



# SNS COLLEGE OF TECHNOLOGY

Re-accredited by NAAC with A+ grade, Accredited by NBA(CSE, IT, ECE, EEE & Mechanical)  
Approved by AICTE, New Delhi, Recognized by UGC, Affiliated to Anna University, Chennai



## Column Oriented Database



**COURSE** : 23CAT603- Database Management System

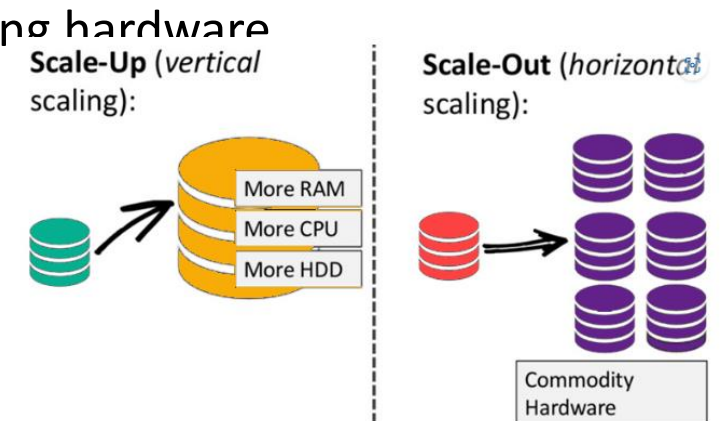
**UNIT V** : Column Oriented Database

**CLASS** : I Semester / I MCA



# Relational Database Management System

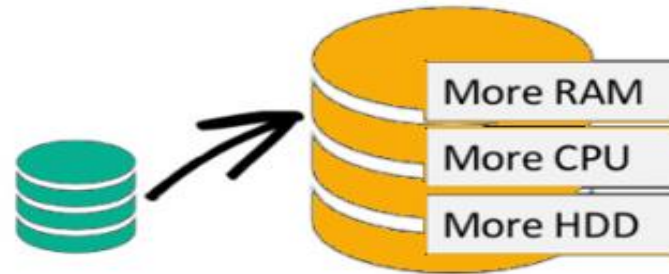
- ❑ Scaling Up Issues when dataset is too big
  - Multiple servers to host database
  - Expensive parallel databases, but not designed for OLTP
  - Master – slave architecture
- ❑ Not designed to be distributed
- ❑ Schema dependent – no flexibility to handle unstructured data
- ❑ Performance is matter when data volume grows
- ❑ “scale up” our systems by upgrading our existing hardware



