

Cuadro Comparativo de Bases de Datos Relacionales, No Relacionales y Multimodelo

Bases de Datos Relacionales

Definición:

Son sistemas que almacenan la información en **tablas** formadas por **filas y columnas**. Cada tabla representa una entidad (por ejemplo, *Estudiante* o *Curso*), y las relaciones entre ellas se definen mediante **claves primarias y foráneas**.

Ejemplo sencillo:

Una tabla "Estudiantes" puede tener las columnas: ID, Nombre, Apellido, DNI, FechaNacimiento. Otra tabla "Cursos" tiene IDCurso, NombreCurso, Créditos. Ambas pueden estar relacionadas por una tabla "Matrícula" que guarda el ID del estudiante y el ID del curso.

Lenguaje usado:

Utilizan **SQL (Structured Query Language)** para consultar y modificar datos.

Ventajas:

- Estructura organizada y consistente.
- Permiten mantener la **integridad de los datos**.
- Excelentes para sistemas donde los datos son estructurados (como bancos o universidades).
- Soportan transacciones seguras (ACID: Atomicidad, Consistencia, Aislamiento y Durabilidad).

Desventajas:

- Dificultan la gestión de datos no estructurados (como imágenes, JSON, o grandes volúmenes).
- Escalabilidad limitada (crecer horizontalmente es más complicado).

Ejemplos de sistemas:

- MySQL
- PostgreSQL
- SQL Server
- Oracle Database

Usos comunes:

- Sistemas financieros.
- Aplicaciones empresariales (ERP, CRM).
- Registros académicos o administrativos.

Bases de Datos No Relacionales (NoSQL)

Definición:

Las bases de datos **No Relacionales** (o **NoSQL**) son sistemas que **no utilizan tablas** ni relaciones estrictas como en las bases de datos relacionales.

Se diseñaron para manejar **grandes volúmenes de datos**, especialmente **no estructurados o semiestructurados**, como documentos, redes sociales, registros de sensores o contenido multimedia.

Estructura:

Los datos se almacenan en formatos más flexibles, según el tipo de base NoSQL:

- **Clave-Valor:** Cada dato se guarda con una clave única (por ejemplo, Redis, Amazon DynamoDB).
- **Documentos:** Guardan información en formato JSON o BSON (por ejemplo, MongoDB, CouchDB).
- **Columnas:** Organizan los datos por columnas en lugar de filas (por ejemplo, Cassandra, HBase).
- **Grafos:** Almacenan nodos y relaciones, ideales para redes sociales (por ejemplo, Neo4j).

Lenguaje usado:

No tienen un lenguaje estándar como SQL.

Cada sistema tiene su propia forma de consulta, por ejemplo:

- MongoDB usa consultas en formato JSON.
- Cassandra utiliza CQL (Cassandra Query Language).
- Neo4j usa Cypher.

Ventajas:

- **Alta velocidad y rendimiento**, especialmente con grandes volúmenes de datos.
- **Escalabilidad horizontal**, es decir, se pueden agregar más servidores fácilmente.
- **Flexibilidad en los esquemas**, no es necesario definir las columnas antes de almacenar datos.
- Adecuadas para aplicaciones distribuidas o en la nube.

Desventajas:

- No garantizan siempre la consistencia total de los datos (a veces priorizan disponibilidad y rendimiento).
- No existe un lenguaje unificado estándar.
- Menor soporte para transacciones complejas.

Ejemplos de sistemas:

- **MongoDB** (documentos)
- **Cassandra** (columnas)
- **Redis** (clave-valor)

- **Neo4j** (grafos)

Usos comunes:

- Aplicaciones web y móviles con gran cantidad de usuarios.
- Sistemas de análisis de datos en tiempo real.
- Plataformas de redes sociales.
- Big Data y análisis de comportamiento de usuarios.

Bases de Datos Multimodelo

Definición:

Las **bases de datos multimodelo** son sistemas modernos que **combinan varios modelos de datos** (relacional, documental, de grafos, clave-valor, etc.) en una sola plataforma.

Esto permite manejar **diferentes tipos de información** sin necesidad de usar varias bases de datos distintas.

En otras palabras, **un mismo sistema** puede almacenar datos estructurados, semiestructurados y no estructurados al mismo tiempo.

Estructura:

Estas bases permiten crear y consultar datos usando distintos enfoques:

- Tablas relacionales (como en SQL).
- Documentos JSON o XML.
- Nodos y relaciones (como en grafos).
- Pares clave-valor.

