

Tarea 1: Investigación sobre modelos de bases de datos

Brayan Téllez Cruz

2 de febrero de 2024

0.1. Modelo Orientado a Objetos

0.1.1. ¿Que es el M.O.O?

El Modelo Orientado a Objetos (M.O.O) es una implementación del paradigma Orientado a Objetos dentro de las bases de datos, es decir busca una representación abstracta del mundo real mediante objetos y sus interacciones definidas a través de clases.

Así como en el modelo relacional existen las entidades mediante una representación de tablas, donde cada registro es una nueva instancia de esa entidad, en el M.O.O existen las clases que definen las características de cada abstracción y sus instancias serán objetos, los cuales implementarán los valores reales de cada atributo definido en una clase.

0.1.2. Sistemas administradores de bases de datos orientados a objetos (OODBMS)

Los sistemas de bases da datos orientados a objetos carecen de un modelo común de datos y fundamentos formales, además que su comportamiento de seguridad y manejo de transacciones no están a la altura de los programas actuales de administradores de bases de datos.

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

0.1.3. Características del M.O.O

Dado que está basado en el paradigma orientado a objetos, realiza una implementación de las bases fundamentales de este, esto quiere decir que implementa: •

- Clases: es decir plantillas que definen los atributos y comportamientos de cada elemento abstracto que es definido.
- Objetos: instancias de las clases, o mejor dicho implementación de los valores especificados para cada uno de los atributos definidos para una clase.
- Herencia: definición jerarquica de clases, en la que una clase es capaz de extender las características implementadas por otra (superclase o clase padre), agregando nuevos atributos.

- Polimorfismo: es decir que cada objeto de una clase o de distintas clases de una misma línea de herencia sea capaz de responder de manera distinta a un mismo mensaje recibido.
- Mensaje: es decir un medio para brindar comunicación entre los objetos.

Además entre las características que definen a un modelo orientado a objetos podemos decir que se encuentran. •

- Cada objeto tiene un nombre, atributos y operaciones.
- Un objeto tiene un estado, un comportamiento y una identidad.
- Los mensajes brindan comunicación entre los objetos.
- Los objetos complejos son creados a partir de objetos simples (tipos de datos) como: enteros, caracteres, cadenas de bytes, valores booleanos, números de punto flotante.
- Los objetos complejos pueden ser conjuntos (sets), listas, arreglos, etc.

Así mismo los OODBMS (Oriented Object DataBase Manager System) deben tomar en cuenta las siguientes operaciones: •

- Ser capaces de definir sus propios tipos de datos.
- El tamaño de los datos puede ser muy grande.
- La duración de las transacciones puede ser larga.
- Debe ser capaz de recuperar rápidamente objetos complejos.
- Poseer un lenguaje de consulta de objetos, un ejemplo es el OQL (Object Query Language).
- Poser mecanismos de seguridad basados en la noción de objeto.
- Integrar funciones para definir reglas deductivas (mecanismos de inferencia, para generar información adicional a partir de los datos almacenados en la base).

0.1.4. Manejadores

1. ObjectDB
2. ZODB

0.1.5. Limitaciones de las bases de datos orientadas a objetos

Existen a día de hoy 3 problemas principales que impiden una definición generalizada de estos modelos.

1. Los sistemas de bases de datos relacionales cuentan con especificaciones claras, pero no es así con los orientados a objetos. Se pueden encontrar muchos textos que describen diferentes modelos, pero no hay uno estándar.
2. 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. 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.[1]

0.2. Modelos NoSQL

0.2.1. Clave-valor

¿Qué es el modelo clave valor?

Una base de datos clave-valor es un tipo de base de datos no relacional, también conocido como base de datos NoSQL, que utiliza un método simple de clave-valor para almacenar datos. Almacena datos como un conjunto de pares clave-valor en los que una clave sirve como un identificador único. Tanto las claves como los valores pueden ser cualquier cosa, sean objetos simples o compuestos. Las bases de datos de clave-valor son altamente divisibles y escalables.

Casos de uso

Puede utilizar sistemas de bases de datos de clave-valor como base de datos principal para su aplicación o para gestionar requisitos específicos. •

- Administración de sesiones: cuando un usuario inicia sesión, normalmente se le asigna una clave única que lo identifica a cada usuario y expira en cierto tiempo, usar una base de datos clave valor en estos casos, muchas veces reduce la carga de información con respecto a una base de datos tradicional.
- Almacenamiento de metadatos: puede actuar como una capa de alto nivel para los datos que están almacenados, por ejemplo en el caso de las cargas de multimedia, entretenimiento, como el streaming en tiempo real y el contenido interactivo.

Ventajas

1. Escalabilidad: como todas las consultas de los usuarios requieren la interacción de datos, las bases de datos a menudo pueden convertirse en un obstáculo para el rendimiento de las aplicaciones. Muchas bases de datos ofrecen soporte integrado para funciones de escalado avanzadas. Se escalan horizontalmente y distribuyen automáticamente los datos entre los servidores para reducir obstáculos en un solo servidor.
2. Facilidad de uso: al seguir el paradigma orientado a objetos, permite a los desarrolladores mapear objetos del mundo real directamente a objetos de software. Varios lenguajes de programación, como Java, también siguen ese mismo paradigma. En lugar de mapear sus objetos de código a varias tablas subyacentes, los ingenieros pueden crear pares clave-valor que coincidan con sus objetos de código. Esto hace que el uso de las tiendas de clave-valor sea más intuitivo para los desarrolladores.
3. Rendimiento: procesan operaciones de lectoescritura constantes con llamadas al servidor de bajo costo. La latencia mejorada y el tiempo de respuesta reducido ofrecen un mejor rendimiento a escala. Se basan en estructuras simples de una sola tabla, en lugar de múltiples tablas interrelacionadas. Evitando por ejemplo los joins.[2]

Desventajas

1. Ausencia de consultas complejas: las bases clave-valor no admiten consultas complejas, por lo que los desarrolladores deben resolver este problema en el código. Además de un límite de datos al momento de filtrar y clasificar.
2. Mala administración del esquema: dado a que los esquemas no son fijos, cualquier persona puede modificar el esquema en el programa de base de datos. Los equipos de desarrollo deben planificar el modelo de datos para evitar errores a largo plazo.

