NonRelational Database

➢ What is and Why NoSQL

NoSQL databases are purpose built for specific data models and have flexible schemas for building modern applications. NoSQL databases are widely recognized for their ease of development, functionality, and performance at scale. This page includes resources to help you better understand NoSQL databases and to get started.

NoSQL databases are a great fit for many modern applications such as mobile, web, and gaming that require flexible, scalable, high-performance, and highly functional databases to provide great user experiences.

- Flexibility: NoSQL databases generally provide flexible schemas that enable faster and more iterative development. The flexible data model makes NoSQL databases ideal for semi-structured and unstructured data.

- Scalability: NoSQL databases are generally designed to scale out by using distributed clusters of hardware instead of scaling up by adding expensive and robust servers. Some cloud providers handle these operations behind-the-scenes as a fully managed service.

- High-performance: NoSQL database are optimized for specific data models and access patterns that enable higher performance than trying to accomplish similar functionality with relational databases.

- Highly functional: NoSQL databases provide highly functional APIs and data types that are purpose built for each of their respective data models.

➢ NoSQL characteristics

It's more than rows in tables—NoSQL systems store and retrieve data from many formats:

key-value stores, graph databases, column-family (Bigtable) stores, document stores, and even rows in tables.

• It's free of joins—NoSQL systems allow you to extract your data using simple interfaces without joins.

• It's schema-free—NoSQL systems allow you to drag-and-drop your data into a folder and then query it without creating an entity-relational model.

It works on many processors—NoSQL systems allow you to store your database on multiple processors and maintain highspeed performance.

• It uses shared-nothing commodity computers—Most (but not all) NoSQL systems leverage low-cost commodity processors that have separate RAM and disk.

• It supports linear scalability—When you add more processors, you get a consistent increase in performance.

It's innovative—NoSQL offers options to a single way of storing, retrieving, and manipulating data. NoSQL supporters (also known as NoSQLers) have an inclusive

• attitude about NoSQL and recognize SQL solutions as viable options. To the NoSQL community, NoSQL means "Not only SQL."

➤ NoSQL databases types

Key-value: Key-value databases are highly partitionable and allow horizontal scaling at scales that other types of databases cannot achieve. Use cases such as gaming, ad tech, and IoT lend themselves particularly well to the key-value data model. Amazon DynamoDB is designed to provide consistent single-digit millisecond latency for any scale of workloads. This consistent performance is a big part of why the Snapchat Stories feature, which includes Snapchat's largest storage write workload, moved to DynamoDB.

Document: In application code, data is represented often as an object or JSON-like document because it is an efficient and intuitive data model for developers. Document databases make it easier for developers to store and query data in a database by using the same document model format that they use in their application code. The flexible, semistructured, and hierarchical nature of documents and document databases allows them to evolve with applications' needs. The document model works well with catalogs, user profiles, and content management systems where each document is unique and evolves over time. Amazon DocumentDB (with MongoDB compatibility) and MongoDB are popular document databases that provide powerful and intuitive APIs for flexible and iterative development.

Graph: A graph database's purpose is to make it easy to build and run applications that work with highly connected datasets. Typical use cases for a graph database include social networking, recommendation engines, fraud detection, and knowledge graphs. Amazon Neptune is a fully-managed graph database service. Neptune supports both the Property Graph model and the Resource Description Framework (RDF), providing the choice of two graph APIs: TinkerPop and RDF/SPARQL. Popular graph databases include Neo4j and Giraph.
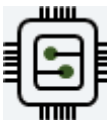
In-memory: Gaming and ad-tech applications have use cases such as leaderboards, session stores, and real-time analytics that require microsecond response times and can have large spikes in traffic coming at any time. Amazon ElastiCache offers Memcached and Redis, to serve low-latency, high-throughput workloads, such as McDonald's, that cannot be served with disk-based data stores. Amazon DynamoDB Accelerator (DAX) is another example of a purpose-built data store. DAX makes DynamoDB reads an order of magnitude faster.cases. Expedia is using more than 150 Amazon ES domains, 30 TB of data, and 30 billion documents for a variety of mission-critical use cases, ranging from operational monitoring and troubleshooting to distributed application stack tracing and pricing optimization.

