

Tarea 1

Modelos de Bases de Datos No Relacionales (NoSQL)

En los últimos años, el crecimiento masivo de datos generados por aplicaciones web, redes sociales, dispositivos móviles y sistemas distribuidos ha impulsado el desarrollo de nuevos modelos de bases de datos distintos al modelo relacional tradicional. Estos sistemas son conocidos como **NoSQL (Not Only SQL)**, y se caracterizan por ofrecer mayor flexibilidad, escalabilidad horizontal y rendimiento en entornos de grandes volúmenes de información.

A diferencia de las bases de datos relacionales, que organizan la información en tablas con filas y columnas, los modelos NoSQL utilizan diferentes estructuras de almacenamiento según el tipo de datos y la necesidad del sistema. Entre los principales modelos se encuentran: orientado a objetos, clave-valor, documental, de grafos y columnar.

1. Modelo Orientado a Objetos

El modelo orientado a objetos almacena la información en forma de objetos, similares a los utilizados en lenguajes de programación orientados a objetos como Java o C++. En este modelo, los datos y sus métodos pueden integrarse dentro de una misma estructura.

Descripción

Cada objeto contiene atributos (datos) y puede incluir relaciones con otros objetos. Esto permite representar estructuras complejas de manera más natural que en un esquema relacional.

Ventajas

- Representación más directa de estructuras complejas.
- Menor necesidad de transformar datos entre la aplicación y la base de datos.
- Buena integración con lenguajes orientados a objetos.

Desventajas

- Menor estandarización que SQL.
- Puede presentar problemas de interoperabilidad con otros sistemas.

Casos de uso

Sistemas de ingeniería, aplicaciones científicas, modelado 3D o sistemas donde los datos son altamente complejos y estructurados.

2. Modelo Clave-Valor

Este es el modelo más simple dentro de NoSQL. Funciona mediante pares donde una clave única se asocia con un valor específico.

Descripción

Cada registro se almacena como una clave y su respectivo valor, el cual puede ser un dato simple o una estructura más compleja.

Ventajas

- Alta velocidad de lectura y escritura.
- Diseño sencillo.
- Ideal para almacenamiento en caché.

Desventajas

- No permite consultas complejas.
- No maneja relaciones entre datos.

Casos de uso

Sistemas de caché, sesiones de usuario, almacenamiento temporal de datos y aplicaciones con alto tráfico web.

3. Modelo Documental

El modelo documental almacena la información en documentos, generalmente en formato JSON o BSON.

Descripción

Cada documento contiene datos estructurados que pueden variar en campos y estructura. No es obligatorio que todos los registros tengan los mismos atributos.

Ventajas

- Gran flexibilidad en la estructura de los datos.

- Escalabilidad horizontal.
- Ideal para aplicaciones web modernas.

Desventajas

- Puede haber redundancia de datos.
- No siempre garantiza integridad referencial como en el modelo relacional.

Casos de uso

Aplicaciones web, comercio electrónico, sistemas de gestión de contenido (CMS), APIs y microservicios.

4. Modelo de Grafos

El modelo de grafos se basa en nodos y relaciones (aristas), lo que permite representar conexiones entre datos de manera eficiente.

Descripción

Los nodos representan entidades y las aristas representan relaciones entre ellas. Este modelo está optimizado para consultas que requieren recorrer relaciones complejas.

Ventajas

- Excelente desempeño en consultas de relaciones complejas.
- Representación natural de redes.

Desventajas

- No es ideal para datos tabulares tradicionales.
- Puede ser más complejo de diseñar.

Casos de uso

Redes sociales, sistemas de recomendación, análisis de fraudes y mapas de relaciones.

5. Modelo Columnar

El modelo columnar organiza los datos por columnas en lugar de filas completas.

Descripción

En lugar de almacenar una fila completa como en el modelo relacional, se almacenan columnas individuales, lo que optimiza ciertas consultas analíticas.

Ventajas

- Alto rendimiento en consultas analíticas.
- Escalabilidad.
- Ideal para Big Data.

Desventajas

- No es óptimo para transacciones pequeñas y frecuentes.
- Mayor complejidad en diseño.

Casos de uso

Análisis de grandes volúmenes de datos, sistemas de inteligencia de negocios y procesamiento masivo de información.

Referencias

- [1] R. Elmasri y S. B. Navathe, *Fundamentos de Sistemas de Bases de Datos*, 7ª ed. México: Pearson Educación, 2016.
- [2] A. Silberschatz, H. F. Korth y S. Sudarshan, *Conceptos de Sistemas de Bases de Datos*, 6ª ed. México: McGraw-Hill Interamericana, 2011.
- [3] C. J. Date, *Introducción a los Sistemas de Bases de Datos*, 8ª ed. México: Pearson Educación, 2004.