

Tarea 1. Modelos de base de datos.

Villeda Tlecuitl José Eduardo

August 20, 2024

1 Modelo orientado a objetos

1.1 Descripción

Es Una base de datos orientada a objetos que guarda información en forma de objetos, tal como se hace en la programación orientada a objetos. A diferencia de las bases de datos tradicionales que usan tablas, estas bases utilizan objetos como unidades básicas de almacenamiento.

Estos ofrecen una forma más natural de almacenar datos, modelándolos como objetos con propiedades y comportamientos. Lo que las hace ideales para gestionar información compleja y relaciones entre datos, superando las limitaciones de las bases de datos relacionales.

1.2 Ventajas.

1. Facilidad para manejar versiones y control de cambios: Las bases de datos orientadas a objetos facilitan el control de versiones de los objetos, permitiendo a las aplicaciones manejar mejor los cambios en los datos a lo largo del tiempo.
2. Mejor manejo de relaciones complejas: Las relaciones entre objetos (como herencia o composición) son naturales en este tipo de bases de datos y pueden ser gestionadas de forma más directa sin necesidad de unir tablas.
3. Mayor cohesión de datos y comportamientos: Los datos y las operaciones relacionadas con ellos están agrupados en los mismos objetos, lo que permite una mayor cohesión y encapsulación, mejorando la mantenibilidad del código.
4. Polimorfismo: El polimorfismo permite que objetos de diferentes clases puedan ser tratados como si fueran de una misma clase, lo que aumenta la flexibilidad.
5. Mayor expresividad: Los lenguajes de consulta de las BDOO suelen ser más expresivos que los de las bases de datos relacionales, lo que facilita la realización de consultas complejas.

1.3 Desventajas

1. Compatibilidad limitada con aplicaciones existentes: Muchas aplicaciones, especialmente las más antiguas, están diseñadas para funcionar con bases de datos relacionales. La conversión o la integración de estas aplicaciones con una base de datos orientada a objetos puede requerir una reescritura significativa del código.
2. Falta de experiencia generalizada: Hay menos profesionales capacitados en el uso de bases de datos orientadas a objetos en comparación con las bases de datos relacionales, lo que puede dificultar encontrar personal con experiencia.
3. Escalabilidad limitada en algunos casos: Para sistemas extremadamente grandes con una alta demanda de transacciones, las bases de datos orientadas a objetos pueden no escalar tan eficientemente como las bases de datos relacionales.
4. Soporte limitado: Aunque están creciendo en popularidad, las bases de datos orientadas a objetos no son tan ampliamente soportadas o utilizadas como las bases de datos relacionales. Esto puede limitar las herramientas disponibles y el soporte técnico.

5. Estandarización: No existe un estándar universal para las BDOO, lo que dificulta la portabilidad de las aplicaciones.

2 Modelos NoSQL

2.1 Descripción

Un modelo NoSQL es un tipo de modelo de base de datos que no sigue la estructura relacional tradicional (esquemas rígidos y uso de tablas), sino que está diseñado para almacenar, procesar y acceder a grandes volúmenes de datos no estructurados o semi-estructurados de una manera flexible y escalable.

2.2 Ventajas

1. **Flexibilidad:** Las bases de datos NoSQL generalmente ofrecen esquemas flexibles que permiten un desarrollo más rápido y más iterativo. El modelo de datos flexible hace que las bases de datos NoSQL sean ideales para datos semiestructurados y no estructurados.
2. **Escalabilidad:** Las bases de datos NoSQL generalmente están diseñadas para escalar horizontalmente usando clústeres distribuidos de hardware, en lugar de escalar añadiendo servidores caros y sólidos.
3. **Altamente funcionales:** Las bases de datos NoSQL proporcionan API altamente funcionales y tipos de datos que están diseñados específicamente para cada uno de sus respectivos modelos de datos.
4. **Alto rendimiento:** Las bases de datos NoSQL están optimizadas para modelos de datos y patrones de acceso específicos. Esto permite un mayor rendimiento que si intentara lograr una funcionalidad similar con bases de datos relacionales.
5. **Alta Disponibilidad y Tolerancia a Fallos:** Los sistemas NoSQL están diseñados para ser distribuidos, lo que proporciona alta disponibilidad y recuperación rápida en caso de fallos.