Search: Many applications output logs to help developers troubleshoot issues. Amazon Elasticsearch Service (Amazon ES) is purpose built for providing near-real-time visualizations and analytics of machine-generated data by indexing, aggregating, and searching semistructured logs and metrics. Amazon ES also is a powerful, high-performance search engine for full-text search use cases. Expedia is using more than 150 Amazon ES domains, 30 TB of data, and 30 billion documents for a variety of mission-critical use cases, ranging from operational monitoring and troubleshooting to distributed application stack tracing and pricing optimization.

➢ What is ACID theorem

In computer science, ACID (atomicity, consistency, isolation, durability) is a set of properties of database transactions intended to guarantee data validity despite errors, power failures, and other mishaps. In the context of databases, a sequence of database operations that satisfies the ACID properties (which can be perceived as a single logical operation on the data) is called a transaction


ACID Properties in DBMS
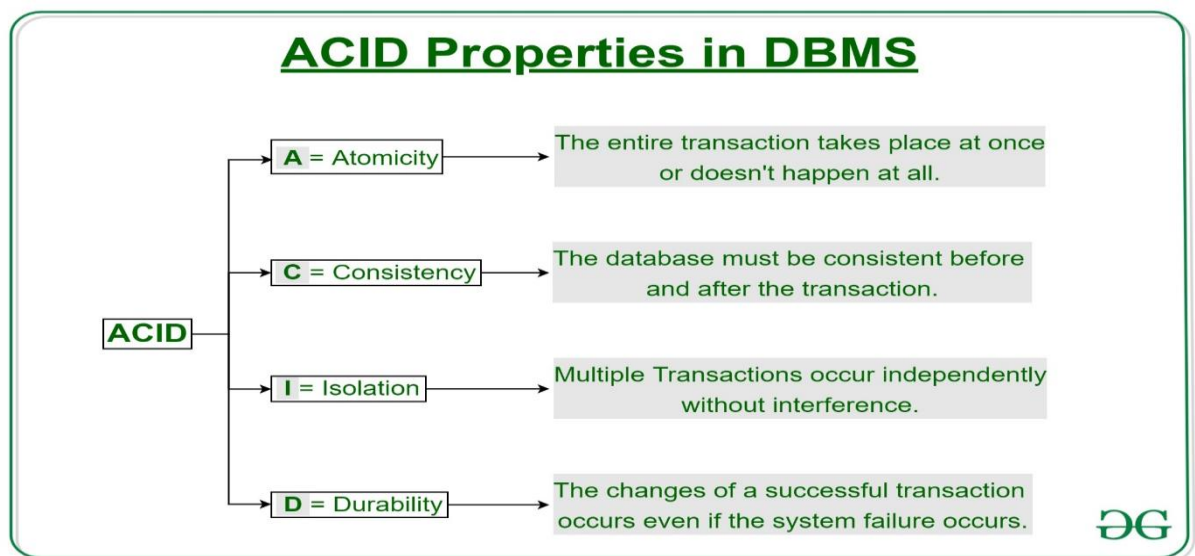
➢ What is CAP theorem

CAP theorem also known as Brewer's theorem was introduced by computer scientist Eric Brewer at Symposium on Principles of Distributed computing in 2000. In CAP theorem, C stands for Consistency, A stands for Availability and P stands for Partition tolerance.

 Today, NoSQL databases are classified based on the two CAP characteristics they support:

