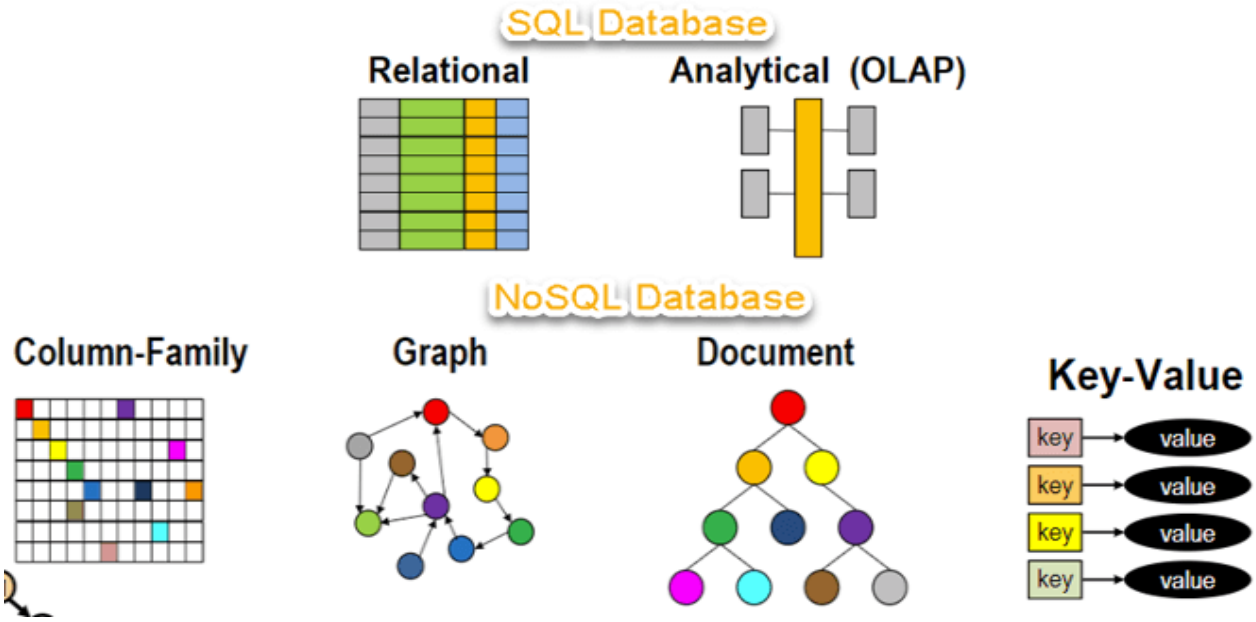
What is NoSQL?

NoSQL Database is a non-relational Data Management System, that does not require a fixed schema. It avoids joins, and is easy to scale. The major purpose of using a NoSQL database is for distributed data stores with humongous data storage needs. NoSQL is used for Big data and real-time web apps. For example, companies like Twitter, Facebook and Google collect terabytes of user data every single day.

NoSQL database stands for "Not Only SQL" or "Not SQL." Though a better term would be "NoREL", NoSQL caught on. Carl Strozz introduced the NoSQL concept in 1998.

Traditional RDBMS uses SQL syntax to store and retrieve data for further insights. Instead, a NoSQL database system encompasses a wide range of database technologies that can store structured, semi-structured, unstructured and polymorphic data. Let's understand about NoSQL with a diagram in this NoSQL database tutorial:

