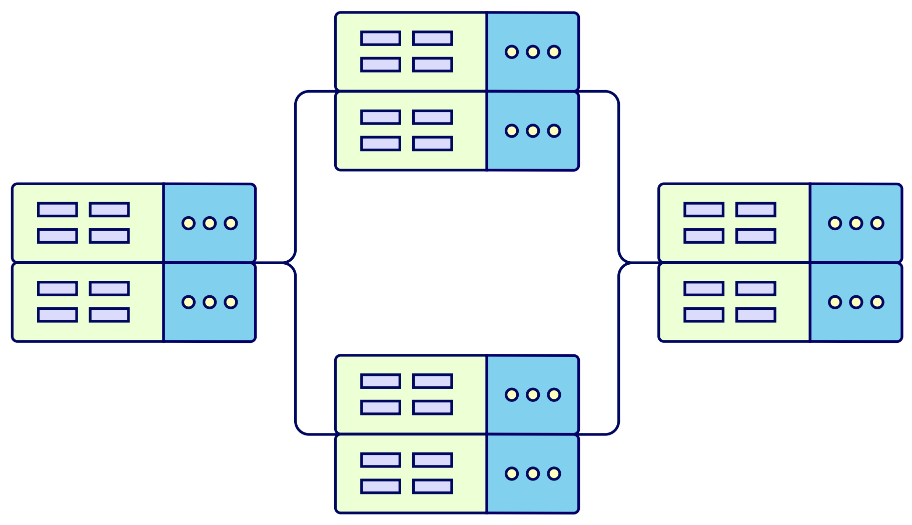
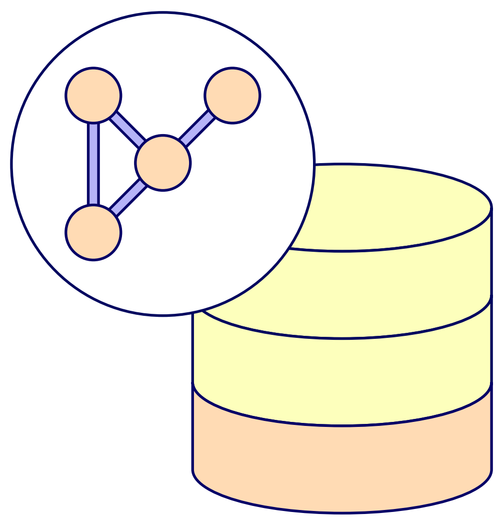
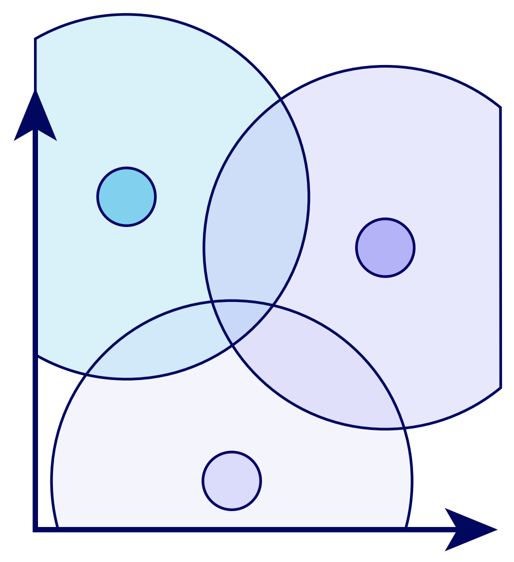
Relational vs. Non-Relational Databases

