

MySQL vs MongoDB

by admin - lunes, agosto 16, 2021

<https://dbandtech.com/mysql-vs-mongodb/>

Hoy en día las bases de datos NoSQL o no-relacionales se han puesto de moda, sobretodo porque ofrecen flexibilidad y alta performance en comparación con las bases de datos relacionales. ¿Pero estas bases de datos son mejores que las relacionales? ¿Conviene usarlas siempre o en determinados casos?

En este versus despejamos esas dudas: te mostramos las ventajas y desventajas de MySQL vs MongoDB, y en qué momentos es conveniente usar una u otra.

BASES DE DATOS NOSQL

Las bases de datos no relacionales surgen de la necesidad de tener sistemas de bases de datos altamente disponibles, distribuidos y escalables horizontalmente.

Estos tienen la particularidad de que no poseen un esquema definido sino que cada documento puede tener distintos datos, sin importar cantidad o tipo de datos.

Positivo: si necesitas un campo más en un registro en específico no es necesario remodelar toda la tabla solo por un par.

Negativo: la salida de datos será variable, por lo que si necesitas una salida más estructurada no es conveniente el uso de bases de datos NoSQL.

Al ser no relacionales no poseen relaciones, por lo que nos podemos olvidar de esos engorrosos joins que enlentecen nuestras consultas. Si hay maneras de agregar relaciones en el caso de que sean necesarias pero la idea en este tipo de base de datos es tener las menos posibles.

Positivo: Mejor performance, consultas más rápidas. Suelen ser pocas colecciones y cada una tiene un propósito.

Negativo: cuando editamos datos hay que hacerlo en todas las colecciones en las que se encuentran.

Las bases de datos no relacionales permiten el escalado horizontal de forma sencilla y poseen una mejor performance a nivel de miles de consultas de lectura y escritura.

BASES DE DATOS SQL

Los datos SQL se manejan con un esquema bien definido y rígido, por lo que todos los datos de una misma tabla tienen la misma estructura cantidad de campos y tipo de dato.

Positivo: nos permite manejar mejor la salida de datos y tener una estructura compleja mejor organizada.

Negativo: agregar nuevos campos requiere modificar la tabla y junto con ella todos los registros

existentes en la misma. La mayoría de las veces la solución no cubre las necesidades del proyecto y es necesario editar el esquema repetidamente.

Permite relaciones entre tablas añadiendo el identificador de una tabla como clave foránea en la otra tabla o generando una nueva tabla para guardar la relación entre ambas tablas.

Positivo: este tipo de relaciones suelen ser más robusta.

Negativo: las relaciones enlentecen las tablas ya que consultas que impliquen los datos de varias tablas deben realizarse mediante joins.

Las bases de datos relacionales permiten el escalado de manera vertical en un mismo servidor, se pueden escalar de manera horizontal pero significa una gran cantidad de horas de trabajo. Tienen mejor desempeño en el manejo de datos para uso estadístico y analítico.

	SQL	NoSQL
Datos	Datos estructurados guardados en tablas	Datos no estructurados guardados en formato BSON – binary JSON
Esquema	Estático	Dinámico
Escalabilidad	Vertical	Horizontal
Transacciones	Soporta transacciones ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad)	Teorema CAP (Consistencia, Disponibilidad y Tolerancia al particionado)
Auto Elasticidad	Requiere downtimes en algunos casos	Automático, no requiere interrupción

MYSQL VS MONGODB: PRINCIPALES DIFERENCIAS

Ahora que ya hemos visto las principales diferencias entre una base de datos relacional y una no relacional nos podemos adentrar en este versus de MySQL y MongoDB. Aún quedan algunas diferencias propias de estas tecnologías por comparar, que nos ayudarán a decidir cuándo es mejor usar una u otra.

