**[MySQL](http://lib.csdn.net/base/mysql" \o "MySQL知识库" \t "_blank)**[**数据库**](http://lib.csdn.net/base/mysql)**常用命令**

1、MySQL常用命令

create database name; 创建数据库

use databasename; 选择数据库

drop database name 直接删除数据库，不提醒

show tables; 显示表

describe tablename; 表的详细描述

select 中加上distinct去除重复字段

mysqladmin drop databasename 删除数据库前，有提示。

显示当前mysql版本和当前日期

select version(),current\_date;

2、修改mysql中root的密码：

shell>mysql -u root -p

mysql> update user set password=password(”xueok654123″) where user=’root’;

mysql> flush privileges //刷新数据库

mysql>use dbname； 打开数据库：

mysql>show databases; 显示所有数据库

mysql>show tables; 显示数据库mysql中所有的表：先use mysql；然后

mysql>describe user; 显示表mysql数据库中user表的列信息）；

3、grant

创建一个可以从任何地方连接服务器的一个完全的超级用户，但是必须使用一个口令something做这个

mysql> grant all privileges on \*.\* to user@localhost identified by ’something’ with

增加新用户

格式：grant select on 数据库.\* to 用户名@登录主机 identified by “密码”

GRANT ALL PRIVILEGES ON \*.\* TO monty@localhost IDENTIFIED BY ’something’ WITH GRANT OPTION;

GRANT ALL PRIVILEGES ON \*.\* TO monty@”%” IDENTIFIED BY ’something’ WITH GRANT OPTION;

删除授权：

mysql> revoke all privileges on \*.\* from root@”%”;

mysql> delete from user where user=”root” and host=”%”;

mysql> flush privileges;

创建一个用户custom在特定客户端it363.com登录，可访问特定数据库fangchandb

mysql >grant select, insert, update, delete, create,drop on fangchandb.\* to custom@ it363.com identified by ‘ passwd’

重命名表:

mysql > alter table t1 rename t2;

4、mysqldump

备份数据库

shell> mysqldump -h host -u root -p dbname >dbname\_backup.sql

恢复数据库

shell> mysqladmin -h myhost -u root -p create dbname

shell> mysqldump -h host -u root -p dbname < dbname\_backup.sql

如果只想卸出建表指令，则命令如下：

shell> mysqladmin -u root -p -d databasename > a.sql

如果只想卸出插入数据的sql命令，而不需要建表命令，则命令如下：

shell> mysqladmin -u root -p -t databasename > a.sql

那么如果我只想要数据，而不想要什么sql命令时，应该如何操作呢？

　　 mysqldump -T./ phptest driver

其中，只有指定了-T参数才可以卸出纯文本文件，表示卸出数据的目录，./表示当前目录，即与mysqldump同一目录。如果不指定driver 表，则将卸出整个数据库的数据。每个表会生成两个文件，一个为.sql文件，包含建表执行。另一个为.txt文件，只包含数据，且没有sql指令。

5、可将查询存储在一个文件中并告诉mysql从文件中读取查询而不是等待键盘输入。可利用外壳程序键入重定向实用程序来完成这项工作。

例如，如果在文件my\_file.sql 中存放有查

询，可如下执行这些查询：

例如，如果您想将建表语句提前写在sql.txt中:

mysql > mysql -h myhost -u root -p database < sql.txt

**转载声明：**本文转自<http://news.newhua.com/news1/program_database/2009/217/0921715343537K7H7IDI2CCI09JCI1DK8FJ4B07B3A04219G561C3JAB.html>

================================================================================

**转 mysql命令**

**一 . 安装与配置MYSQL   
　　二 . 常用mysql命令行命令**　　1 .mysql的启动与停止   
　　启动MYSQL服务 net start mysql   
　　停止MYSQL服务 net stop mysql

　　2 . netstat –na | findstr 3306 查看被监听的端口 , findstr用于查找后面的端口是否存在

　　3 . 在命令行中登陆MYSQL控制台 , 即使用 MYSQL COMMEND LINE TOOL   
　　 语法格式 mysql –user=root –password=123456 db\_name   
　　 或 mysql –u root –p123456 db\_name

　　4 . 进入MYSQL命令行工具后 , 使用status; 或/s 查看运行环境信息

　　5 . 切换连接数据库的语法 : use new\_dbname;   
　　 　   
　　6 . 显示所有数据库 : show databases;   
　　   
　　7 . 显示数据库中的所有表 : show tables;   
　　   
　　8 . 显示某个表创建时的全部信息 : show create table table\_name;   
　　   
　　9 . 查看表的具体属性信息及表中各字段的描述   
　　 Describe table\_name; 缩写形式 : desc table\_name;