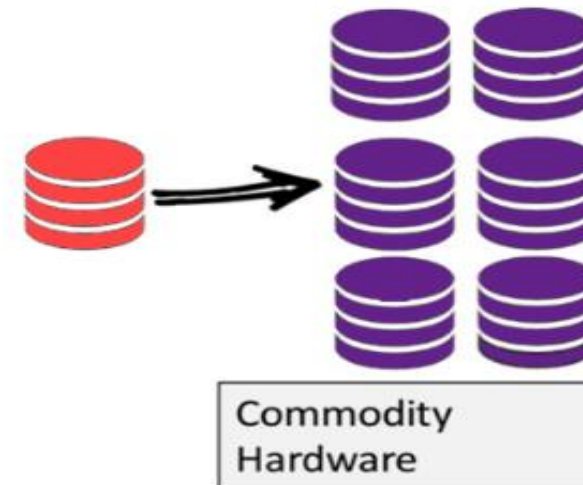


# Scale Up and Scale Out

**Scale-Up** (*vertical scaling*):



**Scale-Out** (*horizontal scaling*):

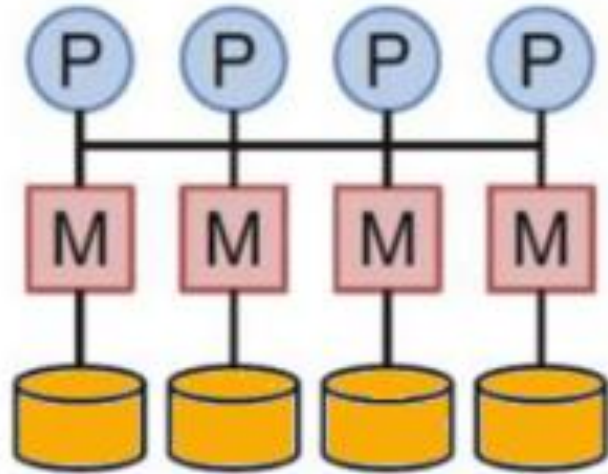


**Add more hardware resources**

**Add more servers in a distributed manner**

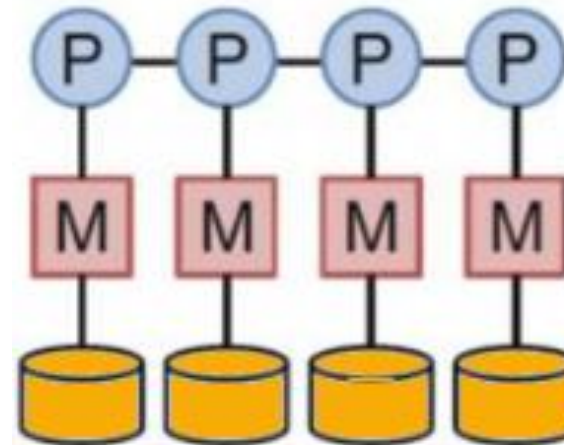


# RDBMS Vs NoSQL



**Shared Memory**  
e.g. "Oracle 11g"

|||||



**Shared Nothing**  
e.g. "NoSQL"



# NoSQL

- Stands for **Not Only SQL**
  - No relation
  - No database
- A class of non-relational storage system
- Doesn't require fixed schema
- Relaxation for one/more ACID properties using CAP theorem

