



链滴

MySQL 数据库备份与恢复

作者: [CloudOf](#)

原文链接: <https://ld246.com/article/1531886876867>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



MySQL数据库备份与恢复——使用mysqldump 导入与导出方法总结

MySQL数据库备份与恢复——使用mysqldump 导入与导出方法总结

mysqldump客户端可用来转储数据库或搜集数据库进行备份或将数据转移到另一个sql服务器(不一定一个mysql服务器)。转储包含创建表和/或装载表的sql语句。

如果在服务器上进行备份，并且表均为myisam表，应考虑使用mysqlhotcopy，因为可以更快地进行备份和恢复。

本文从四部分介绍了mysql数据备份与恢复：第一，mysql基本常识；第二，数据备份与恢复示例；三，mysqldump具体参数说明。

一、MySQL基本常识

(1) 连接mysql:

格式：`mysql -h主机地址 -u用户名 -p用户密码`

1、例1：连接到本机上的MYSQL

键入命令`mysql -uroot -p`，回车后提示你输密码，如果刚安装好MYSQL，超级用户root是没有密码，故直接回车即可进入到MYSQL中了，MYSQL的提示符是：`mysql>`

2、例2：连接到远程主机上的MYSQL

假设远程主机的IP为：10.0.0.1，用户名为root,密码为12356。则键入以下命令：

```
mysql -h10.0.0.1 -uroot -p12356
```

(注: u与root可以不用加空格, 其它也一样)

3、退出MYSQL命令

exit (回车)

(2) 常见命令:

1、显示数据库列表:

show databases;

刚开始时只有两个数据库: mysql和test。mysql库很重要它里面有MYSQL的系统信息, 我们改密码和增用户, 实际上就是用这个库进行操作。

2、显示库中的数据表:

use mysql; //打开库

show tables;

3、显示数据表的结构:

describe 表名;

简写: desc 表名

4、建库:

create database 库名;

5、建表:

use 库名;

create table 表名 (字段设定列表);

6、删库和删表:

drop database 库名;

drop table 表名;

7、将表中记录清空:

delete from 表名;

8、显示表中的记录:

select * from 表名;

二、数据备份与恢复

(1) 备份: 从数据库导出数据:

格式: mysqldump -h链接ip -P(大写)端口 -u用户名 -p密码数据库名 > d:XX.sql(路劲)

示例: mysqldump -h132.72.192.432 -P3307 -uroot -p8888 htgl > d:\htgl.sql;

** (2) 备份导出示例: **

1、导出数据和表结构——将特定数据库特定表中的数据和表格结构和数据全部返回

mysqldump --u b_user -h 101.3.20.33 -p'H_password' -P3306 database_di up_subjects > 010_0630_up_subjects.sql

2、导出表结构却不导出表数据——只返回特定数据库特定表格的表格结构, 不返回数据, 添加 "-d" 命令参数

mysqldump --u b_user -h 101.3.20.33 -p'H_password' -P3306 -d database_di up_subjects > 010_0630_up_subjects.sql

3、导出表结构和满足挑顶条件的表数据——只返回特定数据库中特定表的表格结构和满足特定条件数据

mysqldump --u b_user -h 101.3.20.33 -p'H_password' -P3306 database_di up_subjects --where="ctime>'2017-01-01' and ctime<'2017-06-30'" > 0101_0630_up_subjects.sql

4、导出数据却不导出表结构——只返回特定数据库中特定表格的数据, 不返回表格结构, 添加 "-t" 命令参数

mysqldump --u b_user -h 101.3.20.33 -p'H_password' -t -P3306 database_di up_subjects > 0101_0630_up_subjects.sql

5、导出特定数据库的所有表格的表结构及其数据, 添加 "--databases " 命令参数

mysqldump --u b_user -h 101.3.20.33 -p'H_password' -P3306 --databases test > all_database.sql

** (3) 恢复, 导入数据库数据: **

将导出的本地文件导入到指定数据库

1、系统命令行

格式: mysql -h链接ip -P(大写)端口 -u用户名 -p密码 数据库名 < d:XX.sql(路劲)

mysql -uusername -ppassword db1

2、或mysql命令行

mysql>

user db1;

source tb1_tb2.sql;

3、恢复整个数据库的方法:

mysql -u b_user -h 101.3.20.33 -p'H_password' -P3306 < all_database.sql

**** (4) 具体恢复示例: ****

1、先登录该MySQL服务器，切换要导入数据的数据

具体命令如下：

```
mysql> use test;
```

```
mysql> source /home/test/0101_0630_up_subjects.sql
```

```
Query OK, 0 rows affected (0.01 sec)
```

```
.....
```

```
Query OK, 0 rows affected (0.00 sec)
```

```
Query OK, 9 rows affected (0.00 sec)
```

```
Records: 9 Duplicates: 0 Warnings: 0
```

