

Modelo orientado a objetos

Permite que los diseños y las implementaciones sean portables entre los sistemas que lo soportan.
Primitivas de modelado:

- Los componentes básicos son objetos y literales. Un objeto es una instancia autocontenida de una entidad de interés del mundo real.
- Tienen identificador único. Literal es un valor específico. No tiene identificadores. Puede ser una estructura o un conjunto de valores relacionados.
- Se categorizan en tipos. Cada tipo tiene un dominio específico compartido por todos los objetos y literales de ese tipo. Los tipos también pueden tener comportamientos, que también comparten todos los objetos del mismo.
- Lo que un objeto sabe hacer son sus operaciones. Puede requerir datos de entrada y devolver algún valor de un tipo conocido.
- Las propiedades son sus atributos y las relaciones. El estado viene dado por los valores actuales de sus propiedades.
- Una base de datos es un conjunto de objetos almacenados que pueden ser accedidos por múltiples usuarios y aplicaciones.
- La definición de una base de datos está contenida en un esquema que se ha creado mediante el lenguaje de definición de objetos ODL.

Ejemplos:

En este primer ejemplo, pongamos que tenemos una tienda de material deportivo; en un BDOO guardaremos cada objeto con sus atributos y métodos: unos patines (el objeto), son en línea, azules, para niños, para adultos, etc. (atributos y métodos). Los patines forman parte de la clase «patines», donde podemos incluir también patines negros o patines paralelos. La clase «patines» es una subcategoría de «productos deportivos con ruedas», que incluye también la clase «monopatines». Este objeto también está relacionado a su vez con la clase «deportes de exterior». Cuando accedemos a través del código identificador único del objeto, obtenemos de todos sus métodos y atributos.

Un segundo ejemplo lo podemos ver con una librería; si esta emplease una base de datos orientada a objetos, se podría guardar «libro A» como objeto, con los atributos y métodos siguientes: tapa dura, extensión, autor, etc. El Libro A podría incluirse en la clase «Novela de misterio», dentro de una subclase «Novela norteamericana». Así mismo, estas clases podrían ser subcategorías de «Novelas de ficción».

Ventajas:

- Si estamos trabajando con lenguajes de programación orientados a objetos, son las más adecuadas, puesto que los objetos del código fuente se pueden integrar fácilmente en la base de datos.
- Permite manipular conjuntos de datos complejos de manera más rápida, tanto para guardarse como para su consulta.
- A cada objeto se le asigna de forma automática un código de identificación.
- Alto nivel de confiabilidad en la ejecución de transacciones ACID, es decir, que las transacciones se completan solo cuando no existen conflictos con los datos.
- Cuentan con un mecanismo de almacenamiento en caché que crea réplicas parciales de la base de datos, lo que mejora su rendimiento.
- Son capaces de manejar grandes volúmenes de datos.

Desventajas:

En cuanto a sus desventajas, podemos hablar de una principal y es que el uso de las BDOO está poco extendido aún y esto implica que todavía no hay criterios claros de estandarización y existe poca documentación sobre los proyectos que las han implementado.

La optimización de consultas requiere una comprensión de la implementación de los objetos, para poder acceder a la base de datos de manera eficiente. Sin embargo, esto compromete el concepto de encapsulación.

Base de datos NoSQL Clave - Valor

Esta modalidad de base de datos, key-value database o store en inglés, se basa en una tabla de tan solo dos columnas. En una de ellas se guarda un valor y en la otra, una clave que representa una característica identificativa única. Un valor puede ser sencillo, como una cadena de caracteres o un número entero, o pueden ser objetos complejos (un documento también puede ocupar el lugar de un valor, aunque, entonces se hablaría de una base de datos de documentos). En las bases de datos se pueden incluir también referencias a archivos, así como a tuplas (conjunto de valores).

El contenido de la base de datos puede ser muy heterogéneo, por lo que es posible incluir objetos distintos en una misma columna. Lo mismo se aplica a los rasgos identificativos. En la mayoría de los casos, las claves seguirán un determinado esquema, pero esto no es indispensable. Tanto las cadenas de caracteres como los enteros se pueden constituir siguiendo criterios libres.

Las bases de datos clave-valor son muy efectivas en la consulta y fáciles de escalar, lo cual se deriva de la sencilla estructura del modelo. Asimismo, este modelo de base de datos ofrece una gran velocidad de búsqueda gracias a la sencilla conexión entre la clave y el valor.

Los campos de aplicación de las bases de datos clave-valor se derivan tanto de las ventajas que aportan como de sus limitaciones. Así, se utilizan siempre que se requiere una alta velocidad con un gran volumen de datos, como en las cestas de la compra de las tiendas online o para datos de sesión. En estos casos, la información está dispuesta de forma clara y debe estar disponible en poco tiempo. En sitios web muy conocidos y de mucho tamaño, estas bases de datos deben crear continuamente nuevas entradas y eliminar las antiguas, por lo que el almacenamiento clave-valor resulta perfecto para ello.