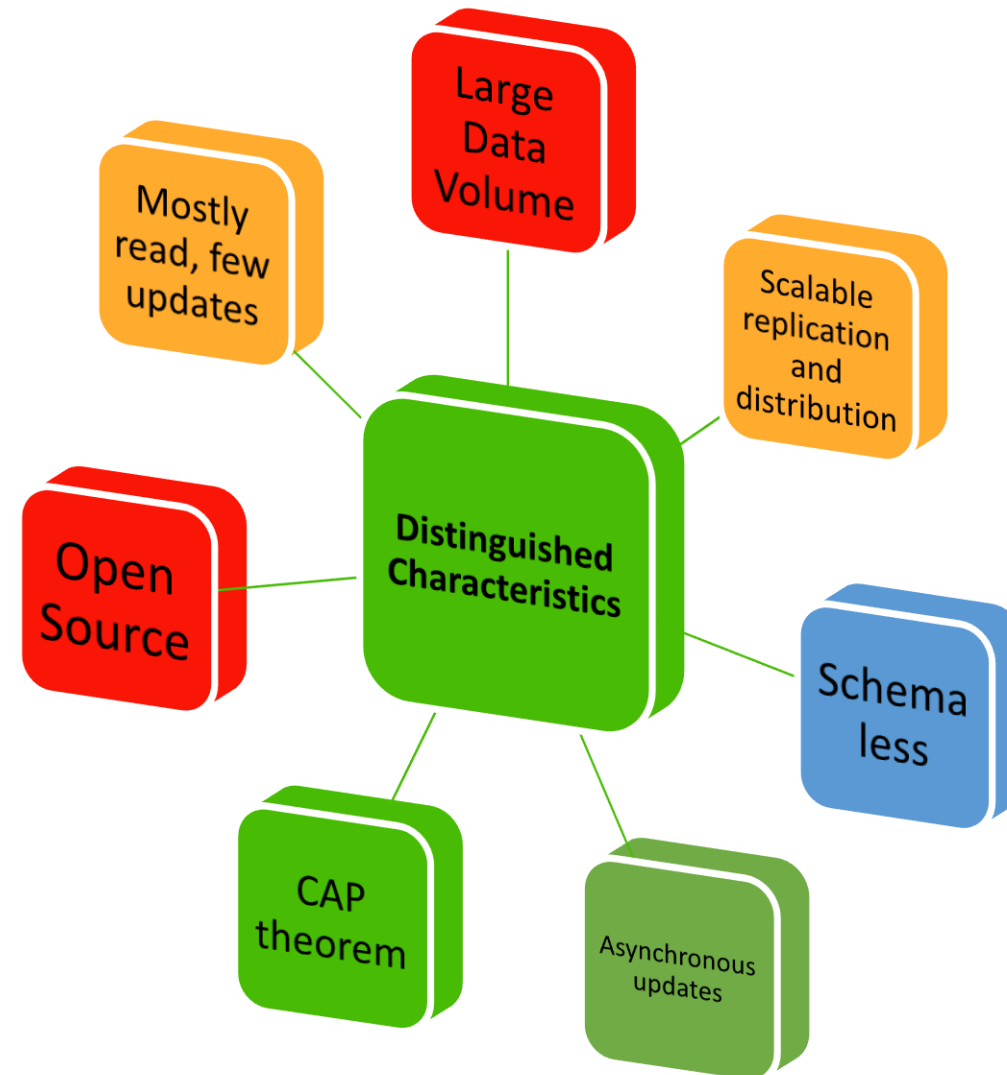
=====

## Why NoSQL?

- Explosion of social media network
- Explosion of large scale web services
- Rising of cloud based solutions
- Open source community
- Not required fixed schema