注释：表示影响的记录为9行，重复的记录有0行，警告的记录有0个

2、直接使用系统命令行

```
mysql -u b_user -h 101.3.20.33 -p'H_password' -P3306 test
```

```
**
```

```
**
```

****三、mysqldump参数说明 ****

(1) mysqldump字符集设置

```
mysqldump -uusername -ppassword --default-character-set=gb2312 db1 table1 > tb1.sql
```

mysqldump客户端可用来转储数据库或搜集数据库进行备份或将数据转移到另一个sql服务器(不一定一个mysql服务器)。转储包含创建表和/或装载表的sql语句。

如果在服务器上进行备份，并且表均为myisam表，应考虑使用mysqlhotcopy，因为可以更快地进行备份和恢复。

(2) 其他参数

```
--all-databases , -A
```

导出全部数据库。

```
mysqldump -uroot -p --all-databases
```

```
--all-tablespaces , -Y
```

导出全部表空间。

```
mysqldump -uroot -p --all-databases --all-tablespaces
```

```
--no-tablespaces , -y
```

不导出任何表空间信息。

```
mysqldump -uroot -p --all-databases --no-tablespaces  
--add-drop-database
```

每个数据库创建之前添加drop数据库语句。

```
mysqldump -uroot -p --all-databases --add-drop-database  
--add-drop-table
```

每个数据表创建之前添加drop数据表语句。(默认为打开状态, 使用--skip-add-drop-table取消选项)

```
mysqldump -uroot -p --all-databases (默认添加drop语句)  
mysqldump -uroot -p --all-databases --skip-add-drop-table (取消drop语句)  
--add-locks
```

在每个表导出之前增加LOCK TABLES并且之后UNLOCK TABLE。(默认为打开状态, 使用--skip-add-locks取消选项)

```
mysqldump -uroot -p --all-databases (默认添加LOCK语句)  
mysqldump -uroot -p --all-databases --skip-add-locks (取消LOCK语句)  
--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。

```
mysqldump -uroot -p --all-databases --allow-keywords  
--apply-slave-statements
```

在'CHANGE MASTER'前添加'STOP SLAVE', 并且在导出的最后添加'START SLAVE'。

```
mysqldump -uroot -p --all-databases --apply-slave-statements  
--character-sets-dir
```

字符集文件的目录

```
mysqldump -uroot -p --all-databases --character-sets-dir=/usr/local/mysql/share/mysql/char  
ets  
--comments
```

附加注释信息。默认为打开, 可以用--skip-comments取消

```
mysqldump -uroot -p --all-databases (默认记录注释)  
mysqldump -uroot -p --all-databases --skip-comments (取消注释)  
--compatible
```


导出的数据将和其它数据库或旧版本的MySQL 相兼容。值可以为ansi、mysql323、mysql40、postgresql、oracle、mssql、db2、maxdb、no_key_options、no_tables_options、no_field_options、

要使用几个值，用逗号将它们隔开。它并不保证能完全兼容，而是尽量兼容。

```
mysqldump -uroot -p --all-databases --compatible=ansi
```

```
--compact
```

导出更少的输出信息(用于调试)。去掉注释和头尾等结构。可以使用选项：--skip-add-drop-table --skip-add-locks --skip-comments --skip-disable-keys

```
mysqldump -uroot -p --all-databases --compact
```

```
--complete-insert, -c
```

使用完整的insert语句(包含列名称)。这么做能提高插入效率，但是可能会受到max_allowed_packet数的影响而导致插入失败。

```
mysqldump -uroot -p --all-databases --complete-insert
```

```
--compress, -C
```

在客户端和服务端之间启用压缩传递所有信息

```
mysqldump -uroot -p --all-databases --compress
```

```
--create-options, -a
```

在CREATE TABLE语句中包括所有MySQL特性选项。(默认为打开状态)

