

Bases de Datos Relacionales vs No Relacionales



Objetivos:

- ◆ Definir las diferencias de las bases de datos relacionales y las no relacionales.
- ◆ Comprender los beneficios de las bases de datos relacionales y no relacionales.
- ◆ Comprender las desventajas de las bases de datos relacionales y no relacionales.



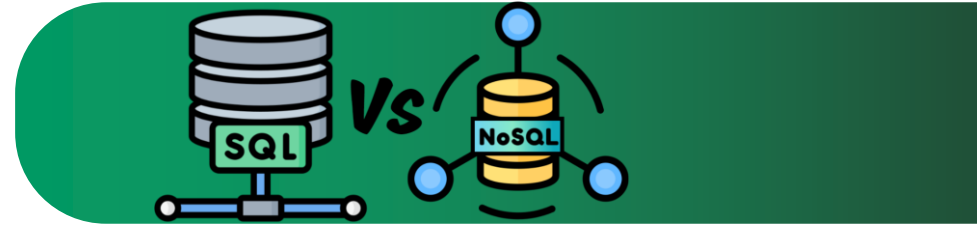
Introducción:

Hay muchas bases de datos utilizadas hoy en día en la industria. Algunas son bases de datos SQL, otras son bases de datos NoSQL. La base de datos convencional es un sistema de base de datos SQL que utiliza un modelo relacional tabular para representar los datos y su relación. La base de datos NoSQL es la base de datos más nueva que proporciona un mecanismo para el almacenamiento y la recuperación de datos que no sea el modelo de relaciones tabulares utilizado en las bases de datos relacionales.

SQL es la mejor base de datos para usar para datos relacionales, especialmente cuando la relación entre conjuntos de datos está bien definida y es altamente navegable. También es mejor para evaluar la integridad de los datos.

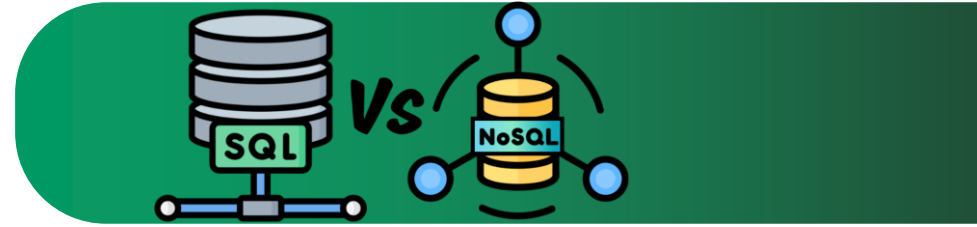
NoSQL es la mejor base de datos para usar para grandes cantidades de datos o conjuntos de datos en constante cambio. También es mejor usarlo cuando tiene modelos de datos flexibles o necesidades que no encajan en el modelo relacional.

A continuación, algunas diferencias entre las bases de datos relacionales y las no relacionales.



Cuadro Comparativo de SQL y NoSQL

Características	SQL	NoSQL
Modelo	Relacional	No relacional
Tipo	Basada en tablas	Varios modelos de bases de datos, como bases de datos orientadas a documentos, gráficas, clave-valor y columnas
Almacenamiento de datos	Los datos individuales (por ejemplo, "Título del libro") se almacenan en filas de una tabla y se asignan a atributos específicos (por ejemplo, "Autor", "Año de publicación", etc.). Los conjuntos de datos se almacenan en tablas separadas y el sistema los fusiona durante búsquedas complejas.	En lugar de tablas, las bases de datos NoSQL utilizan documentos, valores clave, gráficos o columnas.
Esquemas	El tipo de datos y la estructura se definen de antemano. Para almacenar nueva información, toda la base de datos debe personalizarse (y desconectarse).	Flexible. Se pueden agregar inmediatamente nuevos conjuntos de datos. Los datos estructurados, semiestructurados y no estructurados se pueden almacenar juntos; no es necesaria ninguna conversión previa.
Estructura de datos	Mantienen una estructura definida previamente	Su estructura puede ser dinámica
Rendimiento	Utiliza índices para grandes cantidades de datos. Las consultas, los índices y la estructura deben optimizarse para mejorar el rendimiento de los sistemas SQL.	Las bases de datos NoSQL son mucho más potentes porque utilizan servidores en la nube y clústeres de hardware.
Consistencia	Se diseñan para tener la menor redundancia de datos posible	Permite tener redundancia de datos
Modelo de desarrollo	Software de código abierto (como MySQL) o de pago (base de datos Oracle)	Software de código abierto o de pago
Disponibilidad	Para aplicaciones que requieren datos consistentes sin dar posibilidades a errores.	Para aplicaciones que requieren grandes cantidades de datos a consultar
Escala	Escalables verticalmente: aumentar las prestaciones del servidor. Un solo servidor debe admitir todo el sistema de base de datos, lo que provoca una caída en el rendimiento cuando los volúmenes de datos son altos.	Escalable horizontalmente: agregar más servidores. Cada administrador puede agregar nuevos servidores básicos y en la nube, y la base de datos NoSQL envía automáticamente los datos a todos los servidores.