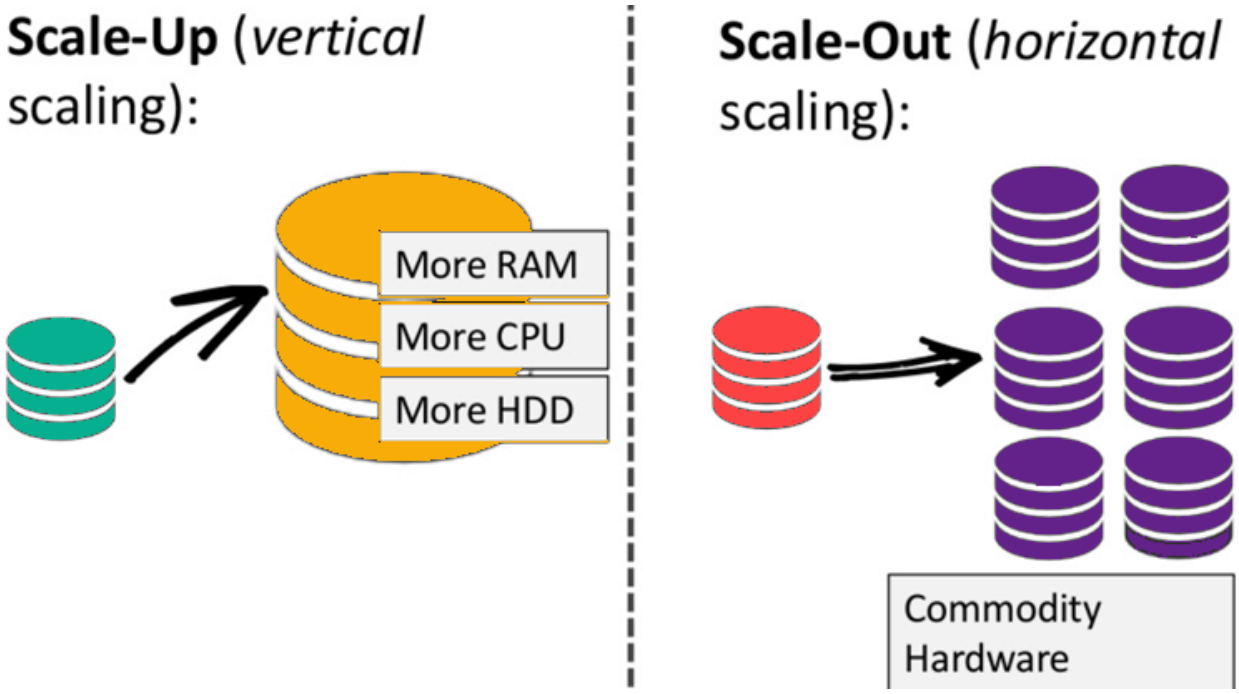


Why NoSQL?

The concept of No SQL databases became popular with Internet giants like Google, Facebook, Amazon, etc. who deal with huge volumes of data. The system response time becomes slow when you use RDBMS for massive volumes of data.

To resolve this problem, we could "scale up" our systems by upgrading our existing hardware. This process is expensive.

The alternative for this issue is to distribute database load on multiple hosts whenever the load increases. This method is known as "scaling out."



NoSQL database is non-relational, so it scales out better than relational databases as they are designed with web applications in mind.

Brief History of NoSQL Databases

- 1998- Carlo Strozzi use the term NoSQL for his lightweight, open-source relational database
- 2000- Graph database Neo4j is launched
- 2004- Google BigTable is launched
- 2005- CouchDB is launched
- 2007- The research paper on Amazon Dynamo is released
- 2008- Facebooks open sources the Cassandra project
- 2009- The term NoSQL was reintroduced