```
mysqldump -uroot -p --all-databases
```

```
--databases, -B
```

导出几个数据库。参数后面所有名字参量都被看作数据库名。

```
mysqldump -uroot -p --databases test mysql
```

```
--debug
```

输出debug信息，用于调试。默认值为：d:t/tmp/mysqldump.trace

```
mysqldump -uroot -p --all-databases --debug
```

```
mysqldump -uroot -p --all-databases --debug=" d:t/tmp/debug.trace"
```

```
--debug-check
```

检查内存和打开文件使用说明并退出。

```
mysqldump -uroot -p --all-databases --debug-check
```

```
--debug-info
```

输出调试信息并退出

```
mysqldump -uroot -p --all-databases --debug-info
```

```
--default-character-set
```

设置默认字符集，默认值为utf8

```
mysqldump -uroot -p --all-databases --default-character-set=utf8
```

```
--delayed-insert
```

采用延时插入方式 (INSERT DELAYED) 导出数据

```
mysqldump -uroot -p --all-databases --delayed-insert
```

```
--delete-master-logs
```

master备份后删除日志. 这个参数将自动激活--master-data。

```
mysqldump -uroot -p --all-databases --delete-master-logs
```

```
--disable-keys
```

对于每个表，用`/*!40000 ALTER TABLE tbl_name DISABLE KEYS ;`和`/*!40000 ALTER TABLE tbl_name ENABLE KEYS ;`语句引用INSERT语句。这样可以更快地导入dump出来的文件，因为它是在插入有行后创建索引的。该选项只适合MyISAM表，默认为打开状态。

```
mysqldump -uroot -p --all-databases
```

```
--dump-slave
```

该选项将主的binlog位置和文件名追加到导出数据的文件中(show slave status)。设置为1时，将会以HANGE MASTER命令输出到数据文件；设置为2时，会在change前加上注释。该选项将会打开--lock-all-tables，除非--single-transaction被指定。该选项会自动关闭--lock-tables选项。默认值为0。

```
mysqldump -uroot -p --all-databases --dump-slave=1
```

```
mysqldump -uroot -p --all-databases --dump-slave=2
```

```
--master-data
```

该选项将当前服务器的binlog的位置和文件名追加到输出文件中(show master status)。如果为1，会输出CHANGE MASTER 命令；如果为2，输出的CHANGE MASTER命令前添加注释信息。该选项打开--lock-all-tables 选项，除非--single-transaction也被指定（在这种情况下，全局读锁在开始导时获得很短的时间；其他内容参考下面的--single-transaction选项）。该选项自动关闭--lock-table选项。

```
mysqldump -uroot -p --host=localhost --all-databases --master-data=1;
```

```
mysqldump -uroot -p --host=localhost --all-databases --master-data=2;
```

```
--events, -E
```

导出事件。


```
mysqldump -uroot -p --all-databases --events
```

```
--extended-insert, -e
```

使用具有多个VALUES列的INSERT语法。这样使导出文件更小，并加速导入时的速度。默认为打开状态，使用--skip-extended-insert取消选项。

```
mysqldump -uroot -p --all-databases
```

```
mysqldump -uroot -p --all-databases--skip-extended-insert （取消选项）
```

```
--fields-terminated-by
```

导出文件中忽略给定字段。与--tab选项一起使用，不能用于--databases和--all-databases选项

```
mysqldump -uroot -p test test --tab=" /home/mysql" --fields-terminated-by=" #"
```

```
--fields-enclosed-by
```

输出文件中的各个字段用给定字符包裹。与--tab选项一起使用，不能用于--databases和--all-databases选项

```
mysqldump -uroot -p test test --tab=" /home/mysql" --fields-enclosed-by=" #"
```

```
--fields-optionally-enclosed-by
```

输出文件中的各个字段用给定字符选择性包裹。与--tab选项一起使用，不能用于--databases和--all-databases选项

```
mysqldump -uroot -p test test --tab=" /home/mysql" --fields-enclosed-by=" #" --fields-optionally-enclosed-by=" #"
```

```
--fields-escaped-by
```

输出文件中的各个字段忽略给定字符。与--tab选项一起使用，不能用于--databases和--all-databases选项

```
mysqldump -uroot -p mysql user --tab=" /home/mysql" --fields-escaped-by=" #"
```

```
--flush-logs
```

开始导出之前刷新日志。

请注意：假如一次导出多个数据库(使用选项--databases或者--all-databases)，将会逐个数据库刷新日志。除使用--lock-all-tables或者--master-data外。在这种情况下，日志将会被刷新一次，相应的以表同时被锁定。因此，如果打算同时导出和刷新日志应该使用--lock-all-tables 或者--master-data和--flush-logs。

```
mysqldump -uroot -p --all-databases --flush-logs
```

```
--flush-privileges
```

在导出mysql数据库之后，发出一条FLUSH PRIVILEGES 语句。为了正确恢复，该选项应该用于导出mysql数据库和依赖mysql数据库数据的任何时候。

