

- **MySql 入门教程**

- **一、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。

首先在打开 DOS 窗口, 然后进入目录 `mysqlbin`, 再键入命令 `mysql -uroot -p`, 回车后提示你输密码, 如果刚安装好 MYSQL, 终极用户 `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、例 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 -uroot -pabcd123
```

(注: `u` 和 `root` 能不用加空格, 其他也相同)

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、例 1: 给 `root` 加个密码 `ab12`。首先在 DOS 下进入目录 `mysqlbin`, 然后键入以下命令

```
mysqladmin -uroot -password ab12
```

注: 因为开始时 `root` 没有密码, 所以 `-p` 旧密码一项就能省略了。

2、例 2: 再将 `root` 的密码改为 `djg345`。

```
mysqladmin -uroot -pab12 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"`

例 1、增加一个用户 `test1` 密码为 `abc`, 让他能在所有主机上登录, 并对所有数据库有查询、插入、修改、删除的权限。首先用以 `root` 用户连入 MYSQL, 然后键入以下命令:

```
grant select,insert,update,delete on *.* to test1@"%" identified by "abc";
```

但例 1 增加的用户是十分危险的, 你想如某个人知道 `test1` 的密码, 那么他就能在

internet 上的所有一台计算机上登录你的 mysql 数据库并对你的数据能有所为所欲为了，解决办法见例 2。

例 2、增加一个用户 test2 密码为 abc,让他只能在 localhost 上登录,并能对数据库 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 test2@localhost identified by "abc";
```

如果你不想 test2 有密码，能再打一个命令将密码消掉。

```
grant select,insert,update,delete on mydb.* to test2@localhost identified by "";
```

下面来看看 MYSQL 中有关数据库方面的操作。注意：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。

又一篇安装的文章

PHP+MySQL+Linux 目前已逐渐成为小型 web 服务器的一种经典组合。在 indows 环境下构筑和调试 MySQL 数据库是许多网站研发者的一种最佳选择。本人在视窗系统 98 环境下初学 MySQL，现将学习过程和经验总结出来供大家参考。

1、下载 mysql-3.23.35-win.zip 并解压；

2、运行 setup.exe;选择 d:\mysql,"typical install"

3、启动 mysql，有如下方法：

方法一：使用 winmysqladmin

1)、进入 d:\mysql\bin 目录，运行 winmysqladmin.exe，在屏幕右下角的任务栏内会有一个带红色的图符

2)、鼠标左键点击该图符，选择“show me”，出现“WinMySQLAdmin”操作界面；首次运行时中间会出现一个对话框需求输入并设置你的用户名和口令

3)、选择“My.INI setup”

4)、在“mysqld file”中选择“mysqld-opt”(win9x)或“mysqld-nt”(winNT)

5)、选择“Pick-up or Edit my.ini values”能在右边窗口内对你的 my.ini 文件进行编辑

6)、选择“Save Modification”保存你的 my.ini 文件

7)、如果你想快速使用 winmysqladmin(开机时自动运行)，选择“Create ShortCut on Start Menu”

8)、测试：

进入 DOS 界面；

在 d:\mysql\bin 目录下运行 mysql，进入 mysql 交互操作界面

输入 show databases 并回车，屏幕显示出当前已有的两个数据库 mysql 和 test

方法二：不使用 winmysqladmin

1)、在 DOS 窗口下，进入 d:/mysql/bin 目录

2)、win9X 下)运行:

mysqld

在 NT 下运行:

mysqld-nt --standalone

3)、此后，mysql 在后台运行

4)、测试 mysql: (在 d:/mysql/bin 目录下)

a)、mysqlshow

正常时显示已有的两个数据库 mysql 和 test

b)、mysqlshow -u root mysql

正常时显示数据库 mysql 里的五个表:

columns_priv

db

host

tables_priv

user

c)、mysqladmin version status proc

显示版本号、状态、进程信息等

d)、mysql test

进入 mysql 操作界面，当前数据库为 test

5)、mysql 关闭方法:

mysqladmin -u root shutdown

4、至此，MySQL 已成功安装，接着能熟悉 MySQL 的常用命令并创建自己的数据库了。

上篇讲了怎么安装并测试 MySQL，环境建好后就能继续我们的学习了。本篇主要熟悉一写常用命令。

1、启动 MySQL 服务器

实际上上篇已讲到怎么启动 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 能随时退出交互操作界面:

```
mysql> quit
```

Bye

你也能用 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>
```

