Modelo Orientado a objetos

El modelado orientado a objetos en bases de datos se refiere al uso de conceptos de programación orientada a objetos en el diseño e implementación de bases de datos. En este enfoque, los datos se almacenan en forma de objetos, que son instancias de clases, y las relaciones entre objetos se representan mediante herencia y composición. El modelo orientado a objetos en DBMS puede ayudar en la integridad de la base de datos, la independencia estructural y de la base de datos, y hacer que el código se parezca a objetos del mundo real. Sin embargo, también puede provocar un acceso complejo a los datos de navegación, transacciones lentas y una curva de aprendizaje pronunciada para los desarrolladores.

Los componentes del modelo de datos orientado a objetos son clases, objetos, atributos, métodos y herencia. Las bases de datos orientadas a objetos almacenan objetos en lugar de datos como números enteros y cadenas, y la relación entre varios datos está implícita en el objeto y se manifiesta como atributos y métodos del objeto.

Ventajas del modelado orientado a objetos en bases de datos:

- Se añade contenido semántico.
- Soporte para objetos complejos.
- La herencia promueve la integridad de los datos.
- La representación visual incluye contenido semántico.
- Capacidades de modelado enriquecidas.
- Expresivo.
- Apoyo.
- Almacenamiento en caché de bases de datos.
- Extensibilidad.
- Fácil de guardar y recuperar datos rápidamente.
- Integración perfecta con lenguajes de programación orientados a objetos.
- Mayor confiabilidad y flexibilidad.
- Mantenimiento reducido.
- Reutilización del código.
- Modelado del mundo real.

Desventajas del modelado orientado a objetos en bases de datos:

- Sistema de navegación complejo.
- Lento desarrollo de estándares.
- Altos gastos generales del sistema.
- Transacciones lentas.

- Falta de modelo de datos universal.
- Falta de apoyo a las opiniones.
- Seguridad.
- Falta de experiencia.
- Competencia.
- Complejidad.
- Mala adopción entre la comunidad.
- La mayoría de los modelos carecen de un modelo de datos universalmente acordado.
- Obstrucción de la base de datos debido a conexiones extensas

Modelos NoSQL

Las bases de datos NoSQL son bases de datos no relacionales que se utilizan para almacenar datos en forma no tabular. Existen diferentes tipos de bases de datos NoSQL, incluidas bases de datos documentales, clave-valor, almacenes de columnas anchas y bases de datos de grafos.

Bases de datos basadas en documentos: este tipo de base de datos NoSQL almacena datos en documentos, generalmente codificados en formato JSON. Cada documento tiene una clave única y los datos se almacenan como un conjunto de documentos. Las bases de datos basadas en documentos se utilizan para almacenar datos semiestructurados y no estructurados, y proporcionan resultados de consultas más estructurados y organizados que las bases de datos de pares clave-valor.

Base de datos clave-valor: este tipo de base de datos NoSQL almacena datos como una matriz asociativa, en la que una clave individual está vinculada con un solo valor de una colección. Las bases de datos de valores clave son simples y fáciles de usar, y son ideales para almacenar, obtener y eliminar datos. Sin embargo, no tienen un lenguaje de consulta y carecen de relaciones dinámicas entre dos documentos.

Bases de datos de columnas anchas: este tipo de base de datos NoSQL almacena datos en tablas con filas y columnas, pero con un esquema flexible que permite agregar nuevas columnas sin alterar los datos existentes. Los almacenes de columnas anchas se utilizan para almacenar grandes cantidades de datos con patrones de consulta predecibles.

Bases de datos de grafos: este tipo de base de datos NoSQL almacena datos en nodos y bordes, que representan entidades y relaciones entre ellas. Las bases de datos de gráficos se utilizan para analizar y recorrer relaciones entre datos y son ideales para aplicaciones como redes sociales, motores de recomendación y detección de fraude.

Ejemplos de manejadores de bases de datos populares asociados con cada tipo:

- Modelo Orientado a Objetos:
 - ZODB (para Python)
 - ObjectDB (para Java).
- Base de Datos Clave-Valor:
 - o Redis
 - Amazon DynamoDB
- Base de Datos Documentales:
 - MongoDB
 - Couchbase
- Base de Datos de Grafos:
 - Neo4i
 - o OrientDB

Bibliografía

"DBMS - Data Models". Online Courses and eBooks Library | Tutorialspoint. Accedido el 12 de septiembre de 2023. [En línea].

Disponible: https://www.tutorialspoint.com/dbms/dbms data models.htm

"Basic Object Oriented Data Model - GeeksforGeeks". GeeksforGeeks. Accedido el 12 de septiembre de 2023. [En línea]. Disponible: https://www.geeksforgeeks.org/basic-object-oriented-data-model/

"Explain the object oriented data model in DBMS". Online Courses and eBooks Library | Tutorialspoint. Accedido el 12 de septiembre de 2023. [En línea].

Disponible: https://www.tutorialspoint.com/explain-the-object-oriented-data-model-in-dbms

IONOS editorial team. "Object-oriented databases: the insider tip in database models". IONOS Digital Guide. Accedido el 12 de septiembre de 2023. [En línea].

Disponible: https://www.ionos.com/digitalguide/hosting/technical-matters/object-oriented-databases/

M. Drake y ostezer. "A Comparison of NoSQL Database Management Systems and Models". DigitalOcean | Cloud Hosting for Builders. Accedido el 12 de septiembre de 2023. [En línea]. Disponible: https://www.digitalocean.com/community/tutorials/a-comparison-of-nosgl-database-management-systems-and-models

"4 Types of NoSQL Databases & When to use them?" Blazeclan. Accedido el 12 de septiembre de 2023. [En línea]. Disponible: https://www.blazeclan.com/blog/dive-deep-types-nosql-databases/