**1. What is SQL and What is NoSQL?**

**A diagram of a graph and a diagram of a graph

AI-generated content may be incorrect.**

**SQL (Structured Query Language):**

SQL is a standardized programming language used for managing and manipulating relational databases. These databases store data in structured formats using tables with rows and columns. SQL allows users to query, insert, update, and delete data with high accuracy and reliability.

**Key Features of SQL Databases:**

* **Relational Data Model:** Uses tables to represent data and relationships between them.
* **Schema-Based:** A fixed schema defines the structure of the data.
* **ACID Compliance:** Ensures Atomicity, Consistency, Isolation, and Durability for reliable transactions.
* **Examples:** MySQL, PostgreSQL, Oracle, Microsoft SQL Server.

**Use Cases:**

* Applications with complex queries and transactions.
* Systems that require data integrity, such as banking and e-commerce platforms.

**NoSQL (Not Only SQL):**

NoSQL refers to a variety of database technologies designed for flexible, scalable storage and retrieval of data that may not require a fixed schema. It is especially useful for handling large volumes of unstructured or semi-structured data.

**Types of NoSQL Databases:**

* **Document-Based (e.g. MongoDB):** Stores data as JSON-like documents.
* **Key-Value Stores (e.g. Redis, DynamoDB):** Uses key-value pairs.
* **Column-Based (e.g. Cassandra, HBase):** Stores data in columns instead of rows.
* **Graph-Based (e.g. Neo4j):** Stores data as nodes and relationships.

**Key Features:**

* **Flexible Schema:** Data structure can change over time.
* **Horizontal Scalability:** Easily handles massive volumes of data across distributed systems.
* **High Performance for Specific Use Cases.**

**Use Cases:**

* Big data applications, real-time web apps, IoT, content management systems, etc.

**2. What Are the Scenarios in Which One Needs to Choose One Over the Other (SQL vs NoSQL)?**

**Scenario 1: Banking or Financial Applications – Choose SQL**

**Reason:**  
These applications require high data accuracy, strict consistency, and complex transactions involving multiple tables.

**Why SQL?**

* Strong ACID compliance ensures reliable financial transactions.
* Relational structure supports complex joins and queries.
* Data integrity and consistency are critical.

**Example:** Core banking systems, accounting software.

**Scenario 2: E-commerce Catalogue – Choose NoSQL**

**Reason:**  
Product data or content types can vary widely and change frequently.

**Why NoSQL?**

* No need to modify schema every time a new product attribute is added.
* Document-based NoSQL databases (like MongoDB) store varied content types efficiently.
* Easy to scale as product listings grow.

**Example:** Amazon-like e-commerce platform, blogging platforms.

**Scenario 3: Social Media Platforms – Choose NoSQL**

**Reason:**  
Social media platforms generate huge volumes of diverse, fast-changing data such as posts, likes, comments, followers, images, and videos.

**Why NoSQL?**

* Flexible schema to handle varied content types (text, images, videos, reactions, etc.).
* High write/read speed for handling real-time user interactions.
* Horizontal scalability to support millions of users and interactions.
* Graph-based NoSQL databases (like Neo4j) can efficiently manage relationships between users (followers, friends, likes).

**Example:** Facebook, Instagram, Twitter-like platforms.

**3. Advantages of SQL Databases**

1. **Structured Data and Schema Enforcement**

* Enforces a strict schema, which ensures data is highly organized and consistent.
* Ideal for applications where data structure does not change frequently.

1. **ACID Compliance**

* Ensures reliable transactions through Atomicity, Consistency, Isolation, and Durability.
* Critical for applications that require high data integrity, such as banking and inventory management.

1. **Powerful Query Capabilities**

* SQL provides powerful and standardized query language for complex data retrieval.
* Supports complex joins, filtering, aggregations, and nested queries.

1. **Mature Ecosystem**

* Widely adopted, well-documented, and supported by a large community.
* Rich tools for reporting, data migration, backups, and performance tuning.

**4. Advantages of NoSQL Databases**

1. **Flexible Schema**

* Schema-less design allows storage of structured, semi-structured, or unstructured data.
* Suitable for applications where the data model evolves frequently.

1. **High Scalability**

* Designed for horizontal scaling across distributed systems.
* Ideal for handling massive datasets and high throughput requirements.

1. **Performance for Specific Use Cases**

* Optimized for high-speed read and write operations.
* Excellent for use cases like caching, real-time analytics, and large-scale web applications.

1. **Variety of Data Models**

* Offers different types (Document, Key-Value, Columnar, Graph), allowing selection based on specific application needs.
* For example, document databases are great for content management; graph databases excel at social networks.