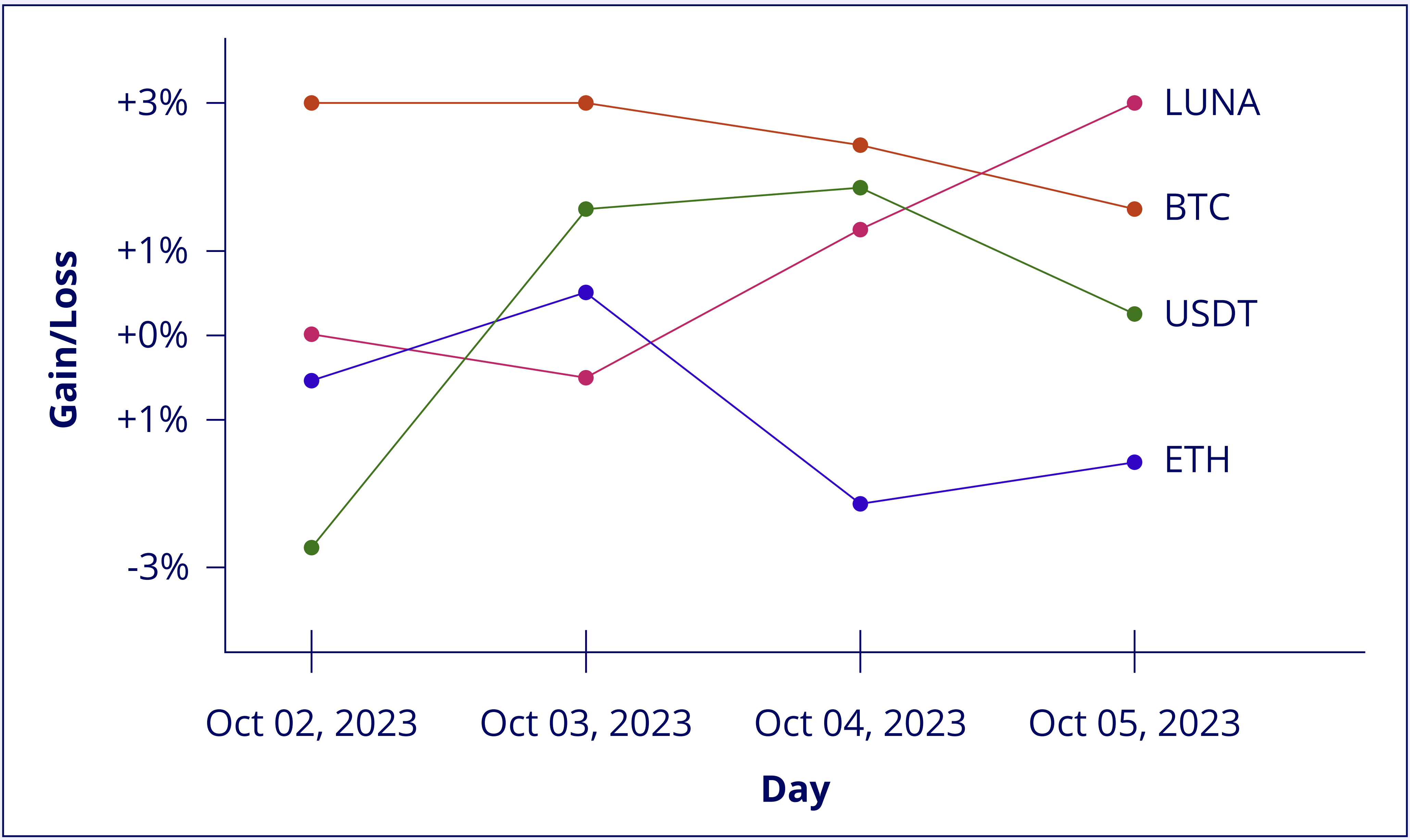
educative

Types of Databases

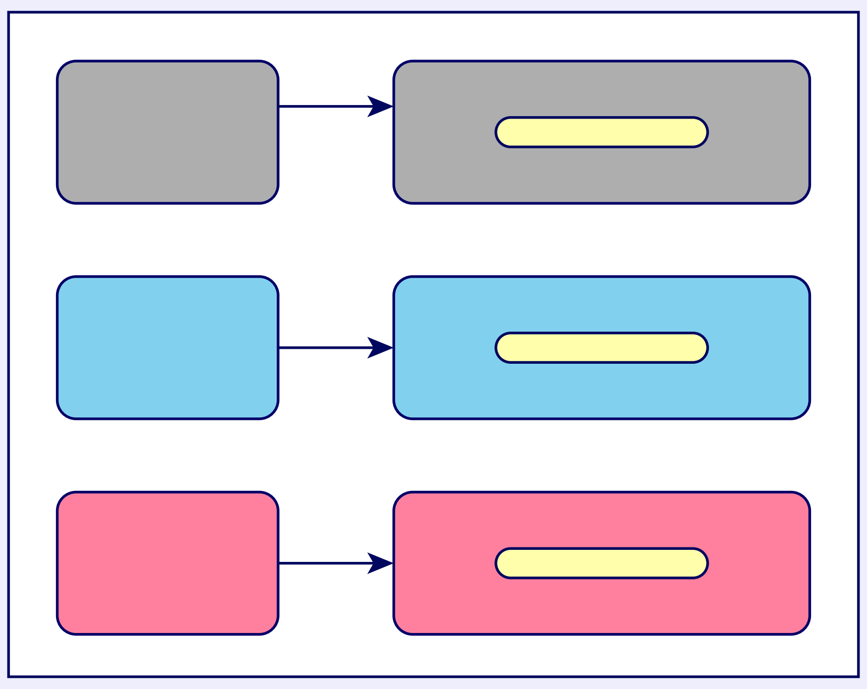
Relational Database	Non-Relational Database	Vector Database
• It stores and organizes structured data into tables. Each table consists of rows and columns. SQL (Structured Query Language) manages and manipulates this data. 	• It's a non-relational database that provides a mechanism for storing and retrieving unstructured data modeled in a non tabular format. 	• A vector database stores data as high-dimensional vectors, representing features or attributes mathematically. 