　**三 。 MySql中的SQL语句**  
　　1 . 数据库创建 : Create database db\_name;   
　　数据库删除 : Drop database db\_name; 删除时可先判断是否存在，写成 : drop database if exits db\_name   
　　   
　　2 . 建表 : 创建数据表的语法 : create table table\_name (字段1 数据类型 , 字段2 数据类型);   
　　 例 : create table mytable (id int , username char(20));   
　　 删表 : drop table table\_name; 例 : drop table mytable;   
　　   
　　8 . 添加数据 : Insert into 表名 [(字段1 , 字段2 , ….)] values (值1 , 值2 , …..);   
　　如果向表中的每个字段都插入一个值,那么前面 [ ] 括号内字段名可写也可不写   
　　 例 : insert into mytable (id,username) values (1,’zhangsan’);   
　　   
　　9 . 查询 : 查询所有数据 : select \* from table\_name;   
　　查询指定字段的数据 : select 字段1 , 字段2 from table\_name;   
　　例 : select id,username from mytable where id=1 order by desc;多表查询语句------------参照第17条实例   
　　   
　　10 . 更新指定数据 , 更新某一个字段的数据（注意，不是更新字段的名字）   
　　Update table\_name set 字段名=’新值’ [, 字段2 =’新值’ , …..][where id=id\_num] [order by 字段 顺序]   
　　例 : update mytable set username=’lisi’ where id=1;   
　　Order语句是查询的顺序 , 如 : order by id desc(或asc) , 顺序有两种 : desc倒序(100—1,即从最新数据往后查询),asc(从1-100)，Where和order语句也可用于查询select 与删除delete   
　　   
　　11 . 删除表中的信息 :   
　　 删除整个表中的信息 : delete from table\_name;   
　　 删除表中指定条件的语句 : delete from table\_name where 条件语句 ; 条件语句如 : id=3;   
　　   
　　12 . 创建数据库用户   
　　一次可以创建多个数据库用户如：   
　　CREATE USER username1 identified BY ‘password’ , username2 IDENTIFIED BY ‘password’….   
　　   
　　13 . 用户的权限控制：grant   
　　 库，表级的权限控制 : 将某个库中的某个表的控制权赋予某个用户   
　　 Grant all ON db\_name.table\_name TO user\_name [ indentified by ‘password’ ];   
　　   
　　14 . 表结构的修改   
　　（1）增加一个字段格式：   
　　alter table table\_name add column (字段名 字段类型); ----此方法带括号   
　　（2）指定字段插入的位置：   
　　alter table table\_name add column 字段名 字段类型 after 某字段；   
　　删除一个字段：   
　　alter table table\_name drop字段名;   
　　（3）修改字段名称/类型   
　　alter table table\_name change 旧字段名 新字段名 新字段的类型;   
　　（4）改表的名字   
　　alter table table\_name rename to new\_table\_name;   
　　（5）一次性清空表中的所有数据   
　　truncate table table\_name; 此方法也会使表中的取号器(ID)从1开始   
　　   
　　15 . 增加主键，外键，约束，索引。。。。(使用方法见17实例)   
　　① 约束（主键Primary key、唯一性Unique、非空Not Null）   
　　② 自动增张 auto\_increment   
　　③外键Foreign key-----与reference table\_name(col\_name列名)配合使用，建表时单独使用   
　　④ 删除多个表中有关联的数据----设置foreign key 为set null ---具体设置参考帮助文档   
　　   
　　16 . 查看数据库当前引擎   
　　 SHOW CREATE TABLE table\_name;   
　　 修改数据库引擎   
　　 ALTER TABLE table\_name ENGINE=MyISAM | InnoDB;   
　　   
　　17 . SQL语句运用实例:   
　　--1 建users表   
　　create table users (id int primary key auto\_increment,nikename varchar(20) not null unique,password varchar(100) not null,address varchar(200), reg\_date timestamp not null default CURRENT\_TIMESTAMP);   
　　   
　　--2 建articles表,在建表时设置外键   
　　create table articles (id int primary key auto\_increment,content longtext not null,userid int,constraint foreign key (userid) references users(id) on delete set null);   
　　   
　　-----------------------------------------------------------------------   
　　--2.1 建articles表,建表时不设置外键   
　　 create table articles (id int primary key auto\_increment,content longtext not null,userid int);   
　　--2.2 给articles表设置外键   
　　 alter table articles add constraint foreign key (userid) references users(id) on delete set null;   
　　------------------------------------------------------------------------   
　　   
　　--3. 向users表中插入数据,同时插入多条   
