

Modelo orientado a objetos

Origen o concepto

Los sistemas de bases de datos orientados a objetos parecen ser la tecnología más prometedora para los próximos años, aunque carecen de un modelo de datos común y de fundamentos formales, además de que su comportamiento en seguridad y manejo de transacciones no están a la altura de los programas actuales de administradores de bases de datos.

Ejemplo representando el almacenamiento de los datos de dos objetos.

Almacenamiento de datos en un OODBMS

Clase	Objetos	Atributos/datos
Empleado	Juan Pérez	Edad: 25
		Puesto: Psicóloga social
		Salario: 8000
	María Suárez	Edad: 23
		Puesto: Pedagoga
		Salario: 15 000

Hay organismos en pro de la estandarización de este tipo de sistemas manejadores de bases de datos, como el OMG (Object Management Group), la CAD Framework Initiative y el grupo de trabajo 'e ANSI (American National Standards Institute).

Algo que apoya esta tendencia es que a pesar de que la ingeniería de software orientada a objetos requiere mucho tiempo de análisis, la mayoría de los proyectos de desarrollo son más cortos y requieren menos personas, además de que la cantidad de código es menor. Veamos su concepto y características más importantes.

Definición

En la actualidad, hay mucha atención hacia los OODBMS, tanto en el terreno de desarrollo como en el teórico, no hay una definición estándar de lo que estos sistemas significan. Existen tres problemas principales que impiden una definición generalizada:

1. **La falta de un modelo de datos común entre los diferentes sistemas:** Los sistemas de bases de datos relacionales cuentan con especificaciones claras dadas por Codd, pero los orientados a

objetos no tienen algo así. Se pueden encontrar muchos textos que describen diferentes modelos, pero no hay uno estándar.

2. **La carencia de fundamentos formales:** El fundamento teórico de la programación orientada a objetos es escaso en comparación con otras áreas como la programación lógica. Además, se carece de definiciones de diversos conceptos.
3. **Una actividad experimental muy fuerte:** Existe mucho trabajo experimental, la mayoría de los desarrollos son sistemas prototipo no comerciales, por eso, no hay trabajo de conceptualización y definición de estándares. El diseño de estos sistemas está orientado por las aplicaciones que los requieren y no por un modelo común.

El problema de estos sistemas es similar al de las bases de datos relacionales a mitad de los setenta. La gente se dedicaba a desarrollar implementaciones en lugar de definir las especificaciones para luego hacer la tecnología que permitiera implementarlas. Se espera que de los prototipos y desarrollos actuales de los OODBMS surja un modelo. Aunque también se corre el riesgo de que alguno de éstos se convierta en el estándar por su demanda en el mercado. A manera de definición podemos decir que un OODBMS debe satisfacer dos criterios:

Debe ser un DBMS
El primer criterio incluye características de cualquier DBMS, como persistencia, administración de almacenamiento secundario, concurrencia, recuperación y facilidad de consultas personalizadas.
Debe ser un sistema orientado a objetos, consistente con los lenguajes de programación orientada a objetos
El segundo criterio corresponde a características que se comparten con la programación orientada a objetos: objetos complejos, identidad de objetos, encapsulación, herencia, sobreescritura y sobrecarga y completa capacidad computacional —computational completeness—.

Ejemplos o Manejadores

Los administradores de bases de datos relacionales cuentan con un lenguaje para realizar procesos computacionales sobre los datos: el SQL. Además, adicionan un lenguaje procedimental que permite la definición de variables, manejo de excepciones, ciclos y estructuras condicionales. Algunos de estos lenguajes son:

PL/sql para Oracle

pl/pgsql para PostgreSQL

Transact-SQL para SQL Server de Microsoft

Modelos NoSQL

Clave-Valor

Origen/Concepto

Este es el tipo más flexible de base de datos NoSQL porque la aplicación tiene un control completo sobre lo que se almacena en el campo de valor sin ninguna restricción.

