

UNIVERSIDAD DE LOS ÁNGELES

INGENIERÍA EN SISTEMAS
COMPUTACIONALES

CUARTO SEMESTRE

DOCENTE: Martin Cruz Medinilla

ESTUDIANTE: Abigail Mendoza Ortega

ÍNDICE

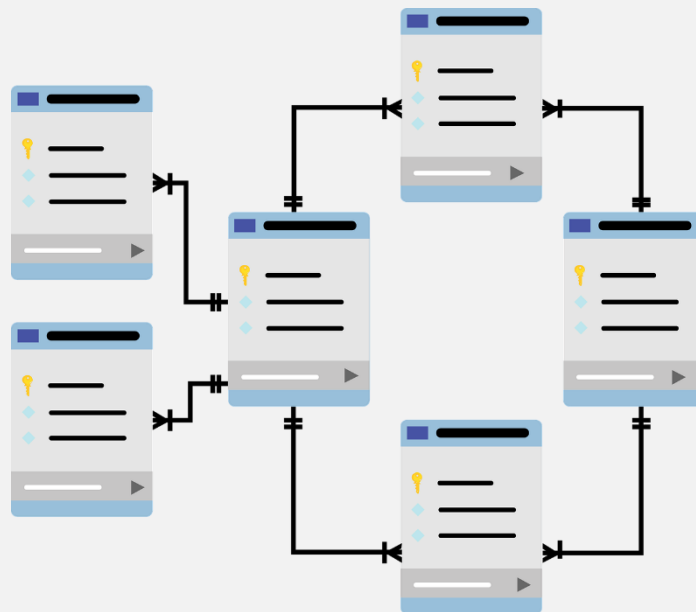
BASES DE DATOS RELACIONALES	2
BASES DE DATOS NOSQL	3
DIFERENCIAS ENTRE BASES DE DATOS RELACIONALES Y BASES DE DATOS NOSQL.....	4
CONCLUSIÓN.....	5

BASES DE DATOS RELACIONALES

Las bases de datos relacionales son un tipo de sistema de gestión de bases de datos (SGBD) que organiza los datos en tablas relacionadas entre sí. Estas tablas están compuestas por filas y columnas, donde cada fila representa una entidad específica y cada columna representa un atributo o característica de esa entidad.

Algunas de sus principales características son:

- Los datos se almacenan en tablas, que consisten en filas (tuplas) y columnas (atributos).
- Las relaciones entre las tablas se establecen mediante claves primarias y claves foráneas. Las claves primarias son columnas únicas que identifican de manera única cada fila en una tabla, mientras que las claves foráneas establecen relaciones entre tablas al hacer referencia a las claves primarias de otras tablas.
- Las bases de datos relacionales garantizan la integridad referencial, lo que significa que las relaciones entre las tablas se mantienen consistentes y que no se pueden crear relaciones entre entidades inexistentes.
- Las operaciones comunes en bases de datos relacionales incluyen la selección, inserción, actualización y eliminación de datos utilizando el lenguaje estructurado de consulta (SQL).
- La normalización es un proceso que se utiliza para organizar los datos de manera eficiente, reduciendo la redundancia y mejorando la integridad de los datos.



BASES DE DATOS NOSQL

Las bases de datos NoSQL (Not Only SQL) son sistemas de gestión de bases de datos diseñados para manejar grandes volúmenes de datos no estructurados. Las bases de datos NoSQL se utilizan en una variedad de aplicaciones y casos de uso, incluyendo aplicaciones web y móviles, análisis de big data, IoT (Internet de las cosas), juegos en línea y sistemas de comercio electrónico.

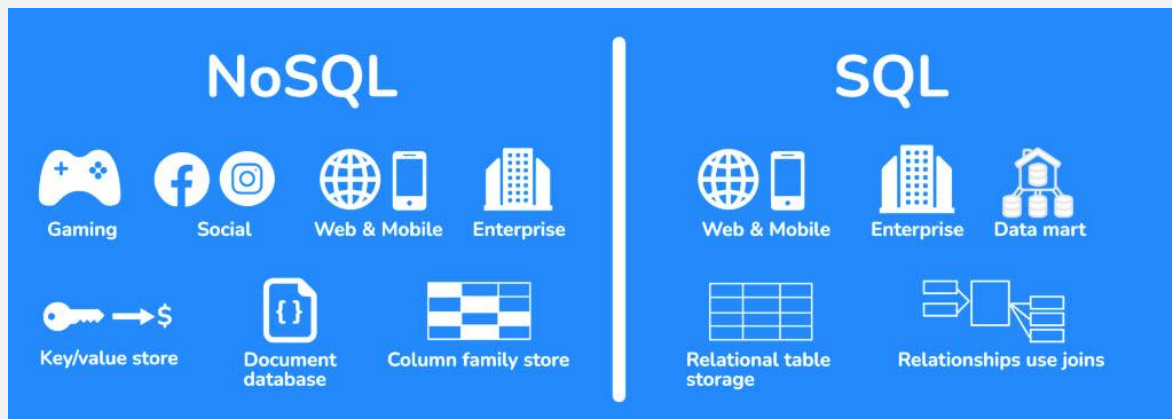
Características:

- Los sistemas NoSQL pueden utilizar varios modelos de datos, como documentos, grafos, columnares o clave-valor, lo que permite almacenar y manipular datos de formas más diversas que los modelos relacionales tradicionales.
- Los sistemas NoSQL están diseñados para escalar horizontalmente, lo que significa que pueden distribuir la carga de trabajo a través de múltiples servidores para manejar grandes volúmenes de datos y altas cargas de trabajo.
- Muchas bases de datos NoSQL están diseñadas para proporcionar alta disponibilidad y tolerancia a fallos mediante la replicación de datos y la distribución geográfica de los servidores.
- A diferencia de las bases de datos relacionales, los sistemas NoSQL no requieren un esquema fijo, lo que permite agregar, modificar o eliminar campos de manera flexible sin afectar el rendimiento o la integridad de los datos.
- Algunos sistemas NoSQL están optimizados para consultas específicas o patrones de acceso a datos, lo que puede mejorar el rendimiento en aplicaciones con requisitos de consulta particularmente exigentes



DIFERENCIAS ENTRE BASES DE DATOS RELACIONALES Y BASES DE DATOS NOSQL

NoSQL	SQL
No es relacional.	Es relacional.
Se basa en documentos.	Se basa en tablas.
Su estructura es dinámica (para bases de datos orientadas a documentos).	Mantiene una estructura definida previamente diseñada.
Permite tener redundancia de datos.	Se diseña para tener la menor redundancia de datos posibles.
Para aplicaciones que requieren grandes cantidades de datos a consultar.	Se usa donde se requieren datos consistentes y sin errores.



CONCLUSIÓN

Elegir entre una base de datos relacional y una base de datos NoSQL depende de varios factores, como los requisitos de la aplicación, el tipo de datos que se van a manejar, el rendimiento esperado, la escalabilidad y como se manejaran esos datos.

En algunos casos, se han utilizado ambos tipos de bases de datos, esto conocido como arquitectura políglota, que puede ser la solución más apropiada para satisfacer las necesidades de una aplicación o sistema más compleja.