

# NoSQL: Not Only SQL Part-1

*Instructor: Mainack Mondal**Scribe: Sourav Poria*

Relational databases have long been the standard for structured data storage, providing robust and organized frameworks based on relational data models. However, NoSQL emerged in the late 2000s to address the limitations of SQL databases, particularly scalability, support for multi-structured data, geo-distribution, and adaptability to agile development processes.

## 1 What is NoSQL?

NoSQL, short for “Not Only SQL,” refers to non-relational databases that store data in non-tabular formats. Unlike relational databases, they lack a fixed schema, providing greater flexibility. They share common characteristics:

- Lack of a rigid schema for more adaptability.
- Distributed architectures ensure high scalability and fault tolerance.
- Non-relational data models, tailored for diverse use cases.

### 1.1 Types of NoSQL Databases

Four main types of NoSQL databases have emerged over time:

- **Document Databases:** Store semi-structured data as JSON-like documents. Ideal for blogging platforms, e-commerce, and real-time analytics.
- **Key-Value Databases:** Use simple key-value pairs for data storage. Suitable for user preferences, shopping carts, and web user profiles.
- **Column-Oriented Databases:** Organize data as a set of columns. Widely used in analytics and recommendation engines.
- **Graph Databases:** Represent data using nodes, edges, and properties. Useful for knowledge graphs and relationship-heavy datasets.

## 2 Why Do We Need NoSQL?

NoSQL databases address specific needs and use cases:

- **Agility:** Adapts to changing requirements without restructuring schemas.
- **Handling Unstructured Data:** Supports diverse data types natively.
- **Auto-Sharding:** Automatically distributes data across nodes for scalability.
- **Polyglot Persistence:** Allows combining multiple data models in a single application.
- **Flexible Storage:** Meets varying storage needs.

### 3 Limitations of SQL

Some common challenges associated with SQL databases include:

- **Object-Relational Impedance Mismatch:** Difficulty aligning object-oriented programming with relational tables.
- **Scalability:** Limited ability to scale horizontally.
- **Schema Rigidity:** Difficulty accommodating evolving data requirements.
- **Expensive Joins:** Performance bottlenecks with complex queries.

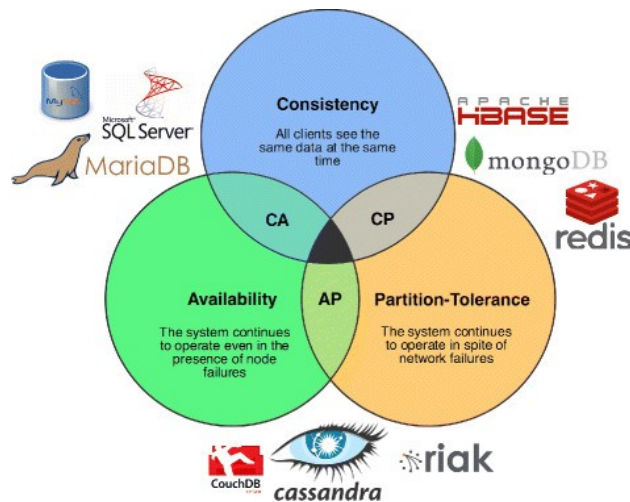
### 4 NoSQL Trade-offs

Although NoSQL databases have gained popularity, they come with trade-offs:

#### 4.1 CAP Theorem

It is impossible for a distributed system to simultaneously provide:

- **Consistency:** All nodes see the same updates in the same order.
- **Availability:** Every request receives a response.
- **Partition Tolerance:** The system continues to function despite network partitions.



#### 4.2 Limited ACID Support

NoSQL databases often provide less robust ACID (Atomicity, Consistency, Isolation, Durability) guarantees, making them unsuitable for some applications.

NoSQL databases usually aren't a good choice for applications that run complex queries and joins. Managing indexes and queries across multiple nodes would be slow and may not return consistent results.

## 5 SQL vs. NoSQL

Table 1: Comparison of SQL and NoSQL

| Criteria           | Use Case                              | SQL                                   | NoSQL                        |
|--------------------|---------------------------------------|---------------------------------------|------------------------------|
| ACID Compliance    | Banking Systems, Inventory Management | Ensures ACID compliance               | Limited ACID compliance      |
| Complex Queries    | Reporting and Analytics               | Supports complex queries with JOINS   | Best for simple queries      |
| Scalability        | Large Data Workloads                  | Limited to vertical scaling           | Horizontal scaling supported |
| Data Relationships | E-commerce Systems                    | Suitable for structured relationships | Less suitable                |
| Data Variety       | Big Data Applications                 | Suitable for structured data          | Ideal for unstructured data  |

## 6 References

- Google Cloud. *What is NoSQL?* <https://cloud.google.com/discover/what-is-nosql>
- Alamgir Qazi. *Introduction to NoSQL*. <https://github.com/alamgirqazi/IntroToNoSQL>.
- T9 - NoSQL. *Slides provided as course materials*.