

## 1、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)

## 2、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

## 3、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 “”;
```

## 4.1 创建数据库

注意：创建数据库之前要先连接Mysql服务器

命令：create database <数据库名>

**例1：建立一个名为xhkdb的数据库**

```
mysql> create database xhkdb;
```

**例2：创建数据库并分配用户**

①CREATE DATABASE 数据库名;

②GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP,ALTER ON 数据库名.\* TO 数据库名@localhost IDENTIFIED BY '密码';

③SET PASSWORD FOR '数据库名'@'localhost' = OLD\_PASSWORD('密码');

依次执行3个命令完成数据库创建。注意：中文“密码”和“数据库”是户自己需要设置的。

## 4.2 显示数据库

命令：show databases （注意：最后有个s）

```
mysql> show databases;
```

注意：为了不再显示的时候乱码，要修改数据库默认编码。以下以GBK编码页面为例进行说明：

**1、修改MYSQL的配置文件：**my.ini里面修改default-character-set=gbk

**2、代码运行时修改：**

①Java代码：jdbc:mysql://localhost:3306/test?useUnicode=true&characterEncoding=gbk

②PHP代码：header("Content-Type:text/html;charset=gb2312");

③C语言代码：int mysql\_set\_character\_set( MYSQL \* mysql, char \* csname);

该函数用于为当前连接设置默认的字符集。字符串csname指定了1个有效的字符集名称。连接校对成为字符集的默认校对。该函数的工作方式与SET NAMES语句类似，但它还能设置mysql- > charset的值，从而影响了由mysql\_real\_escape\_string() 设置的字符集。

## 4.3 删除数据库

命令：drop database <数据库名>

例如：删除名为 xhkdb的数据库

```
mysql> drop database xhkdb;
```

例子1：删除一个已经确定存在的数据库

```
mysql> drop database drop_database;
```

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

例子2: 删除一个不确定存在的数据库

```
mysql> drop database drop_database;
```

```
ERROR 1008 (HY000): Can't drop database 'drop_database'; database doesn't exist
```

//发生错误, 不能删除'drop\_database'数据库, 该数据库不存在。

```
mysql> drop database if exists drop_database;
```

```
Query OK, 0 rows affected, 1 warning (0.00 sec)//产生一个警告说明此数据库不存在
```

```
mysql> create database drop_database;
```

```
Query OK, 1 row affected (0.00 sec)
```

```
mysql> drop database if exists drop_database;//if exists 判断数据库是否存在, 不存在也不产生错误
```

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

## 4.4 连接数据库

命令: use <数据库名>

例如: 如果xhkdb数据库存在, 尝试存取它:

```
mysql> use xhkdb;
```

屏幕提示: Database changed

**use 语句可以通告MySQL把db\_name数据库作为默认（当前）数据库使用, 用于后续语句。该数据库保持为默认数据库, 直到语段的结尾, 或者直到发布一个不同的USE语句:**

```
mysql> USE db1;
```

```
mysql> SELECT COUNT(*) FROM mytable; # selects from db1.mytable
```

```
mysql> USE db2;
```

```
mysql> SELECT COUNT(*) FROM mytable; # selects from db2.mytable
```

**使用USE语句为一个特定的当前的数据库做标记, 不会阻碍您访问其它数据库中的表。下面的例子可以从db1数据库访问作者表, 并从db2数据库访问编辑表:**

```
mysql> USE db1;
```

```
mysql> SELECT author_name,editor_name FROM author,db2.editor
```

```
-> WHERE author.editor_id = db2.editor.editor_id;
```

USE语句被设立出来, 用于与Sybase相兼容。

有些网友问到，连接以后怎么退出。其实，不用退出来，use 数据库后，使用show databases就能查询所有数据库，如果想跳到其他数据库，用

```
use 其他数据库名字
```

就可以了。

## 4.5 当前选择的数据库

命令：mysql> select database();

MySQL中SELECT命令类似于其他编程语言里的print或者write，你可以用它来显示一个字符串、数字、数学表达式的结果等等。如何使用MySQL中SELECT命令的特殊功能？

### 1.显示MYSQL的版本

mysql> select version();

```
+-----+
| version()          |
+-----+
| 6.0.4-alpha-community |
+-----+
1 row in set (0.02 sec)
```

### 2. 显示当前时间

mysql> select now();

```
+-----+
| now()          |
+-----+
| 2009-09-15 22:35:32 |
+-----+
1 row in set (0.04 sec)
```

### 3. 显示年月日

SELECT DAYOFMONTH(CURRENT\_DATE);

