|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nombre de la práctica** | Análisis de Manejadores de bases de Datos | | | **No.** | 1 |
| **Asignatura:** | Administracion de base de datos | **Carrera:** | ISIC | **Duración de la práctica (Hrs)** |  |

**Nombre: Jocelin Reyes Rodriguez**

**Grupo: 3601**

**1.2: Análisis de los Manejadores de Bases de Datos:**

**Actividades**

Contenido

[**1.- Investigación de SGBD:**](#_Toc190584653)

[Oracle:](#_Toc190584654)

[**Modelo de Datos:**](#_Toc190584655)

[**Soporte para Transacciones:**](#_Toc190584656)

[**Escalabilidad:**](#_Toc190584657)

[**Seguridad:**](#_Toc190584658)

[**Herramientas de Analisis y Reportes:**](#_Toc190584659)

[MySQL](#_Toc190584660)

[**Modelo de Datos:**](#_Toc190584661)

[**Soporte para Transacciones:**](#_Toc190584662)

[**Escalabilidad:**](#_Toc190584663)

[**Seguridad:**](#_Toc190584664)

[**Herramientas de Analisis y Reportes:**](#_Toc190584665)

[**Modelo de Datos:**](#_Toc190584666)

[**Soporte para Transacciones:**](#_Toc190584667)

[**Escalabilidad:**](#_Toc190584668)

[**Seguridad:**](#_Toc190584669)

[**Herramientas de Analisis y Reportes:**](#_Toc190584670)

[MongoDB](#_Toc190584671)

[**Modelo de Datos:**](#_Toc190584672)

[**Soporte para Transacciones:**](#_Toc190584673)

[**Escalabilidad:**](#_Toc190584674)

[**Seguridad:**](#_Toc190584675)

[**Herramientas de Analisis y Reportes:**](#_Toc190584676)

[**3.- Estudio de Casos**](#_Toc190584677)

[**Caso 1: MySQL en Facebook**](#_Toc190584678)

[**Caso 2: Microsoft SQL Server en Bank of America**](#_Toc190584679)

[**Caso 3: Oracle en Walmart**](#_Toc190584680)

[**2.- Tabla Comparativa**](#_Toc190584681)

# **1.- Investigación de SGBD:**

## Oracle:

Oracle Database es un sistema de gestión de bases de datos empresarial altamente optimizado para entornos críticos y de alto rendimiento.

### **Modelo de Datos:**

**Tipo:** Relacional (SQL) con soporte para **NoSQL, JSON, XML y datos en columnas**.

**Lenguaje de consulta:** SQL extendido con **PL/SQL para procedimientos almacenados**.

**Soporte para bases de datos autónomas en Oracle Cloud.**

### **Soporte para Transacciones:**

**Cumple con ACID** con optimización avanzada de concurrencia y bloqueo.

**Soporte para transacciones distribuidas y replicación en tiempo real.**

**Oracle RAC (Real Application Clusters):** Permite alta disponibilidad en múltiples servidores.

### **Escalabilidad:**

**Extremadamente escalable** tanto en entornos locales como en la nube.

**Optimizado para procesamiento en paralelo, análisis de datos y AI.**

### **Seguridad:**

**Autenticación multifactor y cifrado avanzado.**

**Políticas de seguridad a nivel de usuario, tabla y fila.**

**Soporte para auditoría y cumplimiento con normativas empresariales (GDPR, HIPAA).**

### **Herramientas de Analisis y Reportes:**

**Oracle BI y Oracle Analytics Cloud.**

**Integración con Power BI, Tableau y sistemas de ETL.**

## MySQL

MySQL es una base de datos relacional de código abierto, ampliamente utilizada en aplicaciones web y sistemas de gestión empresarial debido a su rapidez y facilidad de uso.