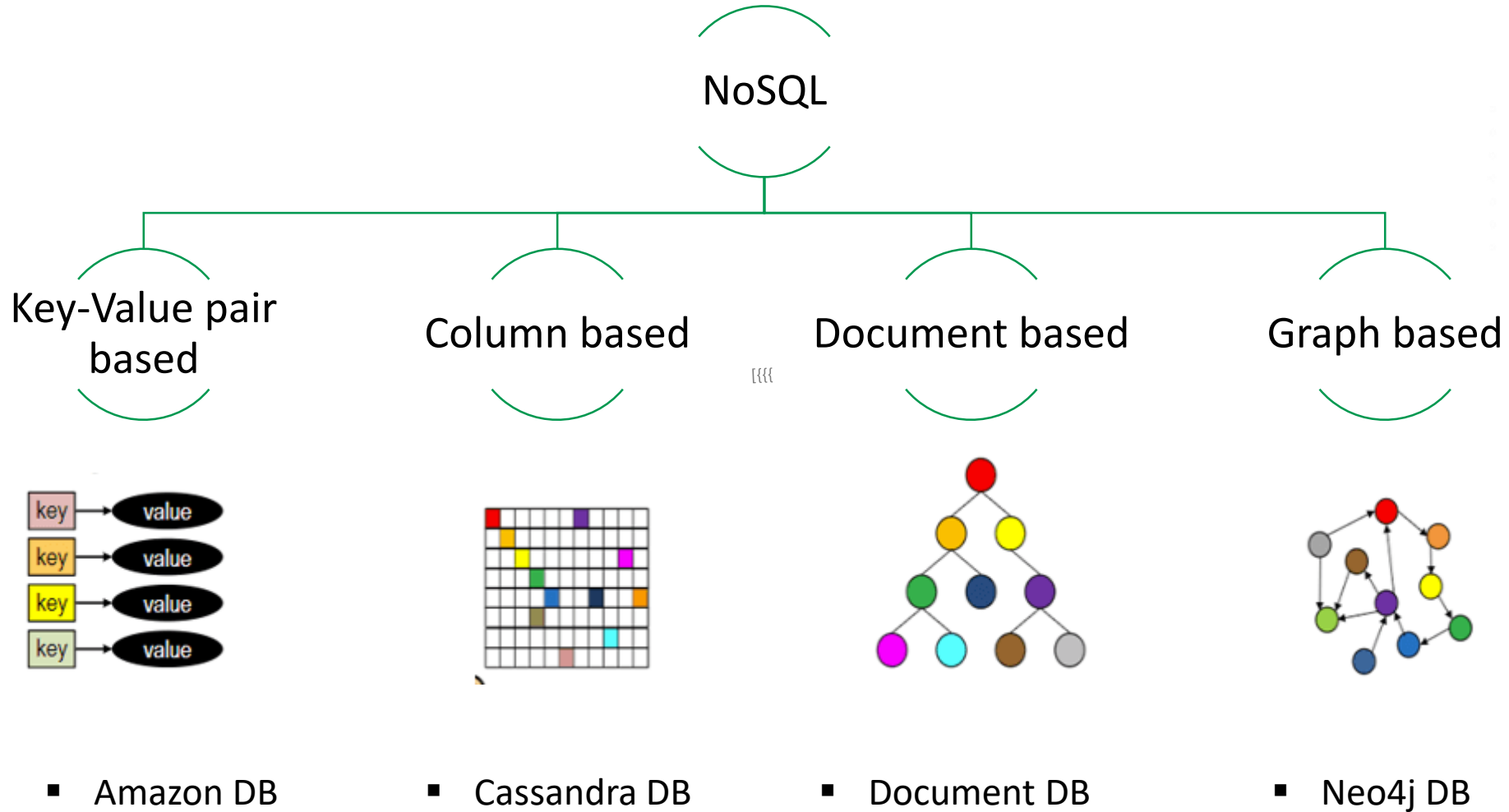


# NoSQL distinguished Characteristics



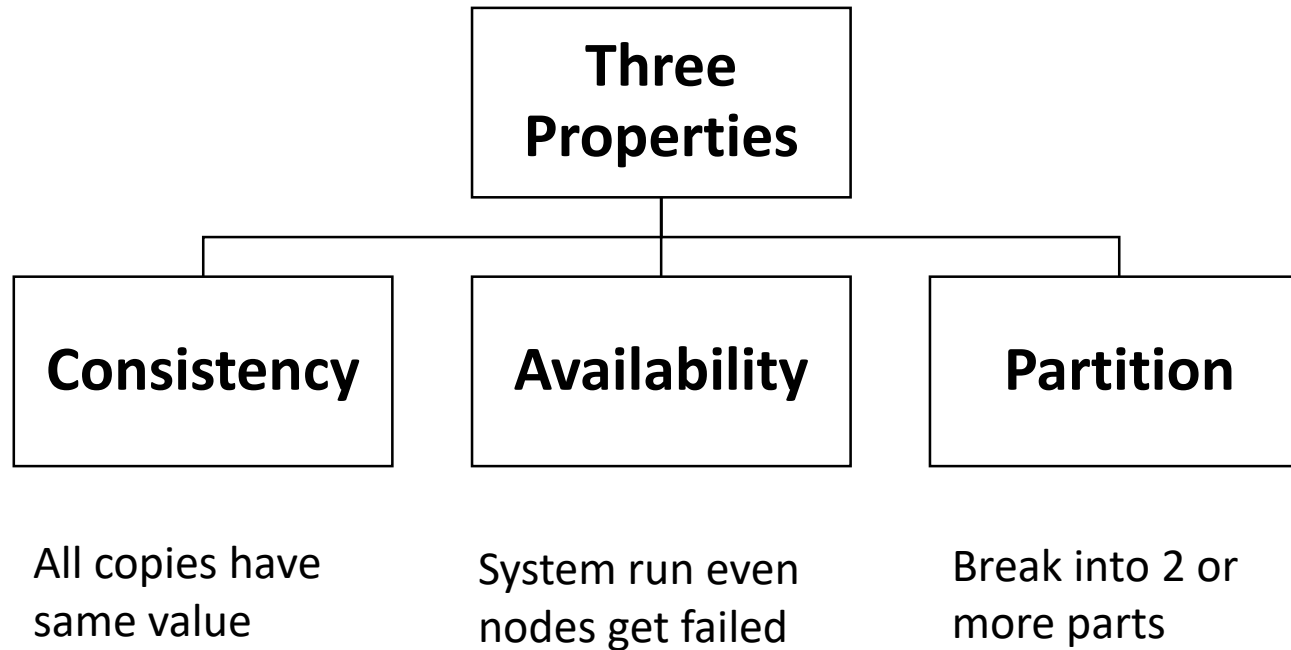


# NoSQL Types





# NoSQL distinguished Characteristics



- ❑ Theorem says at least 2 of 3 properties required for any system
  - Traditional DB choose consistency
  - Web apps choose availability





# Key-Value Pair NoSQL Data Pattern



- ❑ Stores and retrieves data as a key value pair but the value part is stored as a document.
- ❑ Document here can be a form of text, arrays, strings, JSON, XML or any such format
- ❑ Collections are the group of documents that store documents that have similar contents.
- ❑ The document is stored in JSON or XML formats.
- ❑ It is best for semi-structured data and storage retrieval/ managing of documents is easy

Key:1	ID:501
-------	--------

Key:2	ID:501	Name: Ragu
-------	--------	------------

Key:2	ID:501	Name: Ragu	Mail: rs@s.com
-------	--------	------------	----------------