　　insert into users (id,nikename,password,address) values (1,'lyh1','1234',null),(10,'lyh22','4321','湖北武汉'),(null,'lyh333','5678', '北京海淀');   
　　   
　　--4. 向article中插入三条数据   
　　insert into articles (id,content,userid) values (2,'hahahahahaha',11),(null,'xixixixixix',10),(13,'aiaiaiaiaiaiaiaiaiaiaiaia',1),(14,'hohoahaoaoooooooooo',10);   
　　   
　　--5. 进行多表查询，选择users表中ID=10的用户发布的所有留言及该用户的所有信息   
　　select articles.id,articles.content,users.\* from users,articles where users.id=10 and articles.userid=users.id order by articles.id desc;   
　　   
　　--6. 查看数据库引擎类型   
　　show create table users;   
　　   
　　--7. 修改数据库引擎类型   
　　alter table users engine=MyISAM; ---因为users表中ID被设置成外键，执行此句会出错   
　　   
　　--8. 同表查询,已知一个条件的情况下.查询ID号大于用户lyh1的ID号的所有用户   
　　select a.id,a.nikename,a.address from users a,users b where b.nikename='lyh1' and a.id>b.id;   
　　------也可写成   
　　select id,nikename,address from users where id>(select id from users where nikename='lyh1');   
　　   
　　9. 显示年龄比领导还大的员工：   
　　select a.name from users a,users b where a.managerid=b.id and a.age>b.age;   
　　   
　　查询编号为2的发帖人: 先查articles表,得到发帖人的编号,再根据编号查users得到的用户名。   
　　接着用关联查询.   
　　select \* from articles,users得到笛卡儿积,再加order by articles.id以便观察   
　　   
　　使用select \* from articles,users where articles.id=2 筛选出2号帖子与每个用户的组合记录   
　　   
　　再使用select \* from articles,users where articles.id=2 and articles.userid=users.id选出users.id等于2号帖的发帖人id的记录.   
　　   
　　只取用户名:select user where user.id=(select userid from articles where article.id =2)   
　　   
　　找出年龄比小王还大的人:假设小王是28岁,先想找出年龄大于28的人   
　　select \* from users where age>(select age from users where name='xiaowang');   
　　\*\*\*\*\*要查询的记录需要参照表里面的其他记录:   
　　select a.name from users a,users b where b.name='xiaowang' and a.age>b.age   
　　   
　　表里的每个用户都想pk一下.select a.nickname,b.nickname from users a,users b where a.id>b.id ;   
　　   
　　更保险的语句:select a.nickname,b.nickname from (select \* from users order by id) a,(se   
　　lect \* from users order by id) b where a.id>b.id ;   
　　   
　　再查询某个人发的所有帖子.   
　　select b.\* from articles a , articles b where a.id=2 and a.userid=b.userid   
　　   
　　说明: 表之间存在着关系，ER概念的解释，用access中的示例数据库演示表之间的关系.只有innodb引擎才支持foreign key，mysql的任何引擎目前都不支持check约束。   
　　四、字符集出现错误解决办法   
　　出现的问题：   
　　mysql> update users   
　　-> set username='关羽'   
　　-> where userid=2;   
　　ERROR 1366 (HY000): Incorrect string value: '/xB9/xD8/xD3/xF0' for column 'usern   
　　ame' at row 1   
　　向表中插入中文字符时，出现错误。   
　　   
　　mysql> select \* from users;   
　　+--------+----------+   
　　| userid | username |   
　　+--------+----------+   
　　| 2 | ???? |   
　　| 3 | ???? |   
　　| 4 | ?í?ù |   
　　+--------+----------+   
　　3 rows in set (0.00 sec)   
　　表中的中文字符位乱码。   
　　解决办法：   
　　使用命令：   
　　mysql> status;   
　　--------------   
　　mysql Ver 14.12 Distrib 5.0.45, for Win32 (ia32)   
　　   
　　Connection id: 8   
　　Current database: test   
　　Current user: root@localhost   
　　SSL: Not in use   
　　Using delimiter: ;   
　　Server version: 5.0.45-community-nt MySQL Community Edition (GPL)   
　　Protocol version: 10   
　　Connection: localhost via TCP/IP   
　　Server characterset: latin1   
　　Db characterset: latin1   
　　Client characterset: gbk   
　　Conn. characterset: gbk   
　　TCP port: 3306   
　　Uptime: 7 hours 39 min 19 sec   
　　Threads: 2 Questions: 174 Slow queries: 0 Opens: 57 Flush tables: 1 Open ta   