### **Modelo de Datos:**

**Tipo:** Relacional (SQL).

**Estructura:** Basada en **tablas** con relaciones entre los datos mediante **claves primarias y foráneas**.

**Lenguaje de consulta:** SQL estándar con extensiones propias.

**Tipos de datos soportados:** Numéricos, cadenas, fechas, JSON (limitado)

### **Soporte para Transacciones:**

**Cumple con ACID**: Maneja transacciones atómicas, consistentes, aisladas y duraderas.

**Motor de almacenamiento InnoDB:** Soporta **bloqueos de filas**, **rollback**, y **integridad referencial**.

**Motor MyISAM:** No soporta transacciones ni integridad referencial, pero ofrece mayor rapidez en lectura.

### **Escalabilidad:**

**Escalado vertical:** Mejor rendimiento aumentando recursos del servidor.

**Escalado horizontal:** Mediante **sharding** y **replicación maestro-esclavo**, aunque con ciertas limitaciones.

**Soporte para clustering:** Disponible con **MySQL Cluster**, pero requiere configuración avanzada

### **Seguridad:**

**Autenticación por usuarios y roles** con privilegios detallados.

**Cifrado SSL/TLS** para conexiones seguras.

**Protección contra inyección SQL** mediante consultas parametrizadas.

### **Herramientas de Analisis y Reportes:**

**MySQL Workbench**: Herramienta gráfica para diseño y administración.

**Integración con Tableau, Power BI, Metabase** y otros sistemas BI.

Microsoft SQL Server

SQL Server es una base de datos relacional robusta, ideal para entornos empresariales y analítica avanzada.

### **Modelo de Datos:**

Relacional (SQL) con soporte para JSON, XML y datos en columnas.

### **Soporte para Transacciones:**

**Cumple con ACID**: Garantiza Atomicidad, Consistencia, Aislamiento y Durabilidad en todas las transacciones.

**MVCC (Multi-Version Concurrency Control):** Permite múltiples versiones de un mismo dato, evitando bloqueos de lectura/escritura.

**Soporte para transacciones distribuidas:** Coordina transacciones entre múltiples bases de datos o servidores.

**Mecanismo de bloqueo avanzado:** Optimiza la concurrencia evitando bloqueos innecesarios en tablas y registros.

### **Escalabilidad:**

**Escalado vertical:** Mejora el rendimiento aumentando la capacidad del hardware (RAM, CPU, almacenamiento).

**Escalado horizontal:** Mediante **particionamiento de bases de datos, replicación y clustering**.

**Always On Availability Groups:** Permite alta disponibilidad y recuperación ante desastres con múltiples réplicas sincronizadas.

**Optimización para Big Data:** Integración con Apache Spark y Hadoop para procesamiento de datos masivos.

### **Seguridad:**

**Autenticación basada en Active Directory**: Permite acceso controlado con credenciales corporativas.

**Cifrado de datos avanzado:** Soporta cifrado en tránsito y en reposo (Transparent Data Encryption - TDE).

**Dynamic Data Masking:** Oculta datos sensibles en consultas para proteger la información sin modificarla en la base de datos.

**Row-Level Security (RLS):** Permite restringir el acceso a filas específicas de una tabla según el usuario o rol.

**Auditoría avanzada:** Monitorea y registra todas las transacciones para cumplir con normativas de seguridad.

### **Herramientas de Analisis y Reportes:**

**SQL Server Management Studio (SSMS):** Herramienta oficial para administración y consultas.

**SQL Server Reporting Services (SSRS):** Generación de reportes avanzados con visualización interactiva.

**SQL Server Integration Services (SSIS):** Permite la integración y transformación de datos desde múltiples fuentes.

**SQL Server Analysis Services (SSAS):** Optimización de análisis de datos y minería de datos con soporte para cubos OLAP.

**Integración con Power BI, Tableau y Excel para visualización de datos.**

## MongoDB

### **Modelo de Datos:**

MongoDB es una base de datos NoSQL orientada a documentos, ideal para aplicaciones modernas y escalables.

### **Soporte para Transacciones:**

**Tipo:** No relacional (NoSQL), basado en documentos JSON (BSON).

**Estructura:** Flexible, sin esquema fijo.

**Ideal para datos semiestructurados y dinámicos.**

### **Escalabilidad:**

**Escalabilidad horizontal con sharding automático.**

**Soporte nativo para replicación y clustering.**

### **Seguridad:**

**Autenticación basada en roles y cifrado SSL.**

**Control de acceso granular.**

### **Herramientas de Analisis y Reportes:**

**MongoDB Compass**: Herramienta de gestión visual.

**Integración con Power BI, Tableau y Grafana.**

# **3.- Estudio de Casos**

## **Caso 1: MySQL en Facebook**

**Problema o necesidad:**

Facebook necesitaba un sistema de almacenamiento eficiente para gestionar millones de usuarios y sus interacciones en tiempo real, además de sus publicaciones.

