NoSQL Database

It stands for “non-SQL” database, or we can say that it is a non-relational database. It is complementary to RDBMS. It is built on the principle of a distributed system. Therefore, they are natively scalable. NoSQL is the umbrella term comprising of four different types of databases.

Example: MongoDB, Redis etc.

Advantages of NoSQL Database:

1. Semi-structured data
2. Dynamic or flexible schema
3. Non-relational data
4. No need for complex joins
5. Store many TB (or PB) of data
6. Very data-intensive workload
7. Very high throughput for IOPS

Example 1:

You wish to develop an e-commerce website that would contain multiple databases and tables storing user’s login information and staff details including ID, date of joining, phone number, address, email id.

What would be the suitable database choice here out of SQL or NoSQL databases?

Answer 1: The appropriate database here is an SQL database since the data is structured primarily. Transaction-oriented systems, such as customer relationship management tools, accounting software, and e-commerce platforms, benefit greatly from SQL databases.

Example 2:

Let's pretend you're working on the next Google Analytics. You've decided to monitor IP addresses, browsers, and device types. However, you later realise that you might want to keep track of Browser Size as well. It's not straightforward to add an extra column to your table because analytics databases can include millions or billions of records. It would simply be too time-consuming.

What will you choose from SQL and NoSQL databases for this situation?

Answer 2: We should select the NoSQL database here since it has a dynamic and flexible schema that would allow us to add and append rows in the database without changing and re-ordering each row of the tracking table.

The four different types of NoSQL Databases are:

