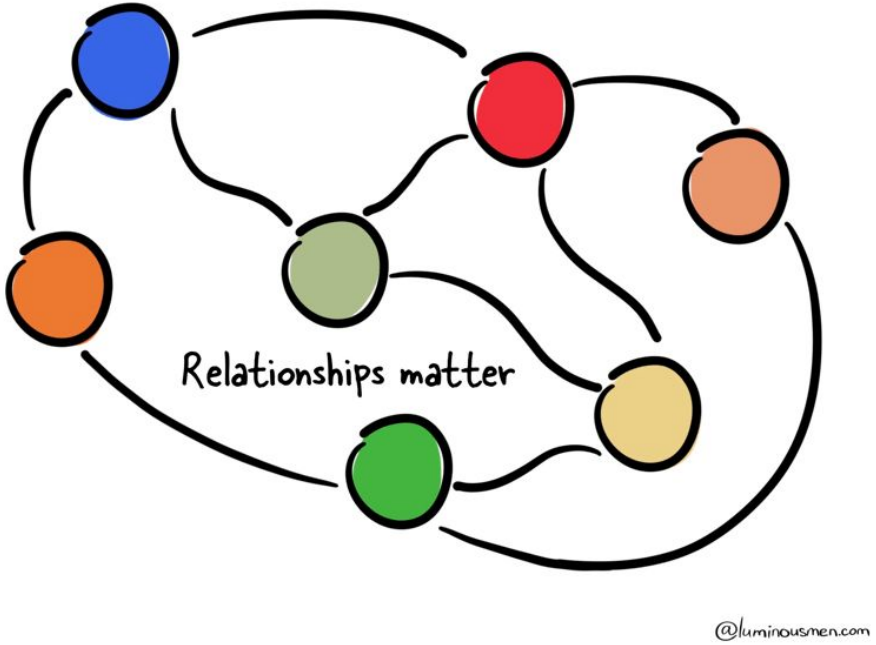
Document Databases: Common Use-cases

- E-commerce Platforms: Storing product catalogs with diverse attributes, user reviews, and inventory data, allowing for flexible representation of product information.
- Content Management Systems (CMS): Ideal for managing articles, user profiles, and comments, where each piece of content can be stored as a document.
- Real-Time Analytics and IoT: Handling varied data structures generated by IoT devices and supporting real-time analytics on this data.
- Examples: MongoDB, Couchbase, and Apache CouchDB

Graph Databases

- Graph databases are a type of NoSQL database that specialize in storing, managing, and querying complex networks of interconnected data.
- They represent data as graphs, consisting of nodes (entities), edges (relationships between entities), and properties (information associated with nodes and edges).
- By leveraging the graph structure, graph databases enable efficient traversal, querying, and analysis of interconnected data.
- They are very useful in applications like social networks and recommendation engines.

Graph Databases



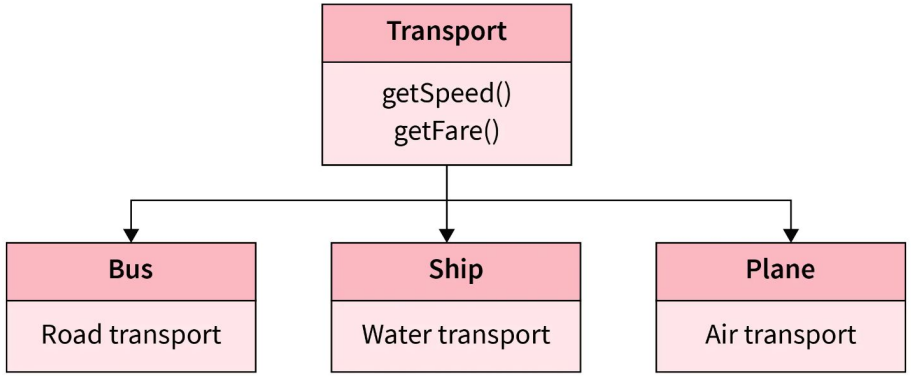
Graph Databases: Common Use-cases

- Social Networks: Managing user profiles and their connections, enabling features like friend recommendations and social graph analysis.
- Recommendation Systems: Analyzing customer preferences, product inventories, and purchase histories to generate personalized product or content recommendations.
- Knowledge Graphs: Building vast repositories of interconnected data for semantic searches, information retrieval, and decision support systems.
- Examples: Neo4j, Amazon Neptune.

Object-Oriented Databases

- Object-oriented databases (OODB) are databases that store and manipulate data as objects.
- These objects are instances of classes, which can encapsulate both data (attributes) and behaviors (methods), mirroring the structure and concepts of object-oriented programming languages like Java, C++, or Python.
- OODBs are particularly well-suited for applications where complex data models are necessary, or the application logic heavily relies on object-oriented principles.
- By allowing developers to work directly with objects in the database, OODBs can simplify the development process and provide a more natural and efficient way to manage complex data structures and relationships.

