



## **Data Base Systems I**

### **Topic:**

Comparison between SQL and NoSQL

### **Name:**

Aoun-Haider

### **ID:**

FA21-BSE-133

### **Assignment:**

02

### **Submitted to:**

Sir Abdul Qayyum

### **Date:**

23-05-2023

## Contents

Abstract .....	03
Introduction .....	03
Data Model .....	03
Query Language .....	03
Scalability .....	03
Data Consistency and Transactions .....	04
Schema Flexibility .....	04
Use Cases .....	04
Comparison .....	04
Conclusion .....	05

## **Abstract:**

This report provides a detailed analysis of the differences between SQL-based databases and NoSQL-based databases. It explores key aspects such as data model, query language, scalability, data consistency and transactions, schema flexibility, and use cases. The report aims to assist readers in understanding the characteristics and advantages of each database category, enabling informed decision-making when choosing the appropriate database management system for specific applications.

## **Introduction:**

In today's data-driven world, choosing the right database management system is crucial for efficient and effective data storage, retrieval, and management. SQL-based databases and NoSQL-based databases represent two distinct approaches to data management, each with its own set of strengths and limitations. This report aims to provide a comprehensive comparison between these two types of databases, highlighting their divergent features and use cases.

## **Data Model:**

SQL databases are based on the relational data model, utilizing structured tables with predefined schemas, while NoSQL databases offer various data models, such as key-value, document, columnar, and graph, providing flexibility for storing unstructured and polymorphic data.

## **Query Language:**

SQL databases employ the Structured Query Language (SQL), allowing complex querying, joins, aggregations, and transactions, while NoSQL databases often employ their own query languages or APIs tailored to specific data models, focusing on simple CRUD operations and basic querying capabilities.

## **Scalability:**

SQL databases typically scale vertically by increasing hardware resources, whereas NoSQL databases are designed for horizontal scalability, distributing data across multiple servers to handle large datasets and high traffic loads.

### **Data Consistency and Transactions:**

SQL databases prioritize strong data consistency and support transactions adhering to the ACID properties, while NoSQL databases may sacrifice strong consistency for scalability, often offering eventual consistency and tunable consistency models.

### **Schema Flexibility:**

SQL databases enforce rigid schemas with fixed columns and data types, necessitating alterations to the table structure for schema modifications, while NoSQL databases provide schema flexibility, allowing dynamic changes without disrupting existing data.

### **Use Cases:**

SQL databases are well-suited for applications requiring complex querying, structured data, and transactions, such as financial systems and e-commerce platforms. NoSQL databases excel in scenarios involving large-scale data ingestion, high write throughput, real-time analytics, and unstructured or semi-structured data, commonly found in social networks, IoT platforms, and recommendation engines.

### **Comparison:**

<b>SQL</b>	<b>NoSQL</b>
Stands for 'Structured Query Language'	Stands for 'Not only SQL'
Relational Database Management System	Non-relational Database System
Data is stored in structured format like table.	Data can be stored in unstructured format like graph or document.
Supports complex joins and relationship	No join or relationships, typically denormalized data
Limited scalability for large data	High scalability for large data
Less flexible with fixed schemas and rigid structure	More flexible with dynamic schemas and structure
Supports for multi-row transactions	Limited or no support for multi-row transactions
Vertically scaled	Horizontally scaled
Strict schema enforcement	Flexible or schema less

<b>For example, MySQL, Oracle, PostgreSQL</b>	<b>For example, MongoDB, Cassandra, Redis</b>
---	---

### **Conclusion:**

In conclusion, SQL-based databases and NoSQL-based databases offer distinct approaches to data management, each with its own strengths and suitability for specific use cases. SQL databases provide strong consistency, complex querying capabilities, and are ideal for structured data and transactional applications. On the other hand, NoSQL databases offer scalability, flexibility, and handle unstructured data efficiently, making them suitable for high-performance applications dealing with large volumes of data. Choosing the appropriate database management system depends on factors such as data structure, scalability requirements, consistency needs, and the nature of the application.