　　bles: 1 Queries per second avg: 0.006   
　　--------------   
　　查看mysql发现Server characterset，Db characterset的字符集设成了latin1，所以出现中文乱码。   
　　   
　　mysql> show tables;   
　　+----------------+   
　　| Tables\_in\_test |   
　　+----------------+   
　　| users |   
　　+----------------+   
　　1 row in set (0.00 sec)   
　　   
　　更改表的字符集。   
　　mysql> alter table users character set GBK;   
　　Query OK, 3 rows affected (0.08 sec)   
　　Records: 3 Duplicates: 0 Warnings: 0   
　　   
　　查看表的结构：   
　　mysql> show create users;   
　　ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that   
　　corresponds to your MySQL server version for the right syntax to use near 'users   
　　' at line 1   
　　mysql> show create table users;   
　　+-------+-----------------------------------------------------------------------   
　　------------------------------------------------------------------------------+   
　　| Table | Create Table   
　　|   
　　+-------+-----------------------------------------------------------------------   
　　------------------------------------------------------------------------------+   
　　| users | CREATE TABLE `users` (   
　　`userid` int(11) default NULL,   
　　`username` char(20) character set latin1 default NULL   
　　) ENGINE=InnoDB DEFAULT CHARSET=gbk |   
　　+-------+-----------------------------------------------------------------------   
　　------------------------------------------------------------------------------+   
　　1 row in set (0.00 sec)   
　　   
　　mysql> desc users;   
　　+----------+----------+------+-----+---------+-------+   
　　| Field | Type | Null | Key | Default | Extra |   
　　+----------+----------+------+-----+---------+-------+   
　　| userid | int(11) | YES | | NULL | |   
　　| username | char(20) | YES | | NULL | |   
　　+----------+----------+------+-----+---------+-------+   
　　2 rows in set (0.02 sec)   
　　   
　　这时向表中插入中文然后有错误。   
　　mysql> insert into users values(88,'中文');   
　　ERROR 1366 (HY000): Incorrect string value: '/xD6/xD0/xCE/xC4' for column 'usern   
　　ame' at row 1   
　　mysql> insert into users values(88,'中文');   
　　ERROR 1366 (HY000): Incorrect string value: '/xD6/xD0/xCE/xC4' for column 'usern   
　　ame' at row 1   
　　   
　　还要更改users表的username的字符集。   
　　mysql> alter table users modify username char(20) character set gbk;   
　　ERROR 1366 (HY000): Incorrect string value: '/xC0/xEE/xCB/xC4' for column 'usern   
　　ame' at row 1   
　　mysql> alter table users modify username char(20) character set gbk;   
　　ERROR 1366 (HY000): Incorrect string value: '/xC0/xEE/xCB/xC4' for column 'usern   
　　ame' at row 1   
　　   
　　因为表中已经有数据，所以更改username字符集的操作没有成\*\*\*   
　　清空users表中的数据   
　　mysql> truncate table users;   
　　Query OK, 3 rows affected (0.01 sec)   
　　   
　　从新更改user表中username的字符集   
　　mysql> alter table users modify username char(20) character set gbk;   
　　Query OK, 0 rows affected (0.06 sec)   
　　Records: 0 Duplicates: 0 Warnings: 0   
　　   
　　这时再插入中文字符，插入成\*\*\*。   
　　mysql> insert into users values(88,'中文');   
　　Query OK, 1 row affected (0.01 sec)   
　　   
　　mysql> select \* from users;   
　　+--------+----------+   
　　| userid | username |   
　　+--------+----------+   
　　| 88 | 中文 |   
　　+--------+----------+   
　　1 row in set (0.00 sec)   
　　mysql>

