IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 5, Issue 1, January 2025

A Study of SQL and NOSQL A Comparative Analysis

Sagar Laxman Bhor, Omkar Sampat Naykodi, Dipak Vilas Shelkande,
Prof. Vivek Vinayak Ghangale Ganesh Govind Bhor
Shankarrao Butte Patil B.Sc. IT College, Junnar, Maharashtra, India

Abstract: *This paper provides a comprehensive comparative analysis of SQL and NoSQL database technologies. SQL databases, characterized by their structured schema and ACID compliance, are ideal for relational data and complex queries. In contrast, NoSQL databases prioritize scalability, flexibility, and performance for unstructured and semi-structured data. The study aims to highlight their key features, use cases, and trade-offs, helping developers and organizations choose the most suitable database technology for their specific needs. This comparative study draws insights from technical journals, books, and online resources to provide a balanced understanding of both database models.*

Keywords: Databases, Commands, Structured Query, Access Control, scalability, ACID compliance

I. INTRODUCTION

Database technologies are essential tools for developers, and understanding their differences is key to building efficient applications. For this research, the team focused on two widely adopted models: SQL and NoSQL. These technologies have gained significant popularity in the industry due to their adaptability and performance. SQL databases are well-known for their structured approach and reliability, while NoSQL databases excel in handling unstructured data and scalability. For beginners, important qualities of a database system include user-friendliness, flexibility, and clear documentation. The team collected insights from diverse sources such as technical journals, books, and online platforms. Below is an overview of these two database types.

1.1 OVERVIEW OF SQL: [1]

SQL stands for Structured Query Language. IBM developed the original version of SQL, originally called SEQUEL, as part of the System R project in the early 1970s. The SEQUEL language has evolved since then, and its name has changed to SQL (Structured Query Language). Many products now support the SQL language. SQL has clearly established itself as the standard relational database language. In 1986, the American National Standards Institute (ANSI) and the International Organization for Standardization (ISO) published an SQL standard, called SQL-86. The basic structure of an SQL query consists of three clauses: SELECT, FROM, WHERE.ANSI published an extended standard for SQL, SQL-89, in 1989. The next version was SQL-92, followed by SQL:1999. The most recent version is SQL:2003.

1.2 FEATURE OF SQL:

- Data Definition Language (DDL)
- Data Manipulation Language (DML)
- Integrity
- View definition
- Transaction control
- Embedded SQL and dynamic SQL
- Authorization

1.3 KEY FUNCTIONS OF SQL:[2]

1. Data Definition: Defines the structure of the database.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 5, Issue 1, January 2025

2. Data Retrieval: Extracts data from the database.
3. Data Manipulation: Handles operations like inserting, updating, and deleting data.
4. Access Control: Manages permissions and access rights for data.

1.4 WHAT SQL DOES:

SQL is used to store, retrieve, manipulate, and manage data within a database management system (DBMS).
SQL allows for the organization of data in a tabular format (with rows and columns), facilitating the establishment of relationships between different tables.

1.5 COMMANDS IN SQL:

- Data Definition Language (DDL): Manages the structure of the database (e.g., creating or tables and schemas).
- Data Manipulation Language (DML): Handles operations like inserting, updating, and deleting data.
- Data Control Language (DCL):
- Controls access to the database using commands like GRANT and REVOKE.
- Data Query Language (DQL):
- Executes queries to retrieve data from the database (e.g., using the SELECT statement).
- Transaction Control Language (TCL):
- Manages transaction-related operations like COMMIT, ROLLBACK, and SAVEPOINT to ensure data integrity

1.6 Use Case of SQL: [3.1]

- Data Science
- Marketing
- Finance
- Healthcare
- Cybersecurity
- Social Media
- Entertainment Industry [3.3].
- Transactional Applications: SQL databases are well-suited for transactional applications that require strong data consistency and ACID compliance, such as banking systems, e-commerce platforms, and ERP (Enterprise Resource Planning) systems
- Reporting and Analytics: SQL databases are commonly used for reporting and analytics applications that involve complex queries, aggregations, and data joins Their support for SQL queries makes them suitable for generating insights from structured datasets

