

RELATIONAL DATABASES



DEFINITION

A relational database is a type of database that organizes and allows access to data that is interconnected. It is based on the relational model, which is a simple and logical method of representing data in tables. (Oracle, n.d., para. 1)

In this structure, each row represents a record with a unique identifier known as the key. Columns contain the attributes of the data. Usually, each record holds a value for every attribute, making it easy to define relationships between different data points. (Oracle, n.d., para. 1)

EXAMPLES

MySQL
PostgreSQL
SQLite
Oracle Database
Microsoft SQL Server



PERFORMANCE CONSIDERATIONS

TABLE 1

Performance consideration of Relational Databases

FEATURES	RELATIONAL DATABASES
Availability	High
Horizontal Scaling	Low
Vertical Scaling	High
Data Storage	Medium to large data
Performance	Low To Medium
Reliability	High (Acid)
Complexity	Medium (Joins)
Flexibility	Low (Strict-Schema)
Suitability	Suitable For OLTP

Note. Adapted from MongoDB, n.d., Inc. © 2024.

USE CASES

When to Use a Relational Database:

- Consistent Data Structure: Optimal for projects where data structure, size, and access frequency are predictable. (MongoDB, n.d., para. 44)
- Advantages of Normalization: Reduces data duplication and anomalies, which can help avoid future vertical scaling. (MongoDB, n.d., para. 45)
- Crucial Relationships: Best suited for scenarios where the relationships between entities are significant. (MongoDB, n.d., para. 46)
- Extensive Support: Offers a wide range of tools and integration capabilities due to the long history of relational database management systems. (MongoDB, n.d., para. 48)

NON-RELATIONAL DATABASES



DEFINITION

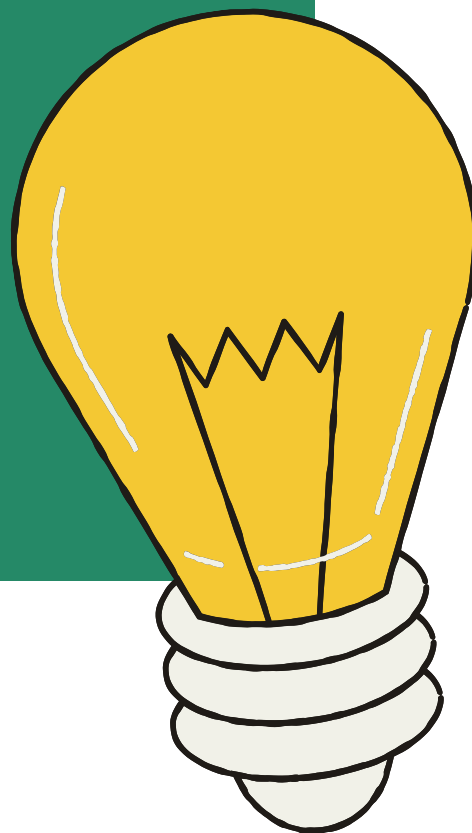
NoSQL, short for "not only SQL," is a type of database that lets you store and access data in a flexible way, not just in tables like traditional databases. Instead, it often uses formats like JSON documents, which allows it to grow quickly and handle large amounts of unstructured data without needing a fixed layout. (IBM, n.d., para. 1)

NoSQL databases are spread out over multiple servers, which helps keep data available and reliable. If one part goes down, the rest can still work. (IBM, n.d., para. 4)

Today, as companies deal with lots of data fast, NoSQL databases are becoming popular for web applications because they are quick, easy to scale, and user-friendly in the age of cloud computing and big data. (IBM, n.d., para. 5)

EXAMPLES

- MongoDB
- Apache Cassandra
- Neo4J
- Redis
- Amazon DynamoDB



PERFORMANCE CONSIDERATIONS

TABLE 2

Performance consideration of Non-Relational Databases

FEATURES	DOCUMENT DATABASE	COLUMN STORE DATABASE	KEY-VALUE STORE DATABASE	GRAPH DATABASE
Performance	High	High	High	Moderate
Availability	High	High	High	High
Flexibility	High	Moderate	High	High
Scalability	High	High	High	Moderate
Complexity	Low	Low	Moderate	High

Note. Adapted from MongoDB, n.d., Inc. © 2024.

ADVANTAGES AND DISADVANTAGES

Advantages

- Flexibility: NoSQL handles unstructured or changing data, ideal for modern apps and big data. (Couchbase Product Marketing, 2023, para. 55)
- Scalability: It can manage large data and lots of traffic easily. (Couchbase Product Marketing, 2023, para. 56)
- Performance: NoSQL works well with big data and complex tasks. (Couchbase Product Marketing, 2023, para. 57)
- Distributed: Data is spread across servers for better speed and scaling. (Couchbase Product Marketing, 2023, para. 58)
- Agile: You can easily add or remove data without changing the whole system. (Couchbase Product Marketing, 2023, para. 59)

Disadvantages

- Limited queries: NoSQL isn't as good for complex searches. (Couchbase Product Marketing, 2023, para. 60)
- No standard language: There's no single query language like SQL, which can make it harder to develop. (Couchbase Product Marketing, 2023, para. 61)
- Fewer tools: It has less support and tools compared to traditional databases. (Couchbase Product Marketing, 2023, para. 64)
- Security: It might not have strong security features like access control or encryption. (Couchbase Product Marketing, 2023, para. 65)

USE CASES

When to Use a Non-Relational Database:

- High Performance: NoSQL databases like Cassandra can read data quickly by using more servers and copying data across them. (Coursera Staff, 2024, para. 50)
- High Availability: NoSQL databases keep copies of data on multiple servers, so even if one server fails, your data stays safe, and the application keeps working. (Coursera Staff, 2024, para. 51)
- Handling Large Data Volumes: NoSQL databases are built to handle huge amounts of unstructured or semi-structured data, making them perfect for big data projects.
- Flexible Data Models: NoSQL is a good option when your data might change over time or doesn't fit into a fixed structure.



REFERENCES

- Oracle Canada. (n.d.). What is a Relational Database (RDBMS)? <https://www.oracle.com/ca-en/database/what-is-a-relational-database/>
- IBM. (n.d.). What is a NoSQL database? <https://www.ibm.com/topics/nosql-databases>
- MongoDB. (n.d.). Relational vs. Non-Relational Databases. <https://www.mongodb.com/resources/compare/relational-vs-non-relational-databases>
- Couchbase Product Marketing. (2023, May 18). Relational vs. Non-Relational Databases: Features and Benefits. <https://www.couchbase.com/blog/relational-vs-non-relational-database/>
- Coursera. (2024, April 19). SQL vs. NoSQL: The Differences Explained + When to Use Each. <https://www.coursera.org/articles/nosql-vs-sql>