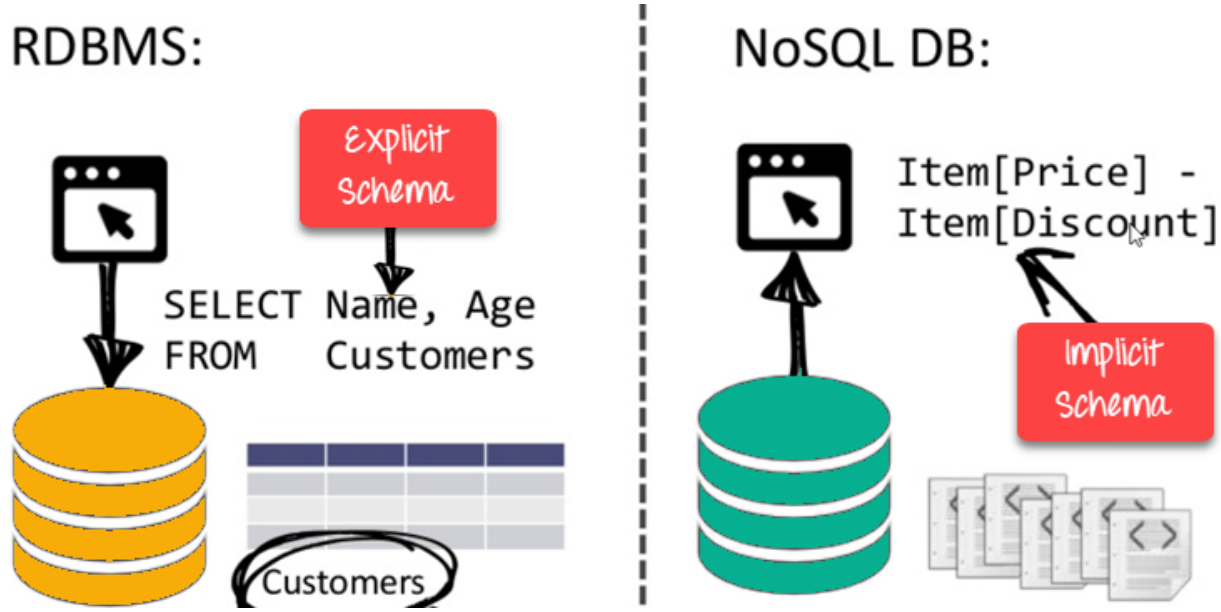
Features of NoSQL

Non-relational

- NoSQL databases never follow the relational model
- Never provide tables with flat fixed-column records
- Work with self-contained aggregates or BLOBs
- Doesn't require object-relational mapping and data normalization
- No complex features like query languages, query planners,referential integrity joins, ACID

Schema-free

- NoSQL databases are either schema-free or have relaxed schemas
- Do not require any sort of definition of the schema of the data
- Offers heterogeneous structures of data in the same domain



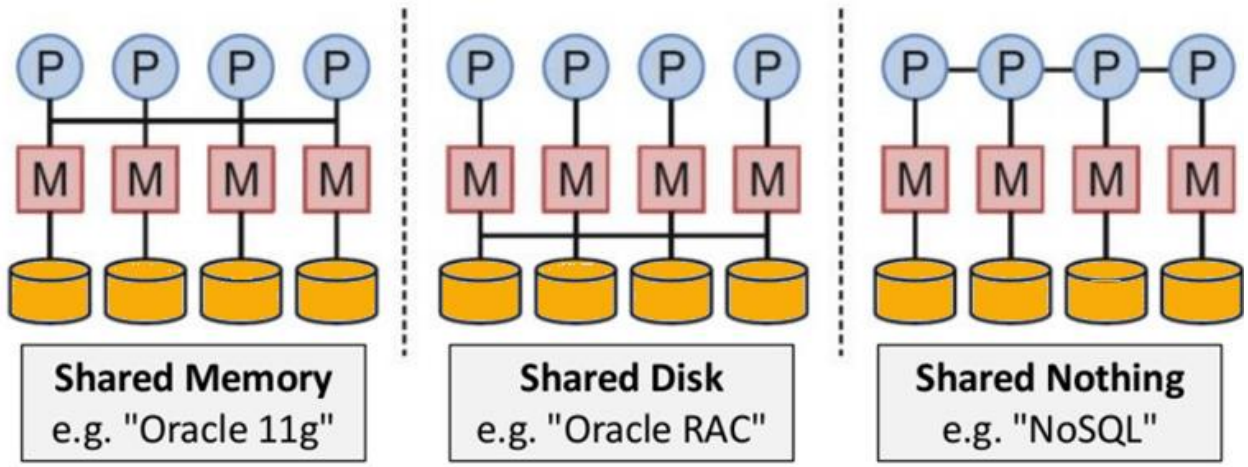
NoSQL is Schema-Free

Simple API

- Offers easy to use interfaces for storage and querying data provided
- APIs allow low-level data manipulation & selection methods
- Text-based protocols mostly used with HTTP REST with JSON
- Mostly used no standard based NoSQL query language
- Web-enabled databases running as internet-facing services

Distributed

- Multiple NoSQL databases can be executed in a distributed fashion
- Offers auto-scaling and fail-over capabilities
- Often ACID concept can be sacrificed for scalability and throughput
- Mostly no synchronous replication between distributed nodes Asynchronous Multi-Master Replication, peer-to-peer, HDFS Replication
- Only providing eventual consistency
- Shared Nothing Architecture. This enables less coordination and higher distribution.