Relational databases, based on tables, rows, and columns, vary in schema and optimization but do not have different types. Similarly, vector databases store data as high-dimensional vectors and do not have different types.

- **Time Series Database (Non-Relational/Relational):**
  - Built on top of relational or non-relational databases.
  - Efficiently stores, retrieves, and manages time-indexed data.
  - Functions like a columnar key-value database with time stamps as keys.



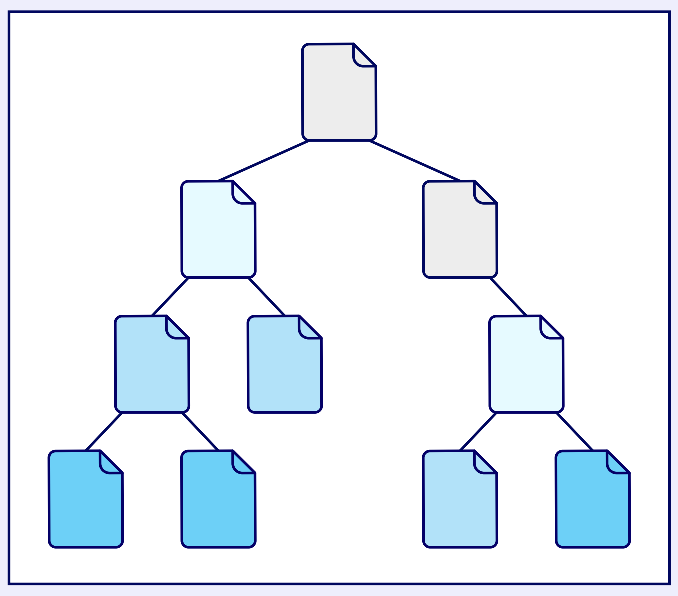
- **Key-Value Stores (Non-Relational):**
  - Stores data in a schema-less way as key-value pairs.
  - Provides fast and simple data access.
  - The popular databases are Redis and Amazon DynamoDB.



