

Bases de datos SQL y NoSQL

¿Qué son las bases de datos SQL?

SQL (Structured Query Language) es un lenguaje de consulta estandarizado para interactuar con bases de datos relacionales. Estas bases de datos organizan los datos en tablas con filas y columnas, donde cada fila representa un registro individual y cada columna representa un atributo específico de ese registro.

Características principales de las bases de datos SQL:

- Estructura rígida: Los datos se almacenan en tablas con un esquema predefinido, lo que garantiza integridad y consistencia.
- Lenguaje de consulta potente: SQL proporciona un lenguaje estructurado para consultar, insertar, actualizar y eliminar datos de manera eficiente.
- Escalabilidad probada: Las bases de datos SQL son altamente escalables y pueden manejar grandes volúmenes de datos.
- Soporte transaccional robusto: Ofrecen sólidas garantías de integridad de datos para transacciones complejas.

Casos de uso comunes para bases de datos SQL:

- Aplicaciones empresariales: Sistemas de contabilidad, gestión de clientes, inventarios y nóminas.
- Sitios web transaccionales: Tiendas en línea, plataformas de banca y portales de reservas.
- Aplicaciones de registro de datos: Sistemas de monitoreo, análisis de rendimiento y registros de eventos.

¿Qué son las bases de datos NoSQL?

NoSQL es un término general que abarca una amplia gama de bases de datos no relacionales. Estas bases de datos no siguen el modelo tradicional de tablas y relaciones, sino que ofrecen mayor flexibilidad y escalabilidad para manejar datos no estructurados o semiestructurados.

Características principales de las bases de datos NoSQL:

- Esquema flexible: Permiten almacenar datos en diversos formatos, como documentos JSON, gráficos o clave-valor, sin un esquema predefinido.

- Escalabilidad horizontal: Pueden distribuirse fácilmente en múltiples servidores para manejar grandes volúmenes de datos en constante crecimiento.
- Alta disponibilidad: Ofrecen replicación de datos y tolerancia a fallos para garantizar un acceso continuo a la información.
- Modelo de datos diverso: Admiten diferentes modelos de datos, como documentos, gráficos y clave-valor, para adaptarse a diversas necesidades.

Casos de uso comunes para bases de datos NoSQL:

- Aplicaciones web a gran escala: Redes sociales, plataformas de comercio electrónico y servicios de transmisión de datos.
- Aplicaciones móviles: Almacenamiento de datos del usuario, notificaciones y análisis en tiempo real.
- Internet de las cosas (IoT): Gestión de grandes volúmenes de datos de sensores y dispositivos conectados.
- Análisis de Big Data: Procesamiento y análisis de conjuntos de datos masivos y no estructurados.

¿Cuándo usar SQL vs NoSQL?

- Estructura de los datos: Si los datos son estructurados y relacionales, SQL es una buena opción. Si los datos son no estructurados, semiestructurados o varían con el tiempo, NoSQL puede ser más adecuado.
- Volumen de datos: Si se manejan grandes volúmenes de datos en constante crecimiento, NoSQL ofrece mejor escalabilidad horizontal.
- Patrones de acceso a datos: Si se necesitan consultas complejas con relaciones entre tablas, SQL es más eficiente. Si se realizan consultas simples o se accede a datos sin relaciones, NoSQL puede ser suficiente.
- Requisitos de disponibilidad: Si se requiere alta disponibilidad y tolerancia a fallos, NoSQL suele ser la mejor opción.

¿Cómo se implementa MySQL en Node JS?

Instalar los módulos

```
npm install express  
npm install mysql
```

Crear y exportar objeto de conexión mysql

```
const mysql = require("mysql");  
  
let db_con = mysql.createConnection({  
  host: "localhost",  
  user: "root",  
  password: "  
});  
  
db_con.connect((err) => {  
  if (err) {  
    console.log("Database Connection Failed !!!", err);  
  } else {  
    console.log("connected to Database");  
  }  
});  
  
module.exports = db_con;
```

Crear servidor

```
const express = require("express");  
const database = require('./sqlConnection');  
  
const app = express();  
  
app.listen(5000, () => {  
  console.log(`Server is up and running on 5000 ...`);  
});
```

Crear ruta para crear base de datos y usarla

```
app.get("/createDatabase", (req, res) => {  
  
  let databaseName = "gfg_db";
```

```
let createQuery = `CREATE DATABASE ${databaseName}`;

// use the query to create a Database.
database.query(createQuery, (err) => {
  if(err) throw err;

  console.log("Database Created Successfully !");

  let useQuery = `USE ${databaseName}`;
  database.query(useQuery, (error) => {
    if(error) throw error;

    console.log("Using Database");

    return res.send(
`Created and Using ${databaseName} Database`);
  })
});
});
```

Requerimiento: Es un servicio que solo se dedica a enviar notificaciones por diversos canales (SMS, push, email)

Se recomienda utilizar una base de datos NoSQL para este servicio de notificaciones. Las bases de datos NoSQL ofrecen mayor flexibilidad, escalabilidad y adaptabilidad a las características de este tipo de aplicación.

- El servicio necesita almacenar una gran cantidad de datos no estructurados, como el contenido de las notificaciones y las preferencias de los usuarios.
- El servicio necesita ser escalable para poder manejar un aumento en el número de usuarios y el volumen de datos.
- El servicio necesita ser flexible para poder adaptarse a nuevos canales de notificación y a nuevos tipos de datos.