**Branch :- Computer Sci. & Engg. Class :- III Year**

**Subject :- DBMS Sem :- V**

**Teacher Manual**

**PRACTICAL NO 9**

**AIM**: Study of Open Source NoSQL Databases

**Theory:-** NoSQL databases (aka "not only SQL") are non-tabular databases and store data differently than relational tables. NoSQL databases come in a variety of types based on their data model. The main types are document, key-value, wide-column, and graph. They provide flexible schemas and scale easily with large amounts of data and high user loads.

**What is a NoSQL database?**

A NoSQL originally referring to non SQL or non relational is a database that provides a mechanism for storage and retrieval of data. This data is modeled in means other than the tabular relations used in relational databases. Such databases came into existence in the late 1960s, but did not obtain the NoSQL moniker until a surge of popularity in the early twenty-first century. NoSQL databases are used in real-time web applications and big data and their use are increasing over time. NoSQL systems are also sometimes called Not only SQL to emphasize the fact that they may support SQL-like query languages.A NoSQL originally referring to non SQL or non relational is a database that provides a mechanism for storage and retrieval of data. This data is modeled in means other than the tabular relations used in relational databases. Such databases came into existence in the late 1960s, but did not obtain the NoSQL moniker until a surge of popularity in the early twenty-first century. NoSQL databases are used in real-time web applications and big data and their use are increasing over time. NoSQL systems are also sometimes called Not only SQL to emphasize the fact that they may support SQL-like query languages.

**Differences between SQL and NoSQL**

The table below summarizes the main differences between SQL and NoSQL databases.

| **Sr.No.** | **Key Points** | **SQL Databases** | **NoSQL Databases** |
| --- | --- | --- | --- |
| 1 | Data Storage Model | Tables with fixed rows and columns | Document: JSON documents, Key-value: key-value pairs, Wide-column: tables with rows and dynamic columns, Graph: nodes and edges |
| 2 | Development History | Developed in the 1970s with a focus on reducing data duplication | Developed in the late 2000s with a focus on scaling and allowing for rapid application change driven by agile and DevOps practices. |
| 3 | Examples | Oracle, MySQL, Microsoft SQL Server, and PostgreSQL | Document: MongoDB and CouchDB, Key-value: Redis and DynamoDB, Wide-column: Cassandra and HBase, Graph: Neo4j and Amazon Neptune |
| 4 | Primary Purpose | General purpose | Document: general purpose, Key-value: large amounts of data with simple lookup queries, Wide-column: large amounts of data with predictable query patterns, Graph: analyzing and traversing relationships between connected data |
| 5 | Schemas | Rigid | Flexible |
| 6 | Scaling | Vertical (scale-up with a larger server) | Horizontal (scale-out across commodity servers) |
| 7 | Multi-Record ACID Transactions | Supported | Most do not support multi-record ACID transactions. However, some—like MongoDB—do. |
| 8 | Joins | Typically required | Typically not required |
| 9 | Data to Object Mapping | Requires ORM (object-relational mapping) | Many do not require ORMs. MongoDB documents map directly to data structures in most popular programming languages. |

**Advantages of NoSQL:**  
There are many advantages of working with NoSQL databases such as MongoDB and Cassandra. The main advantages are high scalability and high availability.

* High scalability –  
   NoSQL database use sharding for horizontal scaling. Partitioning of data and placing it on multiple machines in such a way that the order of the data is preserved is sharding. Vertical scaling means adding more resources to the existing machine whereas horizontal scaling means adding more machines to handle the data. Vertical scaling is not that easy to implement but horizontal scaling is easy to implement. Examples of horizontal scaling databases are MongoDB, Cassandra etc. NoSQL can handle huge amount of data because of scalability, as the data grows NoSQL scale itself to handle that data in efficient manner.
* High availability –  
   Auto replication feature in NoSQL databases makes it highly available because in case of any failure data replicates itself to the previous consistent state.

**Disadvantages of NoSQL:**NoSQL has the following disadvantages.

* Narrow focus –  
   NoSQL databases have very narrow focus as it is mainly designed for storage but it provides very little functionality. Relational databases are a better choice in the field of Transaction Management than NoSQL.
* Open-source –  
   NoSQL is open-source database. There is no reliable standard for NoSQL yet. In other words two database systems are likely to be unequal.
* Management challenge –  
   The purpose of big data tools is to make management of a large amount of data as simple as possible. But it is not so easy. Data management in NoSQL is much more complex than a relational database. NoSQL, in particular, has a reputation for being challenging to install and even more hectic to manage on a daily basis.
* GUI is not available –  
   GUI mode tools to access the database is not flexibly available in the market.
* Backup –  
   Backup is a great weak point for some NoSQL databases like MongoDB. MongoDB has no approach for the backup of data in a consistent manner.
* Large document size –  
   Some database systems like MongoDB and CouchDB store data in JSON format. Which means that documents are quite large (BigData, network bandwidth, speed), and having descriptive key names actually hurts, since they increase the document size.

**Types of NoSQL database:**  
Types of NoSQL databases and the name of the databases system that falls in that category are:

MongoDB falls in the category of NoSQL document based database.

1)Key value store: Memcached, Redis, Coherence

2)Tabular: Hbase, Big Table, Accumulo

3)Document based: MongoDB, CouchDB, Cloudant

**When should NoSQL be used:**

* When huge amount of data need to be stored and retrieved .
* The relationship between the data you store is not that important
* The data changing over time and is not structured.
* Support of Constraints and Joins is not required at database level
* The data is growing continuously and you need to scale the database regular to handle the data.

**CONCLUSION:**  Thus we have studied Open Source NoSQL Databases.