# UNIT-V (Part-2)

OVERVIEW OF NOSQL

- **NoSQL** Database is a non-relational Data Management System that does not require a fixed schema. It avoids joins, and is easy to scale.
- The major purpose of using a NoSQL database is for distributed data stores with humongous data storage needs.
- NoSQL is used for Big data and real-time web apps. For example, companies like Twitter, Facebook and Google collect terabytes of user data every single day.

→ **NoSQL database** stands for "Not Only SQL" or "Not SQL." Though a better term would be "NoREL", NoSQL caught on. Carl Strozz introduced the NoSQL concept in 1998.

- Traditional RDBMS uses SQL syntax to store and retrieve data for further insights. Instead, a NoSQL database system encompasses a wide range of database technologies that can store structured, semi-structured, unstructured and polymorphic data.
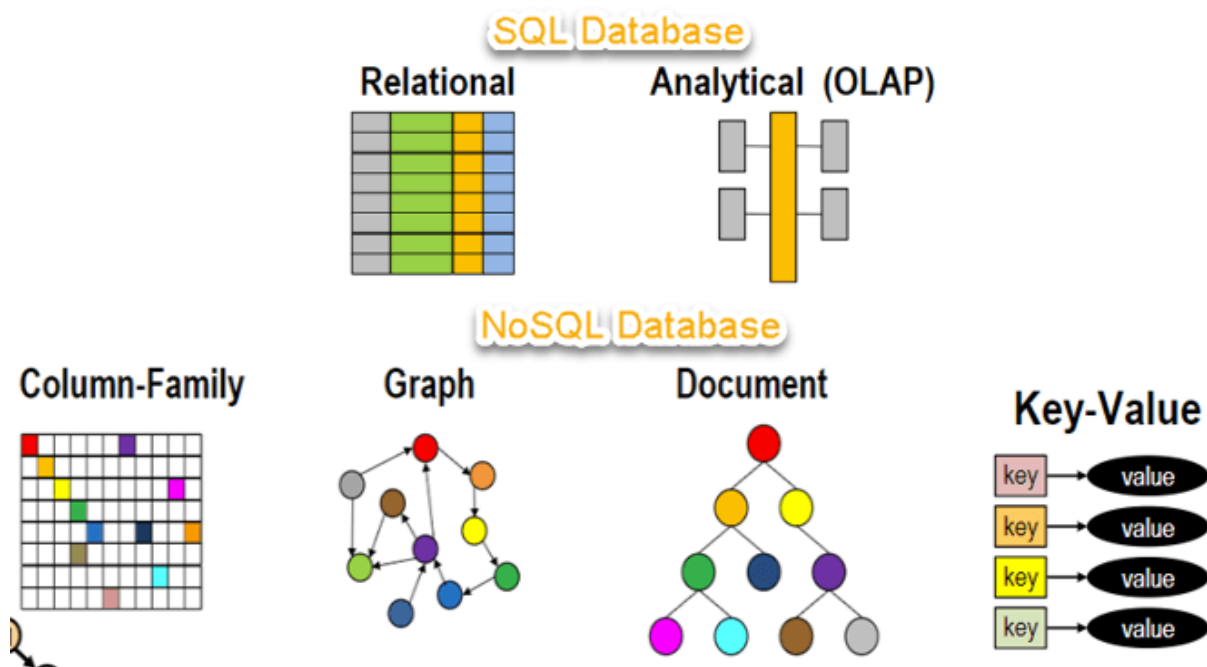
NEED OF NOSQL
- The concept of NoSQL databases became popular with Internet giants like Google, Facebook, Amazon, etc. who deal with huge volumes of data.
- The system response time becomes slow when you use RDBMS for massive volumes of data.