Tabla comparativa, en esta tabla se destacan algunas de las diferencias entre ambos

Característica	MongoDB	MySQL
Cloud, SaaS, Web	si	si
Desarrolladores	MongoDB Inc.	Oracle Corporation
SO	Multiplataforma	Multiplataforma
Lenguaje query	Javascript	SQL
Mapa reducido	si	no
Convercion de DB	si	no
Analisis de performance	si	no
Virtualización	si	no
Modelo de integridad	BASE	ACID
Atomicidad	condicional	si
Aislamiento	no	si
Transacciones	no	si
Integridad referencial	no	si
CAP	CP	CA
Escalabilidad horizontal	si	condicional
Modo de replicación	Maestro-Esclavo	Maestro - Maestro/Esclavo

CARACTERÍSTICAS DE MONGODB

Los datos guardados con MongoDB son de fácil lectura para los usuarios, como ya mencionamos se guarda en formato JSON por lo cual a modo de ejemplo quedan de la siguiente manera:

```
{
  A/c number: 1234567891
  Nombre: "Juan"
  Apellido: "Perez"
  Telefono: "(+1)45878931"
  Direccion: [ {
    Calle: "calle 20"
    Ciudad: "Montevideo"
    CodigoPostal: 20001
  } ]
}
```

Nótese que en los documentos **en lugar de crear relaciones con otra tabla se puede especificar de forma anidada más datos**, como es el caso de la dirección en el ejemplo. Los documentos nos

permiten almacenar toda la información que queramos, lo cual a veces puede ocasionar problemas en la consistencia de los datos.

MongoDB utiliza un **lenguaje de consultas no estructurado** por lo que realizamos las consultas especificando el nombre del documento con las propiedades que queremos filtrar. Este tipo de consultas permite una amplia variedad de operadores, los puedes consultar desde su [sitio oficial](#).

Como vemos en la tabla comparativa MongoDB en el teorema CAP se inclina a **CP (Consistencia y Tolerancia a particiones)** esto significa que todos los clientes acceden a una vista consistente de la base de datos. Lo cual implica que los usuarios de un nodo deben esperar a que los otros nodos se sincronicen para poder ser visibles y editables, en este caso la disponibilidad queda en segundo plano frente a la consistencia.

En el ámbito de la seguridad MongoDB utiliza un **control de acceso basado en roles con privilegios flexibles**, sus características de seguridad incluyen **autenticación, auditoría y autorización**. También **permite el uso de TLS/SSL** con el propósito de encriptar los datos y que solo sean accesibles para el cliente.

VENTAJAS

- Validación de documentos
- Motores de almacenamiento integrado
- Menor tiempo de recuperación ante fallas

DESVENTAJAS

- No es una solución adecuada para aplicaciones con transacciones complejas
- No tiene un reemplazo para las soluciones de herencia
- Aún es una tecnología joven

CARACTERÍSTICAS DE MYSQL

MySQL ya tiene varios años en la vuelta por lo que tiene **una comunidad enorme y más sólida**.

Como todos sabemos a la hora de comenzar a trabajar **con MySQL es necesario crear nuestro quema con las tablas y campos que vamos a necesitar** para guardar nuestros datos. Si eres principiante el esquema se vuelve una prueba y error donde debemos cambiar el modelo cada vez que debemos guardar un dato que no fue contemplado, incluso los expertos muchas veces deben cambiar el esquema sobre la marcha.

En MySQL **las consultas se manejan con un lenguaje estructurado como lo es SQL**, por lo general suele ser bastante simple una vez le agarramos la mano, el mismo implica 2 partes un lenguaje de definición de datos (DDL) y un lenguaje de manipulación de datos (DML).

En la tabla comparativa vemos que MYSQL bajo el teorema CAP se inclina por **CA (Consistencia y**

disponibilidad), esto significa que **la información será consistente entre todos los nodos mientras los mismos estén disponibles**. Esto nos permite leer/escribir desde cualquier nodo y estar seguros de que los datos serán consistentes. Si se realiza una partición entre nodos los datos no estarán sincronizados y no se resolverá hasta que se resuelva la partición.

