**No SQL**

NoSQL, which stands for “not only SQL,” or sometimes “non-SQL” is a non-relational database design that provides flexible schemas for the storage and retrieval of data. NoSQL databases have existed for many years but have only recently become more popular in the era of cloud, big data, and high-volume web and mobile applications. They are chosen today for their attributes around scale, performance, and ease of use.

It's important to emphasize that the "No" in "NoSQL" is an abbreviation for "not only"

and not the actual word "No."

NoSQL databases are built for specific data models and have flexible schemas that allow

programmers to create and manage modern applications. They do not use a traditional row/column/table database design with fixed schemas, and typically not use the structured query language (or SQL) to query data, although some may support SQL or SQL-like interfaces. NoSQLallows data to be stored in a schema-less or free-form fashion. Any data, be It structured, semi-structured, or unstructured, can be stored in any record.

Based on the model being used for storing data, there are four common types of NoSQL

databases: Key-value store, Document-based, Column-based, and graph-based.

* **Key-value store**:

Data in a key-value database is stored as a collection of key-value pairs. The key represents an attribute of the data and is a unique identifier. Both keys and values can be anything from simple integers or strings to complex JSON documents.

Key-value stores are great for storing user session data and user preferences, making real-time recommendations and targeted advertising, and in-memory data caching. However, if you want to be able to query the data on specific data value, need relationships between data values, or need to have multiple unique keys, a key-value store may not be the best fit.

Redis, Memcached, and DynamoDB are some well-known examples in this category.

* **Document-based**:

Document databases store each record and its associated data within a single document. They enable flexible indexing, powerful ad hoc queries, and analytics over collections of documents. Document databases are preferable for eCommerce platforms, medical records storage, CRM platforms, and analytics platforms.

However, if you’re looking to run complex search queries and multi-operation transactions, a document-based database may not be the best option for you.

MongoDB, DocumentDB, CouchDB, and Cloud ant are some of the popular document-based databases.

* **Column-based**:

Column-based models store data in cells grouped as columns of data instead of rows. A logical grouping of columns, that is, columns that are usually accessed together, is called a column family.

For example, a customer’s name and profile information will most likely be accessed together but not their purchase history. So, customer name and profile information data can be grouped into a column family. Since column databases store all cells corresponding to a column as a continuous disk entry, accessing and searching the data becomes very fast. Column databases can be great for systems that require heavy write requests, storing time-series data, weather data, and IoT data. But if you need to use complex queries or change your querying patterns frequently, this may not be the best option for you.

The most popular column databases are Cassandra and HBase.

* **Graph-based**:

Graph-based databases use a graphical model to represent and store data. They are particularly useful for visualizing, analysing, and finding connections between different pieces of data. The circles are nodes, and they contain the data. The arrows represent relationships.

Graph databases are an excellent choice for working with connected data, which is data that contains lots of interconnected relationships. Graph databases are great for social networks, real-time product recommendations, network diagrams, fraud detection, and access management.

But if you want to process high volumes of transactions, it may not be the best choice for you, because graph databases are not optimized for large-volume analytics queries.

Neo4J and CosmosDB are some of the more popular graph databases.

Advantages of NoSQLNoSQL was created in response to the limitations of traditional relational

database technology. The primary advantage of NoSQL is its ability to handle large volumes of structured, semi-structured, and unstructured data.

Some of its other advantages include:

* The ability to run as distributed systems scaled across multiple data centres, which enables them to take advantage of cloud computing infrastructure.
* An efficient and cost-effective scale-out architecture that provides additional capacity and performance with the addition of new nodes; and
* Simpler design, better control over availability, and improved scalability that enables you to be more agile, more flexible, and to iterate more quickly

To summarize the key differences between relational and non-relational databases:

* RDBMS schemas rigidly define how all data inserted into the database must be typed and composed, whereas NoSQL databases can be schema-agnostic, allowing unstructured and semi-structured data to be stored and manipulated.
* Maintaining high-end, commercial relational database management systems is expensive whereas NoSQL databases are specifically designed for low-cost commodity hardware Relational databases, unlike most NoSQL, support ACID-compliance, which ensures reliability of transactions and crash recovery.
* RDBMS is a mature and well-documented technology, which means the risks are perceivable as compared to NoSQL, which is a relatively newer technology. Nonetheless, NoSQL databases are here to stay, and are increasingly being used for mission critical applications.