

1. Cuando conviene mover lógica de PHP a Stored Procedures y cuando NO?

Conviene usar store procedure cuando queremos trabajar datos masivos de información y evitar la interacción de ida y vuelta entre el backend y la bd.

No conviene usar cuando la consulta requiere demasiada complejidad, también aplicar demasiada complejidad dificulta el poder debuggear el store procedure.

2. Diferencia entre vista normal y vista materializada. ¿Cómo simular una materializada en MySQL?

Vista normal se consulta cada vez que llamamos al query lo que genera más consumo de memoria, una vista materializada se almacena en una tabla pero tenemos evitando la sobrecarga de memoria pero tendremos que actualizar manualmente si realizamos algún cambio.

3. Explica cómo evitar condiciones de carrera (race conditions) dentro de un SP.

Podemos usar una transacción para aislar la ejecución hasta que este finalice.

4. Cómo implementar idempotencia en procedimientos que generan documentos fiscales?

Podemos usar una llave única cuando realicemos el procedimiento así si esta se envia n veces solo se ejecutara una sola vez

5. Cómo optimizar un RANK() o ranking en MySQL sin afectar rendimiento masivo?

Podemos usar índices compuestos a las columnas que aplicaremos el RANK en mysql.

6. Estrategia para evitar full table scans en reportes académicos de 10k estudiantes.

Podemos usar índices para optimizar la consulta en los inner y un **between** para tener un filtro por rangos de fecha

7. Manejo correcto de NULL vs " en columnas INT y DECIMAL.

Null se usaría para columnas donde no estemos seguro de la información que recibiremos, en columnas **int/decimal** podemos usar valores predeterminados cuando estos no reciban valores.

8. Cómo estructurar auditoría SQL robusta para facturación.

Podemos usar triggers para registrar los cambios en una tabla y almacenarlos en una tabla de auditoría que nos permitirá un mejor seguimiento de los registros.

9. Cómo manejar collation conflicts en un sistema multi base de datos.

Lo recomendable sería estandarizar todas las bases de datos, podemos aplicar soluciones de collation para cada query.

10. Estrategia para versionar procedimientos en producción sin romper el sistema.

Podemos usar un wrapper para tener una sola sp y dentro podemos especificar que sp usara dependiendo del ambiente(test,prod) que estamos desarrollando.