Non-Relational Databases

by Sophia

WHAT'S COVERED

In this lesson, you will explore the similarities and differences between relational databases and non-relational databases like NoSQL. Specifically, this lesson will cover:

1. Differentiating Between Relational and Non-Relational Databases

In previous lessons, we have talked about the design tradeoffs when designing a database. Depending on what you are building, a relational database might not be the right answer for everything. **Non-relational databases** are databases that do not use the traditional tabular relational model that is used in relational database management systems. You learned about NoSQL, one type of non-relational database, in the previous lesson.

Many companies use a blend of relational databases and non-relational databases to solve business problems. For example, Amazon uses NoSQL databases to deliver information on over a billion products, and when they were building their latest site, they needed a database that could deliver 10,000 pages a second and be easily scalable in capacity. A NoSQL database was their chosen solution for their product pages. For customer data, customer orders, and other information, a relational database is what they chose for that side of their business process flow.

REFLECT

Are the terms "non-relational" and "NoSQL" synonymous?

NoSQL databases are a subset of non-relational databases that have gained prominence for their ability to address specific challenges related to scalability, flexibility, and performance, especially in modern web applications and big data scenarios. Non-relational databases include a broader range of database systems, some of which may not be associated with the characteristics and design principles of NoSQL databases.

Among the key differences between relational and non-relational databases are:

- Tables provide the structure of relational databases, where data is organized into columns and rows. Each table has a fixed **schema**; relationships between tables are established with primary and foreign keys. While non-relational databases use a variety of data models such as key-value, document, columnar, or graph, non-relational databases don't. NoSQL can handle unstructured or semi-structured data because they provide more flexibility in data structures.
- In relational databases, scaling is typically achieved by upgrading hardware (e.g., adding more memory or processing power) to handle increased loads. In contrast, non-relational databases are designed to scale horizontally. Data can be distributed horizontally across multiple servers or clusters, allowing for seamless handling of large-scale data and high traffic loads.
- Relational databases have a rigid schema requiring predefined tables and fields. A schema change can be challenging and involve downtime or complex migration procedures. A non-relational database, on the other hand, offers schema flexibility. Schema changes can be made on the fly without disrupting the system significantly.
- Relational databases use SQL to retrieve and manipulate data. With SQL, you can write powerful queries that perform complex joins, aggregations, and transactions. The query languages used by non-relational databases are often specific to their respective data models. While less expressive than SQL, these query languages are designed to efficiently handle a database's particular data structure and characteristics.
- ACID refers to four database characteristics that ensure database integrity by controlling data changes: Atomicity, Consistency, Isolation, and Durability. You'll learn about them in Unit 5 of this course. Relational databases apply ACID principles strictly, which has a side effect of creating extra work that can affect performance and scalability. Some non-relational databases sacrifice strict ACID properties to achieve performance and scalability gains.
- A relational database is typically used for applications that require structured data with well-defined relationships, complex querying, and transactional integrity. Suitable applications include e-commerce platforms, financial systems, and inventory management. Unlike SQL databases, non-relational databases can handle large volumes of rapidly changing, unstructured, or semi-structured data. These systems are often used in real-time analytics, content management systems, social media platforms, and IoT data storage platforms.

Implementations of relational and non-relational databases can differ based on their purposes. There are many factors to consider when deciding between the two, notably the application requirements, the scalability needs, the data characteristics, and the expected workload.

This course focuses on relational databases, where data is organized into tables in a structured manner. There are relationships and dependencies between the tables. You use SQL to query the data in a specific format through the tables, columns, and rows. Each table contains data pertaining to a certain subject, such as Inventory or Customers. The columns represent the fields, which are individual pieces of data to be stored, such as Name or City. The rows represent the records, which are the instances being described, such as a specific inventory item or customer. This type of relational database structure makes it easy to sort, filter, and compute various calculations with expressive query languages. But you might run into companies using both databases, so it is good to be aware of non-relational databases.

