NoSQL Data Architecture Patterns

Architecture Pattern is a logical way of categorising data that will be stored on the Database. NoSQL is a type of database which helps to perform operations on big data and store it in a valid format. It is widely used because of its flexibilty and wide variety of services.

Architecture Patterns of NoSQL:
The data is stored in NoSQL in any of the following four data architecture patterns.

1. Key-Value Store Database

2. Column Store Database

3. Document Database

4. Graph Database

These are explained as following below.

1. Key-Value Store Database:
This model is one of the most basic models of NoSQL databses. As the name suggests, the data is stored in form of Key-Value Pairs.The key is usually a sequence of strings, integers or characters but can also be a more advanced data type. The value is typically linked or co-related to the key. The key-value pair storage databases generally store data as a hash table where each key is unique. The value can be of any type (JSON, BLOB(Binary Large Object), strings, etc). This type of pattern is usually used in shopping websites or e-commerce applications.

Advantages:
- Can handle large amounts of data and heavy load,
- Easy retrieval of data by keys.

Limitations:
- Complex queries may attempt to involve multiple key-value pairs which may delay performance.
- Data can be involving many-to-many relationships which may collide.

Examples:
- DynamoDB
- Berkeley DB

Key:1	ID:210		

Key:2	ID:411	Email: geeksforgeeks@gmail.com	

Key:3	UID:219	Name: Geek	Age:20

2. Column Store Database:

Rather than storing data in relational tuples, the data is stored in individual cells which are further grouped into columns. Column-oriented databases work only on columns. They store large amounts of data into columns together. Format and titles of the columns can diverge from one row to other. Every column is treated separately.But still, each individual column may contain multiple other columns like traditional databases.
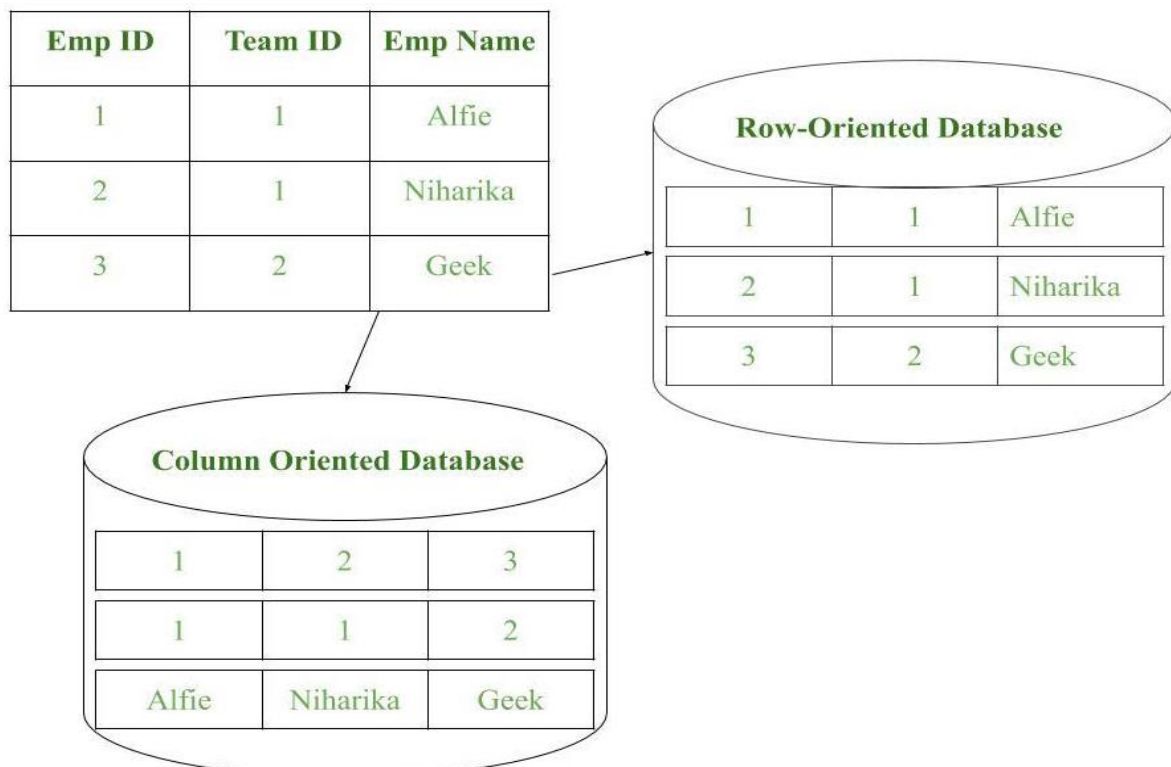
Basically, columns are mode of storage in this type.

Advantages:

- Data is readily available

- Queries like SUM, AVERAGE, COUNT can be easily performed on columns.

