HUGO IVAN ROMERO DUARTE
15-5-24

INVESTIGACIÓN

Las bases de datos SQL y NoSQL son dos tipos de sistemas de gestión de bases de datos que se utilizan para almacenar y recuperar información. Cada uno tiene sus propias características, ventajas y desventajas, y se utilizan en diferentes situaciones según las necesidades del proyecto.

SQL (Lenguaje de Consulta Estructurado) es un lenguaje de programación creado a principios de la década de 1970. SQL se utiliza ampliamente para la gestión de datos en sistemas de gestión de bases de datos relacionales. Este lenguaje permite realizar consultas en bases de datos relacionales, las cuales reconocen las relaciones entre los elementos almacenados. Ejemplos de bases de datos relacionales SQL son MySQL, Oracle, Microsoft SQL Server, PostgreSQL, entre otros.

Las bases de datos NoSQL son bases de datos no relacionales (a diferencia de las bases de datos SQL, que sí lo son). Surgieron a finales de la década de 2000, cuando la productividad del desarrollador se volvió más importante que los costos de almacenamiento. Un punto crucial es que NoSQL no significa que estas bases de datos nunca usen SQL. (Existen bases de datos NoSQL que admiten y utilizan SQL). En cambio, es mejor pensar en NoSQL como "no solo SQL". Ejemplos de bases de datos NoSQL son Google Cloud BigTable, Apache HBase, Redis, MongoDB, Cassandra, entre otros.

Las principales diferencias entre ambos modelos de bases de datos son!:

Las bases de datos SQL almacenan datos de manera estructurada y las NoSQL lo hacen en su formato original.

Las bases de datos SQL utilizan tablas con columnas fijas (atributos) y filas (registros). Estas bases de datos siguen reglas específicas relacionadas con la integridad y la consistencia de los datos.

Las bases de datos NoSQL no se adhieren a este formato rígido y son más flexibles.

Las bases de datos SQL proporcionan una capacidad de escalar baja, en comparación con las NoSQL.

En cuanto a cuándo usar cada una, depende de las necesidades específicas del proyecto. Las bases de datos SQL son ideales para situaciones donde se requiere una estructura de datos consistente y definida, mientras que las bases de datos NoSQL son más adecuadas para proyectos que requieren flexibilidad y escalabilidad

MySQL con TypeScript!:

Instala el paquete mysql2 con npm: npm install mysql2.

Crea un archivo para manejar la conexión a MySQL, por ejemplo dbConnection.ts. En este archivo, utiliza el módulo mysql2 para conectarte a la base de datos.

TypeScript

HUGO IVAN ROMERO DUARTE
15-5-24

INVESTIGACIÓN

```
import mysql from 'mysql2/promise';

async function connect() {
  try {
    const connection = await mysql.createConnection({
      host: 'localhost',
      user: 'tu_usuario',
      password: 'tu_contraseña',
      database: 'nombre_de_tu_db',
    });
    console.log('Conexión a MySQL establecida.');
    return connection;
  } catch (error) {
    console.error('Error al conectar a MySQL:', error);
    throw error;
  }
}

export default connect;
```

Ahora puedes importar esta función en cualquier parte de tu aplicación para interactuar con la base de datos.

MongoDB con TypeScript2:

Instala el paquete mongodb con npm: npm install mongodb.

HUGO IVAN ROMERO DUARTE
15-5-24

INVESTIGACIÓN

Crea un archivo para manejar la conexión a MongoDB, por ejemplo dbConnection.ts. En este archivo, utiliza el módulo mongodb para conectarte a la base de datos.

```
import { MongoClient } from 'mongodb';

const uri = 'mongodb://localhost:27017';  // URL de conexión a tu base de datos
const dbName = 'mydatabase';  // Nombre de tu base de datos

async function connect() {
  try {
    const client = new MongoClient(uri);
    await client.connect();
    const db = client.db(dbName);
    console.log('Conexión a MongoDB establecida.');
    return db;
  } catch (error) {
    console.error('Error al conectar a MongoDB:', error);
    throw error;
  }
}
export default connect;
```

Ahora puedes importar esta función en cualquier parte de tu aplicación para interactuar con la base de datos