Por ejemplo, un sistema multimodelo puede guardar información de usuarios en tablas relacionales, los mensajes en formato documento y las conexiones entre ellos como un grafo, todo dentro de una misma base de datos.

Lenguaje usado:

Usan una **combinación de SQL y lenguajes específicos** según el modelo de datos.

Por ejemplo:

- SQL para tablas.
- JSONPath o consultas anidadas para documentos.
- Cypher para grafos.

Ventajas:

- **Versatilidad:** se adaptan a diferentes tipos de datos y aplicaciones.
- **Eficiencia:** reducen la necesidad de usar varios sistemas de bases de datos.
- **Consultas híbridas:** se pueden mezclar datos relacionales y no relacionales en una sola consulta.
- **Ideal para entornos empresariales complejos** que manejan distintos formatos de información.

Desventajas:

- **Mayor complejidad** en la administración y configuración.
- **Costos más altos** de implementación y mantenimiento.
- Requiere conocimientos avanzados para aprovechar todas sus funciones.

Ejemplos de sistemas:

- **ArangoDB**
- **Microsoft Azure Cosmos DB**
- **OrientDB**
- **Couchbase**

Usos comunes:

- Aplicaciones empresariales con múltiples fuentes de datos.
- Sistemas de análisis avanzado (combinan datos estructurados y no estructurados).
- Plataformas en la nube con diferentes tipos de usuarios y servicios.

Cuadro Comparativo de Bases de Datos

Aspecto	Bases de Datos Relacionales	Bases de Datos No Relacionales (NoSQL)	Bases de Datos Multimodelo
Estructura	Organizadas en tablas con filas y columnas. Cada tabla representa una entidad y se relaciona con otras mediante claves primarias y foráneas.	Utilizan estructuras flexibles como documentos, grafos, pares clave-valor o columnas amplias.	Integran múltiples modelos de datos en una misma base (relacional, documental, grafo, clave-valor, etc.).
Modelo de Datos	Basado en el modelo relacional , propuesto por Edgar F. Codd, con relaciones entre tablas.	Se basa en modelos NoSQL : documento, clave-valor, columna o grafo.	Permite trabajar con varios modelos de datos simultáneamente , en una sola plataforma.
Lenguaje de Consulta	Usa SQL (Structured Query Language) , un lenguaje estandarizado.	Cada sistema tiene su propio lenguaje (por ejemplo: JSON en MongoDB, CQL en Cassandra, Cypher en Neo4j).	Soporta SQL y lenguajes especializados para cada tipo de modelo (SQL + JSON + Cypher).
Tipo de Datos	Datos estructurados , con formato definido (tablas).	Datos no estructurados o semiestructurados , como JSON, texto o multimedia.	Datos estructurados y no estructurados al mismo tiempo.
Escalabilidad	Escalabilidad vertical : se mejora el rendimiento aumentando recursos al mismo servidor.	Escalabilidad horizontal : se añaden más servidores para distribuir los datos.	Puede escalar vertical u horizontalmente , según el modelo de datos.
Integridad de los	Alta integridad,	Menor integridad,	Alta integridad, al

Datos	gracias al uso de reglas y relaciones.	prioriza flexibilidad y rendimiento.	combinar reglas relacionales con flexibilidad NoSQL.
Ventajas	<ul style="list-style-type: none"> - Estructura organizada y estandarizada. - Alta consistencia. - Ideal para sistemas transaccionales. 	<ul style="list-style-type: none"> - Alta escalabilidad. - Gran rendimiento. - Flexibilidad en esquemas. - Ideal para Big Data y nube. 	<ul style="list-style-type: none"> - Versátil y adaptable. - Maneja múltiples modelos. - Permite consultas híbridas.
Desventajas	<ul style="list-style-type: none"> - Difícil de escalar horizontalmente. - Menor rendimiento en datos no estructurados. 	<ul style="list-style-type: none"> - Falta de estandarización. - Menor soporte para transacciones complejas. 	<ul style="list-style-type: none"> - Mayor complejidad técnica. - Costos de implementación elevados.
Ejemplos de Sistemas	MySQL, PostgreSQL, SQL Server, Oracle Database.	MongoDB, Cassandra, Redis, Neo4j.	ArangoDB, Cosmos DB, OrientDB, Couchbase.
Casos de Uso	Sistemas empresariales, financieros, académicos, ERP, CRM.	Redes sociales, aplicaciones móviles, Big Data, IoT.	Empresas con datos variados, análisis en la nube, plataformas integradas.