1.7 Notable Developers of SQL Systems:[4]

- **SQL (Structured Query Language):** Developed by Donald D. Chamberlin and Raymond F. Boyce at IBM in the early 1970s.
- **Oracle Database:** Founded by Larry Ellison, Bob Miner, and Ed Oates in 1977.
- **MySQL:** Developed by Michael Widenius and David Axmark in 1995.
- **PostgreSQL:** Originated from the POSTGRES project led by Michael Stonebraker at the University of California, Berkeley, starting in 1986.
- **Microsoft SQL Server:** Developed by Microsoft, with its first version released in 1989.

1.8 Advantages and Disadvantages of SQL:[5]

Advantages:

- Efficient data retrieval

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 5, Issue 1, January 2025

- Data integrity
- Scalability
- Standardization
- Flexibility

Disadvantages:

- Complexity
- Security risks
- Performance issues
- Cost
- Lack of flexibility

1.8 Overview of NoSQL :[6]

NoSQL, which stands for "Not Only SQL," is a category of database systems that eschews the traditional relational model in favor of more flexible, scalable, and schema-less
approaches. These databases are optimized for handling large volumes of data, high user loads, and dynamic or unstructured data types.

- Origins in the 1960s and 1970s:Early database systems like IBM's IMS and Codasyl supported non-relational models. These systems were primarily hierarchical or network-based and catered to specific data management needs before the dominance of relational databases.
- Rise of Relational Databases in the 1980s: Relational databases became the standard for data management due to their structured query language (SQL) and ACID properties, which ensured consistency and reliability.
- Re-emergence of Non-Relational Models in the 2000s: As web applications grew, relational databases faced challenges with scalability and flexibility. Companies like Google (with Bigtable) and Amazon (with Dynamo) developed distributed storage systems that could handle massive amounts of unstructured data efficiently.
- Coining the Term NoSQL: Johan Oskarsson popularized the term "NoSQL" in 2009 to describe a new wave of databases designed for large-scale web applications. These databases prioritized scalability, flexibility, and performance over strict adherence to relational models.
- Modern NoSQL Databases: The book describes the development of popular NoSQL databases like MongoDB, Cassandra, and CouchDB, highlighting their ability to address specific use cases such as document storage, key-value pair handling, and distributed data management.

1.9 Types of NoSQL

- **Key-Value Store**: It is a Hash Table with keys and values. The keys are user-defined and these keys point to the corresponding values stored in them. The keys are unique identifiers of the values stored in them.
- **Document Store**: It is a document made up of tagged elements. An example is an XML file where the data is stored in the form of tags.
- **Column-family Store**: Each storage block contains data from only one column. That means one block stores all values from one column only, other block stores values from other column and so on.
- **Graph Store**: It is a network of database that has links (edges) and nodes to represent the stored data.

2.1 Use Case of NoSQL:

- Social Media Applications: Handling dynamic and unstructured data, e.g., Facebook, Twitter.
- E-commerce: Managing user preferences and catalog data, e.g., Amazon
- Real-Time Analytics: For applications requiring quick, high-volume data ingestion, example- stock market data.
- IoT Systems: Managing sensor data and logs.
- Big Data and Real-Time Analytics: NoSQL databases are ideal for big data and real- time analytics applications that require scalable

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 5, Issue 1, January 2025

- Content Management and Personalization: NoSQL databases are well-suited for content management systems, recommendation engines, and personalization platforms that handle semi-structured or unstructured data, such as user profiles, preferences, and interactions.

2.2 Feature of NoSQL:[8]

- Distributed Computing
- Scaling
- Flexible Schemas and Rich Query Language
- Ease of use for Developers
- Partition tolerance
- High Availability

2.3 Notable Developers of NoSQL Systems:[9]

- MongoDB: Developed by Dwight Merriman and Eliot Horowitz in 2007.
- Cassandra: Originally developed at Facebook and later open-sourced by the Apache Software Foundation.
- Redis: Created by Salvatore Sanfilippo in 2009.
- HBase: Developed as part of the Apache Hadoop ecosystem.

