



ÍNDICES Y OPTIMIZACIÓN DE CONSULTAS

Diseño y Programación de una BBDD

ÍNDICE

ÍNDICES Y OPTIMIZACIÓN DE CONSULTAS

1. Buenas prácticas en consultas	3
2. Sentencias EXPLAIN y OPTIMIZE TABLE	3
3. Creación de índices.....	4
4. Prepared Statements.....	4
5. Análisis productos previos	5
6. Roles.....	5

1. Buenas prácticas en consultas

Del recurso recomendado **Effective MySQL Optimizing SQL Statements**, se pueden extraer las siguientes buenas prácticas al realizar consultas. Explicar el porqué de las mismas. (Dar un ejemplo)

1. Sustituir los operadores **OR** en lo posible en las cláusulas WHERE y HAVING por el operador IN
2. Realizar búsquedas binarias sobre Índices FULLTEXT para sustituir caracteres comodín.
3. Tener un equilibrio en el número de índices (Índices faltantes o indexación excesiva)
4. Preparar VISTAS o utilizar tablas temporales para sustituir demasiados JOINS en una consulta.
5. Todas las tablas deben tener una clave primaria y según el tipo de búsquedas que se hagan un índice agrupado.

2. Sentencias EXPLAIN y OPTIMIZE TABLE

La instrucción EXPLAIN permite ver cómo se ejecutará una consulta en nuestra Base de Datos. La tabla que devuelve la sentencia EXPLAIN contiene unos campos que son de vital importancia para entender el funcionamiento de esta consulta. Por otra parte, OPTIMIZE TABLE es una instrucción que debe usarse si se ha eliminado gran parte de una tabla o se han hecho muchos cambios en una tabla con filas de tamaño variable para desfragmentarla y recuperar el espacio no usado. Responder las siguientes interrogantes. (Dar un ejemplo)

6. Explicar cómo se utiliza la sentencia EXPLAIN.
7. ¿Para qué se usan las columnas key y type y cuáles son los posibles valores de esta última columna??
8. ¿Cómo se interpretan los valores de la columna **type** y qué acciones podrían tomarse para optimizar la consulta?
9. ¿Cuál es la sintaxis de OPTIMIZE TABLE?

3. Creación de índices

Un índice es estructura que permite la optimización del rendimiento de una Base de Datos al facilitar la recuperación eficiente de los datos minimizando la cantidad de accesos al disco necesarios al consultar los registros. A este respecto explicar cada una de las siguientes preguntas. (Dar un ejemplo de cada una de ellas)

10. ¿Qué sentencia SQL permite la creación de Índices? Indicar la sintaxis.
11. ¿Con cuál sentencia se elimina un índice?
12. ¿Para qué se utilizan los diferentes tipos de índice en MySQL (PRIMARY | UNIQUE | FULLTEXT | SPATIAL)?
13. Explica las características de los tipos de estructuras en las cuales se pueden crear índices (B-TREE y HASH) y cuándo es mejor usar una u otra.
14. ¿Cómo se crea un índice sobre varios campos y cuál sería su utilidad?
15. MySQL ofrece búsqueda FULLTEXT.
Esto permite buscar en las tablas aquellas columnas que contengan textos que ofrezcan las mejores coincidencias de palabras y frases respecto de una búsqueda.
¿Qué condiciones deben darse para poder utilizar este tipo de búsquedas?
16. ¿Qué es una búsqueda BOOLEAN FULLTEXT y por qué es mucho más rápida que utilizar el operador LIKE?
17. ¿Es posible realizar búsquedas FULLTEXT contra varias columnas de una tabla? (Ejemplo)

4. Prepared Statements

Un **Prepared Statement** es un mecanismo que permite que la ejecución de ciertas sentencias SQL que se ejecutan repetidas en el tiempo sean más rápidas. El mecanismo que establecen las Bases de Datos para hacer más eficiente el proceso, es que le indiquemos previamente el tipo de sentencia que vamos a usar, de forma que la base de datos la "precompila" y la guarda en condiciones de ser ejecutada inmediatamente, sin necesidad de analizarla en cada caso. Explica (dando un ejemplo para cada respuesta.)

18. ¿Cuáles son los pasos del proceso para crear un Prepared Statement en MySQL?
19. ¿Qué otros procesos se pueden rescatar de la lectura de los recursos recomendados?

5. Análisis productos previos

Revisad los productos 2 y 3 aplicando los conocimientos adquiridos en este producto.

20. ¿Cómo se podrían optimizar las consultas o qué consultas toman en cuenta los puntos de este producto y por qué?

6. Roles

En relación a los permisos de usuario, MySQL en su versión 8, permite hacer uso de los Roles, los cuales permiten simplificar la administración de usuarios en la Base de Datos. A este respecto, explica:

21. ¿Qué es un rol y cómo se crea un rol?

22. ¿Cómo se asignan usuarios a un rol?

23. ¿Cómo retirar usuarios de un rol?

24. ¿Cómo eliminar un rol y qué se requiere antes de hacerlo?

25. ¿Qué roles se podrían crear para el Gimnasio? (Justificar)