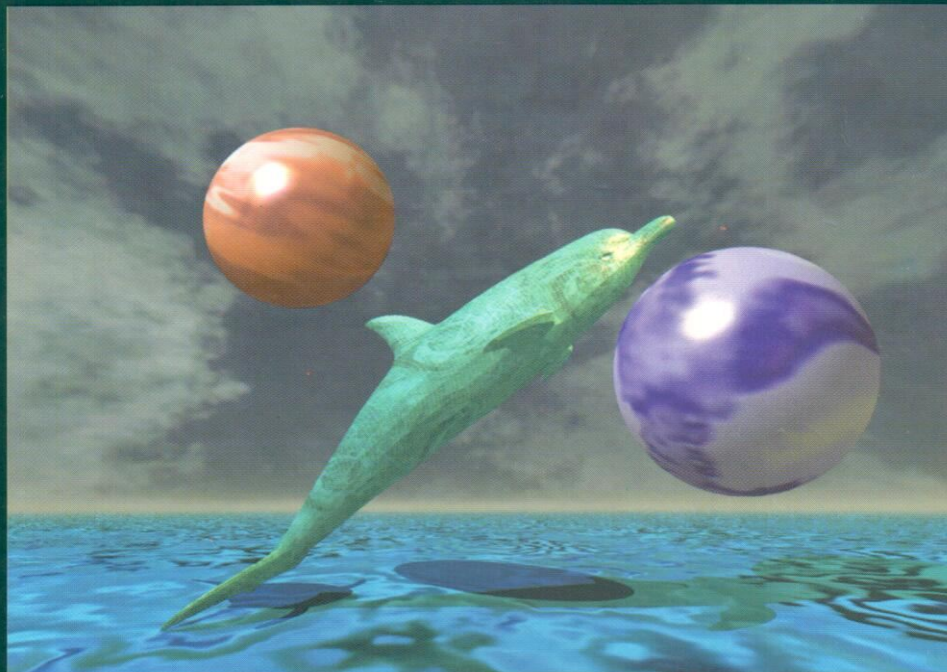


MySQL

para Windows y Linux

2ª EDICIÓN



César Pérez

Alfaomega  Ra-Ma®

También existe un log de actualización que hace el seguimiento de las consultas que modifican la base de datos y que contiene un registro para cada consulta del tipo DELETE, INSERT, REPLACE, CREATE TABLE, DROP TABLE, GRANT y REVOKE. Estas consultas aparecen en el fichero log como sentencias SQL, lo que permitirá restaurar las tablas de la base de datos en caso de una caída del sistema con pérdida de información. Se pueden usar los log de actualización como entrada para *mysql* para restaurar la base de datos.

La sintaxis para habilitar el log general al iniciar el servidor MySQL es la siguiente:

```
mysqld --log [=nombre_ruta]
```

La sintaxis para habilitar el log de actualización al iniciar el servidor MySQL es la siguiente:

```
mysqld --log-update [=nombre_ruta]
```

La opción *nombre_ruta* permite especificar un nombre y ruta en el sistema para el fichero log (distinto del nombre por defecto).

Existe otra sentencia que habilita el logging del archivo de índice.

```
mysqld --log-isam [=nombre_ruta]
```

Estas opciones también se pueden especificar con sintaxis semejante cuando se inicia el servidor MySQL mediante *safe_mysqld* o *mysql.server*.

Es muy conveniente habilitar los archivos log siempre que se inicie el servidor.

COPIAS DE SEGURIDAD

En todo momento debemos disponer de una copia de seguridad de las bases de datos MySQL. Pueden existir colapsos del sistema u operaciones administrativas poco prudentes como DROP DATABASE o DROP TABLE que accidentalmente eliminen o dañen una base de datos. Es necesario estar preparados para este tipo de contingencias habilitando copias de seguridad que permitan la recuperación de la información si es necesario.

En MySQL hay dos métodos para hacer copias de seguridad de las bases de datos. La más conveniente es usar la utilidad *mysqldump*. La otra opción es realizar una copia directa de los archivos de la base de datos mediante las órdenes de copia del sistema operativo *copy*, *cp*, *tar* o *cpio*.

