

MySQL 数据库备份方法

总结了 MySQL 数据库备份及恢复常用命令 mysqldump, source 的用法。

备份 MySQL 数据库的命令

```
mysqldump -hhostname -uusername -ppassword databasename >  
backupfile.sql
```

备份 MySQL 数据库为带删除表的格式

备份 MySQL 数据库为带删除表的格式, 能够让该备份覆盖已有数据库而不需要手动删除原有数据库。

```
mysqldump --add-drop-table -uusername -ppassword databasename >  
backupfile.sql
```

直接将 MySQL 数据库压缩备份

```
mysqldump -hhostname -uusername -ppassword databasename | gzip >  
backupfile.sql.gz
```

备份 MySQL 数据库某个(些)表

```
mysqldump -hhostname -uusername -ppassword databasename  
specific_table1 specific_table2 > backupfile.sql
```

同时备份多个 MySQL 数据库

```
mysqldump -hhostname -uusername -ppassword -databases  
databasename1 databasename2 databasename3 > multibackupfile.sql
```

仅仅备份数据库结构

```
mysqldump -n-no-data -d-databases databasename1 databasename2  
databasename3 > structurebackupfile.sql
```

备份服务器上所有数据库

```
mysqldump -a-all-databases > allbackupfile.sql
```

还原 MySQL 数据库的命令

```
mysql -hhostname -uusername -ppassword databasename < backupfile.sql
```

还原压缩的 MySQL 数据库

```
gunzip < backupfile.sql.gz | mysql -uusername -ppassword  
databasename
```

将数据库转移到新服务器

```
mysqldump -uusername -ppassword databasename | mysql -host=*. *.*.*.*  
-C databasename
```

几个常用用例:

1. 导出整个数据库

```
mysqldump -u 用户名 -p 数据库名 > 导出的文件  
名
```

```
mysqldump -u root -p dataname >dataname.sql
```

这个时候会提示要你输入 root 用户名的密码, 输入密码后 dataname 数据库就成功备份在 mysql/bin/ 目录中.

2. 导出一个表

```
mysqldump -u 用户名 -p 数据库名 表名> 导出的文件名  
mysqldump -u root -p dataname users> dataname_users.sql
```

3. 导出一个数据库结构

```
mysqldump -u wcnc -p -d - add-drop-table  
smgp_apps_wcnc >d:\wcnc_db.sql
```

-d 没有数据 - add-drop-table 在每个 create 语句之前增加一个 drop table

4. 导入数据库

常用 source 命令

进入 mysql 数据库控制台,

如 mysql -u root -p

```
mysql>use 数据库
```

然后使用 source 命令, 后面参数为脚本文件(如这里用到的.sql)

```
mysql>source d:\wcnc_db.sql
```

mysqldump 支持下列选项:

- add-locks

在每个表导出之前增加 LOCK TABLES 并且之后 UNLOCK TABLE。 (为了使得更快地插入到 MySQL)。

- add-drop-table

在每个 create 语句之前增加一个 drop table。

- allow-keywords

允许创建是关键词的列名字。这由表名前缀于每个列名做到。

-c, - complete-insert

使用完整的 insert 语句(用列名字)。

-C, - compress

如果客户和服务器均支持压缩，压缩两者间所有的信息。

- delayed

用 INSERT DELAYED 命令插入行。

-e, - extended-insert

使用全新多行 INSERT 语法。(给出更紧缩并且更快的插入语句)

-#, - debug[=option_string]

跟踪程序的使用(为了调试)。

- help

显示一条帮助消息并且退出。

- fields-terminated-by=...

- fields-enclosed-by=...

- fields-optionally-enclosed-by=...

- fields-escaped-by=...

- fields-terminated-by=...

这些选择与-T 选择一起使用，并且有相应的 LOAD DATA INFILE 子句相同的含义。

LOAD DATA INFILE 语法。

-F, - flush-logs

在开始导出前，洗掉在 MySQL 服务器中的日志文件。

-f, - force,

即使我们在一个表导出期间得到一个 SQL 错误，继续。

-h, - host=..

从命名的主机上的 MySQL 服务器导出数据。缺省主机是 localhost。

-l, - lock-tables.

为开始导出锁定所有表。

-t, - no-create-info

不写入表创建信息(CREATE TABLE 语句)

`-d, -no-data`

不写入表的任何行信息。如果你只想得到一个表的结构导出，这是很有用的！

`-opt`

同 `-quick` `-add-drop-table` `-add-locks` `-extended-insert`

`-lock-tables`。

应该给你为读入一个 MySQL 服务器的尽可能最快的导出。

`-pyour_pass, -password[=your_pass]`

与服务器连接时使用的口令。如果你不指定 “=your_pass” 部分，mysqldump 需要来自终端的口令。

`-P port_num, -port=port_num`

与一台主机连接时使用的 TCP/IP 端口号。(这用于连接到 localhost 以外的主机，因为它使用 Unix 套接字。)

`-q, -quick`

不缓冲查询，直接导出至 stdout;使用 `mysql_use_result()` 做它。

`-S /path/to/socket, -socket=/path/to/socket`

与 localhost 连接时(它是缺省主机)使用的套接字文件。

`-T, -tab=path-to-some-directory`

对于每个给定的表，创建一个 `table_name.sql` 文件，它包含 SQL `CREATE` 命令，和一个 `table_name.txt` 文件，它包含数据。注意：这只有在 mysqldump 运行在 mysqld 守护进程运行的同一台机器上的时候才工作。.txt 文件的格式根据 `-fields-xxx` 和 `-lines-xxx` 选项来定。

`-u user_name, -user=user_name`

与服务器连接时，MySQL 使用的用户名。缺省值是你的 Unix 登录名。

`-O var=option, -set-variable var=option`

设置一个变量的值。可能的变量被列在下面。

`-v, -verbose`

冗长模式。打印出程序所做的更多的信息。

`-V, -version`

打印版本信息并且退出。

`-w, -where='where-condition'`

只导出被选择的记录;注意引号是强制的！

“`-where=user='jimf'`” “`-userid>1`” “`-userid<1`”

最常见的 mysqldump 使用可能制作整个数据库的一个备份：

```
mysqldump -opt database > backup-file.sql
```

但是它对用来自于一个数据库的信息充实另外一个 MySQL 数据库也是有用的：

```
mysqldump -opt database | mysql -host=remote-host -C database
```

由于 mysqldump 导出的是完整的 SQL 语句，所以用 mysql 客户程序很容易就能把数据导入了：

```
shell> mysqladmin create target_db_name
```

```
shell> mysql target_db_name < backup-file.sql
```

就是 shell> mysql 库名 < 文件名