

Modelo Orientado a Objetos

Las bases de datos de objetos son especialmente adecuadas si ya estamos trabajando con lenguajes de programación orientados a objetos, como Java, porque los objetos del código fuente se pueden integrar fácilmente en la base de datos. Si recurrimos a una base de datos relacional, lo que suele ser lo más frecuente, nos costará incorporar objetos complejos a la estructura tabular.

Uno de los inconvenientes de este modelo es que su uso está poco extendido.

La mayoría de los desarrolladores prefieren utilizar bases de datos relacionales, que están generalizadas, bien documentadas y mucho más desarrolladas.

La complejidad de los objetos garantiza que hasta las consultas y anotaciones más complejas puedan llevarse a cabo mucho más rápido que en los modelos relacionales. Sin embargo, si los procesos son sencillos en comparación, no se puede prescindir de la estructura compleja, lo que puede conllevar problemas de ralentización.

Ventajas	Inconvenientes
Los conjuntos de datos complejos pueden guardarse y consultarse de forma rápida y sencilla.	El uso de las bases de datos orientadas a objetos no está muy extendido.
Los códigos de identificación se asignan automáticamente a cada objeto.	En algunas situaciones, la gran complejidad puede acarrear problemas de rendimiento.
Funciona bien con lenguajes de programación orientados a objetos.	

Manejadores:

El gestor orientado a objetos de InterSystems Caché.

El gestor orientado a objetos de InterSystems IRIS.

El gestor orientado a objetos de Giga spaces.

El gestor orientado a objetos de ObjectBox.

El gestor orientado a objetos de ObjectStore.

Modelo NoSQL.

Las bases de datos de NoSQL presentan muchas ventajas en comparación con las bases de datos tradicionales. A diferencia de las bases de datos relacionales, las bases de datos NoSQL están basadas en key-value pairs. Algunos tipos de almacén de bases de datos NoSQL incluyen diferentes tipos de almacenes como por ejemplo el almacén de columnas, de documentos, de key value store, de gráficos, de objetos, de XML y otros modos de almacén de datos.

Algunos tipos de almacén de bases de datos NoSQL incluyen almacenes de columnas, de documentos, de valores de claves, de gráficos, de objetos, de XML y otros modos de almacén de datos.

Podría decirse que las bases de datos NoSQL de código abierto tienen una implementación rentable. Ya que no requieren las tarifas de licencia y pueden ejecutarse en hardware de precio bajo.

Cuando trabajamos con bases de datos NoSQL, ya sean de código abierto o tengan un propietario, la expansión es más fácil y más barata que cuando se trabaja con bases de datos relacionales. Esto se debe a que se realiza un escalado horizontal y se distribuye la carga por todos los nodos. En lugar de realizarse una escala vertical, más típica en los sistemas de bases de datos relacionales.

Por supuesto, las bases de datos NoSQL no son perfectas, y no siempre van a ser la elección ideal.

- La mayoría de las bases de datos NoSQL no admiten funciones de fiabilidad, que son soportadas por sistemas de bases de datos relacionales. Estas características de fiabilidad pueden resumirse en:

“atomicidad, consistencia, aislamiento y durabilidad.” Esto también significa que las bases de datos NoSQL, que no soportan esas características, ofrecen consistencia para el rendimiento y la escalabilidad.

- Con el fin de apoyar las características de fiabilidad y coherencia, los desarrolladores deben implementar su propio código, lo que agrega más complejidad al sistema.
- Esto podría limitar el número de aplicaciones en las que podemos confiar para realizar transacciones seguras y confiables, como por ejemplo los sistemas bancarios.
- Otras formas de complejidad encontradas en la mayoría de las bases de datos NoSQL, incluyen la incompatibilidad con consultas SQL. Esto significa que se necesita un lenguaje de consulta manual, haciendo los procesos mucho más lentos y complejos.

Key Value Store.

En el tipo de almacén Key Value, se utiliza una tabla hash en la que una clave única apunta a un elemento. Las claves pueden ser organizadas por grupos clave lógicos, requiriendo solamente estas claves para ser únicas dentro de su propio grupo. Esto permite tener claves idénticas en diferentes grupos lógicos.