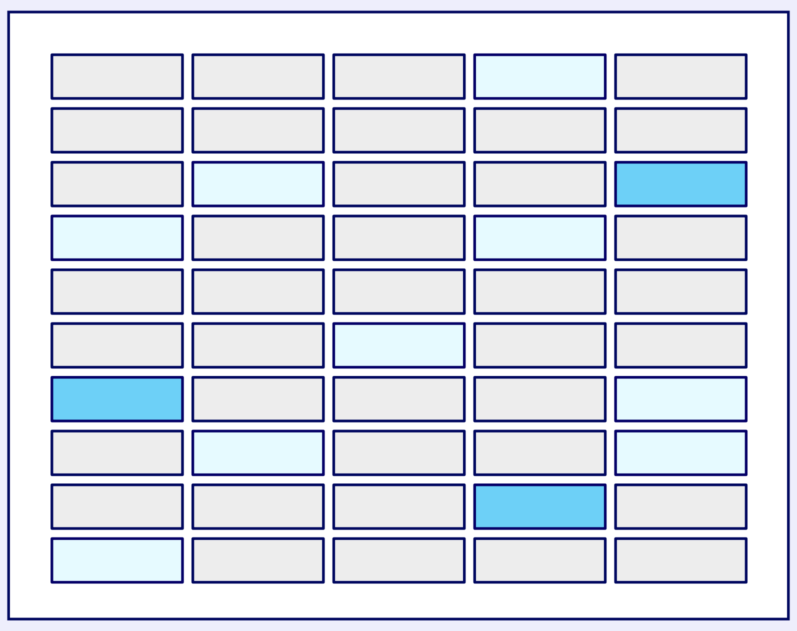
Relational vs. Non-Relational Databases

educative

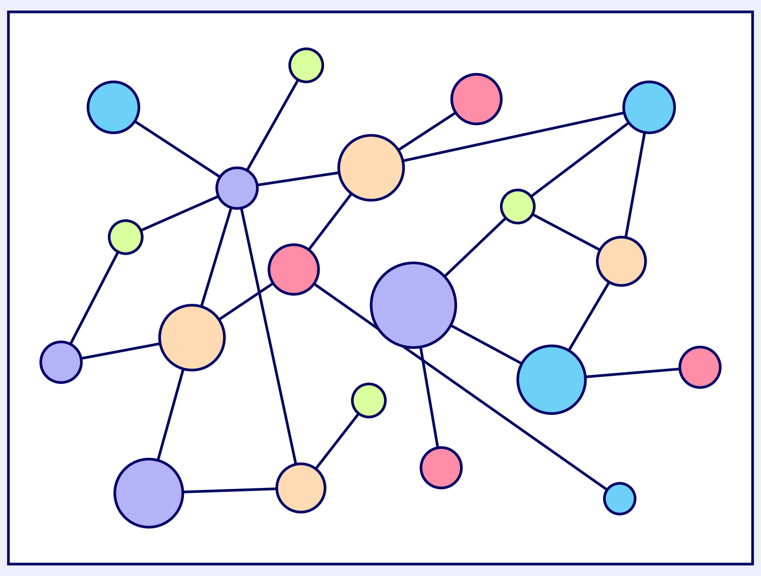
- **Document Stores (Non-Relational):**
  - Stores and retrieves data in flexible JSON or BSON documents.
  - Offers high performance and scalability.
  - The popular databases are MongoDB and Couchbase.



- **Wide-Column Stores (Non-Relational):**
  - Organizes data in columns rather than rows.
  - Allows for efficient storage and retrieval of large datasets.
  - The popular databases are Apache Cassandra and HBase.



- **Graph Databases (Non-Relational):**
  - Models and stores data in terms of entities and their relationships.
  - Makes them suitable for applications involving complex data relationships and network analysis.
  - The popular databases are Neo4j and Amazon Neptune.



