

mysqldump - Guia practica

mysqldump es parte de los "MySQL Client Programs" programas de cliente de MySQL, que puede ser utilizado para generar respaldos de bases de datos y ser usados o incluso para ser transferidos a otro servidor de base datos SQL (No estrictamente tiene que ser un servidor MySQL).

En esta guía práctica de LinuxTotal aprenderás (por ejemplos) los usos y opciones más comunes de este potente comando de MySQL.

Uso básico

Tres formas básicas de invocar **mysqldump** son posibles:

```
1) mysqldump [opciones] nombre_bd [nombre_tabla1 nombre_tabla2 ...]
> respaldo.sql
2) mysqldump [opciones] --databases nombre_bd1 nombre_bd2 >
respaldo.sql
3) mysqldump [opciones] --all-databases > respaldo.sql
```

1) respalda una sola base de datos, indicando su nombre, y opcionalmente una o más tablas de la misma base de datos. Si no se indican tablas, se respaldan todas.

2) respalda una o más bases de datos de forma completa, no se pueden indicar tablas individuales de esta manera.

3) respalda de forma completa todas las bases de datos del servidor MySQL de forma completa, no se pueden indicar tablas individuales de esta manera.

En todos los casos se indica que la salida del comando, es decir, el vaciado del respaldo se guarde en el archivo llamado "respaldo.sql", este archivo es el resultado final, lo que puede utilizarse después para restaurar una base de datos (véase ejemplos más adelante).

Opciones más comunes

mysqldump es un comando con decenas de opciones, para verlas todas utiliza la ayuda en línea que se proporciona a través de *--help*

```
mysqldump --help
```

Lo anterior muestra las posibles opciones específicas de la versión de MySQL que utilices

Ejemplos de uso de **mysqldump**

Respaldo de una sola base de datos completa

```
mysqldump clientes > clientes.sql
```

Respaldo de una sola base de datos con dos tablas

```
mysqldump clientes saldos facturas > clientes.sql
```

Respaldo completo de base de datos clientes y ventas

```
mysqldump --databases clientes ventas >  
respaldo_cli_ven_sep_2011.sql
```

Respaldamos la base de datos clientes pero ignoramos las tablas 'temporal' y 'basura' (Obligatorio indicar base.tabla)

```
mysqldump clientes --ignore-table=clientes.temporal  
--ignore-table=clientes.basura > respaldo_clientes_2011.sql
```

Respaldo completo de todas las bases de datos

```
mysqldump --all-databases > respaldo_full_sep_2011.sql
```

Si se tiene contraseña (como debe ser) se indica usuario y que pregunte por el password

```
mysqldump -u root -p --all-databases > respaldo_full_sep_2011.sql
```

No muy buena idea, pero se puede indicar el password directamente, además nos aseguramos que se indiquen las opciones por defecto más comunes

```
mysqldump -u root -psecreto --all-databases --opt >
respaldo_full_sep_2011.sql
```

Respaldo de una base de datos transaccional tipo InnoDB o BDB asegurando su consistencia

```
mysqldump -u root -p --single-transaction --quick ventas >
respaldo_ventas_2011.sql
```

Todas las bases de datos del host 192.168.0.100 y agregamos los procedimientos almacenados que sean respaldados también.

```
mysqldump -h 192.168.1.100 -u root -p --routines --all-databases >
respaldo_ventas_2011.sql
```

Respaldo de las bases de datos clientes y pedidos, con todas las opciones específicas para re-crear las tablas, además añadimos 'drop database' para asegurarnos que en la restauración se creé desde cero el respaldo, además ignoramos errores..

```
mysqldump -u root -p --create-options --add-drop-database --force
--databases clientes pedidos > respaldo_ven_ped_2011.sql
```

Respaldo completo de un servidor MySQL maestro en replicación, indicando en el respaldo la posición para sincronización con servidores esclavos, además añadimos insertar completos que incluyen los nombres de columnas en sentencias INSERT

```
mysqldump -u root --password=secreto --all-databases --master-data
--complete-insert > respaldo_2011.sql
```

