

Práctica 1.2 Análisis de los Manejadores de Bases de Datos

Objetivo

Analizar y comparar diferentes sistemas de gestión de bases de datos (SGBD) en función de sus características, ventajas y desventajas, para tomar decisiones informadas sobre cuál utilizar en un proyecto específico.

1. Investigación de SGBD

Para este análisis, se seleccionaron tres SGBD populares: **MySQL**, **PostgreSQL** y **MongoDB**.

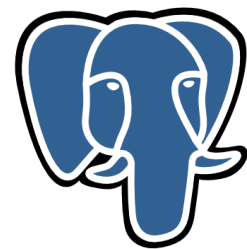
1.1 MySQL

- **Modelo de datos:** Relacional (SQL).
- **Soporte para transacciones:** Cumple con ACID.
- **Escalabilidad:** Escalabilidad vertical, con soporte limitado para distribución horizontal.
- **Seguridad:** Control de acceso basado en roles, autenticación segura y cifrado de datos.
- **Herramientas de análisis y reportes:** MySQL Workbench, integración con BI y herramientas ETL.



1.2 PostgreSQL

- **Modelo de datos:** Relacional (SQL) con soporte para JSONB (semi-estructurado).
- **Soporte para transacciones:** Cumple con ACID y admite transacciones avanzadas.
- **Escalabilidad:** Escalabilidad horizontal con replicación y particionamiento.
- **Seguridad:** Soporte para autenticación LDAP, SSL/TLS y encriptación avanzada.
- **Herramientas de análisis y reportes:** PostgreSQL Studio, soporte para BI.



PostgreSQL

1.3 MongoDB

- **Modelo de datos:** No relacional, basado en documentos JSON.
- **Soporte para transacciones:** Soporte parcial para ACID en versiones recientes.
- **Escalabilidad:** Diseñado para escalabilidad horizontal y distribución de datos.
- **Seguridad:** Control de acceso basado en roles, autenticación y encriptación.
- **Herramientas de análisis y reportes:** MongoDB Compass, integración con BI.



En resumen, cada SGBD tiene fortalezas y debilidades dependiendo del tipo de proyecto:

- **MySQL** es ideal para aplicaciones web y proyectos medianos con estructuras de datos bien definidas.
- **PostgreSQL** es recomendable para sistemas empresariales que requieren robustez y escalabilidad avanzada.
- **MongoDB** es excelente para aplicaciones con datos no estructurados y necesidades de escalabilidad horizontal.

La elección del SGBD depende del tipo de proyecto, el volumen de datos y los requerimientos de escalabilidad y seguridad.

2. Comparación de Características:

Característica	MySQL	PostgreSQL	MongoDB
Modelo de datos	Relacional	Relacional / JSONB	No relacional
Transacciones (ACID)	Sí	Sí (avanzado)	Parcial
Escalabilidad	Vertical	Horizontal con replicación	Horizontal con sharding
Seguridad	Cifrado y roles, autenticación SSL	Cifrado avanzado, autenticación LDAP	Roles, autenticación, encriptación TLS
Facilidad de uso	Alta, interfaz intuitiva	Media, configuración avanzada	Alta, flexible y escalable
Costo	Gratis (comunidad), versiones de pago	Gratis (comunidad), versiones de pago	Gratis (comunidad), versiones de pago
Comunidad y Soporte	Amplia comunidad, documentación extensa	Fuerte comunidad, buen soporte oficial	Creciente comunidad, fuerte respaldo de MongoDB Inc.
Rendimiento	Alto para consultas simples	Alto para transacciones complejas	Alto para grandes volúmenes de datos
Casos de uso recomendados	Aplicaciones web, pequeños y medianos proyectos	Sistemas financieros, bases de datos empresariales	Big data, IoT, aplicaciones en tiempo real

SGBD	Ventajas	Desventajas
MySQL	Fácil de usar, alto rendimiento en consultas simples, amplia documentación y comunidad de soporte.	Limitado en escalabilidad horizontal, menos optimizado para consultas complejas y procesamiento de grandes volúmenes de datos.
PostgreSQL	Potente, soporte para JSON, altamente escalable, robusto para transacciones complejas.	Puede ser más complejo de configurar y administrar, curva de aprendizaje más pronunciada.
MongoDB	Escalabilidad horizontal, almacenamiento flexible para datos no estructurados, ideal para big data.	Menor soporte para ACID, estructura de datos menos rígida que puede generar inconsistencias sin diseño adecuado.

3. Estudio de Casos

El objetivo de este estudio es analizar casos reales donde se haya utilizado cada uno de los sistemas de gestión de bases de datos (SGBD) seleccionados: **MySQL**, **PostgreSQL** y **MongoDB**. Se examinará el problema o necesidad abordada, la razón por la que se eligió el SGBD específico y los resultados obtenidos.

