# NOSQL Database Assignment

## ❖ NOSQL V/S SQL Databases

- **Introduction:**

Databases are essential components of modern software systems, storing and organizing data for efficient retrieval, update, and management. Broadly, databases are categorized into two types: SQL (Structured Query Language) databases and NoSQL (Not Only SQL) databases. Choosing the right database type depends on the application's nature, data structure, scalability needs, and performance requirements.

- **SQL Databases (Relational Databases):**

SQL databases are relational databases that store data in tables (rows and columns). Relationships between tables are established using primary and foreign keys.

- **Examples:** MySQL, PostgreSQL, Oracle Database, Microsoft SQL Server
- **Key Features:**

1. Structured Schema: Tables with fixed columns and predefined data types.
2. ACID Compliance: Ensures Atomicity, Consistency, Isolation, and Durability.
3. Joins: Supports complex queries with JOINs to combine data from multiple tables.
4. Relational Integrity: Enforced through keys and constraints.
5. Standard Query Language: Uses SQL for querying and data manipulation.

- **Use Cases:**

1. Financial applications
2. ERP systems
3. Applications requiring strong data consistency and transactions

- **NoSQL Databases (Non-Relational Databases):**

NoSQL databases are non-relational and store data in a variety of formats: key-value pairs, documents, graphs, or wide-columns. They are designed for scalability, flexibility, and handling large volumes of unstructured or semi-structured data.

- **Types of NoSQL Databases:**

1. Document-Based (e.g., MongoDB)
2. Key-Value Store (e.g., Redis)
3. Column-Oriented (e.g., Cassandra)
4. Graph-Based (e.g., Neo4j)

- **Key Features**

1. Schema-less: No rigid schema; fields can vary across documents.

2. Horizontal Scalability: Easily scales out across servers.
3. High Performance: Optimized for read/write operations.
4. Flexible Data Models: Can handle structured, semi-structured, or unstructured data.
5. Eventual Consistency: Prioritizes availability and partition tolerance over consistency (CAP theorem).

- **Use Cases**

1. Big Data applications
2. Real-time analytics
3. IoT applications

| Aspect | SQL (Relational) | NoSQL (Non-relational) |
|---|---|---|
| **Data Structure** | Tables with rows and columns | Document-based, key-value, column-family, or graph-based |
| **Schema** | Fixed schema (predefined structure) | Flexible schema (dynamic and adaptable) |
| **Scalability** | Vertically scalable (upgrading hardware) | Horizontally scalable (adding more servers) |
| **Data Integrity** | ACID-compliant (strong consistency) | BASE-compliant (more available, less consistent) |
| **Query Language** | SQL (Structured Query Language) | Varies (e.g., MongoDB uses its own query language) |
| **Performance** | Efficient for complex queries and transactions | Better for large-scale data and fast read/write operations |
| **Use Case** | Best for transactional systems (banking, ERP, etc.) | Ideal for big data, real-time web apps, and data lakes |

| Aspect | SQL (Relational) | NoSQL (Non-relational) |
|---|---|---|
| **Examples** | MySQL, PostgreSQL, Oracle, MS SQL Server | MongoDB, Cassandra, CouchDB, Neo4j |

❖ **Features of MongoDB:**

MongoDB is one of the most popular NoSQL document-based databases, developed by MongoDB Inc. It stores data in flexible, JSON-like documents and is designed for scalability, high availability, and developer agility.

MongoDB's database structure consists of the following components:

| Component | Equivalent RDBMS | Description |
|---|---|---|
| **Database** | Database | Stores multiple collections. |
| **Collection** | Table | Groups related documents. |
| **Document** | Row | A BSON object containing key-value pairs. |
| **Field** | Column | Stores data attributes within documents. |

MongoDB offers a wide range of features that make it a preferred choice for modern applications.

**1. Schema-less Database**

Unlike traditional relational databases, MongoDB collections:

- Allow different structures within the same collection.
- Do not require fixed column definitions.
- Enable easy updates and modifications.

**2. Document Oriented**

In MongoDB, all the data stored in the documents instead of tables like in RDBMS. In these documents, the data is stored in fields(key-value pair) instead of rows and columns which

make the data much more flexible in comparison to RDBMS. And each document contains its unique object id.

**3. Indexing**

In MongoDB database, every field in the documents is indexed with primary and secondary indices this makes easier and takes less time to get or search data from the pool of the data. If the data is not indexed, then database search each document with the specified query which takes lots of time and not so efficient.

**4. Scalability**

MongoDB provides horizontal scalability with the help of sharding. Sharding means to distribute data on multiple servers, here a large amount of data is partitioned into data chunks using the shard key, and these data chunks are evenly distributed across shards that reside across many physical servers. It will also add new machines to a running database.

**5. Replication**

MongoDB provides high availability and redundancy with the help of replication, it creates multiple copies of the data and sends these copies to a different server so that if one server fails, then the data is retrieved from another server.

**6. Aggregation**

It allows to perform operations on the grouped data and get a single result or computed result. It is similar to the SQL GROUPBY clause. It provides three different aggregations i.e, aggregation pipeline, map-reduce function, and single-purpose aggregation methods

**7. High Performance**

The performance of MongoDB is very high and data persistence as compared to another database due to its features like scalability, indexing, replication, etc.