DWES 02

Gestión de bases de datos en PHP

1. No pienses en una aplicación web ni en una aplicación de escritorio concreta y cita al menos tres tipos de sistemas que podrían utilizarse para organizar la información del almacén. Estos tipos de sistemas no tienen porqué ser necesariamente una base de datos relacional (existen otras formas de almacenar la información en un ordenador), aunque una base de datos relacional es uno de los tipos de sistemas de almacenamiento de la información que deberías contemplar. En este apartado no se espera que indiques bases de datos concretas (como MySQL, MongoDB, etc.) sino tipos de sistemas de almacenamiento en general, aunque puedes proporcionar un ejemplo de cada tipo (por ejemplo, MySQL es un tipo de base de datos relacional).

Base de datos relacional, que consisten en una serie de contenedores de información denominados tablas, en los que se almacenan registros. Los registros serían las filas de las tablas mientras que sus campos de éstos serían las columnas, es la más común en la actualidad.

Base de datos distribuida, la información no reside en un único servidor de bases de datos, sino que existen diversos servidores que mantienen una porción de datos.

Base de datos orientada a objetos, la información se almacena mediante objetos. Los objetos son conjuntos heterogéneos de datos, tan complejos como sea necesario para modelizar aquella información que se necesite.

Base de datos NoSQL, son aquellas en las que no se usan tablas sino colecciones de elementos. Los elementos almacenados en las colecciones pueden ser heterogéneos, de modo que en una colección podemos almacenar registros con juegos de datos distintos entre sí. Además, este tipo de base de datos no suele usar SQL para realizar consultas, sino programación funcional para hacer filtrados y otros tipos de operaciones.
Las NoSQL son otro modelo de bases de datos muy popular, sin llegar al grado de las relacionales, pero muy frecuente en la actualidad en muchos tipos de aplicaciones donde se requiere variabilidad entre la información almacenada en las colecciones así como velocidad de recuperación de la información.

2. Partiendo del ejercicio anterior, reflexiona sobre cuales son los tipos sistemas más convenientes de cara a desarrollar una aplicación web desde PHP.

Bases de datos relacionales y Bases de datos NoSQL ya que proporcionan un gran rendimiento. Como ya he comentado las bases de datos relacionales son las más extendidas y dentro de estas MySQL o MariaDB.

3. Si usamos una base de datos MySQL o MariaDB, ¿con qué tipo de sistema de almacenamiento de información de los citados por ti en el ejercicio anterior encajaría?

Base de datos relacional

4. Desde PHP se puede acceder a una base de datos MySQL o MariaDB a través de diferentes formas. Cita al menos dos formas diferentes y explica sus diferencias.

MySQLi y PDO.

MySQLi es una extensión de PHP creada con el propósito de facilitar la comunicación entre un script PHP y un servidor MySQL mientras que PDO es una extensión más abstracta que podríamos utilizar con distintas bases de datos (no solo MySQL).

// PDO (indicamos el parámetro mysql para indicar el tipo de base de datos al que nos vamos a conectar)

```
$pdo = new PDO("mysql:host=localhost;dbname=database", 'username', 'password');
```

// mysqli, mediante programación estructurada

```
$mysqli= mysqli_connect('localhost','username','password','database');
```

// mysqli, mediante POO

```
$mysqli = new mysqli('localhost','username','password','database');
```

En MySQLi no es necesario indicar el tipo de base de datos ya que sólo se puede conectar a bases de datos MySQL.

5. Investiga que diferencias hay entre MySQL y Apache Cassandra, y reflexiona sobre que limitaciones o ventajas tendría el uso de una u otra de cara a desarrollar la aplicación web para esta empresa.

¿Qué es Cassandra?

Apache Cassandra es un sistema de base de datos NoSQL distribuido y altamente escalable,diseñado para administrar grandes cantidades de datos en muchos servidores sin un solo punto de falla.

Es ideal para manejar datos de alta velocidad y gran escala, proporcionando alta disponibilidad, tolerancia a fallas y escalabilidad horizontal.

Cassandra se utiliza a menudo en aplicaciones que requieren análisis de datos en tiempo real, IoT y manejo de grandes conjuntos de datos en un entorno distribuido, que tiene una arquitectura peer-to-peer y soporte para esquemas flexibles.

¿Qué es MySQL?

MySQL es un sistema de gestión de bases de datos relacionales (RDBMS) de código abierto que utiliza el lenguaje de consulta estructurado (SQL) para gestionar y consultar bases de datos .