Todo lo que se necesita para hacer frente a los elementos almacenados en la base de datos: es la clave. Los datos se almacenan en una forma de una cadena, JSON o BLOB (objeto grande binario).

Uno de los mayores defectos en esta forma de base de datos es la falta de consistencia a nivel de la base de datos. Esto puede ser añadido por los desarrolladores con su propio código, aunque esto suponga más esfuerzo y tiempo.

La base de datos NoSQL más famosa que se construye en un almacén de valores clave Key Value es DynamoDB de Amazon.

Almacén de documentos.

Los almacenes de documentos son similares a los almacenes de valores clave, porque no tienen un esquema y se basan en un modelo de valor clave. Ambos carecen de coherencia en el nivel de base de datos, lo que hace posible que las aplicaciones proporcionan más fiabilidad.

Las diferencias más significativas son:

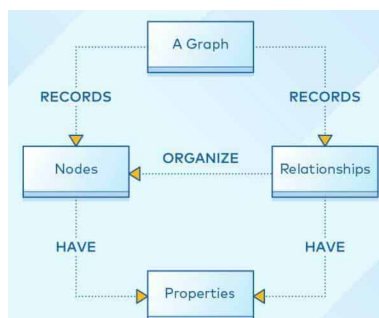
- En el almacén de documentos, los valores (documentos) proporcionan codificación XML, JSON o BSON (JSON codificado binario) para los datos almacenados.

La aplicación de base de datos más popular, que se basa en un almacén de documentos es MongoDB.

Base gráfica.

En una gráfica de una base de datos NoSQL, se utiliza una “estructura de gráfica dirigida” para representar los datos. El gráfico está compuesto por bordes y nodos.

Formalmente, un gráfico, es una representación de un conjunto de objetos, donde algunos pares de objetos están conectados por enlaces. Los objetos interconectados están representados por abstracciones matemáticas, llamadas vértices, y los enlaces que conectan algunos pares de vértices se llaman bordes. Un gráfico sería un tipo de representación de datos compuesto por un conjunto de vértices y de bordes que conectan entre sí, mostrando visualmente su relación matemática.



Esto ilustra la estructura de una base de datos de manera gráfica, donde se usan bordes y nodos para representar y almacenar los datos. Estos nodos están organizados entre sí, y queda representado por los bordes entre los nodos. Tanto los nodos como las relaciones tienen propiedades definidas.

Las bases de datos de gráficos, suelen utilizarse en aplicaciones de redes sociales. Estas permiten a los desarrolladores centrarse más en las relaciones entre los objetos que en los propios objetos. En este contexto, de hecho permiten un entorno escalable y fácil de usar.

Actualmente, InfoGrid y InfiniteGraph son las bases de datos gráficas más populares.

Manejadores:

MongoDB

CouchDB

Redis

Referencias.

"¿Qué es una base de datos orientada a objetos?" KeepCoding Bootcamps.
<https://keepcoding.io/blog/base-de-datos-orientada-a-objetos/> (accedido el 15 de agosto de 2023).

Equipo editorial de IONOS. "Base de datos orientada a objetos: el secreto mejor guardado de los modelos de bases de datos". IONOS Digital Guide.
<https://www.ionos.mx/digitalguide/hosting/cuestiones-tecnicas/base-de-datos-orientada-a-objetos/> (accedido el 15 de agosto de 2023).

"Bases de datos NoSQL: Guía con las ventajas y desventajas". Pandora FMS - The Monitoring Blog.
<https://pandorafms.com/blog/es/bases-de-datos-nosql/#:~:text=Ventajas%20y%20desventajas%20de%20las%20bases%20de%20datos,NoSQL%20y%20las%20bases%20de%20datos%20relacionales:%20> (accedido el 16 de agosto de 2023).

"Principales manejadores". Bases De Datos NoSQL.
<https://bases-de-datos-nosql.blogspot.com/2019/02/principales-manejadores.html> (accedido el 16 de agosto de 2023).