Examples:

- HBase

- Bigtable by Google

- Cassandra

Emp ID	Team ID	Emp Name
1	1	Alfie
2	1	Niharika
3	2	Geek

Row-Oriented Database

1	1	Alfie
2	1	Niharika
3	2	Geek

Column Oriented Database

1	2	3
1	1	2
Alfie	Niharika	Geek

3. Document Database:

The document database fetches and accumulates data in forms of key-value pairs but here, the values are called as Documents. Document can be stated as a complex data structure. Document here can be a form of text, arrays, strings, JSON, XML or any such format. The use of nested

documents is also very common. It is very affective as most of the data created is usually in form of JSONs and is unstructured.

Advantages:

- This type of format is very useful and apt for semi-structured data.

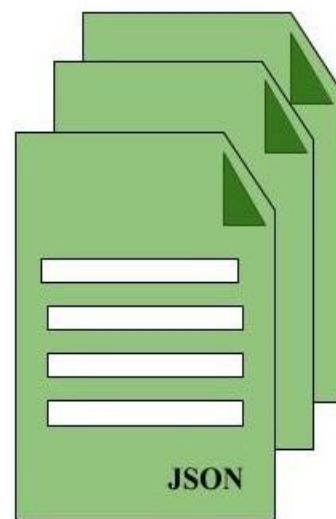- Storage retrieval and managing of documents is easy.

Limitations:

- Handling multiple documents is challenging

- Aggregation operations may not work accurately.

Examples:

- MongoDB

- CouchDB

Figure – Document Store Model in form of JSON documents

4. Graph Databases:
Clearly, this architecture pattern deals with storage and management of data in graphs. Graphs are basically structures that depict connections between two or more objects in some data.The objects or entities are called as nodes and are joined together by relationships called Edges. Each edge has a unique identifier. Each node serves as a point of contact for the graph.This pattern is very commonly used in social networks where there are a large number of entities and each entity has one or many characteristics which are connected by edges. The relational database pattern has tables which are loosely connected, whereas graphs are often very strong and rigid in nature.

Advantages:

- Fastest traversal because of connections.

- Spatial data can be easily handled.

Limitations:
Wrong connections may lead to infinite loops.

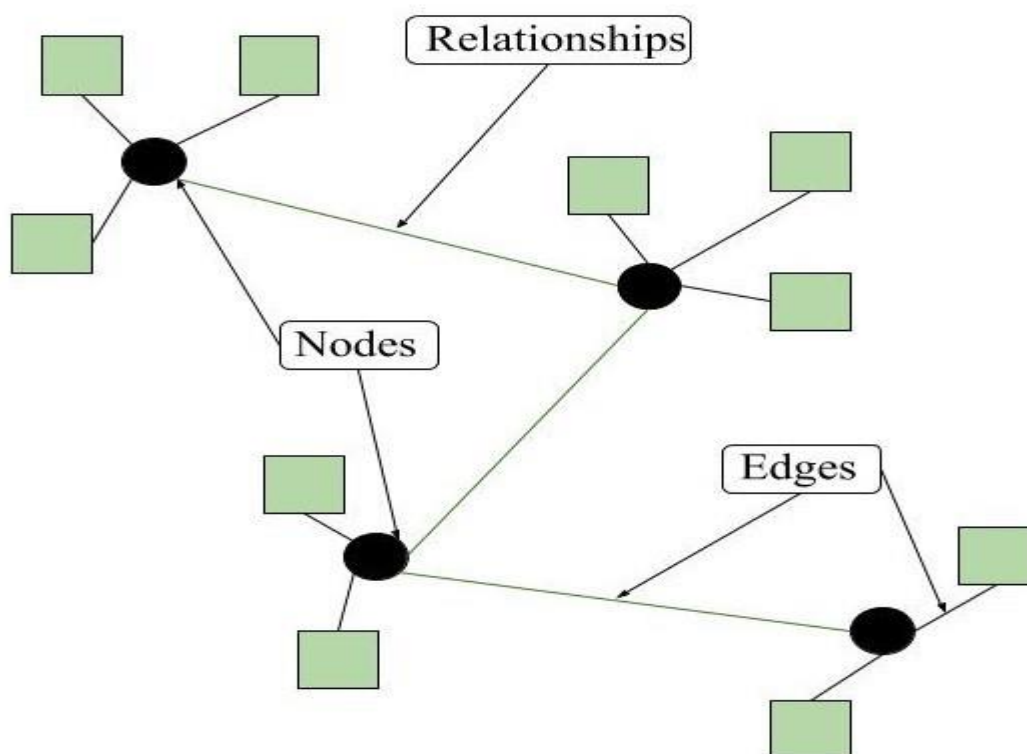Examples:

- Neo4J

- FlockDB(Used by Twitter)

Figure – Graph model format of NoSQL Databases