
PRACTICAL 1

It refers to Non SQL or “Not Only SQL” which is a non-relational database which provides the facility to store and retrieve data in a non-structured manner. It is generally used to store big data and real time web applications.

Characteristics of NoSQL:

- Never follows relational database model.
- It is schema free.
- It provides faster performance.
- Can handle Data variety and huge amounts of data.
- It is both horizontal and vertical scalable. □ It has low cost hardware.

Advantages of NoSQL:

- Flexible Schema.
- Horizontal Scaling.
- Fast queries due to the data model.
- Ease of use for developers.
- Best for big and complex data.

Disadvantages of NoSQL:

- Lack of Standardization.
- Not complete data backup.
- Lack of consistency in the database.

There are four types of NoSQL Databases:

1. Document-Oriented Database
2. Key-Oriented Database
3. Column-Oriented Database
4. Graph

DOCUMENT-ORIENTED DATABASE:

A document database stores data in JSON, BSON, or XML documents. Documents can be stored and retrieved in a form that is much closer to the data objects used in applications, which means less translation is required to use the data in an application. SQL data must often be assembled and disassembled when moving back and forth between applications and storage. It is widely used by the developers as the document is easy to write and update. Use cases include ecommerce platforms, trading platforms, and mobile app development across industries. Example: MonogoDB

KEY-VALUE DATABASE:

The simplest type of NoSQL database is a key-value store. Every data element in the database is stored as a key value pair consisting of an attribute name (or "key") and a value. In a sense, a keyvalue store is like a relational database with only two columns: the key or attribute name (such as "Name") and the value (such as "Abc"). Use cases include shopping carts, user preferences, and user profiles.

Example: Berkeley DB

COLUMN-ORIENTED DATABASE:

While a relational database stores data in rows and reads data row by row, a column store is organized as a set of columns. This means that when you want to run analytics on a small number of columns, you can read those columns directly without consuming memory with the unwanted data. Columns are often of the same type and benefit from more efficient compression, making reads even faster. Columnar databases can quickly aggregate the value of a given column (adding up the total sales for the year, for example). Use cases include analytics. Example: Apache Cassandra

GRAPH:

A graph database focuses on the relationship between data elements. Each element is stored as a node (such as a person in a social media graph). The connections between elements are called links or relationships. In a graph database, connections are first-class elements of the database, stored directly. In relational databases, links are implied, using data to express the relationships. A graph database is optimized to capture and search the connections between data elements, overcoming the overhead associated with joining multiple tables in SQL. Use cases include fraud detection, social networks, and knowledge graphs.

Example: Neo4j

MONGODB:

MongoDB is an open-source document database and leading NoSQL database. It is one of the popular NoSQL database which is object-oriented database. It stores the data in JSON format meaning fields can vary from document to document and data structure can be changed over time. MongoDB is a distributed database at its core, so high availability, horizontal scaling, and geographic distribution are built in and easy to use.

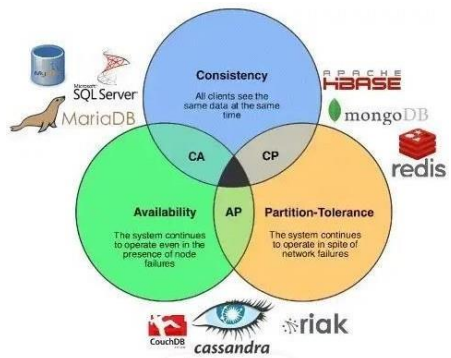
Eg: {"name": "Kushal Gupta", "branch": "IT", "age": "19"}

MongoDB can be used to store which is either larger, unstable or distributed data.

CAP (CONSISTENCY, AVAILABILITY, PARTITION TOLERANCE) THEOREM:

- Consistency: Data should be consistent
- Availability: Availability should be high
- Partition Tolerance: Data should be partitioned. i.e., Data should be able to tolerate further partitions of database

According to CAP theorem, it is not possible by any single database to achieve all these three properties. However, two of the three properties can be achieved alternatively by any database as



shown below:

BASE (BASICALLY AVAILABLE, SOFT STATE, EVENTUAL CONSISTENCY) THEOREM:

- Basically Available: Guarantees the availability of the data. There will be a response to any request.
- Soft State: The state of the system could change over time.
- Eventual consistency: The system will eventually become consistent once it stops receiving input.