- CP database: A CP database delivers consistency and partition tolerance at the expense of availability. When a partition occurs between any two nodes, the system has to shut down the non-consistent node (i.e., make it unavailable) until the partition is resolved.
- AP database: An AP database delivers availability and partition tolerance at the expense of consistency. When a partition occurs, all nodes remain available but those at the wrong end of a partition might return an older version of data than others. (When the partition is resolved, the AP databases typically resync the nodes to repair all inconsistencies in the system.)
- CA database: A CA database delivers consistency and availability across all nodes. It can't do this if there is a partition between any two nodes in the system, however, and therefore can't deliver fault tolerance.
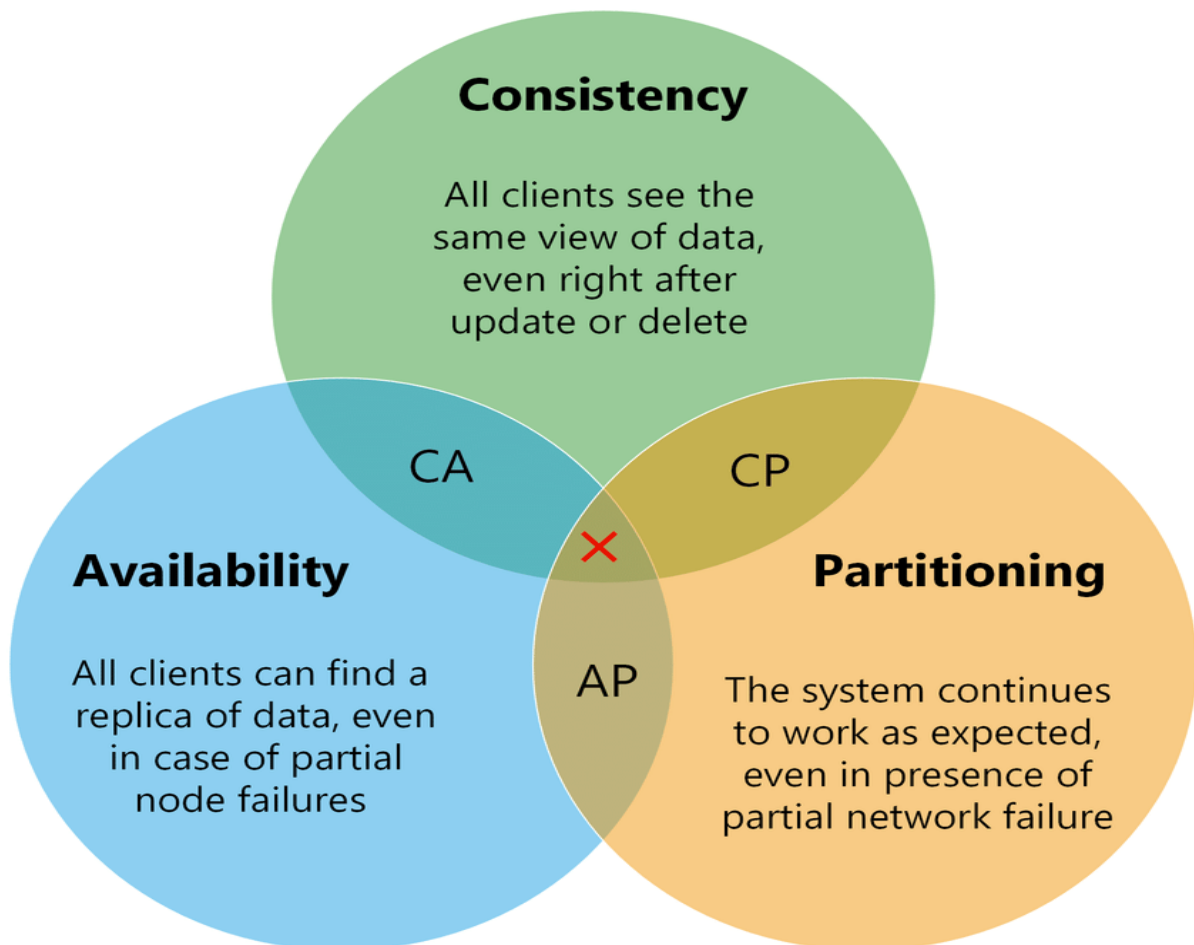
➢ NoSQL advantages

One of the benefits of NoSQL over SQL is that the database can grow horizontally, we can add data without losing the structure, and in order for the performance to remain at the same level, we just need to update the equipment or add more hosts, rather than buying more servers.

It can store and work with enormous amounts of data; for example, NoSQL databases are popular among social media apps.

The data can be distributed across many cheap commodity servers, and the performance will stay at the highest level.

The maintenance is cheap, as the database has built-in functions like automatic repair and simpler data models.

No schema is required to load data into a NoSQL database because the format and model can be changed at any time.

The system memory is used to store cache data.