**Amazon SimpleDB, CouchDB, MongoDB are popular NoSQL**



# Document oriented NoSQL Data Pattern



- ❑ Data is stored in 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.
- ❑ Key-value pair storage databases store data as a hash table where each key is unique, and the value can be a JSON, BLOB(Binary Large Objects), string, etc.
- ❑ It can handle large amounts of data and heavy load and easy retrieval of data by keys

```
Document 1
Id: "001",
Name: "Alex Bill",
Phone: "+001 234 5341",
Department: "Finance"
```

```
Document 2
Id: "003",
Name: "Alex Bill",
Phone: {
  Home: "+001 234 5341",
  Office: "+001 111 2759"
}
Department: "Finance"
```

**Redis, Dynamo, Riak are some NoSQL examples**



# Document oriented NoSQL Data Pattern



- ❑ Stores and retrieves data as a key value pair but the value part is stored as a document.
- ❑ Document here can be a form of text, arrays, strings, JSON, XML or any such format
- ❑ Collections are the group of documents that store documents that have similar contents.
- ❑ The document is stored in JSON or XML formats.
- ❑ It is best for semi-structured data and storage retrieval/ managing of documents is easy

## Document 1

```
Id: "001",  
Name: "Alex Bill",  
Phone: "+001 234 5341",  
Department: "Finance"
```

## Document 2

```
Id: "003",  
Name: "Alex Bill",  
Phone: {  
  Home: "+001 234 5341",  
  Office: "+001 111 2759"  
}  
Department: "Finance"
```

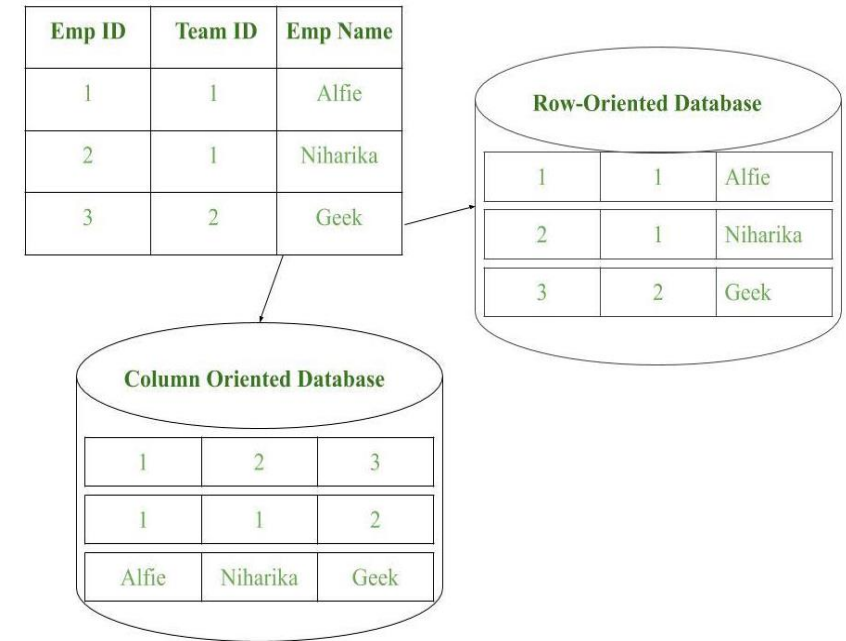
**Amazon SimpleDB, CouchDB,  
MongoDB are popular NoSQL**



# Column oriented NoSQL Data Pattern



- ❑ Data is stored in individual cells which are further grouped into columns. Column-oriented databases work only on columns
- ❑ 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