1. Key-value store
2. Document store
3. Column-oriented database
4. Graph database

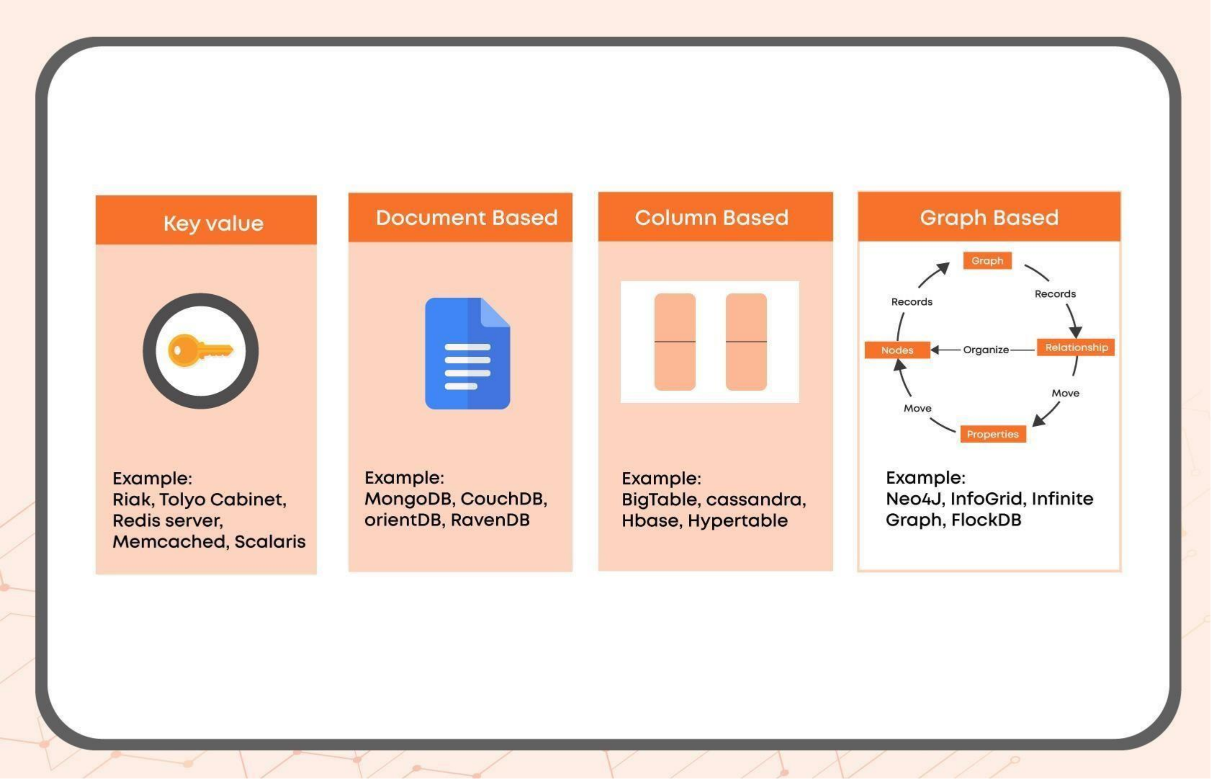


Fig: Different types of NoSQL Databases

1. **<u>Key-Value Database:</u>**

   It stores data in pairs of keys and values. Therefore, it is a non-relational database because the data is not stored in a table but in the form of the key-value method. It is widely used as a caching solution. Suppose we need to store the names of mentors working at Coding Ninjas. The "name" would be the key, and the value would be different names of mentors.

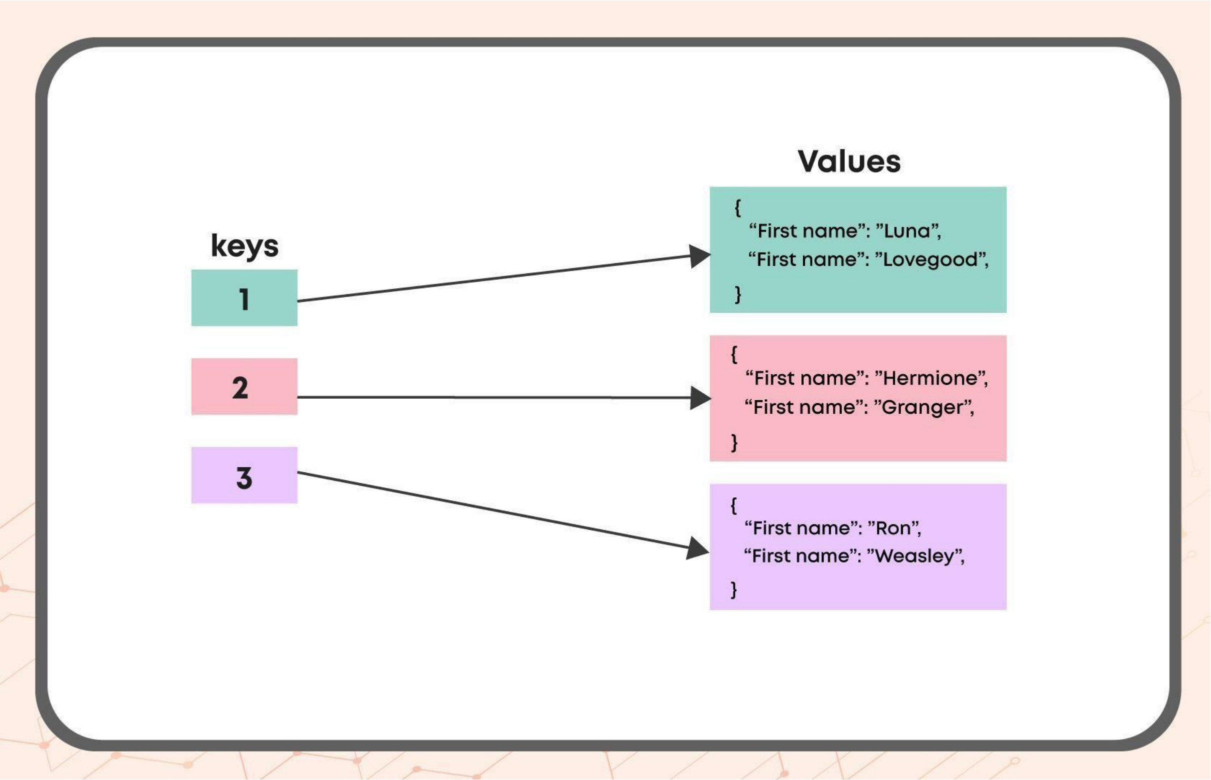
   The TTL column (time to live) is a helpful feature in this database; it may be set differently for each entry and indicates when it should be destroyed from the database.