2.3 Desventajas

1. **Seguridad Limitada:** Algunas bases de datos NoSQL no tienen capacidades de seguridad avanzadas (como control de acceso basado en roles) tan robustas como las bases de datos relacionales.
2. **Falta de Estándares Universales:** A diferencia de SQL, NoSQL no tiene un estándar de lenguaje universal para consultas, lo que puede generar incompatibilidades entre sistemas.
3. **Administración Compleja en Grandes Implementaciones:** Aunque NoSQL es escalable, administrar un sistema distribuido puede ser complejo y puede requerir un esfuerzo adicional para mantener la coherencia y la disponibilidad.
4. **Duplicación de Datos:** En muchos sistemas NoSQL (como las bases de datos de documentos o clave-valor), es común duplicar datos para mejorar el rendimiento de las consultas, lo que puede resultar en un uso ineficiente del almacenamiento y la necesidad de gestionar la coherencia entre los duplicados.
5. **Menor Madurez:** Algunas soluciones NoSQL pueden no estar tan maduras o tan optimizadas como las bases de datos relacionales, lo que podría resultar en problemas de estabilidad o soporte.

2.4 Casos de uso

Puede usar bases de datos NoSQL para crear una amplia variedad de aplicaciones móviles, de Internet of Things (Internet de las cosas (IoT)), de juegos y web de alto rendimiento que proporcionan excelentes experiencias de usuario a escala.

- **Aplicaciones de alta disponibilidad:** Las bases de datos NoSQL distribuidas son excelentes para crear aplicaciones de alta disponibilidad y baja latencia para mensajería, redes sociales, uso compartido de archivos y más. Por ejemplo, Snapchat tiene más de 290 millones de usuarios que envían miles de millones de imágenes y mensajes de video a diario. Utiliza sistemas de bases de datos NoSQL para reducir la latencia media del envío de mensajes en un 20
- **Administración de datos en tiempo real:** Puede ofrecer recomendaciones en tiempo real, personalización y experiencias de usuario mejoradas con bases de datos NoSQL. Por ejemplo, Disney+ ofrece su amplia biblioteca de contenido digital a más de 150 millones de suscriptores mediante la tecnología de base de datos NoSQL.
- **Seguridad en la nube:** Puede usar bases de datos de grafos para descubrir rápidamente relaciones complejas en sus datos. Por ejemplo, Wiz reimaginó la seguridad en la nube como un gráfico con Amazon Neptune. Wiz ayuda a sus clientes a mejorar su postura de seguridad al identificar y corregir rápidamente los riesgos más críticos.

3 Base de datos clave-valor

El modelo clave-valor es uno de los tipos más simples y fundamentales de bases de datos NoSQL. Almacena datos como pares clave-valor, donde cada valor se asocia a una clave única. Una forma de imaginar esta base de datos es como un diccionario donde cada palabra (clave) tiene una definición (valor). Este modelo es extremadamente eficiente para búsquedas rápidas y es especialmente útil en aplicaciones que requieren acceso rápido a datos mediante una clave específica.

3.1 Ventajas

1. **Fácil de Implementar:** Su simplicidad hace que sea fácil de implementar y gestionar en comparación con modelos de datos más complejos.
2. **Alto Rendimiento en Aplicaciones Distribuidas:** Las bases de datos clave-valor están diseñadas para distribuirse entre varios servidores, manteniendo un rendimiento elevado incluso a gran escala.
3. **Bajo Costo de Mantenimiento:** Al no tener un esquema rígido, el modelo clave-valor requiere menos esfuerzo para mantener en comparación con los sistemas relacionales.
4. **Bajo Consumo de Recursos:** Al ser un modelo simple y ligero, las bases de datos clave-valor suelen consumir menos recursos en comparación con bases de datos relacionales.
5. **Soporte Multiplataforma:** Las bases de datos clave-valor suelen ser compatibles con múltiples lenguajes de programación y plataformas, facilitando su integración en diferentes entornos tecnológicos.