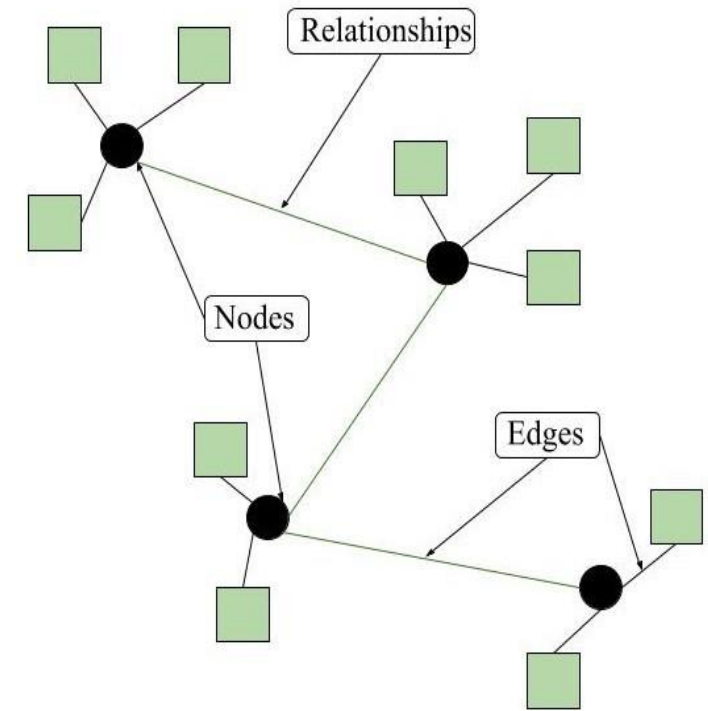


**HBase, Cassandra, HBase, Hypertable are NoSQL query examples of column based database**



# Graph Based NoSQL Data Pattern

- ❑ It deals with the 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 node serves as a point of contact for the graph.
- ❑ 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

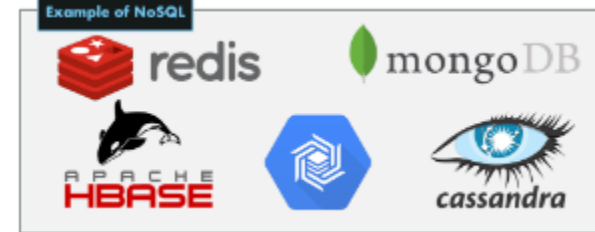


Neo4J, Infinite Graph, OrientDB are few databases



# SQL vs NoSQL

SQL	NoSQL
A relational database	A non-relational database
Needs a predefined schema for structured data	Have a dynamic schema for unstructured data
SQL databases are table based databases	NoSQL Databases are document / key-value pair/ graph/ column based
Better fit for complex queries	Not fit for complex queries
Vertically scalable	Horizontally scalable
Database based on ACID properties	Based on CAP Theorem










# NoSQL

Advantages	Disadvantages
<ul style="list-style-type: none"><li>● Handles big data</li><li>● Easy Replication</li><li>● It can handle structured, semi-structured, and unstructured data with equal effect</li><li>● It don't need a dedicated high-performance server</li><li>● It serve as the primary data source for online applications.</li><li>● Excels at distributed database and multi-data center operations</li><li>● Offers a flexible schema design which can easily be altered without downtime</li></ul>	<ul style="list-style-type: none"><li>● No standardization rules</li><li>● Limited query capabilities</li><li>● Doesn't work as well with relational data</li><li>● It does not offer any traditional database capabilities</li></ul>



# NoSQL Technologies

 <b>Relational</b>	 <b>Key/ Value</b>	 <b>Column Family</b>	 <b>Document</b>	 <b>Graph</b>
<ul style="list-style-type: none"><li>• Window Azure SQL Database</li><li>• SQL Server</li><li>• Oracle</li><li>• MySQL</li><li>• SQL Compact</li><li>• SQLite</li><li>• Postgres</li></ul>	<ul style="list-style-type: none"><li>• Window Azure Blob Storage</li><li>• Window Azure Table Storage</li><li>• Window Azure Cache</li><li>• Redis</li><li>• Memcached</li><li>• Risk</li></ul>	<ul style="list-style-type: none"><li>• Cassandra</li><li>• HBase</li><li>• This is a sample text that you can edit.</li></ul>	<ul style="list-style-type: none"><li>• MongoDB</li><li>• RavenDB</li><li>• CouchDB</li><li>• This is a sample text that you can edit.</li></ul>	<ul style="list-style-type: none"><li>• Neo4J</li><li>• This is a sample text that you can edit.</li><li>• This is a sample text that you can edit.</li></ul>





## References

- <https://www.tutorialspoint.com/NoSQL-Databases>
- <https://www.geeksforgeeks.org/nosql-data-architecture-patterns>
- <https://www.mongodb.com/nosql-explained>
- Shannon Bradshaw, Eoin Brazil, and Kristina Chodorow, “MongoDB: the Definitive Guide”, O’Reilly Media, 3<sup>rd</sup> Edition