Respaldamos solo el esquema de clientes sin registros

```
mysqldump --no-data clientes > respaldo_esquema_clientes_2011.sql
```

Se produce una salida compatible para restaurar la base de datos en Oracle

```
mysqldump --compatible=oracle --databases clientes >
respaldo_clientes_oracle_2011.sql
```

mysqldump con gzip

Al mismo tiempo que realizamos el respaldo podemos comprimirlo para ahorrar espacio en nuestros respaldos.

```
mysqldump -u root -p --all-databases | gzip > respaldo_2011.sql.gz
```

Para descomprimir lo anterior y dejar el archivo listo para la restauración en si utiliza `gunzip`

```
gunzip respaldo_2011.sql.gz
```

Restaurando el respaldo

Hay varias maneras de lograr la restauración del respaldo. Con el comando `mysql`:

```
mysql -u root -p < respaldo.sql
```

Si se utilizó `gzip` para comprimir el respaldo, se puede descomprimir y restaurar en una sola línea:

```
gunzip < respaldo.sql.gz | mysql -u root -p
```

Si el respaldo contiene una sola base de datos y no contiene sentencias 'drop database' y 'create database', se puede entonces indicar la base de datos donde se debe realizar la restauración:

```
mysql -u root -p clientes < respaldo_clientes.sql
```

Lo anterior (cuando ya existe la base de datos) también se puede lograr con el comando `mysqlimport`

```
mysqlimport -u root -p clientes respaldo_clientes.sql
```

Es posible también utilizar la opción "-e" (execute) de `mysql`, que permite ejecutar un archivo con sentencias SQL, y un respaldo es exactamente eso, un largo script con sentencias SQL para recrear esquema y datos:

```
mysql -u root -p -e "source /ruta/a/respaldo.sql"
```

Un simple script para respaldos automáticos por medio de **cron**

El siguiente es un ejemplo de un script muy simple para respaldar todas las bases de datos con las opciones más completas de manera automática, además se comprime. El archivo que se genera contiene la fecha y la hora al momento del respaldo, por medio de **cron** se programa su ejecución a las 11:00 de la noche todos los días.

Solo si la ejecución del comando fue correcta y termina, anexa un pequeño mensaje al final de un archivo de bitácora, así podemos consultarlo y observar que todos los días terminó correctamente

Guardar el respaldo como respaldoMySQL.sh y cambia sus permisos de ejecución que permita ejecución **chmod 700 respaldoMySQL.sh** (ver artículos relacionados más abajo para enlace de este comando)

```
#!/bin/bash
# fecha y hora, ejemplo 20110920_2300
# archivo termina como 'mysql_20110920_2300.sql.gz'

fecha=`date +%Y%m%d_%H%M`
archivo="mysql_`date +%Y%m%d_%H%M`.sql.gz"

# elimina la opción --master-data si tu servidor no es maestro en
replicación

mysqldump --user=root --password=xyz --all-databases --opt
--routines \
--add-drop-database --add-drop-table --complete-insert
--create-options \
--master-data | gzip > /ruta/a/respaldos/$archivo && \
echo "Respaldo realizado exitosamente el `date`" >>
/ruta/a/bitacora.txt

#adicionalmente si se tiene otro servidor linux, copiamos el archivo
#ya generado de respaldo por medio de scp de ssh y así aseguramos
#que el respaldo esté en otra locación, logrando redundancia (en
#los artículos #relacionados chécate "Autentificación SSH con llave
privada")
# El usuario que ejecuta el comando es 'userdba' que existe en ambos
servidores

su -c "scp -pq /ruta/a/respaldos/$archivo
192.168.1.100:/ruta/a/respaldosmysql/. " userdba \
&& echo "Copia a servidor remoto se completó con éxito el `date`" >>
/ruta/a/bitacora.txt
```

La línea **cron** correspondiente para su ejecución programada quedaría de la siguiente manera en */etc/crontab*

```
0 23 * * * root /ruta/a/respaldoMySQL.sh
```