Optimización en MySQL

Bases de datos II

3º - Grado en Ingeniería Informática

Índice de Contenidos

[Introducción 3](#_Toc5840603)

[Optimizar a nivel de base de datos 3](#_Toc5840604)

[Optimizar a nivel de hardware 3](#_Toc5840605)

[Procesos de optimización implementados por MySQL 3](#_Toc5840606)

[Sentencia EXPLAIN 3](#_Toc5840607)

[Motores de almacenamiento MyISAM e InnoDB 3](#_Toc5840608)

[Índices 3](#_Toc5840609)

[Sentencia OPTIMIZE Table 4](#_Toc5840610)

[Sentencia INSERT DELAYED 4](#_Toc5840611)

[Slow Query Log 4](#_Toc5840612)

[Variables de sistema: long-query-time 4](#_Toc5840613)

[Show ProcessList 4](#_Toc5840614)

[Conclusiones 4](#_Toc5840615)

[Bibliografía 4](#_Toc5840616)

# Introducción

El rendimiento de una base de datos se basas en diversos factores, como las tablas, sentencias y la configuración que utiliza, siendo estos factores a nivel de la propia base de datos. Además, se requiere de una optimización a nivel de hardware, minimizando y haciendo lo más eficiente posible las operaciones de la CPU y las de entrada y salida.

## Optimizar a nivel de base de datos

Las tablas deben estar estructuradas correctamente: que las columnas tengan el tipo de dato adecuado y que cada tabla tenga un número apropiado de columnas.

Uso correcto del indexado, lo que se abordará posteriormente en detalle.

Para favorecer el rendimiento y la escalabilidad se debería usar el motor de almacenamiento más adecuado para cada tabla, siendo en ocasiones mejor el uso de motores transaccionales y, en otros casos, los motores no transaccionales.

## Optimizar a nivel de hardware

A medida que la base de datos se usa con mayor asiduidad, esta podría alcanzar su límite hardware, siendo necesario modificar la aplicación de la base de datos o reconfigurar el servidor de llegar a este punto. Los cuellos de botella de los sistemas suelen ser: las búsquedas en disco, la lectura y escritura en disco, los ciclos del procesador o, aunque en menor medida, el ancho de banda de la memoria.

# Procesos de optimización implementados por MySQL

## Sentencia EXPLAIN

La sentencia EXPLAIN proporciona información sobre como MySQL ejecuta sentencias, en concreto las sentencias SELECT, DELETE, INSERT, REPLACE y UPDATE.

Cuando se utiliza EXPLAIN con una sentencia explicable, MySQL muestra información desde el optimizador explicando cómo se procesaría la sentencia, incluyendo información sobre las como se unen las tablas y en que orden.

Con la ayuda de EXPLAIN se puede ver donde se deberían añadir índices en las tablas de forma que la ejecución de la consulta se vea acelerada al utilizar índices para encontrar filas. También es útil para comprobar si el optimizador está uniendo tablas en el orden óptimo.

EXPLAIN retorna una fila de información por cada tabla utilizada en la sentencia SELECT.

## Motores de almacenamiento MyISAM e InnoDB

## Índices

La creación de índices en una o varias de las columnas objetivo de una consulta es la mejor manera de mejorar el rendimiento de las operaciones SELECT. Estos índices actúan como puntero a las filas de la tabla, permitiendo a la consulta determinar rápidamente que filas cumplen cierta condición en la cláusula WHERE y la recuperación del resto de información de dichas filas. Tampoco es bueno crear índices sin control en todas las columnas de una consulta ya hacen a MySQL gastar espacio y tiempo en determinar que índices se utilizarán.

## Sentencia OPTIMIZE Table

Once your data reaches a stable size, or a growing table has increased by tens or some hundreds of megabytes, consider using the OPTIMIZE TABLE statement to reorganize the table and compact any wasted space. The reorganized tables require less disk I/O to perform full table scans. This is a straightforward technique that can improve performance when other techniques such as improving index usage or tuning application code are not practical.

OPTIMIZE TABLE copies the data part of the table and rebuilds the indexes. The benefits come from improved packing of data within indexes, and reduced fragmentation within the tablespaces and on disk. The benefits vary depending on the data in each table. You may find that there are significant gains for some and not for others, or that the gains decrease over time until you next optimize the table. This operation can be slow if the table is large or if the indexes being rebuilt do not fit into the buffer pool. The first run after adding a lot of data to a table is often much slower than later runs.

## Sentencia INSERT DELAYED

A la hora de optimizar sentencias INSERT se debería fusionar múltiples operaciones pequeñas en una sola; realizar el menor número de conexiones con la tabla donde se van a insertar las nuevas tuplas y mandar el mayor de estas de una sola vez, retrasando la actualización de índices y la comprobación de consistencia.

Para mejorar la velocidad de las sentencias INSERT hay que usar sentencias INSERT con múltiples listas de VALUE cuando se desee insertar varias tuplas a la vez, usar la sentencia LOAD DATA cuando se cargue una tabla de un archivo de texto o insertar valores explícitamente solo cuando el valor a insertar es distinto del valor por defecto de la columna.

La opción DELAYED es una extensión de MySQL al estándar SQL, que se encuentra actualmente obsoleta desde MySQL 5.6. Anteriormente, cuando un cliente usaba esta sentencia sobre una tabla, la tupla se almacenaba en una cola para insertarse cuando esta tabla no estuviera en uso.

En MySQL 8.0 el servidor reconoce la opción DELAYED, pero la ignora y trata la sentencia como un simple INSERT. Se planea eliminar esta palabra reservada en futuras versiones.

## Slow Query Log

## Variables de sistema: long-query-time

## Show ProcessList

# Conclusiones

# Bibliografía

manual de referencia de MySQL - <https://dev.mysql.com/doc/>