Application

Relational Database	Non-Relational Database	Vector Database
1. Enterprise Resource Planning (ERP) Systems: • **Use case:** Managing company operations such as accounting, human resources, and supply chain management. • **Examples:** SAP and Oracle ERP SAP · Oracle ERP	**1. Content Management Systems (CMS):** • **Use case:** Storing and retrieving large volumes of unstructured data such as blogs, articles, and multimedia. • **Examples:** WordPress (with NoSQL for scalability) and Couchbase WordPress · Couchbase	**1. Fraud Detection:** • **Use case:** Identifying patterns and anomalies in transaction data represented as vectors to detect fraudulent activities. • **Examples:** Pinecone and Milvus Pinecone · Milvus

Relational vs. Non-Relational Databases

educative

2. Customer Relationship Management (CRM) Systems:

- **Use case:** Tracking customer interactions, sales processes, and customer support.
- **Examples:** Salesforce and Microsoft Dynamics

salesforce

Salesforce

Microsoft Dynamics

2. Real-Time Big Data Analytics:

- **Use case:** Handling large-scale, distributed data for real-time analytics in applications like social media monitoring and IoT data processing.
- **Examples:** Apache Cassandra and MongoDB

cassandra

Apache Cassandra

mongoDB

MongoDB

2. Semantic Search:

- **Use case:** Improving search results by understanding the semantic meaning of queries and documents rather than relying solely on keyword matching.
- **Examples:** Qdrant

drant

Qdrant

Pros and Cons of Database Types

Relational Databases

- **Pros:**
  - Structured data model
  - Reduced redundancy
  - ACID transactions
  - Data integrity
  - SQL support
- **Cons:**
  - No support for unstructured data
  - Lack of flexibility (rigid schema)
  - Limited scalability
  - Complex interface

Non-Relational Databases

- **Pros:**
  - Flexible schema
  - Horizontal scalability
  - High performance for unstructured data
  - High availability and fault tolerance
- **Cons:**
  - No standardized query language
  - Inefficiency with complex queries
  - Lack of consistency
  - Lack of support for analytics

Vector Database

- **Pros:**
  - Fast similarity searches
  - Scalable solutions
  - Integration with machine learning
  - Managing high-dimensional data
  - Support for embeddings
- **Cons:**
  - Limited support for complex queries
  - Data type constraints
  - Reliance on vector representations

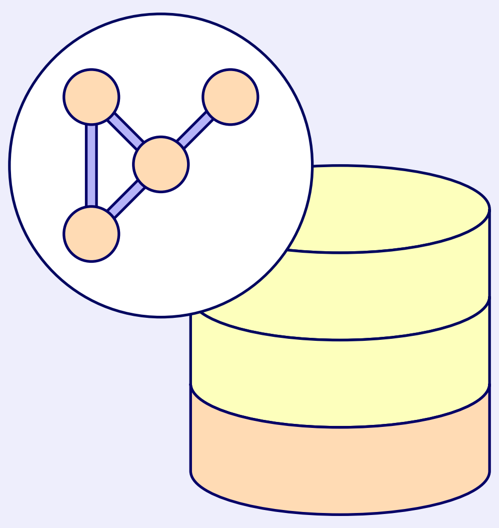
Open Source vs. Closed Source Databases

	SQL	NoSQL	Vector
Open Source	MySQL, MariaDB	HBASE, neo4j, cassandra, COUCHBASE, mongoDB	zilliz, milvus, drant, deeplake
Closed Source	ORACLE, SQL Server		Pinecone

Scenario-Based Example

You are designing an analytics system for a social media platform. The system needs to track user interactions, such as likes, comments, shares, and demographics. Which database type, relational or non-relational, is better suited for this system?

Solution:
A non-relational database is better suited for this system due to its flexibility in handling diverse data structures and scalability for the platform's growth. MongoDB is ideal for this scenario. MongoDB's flexible schema and powerful queries handle diverse user data.



Assess Your Understanding!

You're designing a personalized music recommendation system for a streaming service. The system must handle song metadata, user listening history, personalized playlist generation, similarity searches, and audio feature embeddings. Which database type, relational, non-relational, or vector database, is better suited for this system?

Write your answers in the comment section below!