Bases de datos relacionales y no relacionales

Javier Andrés Serna Bañol

Enero 2024.

Centro de tecnologías agroindustriales (SENA)

Valle del cauca

Análisis y desarrollo de software

**Tabla de Contenidos**

[¿Qué es una base de datos relacional? 3](#_Toc157070212)

[Diferencias 5](#_Toc157070213)

[Tabla 1. *Cuadro comparativo*. 7](#_Toc157070214)

[Lista de referencias 8](#_Toc157070215)

## ¿Qué es una base de datos relacional?

* Una base de datos relacional “SQL” organiza los datos en filas y columnas, que en conjunto forman una tabla. Los datos normalmente se estructuran en varias tablas, que se pueden unir a través de una clave principal o una clave externa.
* Estos identificadores únicos demuestran las diferentes relaciones que existen entre las tablas, y estas relaciones generalmente se ilustran a través de diferentes tipos de modelos de datos.
* Por ejemplo, imagine que su empresa mantiene una tabla de base de datos con información de clientes, que contiene datos de la empresa a nivel de cuenta. También puede haber una tabla diferente, que describe todas las transacciones individuales que se alinean con esa cuenta. Juntas, estas tablas pueden proporcionar información acerca de las diferentes industrias que compran un producto de software específico.

¿Qué es una base de datos no relacional?

* NoSQL, también conocido como "no solo SQL", "no SQL", es un enfoque para el diseño de bases de datos que permite el almacenamiento y la consulta de datos fuera de las estructuras tradicionales que se encuentran en las bases de datos relacionales. Si bien aún puede almacenar datos que se encuentran dentro de los sistemas de gestión de bases de datos relacionales (RDBMS), los almacena de manera diferente a un RDBMS. La decisión de utilizar una base de datos relacional frente a una no relacional es en gran medida contextual y varía según el caso de uso.
* En lugar de la estructura tabular típica de una base de datos relacional, las bases de datos NoSQL albergan datos dentro de una estructura de datos, como un documento JSON. Dado que este diseño de base de datos no relacional no requiere un esquema, ofrece escalabilidad rápida para gestionar grandes conjuntos de datos normalmente no estructurados.
* NoSQL también es un tipo de base de datos distribuida, lo que significa que la información se copia y almacena en varios servidores, que pueden ser remotos o locales. De esta manera, se garantiza la disponibilidad y la confiabilidad de los datos. Si algunos de los datos se desconectan, el resto de la base de datos puede continuar ejecutándose.

# Diferencias

* A la hora de elegir el desarrollo de cualquier tipo de aplicación o proyecto, definir el tipo de sistema de gestión de datos es fundamental.
* Se tratan de modelos distintos con sus ventajas y desventajas. No podemos decir que uno sea mejor que otro, sino que ambos tienen sus casos de uso.

Vamos a identificar cuáles son las principales diferencias entre las bases de datos relacionales y no relacionales.

**Estructura:** las bases de datos SQL guardan información de manera estructurada, sumando robustez y seguridad, pero restando escalabilidad. Las bases de datos no relacionales guardan información en documentos, lo que aumenta la flexibilidad, pero puede restarle robustez y orden.

**Escalabilidad:** la capacidad para escalar de las bases de datos relacionales es baja en comparación con las NoSQL. Esto es algo que se vuelve más evidente cuando hablamos de grandes volúmenes de datos y altos niveles de concurrencia.

**Flexibilidad:** los proyectos que tienen una estructura sólida y permanente tienden a utilizar bases de datos relacionales, ya que su estructura es robusta. Sin embargo, en caso de que haya que incorporar cambios importantes, esta no se comporta adecuadamente. Por lo contrario, las NoSQL son totalmente flexibles.

**Compatibilidad:** las bases de datos relacionales se pueden utilizar en todo tipo de aplicaciones, por lo que se pueden utilizar para proyectos muy variados. Sin embargo, el diseño libre de las NoSQL las hace más apropiadas para proyectos muy grandes, que pueden crecer rápidamente, pueden tener picos de demanda o cuando las necesidades no se pueden predecir.

**Consistencia:** la estructura de las bases de datos relacionales ofrece una alta integridad y consistencia en los datos. A causa de ello, la velocidad no es una de sus características, algo que sí ofrece NoSQL en detrimento de una menor integridad.

**Agilidad:** las bases de datos SQL son más rápidas en proyectos donde la información debe estar siempre bien estructurada. Sin embargo, si esto no es posible, las NoSQL ofrecerán un mayor rendimiento.

## Tabla 1. *Cuadro comparativo*.

|  |  |  |
| --- | --- | --- |
| ***Aspecto*** | ***Base de Datos Relacional*** | ***Base de Datos No Relacional*** |
| **Estructura de Datos** | Utiliza tablas para organizar los datos | Utiliza varios modelos, como documentos, grafos, clave-valor o columnares. |
| **Esquema** | Esquema fijo y predefinido | Esquema dinámico y flexible |
| **Escalabilidad Vertical** | Mejor para escalabilidad vertical | Mejor para escalabilidad horizontal |
| **Transacciones ACID** | Cumple con propiedades ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad) | Algunas bases de datos no relacionales pueden comprometer algunas propiedades ACID para mejorar el rendimiento y la escalabilidad. |
| **Escrituras vs Lecturas** | Mejor rendimiento en operaciones de lectura | Mejor rendimiento en operaciones de escritura y actualización. |
| **Complejidad del Modelo** | Modelos de datos más complejos | Modelos de datos más simples |
| **Flexibilidad de Esquema** | Menos flexible en términos de cambios en el esquema | Más flexible, permite cambios en el esquema sin afectar la integridad de los datos existentes. |
| **Ejemplos de Tecnologías** | MySQL, PostgreSQL, Oracle | MongoDB, Cassandra, Redis, Neo4j |
| **Casos de Uso Comunes** | Aplicaciones con relaciones y consultas complejas | Aplicaciones con grandes volúmenes de datos, necesidades de escalabilidad horizontal, y cambios frecuentes en el esquema. |
|  |  |  |

# Lista de referencias

<https://www.incentro.com/es-ES/blog/base-de-datos-relacional-vs-no-relacional>

<https://www.ibm.com/mx-es/topics/relational-databases>

<https://www.ibm.com/es-es/topics/nosql-databases>