Object-Oriented Databases



SCALER Topics

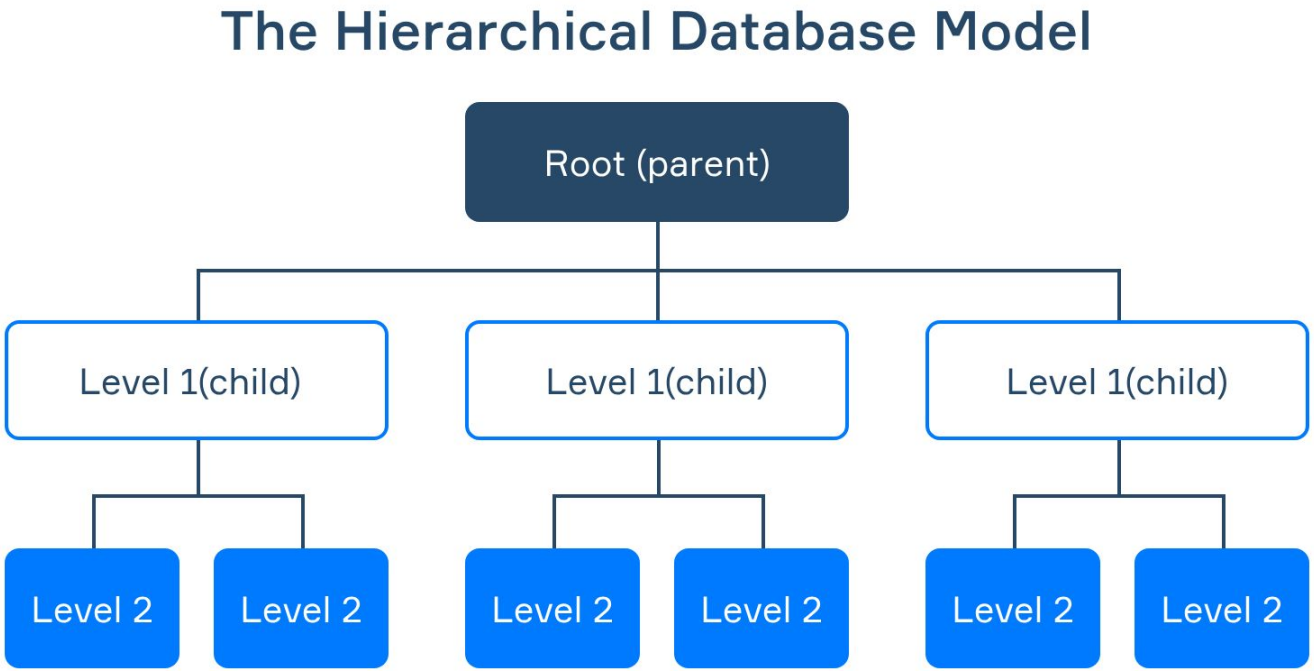
Object-Oriented Databases: Common Use-cases

- Object-Oriented Applications: Applications developed using OOP languages that require a seamless persistence mechanism for storing and retrieving objects without the need to convert them to a different format (object-relational mapping).
- Multimedia Databases: Storing, organizing, and retrieving multimedia items like images, videos, and audio files, which can benefit from the encapsulation of both data and behaviors (e.g., methods to play or edit).
- Examples: ObjectDB, db4o

Hierarchical Databases

- Hierarchical databases organize data into a tree-like structure where data is stored in records, and each record has a single parent record but can have a multiple children, establishing a one-to-many relationship between records.
- Hierarchical databases were popular in the early days of computing, particularly in mainframe systems. They were commonly used for file systems, where directories and files naturally fit into a hierarchical structure.
- However, they have largely been replaced by other database models, such as relational databases and NoSQL databases, which offer more flexibility and better support for complex relationships.

Hierarchical Databases



Hierarchical Databases: Common Use-cases

- Organizational Structures: Managing data in organizational charts where each entity (e.g., employee) has a clear hierarchical relationship.
- File Systems: The directory structure of file systems is a classic example of hierarchical data, where folders have subfolders and files.
- Examples: IBM IMS, Windows Registry

Summary

Database Type	Data Structure	Use Cases	Advantages	Examples
Relational Databases	Tables with rows and columns, structured relationships (SQL-based)	Enterprise applications, banking, e-commerce platforms	Data integrity, complex queries	MySQL, PostgreSQL, Oracle DB
Key-Value Store	Key-value pairs	Caching, session, storage, real-time data processing	Simple, fast retrieval, highly scalable	Redis, DynamoDB
Document Databases	Semi-structured documents	Content management, real-time analytics, IoT	Flexible schema, fast reads/writes, good for evolving data	MongoDB, Couchbase,Apache Couchbase
Graph Databases	Graphs, nodes, edges, properties	Social networks, recommendation systems, knowledge graphs	Efficient traversal of connected data, flexible querying	Neo4j, Amazon Neptune
Object-Oriented Databases	Objects (similar to OOP languages)	Object-oriented applications, multimedia databases	Seamless OOP integration, efficient object management	ObjectDB, db4o
Hierarchical Databases	Tree-like structure (parent-child relationships)	Organizational charts, file systems	Efficient for one-to-many relationships	IBM IMS, Windows Registry