1. Caso de estudio: MySQL en Facebook (Primeros años)

Problema o necesidad

Facebook, en sus primeros años de desarrollo, necesitaba una base de datos confiable y escalable para manejar una creciente cantidad de usuarios y sus interacciones. Se requería una solución eficiente para almacenar y recuperar grandes volúmenes de datos de manera rápida.

Razón para elegir MySQL

- MySQL ofrecía un rendimiento óptimo en la lectura de datos, lo cual era fundamental para mostrar contenido en los perfiles de los usuarios con rapidez.
- Era una solución de código abierto, lo que permitió a Facebook personalizar su funcionamiento según sus necesidades.
- Contaba con una amplia comunidad y documentación que facilitaba su implementación y soporte técnico.

Resultados y beneficios

- Se logró un sistema rápido y eficiente para la gestión de datos de usuarios.
- MySQL permitió una integración con Memcached para mejorar la velocidad de acceso a datos en caché.
- Facebook eventualmente migró partes de su arquitectura a otras soluciones más escalables, pero MySQL fue clave en sus primeras etapas.



2. Caso de estudio: PostgreSQL en Instagram

Problema o necesidad

Instagram, tras su adquisición por Facebook, necesitaba una base de datos robusta y confiable para manejar millones de imágenes y datos de usuarios. Se requería un SGBD con soporte para transacciones complejas y alta disponibilidad.

Razón para elegir PostgreSQL

- PostgreSQL proporcionaba un excelente soporte para integridad de datos y transacciones ACID, lo que garantizaba la consistencia en las operaciones de la plataforma.
- Su capacidad para manejar datos JSONB permitió una flexibilidad en el almacenamiento de metadatos de imágenes y usuarios.
- Contaba con herramientas avanzadas de replicación y escalabilidad, facilitando la gestión de bases de datos distribuidas.

Resultados y beneficios

- Se logró mejorar la estabilidad y rendimiento en el manejo de grandes volúmenes de datos.
- PostgreSQL permitió realizar consultas complejas de manera eficiente, lo cual optimizó la experiencia de usuario.
- Instagram pudo escalar su arquitectura de manera más controlada, manteniendo la integridad de los datos.



3. Caso de estudio: MongoDB en eBay

Problema o necesidad

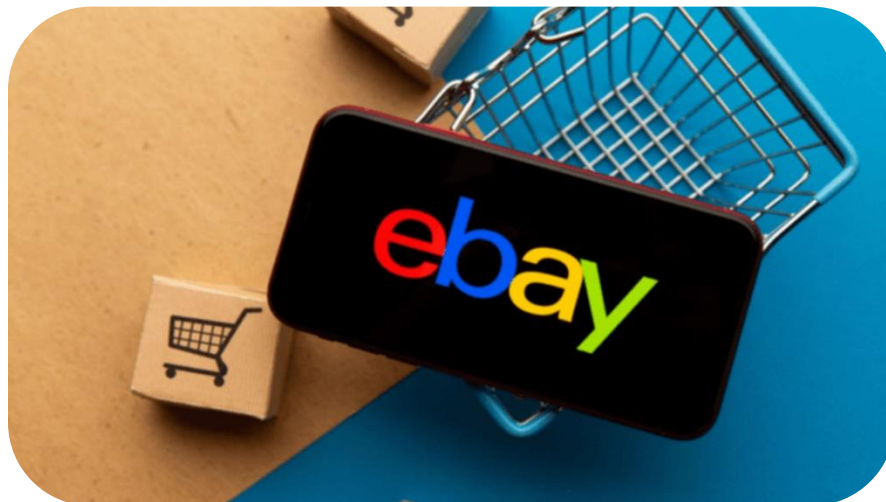
eBay, una de las plataformas de comercio electrónico más grandes del mundo, necesitaba una base de datos escalable y flexible para manejar datos semiestructurados relacionados con productos, usuarios y transacciones.

Razón para elegir MongoDB

- MongoDB permitió un almacenamiento flexible en formato JSON, ideal para gestionar grandes volúmenes de datos de diferentes tipos.
- Su capacidad de escalabilidad horizontal permitió distribuir la carga en múltiples servidores.
- La capacidad de indexación avanzada mejoró la velocidad de búsqueda de productos en la plataforma.

Resultados y beneficios

- Se logró una mejor experiencia de usuario gracias a la rápida recuperación de información.
- eBay pudo manejar de manera eficiente datos no estructurados y realizar cambios en la estructura de la base de datos sin afectar el rendimiento.
- La integración con otras tecnologías permitió mejorar la recomendación de productos y personalizar la experiencia de compra.



En conclusión, cada uno de los SGBD analizados ha demostrado su valor en distintos contextos:

- **MySQL** es una opción sólida para proyectos que requieren rapidez en lectura y facilidad de implementación.
- **PostgreSQL** es ideal para aplicaciones que demandan transacciones complejas y alta integridad de datos.
- **MongoDB** se adapta bien a entornos con grandes volúmenes de datos semiestructurados y necesidades de escalabilidad.

La elección de un SGBD debe considerar factores como la estructura de datos, escalabilidad y necesidades específicas del proyecto.