```
+-----+
```

```
| DAYOFMONTH(CURRENT_DATE) |
```

```
+-----+
```

```
|          15 |
```

```
+-----+
```

```
1 row in set (0.01 sec)
```

```
SELECT MONTH(CURRENT_DATE);
```

```
+-----+
```

```
| MONTH(CURRENT_DATE) |
```

```
+-----+
```

```
|          9 |
```

```
+-----+
```

```
1 row in set (0.00 sec)
```

```
SELECT YEAR(CURRENT_DATE);
```

```
+-----+
```

```
| YEAR(CURRENT_DATE) |
```

```
+-----+
```

```
|       2009 |
```

```
+-----+
```

```
1 row in set (0.00 sec)
```

#### 4. 显示字符串

```
mysql> SELECT "welecome to my blog!";
```

```
+-----+
```

```
| welecome to my blog! |
```

```
+-----+
```

```
| welecome to my blog! |
```

```
+-----+
```

```
1 row in set (0.00 sec)
```

#### 5. 当计算器用

```
select ((4 * 4) / 10 ) + 25;
```

```

+-----+
| ((4 * 4) / 10 ) + 25 |
+-----+
|          26.60 |
+-----+
1 row in set (0.00 sec)

```

## 6. 串接字符串

```

select CONCAT(f_name, " ", l_name)
AS Name
from employee_data
where title = 'Marketing Executive';

```

```

+-----+
| Name      |
+-----+
| Monica Sehgal |
| Hal Simlai   |
| Joseph Irvine |
+-----+
3 rows in set (0.00 sec)

```

注意：这里用到CONCAT()函数，用来把字符串串接起来。另外，我们还用到以前学到的AS给结果列'CONCAT(f\_name, " ", l\_name)'起了个假名。

## 5.1 创建数据表

命令：create table <表名> ( <字段名1> <类型1> [...<字段名n> <类型n>]);

例如，建立一个名为MyClass的表，

字段名	数字类型	数据宽度	是否为空	是否主键	自动增加	默认值
id	int	4	否	primary key	auto_increment	
name	char	20	否			
sex	int	4	否			0
degree	double	16	是			

```
mysql> create table MyClass(  
> id int(4) not null primary key auto_increment,  
> name char(20) not null,  
> sex int(4) not null default '0',  
> degree double(16,2));
```

## 5.3 删除数据表

命令：drop table <表名>

例如：删除表名为 MyClass 的表

```
mysql> drop table MyClass;
```

DROP TABLE用于取消一个或多个表。您必须有每个表的DROP权限。所有的表数据和表定义会被取消，所以使用本语句要小心！

注意：对于一个带分区的表，DROP TABLE会永久性地取消表定义，取消各分区，并取消储存在这些分区中的所有数据。DROP TABLE还会取消与被取消的表有关联的分区定义（.par）文件。

对与不存在的表，使用IF EXISTS用于防止错误发生。当使用IF EXISTS时，对于每个不存在的表，会生成一个NOTE。

RESTRICT和CASCADE可以使分区更容易。目前，RESTRICT和CASCADE不起作用。

## 5.4 表插入数据

命令：insert into <表名> [( <字段名1>[,...<字段名n > ])] values ( 值1 ), ( 值n )

例如：往表 MyClass中插入二条记录, 这二条记录表示：编号为1的名为Tom的成绩为96.45, 编号为2的名为Joan 的成绩为82.99, 编号为3的名为Wang 的成绩为96.5。

```
mysql> insert into MyClass values(1,'Tom',96.45),(2,'Joan',82.99), (2,'Wang', 96.59);
```

注意：insert into每次只能向表中插入一条记录。

## 5.5 查询表中的数据



## 1)、查询所有行

命令：select <字段1, 字段2, ...> from <表名> where <表达式>

例如：查看表 MyClass 中所有数据

```
mysql> select * from MyClass;
```

## 2)、查询前几行数据

例如：查看表 MyClass 中前2行数据

```
mysql> select * from MyClass order by id limit 0,2;
```

select一般配合where使用，以查询更精确更复杂的数据。

# 5.6 删除表中数据

命令：delete from 表名 where 表达式

例如：删除表 MyClass中编号为1 的记录