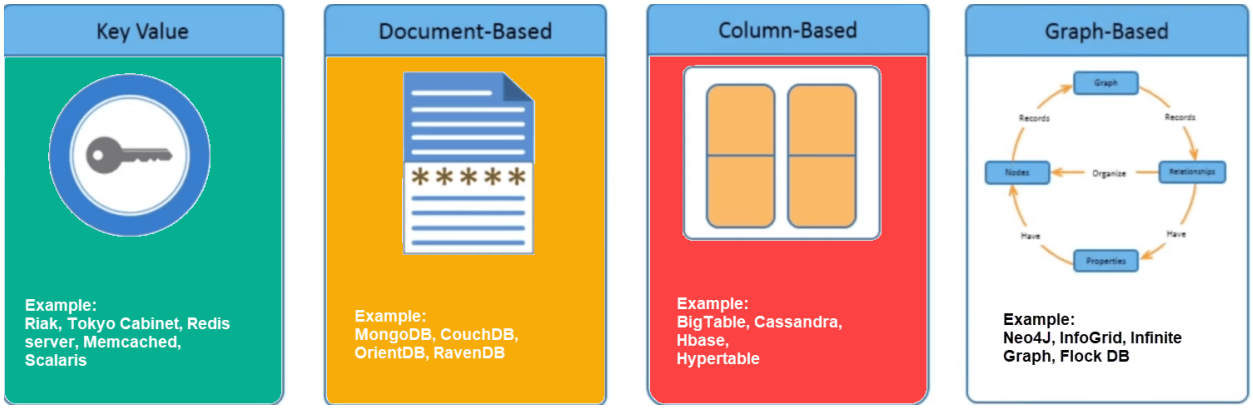
NoSQL is Shared Nothing.

Types of NoSQL Databases

NoSQL Databases are mainly categorized into four types: Key-value pair, Column-oriented, Graph-based and Document-oriented. Every category has its unique attributes and limitations. None of the above-specified database is better to solve all the problems. Users should select the database based on their product needs.

Types of NoSQL Databases:

- Key-value Pair Based
- Column-oriented Graph
- Graphs based
- Document-oriented



Key Value Pair Based

Data is stored in key/value pairs. It is designed in such a way to handle lots of data and heavy load.

Key-value pair storage databases store data as a hash table where each key is unique, and the value can be a JSON, BLOB(Binary Large Objects), string, etc.

For example, a key-value pair may contain a key like “Website” associated with a value like “Guru99”.

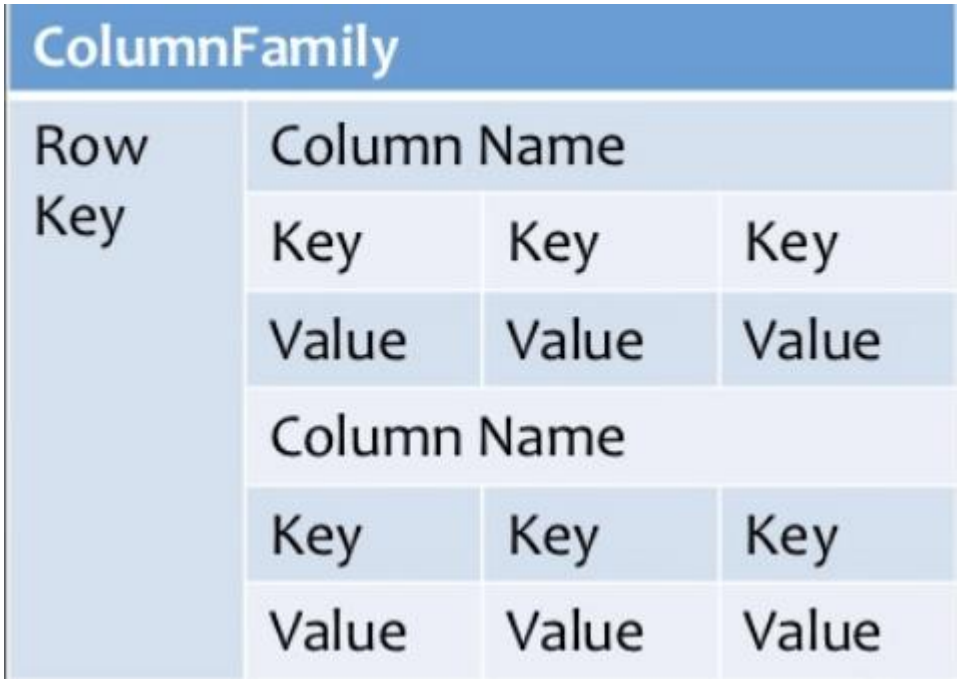
Key	Value
Name	Joe Bloggs
Age	42
Occupation	Stunt Double
Height	175cm
Weight	77kg