**转载声明：**本文转自<http://hi.baidu.com/zhjlabm/blog/item/b939fc3307a1d445ad4b5fbd.html>

================================================================================

**学习MySQL常用操作命令**

1、启动MySQL服务器

实际上上篇已讲到如何启动MySQL。两种方法： 一是用winmysqladmin，如果机器启动时已自动运行，则可直接进入下一步操作。 二是在DOS方式下运行 d:mysqlbinmysqld

2、进入mysql交互操作界面

在DOS方式下，运行： d:mysqlbinmysql

出现: mysql 的提示符，此时已进入mysql的交互操作方式。

如果出现 "ERROR 2003: Can′t connect to MySQL server on ′localhost′ (10061)“，

说明你的MySQL还没有启动。

3、退出MySQL操作界面

在mysql>提示符下输入quit可以随时退出交互操作界面：   
mysql> quit   
Bye   
你也可以用control-D退出。

4、第一条命令

mysql> select version(),current\_date();   
+----------------+-----------------+   
| version() | current\_date() |   
+----------------+-----------------+   
| 3.23.25a-debug | 2001-05-17 |   
+----------------+-----------------+   
1 row in set (0.01 sec)   
mysql>

此命令要求mysql服务器告诉你它的版本号和当前日期。尝试用不同大小写操作上述命令，看结果如何。结果说明mysql命令的大小写结果是一致的。

练习如下操作：

mysql>Select (20+5)\*4;   
mysql>Select (20+5)\*4,sin(pi()/3);   
mysql>Select (20+5)\*4 AS Result,sin(pi()/3); (AS: 指定假名为Result)

5、多行语句  
    一条命令可以分成多行输入，直到出现分号“；”为止：   
<ccid\_nobr>  
<table width="400" border="1" cellspacing="0" cellpadding="2"   
bordercolorlight = "black" bordercolordark = "#FFFFFF" align="center">  
<tr>  
<td bgcolor="e6e6e6" class="code" style="font-size:9pt">  
<pre><ccid\_code> mysql> select   
-> USER()   
-> ,   
-> now()   
->;   
+--------------------+---------------------+   
| USER() | now() |   
+--------------------+---------------------+   
| ODBC@localhost | 2001-05-17 22:59:15 |   
+--------------------+---------------------+   
1 row in set (0.06 sec)   
mysql>

注意中间的逗号和最后的分号的使用方法。

6、一行多命令

输入如下命令：

mysql> SELECT USER(); SELECT NOW();   
+------------------+   
| USER() |   
+------------------+   
| ODBC@localhost |   
+------------------+   
1 row in set (0.00 sec)

+---------------------+   
| NOW() |   
+---------------------+   
| 2001-05-17 23:06:15 |   
+---------------------+   
1 row in set (0.00 sec)   
mysql>

注意中间的分号，命令之间用分号隔开。

7、显示当前存在的数据库

mysql> show databases;   
+----------+   
| Database |   
+----------+   
| mysql |   
| test |   
+----------+   
2 row in set (0.06 sec)   
mysql>

8、选择数据库并显示当前选择的数据库

mysql> USE mysql   
Database changed   
mysql>   
(USE 和 QUIT 命令不需要分号结束。）   
mysql> select database();   
+---------------+   
| database() |   
+---------------+   
| mysql |   
+---------------+   
1 row in set (0.00 sec)

9、显示当前数据库中存在的表   
mysql> SHOW TABLES;

10、显示表(db)的内容   
mysql>select \* from db;

11、命令的取消

当命令输入错误而又无法改变（多行语句情形）时，只要在分号出现前就可以用 c来取消该条命令

mysql> select   
-> user()   
-> c   
mysql>

这是一些最常用的最基本的操作命令，通过多次练习就可以牢牢掌捂了

==========================================================================

**mysql命令**

[**测试**](http://lib.csdn.net/base/softwaretest)环境：mysql 5.0.45  
【注：可以在mysql中通过mysql> SELECT VERSION();来查看数据库版本】  
整理：leo

**一、连接MYSQL。**  
格式： mysql -h主机地址 -u用户名 －p用户密码

1、连接到本机上的MYSQL。  
首先打开DOS窗口，然后进入目录mysql/bin，再键入命令mysql -u root -p，回车后提示你输密码.注意用户名前可以有空格也可以没有空格，但是密码前必须没有空格，否则让你重新输入密码.  
如果刚安装好MYSQL，超级用户root是没有密码的，故直接回车即可进入到MYSQL中了，MYSQL的提示符是： mysql>  
2、连接到远程主机上的MYSQL。假设远程主机的IP为：110.110.110.110，用户名为root,密码为abcd123。则键入以下命令：  
mysql -h110.110.110.110 -u root -p 123; （注:u与root之间可以不用加空格，其它也一样）  
3、退出MYSQL命令： exit （回车）

**二、修改密码。**  
格式：mysqladmin -u用户名 -p旧密码 password 新密码  
1、给root加个密码ab12。首先在DOS下进入目录mysql/bin，然后键入以下命令  
mysqladmin -u root -password ab12  
注：因为开始时root没有密码，所以-p旧密码一项就可以省略了。  
2、再将root的密码改为djg345。  
mysqladmin -u root -p ab12 password djg345

**三、增加新用户。**  
（注意：和上面不同，下面的因为是MYSQL环境中的命令，所以后面都带一个分号作为命令结束符）  
格式：grant select on 数据库.\* to 用户名@登录主机 identified by “密码”  
1、增加一个用户test1密码为abc，让他可以在任何主机上登录，并对所有数据库有查询、插入、修改、删除的权限。首先用root用户连入MYSQL，然后键入以下命令：  
grant select,insert,update,delete on \*.\* to [email=test1@”%]test1@”%[/email]” Identified by “abc”;  
但增加的用户是十分危险的，你想如某个人知道test1的密码，那么他就可以在internet上的任何一台电脑上登录你的mysql数据库并对你的数据可以为所欲为了，解决办法见2。

