

Unidad Didáctica 16 Optimización en MySQL

Optimización del diseño de Bases de Datos

Es importante hacer un diseño óptimo de la base de datos, de forma que sea más fácil construir aplicaciones óptimas y mejorar su rendimiento.

Para ello se deben crear con la ideal general de minimizar el espacio que ocupan en diseño de modo que se reduzca el flujo de entrada salida en disco.

En este sentido se puede tener en cuenta lo siguiente:

Usar los tipos de datos menores posibles siempre que se ajusten a los requisitos. Por ejemplo un MEDIUMINT ocupa un 25% menos que un INT:

Siempre que sea posible usaremos NOT NULL en la definición de los campos, ya que facilita el uso de índices y evita la comprobación en consultas en que se comprueba si cierto valor es nulo.

Comandos de mantenimiento de Tablas

MySQL dispone de una serie de comandos que facilitan la optimización de sentencias SQL.

1ª) ANALYZE TABLE

Este comando analiza y almacena la distribución de claves de una tabla. Funciona con tablas MyISAM, BDB e InnoDB. Es útil cuando se ha cargado una tabla con gran cantidad de datos, para actualizar el índice.

Sintaxis:

```
ANALYZE [NO_WRITE_TO_BINLOG | LOCAL] TABLE nbTabla [, nbTabla]....
```

MySQL usa la distribución de claves almacenadas para decidir el orden en que se harán las consultas que usen combinaciones (join) de tablas. También las usa para decidir que índices tiene que utilizar para una tabla específica dentro de la consulta.

El comando devuelve el nombre de la tabla el tipo de mensaje (error, info, nota o warning) y el mensaje informativo.

ANALYZE TABLE clientes

Se puede comprobar la distribución de claves almacenada con el comando SHOW INDEX. Si la tabla no ha cambiado desde el último comando ANALYZE TABLE, la tabla no vuelve a analizarse.

De forma predeterminada, ANALYZE TABLE se escribe en el log binario para que se repliquen en los esclavos de replicación. Esta acción se puede suprimir con la palabra clave NO_WRITE_TO_BINLOG opcional o su alias LOCAL.

2ª) REPAIR TABLE

Solo se utiliza en tablas MyISAM y ARCHIVE. Su función es reparar tablas corruptas (con datos y/o índices erróneos). Normalmente no debe usarse salvo que ocurra algún error grave en el sistema.

Sintaxis:

```
REPAIR [NO_WRITE_TO_BINLOG | LOCAL] TABLE nbTabla [, nbTabla]....  
[QUICK] [EXTENDED] [USE_FRM]
```

Donde las cláusulas son las mismas que ANALYZE salvo:

- **QUICK**, se utiliza para reparar únicamente los archivos de índices.
- **EXTENDEND**, para crear el índice fila a fila en lugar de crearlo de golpe, que es menos seguro.
- **USE_FRM**, cuando el fichero MVI de índices está ausente o corrupto, para lo cual se basa en el fichero .frm correspondiente.

Su ejecución devuelve el nombre de la tabla procesada, la operación realizada, el tipo de mensaje y un mensaje descriptivo del estado de la tabla.

REPAIR TABLE clientes

3ª) OPTIMIZE TABLE

Solo para tablas MyIsam, InnoDB y ARCHIVE se usa si hemos hecho gran cantidad de borrados o modificaciones sobre tablas con tipos de variables VARCHAR, VARBINARY, BLOB o TEXT. De este modo se reutiliza el espacio y desfragmenta la tabla optimizando el espacio ocupado por los datos.

Sintaxis:

```
OPTIMIZE [NO_WRITE_TO_BINLOG | LOCAL] TABLE nbTabla [, nbTabla]....
```

Su ejecución devuelve el nombre de la tabla procesada, la operación realizada, el tipo de mensaje y un mensaje descriptivo del estado de la tabla.

OPTIMIZE TABLE clientes

4ª) CHECK TABLE

Revisa tablas y vistas MyISAM, InnoDB y ARCHIVE.

Sintaxis:

```
CHECK TABLE nbTabla [, nbTabla]...[option]
```

```
Option = [FOR UPGRADE | QUICK | FAST | MEDIUM | EXTENDED | CHANGED].
```

Donde el valor **option** determina incompatibilidades de versión (FOR UPGRADE) o el grado (creciente) de profundidad en la revisión de la tabla.

La salida es similar a REPAIR salvo que la operación es check. Si el mensaje obtenido no es **ok** o **Table up to date** es conveniente usar REPAIR para corregir los posibles errores.

CHECK TABLE clientes

5ª) CHECKSUM TABLE

Genera una suma de verificaciones de la tabla. Cualquier cambio que hagamos a la misma producirá una suma completamente distinta. De este modo podremos saber si ha sido alterada de manera fortuita o sin que nos demos cuenta.

Sintaxis:

```
CHECKSUM TABLE nbTabla [, nbTabla]...[QUICK] [EXTENDED]
```

La cláusula QUICK permite hacer esta suma mucho más rápida aunque debe estar activada con la opción **CHECKSUM = 1** cuando creamos la tabla.

CHECKSUM TABLE clients

Comandos SQL de optimización y Seguridad	
EXPLAIN	Permite saber cómo se va a ejecutar cierta consulta.
REPAIR	Repara tablas MyISAM corruptas.
CHECK TABLE	Chequea una tabla para encontrar errores. Sirve para tablas MyISAM e InnoDB.
ANALYZE	Analiza la distribución de claves para una tabla. Sirve para tablas MyISAM e InnoDB.
OPTIMIZE	Cuando se han hecho muchas modificaciones a una tabla actualizando estadísticas y reordenando índices.
CHECKSUM	Devuelve la suma de verificación de una tabla.
SHOW	Conjunto de comandos para mostrar información diversa sobre toda clase de objetos.