**¿Por qué se eligió MySQL?**

* Es un SGBD de código abierto, lo que permitió personalizarlo.
* Tiene un alto rendimiento y escalabilidad.
* Soporta replicación, lo que ayuda en la distribución de datos.

**Resultados y beneficios:**

* Facebook optimizó MySQL con su propia infraestructura para manejar grandes volúmenes de datos.
* Se logró alta disponibilidad y rapidez en las consultas.
* La arquitectura escalable permitió que la red social creciera sin problemas.

## **Caso 2: Microsoft SQL Server en Bank of America**

**Problema o necesidad:**

Bank of America necesitaba una base de datos segura y escalable para gestionar transacciones bancarias en tiempo real, cumplir con regulaciones financieras y ofrecer alta disponibilidad a sus clientes.

**¿Por qué se eligió Microsoft SQL Server?**

* Ofrece seguridad avanzada con cifrado y control de accesos.
* Tiene un excelente soporte para transacciones ACID, garantizando integridad de datos.
* Permite integración con herramientas de análisis y BI (como Power BI).
* Su capacidad de replicación y clustering ofrece alta disponibilidad.

**Resultados y beneficios:**

* Se mejoró la velocidad y confiabilidad en el procesamiento de transacciones.
* Redujo el riesgo de fraude mediante auditoría y seguridad avanzada.
* Facilitó el análisis de datos en tiempo real para la toma de decisiones estratégicas. Logró escalabilidad sin interrupciones en el servicio.

## **Caso 3: Oracle en Walmart**

**Problema o necesidad:**

Walmart requería una base de datos robusta para manejar su cadena de suministro y millones de transacciones diarias.

**¿Por qué se eligió Oracle?**

* Proporciona una arquitectura segura y de alto rendimiento.
* Tiene herramientas avanzadas para análisis de datos.
* Soporta transacciones de gran escala sin afectar el rendimiento.

**Resultados y beneficios:**

* Walmart logró mejorar la eficiencia en la gestión de inventarios.
* Redujo costos operativos gracias a la automatización de procesos.
* Pudo realizar análisis predictivos para optimizar su cadena de suministro.

