

Modelo SQL

Representación abstracta de los datos de una base de datos relacional. Es un conjunto de definiciones de tablas, columnas, restricciones y relaciones. El modelo SQL se utiliza para describir la estructura de una base de datos, así como las reglas que rigen los datos que se pueden almacenar en ella.

Se compone de los siguientes elementos:

Tablas: colección de datos relacionados entre sí. Las tablas se componen de filas y columnas.

Columnas: atributo de los datos almacenados en una tabla.

Restricciones: reglas que definen cómo pueden almacenarse los datos en una tabla. Las

Relaciones: conexión entre dos o más tablas. Las relaciones permiten que los datos de diferentes tablas se relacionen entre sí.

Modelo orientado a objetos: se basa en el paradigma de la programación orientada a objetos. En este modelo, los datos se representan como objetos, que tienen atributos y métodos. Los atributos representan los datos del objeto, mientras que los métodos representan las acciones que puede realizar el objeto.

Los sistemas de bases de datos orientados a objetos ofrecen una serie de ventajas respecto a los sistemas de bases de datos relacionales, tales como:

Flexibilidad: Los sistemas de bases de datos orientados a objetos son más adaptables que sus contrapartes relacionales, ya que pueden representar información de diversos tipos y configuraciones.

Reutilización: Estos sistemas pueden ser reutilizados para construir sistemas similares, lo que facilita la creación de aplicaciones relacionadas.

Escalabilidad: Los sistemas de bases de datos orientados a objetos son escalables y pueden adaptarse a sistemas de mayor envergadura.

Desventajas:

Complejidad: Los sistemas de bases de datos orientados a objetos pueden resultar complejos de comprender y mantener debido a la riqueza de sus características y estructuras.

Eficiencia: En ciertos contextos de aplicaciones, los sistemas de bases de datos orientados a objetos pueden no ser tan eficientes como los sistemas relacionales, lo que puede afectar el rendimiento en escenarios específicos.

Manejadores: Objectivity/DB, Versant, ObjectStore

Modelo no SQL

Un modelo NoSQL es un modelo de datos para bases de datos que no se basa en el modelo relacional. Los modelos NoSQL se utilizan para almacenar datos que no se ajustan bien al modelo relacional, como datos semiestructurados o no estructurados. Se dividen en diferentes categorías, según el tipo de datos que almacenan:

Modelo clave-valor: es un modelo de datos para bases de datos que almacena los datos como pares de clave-valor. Cada par de clave-valor consiste en una clave única y un valor asociado. La clave se utiliza para identificar el valor.

Ventajas:

Escalabilidad: Los modelos clave-valor se pueden escalar fácilmente para adaptarse a cargas de trabajo más grandes.

Eficiencia: Los modelos clave-valor son eficientes para operaciones de búsqueda y escritura.

Simpleza: Los modelos clave-valor son fáciles de entender y utilizar.

Desventajas:

Flexibilidad: Los modelos clave-valor no son tan flexibles como los modelos relacionales.

Relaciones: Los modelos clave-valor no pueden representar relaciones entre datos.

Complejidad: Las aplicaciones que utilizan modelos clave-valor pueden ser complejas de diseñar y desarrollar.

Manejadores: Redis, Couchbase, Amazon DynamoDB.

Modelo documental: modelo de datos para bases de datos que almacena los datos como documentos. Los documentos son unidades de datos independientes que pueden contener texto, imágenes, audio y video.

Un modelo documental es un modelo de datos para bases de datos que almacena los datos como documentos. Los documentos son unidades de datos independientes que pueden contener texto, imágenes, audio y video.

Los modelos documentales son una opción popular para aplicaciones que necesitan almacenar grandes cantidades de datos no estructurados o semiestructurados. Por ejemplo, un modelo documental podría utilizarse para almacenar datos sobre un sitio web. En este caso, los documentos podrían representar páginas web, imágenes, videos y otros tipos de contenido.

Ventajas:

Flexibilidad: Los modelos documentales son muy flexibles y se pueden utilizar para representar datos de diferentes tipos y estructuras.

Reutilización: Los modelos documentales se pueden reutilizar para crear aplicaciones similares.

Escalabilidad: Los modelos documentales se pueden escalar fácilmente para adaptarse a cargas de trabajo más grandes.

Desventajas:

Complejidad: Los modelos documentales pueden ser complejos de entender y mantener.

Eficiencia: Los modelos documentales pueden no ser tan eficientes como los modelos relacionales para ciertos tipos de operaciones.

Los modelos documentales son una opción popular para aplicaciones que necesitan almacenar grandes cantidades de datos no estructurados o semiestructurados.

Manejadores: MongoDB, CouchDB, Elasticsearch, Solr, Azure Cosmos DB

Modelo orientado a grafos: almacena los datos como un grafo. Un grafo es una estructura de datos que consta de nodos y aristas. Los nodos representan entidades, mientras que las aristas representan relaciones entre entidades.

Ventajas:

Representación de relaciones complejas: Los modelos orientados a grafos son ideales para representar relaciones complejas entre datos.

Eficiencia para consultas: Los modelos orientados a grafos son eficientes para consultas que involucran relaciones entre datos.

Flexibilidad: Los modelos orientados a grafos son muy flexibles y se pueden utilizar para representar datos de diferentes tipos y estructuras.

Desventajas:

Complejidad: Los modelos orientados a grafos pueden ser complejos de entender y mantener.

Eficiencia para operaciones de escritura: Los modelos orientados a grafos pueden no ser tan eficientes como los modelos relacionales para operaciones de escritura.

Los modelos orientados a grafos son una opción popular para aplicaciones que necesitan representar relaciones complejas entre datos.

Manejadores: Neo4j, GraphDB, ArangoDB, TigerGraph, Amazon Neptune

Referencias:

ISO/IEC 9075-1:2011, Information technology -- Database languages -- SQL -- Part 1: Framework.

Fielding, R. T., Taylor, N., & Lafon, Y. (2009). Scalable, distributed web-based information systems. ACM Transactions on Internet Technology, 10(4), 1-50.

Fielding, R. T., Taylor, N., & Lafon, Y. (2009). Scalable, distributed web-based information systems. ACM Transactions on Internet Technology, 10(4), 1-50.

Elmasri, R., & Navathe, S. B. (2022). Fundamentals of Database Systems (7th ed.). Pearson Education.