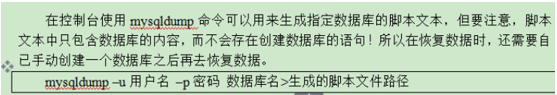
MySQL数据库的备份与恢复（mysqldump）

# 生成SQL脚本，导出数据



注意：mysqldump是windows或linux下就可以使用的命令，不需要进入数据库mysql。但是，恢复数据必须需要进入数据库。

示例：

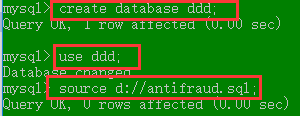
导出数据：



注意：后面不要带分号，密码可以直接写在-p后面，不要有空格。

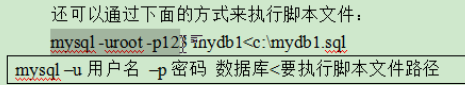
恢复数据库：

首先创建一个新的数据库，然后利用source恢复数据即可。

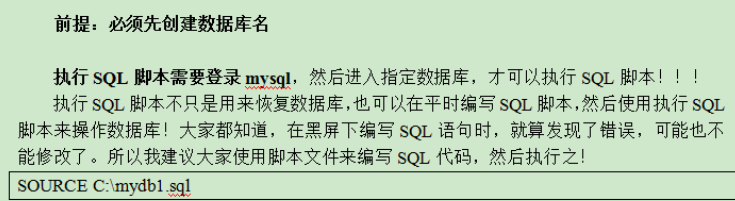


# 执行SQL脚本，恢复数据

## 方法1：不需要登录MySQL，但是本身需要存在待恢复数据的数据库。



## 方法2：需要先进入MySQL。



# ****mysqldump的几种常用方法：****

**（1）导出整个数据库(包括数据库中的数据）**

**mysqldump -u username -p dbname > dbname.sql**

**（2）导出数据库结构（不含数据）**

**mysqldump -u username -p -d dbname > dbname.sql**

**（3）导出数据库中的某张数据表（包含数据）**

**mysqldump -u username -p dbname tablename > tablename.sql**

**（4）导出数据库中的某张数据表的表结构（不含数据）**

**mysqldump -u username -p -d dbname tablename > tablename.sql**

# mysqldump常用参数说明：

**–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)

**–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)

**–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-keysmysqldump -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参数的影响而导致插入失败。mysqldump -uroot -p –all-databases –complete-insert

**–compress**, -C在客户端和服务器之间启用压缩传递所有信息mysqldump -uroot -p –all-databases –compress

**–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:o,/tmp/mysqldump.tracemysqldump -uroot -p –all-databases –debugmysqldump -uroot -p –all-databases –debug=” d:t:o,/tmp/debug.trace”

**–debug-info**输出调试信息并退出mysqldump -uroot -p –all-databases –debug-info

**–default-character-se**t设置默认字符集，默认值为utf8mysqldump -uroot -p –all-databases –default-character-set=latin1

**–delayed-insert**采用延时插入方式（INSERT DELAYED）导出数据mysqldump -uroot -p –all-databases –delayed-insert

–events, -E导出事件。mysqldump -uroot -p –all-databases –events

**–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，相应的所以表同时被锁定。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

**–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

–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–single-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。对于支持事务的表例如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件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

–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

–password, -p连接数据库密码

–port, -P连接数据库端口号

–user, -u指定连接的用户名。

# mysqldump常用实例：

mysqldump常用于数据库的备份与还原，在备份的过程中我们可以根据自己的实际情况添加以上任何参数，假设有数据库test\_db，执行以下命令，即可完成对整个数据库的备份：

**mysqldump -u root -p test\_db > test\_db.sql**

如要对数据进行还原，可执行如下命令：

**mysql -u username -p test\_db < test\_db.sql**

还原数据库操作还可以使用以下方法：

mysql> **source test\_db.sql**

# 更简便的操作：利用管道流：

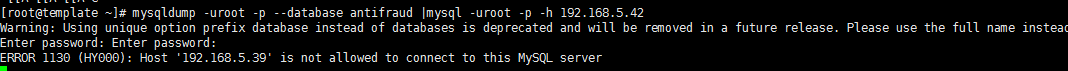
mysqldump –uroot –p password --database antifraud | mysql –uroot –p password –h 192.168.5.42

意义：直接将 192.168.5.42上的mysql数据库中的antifraud数据库拷贝到当前mysql的下。

如果两个密码都没有显式指出，可以需要连续输入两个密码，回车即可。

并没有成功，因为两个服务器之间相连接也需要密码。





# mysql中把一个表的数据导入另一个表中

## 如果两张张表（导出表和目标表）的字段一致，并且希望插入全部数据，可以用这种方法：（此方法只适合导出两表在同一database）

**INSERT INTO 目标表 SELECT \* FROM 来源表;**

例如，要将 articles 表插入到 newArticles 表中，则可以通过如下SQL语句实现：INSERT INTO newArticles SELECT \* FROM articles;

## 如果只希望导入指定字段，可以用这种方法：

**INSERT INTO 目标表 (字段1, 字段2, ...) SELECT 字段1, 字段2, ... FROM 来源表;**

请注意以上两表的字段必须一致（字段类型），否则会出现数据转换错误。

## 如何将一个mysql数据库中的一个表导入到另一个mysql数据库中。

（db1为原数据库，db2为要导出到的数据库，fromtable 是要导出的表名）

方法1：进入到db2；

use db2；

创建一个新表：**create table newtablename （as）select \* from db1.fromtable；**

插入数据：**insert into desttablename （as）select \* from db1.fromtable;**

方法2：

在Windows或Linux下，执行命令：

（1）**导出整个数据库**：

**mysqldump –u username –p password --database dbname > ‘/path/xxx.sql’；**

（2）**导出一个表格**：

**mysqldump -u root -p db1 fromtable file=d:/fromtable.sql;**

输入秘密，root为用户名

登录db2 执行 **source d:/fromtable.sql;**

**方法3：见8所示。**

# load 、infile、outfile

## select \* from deallist into outfile 'e:\\deallist.csv' fields terminated by ',' lines terminated by '\n';

## 之前由于没有指定字符集，总是导致错误。

**load** data local **infile** 'e: \\deallist.csv' into table deallist **character set UTF8** **fields terminated by ',' ENCLOSED BY '"' lines terminated by '\n'**;

导入一个文件的时候，指定字符集。**character set utf8**

# 关于导入文件大小限制问题的解决：

默认情况下：mysql 对导入文件大小有限制的，最大为2M，所以当文件很大时候，直接无法导入，下面就这个问题的解决列举如下：

**修改相关参数：**

影响mysql导入文件大小的参数有三个：

memory\_limit=128M,upload\_max\_filesize=2M,post\_max\_size=8M

修改upload\_max\_filesize=200 M 这里修改满足你需要的大小，

可以同时修改其他两项memory\_limit=250M post\_max\_size=200M

这样就可以导入200M以下的.sql文件了。