It is one of the most basic NoSQL database example. This kind of NoSQL database is used as a collection, dictionaries, associative arrays, etc. Key value stores help the developer to store schema-less data. They work best for shopping cart contents.

Redis, Dynamo, Riak are some NoSQL examples of key-value store DataBases. They are all based on Amazon’s Dynamo paper.

Column-based

Column-oriented databases work on columns and are based on BigTable paper by Google. Every column is treated separately. Values of single column databases are stored contiguously.



Column based NoSQL database

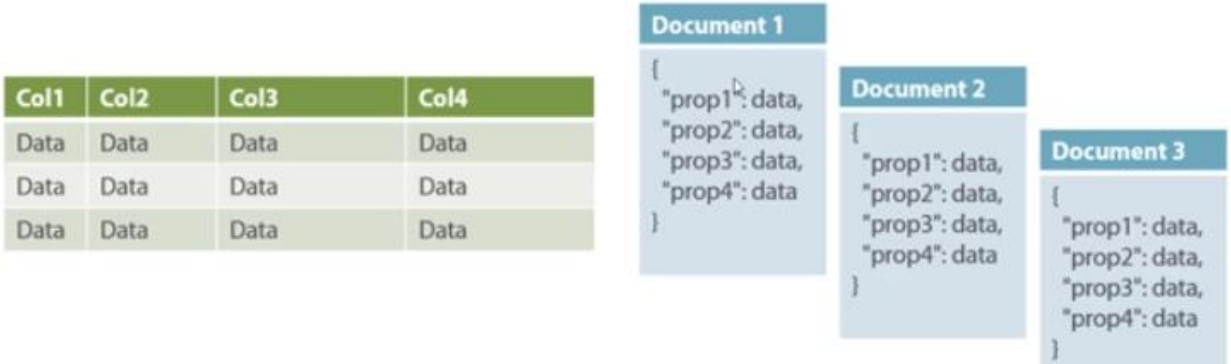
They deliver high performance on aggregation queries like SUM, COUNT, AVG, MIN etc. as the data is readily available in a column.

Column-based NoSQL databases are widely used to manage data warehouses, business intelligence, CRM, Library card catalogs,

HBase, Cassandra, HBase, Hypertable are NoSQL query examples of column based database.

Document-Oriented

Document-Oriented NoSQL DB stores and retrieves data as a key value pair but the value part is stored as a document. The document is stored in JSON or XML formats. The value is understood by the DB and can be queried.