   <u>Example:</u> Redis

   <u>When should we use Key-Value DB?</u>

   Because it is fast and does not require extensive queries, it is primarily used for caching. The TTL function is also highly beneficial for caching.

   It can also be used for any other type of data with a key-value format that requires quick querying.



2. **Document Database:**

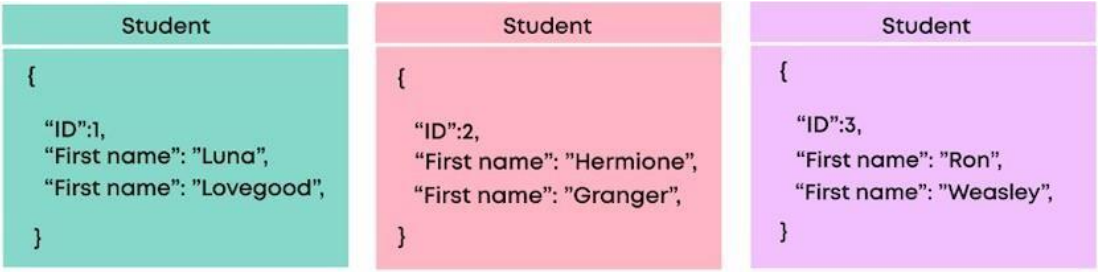
Data is stored in the form of JSON like documents. It combines the concepts of RDBMS and NoSQL databases. It combines the relationship concept from RDBMS and dynamic schema and horizontal scaling from NoSQL databases.

Example: MongoDB

When should we use Document DB?

Data analysis: This database facilitates parallel computations because separate records are not logically or structurally dependent on one another.

This makes it simple to run big data analytics on our data.

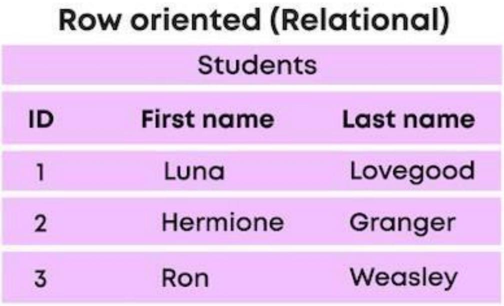


3. **Columnar Database:**

A columnar database is one in which the columns are stored together instead of rows. Most of the operations are column-oriented in an application, and because of that, the aggregation in such databases is rapid. It is widely used for running analytical queries or Time Series Data.
Example: Cassandra

When should we use Columnar DB?
When we query on a subset of your data's columns.
Because it just needs to read these specific columns, columnar DB conducts such queries quickly (while row-based DB would have to read the entire data).

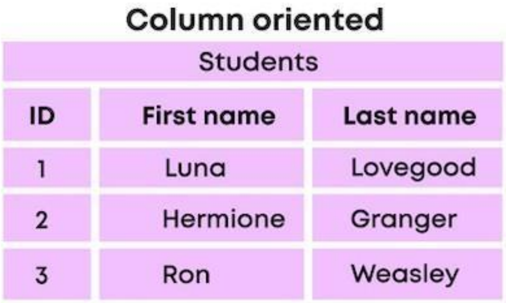
- Each column represents a feature, which is frequent in data science. Data scientist frequently trains models with subsets of the features and frequently examine feature-score relationships (correlation, variance, significance).
- They often save a lot more attributes in our logs database but only use a few in each query, which is also common with logs.



Row oriented (Relational)

Students

ID	First name	Last name
1	Luna	Lovegood
2	Hermione	Granger
3	Ron	Weasley



Column oriented

Students

ID	First name	Last name
1	Luna	Lovegood
2	Hermione	Granger
3	Ron	Weasley

4. Graph Database:

Graph database represents and stores entities and relationships in the form of graph data structure. It is majorly used for social networks and applications incorporating relationships with entities like google maps

.

Example: Neo4j

When should we use Graph DB?

When your data is in the form of a graph, such as knowledge graphs or social networks.

Summary on usage

Key-Value Database	Caching
Graph Database	A graph like data for example social networks.
Columnar Database	If querying on columns is required
Document Database	Flexible Schema or we might need to change the schema in future