

Modelo orientado a objetos

Descripción

Las bases de datos orientadas a objetos son un modelo relativamente nuevo diseñado para almacenar objetos completos, incluyendo su estado y comportamiento. Este enfoque sigue los principios de la programación orientada a objetos y permite a los usuarios definir operaciones sobre los datos como parte integral de la base de datos. Las operaciones constan de una interfaz (nombre y tipos de datos de los argumentos) y una implementación separada, lo que proporciona independencia entre programas y operaciones. Esto permite a las aplicaciones operar sobre los datos invocando a estas operaciones sin preocuparse por la implementación subyacente.

Ventajas

- ☒ Permite el modelado de datos complejos de forma rápida y ágil.
- ☒ Posibilidad de reutilizar y compartir código.
- ☒ Mejor integración con los lenguajes de programación que se basan en el paradigma de la orientación a objetos.
- ☒ Consultas mejoradas y más intuitivas que las de un sistema relacional

Desventajas

- ☒ La transición a bases de datos orientadas a objetos puede plantear desafíos en términos de compatibilidad con herramientas y sistemas existentes que pueden no estar diseñados para trabajar con este modelo.

Casos de Uso

- Aplicaciones que requieren modelar datos de manera compleja, como sistemas de diseño asistido por computadora (CAD) o simulaciones científicas.
- Útil en aplicaciones empresariales que tienen una lógica de negocio compleja y requieren una representación más natural de sus datos.
- Especialmente beneficioso cuando se trabaja en entornos de desarrollo de software orientado a objetos, donde los conceptos del modelo se alinean con la lógica de programación.

Modelo clave-valor

Descripción

El modelo de base de datos clave-valor es un tipo de base de datos NoSQL que ofrece una gran flexibilidad, ya que permite almacenar datos sin una estructura predefinida. En este modelo, la información se organiza en un diccionario de pares clave-valor, donde cada elemento tiene una clave única y un valor asociado. La clave actúa como un identificador, similar a un ID en una base de datos SQL, mientras que el valor puede ser cualquier tipo de datos. Este enfoque es considerado el más simple dentro de las bases de datos NoSQL y es eficaz para almacenar en caché y gestionar información de sesión del usuario, como carritos de compra. Algunas de las bases de datos basadas en este modelo son:

Cassandra Oracle Nosql BigTable Dynamo Redis Ryak

Ventajas

- ✓ No hay restricciones en la estructura de los datos, lo que proporciona flexibilidad para adaptarse a diferentes necesidades de la aplicación.
- ✓ Acceso rápido a los datos a través de la clave primaria, lo que hace que estas bases de datos sean eficientes y rápidas.
- ✓ Son fácilmente escalables al distribuirse en clústeres de servidores, lo que garantiza disponibilidad y rendimiento constante.
- ✓ Al trabajar en memoria o en caché, estas bases de datos maximizan el rendimiento y son adecuadas para aplicaciones que requieren acceso rápido a datos.

Desventajas

- ✗ No es eficiente para manejar relaciones entre datos, ya que está diseñado para recuperar información a través de claves.
- ✗ No permite realizar consultas directas sobre los valores almacenados, lo que limita su utilidad en casos donde se necesitan búsquedas avanzadas.
- ✗ No se recomienda para aplicaciones que requieren transacciones complejas, ya que no garantiza la consistencia en tiempo real.
- ✗ Requiere que se especifique la llave para recuperar el valor, y los valores no son indexados, lo que podría afectar la eficiencia en ciertos casos.
- ✗ No es ideal cuando necesita extraer varios registros a la vez, lo que puede afectar la eficiencia en operaciones que involucran conjuntos de datos extensos.

Caos de Uso

- Útil para almacenar en caché datos temporales y mejorar el rendimiento de las aplicaciones.
- Adecuado para gestionar información de sesión del usuario, como carritos de compra en aplicaciones de comercio electrónico. Un ejemplo de implementación exitosa de este modelo es [Amazon DynamoDB](#), que utiliza estas bases de datos para la venta de productos en línea.
- Se utiliza comúnmente para implementar módulos de logging donde se requiere alta velocidad y cacheo de datos.
- En casos donde se necesita acceso rápido a datos sin la complejidad de estructuras de tablas, como en aplicaciones de alta velocidad y rendimiento.

Modelo orientado a documentos

Descripción

Las bases de datos orientadas a documentos, también denominadas bases de datos de almacenamiento de documentos son un tipo de base de datos NoSQL diseñadas para almacenar, recuperar y gestionar datos semiestructurados. A diferencia de las bases de datos relacionales, no es necesario definir previamente un esquema fijo, permitiendo una mayor flexibilidad en la representación de la información. Estas bases de datos almacenan datos en formato de documentos, que pueden ser JSON, XML o BSON, brindando una estructura jerárquica y flexible. Algunas de las bases de datos basadas en este modelo son:

MongoDB	IBM Lotus Domino	SimpleDB	CouchBase	BaseX
---------	------------------	----------	-----------	-------

Ventajas

- ✓ No se requiere un esquema fijo, lo que facilita la adaptación a cambios en la estructura de los datos sin necesidad de modificaciones extensas.
- ✓ Almacena datos en formato de documentos (JSON, XML, BSON), facilitando la coherencia y minimizando la necesidad de conversiones al trabajar con aplicaciones.
- ✓ La naturaleza flexible y semiestructurada de los documentos permite una evolución natural de los datos a medida que cambian los requisitos de la aplicación.