Relational Vs. Document

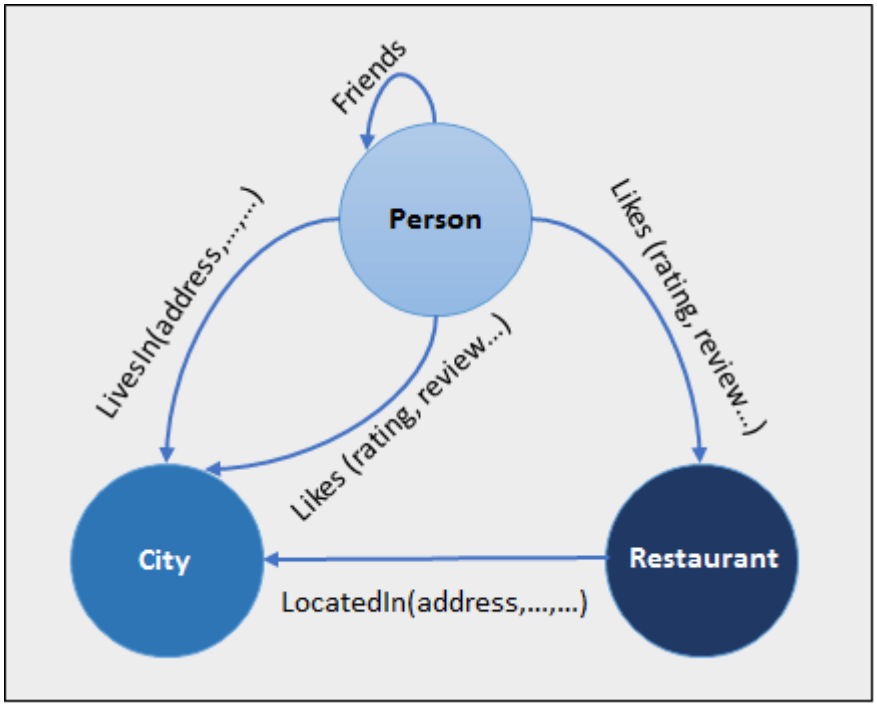
In this diagram on your left you can see we have rows and columns, and in the right, we have a document database which has a similar structure to JSON. Now for the relational database, you have to know what columns you have and so on. However, for a document database, you have data store like JSON object. You do not require to define which make it flexible.

The document type is mostly used for CMS systems, blogging platforms, real-time analytics & e-commerce applications. It should not use for complex transactions which require multiple operations or queries against varying aggregate structures.

Amazon SimpleDB, CouchDB, MongoDB, Riak, Lotus Notes, MongoDB, are popular Document originated DBMS systems.

Graph-Based

A graph type database stores entities as well the relations amongst those entities. The entity is stored as a node with the relationship as edges. An edge gives a relationship between nodes. Every node and edge has a unique identifier.



Compared to a relational database where tables are loosely connected, a Graph database is a multi-relational in nature. Traversing relationship is fast as they are already captured into the DB, and there is no need to calculate them.

Graph base database mostly used for social networks, logistics, spatial data.

Neo4J, Infinite Graph, OrientDB, FlockDB are some popular graph-based databases.

Advantages of NoSQL

- Can be used as Primary or Analytic Data Source
- Big Data Capability
- No Single Point of Failure
- Easy Replication
- No Need for Separate Caching Layer
- It provides fast performance and horizontal scalability.
- Can handle structured, semi-structured, and unstructured data with equal effect
- Object-oriented programming which is easy to use and flexible
- NoSQL databases don't need a dedicated high-performance server
- Support Key Developer Languages and Platforms
- Simple to implement than using RDBMS
- It can serve as the primary data source for online applications.
- Handles big data which manages data velocity, variety, volume, and complexity
- Excels at distributed database and multi-data center operations
- Eliminates the need for a specific caching layer to store data
- Offers a flexible schema design which can easily be altered without downtime or service disruption

Disadvantages of NoSQL

- No standardization rules
- Limited query capabilities
- RDBMS databases and tools are comparatively mature
- It does not offer any traditional database capabilities, like consistency when multiple transactions are performed simultaneously.
- When the volume of data increases it is difficult to maintain unique values as keys become difficult
- Doesn't work as well with relational data
- The learning curve is stiff for new developers
- Open source options so not so popular for enterprises.

Applications of NoSQL Databases

1. **Data Mining**

When it comes to data mining, NoSQL databases are useful in retrieving information for data mining uses. Particularly when it's about large amounts of data, NoSQL databases store data points in both structured and unstructured formats leading to efficient storage of big data.

Perhaps when a user wishes to mine a particular dataset from large amounts of data, one can make use of NoSQL databases, to begin with. Data is the building block of technology that has led mankind to such great heights.

Therefore, one of the most essential fields where NoSQL databases can be put to use is data mining and data storage.

(Also read - Top 10 Data Mining Tools)

2. **Social Media Networking Sites**

Social media is full of data, both structured and unstructured. A field that is loaded with tons of data to be discovered, social media is one of the most effective applications of NoSQL databases.