- To resolve this problem, we could "scale up" our systems by upgrading our existing hardware. This process is expensive.
- The alternative for this issue is to distribute database load on multiple hosts whenever the load increases. This method is known as "scaling out."
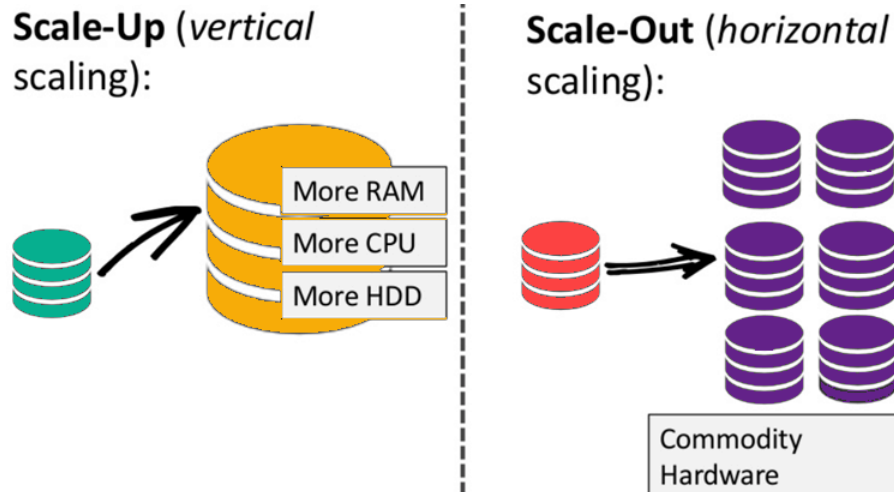
→ NoSQL databases are non-relational, so they scale out better than relational databases as they are designed with web applications in mind.

BRIEF HISTORY OF NOSQL DATABASES

- 1998- Carlo Strozzi use the term NoSQL for his lightweight, open-source relational database
- 2000- Graph database Neo4j is launched
- 2004- Google BigTable is launched
- 2005- CouchDB is launched
- 2007- The research paper on Amazon Dynamo is released
- 2008- Facebook's open sources the Cassandra project
- 2009- The term NoSQL was reintroduced

TYPES OF NOSQL DATABASES

NoSQL Databases are mainly categorized into four types:

- Key-value Pair Based
- Column-oriented Graph
- Graphs based
- Document-oriented

Every category has its unique attributes and limitations. None of the above-specified databases is better to solve all the problems. Users should select the database based on their product needs.

Key Value Pair Based

- Data is stored in key/value pairs. It is designed in such a way to handle lots of data and heavy load.
- Key-value pair storage databases store data as a hash table where each key is unique, and the value can be a JSON(JavaScript Object Notation), BLOB(Binary Large Objects), string, etc
- It is one of the most basic NoSQL database examples. This kind of NoSQL database is used as a collection, dictionaries, associative arrays, etc.
- Key value stores help the developer to store schema-less data. They work best for shopping cart contents.
- Redis, Dynamo, Riak are some NoSQL examples of key-value store DataBases.

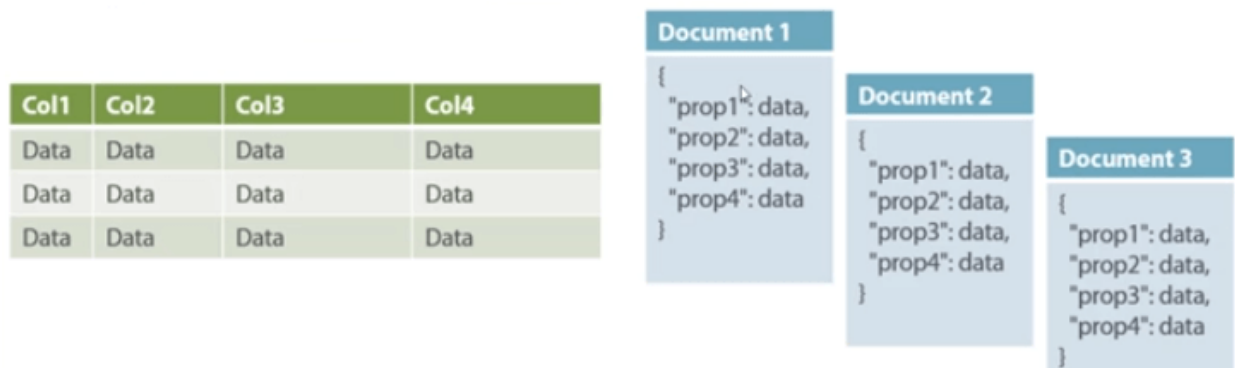| Key	Value
Name	Joe Bloggs
Age	42
Occupation	Stunt Double
Height	175cm
Weight	77kg

Column-based

- Column-oriented databases work on columns and are based on BigTable paper by Google. Every column is treated separately. Values of single column databases are stored contiguously.
- They deliver high performance on aggregation queries like SUM, COUNT, AVG, MIN etc. as the data is readily available in a column.
- Column-based NoSQL databases are widely used to manage data warehouses, business intelligence, CRM, Library card catalogs,
- HBase, Cassandra, HBase, Hypertable are NoSQL query examples of column based databases.

