



TIC



▶ TALENTO TECH

Lección 2: Bases de datos no relacionales



Tiempo de ejecución: 2 horas

PLANTEAMIENTO DE LA SESIÓN

MATERIALES

En esta sesión se explorarán las bases de datos no relacionales o NoSQL. A diferencia de las bases de datos relacionales, no existe un modelo entidad- relación sino que existen muchos tipos de modelos dependiendo de la base de datos y lenguajes de consulta diferentes al estándar SQL. Al finalizar la sesión se comprenderán los conceptos fundamentales de bases de datos NoSQL y se identificarán los diferentes tipos de bases de datos NoSQL.



TIC



CONTENIDO

Las bases de datos NoSQL (not only SQL) son una alternativa a las bases de datos relacionales. Se caracterizan por ser flexibles en la forma de almacenar información y por que pueden escalar a grandes cantidades de datos fácilmente. Esto las hace muy buenas almacenando datos no estructurados. Las principales diferencias con las bases de datos relacionales son:

1. Las bases de datos emplean modelos de datos diferentes al modelo entidad-relación. Por ejemplo existen bases de datos llave-valor, bases de datos documentales, bases columnares o grafales, que almacenan los datos y se consultan de formas muy diferentes a las bases de datos relacionales.



TIC

2. Las bases de datos NoSQL tienen una escalabilidad más sencilla que las bases de datos SQL. Esto se debe a que es más fácil realizar un almacenamiento de datos distribuido por la forma en la que están ordenados.

3. Al no tener una estructura rígida de entidades (tablas, relaciones y modelos predefinidos) las bases de datos no relacionales pueden emplearse en aplicaciones con requerimientos especializados.

Este tipo de bases de datos son ideales para emplearse en casos en donde se tienen muchos datos no estructurados, como es el caso de almacenar documentos, imágenes, videos, archivos de audio y todo tipo de información diferente a tablas (datos estructurados).



TIC



También pueden ser empleadas para casos en los que se requieren accesos rápidos y masivos, como por ejemplo juegos en línea, aplicaciones de redes sociales que necesitan enviar mensajes y multimedia en tiempo real y sistemas de mensajería instantánea cuya estructura debe ser flexible y debe poderse consultar rápidamente (un mensaje puede contener videos, fotos, enlaces, ubicaciones, stickers, sistemas de pagos y muchas cosas más).

Otro caso de uso es en los sistemas de internet de las cosas o IoT en donde los dispositivos conectados a internet necesitan consultar datos de forma rápida y pueden existir muchos dispositivos reportando diferentes datos que pertenecen a ciertos grupos o canales de información..



Es común encontrar uso de bases de datos no relacionales en sistemas de streaming de audio y de video, con lo que se gestiona información de usuarios, consultas, como, por ejemplo, un usuario quiere ver determinada película o escuchar cierta canción, o incluso en plataformas comunes que muestran videos en demanda.

Si bien las bases de datos NoSQL ofrecen muchas ventajas, también tienen algunas desventajas que deben ser consideradas antes de elegir una:

1. Falta de estandarización: No existe un lenguaje de consulta universal para las bases de datos NoSQL. Cada tipo de base de datos NoSQL tiene su propio lenguaje de consulta, lo que puede dificultar el aprendizaje y la comparación entre diferentes sistemas.



TIC





TIC



2. **Soporte técnico:** Las bases de datos NoSQL son generalmente de código abierto y no tienen el mismo nivel de soporte técnico que las bases de datos relacionales comerciales. Esto significa que puedes tener que resolver los problemas por tu cuenta o buscar ayuda en la comunidad.

3. **ACID:** Algunas bases de datos NoSQL no ofrecen las mismas garantías de atomicidad, consistencia, aislamiento y durabilidad (ACID) que las bases de datos relacionales. Esto puede ser un problema para aplicaciones que requieren un alto nivel de integridad de datos.



TIC

4. **Seguridad:** La seguridad puede ser una preocupación con las bases de datos NoSQL, especialmente si se utilizan en aplicaciones que manejan datos sensibles. Es importante elegir una base de datos NoSQL que tenga características de seguridad sólidas y deben seguirse buenas prácticas de seguridad de la información en el desarrollo de sistemas conectados a estas bases de datos.

5. **Complejidad:** La gestión de bases de datos NoSQL puede ser más compleja que la gestión de bases de datos relacionales. Esto se debe a que las bases de datos NoSQL son más flexibles y requieren más configuración y ajuste.





6. Complejidad en las consultas: Cuando una base de datos crece, las consultas tienden a tener muchas alternativas y opciones a tener en cuenta. Por ejemplo, en una colección de documentos cuya estructura es variable, el desarrollo de consultas puede tener errores que aparecen a lo largo del uso de la aplicación. Así mismo, el no tener ciertas estructuras estandarizadas, dificulta algunos procesos en el desarrollo. Otra dificultad general es que las bases de datos no SQL no están diseñadas para tener operaciones de unión (join) por lo que tratar de hacer un join va a ser un proceso complejo que requiere programar muchas reglas.

7. Falta de herramientas: al ser una tecnología relativamente nueva, no hay tantas herramientas para el desarrollo como en el caso de las bases de datos relacionales. Sin embargo si existen algunas herramientas que pueden apoyar el uso de tales sistemas.



