Actividad 1

¡-) diferencias de base de datos relacionales y no relacionales

R/ Bases de Datos Relacionales:

Estructura Tabular:

Los datos se organizan en tablas con filas y columnas.

Cada fila representa una entidad y cada columna es un atributo de esa entidad.

Ejemplo: MySQL, PostgreSQL, Oracle.

Esquema Fijo y Riguroso:

Requieren un esquema predefinido que especifica la estructura de la base de datos antes de almacenar datos.

Cambiar el esquema puede ser complicado y costoso.

Transacciones ACID:

Adherencia a propiedades ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad).

Garantizan la integridad de los datos incluso en situaciones de fallo.

Escalabilidad Vertical:

Mejor rendimiento con una escalabilidad vertical, aumentando la capacidad de la máquina.

Bases de Datos No Relacionales (NoSQL):

Diversidad de Modelos de Datos:

Pueden utilizar modelos como clave-valor, documentos, grafos o columnares, adaptándose a diferentes tipos de datos y requisitos.

Ejemplos: MongoDB (documentos), Cassandra (columnares), Redis (clave-valor).

Esquema Dinámico:

No requieren un esquema fijo y permiten la adición de campos sin cambiar la estructura general.

Mayor flexibilidad para manejar datos no estructurados o semi-estructurados.

Escalabilidad Horizontal:

Suelen escalar horizontalmente, distribuyendo la carga entre múltiples servidores o nodos.

Adecuado para manejar grandes volúmenes de datos y altas cargas de trabajo.

Consistencia eventual:

Algunas bases de datos NoSQL sacrifican la consistencia inmediata (eventual consistency) para lograr una mayor disponibilidad y tolerancia a fallos.

Uso Específico:

Suelen ser más adecuadas para casos de uso específicos como almacenamiento y recuperación rápida de datos no estructurados o semi-estructurados.

La elección entre una base de datos relacional y no relacional generalmente depende de los requisitos específicos del proyecto, como la estructura de los datos, la escalabilidad, la consistencia y la flexibilidad. No hay una respuesta única para todos los casos, y la elección puede depender del contexto y las necesidades del sistema.

2-) Cuadro comparativo ventajas/ desventajas

|  |  |  |
| --- | --- | --- |
| Aspecto | Bases de Datos Relacionales | Bases de Datos NoSQL |
| Modelo de Datos | Estructura tabular, relaciones definidas | Diversidad de modelos: clave-valor, documentos, grafos, columnares, etc. |
| Esquema | Esquema fijo y riguroso | Esquema dinámico, adaptable, permite cambios sin alterar la estructura global |
| Escalabilidad | Escalabilidad vertical (añadir recursos a una máquina) | Escalabilidad horizontal (añadir nodos o servidores) |
| Consistencia | Transacciones ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad) | Consistencia eventual (algunos sacrifican la consistencia inmediata) |
| Flexibilidad | Menos flexible en términos de tipos de datos y cambios en el esquema | Mayor flexibilidad para datos no estructurados o semi-estructurados |
| Rendimiento | Buen rendimiento para consultas complejas y relaciones bien definidas | Rendimiento eficiente para operaciones específicas, pero puede variar |
| Ejemplos de Uso | Sistemas donde las relaciones entre datos son críticas (Sistemas de gestión, aplicaciones empresariales) | Situaciones donde la escalabilidad y la flexibilidad son más importantes (Big Data, aplicaciones web, IoT) |
| Ejemplos de Tecnologías | MySQL, PostgreSQL, Oracle | MongoDB, Cassandra, Redis, CouchDB |
| Costo | A menudo, licencias y costos asociados con sistemas RDBMS | A menudo, código abierto y gratuito, con opciones comerciales disponibles |
| Aprendizaje Inicial | Requiere entender relaciones y diseño de esquemas | Menos restricciones en diseño, puede ser más fácil de aprender inicialmente |

3-) Uso para les base datos no relacionales/relacionales

R/

No relacionales

1. Manejo de Grandes Volúmenes de Datos (Big Data):

Las bases de datos NoSQL son efectivas para gestionar grandes cantidades de datos distribuidos en varios nodos o servidores.

Aplicaciones Web y Móviles:

1. Son ideales para entornos web y móviles donde la escalabilidad horizontal es crucial para manejar un gran número de usuarios concurrentes.

Almacenamiento de Datos No Estructurados o Semi-Estructurados:

1. Pueden manejar datos flexibles y no estructurados, como documentos JSON, XML, o datos de tipo clave-valor.

Sistemas de Gestión de Contenido (CMS):

1. Las bases de datos NoSQL son útiles para almacenar y recuperar contenido multimedia, como imágenes, videos y texto, en sistemas de gestión de contenido.

Análisis en Tiempo Real:

1. Se utilizan en situaciones que requieren análisis en tiempo real de datos, como sistemas de recomendación, análisis de registros y monitoreo en tiempo real.

Aplicaciones de Redes Sociales:

1. Son adecuadas para el manejo de perfiles de usuarios, relaciones y actividades en redes sociales.

(Internet de las cosas):

1. Pueden manejar grandes flujos de datos generados por dispositivos IoT debido a su capacidad de escalabilidad horizontal.

Juegos en Línea:

1. Para juegos en línea y plataformas de juego, donde la velocidad y la escalabilidad son críticas para manejar grandes cantidades de usuarios concurrentes.

Desarrollo Ágil y Prototipado Rápido:

1. Permiten cambios de esquema sin restricciones, facilitando el desarrollo ágil y prototipado rápido.

Aplicaciones con Cambios Frecuentes en el Esquema:

1. Son útiles en situaciones donde el esquema de datos puede cambiar con frecuencia debido a la evolución de la aplicación.

Relacionales

1. Sistemas de Gestión Empresarial (ERP):
2. Se utilizan para gestionar datos empresariales como finanzas, recursos humanos, inventario y procesos empresariales.

Sistemas de Gestión de Bases de Datos Tradicionales:

1. Donde las relaciones entre los datos son cruciales y necesitan ser gestionadas de manera rigurosa.

Sistemas Financieros:

1. Para el manejo de transacciones financieras, seguimiento de cuentas, informes y análisis.

Sistemas de Reservas y Comercio Electrónico:

1. En aplicaciones que involucran transacciones, gestión de inventario y relaciones de productos.

Sistemas de Información Geográfica (GIS):

1. Para el almacenamiento y manejo de datos espaciales y geográficos.

Sistemas de Recursos Humanos:

1. En la gestión de datos relacionados con empleados, salarios, beneficios y evaluaciones de desempeño.

Aplicaciones de Misión Crítica:

1. Donde la integridad de los datos y la consistencia son fundamentales, como sistemas médicos y aeroespaciales.

Sistemas de Información de Clientes (CRM):

1. Para almacenar y gestionar información sobre clientes, relaciones comerciales y actividades de ventas.

Aplicaciones Analíticas Complejas:

1. Cuando se requieren consultas complejas y análisis de datos utilizando lenguajes como SQL.

Sistemas de Información de Inventarios y Logística:

Para el seguimiento y gestión de inventarios, así como la planificación logística.