```
mysqldump -uroot -p --all-databases --flush-privileges
```

`--force`

在导出过程中忽略出现的SQL错误。

```
mysqldump -uroot -p --all-databases --force
```

`--help`

显示帮助信息并退出。

```
mysqldump --help
```

`--hex-blob`

使用十六进制格式导出二进制字符串字段。如果有二进制数据就必须使用该选项。影响到的字段类型 BINARY、VARBINARY、BLOB。

```
mysqldump -uroot -p --all-databases --hex-blob
```

`--host, -h`

需要导出的主机信息

```
mysqldump -uroot -p --host=localhost --all-databases
```

`--ignore-table`

不导出指定表。指定忽略多个表时，需要重复多次，每次一个表。每个表必须同时指定数据库和表名
例如：`--ignore-table=database.table1 --ignore-table=database.table2`

```
mysqldump -uroot -p --host=localhost --all-databases --ignore-table=mysql.user
```

`--include-master-host-port`

在`--dump-slave`产生的'`CHANGE MASTER TO..`'语句中增加'`MASTER_HOST=, MASTER_PORT=`'

```
mysqldump -uroot -p --host=localhost --all-databases --include-master-host-port
```

`--insert-ignore`

在插入行时使用INSERT IGNORE语句。

```
mysqldump -uroot -p --host=localhost --all-databases --insert-ignore
```

`--lines-terminated-by`

输出文件的每行用给定字符串划分。与`--tab`选项一起使用，不能用于`--databases`和`--all-databases`项。

```
mysqldump -uroot -p --host=localhost test test --tab="/tmp/mysql" --lines-terminated-b  
=" ##"
```

`--lock-all-tables, -x`

提交请求锁定所有数据库中的所有表，以保证数据的一致性。这是一个全局读锁，并且自动关闭`--sing
e-transaction` 和`--lock-tables` 选项。

```
mysqldump -uroot -p --host=localhost --all-databases --lock-all-tables
```

```
--lock-tables, -l
```

开始导出前，锁定所有表。用READ LOCAL锁定表以允许MyISAM表并行插入。对于支持事务的表如InnoDB和BDB，--single-transaction是一个更好的选择，因为它根本不需要锁定表。

请注意当导出多个数据库时，--lock-tables分别为每个数据库锁定表。因此，该选项不能保证导出文中的表在数据库之间的逻辑一致性。不同数据库表的导出状态可以完全不同。

```
mysqldump -uroot -p --host=localhost --all-databases --lock-tables
```

```
--log-error
```

附加警告和错误信息到给定文件

```
mysqldump -uroot -p --host=localhost --all-databases --log-error=/tmp/mysqldump_error_l  
g.err
```

```
--max_allowed_packet
```

服务器发送和接受的最大包长度。

```
mysqldump -uroot -p --host=localhost --all-databases --max_allowed_packet=10240
```

```
--net_buffer_length
```

TCP/IP和socket连接的缓存大小。

```
mysqldump -uroot -p --host=localhost --all-databases --net_buffer_length=1024
```

```
--no-autocommit
```

使用autocommit/commit 语句包裹表。

```
mysqldump -uroot -p --host=localhost --all-databases --no-autocommit
```

```
--no-create-db, -n
```

只导出数据，而不添加CREATE DATABASE 语句。

```
mysqldump -uroot -p --host=localhost --all-databases --no-create-db
```

```
--no-create-info, -t
```

只导出数据，而不添加CREATE TABLE 语句。

```
mysqldump -uroot -p --host=localhost --all-databases --no-create-info
```

```
--no-data, -d
```

不导出任何数据，只导出数据库表结构。

```
mysqldump -uroot -p --host=localhost --all-databases --no-data
```

```
--no-set-names, -N
```

等同于--skip-set-charset

```
mysqldump -uroot -p --host=localhost --all-databases --no-set-names
```

--opt

等同于--add-drop-table, --add-locks, --create-options, --quick, --extended-insert, --lock-tables --set-charset, --disable-keys 该选项默认开启, 可以用--skip-opt禁用.

```
mysqldump -uroot -p --host=localhost --all-databases --opt
```

--order-by-primary

如果存在主键, 或者第一个唯一键, 对每个表的记录进行排序。在导出MyISAM表到InnoDB表时有, 但会使得导出工作花费很长时间。

```
mysqldump -uroot -p --host=localhost --all-databases --order-by-primary
```

--password, -p

连接数据库密码

--pipe(windows系统可用)

使用命名管道连接mysql

```
mysqldump -uroot -p --host=localhost --all-databases --pipe
```

--port, -P

连接数据库端口号

--protocol

使用的连接协议, 包括: tcp, socket, pipe, memory.