# **2.- Tabla Comparativa**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | MySQL | PostgreSQL | Oracle | MongoDB | Microsoft SQL Server |
| Modelo de datos | Relacional  Organiza los datos en tablas con filas y columnas, usando claves primarias y foráneas para mantener relaciones.  Ventajas   * Eficiente y rápido en consultas. * Integridad y seguridad con claves y permisos. * Escalable para grandes volúmenes de datos. * Soporta ACID (transacciones confiables).   Desventajas   * Escalabilidad horizontal limitada frente a NoSQL. * Menos flexible para datos no estructurados. * Bloqueos en alta concurrencia. * Consultas complejas lentas | Relacional-Orientado a Objetos.  Ventajas   * Alta flexibilidad: Soporta JSON, XML y datos geoespaciales. * Escalabilidad: Maneja grandes volúmenes con particionamiento y paralelización. * Seguridad avanzada: Cifrado, autenticación y control de acceso detallado.   Desventajas   * Más consumo de recursos en comparación con MySQL. * Curva de aprendizaje más pronunciada debido a su complejidad. * Menor velocidad en lecturas simples que MySQL en algunos casos. | Relacional y Multimodelo  Oracle usa un modelo relacional pero también soporta datos JSON, XML, gráficos y geoespaciales.  Ventajas   * Alto rendimiento y escalabilidad en grandes volúmenes de datos. * Seguridad avanzada con cifrado, auditoría y control de accesos detallado. * Soporte para transacciones ACID, garantizando integridad de datos.   Desventajas   * Costoso, especialmente para grandes implementaciones * Complejidad en administración, requiere conocimientos avanzados. | Orientado a Documentos  donde los datos se almacenan en colecciones como documentos JSON/BSON, en lugar de tablas relacionales.  Ventajas   * Alta flexibilidad: No necesita esquemas rígidos. * Escalabilidad horizontal: Se adapta bien a grandes volúmenes. * Alto rendimiento en lecturas y escrituras rápidas.   Desventajas   * No garantiza ACID en todas las operaciones * Mayor uso de almacenamiento debido a la duplicación de datos. * Menos eficiente en consultas complejas comparado con SQL. | Relacional y Multimodelo  incluye capacidades multimodelo para manejar JSON, XML  Ventajas   * Alto rendimiento con optimización de consultas e índices avanzados. * Seguridad robusta con cifrado, autenticación y control de acceso detallado. * Escalabilidad con soporte para grandes volúmenes de datos y clustering.   Desventajas   * Alto consumo de recursos, requiere hardware potente. * Costos elevados, especialmente en versiones empresariales. * Curva de aprendizaje más compleja para principiantes. |
| Soporte para transacciones | MySQL admite transacciones con InnoDB, siguiendo el modelo ACID para garantizar integridad y confiabilidad. Usa COMMIT, ROLLBACK y niveles de aislamiento para controlar la ejecución segura de operaciones.  Ventajas   * Garantiza integridad de datos con ACID. * Permite revertir cambios en caso de errores. * Soporte para distintos niveles de aislamiento.   Desventajas   * No todos los motores lo soportan (ej. MyISAM). * Mayor consumo de recursos y riesgo de bloqueos. * Puede ser más lento en alta concurrencia. | PostgreSQL ofrece un soporte completo cumpliendo con ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad). Utiliza COMMIT, ROLLBACK y SAVEPOINT para manejar operaciones de manera segura.  Ventajas   * Cumplimiento estricto de ACID * Soporte de transacciones anidadas con SAVEPOINT. * Manejo eficiente de concurrencia con MVCC (Multiversion Concurrency Control).   Desventajas   * Mayor consumo de recursos en comparación con MySQL. * Complejidad en administración para optimizar rendimiento. * Posible crecimiento del almacenamiento por MVCC si no se gestiona bien el VACUUM. | Oracle ofrece transacciones completas con ACID, utilizando COMMIT, ROLLBACK y SAVEPOINT. Soporta transacciones distribuidas y control de concurrencia optimista.  Ventajas   * Alta confiabilidad y consistencia. * Optimización de concurrencia y transacciones distribuidas. * Recuperación eficiente con FLASHBACK.   Desventajas   * Elevado consumo de recursos. * Alta complejidad administrativa. * Bloqueos posibles en alta concurrencia. * Costos elevados en licencias. | MongoDB soporta transacciones ACID desde la versión 4.0, permitiendo transacciones multi-documento con COMMIT y ABORT.  Ventajas   * Soporte ACID para transacciones multi-documento. * Flexibilidad en bases de datos NoSQL. * Recuperación confiable ante fallos.   Desventajas   * Menor rendimiento que en bases SQL. * Solo en replica sets y versiones recientes. * Mayor uso de recursos en transacciones complejas. | SQL Server soporta transacciones ACID con COMMIT, ROLLBACK y SAVEPOINT, y permite transacciones distribuidas y varios niveles de aislamiento.  Ventajas   * Cumplimiento ACID y alta confiabilidad. * Soporte para transacciones distribuidas. * Recuperación ante fallos con log de transacciones.   Desventajas   * Mayor uso de recursos en alta concurrencia. * Posibles bloqueos y deadlocks. * Licenciamiento costoso. |