```
mysql> delete from MyClass where id=1;
```

下面是一个删除数据前后表的对比。

FirstName	LastName	Age
Peter	Griffin	35
Glenn	Quagmire	33

下面以PHP代码为例删除 "Persons" 表中所有 LastName='Griffin' 的记录：

```
<?php
```

```
    $con = mysql_connect("localhost","peter","abc123");
```

```
    if (!$con)
```

```
    {
```

```
        die('Could not connect: ' . mysql_error());
```

```
    }
```

```
mysql_select_db("my_db", $con);  
  
mysql_query("DELETE FROM Persons WHERE LastName='Griffin'"); mysql_close($con);  
?>
```

在这次删除之后，表是这样的：

FirstName	LastName	Age
Glenn	Quagmire	33

## 5.7 修改表中数据

语法：update 表名 set 字段=新值,... where 条件

```
mysql> update MyClass set name='Mary' where id=1;
```

**例子1：单表的MySQL UPDATE语句：**

```
UPDATE [LOW_PRIORITY] [IGNORE] tbl_name SET col_name1=expr1 [, col_name2=expr2  
...] [WHERE where_definition] [ORDER BY ...] [LIMIT row_count]
```

**例子2：多表的UPDATE语句：**

```
UPDATE [LOW_PRIORITY] [IGNORE] table_references SET col_name1=expr1 [,  
col_name2=expr2 ...] [WHERE where_definition]
```

UPDATE语法可以用新值更新原有表行中的各列。SET子句指示要修改哪些列和要给予哪些值。WHERE子句指定应更新哪些行。如果没有WHERE子句，则更新所有的行。如果指定了ORDER BY子句，则按照被指定的顺序对行进行更新。LIMIT子句用于给定一个限值，限制可以被更新的行的数目。

## 5.8 增加字段

命令：**alter table** 表名 **add**字段 类型 其他;

例如：在表MyClass中添加了一个字段passtest，类型为int(4)，默认值为0

```
mysql> alter table MyClass add passtest int(4) default '0'
```

**加索引**

mysql> alter table 表名 add index 索引名 (字段名1[, 字段名2 ...]);

例子: mysql> alter table employee add index emp\_name (name);

### 加主关键字的索引

mysql> alter table 表名 add primary key (字段名);

例子: mysql> alter table employee add primary key(id);

### 加唯一限制条件的索引

mysql> alter table 表名 add unique 索引名 (字段名);

例子: mysql> alter table employee add unique emp\_name2(cardnumber);

### 删除某个索引

mysql> alter table 表名 drop index 索引名;

例子: mysql> alter table employee drop index emp\_name;

### 增加字段:

mysql> ALTER TABLE table\_name ADD field\_name field\_type;

### 修改原字段名称及类型:

mysql> ALTER TABLE table\_name CHANGE old\_field\_name new\_field\_name field\_type;

### 删除字段:

MySQL ALTER TABLE table\_name DROP field\_name;

## 5.9 修改表名

命令: rename table 原表名 to 新表名;

例如: 在表MyClass名字更改为YouClass

mysql> rename table MyClass to YouClass;

当你执行 RENAME 时, 你不能有任何锁定的表或活动的事务。你同样也必须有对原初表的 ALTER 和 DROP 权限, 以及对新表的 CREATE 和 INSERT 权限。

如果在多表更名中，MySQL 遭遇到任何错误，它将对所有被更名的表进行倒退更名，将每件事物退回到最初状态。

RENAME TABLE 在 MySQL 3.23.23 中被加入。

## 6、备份数据库

命令在DOS的[url=file://\mysql\bin]\mysql\bin[/url]目录下执行

### 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
```

例如，将aaa库备份到文件back\_aaa中：

```
[root@test1 root]# cd /home/data/mysql
```

```
[root@test1 mysql]# mysqldump -u root -p --opt aaa > back_aaa
```

## 7.1 一个建库和建表的实例1

```
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。

## 7.2 一个建库和建表的实例2

```
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("", "glchengang", "深圳一中", "1976-10-10");
```

```
insert into teacher values("", "jack", "深圳一中", "1975-12-23");
```

注：在建表中

1、将ID设为长度为3的数字字段:int(3); 并让它每个记录自动加一:auto\_increment; 并不能为空:not null; 而且让他成为主字段 primary key。

2、将NAME设为长度为10的字符字段

3、将ADDRESS设为长度50的字符字段，而且缺省值为深圳。

4、将YEAR设为日期字段。

来源： <http://www.cnblogs.com/zhangzhu/archive/2013/07/04/3172486.html>

---