Base de datos NoSQL documentales

Las bases de datos documentales, en inglés, document stores, se utilizan para la administración de datos semiestructurados. Se trata de datos que no siguen una estructura fija, sino que llevan la estructura casi en sí misma. Sin embargo, con ayuda de marcadores dentro de estos datos, la información puede ordenarse. Debido a la falta de una estructura clara, estos datos no son adecuados para bases de datos relacionales porque su información no puede clasificarse en tablas.

Una base de datos documental crea un par simple: un documento específico se asigna a una clave. En este documento, que puede tener formato XML, JSON o YAML, por ejemplo, se puede encontrar la información propiamente dicha. Como la base de datos no requiere un esquema específico, en un almacén de documentos pueden integrarse diferentes clases de documentos. Los cambios en los documentos no afectan a la base de datos.

Ventajas y desventajas

En las bases de datos relacionales clásicas, debe haber un campo para cada dato, y esto en cada entrada. Si la información no está disponible, la celda queda vacía, pero debe haber una. Las bases de datos documentales son mucho más flexibles: la estructura de los documentos no tiene por qué ser coherente, lo que permite almacenar conjuntos de datos no estructurados muy grandes en una sola base de datos.

También es más fácil integrar datos nuevos: mientras que, en una base de datos relacional, el nuevo punto de información se debe insertar en todos los registros de datos, en una base de datos orientada a documentos basta con integrar la información nueva en solo unos pocos registros. El contenido adicional se puede añadir a otros documentos, pero esto no es necesario.

En estas bases de datos, además, la información no se reparte en varias tablas enlazadas. Todo se almacena en el mismo lugar, lo que puede mejorar el rendimiento. Sin embargo, esta ventaja de velocidad sólo puede explotarse en las bases de datos documentales, siempre y cuando no se intente insertar elementos relacionales, porque las referencias no encajan en el concepto documental. Si, pese a todo, se intenta conectar a los documentos entre sí, el sistema se vuelve muy complejo y voluminoso. Cuando se trata de datos muy interconectados, se recomienda usar un sistema de base de datos relacional.

Base de datos NoSQL grafos

Una base de datos orientada a grafos o graph database se basa, como bien dice su nombre, en grafos, un conjunto de objetos (vértices y aristas) que permite representar datos interconectados, así como las relaciones entre ellas, de forma comprensible y como un único y más amplio conjunto de datos. Los grafos están formados por nodos o vértices, que son propiedades de datos u objetos claramente señalizados e identificables, y aristas o arcos, que representan las relaciones entre los objetos. Gráficamente, estos dos componentes tienen forma de puntos y líneas, respectivamente. Las aristas tienen un extremo inicial y uno final, mientras que cada nodo siempre contiene un número concreto de relaciones a otros nodos, ya sean de entrada o de salida.

Las bases de datos de grafos pueden usarse en contextos variados y con fines muy distintos, ya que permiten analizar información y entender, evaluar y aprovechar los procesos y las relaciones.

Un ejemplo típico del uso de bases de datos de grafos es el análisis de las relaciones entre los usuarios de las redes sociales o de los hábitos de compra de los clientes en tiendas online. Gracias a los diferentes datos y a las relaciones entre ellos se pueden realizar, por ejemplo, recomendaciones personalizadas de compra o de amistades, creando así redes de usuarios y de productos. Las empresas pueden aprovechar, además, la posibilidad de obtener información acerca de las búsquedas, del historial de clics y de otros comportamientos del usuario y crear con ella perfiles completos de clientes. En la gestión de la cadena de suministro se usan bases de datos de grafos para monitorizar todos los procesos, desde el diseño hasta las ventas. Otro uso importante de las bases de datos orientadas a grafos son los análisis de riesgo, la detección de fraudes y la búsqueda de errores.

Ventajas:

- La velocidad de búsqueda depende únicamente del número de relaciones concretas, no del conjunto de datos
- Resultados en tiempo real
- Presentación intuitiva y resumida de las relaciones
- Estructuras flexibles y ágiles

Desventajas:

- Es difícil de escalar, por estar diseñado para arquitecturas con un solo servidor
- Sin lenguaje de consulta consistente

Bibliografía

J. Piqueres Torres. "BASES DE DATOS ORIENTADAS A OBJETOS". I.E.S. San Vicente. <https://iessanvicente.com/colaboraciones/bdOO.pdf> (accedido el 16 de agosto de 2023).

Facultad de Estadística e Informática. "Manejadores de Bases de Datos". Universidad Veracruzana. <https://www.uv.mx/personal/ermeneses/files/2019/02/Clase-BasesDeDatosNoEstructuradas-II.pdf> (accedido el 16 de agosto de 2023).

"5.1 Conceptos y tipos de bases de datos NoSQL (documental, columnar, clave/valor y de grafos) | Prácticas de Tecnologías de Gestión y Manipulación de Datos". GitHub Pages. <https://gltaboada.github.io/tgdbook/conceptos-y-tipos-de-bases-de-datos-nosql-documental-columnar-clavevalor-y-de-grafos.html> (accedido el 16 de agosto de 2023).

"¿Qué es una base de datos orientada a grafos?" Oracle | Cloud Applications and Cloud Platform. <https://www.oracle.com/mx/autonomous-database/what-is-graph-database/> (accedido el 16 de agosto de 2023).