| Escalabilidad | MySQL es escalable verticalmente (mejorando hardware) y horizontalmente (replicación, sharding, clustering).  Ventajas   * Escalabilidad vertical fácil. * Replicación para alta disponibilidad. * Sharding para grandes volúmenes de datos.   Desventajas   * Escalabilidad horizontal compleja. * Sharding difícil de gestionar. * Rendimiento limitado sin optimización. | PostgreSQL es escalable verticalmente y horizontalmente (con replicación y herramientas como Citus para sharding).  Ventajas   * Escalabilidad vertical fácil. * Replicación síncrona y asíncrona. * Herramientas de sharding como Citus.   Desventajas   * Sharding nativo limitado. * Configuración avanzada requerida. * Complejo en grandes volúmenes distribuidos. | Oracle es escalable verticalmente y horizontalmente con RAC, sharding y replicación.  Ventajas   * RAC para alta disponibilidad. * Sharding nativo para grandes volúmenes. * Soporte para grandes aplicaciones empresariales.   Desventajas   * Costoso y complejo de gestionar. * Requiere personal especializado. | MongoDB es escalable horizontalmente con sharding y replicación para alta disponibilidad.  Ventajas   * Escalabilidad horizontal fácil. * Alta disponibilidad con replicación. * Ideal para grandes volúmenes de datos distribuidos.   Desventajas   * Sharding complejo. * No tan eficiente en consultas complejas. | SQL Server es escalable verticalmente y horizontalmente mediante clustering, replicación y particionamiento de datos.  Ventajas   * Escalabilidad vertical fácil. * Clustering y replicación para alta disponibilidad. * Particionamiento de datos para distribuir la carga. * Optimización de consultas complejas.   Desventajas   * Mayor consumo de recursos en grandes entornos. * Complejidad en la gestión de escalabilidad horizontal. * Licenciamiento costoso para características avanzadas. |
| Seguridad | MySQL ofrece autenticación, control de acceso, cifrado y SSL/TLS para conexiones seguras.  Ventajas   * Autenticación avanzada y control de acceso. * Cifrado de datos en reposo y en tránsito. * SSL/TLS para conexiones seguras.   Desventajas   * Cifrado limitado en algunas versiones. * Configuración manual para seguridad óptima. * Funciones avanzadas solo en versiones comerciales. | PostgreSQL ofrece autenticación, cifrado en reposo y en tránsito, y SSL/TLS para conexiones seguras.  Ventajas   * Autenticación avanzada y control por roles. * Cifrado de datos y conexiones seguras. * Permisos granulares.   Desventajas   * Cifrado en reposo requiere herramientas externas. * Configuración compleja. * Funciones avanzadas requieren extensiones. | Oracle ofrece autenticación avanzada, cifrado de datos en reposo y en tránsito, y auditoría detallada.  Ventajas   * Autenticación y control de acceso granulares. * Cifrado completo de datos. * Auditoría y Data Masking.   Desventajas   * Costoso y requiere licencias adicionales. * Configuración compleja. * Requiere personal especializado. | MongoDB ofrece autenticación, control de acceso, cifrado en reposo y en tránsito, y auditoría.  Ventajas   * Autenticación y control de acceso por roles. * Cifrado de datos y auditoría. * Integración con LDAP.   Desventajas   * Cifrado en reposo requiere configuración extra. * Algunas funciones avanzadas solo en versiones comerciales. * Gestión de seguridad compleja en grandes implementaciones. | SQL Server ofrece autenticación, cifrado (TDE, SSL/TLS), y auditoría.  Ventajas   * Autenticación y control de acceso sólidos. * Cifrado completo de datos. * Auditoría avanzada.   Desventajas   * Funciones avanzadas solo en versiones Enterprise. * Mayor consumo de recursos con cifrado. * Configuración compleja. |
| Herramientas de análisis y reportes | MySQL usa herramientas externas como MySQL Workbench, Grafana, Tableau y Pentaho.  Ventajas   * Compatible con herramientas de BI. * Workbench ofrece consultas básicas. * Integración con Python y R.   Desventajas   * No tiene herramientas nativas avanzadas. * Consultas analíticas pueden ser lentas. * Requiere herramientas externas. | PostgreSQL usa herramientas externas como pgAdmin, Metabase, Tableau, Power BI y Grafana para análisis y reportes.  Ventajas   * pgAdmin para consultas y reportes básicos. * Compatible con BI y análisis avanzado. * Soporte para SQL avanzado y JSON.   Desventajas   * No tiene herramientas nativas avanzadas. * Consultas complejas requieren optimización. * Requiere integración con herramientas externas. | Oracle ofrece herramientas como Oracle BI, Oracle Analytics Cloud, SQL Developer y compatibilidad con Power BI y Tableau.  Ventajas   * Oracle BI y Analytics Cloud para reportes avanzados. * SQL Developer para consultas y análisis. * Alta integración con herramientas de BI.   Desventajas   * Alto costo en herramientas avanzadas. * Curva de aprendizaje elevada. * Requiere infraestructura potente para grandes volúmenes de datos. | herramientas como MongoDB Charts, Compass, Tableau, Power BI y Grafana para visualización y análisis de datos.  Ventajas   * MongoDB Charts para reportes nativos. * Compass para análisis básico de datos. * Compatibilidad con BI y visualización externa.   Desventajas   * MongoDB Charts es básico comparado con BI avanzados. * Consultas analíticas menos eficientes que en bases SQL. * Depende de herramientas externas para reportes complejos. | SQL Server ofrece SQL Server Reporting Services (SSRS), SQL Server Analysis Services (SSAS) y compatibilidad con Power BI, Tableau y Excel.  Ventajas   * SSRS para reportes detallados. * SSAS para análisis avanzado y cubos OLAP. * Integración nativa con Power BI y Excel.   Desventajas   * SSRS y SSAS requieren configuración avanzada. * Algunas funciones avanzadas solo en versiones Enterprise. * Curva de aprendizaje alta para análisis complejos. |
