

NoSQL Introduction



Agenda

- Introduction to NoSQL
- What is NoSQL?
- Types of NoSQL Databases
- Key Characteristics
- Benefits and Drawbacks
- When to Use NoSQL
- Use Cases



Introduction

What are Databases?

- Databases are structured collections of data used for storing, managing, and retrieving information.

Traditional Relational Databases

- Traditional relational databases have served as the foundation for data storage for decades.
- They use structured tables and SQL for data management.

The Need for NoSQL

- Modern applications generate vast amounts of unstructured or semi-structured data.
- Traditional databases struggle to handle the scale and flexibility needed for these data types.



What is NoSQL?

- NoSQL stands for "Not Only SQL."
- NoSQL databases offer flexible data models and are designed to handle diverse, unstructured, or semi-structured data.
- They are not limited to SQL query language and relational data models.

Types of NoSQL Databases

Document Store:

- MongoDB and CouchDB.
- Store data as documents in formats like JSON or XML.

Key-Value Store:

- Redis and DynamoDB.
- Store data as key-value pairs.

Column Family Store:

- Apache Cassandra and HBase.
- Store data in columns rather than rows.

Graph Database:

- Neo4j and Amazon Neptune.
- Designed for managing and querying graph-structured data.

Key Characteristics

Schema-less:

- NoSQL databases do not require a predefined schema.
- Data can be added or changed without structural constraints.

Horizontal Scalability:

- NoSQL databases can easily scale out by adding more servers.

High Availability:

- Emphasize fault tolerance and data redundancy.

Distributed Architecture:

- Designed to work in distributed environments, often spanning multiple servers or data centers.

CAP Theorem:

- The CAP theorem (Consistency, Availability, Partition tolerance) outlines trade-offs in distributed systems.

Benefits of NoSQL

Scalability:

- NoSQL databases are highly scalable horizontally, accommodating growth.

Flexibility:

- They offer flexibility in adapting to changing data requirements.

High Performance:

- NoSQL databases can provide exceptional performance for specific use cases.

Schema Evolution:

- Changes to data structure can be made without system downtime.

Cost-Effective:

- NoSQL databases can be cost-effective for workloads that require high performance and scale.

Drawbacks of NoSQL

Lack of ACID Compliance:

- NoSQL databases often prioritize scalability and flexibility over strict ACID compliance.

Consistency Challenges:

- Discuss the trade-offs in terms of eventual consistency.

Limited Ad-hoc Query Capabilities:

- Mention challenges in performing complex, ad-hoc queries compared to traditional databases.

Learning Curve:

- Adapting to NoSQL databases may require a learning curve.

Not a One-Size-Fits-All Solution:

- NoSQL databases are not suitable for all data and application types.



When to Use NoSQL

Use Cases:

- NoSQL databases are well-suited for specific scenarios.

Real-time Big Data Processing:

- Ideal for real-time analytics and handling large datasets.

High Read/Write Workloads:

- Excellent for applications with a high volume of simultaneous read and write operations.

Flexible Data Models:

- Beneficial for rapidly changing or evolving data structures.

Rapid Development:

- Supports agile development and quick iteration.



Use Cases

Examples in Various Industries:

NoSQL databases find extensive applications across a wide range of industries. Let's explore some real-world use cases to illustrate their versatility:

Use Cases : Social Media

- Social media platforms handle vast amounts of user-generated content, from text and images to videos and interactions.
- NoSQL databases, such as MongoDB, enable the efficient storage and retrieval of this diverse and unstructured data.
- Relationships among users, their posts, and social connections are effectively managed with graph databases like Neo4j.

Use Cases : IoT

- In the IoT landscape, countless devices continuously generate data, from sensors in smart homes to machinery on factory floors.
- NoSQL databases like Apache Cassandra and MongoDB play a pivotal role in processing and analyzing this real-time, high-velocity data.
- They provide scalability and flexibility to accommodate dynamic data volumes and structures.

Use Cases : E-commerce

- E-commerce platforms rely on NoSQL databases to deliver seamless and personalized experiences to customers.
- Product catalogs are dynamic, with constantly changing attributes and relationships.
- NoSQL databases offer the agility to handle evolving product data, and they excel in personalizing recommendations and optimizing inventory management.

Use Cases :Content Management

- Content management systems (CMS) deal with a myriad of unstructured content, such as articles, images, audio, and metadata.
- NoSQL databases, including Couchbase and MongoDB, efficiently manage unstructured content, enabling quick retrieval and dynamic content updates.
- Metadata, which plays a crucial role in search and categorization, can be stored and queried effectively.

Use Cases :Content Management

MongoDB:

- MongoDB is a widely used document-oriented NoSQL database.

Cassandra:

- Apache Cassandra excels in managing large datasets across multiple nodes.

Redis:

- Redis is a high-performance in-memory key-value store.

Neo4j:

- Neo4j is a popular graph database used for complex relationship modeling.

HBase:

- HBase is a prominent wide-column store, suitable for massive data storage.

Couchbase:

- Couchbase is known for its support in interactive applications.



THANK YOU