MySQL utiliza un **modelo de seguridad basado en privilegios**, a cada usuario se le asigna privilegios sobre la base de datos de tipo CREATE, SELECT, INSERT, UPDATE entre otros. MySQL también utiliza **encriptación para la conexión entre el server y el cliente del tipo SSL**.

VENTAJAS

- Soporta transacciones atómicas
- Soporta el uso de JOIN
- Seguridad basada en privilegios con uso de contraseña
- Es una tecnología madura

DESVENTAJAS

- Difícil de escalar
- Problemas de estabilidad
- No es un desarrollo impulsado por la comunidad

PERFORMANCE Y VELOCIDAD

Los datos de la gráfica fueron tomados en base a 1.000.000 registros con MySQL 5.7.9 y MongoDB 3.2.0 utilizando las configuraciones por defecto en un servidor Ubuntu con 8 CPUs virtuales y 32 GB de ram en entornos separados.



MongoDB es más rápido que MySQL gracias a su capacidad para manejar grandes cantidades de datos no

estructurados, permitiendo realizar consultas de manera sensible al workload (carga de trabajo). Mientras que MySQL suele ser más lento al momento de manejar grandes bases de datos.

Si vamos a juzgar qué base de datos debemos utilizar dependiendo la performance pues esto dependerá de qué tan grande sea y del volumen de datos que vayamos a manejar. Cuanto más grande sea nuestra base de datos y manejo de grandes volúmenes de información más nos convendrá utilizar una base de datos no relacional como mongoDB. En cambio sí sabemos que nuestra base de datos no crecerá mucho y los datos insertados tendrán siempre el mismo formato, pues nos conviene más MySQL.

¿CUANDO ELEGIR UNA U OTRA?

MongoDB	MySQL
<p>ALTA DISPONIBILIDAD Ideal si necesitas una alta disponibilidad de los datos con recuperacion automatica, rapida e instantanea</p> <p>FRAGMENTACIÓN Si piensas que creceras mucho en el futuro te conviene MongoDB ya que tienen incorporada la fragmentación</p> <p>ESQUEMA INESTABLE Ideal si tienes un esquema inestable y cambiante para evitar cambiar el esquema repetidamente</p> <p>SIN ADMINISTRADOR DE BD Si no quieres usar un administrador de base de datos. Si crecees demasiado deberas utilizar uno de todas maneras.</p> <p>EN LA NUBE Si todos tus servicios estan basados en la nube. MongoDB te resultara mas conveniente.</p>	<p>BAJO MANTENIMIENTO Si recién comienzas y no crees que creceras muchos MySQL te ayudara con su facil y bajo manteniimiento.</p> <p>PRESUPUESTO LIMITADO Ideal si necesitas alta performance con un presupuesto limitado.</p> <p>ESQUEMA FIJO Ideal si tu esquema no cambia y siempre guardas los mismos tipos de datos. Por ejemplo un blog.</p> <p>TRANSACCIONES ALTAS Si requieres de una alta tasa de transacciones</p> <p>SEGURIDAD DE LOS DATOS Si la seguridad de los datos es tu priridad, MySql es el manejador de bases de datos más seguro.</p>

Cuando elegir una u otra siempre va a depender de las necesidades y características de tu proyecto, ninguna es mejor que la otra simplemente son diferentes y funcionan mejor para diferentes casos.

MySQL ofrece alto rendimiento, flexibilidad, protección confiable de datos, alta disponibilidad y es fácil de administrar. Si se indexa correctamente puede mejorar notoriamente el rendimiento, facilitar las consultas y mejorar la robustez. Cabe destacar que es una mejor opción si vamos a realizar transacciones complejas.

Por otro lado MongoDB como ya hemos mencionado es una mejor opción si no tienes un esquema definido o manejas datos no estructurados. Si manejas grandes volúmenes de datos y bases de datos muy grandes MongoDB te ofrecerá un mejor rendimiento, por otra parte si necesitas almacenar muchos datos como documentos/archivos MongoDB es una mejor opción que te ayudara muchísimo.

En tu caso... ¿cuál prefieres para tus proyectos?

[Fuente.](#)