2、增加一个用户test2密码为abc,让他只可以在localhost上登录，并可以对数据库mydb进行查询、插入、修改、删除的操作（localhost指本地主机，即MYSQL数据库所在的那台主机），  
这样用户即使用知道test2的密码，他也无法从internet上直接访问数据库，只能通过MYSQL主机上的web页来访问了。  
grant select,insert,update,delete on mydb.\* to [email=test2@localhost]test2@localhost[/email] identified by “abc”;  
如果你不想test2有密码，可以再打一个命令将密码消掉。  
grant select,insert,update,delete on mydb.\* to [email=test2@localhost]test2@localhost[/email] identified by “”;  
下篇我是MYSQL中有关数据库方面的操作。注意：你必须首先登录到MYSQL中，以下操作都是在MYSQL的提示符下进行的，而且每个命令以分号结束。

**一、操作技巧**  
1、如果你打命令时，回车后发现忘记加分号，你无须重打一遍命令，只要打个分号回车就可以了。  
也就是说你可以把一个完整的命令分成几行来打，完后用分号作结束标志就OK。  
2、你可以使用光标上下键调出以前的命令。

**二、显示命令**  
1、显示当前数据库服务器中的数据库列表：  
mysql> SHOW DATABASES;  
注意：mysql库里面有MYSQL的系统信息，我们改密码和新增用户，实际上就是用这个库进行操作。

2、显示数据库中的数据表：  
mysql> USE 库名；  
mysql> SHOW TABLES;

3、显示数据表的结构：  
mysql> DESCRIBE 表名;

4、建立数据库：  
mysql> CREATE DATABASE 库名;

5、建立数据表：  
mysql> USE 库名;  
mysql> CREATE TABLE 表名 (字段名 VARCHAR(20), 字段名 CHAR(1));

6、删除数据库：  
mysql> DROP DATABASE 库名;

7、删除数据表：  
mysql> DROP TABLE 表名；

8、将表中记录清空：  
mysql> DELETE FROM 表名;

9、显示表中的记录：  
mysql> SELECT \* FROM 表名;

10、往表中插入记录：  
mysql> INSERT INTO 表名 VALUES (”hyq”,”M”);

11、更新表中数据：  
mysql-> UPDATE 表名 SET 字段名1=’a',字段名2=’b’ WHERE 字段名3=’c';

12、用文本方式将数据装入数据表中：  
mysql> LOAD DATA LOCAL INFILE “D:/mysql.txt” INTO TABLE 表名;

13、导入.sql文件命令：  
mysql> USE 数据库名;  
mysql> SOURCE d:/mysql.sql;

14、命令行修改root密码：  
mysql> UPDATE mysql.user SET password=PASSWORD(’新密码’) WHERE User=’root’;  
mysql> FLUSH PRIVILEGES;

15、显示use的数据库名：  
mysql> SELECT DATABASE();

16、显示当前的user：  
mysql> SELECT USER();

**三、一个建库和建表以及插入数据的实例**  
drop database if exists school; //如果存在SCHOOL则删除  
create database school;           //建立库SCHOOL  
use school;   //打开库SCHOOL  
create table teacher //建立表TEACHER  
(  
id int(3) auto\_increment not null primary key,  
name char(10) not null,  
address varchar(50) default ‘深圳’,  
year date  
); //建表结束

//以下为插入字段  
insert into teacher values(”,’allen’,'大连一中’,'1976-10-10′);  
insert into teacher values(”,’jack’,'大连二中’,'1975-12-23′);

**如果你在mysql提示符键入上面的命令也可以，但不方便调试。**  
（1）你可以将以上命令原样写入一个文本文件中，假设为school.sql，然后复制到c://下，并在DOS状态进入目录[url=file:////mysql//bin]//mysql//bin[/url]，然后键入以下命令：  
mysql -uroot -p密码 < c://school.sql  
如果成功，空出一行无任何显示；如有错误，会有提示。（以上命令已经调试，你只要将//的注释去掉即可使用）。  
（2）或者进入命令行后使用 mysql> source c://school.sql; 也可以将school.sql文件导入数据库中。

**四、将文本数据转到数据库中**  
1、文本数据应符合的格式：字段数据之间用tab键隔开，null值用[url=file:////n]//n[/url]来代替.例：  
3 rose 大连二中 1976-10-10  
4 mike 大连一中 1975-12-23  
假设你把这两组数据存为school.txt文件，放在c盘根目录下。  
2、数据传入命令 load data local infile “c://school.txt” into table 表名;  
注意：你最好将文件复制到[url=file:////mysql//bin]//mysql//bin[/url]目录下，并且要先用use命令打表所在的库。

**五、备份数据库**1.导出整个数据库  
导出文件默认是存在mysql/bin目录下  
mysqldump -u 用户名 -p 数据库名 > 导出的文件名  
mysqldump -u user\_name -p123456 database\_name > outfile\_name.sql