Ejemplos/Manejadores

- Redis
- Memcached
- Amazon DynamoDB
- Azure Cosmos DB for table
- Azure Cache for Redis
- Azure Table Storage

Documentales

Origen/Concepto

También denominadas bases de datos orientadas a documentos o de almacenamiento de documentos, estas se utilizan para almacenar, recuperar y gestionar datos semiestructurados. No es necesario especificar los campos que debe contener cada documento.

Ejemplos/Manejadores

- MongoDB
- Amazon DocumentDB
- Azure Cosmos DB

Grafos

Origen/Concepto

Esta base de datos organiza los datos como nodos y relaciones, que muestran las conexiones entre nodos. Esto permite una representación de los datos más rica y completa. Las bases de datos de gráficos se aplican en redes sociales, sistemas de reservas y detección de fraudes.

Ejemplos/Manejadores

- Neo4J
- Amazon Neptune
- Grapa API de Azure Cosmos DB

Ventajas/Desventajas

Ventajas: Dado que las empresas y las organizaciones necesitan innovar con rapidez, la clave y su objetivo pasa por ser capaces de mantenerse ágiles y seguir operando a cualquier escala. Las bases de datos NoSQL ofrecen esquemas flexibles y también admiten una variedad de modelos de datos que son perfectos para crear aplicaciones que requieran grandes volúmenes de datos y tiempos de respuesta o latencia bajos, por ejemplo, juegos en línea y aplicaciones web de comercio electrónico.

Desventajas: Las bases de datos NoSQL suelen utilizar datos no normalizados, de modo que admiten los tipos de aplicaciones que utilizan menos tablas (o contenedores) y cuyas relaciones de datos no se modelan mediante referencias, sino como registros (o documentos) incrustados. Muchas aplicaciones empresariales clásicas del back-office utilizadas en finanzas, contabilidad y planificación de recursos empresariales usan datos altamente normalizados para evitar anomalías y duplicaciones. Estos son en general los tipos de aplicaciones que no resultan adecuados para las bases de datos NoSQL.

Otra elemento distintivo de las bases de datos NoSQL es la complejidad de las consultas. Las bases de datos NoSQL funcionan fenomenalmente bien con las consultas realizadas en relación con una única tabla. Sin embargo, cuando crece la complejidad de las consultas, las bases de datos relacionales resultan más adecuadas. Normalmente, la base de datos NoSQL no ofrece uniones complejas, subconsultas ni anidamiento de consultas en una cláusula WHERE.

Sin embargo, a veces no es necesario elegir entre bases de datos relacionales y no relacionales. En muchas ocasiones, las empresas han optado por bases de datos que ofrecen un modelo convergente, en el que pueden emplear una combinación de modelos de datos relacionales y no relacionales. Este enfoque híbrido ofrece más flexibilidad para el manejo de diferentes tipos de datos, al tiempo que garantiza la consistencia de lectura y escritura sin afectar al rendimiento.

Bibliografía

- [1] CUAIEED | Coordinación de Universidad Abierta, Innovación Educativa y Educación a Distancia, «Modelo Orientado a Objetos,» [En línea]. Available: https://programas.cuaed.unam.mx/repositorio/moodle/pluginfile.php/782/mod_resource/content/8/contenido/index.html. [Último acceso: 14 agosto 2023].
- [2] AWS, «Bases de datos no relacionales,» [En línea]. Available: <https://aws.amazon.com/es/nosql/>. [Último acceso: 14 agosto 2023].
- [3] IBM, «¿Qué son las bases de datos NoSQL?,» [En línea]. Available: <https://www.ibm.com/mx-es/topics/nosql-databases>. [Último acceso: 14 agosto 2023].
- [4] Oracle México, «¿Qué es NoSQL?,» [En línea]. Available: <https://www.oracle.com/mx/database/nosql/what-is-nosql/>. [Último acceso: 14 agosto 2023].
- [5] Microsoft Corporation, «Base de datos NoSQL: ¿qué es NoSQL?,» [En línea]. Available: <https://azure.microsoft.com/es-mx/resources/cloud-computing-dictionary/what-is-nosql-database>. [Último acceso: 14 agosto 2023].