



## **Universidad Autónoma de Querétaro**

### **Facultad de Informática**

**Actividad:** Investigación acerca de las base de datos y su implementación

**Alumno:** Duarte Fregoso Josafat Axel

**Profesor:** Enrique Aguilar

**Fecha:** 17/05/2024

En el mundo moderno de la tecnología, la gestión y almacenamiento de datos es fundamental para el funcionamiento de cualquier aplicación. Las bases de datos, que se dividen principalmente en SQL y NoSQL, juegan un papel crucial en esta tarea.

## **Bases de Datos SQL y NoSQL**

Las bases de datos SQL, o de lenguaje de consulta estructurada, se basan en un esquema relacional y utilizan tablas para organizar datos. Este tipo de bases de datos se caracteriza por tener un esquema rígido y estructurado, lo que facilita la integridad y consistencia de los datos. Entre las características destacadas de SQL se encuentran las transacciones ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad), que garantizan que las operaciones de la base de datos se realicen de manera confiable. Ejemplos comunes de bases de datos SQL incluyen MySQL, PostgreSQL y SQL Server.

Por otro lado, las bases de datos NoSQL, o "Not Only SQL", son un conjunto diverso de tecnologías diseñadas para manejar grandes volúmenes de datos no estructurados y semiestructurados. Estas bases de datos ofrecen esquemas dinámicos, lo que permite una mayor flexibilidad en la gestión de datos. NoSQL se subdivide en varias categorías, como bases de datos de documentos (MongoDB), de columnas (Cassandra), de clave-valor (Redis) y de gráficos (Neo4j). La escalabilidad horizontal y la capacidad de manejar datos en tiempo real son características prominentes de las bases de datos NoSQL.

## **Aplicaciones y Usos de SQL y NoSQL**

Las bases de datos SQL son ideales para aplicaciones empresariales donde la consistencia y la integridad de los datos son críticas. Se utilizan comúnmente en sistemas financieros, de contabilidad y de gestión de relaciones con clientes (CRM). La estructura rígida y las relaciones complejas entre tablas hacen que SQL sea adecuado para escenarios donde el esquema de datos no cambia con frecuencia y donde las transacciones deben ser fiables.

En contraste, las bases de datos NoSQL son preferidas para aplicaciones que requieren alta escalabilidad y capacidad para manejar grandes volúmenes de datos en tiempo real. Estas incluyen redes sociales, sistemas de gestión de contenidos y aplicaciones de big data. La

flexibilidad en el esquema y la capacidad de escalar horizontalmente permiten a NoSQL adaptarse rápidamente a los cambios en los requisitos de datos y soportar cargas de trabajo variables.

## Implementación en Node.js

Node.js, un entorno de ejecución para JavaScript, facilita la integración tanto con bases de datos SQL como NoSQL. Para SQL, se puede utilizar PostgreSQL mediante el módulo `pg`, lo que permite ejecutar consultas y manejar transacciones de manera eficiente.

Por ejemplo:

```
const { Pool } = require('pg');
const pool = new Pool({
  user: 'user',
  host: 'localhost',
  database: 'mydb',
  password: 'password',
  port: 5432,
});

pool.query('SELECT NOW()', (err, res) => {
  console.log(err, res);
  pool.end();
});
```

Para NoSQL, MongoDB es una opción popular que se puede integrar mediante el módulo mongodb. Esto permite la manipulación de datos no estructurados de manera flexible y eficiente.

Un ejemplo de su implementación sería:

```
const { MongoClient } = require('mongodb');
const url = 'mongodb://localhost:27017';
const dbName = 'mydb';

MongoClient.connect(url, function(err, client) {
  if (err) throw err;
  const db = client.db(dbName);
  db.collection('documents').find({}).toArray(function(err, docs) {
```

```
    if (err) throw err;
    console.log(docs);
    client.close();
  });
});
```

## **Elección de Base de Datos para un Sistema de Notificaciones**

Consideremos un sistema de notificaciones que necesita enviar mensajes a través de diferentes canales, como SMS, notificaciones push y correo electrónico. Un patrón de diseño orientado a la mensajería es adecuado para este propósito, ya que puede manejar eficientemente el envío de notificaciones y escalar según sea necesario.

Para este caso, se recomienda utilizar una base de datos NoSQL, específicamente MongoDB. La razón principal es la flexibilidad del esquema que MongoDB ofrece, permitiendo almacenar diferentes tipos de notificaciones sin necesidad de modificar el esquema de la base de datos. Además, la escalabilidad horizontal de MongoDB es ideal para aplicaciones que requieren manejar grandes volúmenes de datos y crecer de manera eficiente. Su rendimiento en tiempo real garantiza que las notificaciones se envíen de manera oportuna, y su alta disponibilidad, proporcionada por la replicación y la tolerancia a fallos, asegura que el sistema de notificaciones esté siempre en funcionamiento.

## Referencias

- Connolly, T., & Begg, C. (2015). \*Database Systems: A Practical Approach to Design, Implementation, and Management\* (6th ed.). Pearson.
- Sadalage, P. J., & Fowler, M. (2012). \*NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence\*. Addison-Wesley Professional.
- Chodorow, K. (2013). \*MongoDB: The Definitive Guide\*. O'Reilly Media.
- PostgreSQL Global Development Group. (2020). \*PostgreSQL 13 Documentation\*. <https://www.postgresql.org/docs/13/index.html>
- MongoDB Inc. (2020). \*MongoDB Manual\*. <https://docs.mongodb.com/manual/>