

Universidad Nacional Autónoma de México
Facultad de Ingeniería
Bases de Datos
Tarea 1

Blanco Pulido Gabriel Alonzo

13 de agosto de 2024

1. Modelo OODB

En una base de datos orientada a objetos, los componentes se almacenan como objetos y no como datos, al igual que en una de tipo relacional. En una OODB es posible trabajar de manera más sencilla con datos complejos, como objetos anidados y colecciones. Este tipo de bases de datos se compone de elementos fundamentales, tales como:

Objetos: Un objeto es una instancia de una clase, siendo un modelo que puede contar o no con atributos y métodos.

Atributos y métodos: Los atributos son la información esencial que compone a un objeto, mientras que los métodos son las acciones que pueden realizar dichos objetos.

Clases: Es la estructura básica que contiene toda la información del objeto.

Apuntadores: Son de utilidad para tener las direcciones de los objetos y obtenerlos de manera más rápida o relacionarlos entre sí.

Una de las principales características de las OODB es la persistencia de datos de las instancias. En la POO las instancias están almacenadas en memoria transitoria, por lo tanto, al terminar la ejecución del programa, dicha información se pierde y es necesario inicializarla nuevamente, mientras que en una OODB la información de la instancia queda almacenada incluso después de finalizar la ejecución del programa con la transparencia necesaria para poder ser recuperada con lenguaje orientado a objetos.

Manejadores: Versant Object Database, DB4O, ObjectDB, GemStone/S

2. Modelos NoSQL

Descripción, ventajas-desventajas, casos de uso-manejadores

Las bases de datos NoSQL "Not Only SQL" son bases de datos diseñadas para soportar largos volúmenes de información semi-estructurada o no estructurada para almacenamiento o consulta de información, siendo más flexibles que las bases de datos relacionales tradicionales. Hay diversos tipos de bases de datos NoSQL, entre los que se encuentran:

2.1. Clave-Valor

Es un modelo de datos que no sigue un esquema, organizando los datos en el sistema clave-valor, en el que cada elemento cuenta con una clave específica y su respectivo valor, ofreciendo un gran rendimiento al escribir o recibir información ya que usualmente estos datos se encuentran cargados en la memoria del dispositivo.

Ventajas

1. **Simplicidad.** Son bastante simples de implementar y comprender por ser implementadas como diccionarios, además de que una llave puede contener información de diversos

tipos.

2. **Flexibilidad.** Pueden almacenar información de cualquier tipo sin necesidad de tener un modelo a seguir.
3. **Rendimiento.** Al poder ser usadas como tablas hash, tanto la búsqueda como escritura de información es bastante rápida.

Desventajas

1. **Inconsistencias.** El no tener una estructura puede generar inconsistencias y dificultades para relacionar la información.
2. **Consultas complejas.** No es posible hacer consultas más allá de dar una clave y recibir un valor, por lo tanto no es posible hacer consultas que requieran subconsultas, ordenar o filtrar resultados en base a un campo.
3. **Redundancia.** Es posible que al modificar o escribir información se genere redundancia en los datos, teniendo la misma información en llaves diferentes.

Manejadores: Amazon DynamoDB, Redis, Aerospike, Couchbase.

2.2. Documentales

Este tipo de base de datos almacena la información en forma de documentos, siendo los más comunes JSON, XML y BSON. Los datos almacenados pueden ser de diferentes tipos, ya sean números, arreglos, caracteres, booleanos, objetos, etc., siendo de ayuda para la gestión de datos semi-estructurados. Ejemplo de la estructura común de un documento:

```
1 {  
2   "_id": "64a3b2c09f1b4e2aafc4a678",  
3   "make": "Toyota",  
4   "model": "Skyline",  
5   "year": 2000,  
6   "productionDetails": {  
7     "factory": "Toyota Manufacturing Plant",  
8     "line": "Assembly Line 4"  
9   },  
10  "specifications": {  
11    "engine": {  
12      "type": "I6",  
13      "displacement": "3.5L",  
14      "horsepower": 301  
15    }  
16  }
```

Ventajas

1. **Flexibilidad.** No hay restricciones en el formato y estructura de la información que será almacenada, haciendo posible agregar o quitar información sin afectar a otros documentos, siendo independientes unos de otros.
2. **Escalabilidad.** Al ser documentos independientes, pueden contener grandes volúmenes de información no estructurada.
3. **Rendimiento.** Además de poder realizar múltiples consultas, estas se hacen de manera rápida, haciéndolas eficientes en casos donde se requiera recuperar bastante información.

Desventajas

1. **Inconsistencias.** La consistencia entre las versiones de un documento podría llegar a ser un problema, ya que podrían llegar a corromper los datos.
2. **Consultas complejas.** El realizar consultas que requieran revisar más de un documento puede llegar a ser problemático ya que no están relacionados.
3. **Seguridad.** Este tipo de documentos suelen tener volúmenes de información extensos, por lo que en caso de una filtración la cantidad de información comprometida podría ser significativa.

Manejadores: MongoDB, Couchbase, Elasticsearch, Firebase Realtime Database.

2.3. Grafos

Este tipo de base de datos almacena la información en forma de nodos, aristas y propiedades, siendo los nodos una representación de la información almacenada y las aristas la relación que tengan entre sí. En bases de datos con información bastante interconectada son de bastante utilidad para identificar patrones o relaciones complejas entre la información.

Ventajas

1. **Flexibilidad.** El modelo puede adaptarse a los cambios que sean necesarios hacer a lo largo de su implementación.
2. **Escalabilidad.** Es posible guardar información en diferentes servidores, haciendo mayor el volumen de datos que se puede manejar.
3. **Rendimiento.** Al estar la información interconectada, consultas que requieran información almacenada en diferentes nodos pueden ser realizadas muy eficientemente.

Desventajas

1. **Complejidad.** Son más complicadas de diseñar y mantener, por lo tanto, solamente personas con cierto conocimiento podrán manejarlas.
2. **Uso limitado.** No son la primera opción al momento de querer implementar una base de datos del tipo NoSQL, por lo que su uso recae en casos específicos.

Manejadores: Neo4j, Amazon Neptune, ArangoDB, OrientDB.

3. Bibliografía

IBM, “¿Qué es una base de datos NoSQL?” IBM. <https://www.ibm.com/es-es/topics/nosql-databases>

A. Williams, “What is a Document Database?” Knowledge Base by phoenixNAP, Jan. 11, 2024. <https://phoenixnap.com/kb/document-database>

MongoDB, “What is an Object-Oriented Database?” MongoDB. <https://www.mongodb.com/resources/basics/databases/what-is-an-object-oriented-database>

MongoDB, “What is NoSQL? NoSQL databases explained” MongoDB. <https://www.mongodb.com/resources/basics/databases/nosql-explained>

GeeksforGeeks, “Key-Value Data Model in NoSQL,” GeeksforGeeks, Feb. 17, 2022. <https://www.geeksforgeeks.org/key-value-data-model-in-nosql/>

GeeksforGeeks, “Introduction to Graph Database on NoSQL,” GeeksforGeeks, Feb. 22, 2023. <https://www.geeksforgeeks.org/introduction-to-graph-database-on-nosql/>