```
mysqldump -uroot -p --host=localhost --all-databases --protocol=tcp
```

--quick, -q

不缓冲查询, 直接导出到标准输出。默认为打开状态, 使用--skip-quick取消该选项。

```
mysqldump -uroot -p --host=localhost --all-databases
```

```
mysqldump -uroot -p --host=localhost --all-databases --skip-quick
```

--quote-names, -Q

使用(`)引起表和列名。默认为打开状态, 使用--skip-quote-names取消该选项。

```
mysqldump -uroot -p --host=localhost --all-databases
```

```
mysqldump -uroot -p --host=localhost --all-databases --skip-quote-names
```

--replace

使用REPLACE INTO 取代INSERT INTO.

```
mysqldump -uroot -p --host=localhost --all-databases --replace  
--result-file, -r
```

直接输出到指定文件中。该选项应该用在使用回车换行对（\r\n）换行的系统上（例如：DOS，Windows）。该选项确保只有一行被使用。

```
mysqldump -uroot -p --host=localhost --all-databases--result-file=/tmp/mysqldump_result_file.txt  
--routines, -R
```

导出存储过程以及自定义函数。

```
mysqldump -uroot -p --host=localhost --all-databases --routines  
--set-charset
```

添加'SET NAMES default_character_set'到输出文件。默认为打开状态，使用--skip-set-charset关闭选项。

```
mysqldump -uroot -p --host=localhost --all-databases  
mysqldump -uroot -p --host=localhost --all-databases --skip-set-charset  
--single-transaction
```

该选项在导出数据之前提交一个BEGIN SQL语句，BEGIN 不会阻塞任何应用程序且能保证导出时数据库的一致性状态。它只适用于多版本存储引擎，仅InnoDB。本选项和--lock-tables 选项是互斥的，为LOCK TABLES 会使任何挂起的事务隐含提交。要想导出大表的话，应结合使用--quick 选项。

```
mysqldump -uroot -p --host=localhost --all-databases --single-transaction  
--dump-date
```

将导出时间添加到输出文件中。默认为打开状态，使用--skip-dump-date关闭选项。

```
mysqldump -uroot -p --host=localhost --all-databases  
mysqldump -uroot -p --host=localhost --all-databases --skip-dump-date  
--skip-opt
```

禁用-opt选项。

```
mysqldump -uroot -p --host=localhost --all-databases --skip-opt  
--socket,-S
```

指定连接mysql的socket文件位置，默认路径/tmp/mysql.sock

```
mysqldump -uroot -p --host=localhost --all-databases --socket=/tmp/mysqld.sock  
--tab,-T
```

为每个表在给定路径创建tab分割的文本文件。注意：仅仅用于mysqldump和mysqld服务器运行在同机器上。注意使用--tab不能指定--databases参数

```
mysqldump -uroot -p --host=localhost test test --tab="/home/mysql"
```

--tables

覆盖--databases (-B)参数，指定需要导出的表名，在后面的版本会使用table取代tables。

```
mysqldump -uroot -p --host=localhost --databases test --tables test
```

--triggers

导出触发器。该选项默认启用，用--skip-triggers禁用它。

```
mysqldump -uroot -p --host=localhost --all-databases --triggers
```

--tz-utc

在导出顶部设置时区TIME_ZONE='+00:00'，以保证在不同时区导出的TIMESTAMP 数据或者数据移动其他时区时的正确性。

```
mysqldump -uroot -p --host=localhost --all-databases --tz-utc
```

--user, -u

指定连接的用户名。

--verbose, --v

输出多种平台信息。

--version, -V

输出mysqldump版本信息并退出

--where, -w

只转储给定的WHERE条件选择的记录。请注意如果条件包含命令解释符专用空格或字符，一定要将件引用起来。

```
mysqldump -uroot -p --host=localhost --all-databases --where=" user=' root' "
```

--xml, -X

导出XML格式.

```
mysqldump -uroot -p --host=localhost --all-databases --xml
```

--plugin_dir

客户端插件的目录，用于兼容不同的插件版本。

```
mysqldump -uroot -p --host=localhost --all-databases--plugin_dir=" /usr/local/lib/plugin"
```

--default_auth

客户端插件默认使用权限。

```
mysqldump -uroot -p --host=localhost --all-databases--default-auth="" /usr/local/lib/plugin
```

参考网址：

1、 <http://www.jb51.net/article/73230.htm>

2、 <http://www.cnblogs.com/chenmh/p/5300370.html>