Desventajas

- ✗ Puede resultar desafiante para transacciones complejas que requieren una estructura de datos más rígida.
- ✗ Menos eficiente para consultas que involucran relaciones complejas entre los datos.

Casos de Uso

- Ideal para almacenar y gestionar contenido de sitios web, blogs y otros sistemas que manejan datos semiestructurados.
- Adecuado para almacenar información de perfiles de usuario, donde cada usuario puede tener atributos específicos sin seguir una estructura estricta.
- Recomendado para el almacenamiento de registros de eventos y logs del sistema, donde la flexibilidad en la estructura es crucial para capturar datos diversos.
- Útil para recopilar datos heterogéneos de la web, como tweets, publicaciones de blogs y otra información variada.

Modelo de Grafos

Descripción

El modelo de grafos es un tipo de datos que organiza la información en nodos y relaciones, permitiendo representar conexiones entre nodos. Los nodos representan entidades como objetos, ubicaciones o personas, mientras que las aristas definen las relaciones entre estos nodos. Este enfoque proporciona una representación completa de los datos, especialmente adecuada para situaciones donde las conexiones son fundamentales.

Las bases de datos de grafos se utilizan para almacenar y gestionar redes de conexiones entre elementos, con ejemplos de aplicaciones como redes sociales, sistemas de reservas, detección de fraudes y gráficos de conocimiento. Algunas de las bases de datos basadas en este modelo son:

Neo4j	InfiniteGraph	HyperGraphDB	DEX/Sparksee	OrientDB
-------	---------------	--------------	--------------	----------

Ventajas

- ✓ Permite representar relaciones complejas de manera intuitiva y estructurada.
- ✓ Facilita la consulta eficiente de información relacionada y la identificación de patrones en conjuntos de datos conectados.

- ✓ No hay límites en la cantidad y tipo de relaciones que un nodo puede tener, proporcionando flexibilidad en la modelación de datos.
- ✓ Facilita la creación y ejecución de aplicaciones que funcionen con conjuntos de datos con un nivel alto de conexión.
- ✓ Se puede usar la teoría de grafos para recorrer la base de datos, encontrar la ruta más corta, medidas de centralidad y adyacencias.

Desventajas

- ✗ No se recomienda para realizar operaciones de agregación, particiones o escalado horizontal, ya que recorrer grafos en diferentes ubicaciones puede ser costoso y difícil de acceder.
- ✗ Puede no ser eficiente para operaciones que no involucren relaciones, como operaciones de agregación masiva.

Casos de Uso

- Modelar y analizar conexiones entre usuarios, amistades, interacciones, etc.
- Identificar patrones de preferencias y comportamientos para ofrecer recomendaciones personalizadas.
- Análisis de patrones de comportamiento y relaciones para identificar actividades fraudulentas.
- Modelado de conexiones en sistemas de transporte como trenes, autopistas, etc.

Fuentes de información

[1] Oracle, "¿Qué es NoSQL?," Oracle México, [En línea]. Disponible: <https://www.oracle.com/mx/database/nosql/what-is-nosql/>. [Accedido: 01, febrero, 2024].

[2] IBM, "¿Qué son las bases de datos NoSQL?," IBM España, [En línea]. Disponible: <https://www.ibm.com/es-es/topics/nosql-databases>. [Accedido: 01, febrero, 2024].

[3] Amazon Web Services, "¿Qué es NoSQL?," Amazon Web Services, [En línea]. Disponible: <https://aws.amazon.com/es/nosql/>. [Accedido: 01, febrero, 2024].

[4] The Data Schools, "NoSQL: Modelos de base de datos no relacionales," The Data Schools, [En línea]. Disponible: <https://thedata.schools.com/que-es/nosql/>. [Accedido: 01, febrero, 2024].

[5] M. Antonio García Téllez, "Uso de modelos de programación asíncronos administrados por eventos y bases de datos NoSQL en sistemas de puntos de venta para PyMES," Universidad Nacional Autónoma de México, 2019. [En línea]. Disponible: <http://www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/16709/Tesis.pdf>. [Accedido: 01, febrero, 2024].

[6] I. Moya Pérez y V. Jesús Rivera Estrada, "Bases de datos NoSQL: solución en sistemas distribuidos y de alto desempeño.," Universidad Nacional Autónoma de México, 2019. [En línea]. Disponible: <http://www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/4199/Tesis.pdf?sequence=1>. [Accedido: 01, febrero, 2024].

[7] C. Lorenzo Bragado, "Bases de Datos," CENTRO DE ESTUDIOS FINANCIEROS, 2013.