Se utiliza ampliamente para aplicaciones web y soluciones de nivel empresarial debido a su rendimiento, confiabilidad y facilidad de uso.

MySQL es conocido por su escalabilidad, ya que admite bases de datos de gran escala y múltiples usuarios simultáneamente y, a menudo, se implementa junto con servidores web como Apache en una pila LAMP (Linux , Apache, MySQL, PHP).

Diferencia entre Cassandra y MySQL

Casandra MySQL

1. Desarrollado por Apache Software Foundation y lanzado en julio de 2008.
   Desarrollado por Oracle y lanzado en mayo de 1995.

2. Cassandra está escrito únicamente en lenguaje Java.
   MySQL está escrito en lenguajes C y C++ .

3. Cassandra es una base de datos de tipo NoSQL.
   MySQL es una base de datos de tipo RDBMS.

4. No proporciona propiedades ACID, pero se puede ajustar para admitirlas.
   MySQL proporciona propiedades ACID .

5. El rendimiento de lectura es muy eficiente en Cassandra y toma un tiempo O(1).
   MySQL requiere leer desde múltiples tablas usando JOIN, lo que toma un tiempo O(log(n)).

6. El rendimiento de escritura en Cassandra también es muy alto y eficiente.
   Escribir en MySQL requiere una búsqueda previa, lo que ralentiza el rendimiento de escritura.

7. Cassandra no proporciona el concepto de integridad referencial (sin claves externas).
   MySQL proporciona integridad referencial y admite claves externas.

8. Cassandra proporciona métodos de consistencia eventual y consistencia inmediata.
   MySQL solo proporciona consistencia inmediata.

9. Sistema operativo de servidor para Cassandra: BSD, Linux, OS X, Windows.
   Sistema operativo de servidor para MySQL: FreeBSD, Linux, OS X, Solaris, Windows.

10. Empresas famosas que utilizan Cassandra: Hulu, Instagram, Netflix, Reddit, etc.
    Empresas famosas que utilizan MySQL: Airbnb, Pinterest, Slack, Udemy, Twitter, etc.

Cuándo utilizar Cassandra

1. Escalabilidad y arquitectura distribuida

Cassandra está diseñado para sistemas distribuidos a gran escala que necesitan manejar cantidades masivas de datos en múltiples nodos y ubicaciones geográficas.
Se destaca en entornos que requieren escalamiento horizontal (agregar más máquinas para distribuir la carga).

2. Alto rendimiento de escritura

Si su aplicación exige escrituras de alta velocidad y maneja grandes volúmenes de datos, Cassandra es ideal.
Está optimizado para operaciones de escritura intensiva, lo que lo hace adecuado para sistemas de registro, aplicaciones de IoT y análisis en tiempo real.

3. Tolerancia a fallos y disponibilidad

Cassandra está diseñado para alta disponibilidad y tolerancia a fallas.
Asegura que no haya ningún punto de falla, lo que significa que su aplicación puede seguir funcionando sin problemas incluso si algunos nodos fallan.

Cuándo utilizar MySQL

1. Datos estructurados con relaciones

MySQL es una base de datos relacional ideal para aplicaciones que requieren datos estructurados y relaciones complejas entre entidades.
Si necesita imponer integridad referencial mediante claves externas o realizar consultas complejas que involucran múltiples tablas con JOIN , MySQL es la mejor opción.

2. Cumplimiento de ACID

MySQL proporciona propiedades ACID completas (atomicidad, consistencia, aislamiento, durabilidad) , lo que garantiza transacciones confiables y consistentes.
Esto hace que MySQL sea un fuerte candidato para sistemas que requieren integridad transaccional estricta, como aplicaciones financieras, sistemas de comercio electrónico y procesamiento de pedidos.

3. Consistencia inmediata

Si su aplicación necesita una fuerte consistencia inmediata (todas las lecturas deben reflejar las últimas escrituras).
MySQL es más adecuado, ya que garantiza la consistencia inmediata por diseño.
Conclusión
La elección entre Cassandra y MySQL depende en gran medida de los requisitos de arquitectura, escalabilidad y consistencia de su aplicación. Cassandra es la opción preferida para aplicaciones distribuidas, de gran volumen y con gran cantidad de escrituras, que necesitan tolerancia a fallas y alta disponibilidad. MySQL , por otro lado, es más adecuado para aplicaciones que necesitan administración de datos estructurados, una sólida integridad transaccional y consistencia inmediata.