| Uso | MySQL se usa en aplicaciones web, sistemas de gestión de contenido (CMS), e-commerce y bases de datos empresariales. Es popular en WordPress, Magento y aplicaciones LAMP.  Ventajas   * Rápido y eficiente para aplicaciones web. * Fácil de usar e integrar con múltiples lenguajes. * Código abierto y comunidad activa.   Desventajas   * Menos eficiente en consultas complejas comparado con otros motores. * Limitado en escalabilidad horizontal. * Algunas funciones avanzadas solo en versiones comerciales. | PostgreSQL se usa en aplicaciones empresariales, análisis de datos, GIS, fintech y sistemas que requieren alta integridad y escalabilidad.  Ventajas   * Soporta consultas complejas y grandes volúmenes de datos. * Extensibilidad y compatibilidad con JSON, XML y NoSQL. * Código abierto y altamente personalizable.   Desventajas   * Consumo de recursos mayor que MySQL en ciertas operaciones. * Curva de aprendizaje más alta. * Menos herramientas de administración visual que otros motores. | Oracle se usa en aplicaciones empresariales y grandes bases de datos.  Ventajas   * Alta disponibilidad y confiabilidad. * Soporta grandes volúmenes de datos y consultas complejas. * Características avanzadas.   Desventajas   * Costoso. * Requiere personal especializado. * Configuración compleja. | MongoDB se usa en aplicaciones web, móviles, IoT y sistemas que requieren alta escalabilidad y flexibilidad. Es ideal para datos no estructurados y proyectos ágiles.  Ventajas   * Alta escalabilidad horizontal. * Flexible con datos no estructurados y esquemas dinámicos. * Rendimiento rápido para operaciones de lectura/escritura.   Desventajas   * Menos adecuado para consultas complejas. * Falta de integridad referencial en comparación con bases de datos SQL. * Requiere más almacenamiento debido a su estructura. | Microsoft SQL Server se usa en aplicaciones empresariales, sistemas de gestión de bases de datos, análisis de datos y almacenamiento de grandes volúmenes de información en sectores como finanzas, salud y comercio.  Ventajas   * Alta fiabilidad y rendimiento. * Integración fácil con otros productos Microsoft. * Soporta grandes volúmenes de datos y consultas complejas.   Desventajas   * Costoso en comparación con otras bases de datos. * Requiere hardware robusto y licencias caras. * Curva de aprendizaje elevada en configuraciones avanzadas. |
| Costo | MySQL es gratuito en su versión comunitaria, con costos adicionales en la versión Enterprise.  Ventajas   * Gratis para proyectos pequeños. * Costo bajo y soporte opcional.   Desventajas   * Versiones comerciales pueden ser caras. * Licencia GPL tiene restricciones en entornos comerciales. | PostgreSQL es totalmente gratuito y de código abierto bajo la licencia PostgreSQL, sin costos de licencias.  Ventajas   * Completamente gratuito y sin restricciones de uso. * Sin costos adicionales para funciones avanzadas. * Soporte de una comunidad activa.   Desventajas   * Sin soporte comercial oficial, a menos que se contraten servicios externos. * Costos de infraestructura pueden aumentar con escalabilidad y grandes volúmenes de datos. | Oracle es costoso, con licencias comerciales y costos de soporte.  Ventajas   * Soporte y características avanzadas. * Alta fiabilidad y rendimiento.   Desventajas   * Costos elevados. * Modelo de precios complejo. * Requiere infraestructura cara. | MongoDB ofrece una versión gratuita (MongoDB Community) y versiones comerciales como MongoDB Atlas con precios según el uso, características y soporte.  Ventajas   * Versión gratuita para proyectos pequeños. * Escalabilidad en la versión comercial sin costos iniciales altos. * Modelo de precios flexible según el uso.   Desventajas   * Versiones comerciales pueden ser caras dependiendo de la escala y soporte. * Costos adicionales por almacenamiento y procesamiento en la nube (MongoDB Atlas) | Microsoft SQL Server tiene licencias comerciales con precios que varían según la edición (Express, Standard, Enterprise) y el número de usuarios o procesadores.  Ventajas   * Licencia flexible según las necesidades (por usuario o por procesador). * Edición gratuita disponible (SQL Server Express) para aplicaciones pequeñas. * Escalabilidad en versiones comerciales.   Desventajas   * Altos costos de licencias para versiones Standard y Enterprise. * Costo adicional por soporte y actualizaciones. * Requiere infraestructura potente para grandes implementaciones. |