Manejadores

1. DinamoDB
2. MemCatcher
3. Redis

0.2.2. Documentos

¿Qué es el modelo de documentos?

Una base de datos de documentos es un tipo de base de datos NoSQL que se puede usar para almacenar y consultar datos como tipo JSON.

Se emplea este formato gracias a la flexibilidad que este tiene abierto al intercambio de datos legibles entre maquinas y humanos.

Manejadores

1. Firebase
2. MongoDB
3. CautchDB

Casos de uso

Dada su alta reciliencia a cambios, pueden ser buenas alternativas por ejemplo en administración de contenidos, creación de catálogos, gestión de sensores, etc.

Puede ser construido para poder representar las vistas de una aplicación, lo que causa que la información pueda presentar redundancia de la información en multiples sitios.

Ventajas

1. Facilidad de desarrollo: los documentos JSON se asignan a objetos de un tipo común a todos los lenguajes de programación, lo que permite a los desarrolladores interactuar con estos objetos facilmente desde el código-
2. Esquema flexible: en este tipo de nases de datos es posible crear varios documentos con diferentes campos dentro de la misma colección. Lo que puede ser una ventaja si contamos con datos no estructurados, se vuelve un problema cuando buscamos estandarizarlos.
3. Rendimiento a escala: las bases de datos basadas en documentos ofrecen distribución integrada, ademas de ser facilmente escalable tanto en forma horizontal como vertical.[3]

Desventajas

1. Inconsistencia en los datos: el desarrollador debe manejar el proceso de actualización de la información para realizar las actualizaciones de forma correcta, en caso contrario existirá inconsistencia en la información.

0.2.3. Grafos

¿Qué es el modelo de grafos?

Es una plataforma especializada y de un solo propósito para crear y manipular grafos. Los grafos contienen nodos, bordes y propiedades que se utilizan para representar y almacenar datos de una forma que no le permiten las bases de datos relacionales.

La analítica de grafos es otro término de uso común y hace referencia específicamente al proceso de analizar información en un formato de grafo utilizando como puntos los nodos y las relaciones como bordes. Se precisa para ello de una base de datos que admita formmato de grafos,

Ventajas

El formato de grafos proporciona una plataforma flexible para detectar conexiones distantes o analizar datos en función de aspectos como la fuerza y calidad de la relación. Esto permite explorar y descubrir conexiones y patrones en redes sociales, IdC, big data, almacenes de datos y también en transacciones complejas para multiples casos de uso comercial, como detección de fraudes en la banca o vista integral del cliente.

Debido a que las bases de datos de este tipo son grafos literalmente, es posible aplicarles los algoritmos de estos como búsquedas e inserciones, lo que permite obtener información en segundos con los algoritmos adecuados en lugar en vez de días, ademas de que no necesitan consultas demasiado complejas.

Las bases orientadas a grafos por tal razón son altamente flexibles y potentes. Gracias al formato del grafo, se pueden determinar relaciones complejas para obtener información más compleja con menos esfuerzo.

Manejadores

1. Neo4j
2. OrientDB
3. TigerGraph
4. GraphEngine

Casos de uso

Algunos de los usos que se les pueden dar a este tipo de bases de datos son.

1. Análisis de medios sociales: entendimiento de las relaciones entre los individuos y la información que reciben y que tanto interactuan o la comparten.
2. Fraude con tarjetas de crédito: buscar cuales son las transferencias recientes que han dado paso al fraude.
3. Blanqueo de dinero: seguimiento de procesos ilegales que dan paso al blanqueo de dinero.[4]

Desventajas

Algunas de las desventajas de este tipo de bases de datos son la complejidad y el espacio que emplean, pues al necesitar guardar todas las relaciones existentes entre diferentes nodos del grafo, conforme más datos sean agregados a la base de datos, mayor cantidad de relaciones tendrán que ser almacenadas y con ello la complejidad de las mismas crece.

1. Presenta consultas difíciles de realizar debido a la complejidad de sus lenguajes de consultas. Un ejemplo es el lenguaje Cípher”.

0.3. Bibliografía

- [1] Unidad de Apoyo para el Aprendizaje, ”Modelo Orientado a Objetos”, https://programas.cuaed.unam.mx/repositorio/moodle/pluginfile.php/782/mod_resource/content/8/contenido/index.html
AWS, ”¿Qué es una base de datos de clave-valor?”, <https://aws.amazon.com/es/nosql/key-value/>
- 3 AWS, ”¿Qué es una base de datos de documentos?”, <https://aws.amazon.com/es/nosql/document/>
- 4 Oracle, ”Definición de base de datos orientada a grafos”, <https://www.oracle.com/mx/autonomous-database/what-is-graph-database/>