2.4 Differences Between SQL and NOSQL:[10]

SQL	NOSQL
Relational Database Management System (RDBMS)	Non-relational or distributed database system.
These databases have fixed or static or predefined schema	They have a dynamic schema
These databases are not suited for hierarchical data storage.	These databases are best suited for hierarchical data storage.
These databases are best suited for complex queries	These databases are not so good for complex queries
Vertically Scalable	Horizontally scalable
Follows ACID property	Follows CAP (consistency, availability, partition tolerance)
Examples: MySQL, PostgreSQL ,Oracle, MS-SQL Server, etc	Examples: MongoDB, HBase, Neo4j, Cassandra, etc

2.6 Advantages and Disadvantages of NoSQL:[11]

Advantages:

- Flexible Data Structures
- Scalability
- High Performance
- Availability
- Cost-Effective

Disadvantages:

- Limited Query Capability
- Data Consistency
- Lack of Standardization
- Limited Tooling
- Limited ACID Compliance

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 5, Issue 1, January 2025

II. DISCUSSION

The choice between SQL and NoSQL databases depends on the specific requirements of a project. SQL databases offer a structured and reliable approach, making them suitable for applications requiring complex queries and strict data integrity, such as banking systems and ERP platforms. Their standardization and support for advanced querying tools further strengthen their use in reporting and analytics. On the other hand, NoSQL databases are designed to handle the challenges of modern web applications, including scalability, distributed computing, and flexible data structures. These databases excel in scenarios like social media, IoT, and big data analytics, where the data is often unstructured and rapidly evolving.

It is also important to consider the trade-offs. While SQL ensures consistency and reliability, it may face challenges in scaling horizontally. Conversely, NoSQL's flexibility and scalability come at the cost of weaker consistency guarantees in certain cases. This discussion underlines that understanding the strengths and limitations of each database type is critical for selecting the right tool for a given application.

III. CONCULSION

Both SQL and NoSQL databases play crucial roles in the modern technological landscape, with each offering unique strengths and addressing specific challenges. SQL databases continue to dominate fields requiring structured data, complex queries, and robust data integrity. In contrast, NoSQL databases provide the flexibility and scalability needed for big data, real-time applications, and systems with dynamic data models. This comparative analysis emphasizes the importance of aligning database choice with the application's needs, ensuring optimal performance, scalability, and reliability. Future developments in database technologies may further blur the lines between these two models, providing even more versatile solutions.

IV. ACKNOWLEDGMENT

The authors wish to express their gratitude to the developers and pioneers of both SQL and NoSQL systems, whose innovations have transformed the way data is managed and utilized.

We also acknowledge the contributions of researchers, educators, and practitioners who provided valuable insights through their work in technical journals, books, and online platforms. Special thanks to our peers and mentors for their feedback and support in refining this review.

REFERENCES

[1]. Database system concept - Abraham Silberschatz, Henry F. Korth, S. Sudarshan (page 75 to 76)
[2]. https://www.javatpoint.com/dbms-sql-command
[3.1]. 1 to 7 points (https://www.projectpro.io/article/applications-of-sql/834)
[3.2]. 8 and 9 point https://www.geeksforgeeks.org/what-are-the-advantages-and- disadvantages-of-using-sql-vs-nosql-databases/#use-cases-for-sql-databases
[4]. https://www.geeksforgeeks.org/top-sql-databases-to-learn/
[5]. https://aspiringyouths.com/advantages-disadvantages/sql/
[6]. NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence – Pramod J. Sadalage and Martin Fowler (Pages no: 192)
[7]. https://www.geeksforgeeks.org/what-are-the-advantages-and-disadvantages-of-using-sql- vs-nosql-databases/#use-cases-for-nosql-databases
[8]. https://www.mongodb.com/resources/basics/databases/nosql-explained
[9]. https://www.geeksforgeeks.org/open-source-nosql-databases/
[10]. https://www.geeksforgeeks.org/difference-between-sql-and-nosql/
[11]. https://aspiringyouths.com/advantages-disadvantages/nosql/