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 user\_name -p database\_name table\_name > outfile\_name.sql

3.导出一个数据库结构  
mysqldump -u user\_name -p -d –add-drop-table database\_name > outfile\_name.sql  
-d 没有数据 –add-drop-table 在每个create语句之前增加一个drop table

4.带语言参数导出  
mysqldump -uroot -p –default-character-set=latin1 –set-charset=gbk –skip-opt database\_name > outfile\_name.sql

**六、导入数据库**

例如：数据库名为 dbTest

首先，进入mysql

mysql -uroot -p123456

然后，创建数据库

create dbTest

exit

最后，导入数据库

mysql -uroot -p123456 dbTest < dbTest\_bk.sql

**数据库及表导出导入示例：**

导出数据库  
mysqldump -u root -p123456 gameTop > gameTop\_db.sql  
  
导出数据库的表  
mysqldump -u -p123456 root gameTop gametop800 > gameTop\_table.sql  
  
导出数据库的特定表  
mysqldump -u root -p123456  gameTop --table gametop800  > gameTop\_table.sql  
  
  
  
导出数据库（只导结构，不要数据）  
mysqldump -u root -p123456 --opt -d gameTop > gameTop\_db.sql

**或者**

mysqldump -u root -p123456  -d gameTop > gameTop\_db.sql  
  
导出数据库（只导数据，不要结构）  
mysqldump -u root -p123456  -t gameTop > gameTop\_db.sql  
  
**注：**不加 -d 和 -t 则既导出结构，也导出数据

**mysqldump导出抛出异常：**

mysqldump: Got error: 1044: Access denied for user 'username'@'%' to database 'dbname' when using LOCK TABLES

解决办法，添加参数 **--skip-lock-tables**：

mysqldump **--skip-lock-tables** -h172.88.12.102 -username-pAnJnVs3C2tYXyTwV dbname> dbname\_bk.sql

--------------------------  
**导入数据库：**  
  
登录MySQL:      mysql -uroot -p123456  
创建数据库：    create database gameTop;

导入数据库：

mysql -uroot -p123456 gameTop < gameTop\_db.sql

导入数据库表：

mysql -uroot -p123456 gameTop  gametop800 <  gameTop\_table.sql

导入数据库表：  
mysql -uroot -p123456 gameTop  < gameTop\_table.sql   （不指定表名）

**转载声明：**本文转自<http://blog.csdn.net/networld2002/archive/2009/04/23/4103407.aspx>

==================================================================================

1:使用SHOW语句找出在服务器上当前存在什么数据库：  
mysql> SHOW DATABASES;

2:创建一个数据库MYSQLDATA  
mysql> CREATE DATABASE MYSQLDATA;

3:选择你所创建的数据库  
mysql> USE MYSQLDATA; (按回车键出现Database changed 时说明操作成功！)

4:查看现在的数据库中存在什么表  
mysql> SHOW TABLES;

5:创建一个数据库表  
mysql> CREATE TABLE MYTABLE (name VARCHAR(20), sex CHAR(1));

6:显示表的结构：  
mysql> DESCRIBE MYTABLE;

7:往表中加入记录  
mysql> insert into MYTABLE values (”hyq”,”M”);

8:用文本方式将数据装入数据库表中（例如D:/mysql.txt）  
mysql> LOAD DATA LOCAL INFILE “D:/mysql.txt” INTO TABLE MYTABLE;

9:导入.sql文件命令（例如D:/mysql.sql）  
mysql>use database;  
mysql>source d:/mysql.sql;

10:删除表  
mysql>drop TABLE MYTABLE;

11:清空表  
mysql>delete from MYTABLE;

12:更新表中数据  
mysql>update MYTABLE set sex=”f” where name=’hyq’;