| Comunidad de soporte | MySQL tiene una comunidad activa con foros, grupos de discusión y recursos en línea, además de la opción de soporte comercial mediante MySQL Enterprise.  Ventajas   * Comunidad grande y activa, con abundante documentación y tutoriales gratuitos. * Soporte en foros y plataformas como Stack Overflow. * Acceso a herramientas de diagnóstico y optimización.   Desventajas   * Soporte limitado en la versión gratuita (solo foros y documentación). * Falta de soporte directo o personalizado en la comunidad. * Tiempo de respuesta puede ser lento para problemas complejos sin soporte comercial. | PostgreSQL tiene una comunidad activa con foros, listas de correo y documentación.  Ventajas   * Gran comunidad y documentación gratuita. * Soporte gratuito en foros y grupos. * Contribuciones constantes de expertos.   Desventajas   * Tiempo de respuesta lento en problemas complejos. * No hay soporte comercial oficial. * Resolución difícil sin soporte especializado. | Oracle tiene comunidad activa y soporte comercial oficial.  Ventajas   * Soporte comercial rápido. * Comunidad activa y gran documentación.   Desventajas   * Soporte gratuito limitado. * Soporte comercial costoso. | MongoDB tiene una comunidad activa con foros, canales en Slack y documentación, además de ofrecer soporte comercial a través de MongoDB Atlas.  Ventajas   * Comunidad activa con foros, Slack y recursos en línea. * Documentación extensa y tutoriales gratuitos. * Soporte comercial disponible con MongoDB Atlas.   Desventajas   * Soporte gratuito limitado en comparación con la opción comercial. * Tiempo de respuesta puede ser lento en problemas avanzados sin soporte comercial. * Falta de soporte oficial gratuito personalizado. | Microsoft SQL Server tiene una comunidad activa en foros, blogs, y documentación oficial, además de ofrecer soporte comercial a través de Microsoft Support.  Ventajas   * Comunidad activa con foros y recursos gratuitos. * Amplia documentación oficial de Microsoft. * Soporte comercial rápido y especializado disponible.   Desventajas   * Soporte gratuito limitado para problemas avanzados. * Soporte comercial costoso. * Respuestas en la comunidad pueden ser lentas para problemas complejos. |
| Rendimiento | MySQL es conocido por su alto rendimiento en aplicaciones web y bases de datos de tamaño medio, con operaciones de lectura/escritura rápidas.  Ventajas   * Rendimiento rápido en consultas simples y transacciones. * Optimización para operaciones de lectura. * Escalabilidad eficiente en implementaciones pequeñas a medianas.   Desventajas   * Rendimiento puede disminuir en consultas complejas o grandes volúmenes de datos. * Falta de soporte nativo para consultas analíticas avanzadas. * Escalabilidad limitada en bases de datos extremadamente grandes. | PostgreSQL ofrece alto rendimiento en consultas complejas, manejo de grandes volúmenes de datos y soporte para transacciones ACID.  Ventajas   * Excelente rendimiento en consultas complejas y operaciones analíticas. * Optimización avanzada para grandes bases de datos. * Soporta consultas concurrentes de alto rendimiento.   Desventajas   * Menor rendimiento en operaciones simples en comparación con MySQL. * Mayor consumo de recursos para consultas simples o en aplicaciones de bajo volumen. * Puede ser más lento en algunos casos de escalabilidad horizontal. | Oracle ofrece un rendimiento excepcional en bases de datos de misión crítica, grandes volúmenes de datos y transacciones complejas.  Ventajas   * Alta capacidad de procesamiento para grandes volúmenes de datos y transacciones. * Optimización avanzada con características como particionamiento y paralelismo. * Excelente manejo de concurrencia y transacciones.   Desventajas   * Mayor consumo de recursos en comparación con otras bases de datos. * Costos adicionales para optimización y características avanzadas. * Requiere hardware potente para rendir a su máxima capacidad. | MongoDB ofrece un buen rendimiento en operaciones de escritura rápida y escalabilidad horizontal, especialmente para grandes volúmenes de datos no estructurados.  Ventajas   * Excelente para grandes volúmenes de datos no estructurados. * Escalabilidad horizontal mediante sharding. * Rendimiento rápido en escrituras y consultas simples.   Desventajas   * Menor rendimiento en consultas complejas. * Falta de optimización para operaciones transaccionales complejas. * Puede requerir recursos significativos en configuraciones grandes y distribuidas. | Microsoft SQL Server ofrece alto rendimiento en operaciones complejas, consultas analíticas y transacciones a gran escala, con optimizaciones avanzadas.  Ventajas   * Rendimiento optimizado para grandes bases de datos y transacciones complejas. * Soporte para consultas analíticas y procesamiento de grandes volúmenes de datos.   Mejoras en rendimiento con herramientas como el SQL Server Profiler y optimización automática.  Desventajas   * Mayor consumo de recursos para consultas simples. * Escalabilidad limitada sin configuraciones adicionales. * Requiere infraestructura robusta y costos adicionales para rendimiento óptimo. |