Cuadro Comparativo de Beneficios de SQL Vs NoSQL

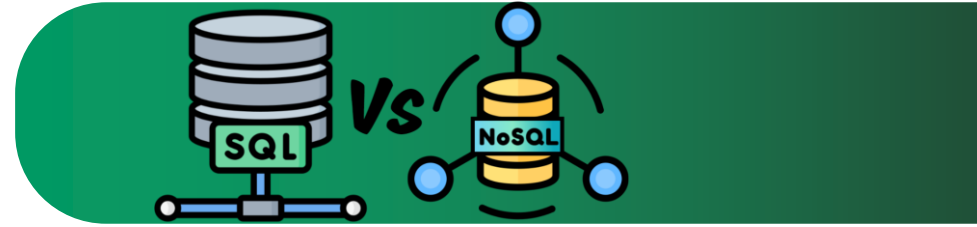
Beneficios

SQL

- ◆ Alta velocidad: mediante las consultas SQL, el usuario puede recuperar de forma rápida y eficiente una gran cantidad de registros de una base de datos.
- ◆ No se necesita codificación: en el SQL estándar, es muy fácil administrar el sistema de base de datos. No requiere una cantidad sustancial de código para administrar el sistema de base de datos.
- ◆ Estándares bien definidos: las bases de datos SQL que están siendo utilizadas por ISO y ANSI utilizan desde hace mucho tiempo.
- ◆ Portabilidad: SQL se puede utilizar en computadoras portátiles, PC, servidores e incluso algunos teléfonos móviles.
- ◆ Lenguaje interactivo: SQL es un lenguaje de dominio utilizado para comunicarse con la base de datos. También se utiliza para recibir respuestas a las preguntas complejas en segundos.
- ◆ Vista de datos múltiples: Usando el lenguaje SQL, los usuarios pueden crear diferentes vistas de la estructura de la base de datos.

NoSQL

- ◆ Modelo de datos flexible: las bases de datos NoSQL son altamente flexibles, ya que pueden almacenar y combinar cualquier tipo de datos, tanto estructurados como no estructurados, a diferencia de las bases de datos relacionales que pueden almacenar datos solo de manera estructurada.
- ◆ Evolución del modelo de datos: las bases de datos NoSQL le permiten actualizar dinámicamente el esquema para evolucionar con los requisitos cambiantes, al tiempo que garantiza que no cause interrupciones ni tiempo de inactividad en su aplicación.
- ◆ Escalabilidad elástica: las bases de datos NoSQL pueden escalar para adaptarse a cualquier tipo de crecimiento de datos mientras se mantienen bajo costo.
- ◆ Alto rendimiento: las bases de datos NoSQL están diseñadas para un gran rendimiento, medido en términos de rendimiento (es una medida del rendimiento general) y latencia (es el retraso entre la solicitud y la respuesta real).
- ◆ Código abierto: las bases de datos NoSQL no requieren costosas tarifas de licencia y pueden ejecutarse en hardware económico, lo que hace que su implementación sea rentable.



Cuadro Comparativo de Beneficios de SQL Vs NoSQL

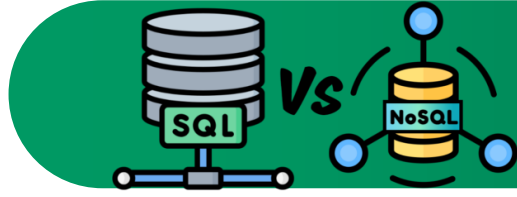
Desventajas

SQL

- ◆ Interfaces: aunque no hay codificación compleja involucrada, el proceso de interfaz es complejo.
- ◆ Interfaz compleja: dado que SQL tiene una estructura compleja, se vuelve difícil para ciertos usuarios acceder a él.
- ◆ Implementación: ciertas bases de datos implementan extensiones propietarias de SQL estándar para garantizar el bloqueo del proveedor.
- ◆ Sólo control parcial: dado que hay ciertas reglas y condiciones ocultas, los programadores que usan SQL no tienen poder sobre la base de datos.
- ◆ Gasto: los gastos involucrados en la operación de SQL son demasiados, lo que dificulta la incorporación del proveedor.

NoSQL

- ◆ Enfoque estrecho: las bases de datos NoSQL tienen un enfoque muy estrecho, ya que están diseñadas principalmente para el almacenamiento, pero proporcionan muy poca funcionalidad. Las bases de datos relacionales son una mejor opción en el campo de la gestión de transacciones que NoSQL.
- ◆ Código abierto: NoSQL es una base de datos de código abierto. Todavía no existe un estándar confiable para NoSQL. En otras palabras, es probable que dos sistemas de bases de datos sean desiguales.
- ◆ Desafío de gestión: el propósito de las herramientas de big data es hacer que la gestión de una gran cantidad de datos sea lo más simple posible. Pero no es tan fácil.
- ◆ GUI no está disponible: las herramientas de modo GUI para acceder a la base de datos no están disponibles de manera flexible en el mercado.
- ◆ Copia de seguridad: la copia de seguridad es un gran punto débil para algunas bases de datos NoSQL como MongoDB. MongoDB no tiene un enfoque para la copia de seguridad de datos de manera consistente.



Bases de Datos Relacionales vs No Relacionales



Infografía:

- ◆ <https://www.ionos.com/digitalguide/hosting/technical-matters/nosql/>
- ◆ <https://www.ibm.com/cloud/blog/sql-vs-nosql>