La sentencia *mysqldump* es más lenta que la copia directa, pero es mucho más segura porque opera en cooperación con el servidor MySQL evitando los problemas de la realización de copias mientras se está produciendo una operación en la base de datos. Además, genera archivos de texto transportables a otras máquinas que harán posible la recuperación de la información en otro sistema distinto del nuestro.

Copias de seguridad con MySQLDUMP (backups)

La utilidad *mysqldump* escribe el contenido de las tablas de la base de datos en ficheros de texto que pueden usarse para copias de seguridad de bases de datos, para mover bases de datos a otro servidor o para crear una base de datos de pruebas basada en una ya existente. Su sintaxis es la siguiente:

```
mysqldump [opciones] nombre_base_de_datos [nombre_tabla]...
```

Cuando no se especifica el nombre de tabla o tablas, el proceso afecta a todas las tablas de la base de datos.

La sintaxis más habitual para volcar la base de datos completa en un fichero de backup es la siguiente:

```
mysqldump nombre_base_de_datos > fichero_de_backup
```

No obstante, suele utilizarse la opción `--opt`, que permite afinar la copia mediante el uso de instrucciones `DROP TABLE`, `LOCK TABLE` y otras previas a la copia para que ésta sea más segura. La sintaxis será:

```
mysqldump --opt nombre_base_de_datos > fichero_de_backup
```

Las opciones más importantes de *mysqldump* ya se estudiaron en el capítulo anterior.

Es conveniente nombrar adecuadamente el fichero de backup añadiendo a su nombre la fecha y hora en que se realiza la copia. Además, si el tamaño de la copia es muy grande, puede comprimirse el backup a la vez que se realiza mediante una sentencia similar a la siguiente:

```
mysqldump nombre_base_de_datos | gzip > fichero_de_backup
```

No olvidemos que ante tamaños muy altos de la base de datos se puede hacer backup tabla a tabla o en grupos de tablas.

```
mysqldump nombre_base_de_datos tabla1 tabla2 tabla3 | gzip > fichero_de_backup
```

Recuperación de una base de datos

La primera cuestión a tener en cuenta ante la recuperación de una base de datos es el nivel de recuperación que se desea. Para recuperar la base de datos incluyendo las tablas de concesión es necesario comenzar arrancando el servidor con la opción `--skip-grant-tables` como sigue:

```
mysqld --skip-grant-tables
```

Además, después de restaurar las tablas será necesario indicar al servidor que cargue las tablas de concesión y empiece a usarlas mediante la sentencia:

```
mysqladmin flush-privileges
```

Para restaurar las tablas se pueden utilizar los archivos de backup generados por `mysqldump` como entrada para `mysql` mediante la sintaxis:

```
mysql nombre_base_datos < fichero_de_backup
```

Si la copia de seguridad de la base de datos se ha realizado con órdenes de sistema operativo como `copy`, `cp` o `tar`, se realizará la recuperación copiando los ficheros a los directorios adecuados. En este caso es necesario conocer muy bien la estructura de directorios de las bases de datos MySQL para volver a situar cada fichero en su sitio. Además, el servidor ha de desconectarse antes de iniciar la copia y reiniciarlo al finalizar la tarea de copia.

Una tercera opción de recuperación de una base de datos MySQL es el uso de los ficheros log de actualización que contienen la sintaxis SQL de la construcción de todas las tablas de la base de datos como ficheros de entrada para el comando `mysql`. Si los ficheros de log de actualización son `log1`, `log2` y `log3`, la recuperación se hará como sigue:

```
mysql nombre_base_datos < log1
```

```
mysql nombre_base_datos < log2
```

```
mysql nombre_base_datos < log3
```

En este caso suele utilizarse la opción `--one database` como sigue:

```
mysql --one-database nombre_base_datos < log1
```

```
mysql --one-database nombre_base_datos < log2
```

```
mysql --one-database nombre_base_datos < log3
```