

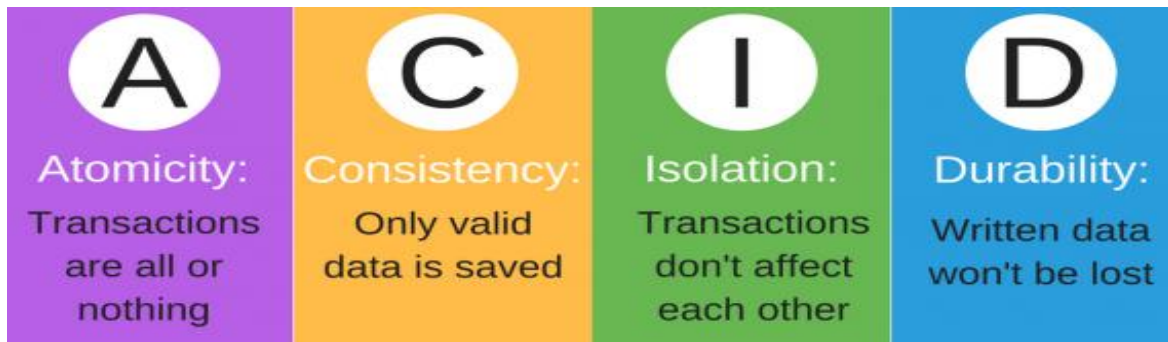
# "MongoDB and NoSQL"

Documented By: HARSHA K

## 1. Introduction to SQL Databases

Structured Query Language (**SQL**) databases have been the foundation of data management for decades. They are **Relational Database Management Systems (RDBMS)** like **MySQL, PostgreSQL, Oracle DB**, etc., where data is stored in **tables (rows and columns)** with clearly defined **schemas**.

SQL databases are **ACID-compliant**, meaning they ensure:



### Strengths of SQL:

- Strong structure & data integrity.
- Ideal for complex queries & joins.

## 2. Limitations of SQL Databases

Despite their strength, SQL databases face challenges in today's modern, scalable, real-time applications.

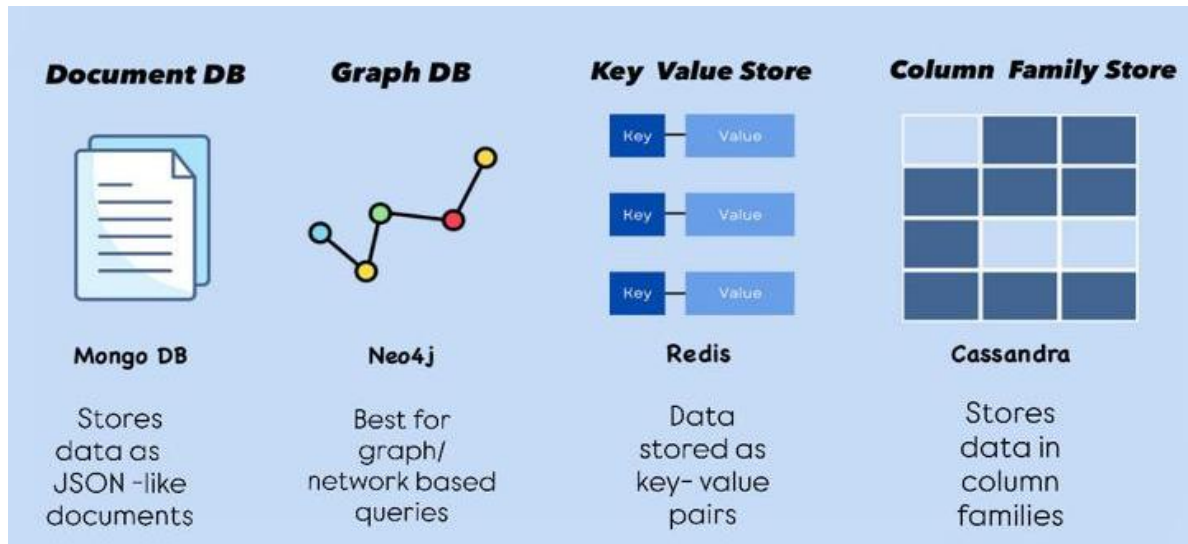
### Major Limitations:

- **Rigid Schema:** Changing table structure is hard.
- **Vertical Scaling:** Can only increase performance by upgrading a single server.
- **Poor for Unstructured Data:** Not ideal for JSON, images, documents, etc.
- **Complex Joins at Scale:** Difficult to handle high-volume joins in big data.
- **Limited Performance for Real-Time Apps:** High read/write at scale can be problematic.

These challenges led to the rise of **NoSQL** databases.

### 3. Introduction to NoSQL Databases

**NoSQL** means "**Not Only SQL**". These databases are designed to handle **large-scale, unstructured or semi-structured data**. They are **schema-less**, support **horizontal scaling**, and come in various types:



#### Strengths of NoSQL:

- **Flexible Schema:** Fields can be added/removed easily.
- **High Performance:** Optimized for read/write at scale.
- **Horizontal Scalability:** Add more machines to scale.
- **Ideal for Modern Applications:** Real-time data, mobile, IoT, social apps.

### 4. Limitations of NoSQL Databases

Although NoSQL provides many benefits, it's not perfect.

#### Challenges with NoSQL:

- **Lack of Standardization:** No single query language like SQL.
- **Weaker Consistency Guarantees:** BASE instead of ACID (Eventually Consistent).
- **Limited Support for Complex Joins:** You may need to manually handle relations.
- **Learning Curve:** Query language and data modeling are often different per DB.

## 5. MongoDB

**MongoDB** is the **most popular NoSQL database**, known for its flexibility, performance, and developer-friendly design. It is a **document-oriented** database, where each record is a **JSON-like document** stored in a **collection**.



## 6. Why MongoDB Over SQL and Other NoSQL DBs?

### Advantages Over SQL:

- No rigid schema – can evolve the model during development.
- JSON-like storage – perfect for APIs and modern apps.
- Easier scaling – add servers easily.
- High-speed reads/writes without complex joins.