ColumnFamily			
Row Key	Column Name		
	Key	Key	Key
	Value	Value	Value
	Column Name		
	Key	Key	Key
	Value	Value	Value

Document-Oriented

- Document-Oriented NoSQL DB stores and retrieves data as a key value pair but the value part is stored as a document. The document is stored in JSON or XML formats. The value is understood by the DB and can be queried.
- The document type is mostly used for CMS systems, blogging platforms, real-time analytics & e-commerce applications. It should not be used for complex transactions which require multiple operations or queries against varying aggregate structures.
- Amazon SimpleDB, CouchDB, MongoDB, Riak, Lotus Notes, MongoDB, are popular Document originated DBMS systems.

Col1	Col2	Col3	Col4
Data	Data	Data	Data
Data	Data	Data	Data
Data	Data	Data	Data

Document 1
```
{
  "prop1": data,
  "prop2": data,
  "prop3": data,
  "prop4": data
}
```

Document 2
```
{
  "prop1": data,
  "prop2": data,
  "prop3": data,
  "prop4": data
}
```

Document 3
```
{
  "prop1": data,
  "prop2": data,
  "prop3": data,
  "prop4": data
}
```

Graph-Based

A graph type database stores entities as well the relations amongst those entities.
The entity is stored as a node with the relationship as edges.
An edge gives a relationship between nodes. Every node and edge has a unique identifier.
Compared to a relational database where tables are loosely connected, a Graph database is multi-relational in nature. Traversing relationships is fast as they are already captured into the DB, and there is no need to calculate them.
Graph based database mostly used for social networks, logistics, spatial data.
Neo4J, Infinite Graph, OrientDB, FlockDB are some popular graph-based databases.
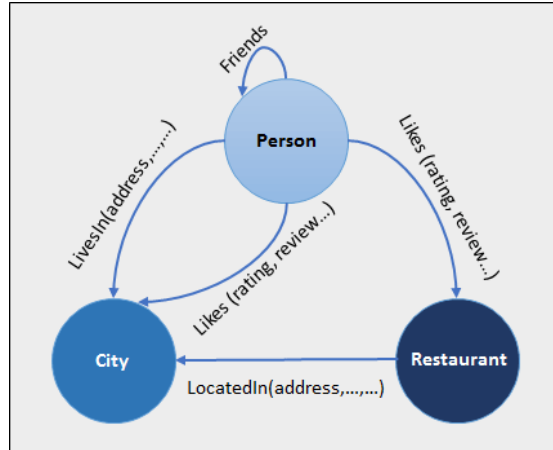
Differences between NoSQL and Relational database

NoSQL Database	Relational Database
NoSQL Database supports a very simple query language.	Relational Database supports a powerful query language.
NoSQL Database has no fixed schema.	Relational Database has a fixed schema.
NoSQL Database is only eventually consistent.	Relational Database follows acid properties. (Atomicity, Consistency, Isolation, and Durability)
NoSQL databases don't support transactions (support only simple transactions).	Relational Database supports transactions (also complex transactions with joins).
NoSQL Database is used to handle data coming in at high velocity.	Relational Database is used to handle data coming in low velocity.
Data arrives from many locations.	Data in relational databases arrive from one or few locations.
NoSQL databases can manage structured, unstructured and semi-structured data.	Relational database manages only structured data.
NoSQL databases have no single point of failure.	Relational databases have a single point of failure with failover.
NoSQL databases can handle big data or data in a very high volume .	NoSQL databases are used to handle moderate volumes of data.
NoSQL has a decentralized structure.	Relational database has a centralized structure.

NoSQL database gives both read and write scalability.	Relational databases give read scalability only.
NoSQL databases are deployed in horizontal fashion.	Relation database is deployed in vertical fashion.