TIC



Motores de bases de datos NoSQL



TIC



Existen varios tipos de bases de datos NoSQL, cada una de ellas con ventajas y aplicaciones en ciertos campos específicos. Algunos de ellos son:

Bases de datos de clave-valor:

Son sistemas cuyas consultas se hacen a través de claves que retornan un valor almacenado. La complejidad de las consultas es baja y la velocidad de acceso es muy alta. Al ser unas bases de datos simples, son fáciles de implementar y de escalar. También son rápidas para realizar operaciones de lectura o de escritura. Sin embargo, no son adecuadas cuando los datos tienen relaciones entre ellos, ya que, encontrar un dato relacionado con otro es costoso computacionalmente. Algunos de los motores de bases de datos clave-valor más populares son Redis, Memcached y Amazon DynamoDB.



BASES DE DATOS DOCUMENTALES

Son bases de datos diseñadas para almacenar conjuntos de información en formato JSON o XML. Estos datos conservan una estructura (marcas u objetos de JSON) sin embargo pueden ser diferentes las estructuras internas de cada archivo. Al tener cierta estructura se pueden realizar consultas complejas, como por ejemplo encontrar relaciones entre algunos datos. También pueden escalarse fácilmente por la forma en la que guardan los datos. Entre los programas mas populares se encuentra MongoDB, couchDB y DocumentDB



TIC





TIC



BASES DE DATOS DE GRAFOS

Un grafo es una estructura de datos que consiste en un conjunto de nodos y un conjunto de aristas que conectan los nodos. Los nodos pueden representar entidades como personas, lugares o cosas, y las aristas pueden representar relaciones entre esas entidades. Hay datos que se pueden almacenar mejor como grafos.

Tal es el caso de una red social, en donde los nodos pueden representar usuarios y las aristas las relaciones de amistad. También se encuentran en las bases de datos de mapas de carreteras, en donde ciudades o puntos de interés son nodos y las aristas son las vías que interconectan esos lugares.



TIC



La ventaja de usar grafos para almacenar información, es que se pueden representar relaciones complejas de forma intuitiva y natural, como el caso de los lugares y vías. Empleando algoritmos de redes es posible encontrar rutas que conectan un nodo con otro y que tienen que pasar por diferentes nodos en el camino, esto ayuda a resolver problemas como ¿Cuál es la ruta mas corta o mas rápida para llegar de un lugar a otro?

En el caso de las bases de datos de grafos, se almacenan nodos y relaciones como es el caso de los sistemas Neo4j, TinkerPop y amazon neptune.



TIC



BASES DE DATOS DE COLUMNAS

Son sistemas que almacenan los datos en formato columna, similar a como funcionan las hojas de cálculo. En una base de datos, una columna es un conjunto de valores relacionados que se almacenan juntos. Cada columna tiene un nombre y un tipo de dato, como cadena de caracteres, número entero o fecha. Por ejemplo, en una tabla de clientes, podría haber una columna para el nombre del cliente, otra para la dirección y otra para el número de teléfono.

Las bases de datos de columnas están diseñadas para almacenar grandes cantidades de datos de forma eficiente. En lugar de almacenar los datos en filas como las bases de datos relacionales tradicionales, los datos se almacenan en columnas.



Esto permite un acceso más rápido a los datos para ciertos tipos de consultas, especialmente aquellas que solo necesitan un subconjunto de las columnas, es decir, el solo hecho de cambiar la forma en la que se representan los datos en memoria da un incremento notable del rendimiento de almacenar los datos y consultarlos. Otra de las ventajas de tener los datos como columnas, es que se pueden comprimir, requiriendo menos espacio para guardar grandes cantidades de datos.

Como ejemplo de motores de bases de datos de columnas se tiene Cassandra y Hbase. Un ejemplo interesante del uso de cassandra es para el almacenamiento y consulta de sistemas de comunicación en video juegos, en donde audio y texto se debe compartir en tiempo real (datos del mismo tipo almacenado en formato de columnas, como son chats, el orden de llegada es el mismo en el que se muestran los eventos).



TIC



BASE DE DATOS ORIENTADA A OBJETOS



TIC



Las bases de datos orientadas a objetos (BDOO) son un tipo de base de datos que almacena datos en forma de objetos, similar a como se programan en lenguajes orientados a objetos. Los objetos encapsulan datos y métodos, lo que permite un modelado más natural de datos complejos y relaciones entre ellos. Existen diversos motores de BDOO, como:

- Objectivity/DB: BDOO pura, robusta y escalable.
- ZODB: BDOO de código abierto, ligera y fácil de usar.
- Versant: BDOO escalable con soporte para transacciones ACID.
- ObjectDB: BDOO multiplataforma con enfoque en rendimiento.



Las bases de datos orientadas a objetos se emplean en donde se requiere modelado de datos con sus métodos e interfaces, y un rendimiento en donde, en las operaciones puedan invocar métodos. Por ejemplo, sistemas de información geográfica en donde una ubicación puede tener métodos (escalamiento, cambios de proyecciones, entre otros).

Sin embargo, las bases de datos orientadas a objetos pueden ser complejas, las consultas tienden a ser más complicadas que utilizar SQL. Tampoco existe un estándar de consultas para tales bases de datos.

Existen también algunas bases de datos de funcionalidades especializadas, como es el caso de las bases de datos de series temporales, las bases de datos espaciales, que contienen columnas con geometrías espaciales y operaciones asociadas a topología y bases de datos multivalor que tienen asociaciones de múltiples valores con una misma clave.



TIC