One of the most common examples of a non-relational database is a NoSQL database, which uses an approach to database design that enables the storage and querying of data outside the traditional structures found in relational databases. NoSQL databases have a flexible data model, which makes it easy to store and combine data of any structure and allows you to modify the schema without any downtime dynamically. In contrast, making changes to the data model in a relational database can take time and planning to implement. Since the NoSQL databases are built on performance and scalability, they are set up to quickly scale out, offering unlimited growth compared to relational databases. They are also designed for highly available systems to provide a consistent, high-quality experience for all users worldwide.

Relational databases, on the other hand, use expressive query languages to access and modify data. They also use indexes within the database to provide efficient access to data rather than maintained in the application code for non-relational databases. With the relational models, there’s much stronger consistency through the primary and foreign keys, such that applications should be able to read what has been written to the database immediately.

TERMS TO KNOW

Schema

The structure of a database, including its tables and the relationships between them.

Non-Relational Databases

Databases that do not use the traditional tabular relational model that is used in relational database management systems.

2. Non-Relational Databases

Non-relational databases are databases that do not use the traditional tabular relational model that is used in relational database management systems. In this section, you will learn about the main types on non-relational databases and when to use them.

2a. When to Use a Non-Relational Database

If you're working with data that isn’t clearly defined and organized, and you can't establish defined tables and relationships in the dataset, you might want to consider a non-relational database. The information gathered in a non-relational database doesn't need to be segmented into tables and defined relationships.

EXAMPLE A messaging application like Facebook Messenger is an excellent example of the need for a non-relational database. With so much unstructured information that gets included within a message, the messages must be stored in a non-relational database for efficiency.

2b. Main Types of Non-Relational Databases

There are four main types of non-relational databases: document-oriented, key-value stores, wide-column stores, and graph stores.

Document-oriented databases focus on pairing a key with a complex data structure. It is similar to storing information in a Word document and just adding more information into that document. You must search through that document if you need to find the information you are looking for.

Key-value stores are databases that use different keys, where each key is only associated with one value. An example is a dictionary where the word is the key, and the value is the word's definition. This is one of the simplest types of non-relational databases. The Registry in Microsoft Windows stores system settings in this type of database.

Wide-column stores use tables, rows, and columns similar to a relational database, but the names and format of the columns can vary from row to row in the same table.

Graph stores use graph structures to link data through edges, nodes, and properties. It can be one of the more complex types of non-relational databases.

TERMS TO KNOW

Document-Oriented Database
A non-relational database that pairs a key with a complex data structure.

Key-Value Store
A simple non-relational database that pairs each key with a single value.

Wide-Column Store
A non-relational database that allows the names and format of columns of its tables to vary from row to row.

Graph Store
A non-relational database that links data through edges, nodes, and properties.

SUMMARY

In this lesson, you learned that there are **different uses for non-relational and relational databases** depending on the organizational need. Organizations need to decide which database design is needed for their specific purpose. Relational databases use structured data tables and a formal query language to offer consistency and efficiency, while **non-relational databases** like NoSQL have unstructured data storage that provides better flexibility and scalability than relational databases can usually achieve. You learned **when to use a non-relational database** and explored the four **main types of non-relational databases**: document-oriented, key-value, wide-column store, and graph store.

Source: THIS TUTORIAL WAS AUTHORED BY DR. VINCENT TRAN, PHD (2020) AND FAITHE WEMPEN (2024) FOR SOPHIA LEARNING. PLEASE SEE OUR **TERMS OF USE**.

TERMS TO KNOW

Document-Oriented Database

A non-relational database that pairs a key with a complex data structure.

Graph Store

A non-relational database that links data through edges, nodes, and properties.

Key-Value Store

A simple non-relational database that pairs each key with a single value.

NoSQL Database

A database that can work with semi-structured or unstructured data by enabling the storing and querying of data outside the traditional structures found in relational databases.

Non-Relational Databases

Databases that do not use the traditional tabular relational model that is used in relational database management systems.

Schema

The structure of a database, including its tables and the relationships between them.

Wide-Column Store

A non-relational database that allows the names and format of columns of its tables to vary from row to row.