3.2 Desventajas

1. **Falta de Herramientas Avanzadas de Gestión de Datos:** En comparación con las bases de datos relacionales, las bases de datos clave-valor suelen tener menos herramientas avanzadas para la gestión y el monitoreo de datos.
2. **Duplicación de Datos:** A menudo es necesario duplicar datos en varias claves para optimizar el rendimiento de las consultas, lo que puede aumentar el uso del almacenamiento y complicar la coherencia.
3. **Manejo Complejo de Versiones de Datos:** En sistemas donde los valores cambian frecuentemente, gestionar versiones de datos puede ser complicado, especialmente sin un esquema o mecanismo estructurado para controlar versiones.
4. **Limitaciones en Consultas Analíticas:** No está diseñado para consultas analíticas complejas o el análisis en profundidad de datos. Las consultas más allá de buscar por clave pueden ser difíciles de implementar.

5. **No Adecuado para Consultas Complejas:** El modelo clave-valor no soporta consultas basadas en múltiples criterios o búsquedas dentro de los valores; las consultas están limitadas a las claves.

3.3 Casos de uso

- **IoT (Internet de las Cosas):** Almacenan datos de sensores, eventos y comandos de dispositivos conectados.
- **Sistemas de mensajería:** Utilizadas como brokers de mensajes para almacenar y entregar mensajes entre diferentes aplicaciones.
- **Juegos en línea:** Manejan datos de estado de los juegos, puntajes, logros y otros datos de los jugadores en tiempo real.
- **Caché de Datos:** Para almacenar datos temporales y acceder a ellos rápidamente, como sesiones de usuario o resultados de consultas frecuentes.
- **Gestión de Sesiones:** Guardar el estado de la sesión de los usuarios en aplicaciones web.

4 Bases de datos grafos

Un modelo de base de datos de grafos es un tipo de base de datos NoSQL diseñado para representar y gestionar datos interrelacionados de manera eficiente. En lugar de usar tablas o documentos, los datos se organizan en nodos (entidades) y aristas (relaciones), donde los nodos representan objetos del mundo real y las aristas definen cómo estos objetos están conectados entre sí. Este enfoque es ideal para analizar redes complejas y explorar patrones en datos interconectados, como redes sociales, mapas de rutas o sistemas de recomendación.

4.1 Ventajas

1. **Soporte para Relaciones Polimórficas:** Los grafos permiten la representación de diferentes tipos de relaciones entre los mismos nodos, lo que facilita la modelización de escenarios complejos.
2. **Consulta en Tiempo Real de Redes Dinámicas:** El modelo de grafos es especialmente útil en sistemas donde las relaciones cambian frecuentemente y se necesita análisis en tiempo real.
3. **Escalabilidad:** Las bases de datos de grafos modernas pueden escalar horizontalmente para manejar grandes volúmenes de datos distribuidos en diferentes nodos.
4. **Mejores Análisis de Redes:** El modelo de grafos es ideal para análisis de redes complejas, como detección de comunidades, centralidad de nodos y análisis de influencias.
5. **Modelado Natural de Datos Conectados:** Los grafos proporcionan una representación intuitiva de datos altamente interconectados, como redes sociales, rutas de transporte o redes de telecomunicaciones.

4.2 Desventajas

1. **Problemas de Rendimiento en Redes Enormes:** Aunque están optimizados para consultas de relaciones, el rendimiento puede disminuir en redes extremadamente grandes sin un diseño cuidadoso.
2. **Escasez de Desarrolladores con Experiencia:** Existe una menor cantidad de desarrolladores especializados en bases de datos de grafos en comparación con otros tipos de bases de datos, lo que puede dificultar la contratación.
3. **Falta de Estándares de Interoperabilidad:** Existe menos estandarización entre las bases de datos de grafos, lo que puede dificultar la interoperabilidad entre diferentes sistemas de grafos.
4. **Alto Consumo de Memoria:** Los grafos grandes pueden consumir grandes cantidades de memoria, especialmente cuando los nodos y relaciones tienen muchas propiedades.

5. **Mayor Costo Operativo en Infraestructura Distribuida:** Las bases de datos de grafos distribuidas a gran escala pueden requerir una infraestructura más compleja y costosa, así como un mantenimiento más especializado.