**Caso 1: MySQL en Facebook**

**Problema o necesidad:**

Facebook necesitaba un sistema de almacenamiento eficiente para gestionar millones de usuarios y sus interacciones en tiempo real, además de sus publicaciones.

**¿Por qué se eligió MySQL?**

* Es un SGBD de código abierto, lo que permitió personalizarlo.
* Tiene un alto rendimiento y escalabilidad.
* Soporta replicación, lo que ayuda en la distribución de datos.

**Resultados y beneficios:**

* Facebook optimizó MySQL con su propia infraestructura para manejar grandes volúmenes de datos.
* Se logró alta disponibilidad y rapidez en las consultas.
* La arquitectura escalable permitió que la red social creciera sin problemas.

Caso 2: Microsoft SQL Server en Bank of America

**Problema o necesidad:**

Bank of America necesitaba una base de datos segura y escalable para gestionar transacciones bancarias en tiempo real, cumplir con regulaciones financieras y ofrecer alta disponibilidad a sus clientes.

**¿Por qué se eligió Microsoft SQL Server?**

* Ofrece seguridad avanzada con cifrado y control de accesos.
* Tiene un excelente soporte para transacciones ACID, garantizando integridad de datos.
* Permite integración con herramientas de análisis y BI (como Power BI).
* Su capacidad de replicación y clustering ofrece alta disponibilidad.

**Resultados y beneficios:**

* Se mejoró la velocidad y confiabilidad en el procesamiento de transacciones.
* Redujo el riesgo de fraude mediante auditoría y seguridad avanzada.
* Facilitó el análisis de datos en tiempo real para la toma de decisiones estratégicas. Logró escalabilidad sin interrupciones en el servicio.

**Conclucion:**

La adopción de Microsoft SQL Server permitió a Bank of America garantizar la seguridad, escalabilidad y alta disponibilidad necesarias para gestionar transacciones bancarias en tiempo real. Gracias a su soporte para transacciones ACID, seguridad avanzada y herramientas de análisis, la entidad mejoró la confiabilidad en el procesamiento de datos, redujo riesgos de fraude y optimizó la toma de decisiones estratégicas. Este caso demuestra cómo una infraestructura de base de datos robusta es clave para el sector financiero.

Caso 3: Oracle en Walmart

**Problema o necesidad:**

Walmart requería una base de datos robusta para manejar su cadena de suministro y millones de transacciones diarias.

**¿Por qué se eligió Oracle?**

* Proporciona una arquitectura segura y de alto rendimiento.
* Tiene herramientas avanzadas para análisis de datos.
* Soporta transacciones de gran escala sin afectar el rendimiento.

**Resultados y beneficios:**

* Walmart logró mejorar la eficiencia en la gestión de inventarios.
* Redujo costos operativos gracias a la automatización de procesos.
* Pudo realizar análisis predictivos para optimizar su cadena de suministro.

**Conclucion:**

La implementación de Oracle permitió a Walmart optimizar la gestión de su cadena de suministro mediante una arquitectura segura y de alto rendimiento. Gracias a sus herramientas avanzadas de análisis de datos y su capacidad para manejar transacciones a gran escala, la empresa mejoró la eficiencia operativa, redujo costos y aprovechó el análisis predictivo para optimizar su logística. Este caso demuestra la importancia de contar con una base de datos robusta en empresas con alto volumen de operaciones.