

TAREA 1

Nombre del alumno: Molina Véjar Aarón Gael

Profesor: Ing. Fernando Arreola Franco

Bases de datos

Modelo de bases de datos orientado a objetos

El modelo de bases de datos orientado a objetos (OODBMS, por sus siglas en inglés) es un sistema de gestión de bases de datos que integra conceptos de la programación orientada a objetos con las bases de datos. Este modelo almacena datos en forma de objetos, similares a las instancias de clases en la programación orientada a objetos. Cada objeto contiene tanto datos como métodos, lo que permite que los datos sean manipulados directamente por las funciones del objeto.

Ventajas

- Integración con programación orientada a objetos: Facilita la implementación y el mantenimiento del software, ya que los desarrolladores pueden usar el mismo modelo conceptual en la base de datos y en el código de la aplicación.
- Herencia y reutilización: Permite la reutilización de código y la herencia de características entre objetos, lo que puede reducir la duplicación de datos.
- Flexibilidad: Capacidad para manejar datos complejos, como multimedia o gráficos, de manera más eficiente que los modelos relacionales tradicionales.
- Encapsulamiento: Los objetos pueden encapsular datos y comportamientos, lo que mejora la modularidad y la gestión de cambios.

Desventajas

- Curva de aprendizaje: Requiere que los desarrolladores estén familiarizados con la programación orientada a objetos y con el manejo de bases de datos, lo que puede aumentar la complejidad del desarrollo.
- Menor estandarización: A diferencia de los sistemas de bases de datos relacionales (RDBMS), los OODBMS no están tan estandarizados, lo que puede llevar a problemas de interoperabilidad.
- Escalabilidad limitada: Aunque son eficientes en manejar datos complejos, pueden tener limitaciones en cuanto a la escalabilidad y el rendimiento en comparación con los RDBMS para grandes volúmenes de datos.

Casos de uso

- Sistemas CAD/CAM: Utilizados en diseño y fabricación asistida por computadora, donde se manejan objetos complejos que representan componentes y ensamblajes.
- Aplicaciones multimedia: Donde es necesario manejar y manipular datos de audio, video, y gráficos.
- Sistemas de información geográfica (SIG): Utilizados para almacenar y manipular datos espaciales complejos.

Modelos de datos NoSQL

Clave-valor

El modelo de bases de datos clave-valor es una estructura de datos simple que almacena pares de clave y valor. Las claves son únicas y se utilizan para acceder a los valores correspondientes, que pueden ser de cualquier tipo, desde cadenas de texto hasta objetos complejos.

Ventajas

- Alta velocidad: Debido a su simplicidad, las bases de datos clave-valor suelen ser extremadamente rápidas en operaciones de lectura y escritura.
- Escalabilidad horizontal: Es fácil de escalar, ya que se pueden distribuir los datos entre múltiples nodos de manera eficiente.
- Flexibilidad en los datos: Permite almacenar valores sin una estructura fija, lo que es útil para datos no estructurados o semiestructurados.

Desventajas

- Limitada en consultas: No es ideal para realizar consultas complejas o relaciones entre datos, ya que sólo permite operaciones de búsqueda basadas en claves.
- Falta de estructura: La ausencia de un esquema definido puede dificultar la integridad de los datos y la realización de ciertas operaciones.

Casos de uso

- Cachés: Usadas para almacenar datos temporales que requieren acceso rápido.
- Carritos de compra en línea: Donde se almacenan elementos seleccionados por el usuario.

Documentales

Las bases de datos documentales almacenan datos en formato de documentos, como JSON, BSON o XML. Cada documento contiene pares de clave-valor y puede tener una estructura flexible y variada.

Ventajas

- Flexibilidad de esquema: Permiten almacenar documentos con estructuras distintas dentro de la misma base de datos, lo que es ideal para datos que cambian frecuentemente.
- Consultas eficientes: Los documentos pueden ser indexados, lo que permite realizar consultas complejas de manera eficiente.
- Manejo de datos complejos: Adecuadas para manejar datos anidados y estructuras jerárquicas.

Desventajas

- Redundancia de datos: Debido a la naturaleza flexible del esquema, puede haber redundancia de datos.
- Complejidad en la gestión de datos: El manejo de datos altamente anidados puede volverse complicado y afectar el rendimiento.

Casos de uso

- Sistemas de gestión de contenido (CMS): Donde se maneja contenido web que puede variar en estructura.
- Aplicaciones móviles y web: Donde la flexibilidad de datos es crucial debido a la naturaleza cambiante de las aplicaciones.

Grafos

Las bases de datos de grafos están diseñadas para almacenar y manejar datos que están altamente interconectados. En lugar de tablas, utilizan nodos (entidades) y aristas (relaciones) para representar los datos.

Ventajas

- Modelado natural de relaciones: Son ideales para representar relaciones complejas entre datos, como redes sociales, recomendaciones o análisis de rutas.
- Consultas de relaciones eficientes: Permiten realizar consultas sobre las relaciones entre los datos de manera más eficiente que las bases de datos relacionales.

- Flexibilidad: Pueden manejar cambios en las relaciones y en la estructura de los datos sin necesidad de alterar el esquema.

Desventajas

- Complejidad en el escalado: Aunque son eficientes para ciertos tipos de consultas, escalar bases de datos de grafos puede ser más complejo que escalar otros modelos de NoSQL.
- Herramientas limitadas: La disponibilidad de herramientas y soporte para bases de datos de grafos es más limitada en comparación con otros tipos de bases de datos.

Casos de uso

- Redes sociales: Para modelar y analizar relaciones entre usuarios.
- Sistemas de recomendación: Donde se analizan relaciones entre productos, usuarios, y preferencias.
- Análisis de fraude: En sectores como la banca, para detectar patrones sospechosos en transacciones.

Referencias

- R. Elmasri and S. B. Navathe, *Fundamentals of Database Systems*, 7th ed. Boston, MA, USA: Addison-Wesley, 2016.
- A. Shankar and S. Balachandran, "NoSQL: Databases for the Cloud," *IEEE Internet Computing*, vol. 18, no. 5, pp. 78-81, 2014.
- C. Strauch, "NoSQL Databases," *Information Services & Use*, vol. 31, no. 3-4, pp. 249-257, 2011.