4.3 Casos de uso

- **Análisis de fraude:** Detectar patrones de fraude al analizar transacciones y relaciones entre cuentas.
- **Redes Sociales:** Representar y analizar relaciones entre usuarios, como amigos, seguidores, grupos, etc.
- **Optimización de Rutas y Redes de Transporte:** Encontrar rutas más cortas o optimizar redes de transporte basadas en las relaciones entre nodos (puntos de conexión).
- **IoT:** Analizar la conectividad entre dispositivos y sensores.
- **Sistemas de recomendación:** Sugerir productos, películas o contenido basado en el comportamiento del usuario y las relaciones entre elementos.

4.4 El modelo de bases de datos documentales

Las bases de datos documentales son un tipo de base de datos NoSQL que almacenan datos en documentos. Estos documentos son estructuras de datos flexibles y semiestructuradas, similares a los objetos JSON (JavaScript Object Notation) o BSON (Binary JSON). A diferencia de las bases de datos relacionales, que imponen un esquema rígido, las bases de datos documentales permiten que los documentos tengan diferentes estructuras dentro de una misma colección.

4.5 Ventajas

1. **Flexibilidad en el Esquema:** Los documentos pueden tener estructuras variables, lo que facilita la evolución del modelo de datos sin necesidad de modificar la estructura subyacente.
2. **Alta Disponibilidad y Tolerancia a Fallos:** Muchas bases de datos documentales son distribuidas, lo que les permite mantener la alta disponibilidad y recuperarse rápidamente de fallos.
3. **Almacenamiento Agnóstico de Datos:** Los documentos pueden contener datos en una variedad de formatos y tipos, lo que les permite almacenar contenido diverso sin tener que transformarlo.
4. **Fácil de usar:** Su modelo de datos es intuitivo y se integra bien con lenguajes de programación modernos.
5. **Escalabilidad:** Pueden manejar grandes volúmenes de datos y escalar horizontalmente.

4.6 Desventajas

1. **Mayor Tamaño de Almacenamiento:** La redundancia y la falta de normalización pueden hacer que los documentos ocupen más espacio en comparación con bases de datos relacionales.
2. **Dificultades en relaciones complejas:** Aunque las bases de datos documentales pueden manejar relaciones, modelar relaciones complejas y jerárquicas puede ser más desafiante que en las bases de datos relacionales.
3. **Diferentes formatos:** Aunque JSON es un formato común, no existe un estándar universal para la estructura de los documentos, lo que puede dificultar la migración entre diferentes sistemas.
4. **Complejidad en la Gestión de Esquemas Dinámicos:** Aunque el esquema flexible es una ventaja, puede ser difícil gestionar datos sin un esquema fijo a medida que crecen las colecciones.
5. **Licencias:** Algunas bases de datos documentales, especialmente las comerciales, pueden tener costos de licencia significativos, especialmente para grandes despliegues.

4.7 Casos de uso

- **Contenido multimedia:** Almacenamiento de imágenes, videos y otros tipos de contenido.
- **Comercio Electrónico:** Para manejar catálogos de productos donde las características de los productos pueden variar ampliamente.
- **Análisis de datos:** Almacenamiento de datos de registro, eventos y métricas.
- **Aplicaciones Web y Móviles:** Donde los datos JSON son nativos y la flexibilidad es crucial.
- **Aplicaciones de Configuración:** Almacenar configuraciones dinámicas que cambian frecuentemente y no siguen un esquema fijo.

References

- “Bases de datos documentales — Qué son, marcas y usos”. GraphEverywhere. Accedido el 20 de agosto de 2024. [En línea]. Disponible: <https://www.grapheverywhere.com/bases-de-datos-documentales/>
- “¿Qué es una base de datos de clave-valor?” Amazon Web Services, Inc. Accedido el 20 de agosto de 2024. [En línea]. Disponible: <https://aws.amazon.com/es/nosql/key-value/>
- “¿Qué es una base de datos NoSQL? — IBM”. IBM - United States. Accedido el 20 de agosto de 2024. [En línea]. Disponible: <https://www.ibm.com/mx-es/topics/nosql-databases: :text=NoSQL>,
- “Principios básicos de las bases de datos NoSQL en grafo”. Tecnología++. Accedido el 20 de agosto de 2024. [En línea]. Disponible: <https://blogs.uoc.edu/informatica/es/introduccion-a-las-bases-de-datos-nosql-en-grafo/: :text=Las>