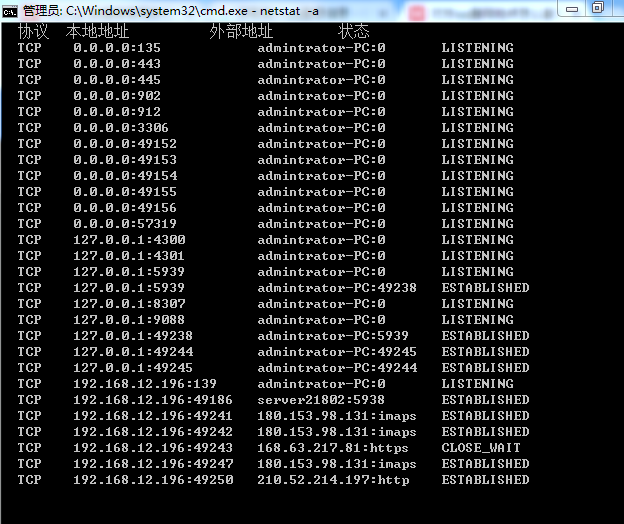
**以下是无意中在网络看到的使用MySql的管理心得,**  
在windows中MySql以服务形式存在，在使用前应确保此服务已经启动，未启动可用net start mysql命令启动。而[**Linux**](http://lib.csdn.net/base/linux)中启动时可用“/etc/rc.d/init.d/mysqld start”命令，注意启动者应具有管理员权限。  
刚安装好的MySql包含一个含空密码的root帐户和一个匿名帐户，这是很大的安全隐患，对于一些重要的应用我们应将安全性尽可能提高，在这里应把匿名帐户删除、 root帐户设置密码，可用如下命令进行：  
use mysql;  
delete from User where User=”";  
update User set Password=PASSWORD(’newpassword’) where User=’root’;

如果要对用户所用的登录终端进行限制，可以更新User表中相应用户的Host字段，在进行了以上更改后应重新启动数据库服务，此时登录时可用如下类似命令：  
mysql -uroot -p;  
mysql -uroot -pnewpassword;  
mysql mydb -uroot -p;  
mysql mydb -uroot -pnewpassword;

**上面命令参数是常用参数的一部分，详细情况可参考文档。此处的mydb是要登录的数据库的名称。**  
在 进行开发和实际应用中，用户不应该只用root用户进行连接数据库，虽然使用root用户进行测试时很方便，但会给系统带来重大安全隐患，也不利于管理技 术的提高。我们给一个应用中使用的用户赋予最恰当的数据库权限。如一个只进行数据插入的用户不应赋予其删除数据的权限。

MySql的用户管理是通过 User表来实现的，添加新用户常用的方法有两个，一是在User表插入相应的数据行，同时设置相应的权限；二是通过GRANT命令创建具有某种权限的用 户。其中GRANT的常用用法如下：  
grant all on mydb.\* to NewUserName@HostName identified by “password” ;  
grant usage on \*.\* to NewUserName@HostName identified by “password”;  
grant select,insert,update on mydb.\* to NewUserName@HostName identified by “password”;  
grant update,delete on mydb.TestTable to NewUserName@HostName identified by “password”;

若 要给此用户赋予他在相应对象上的权限的管理能力，可在GRANT后面添加WITH GRANT OPTION选项。而对于用插入User表添加的用户，Password字段应用PASSWORD 函数进行更新加密，以防不轨之人窃看密码。对于那些已经不用的用户应给予清除，权限过界的用户应及时回收权限，回收权限可以通过更新User表相应字段， 也可以使用REVOKE操作。  
下面给出本人从其它资料(www.cn-[**Java**](http://lib.csdn.net/base/javase).com)获得的对常用权限的解释：  
全局管理权限：  
FILE: 在MySQL服务器上读写文件。  
PROCESS: 显示或杀死属于其它用户的服务线程。  
RELOAD: 重载访问控制表，刷新日志等。  
SHUTDOWN: 关闭MySQL服务。  
数据库/数据表/数据列权限：  
ALTER: 修改已存在的数据表(例如增加/删除列)和索引。  
CREATE: 建立新的数据库或数据表。  
DELETE: 删除表的记录。  
DROP: 删除数据表或数据库。  
INDEX: 建立或删除索引。  
INSERT: 增加表的记录。  
SELECT: 显示/搜索表的记录。  
UPDATE: 修改表中已存在的记录。  
**特别的权限：**ALL: 允许做任何事(和root一样)。  
USAGE: 只允许登录–其它什么也不允许做。



下面的mysql命令是把select的mytable表中的数据导出到/home/db\_bak2012文件。

select \* from mytable where status!=**0** and name!='' into outfile '/home/db\_bak2012'

fields terminated by '|' enclosed by '"' lines terminated by '\r\n' ;

导入刚才备份的数据，可以使用load file方法，下面的mysql命令，把导出的数据导入了mytable\_bak的表中：

load data infile '/home/db\_bak2012' into table mytable\_bak

fields terminated by '|' enclosed by '"' lines terminated by '\r\n' ;

BACK TABLE tbl\_name TO '/tmp/db\_name/';

RESTORE TABLE FROM '/tmp/db\_name/';

Mysql 数据库 客户端手动事务控制，只操作当前客户端。

SHOW VARIABLES LIKE '%autocommit%';

SET @@autocommit=0;

SELECT @@tx\_isolation;

SET SESSION TX\_ISOLATION='READ-COMMITTED';

BEGIN;

DELETE FROM dm\_user\_role WHERE user\_id='518564';

COMMIT;