

Base de datos orientado a objetos

Una base de datos orientada a objetos es aquella que implementa dentro de un sistema de información la representación de datos en forma de objetos. Estas bases de datos se diferencian de las bases de datos relacionales clásicas, debido a que no responden a un sistema de tablas para registrar su información.

La mayoría de las bases de objetos ofrecen lenguajes de consultas para encontrar los objetos que pertenecen a la base. El acceso a los datos en estas bases es más rápido debido a que los objetos pueden ser recuperados de forma directa sin búsqueda específica.

Ventajas de las Bases de datos orientadas a objetos

Lenguajes de Consulta:

Entre los aspectos más positivos que poseen dichas bases de datos se encuentra su lenguaje de consulta.

Persistencia transparente:

Esta es una característica muy importante. Se refiere a la posibilidad de la base de datos de utilizar uno o varios objetos a través del lenguaje de programación para obtener análisis o manipulación de la data.

Desventajas de las bases de datos orientadas a objetos

Al igual que diversas bases de datos NoSQL, carecen de madurez en el desarrollo. Existe poca experiencia y documentación sobre proyectos que las implementen. Esto pudiese extender los tiempos de desarrollo.

Base de datos objeto relacional

El modelo de base de datos objeto-relacional integra los conceptos de la tradicional base de datos relacional y los conceptos de paradigma de objetos que se utiliza en la programación orientada a objetos (POO).

El objetivo de este concepto es poder aplicar la tecnología madura de bases de datos relacionales sobre la organización de los datos complejos es decir datos de texto e imagen, mapas, datos en el rango de audio etc. Las bases de datos Objeto-relacional son compatibles con estos objetos de datos y las operaciones de mayor complejidad.

En bases de datos orientadas a objetos, los usuarios pueden definir operaciones sobre los datos como parte de la definición de la base de datos. Una operación (llamada función) se especifica en dos partes. La interfaz (o signatura) de una operación incluye el nombre de la operación y los tipos de datos de sus argumentos (o parámetros). La implementación (o método) de la operación se especifica separadamente y puede modificarse sin afectar la interfaz. Los programas de aplicación de los usuarios pueden operar sobre los datos invocando a dichas operaciones a través de sus nombres y argumentos, sea cual sea la forma en la que se han implementado. Esto podría denominarse independencia entre programas y operaciones.

Una base de datos orientada a objetos es una base de datos que incorpora todos los conceptos importantes del paradigma de objetos

Encapsulación - Propiedad que permite ocultar la información al resto de los objetos, impidiendo así accesos incorrectos o conflictos.

Herencia - Propiedad a través de la cual los objetos heredan comportamiento dentro de una jerarquía de clases.

Polimorfismo - Propiedad de una operación mediante la cual puede ser aplicada a distintos tipos de objetos.

De este modo, se genera la posibilidad de guardar objetos más complejos en una sola tabla con referencias a otras relaciones, con lo que se acerca más al paradigma de programación orientada a objetos.

Características de estos datos complejos:

Colecciones: También conocidos como conjuntos, este tipo de datos clasifican los arrays y los conjuntos en que los elementos pueden aparecer varias veces.

Tipos estructurados: Los tipos estructurados permiten representación directa de los atributos compuestos en los diagramas entidad-relación.

Objetos de gran tamaño: Desde ya hace varios años que se necesita almacenar datos con atributos muy grandes (Varios Mbytes), como libros, canciones, etc. E incluso aún más grandes; como mapas de alta resolución, video, etc. que puede llegar fácilmente a los Gbytes.

Bases de datos NoSQL

El uso de las bases de datos NoSQL se hace presente hoy en día de forma frecuente. Las aplicaciones y los softwares que más utilizamos las incluyen de forma directa en su arquitectura. Algún tiempo atrás fue toda una revolución la aparición de las BBDD construidas con SQL. Han cumplido y lo han hecho bien, pero las necesidades del mundo digital avanzan a pasos agigantados y lamentablemente las BBDD SQL tienen algunas limitaciones.

Esta es una de las razones de la aparición de las Bases de datos NoSQL. La evolución tecnológica de la sociedad y la participación masiva de usuarios que producen y alojan contenido cada segundo, exigía una forma más flexible de almacenar, ordenar y captar mayor cantidad de datos. Eso es lo que ha ayudado a solucionar el NoSQL.

Aunque son conocidas desde la década de los 60 del pasado siglo, su auge actual viene determinado por el uso que, de estos sistemas han hecho las principales compañías de internet como Amazon, Google, Twitter y Facebook.

Las características comunes entre todas las implementaciones de bases de datos NoSQL suelen ser las siguientes:

Consistencia Eventual: A diferencia de las bases de datos relacionales tradicionales, en la mayoría de sistemas NoSQL, no se implementan mecanismos rígidos de consistencia que garanticen que cualquier cambio llevado a cabo en el sistema distribuido sea visto.

Flexibilidad en el esquema: En la mayoría de base de datos NoSQL, los esquemas de datos son dinámicos; es decir, a diferencia de las bases de datos relacionales en las que, la escritura de los datos debe adaptarse a unas estructuras(o tablas, compuestas a su vez por filas y columnas.

Escalabilidad horizontal: Por escalabilidad horizontal se entiende la posibilidad de incrementar el rendimiento del sistema añadiendo, simplemente, más nodos (servidores) e indicando al sistema cuáles son los nodos disponibles.

Estructura distribuida: Generalmente los datos se distribuyen, entre los diferentes nodos que componen el sistema.

Pese a todas las opciones proporcionadas por el auge de las bases de datos NoSQL, esto no significa la desaparición de las bases de datos de RDBMS ya que son tecnologías complementarias.