From comments to posts, user-related information to advertising, social media marketing requires NoSQL databases to be implemented in certain ways to retrieve useful information that can be helpful in certain ways.

Social media sites like Facebook and Instagram often approach open-source NoSQL databases to extract data that helps them keep track of their users and the activities going on around their platforms.

3. **Software Development**

The third application that we will be looking at is software development. Software development requires extensive research on users and the needs of the masses that are met through software development.

However, a developer must be able to scan through data that is available.

Perhaps NoSQL databases are always useful in helping software developers keep a tab on their users, their details, and other user-related data that is important to be noted. That said, NoSQL databases are surely helpful in software development.

What is NewSQL?

NewSQL is a class of relational database management systems (RDBMS) that aims to provide the scalability and performance of NoSQL databases while maintaining the transactional consistency and reliability of traditional SQL databases. It is designed to handle large-scale, high-velocity data processing and analytics workloads.

How NewSQL Works

NewSQL databases are built on distributed architectures and leverage techniques such as sharding, replication, and parallel processing to achieve high scalability. They also employ innovative query optimization and indexing techniques to ensure fast query performance and efficient data retrieval.

Why NewSQL is Important

NewSQL databases offer several benefits over traditional SQL databases and NoSQL databases:

- **Scalability:** NewSQL databases can scale horizontally across multiple machines, allowing them to handle massive amounts of data and high concurrency workloads.
- **ACID Compliance:** Unlike most NoSQL databases, NewSQL databases support ACID (Atomicity, Consistency, Isolation, Durability) properties, ensuring data consistency and reliability.
- **Performance:** NewSQL databases are optimized for high-performance data processing and analytics, enabling faster query execution and real-time insights.
- **Flexibility:** NewSQL databases provide the flexibility to handle both structured and semi-structured data, making them suitable for a wide range of use cases.

Important NewSQL Use Cases

NewSQL databases are particularly well-suited for the following use cases:

- **Online Transaction Processing (OLTP):** NewSQL databases excel at handling high-volume transactional workloads, making them suitable for applications that require real-time data processing and low-latency responses.
- **Analytics and Business Intelligence:** NewSQL databases can efficiently process complex analytical queries and enable real-time analytics, empowering businesses to derive valuable insights from their data.
- **Data Warehousing and Data Lakes:** NewSQL databases can serve as the backbone for data warehousing and data lake architectures, providing a scalable and reliable foundation for storing and processing big data.

Technologies Related to NewSQL

There are several technologies and terms closely related to NewSQL:

- **NoSQL:** NoSQL databases offer high scalability and flexibility but sacrifice some level of ACID compliance. NewSQL databases bridge the gap between traditional SQL and NoSQL by providing scalable ACID-compliant solutions.
- **Distributed Databases:** NewSQL databases are distributed databases that span multiple machines or nodes, allowing for horizontal scalability and fault tolerance.
- **In-Memory Databases:** Some NewSQL databases leverage in-memory storage and processing to achieve even higher performance and lower latency.

Why Dremio Users Should Know About NewSQL

Dremio is an advanced data lakehouse platform that enables businesses to easily explore, analyze, and extract insights from their data lake. While Dremio has powerful native data processing capabilities, it can also integrate with other data technologies to provide a comprehensive data solution.

NewSQL databases, with their ability to handle large-scale data processing and analytics workloads, can be a valuable addition to the Dremio ecosystem. By integrating NewSQL

databases with Dremio, users can further optimize their data processing pipelines, enhance query performance, and unlock the full potential of their data lakehouse environment.

Dremio offers unique capabilities that differentiate it from NewSQL databases:

- **Virtual Data Reflections:** Dremio's Data Reflections technology enables high-performance data acceleration by automatically creating and maintaining optimized views of the underlying data. This can significantly speed up query execution and improve overall performance.
- **Data Virtualization:** Dremio allows users to virtually integrate and query data from multiple sources without the need for data movement or ETL processes. This provides a unified view of the data, simplifying analytics and reducing data duplication.
- **Self-Service Data Preparation:** Dremio's data preparation capabilities enable users to quickly and easily transform and shape their data without relying on complex SQL queries. This empowers data analysts and scientists to explore and derive insights from data on their own.

What is SQL?

