

# **Introduction to NoSQL Databases**

Recitation 9

# What is a NoSQL Database?

Comparison of NoSQL Databases and Relational Databases

Feature	NoSQL Databases	Relational Databases
Data Model	Flexible and schema-less	Structured and schema-based
Scalability	Horizontally scalable	Vertically scalable
Data Structure	Key-value, document, columnar, graph	Tables with rows and columns
Query Language	Varies by database (e.g., MongoDB uses a JSON-like query language)	Structured Query Language (SQL)
Data Integrity	Eventual consistency	ACID (Atomicity, Consistency, Isolation, Durability) properties
Performance	High read and write throughput	Optimized for structured queries
Scaling	Easy to scale horizontally by adding more servers	Scaling requires vertical hardware upgrades
Use Cases	Big data, real-time analytics, content management systems	Enterprise applications, financial systems, e-commerce

# Advantages of NoSQL Databases

## Key Advantages

- **Scalability:** NoSQL databases are highly scalable and can handle large amounts of data without sacrificing performance.
- **Flexibility:** NoSQL databases allow for flexible data models, making it easier to adapt to changing requirements and schema-less data.
- **High Performance:** NoSQL databases are designed for high-speed data retrieval and processing, making them ideal for applications that require real-time data access.
- **Horizontal Scaling:** NoSQL databases can be easily scaled horizontally by adding more servers to distribute the data load, ensuring high availability and fault tolerance.

```
{
  "FirstName"      : "Sam",
  "LastName"       : "Jackson",
  "employeeID"     : 5698523,
  "Designation"    : "Manager",
  "LanguageExpertise" : ["Java", "C#", "Python"]
  "Car"            : {
    "makeModel" : "Maruti Suzuki Swift",
    "makeYear"  : 2017,
    "color"     : "Red",
    "type"      : "Hatchback",
  }
}
```

# Disadvantages of NoSQL Databases

## Key Disadvantages

- **Lack of Standardization:** NoSQL databases lack a standardized query language, making it difficult to switch between different databases or integrate them into existing systems.
- **Limited Querying Capabilities:** NoSQL databases often have limited querying capabilities compared to SQL databases, making complex queries and joins more challenging.
- **Potential Data Inconsistency:** NoSQL databases prioritize scalability and performance over data consistency, which can lead to potential data inconsistencies in certain scenarios.
- **Complexity and Additional Development Effort:** NoSQL databases can be more complex to work with and may require additional development effort compared to traditional SQL databases.

# Examples of NoSQL Databases: MongoDB



## E-commerce

MongoDB is widely used in e-commerce applications for its ability to handle large amounts of product data and provide real-time inventory management.



## Social Media

MongoDB is utilized by social media platforms to store and retrieve user-generated content, such as posts, comments, and user profiles, due to its flexible schema and scalability.

# How to Connect to a MongoDB Database?



- Go to <https://www.mongodb.com/cloud/atlas/register>
- Create your account
- Create a database cluster as a shared cluster, which is free
- Set a user for your database from Database Access tab on the left panel
- Go back to your database section from left panel and click connect
- Select your development environment and copy the connection string
- Follow the steps from recitation codes:  
<https://github.com/hasan-ert/cs306-202302-recit>
- You may also check out this video:  
<https://www.youtube.com/watch?v=3wNvKybVyal>