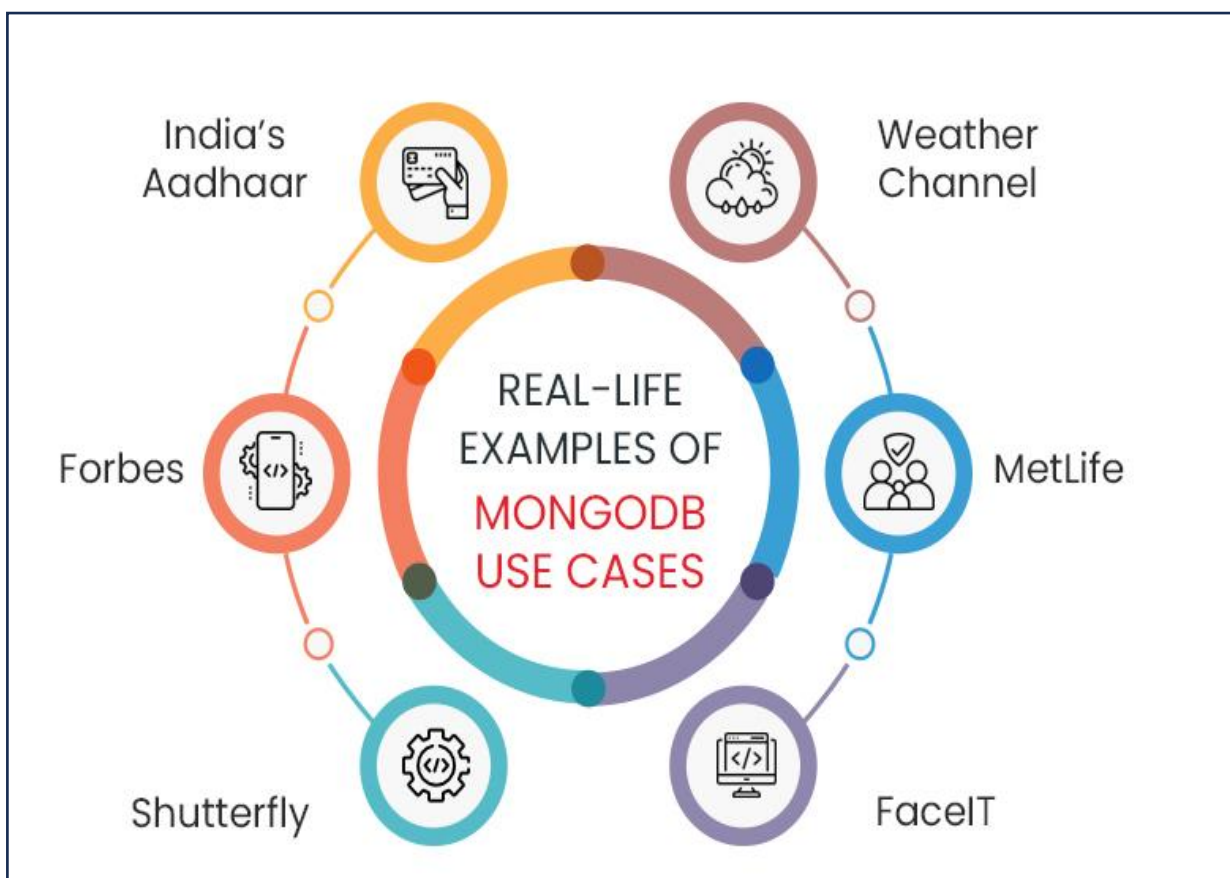
### Advantages Over Other NoSQL:

- **More powerful queries:** Rich query language (unlike simple key-value stores).
- **Better tooling & ecosystem:** Compass, Atlas (cloud), robust drivers.
- **Strong community & support:** Popular among startups and enterprises.

## 7. Use Cases of MongoDB

MongoDB has become a preferred database for many modern applications, especially those that require flexibility, scalability, and high availability. Here are some key areas where MongoDB is effectively used:

- **E-commerce Platforms**  
Handles dynamic product catalogs with varying fields like size, color, or brand.
- **Social Media Applications**  
Efficiently stores posts, comments, likes, and user interactions in a flexible format.
- **Internet of Things (IoT)**  
Ingests and manages high volumes of sensor data from various connected devices.
- **Content Management Systems (CMS)**  
Stores diverse content types like articles, images, and tags with ease of updates.
- **Mobile and Web App Backends**  
Provides fast performance and flexible schemas for scalable app development.
- **Real-Time Analytics**  
Enables quick insights from large, changing datasets for dashboards and personalization.



## 8. Conclusion

SQL databases (like MySQL, PostgreSQL, Oracle) have long been the standard for data storage. They are excellent for scenarios requiring strict data integrity, complex relationships, and transactional processing—such as banking, inventory, or ERP systems.

However, as applications became more distributed and data became semi-structured or unstructured (e.g., logs, media, user preferences), SQL databases showed limitations:

- Rigid schemas
- Complex joins
- Horizontal scaling difficulties

NoSQL databases like MongoDB emerged as a response to these challenges. They are schema-less, horizontally scalable, and designed for handling high-volume, high-velocity data.

### **Why MongoDB Stands Out:**

- Uses a document model (BSON/JSON), making it intuitive for developers
- Supports indexing, aggregation, and rich queries
- Easy horizontal scaling with built-in sharding
- Flexible schema enables agile development

### **We can Use SQL when:**

- Data is structured and relational
- You need multi-table transactions and consistency

### **We can Use MongoDB when:**

- Data is semi-structured, varied, or changes often
- You prioritize speed, scalability, and developer productivity.