SQL, also known as Structured Query Language, is a flexible programming language specifically designed for the management and manipulation of relational databases. It acts as a standardized communication tool between databases and applications. SQL empowers users to define, access, and modify data within a database system.

With a rich set of commands, SQL enables tasks like creating and modifying database structures (DDL), manipulating data (DML), and managing access and permissions (DCL). It excels in executing complex operations such as sorting, joining, aggregating, and grouping data. It also ensures an efficient data retrieval.

SQL finds extensive applications across various fields, serving as an essential tool for database administrators, developers, and data analysts. It is widely supported by popular database management systems like MySQL, Oracle, SQL Server, and PostgreSQL.

Advanced SQL allows individuals to proficiently handle and interact with databases, which helps in enabling critical tasks like data retrieval, manipulation, and database administration. Its user-friendly nature, adaptability, and widespread adoption have solidified SQL's position as an indispensable language in data management and analysis.

What is NO SQL?

NoSQL, also known as "Not Only SQL," represents a category of databases that offers an alternative approach to data management compared to traditional SQL databases. It provides a flexible and scalable solution for storing and handling diverse data types, such as unstructured and semi-structured data which is found in social media posts, sensor data, and real-time streams.

Unlike SQL databases, NoSQL databases do not adhere to a rigid schema and instead employ various data models, including key-value stores, document databases, column-family stores, and graph databases. SQL's versatility enables dynamic data modeling to cater to the specific requirements of various applications which make it adaptable and customizable.

NoSQL databases prioritize scalability, high performance, and fault tolerance. All these features make the databases well-suited for those situations which invlove large data volumes, rapid data processing, and distributed computing. They excel at handling the complexities of modern data environments.

Organizations adopting NoSQL databases can leverage their capabilities to overcome challenges posted by big data, real-time processing, and dynamic data structures. However, it is crucial to carefully assess the requirements before selecting a NoSQL database solution that best aligns with the specific application needs.

What is New SQL?

NewSQL represents a category of database systems that blend the advantages of traditional SQL (relational) databases with the scalability and performance benefits offered by NoSQL databases. NewSQL aims to address the limitations of SQL databases, particularly in distributed environments which require high scalability.

These databases retain the essential ACID (Atomicity, Consistency, Isolation, Durability) properties for data integrity while implementing innovative techniques for handling large datasets and achieving impressive scalability. Through methods such as sharding, replication, and distributed architectures, NewSQL databases efficiently manage extensive workloads and meet the demands of modern applications.

Unlike NoSQL databases, NewSQL databases maintain compatibility with SQL, providing users with familiar SQL-based interfaces, complex queries, and transaction support. This makes them a compelling choice for organizations seeking both the scalability of NoSQL and the reliability of SQL.

NewSQL databases uses the flexibility and scalability of NoSQL databases while preserving the integrity and consistency of SQL databases.

Difference between SQL vs NO SQL vs New SQL

Below table illustrates the distinctions among SQL, NoSQL, and NewSQL databases:

S. No.	SQL	NO SQL	NEW SQL
1.	SQL databases organize data in structured tables.	NoSQL databases support various data models.	NewSQL databases maintain a relational data structure.

2.	SQL databases employ a rigid data schema that defines the structure and organization of data.	NoSQL databases have a flexible schema.	NewSQL databases also support a flexible schema.
3.	SQL is the standard query language used.	NoSQL databases have diverse query languages.	NewSQL databases also use SQL for queries.
4.	SQL databases have limited scalability.	NoSQL databases offer high scalability.	NewSQL databases provide enhanced scalability.
5.	SQL databases adhere to ACID properties	NoSQL databases have varying compliance.	NewSQL databases maintain ACID compliance.
6.	SQL databases generally do not have a distributed architecture as a characteristic feature.	NoSQL databases are specifically designed to operate in distributed environments.	NewSQL databases are specifically engineered with distributed architecture as a core principle.
7.	SQL databases support transactions.	NoSQL databases may have varied transaction support.	NewSQL databases offer transaction support.
8.	SQL databases are suitable for traditional applications with structured data.	NoSQL databases excel with unstructured and diverse data.	NewSQL databases cater to scalable applications with relational data and ACID compliance.