此命令需求 mysql 服务器告诉你他的版本号和当前日期。尝试用不同大小写操作上述命令，看结果怎么。

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

一条命令能分成多行输入，直到出现分号“；”为止：

```
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 命令不必分号结束。)

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动（多行语句情形）时，只要在分号出现前就能用 **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，通过多次练习就能牢牢掌捂了。

了解了一些最基本的操作命令后，我们再来学习怎么创建一个数据库和数据库表。

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;  
+-----+  
| Database |  
+-----+  
| mysql|  
| test |  
+-----+  
3 rows in set (0.00 sec)
```

2、创建一个数据库 **abccs**

```
mysql> CREATE DATABASE abccs;
```

注意不同操作系统对大小写的敏感。

3、选择你所创建的数据库

```
mysql> USE abccs  
Database changed
```

此时你已进入你刚才所建立的数据库 **abccs**。

4、 创建一个数据库表

首先看目前你的数据库中是否存在什么表：

```
mysql> SHOW TABLES;  
Empty set (0.00 sec)
```

说明刚才建立的数据库中还没有数据库表。下面来创建一个数据库表 **mytable**：

我们要建立一个你公司员工的生日表，表的内容包含员工姓名、性别、出生日期、出生城市。

```
mysql> CREATE TABLE mytable (name VARCHAR(20), sex CHAR(1),  
-> birth DATE, birthaddr VARCHAR(20));  
Query OK, 0 rows affected (0.00 sec)
```

由于 name、birthadd 的列值是变化的，因此选择 VARCHAR，其长度不一定是 20。能选择从

1 到 255 的所有长度，如果以后需要改动他的字长，能使用 ALTER TABLE 语句。）；

性别只需一个字符就能表示："m"或"f"，因此选用 CHAR(1)；

birth 列则使用 DATE 数据类型。

创建了一个表后，我们能看看刚才做的结果，用 SHOW TABLES 显示数据库中有哪些表：

```
mysql> SHOW TABLES;
```

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

```
| Tables in menagerie |
```

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

```
| mytables|
```

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

5、显示表的结构：

```
mysql> DESCRIBE mytable;
```

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

```
| Field | Type | Null | Key | Default | Extra |
```

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

```
| name | varchar(20) | YES | | NULL | |
```

```
| sex | char(1) | YES | | NULL | |
```

```
| birth | date | YES | | NULL | |
```

```
| deathaddr | varchar(20) | YES | | NULL | |
```

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

```
4 rows in set (0.00 sec)
```

6、 往表中加入记录

我们先用 SELECT 命令来查看表中的数据：

```
mysql> select * from mytable;
```

```
Empty set (0.00 sec)
```

这说明刚才创建的表还没有记录。

加入一条新记录：

```
mysql> insert into mytable
```

```
-> values ('abccs','f','1977-07-07','china');
```

```
Query OK, 1 row affected (0.05 sec)
```

再用上面的 SELECT 命令看看发生了什么变化。

我们能按此方法一条一条地将所有员工的记录加入到表中。

7、用文本方式将数据装入一个数据库表

如果一条一条地输入，非常麻烦。我们能用文本文件的方式将所有记录加入你的数据库表中。

创建一个文本文件“mysql.txt”，每行包含一个记录，用定位符(tab)把值分开，并且以在

CREATE TABLE 语句中列出的列次序给出，例如：

```
abccs f 1977-07-07 china
mary f 1978-12-12 usa
tom m 1970-09-02 usa
```

使用下面命令将文本文件“mytable.txt”装载到 mytable 表中:

```
mysql> LOAD DATA LOCAL INFILE "mytable.txt" INTO TABLE pet;
```

再使用如下命令看看是否已将数据输入到数据库表中:

```
mysql> select * from mytable;
```

上篇我们学会了怎么创建一个数据库和数据库表,并知道怎么向数据库表中添加记录。
那么我们怎么从数据库表中检索数据呢?

1、从数据库表中检索信息

实际上,前面我们已用到了 SELECT 语句,他用来从数据库表中检索信息。

select 语句格式一般为:

SELECT 检索关键词 FROM 被检索的表 WHERE 检索条件(可选)

以前所使用的“*”表示选择所有的列。

下面继续使用我们在上篇文章中创建的表 mytable:

2、查询所有数据:

```
mysql> select * from mytable;
+-----+-----+-----+-----+
| name | sex | birth | birthaddr |
+-----+-----+-----+-----+
| abccs|f| 1977-07-07 | china |
| mary |f| 1978-12-12 | usa |
| tom  |m| 1970-09-02 | usa |
+-----+-----+-----+-----+
3 row in set (0.00 sec)
```

3、修正错误记录:

如果 tom 的出生日期有错误,应该是 1973—09—02,则能用 update 语句来修正:

```
mysql> update mytable set birth = "1973-09-02" where name = "tom";
```

再用 2 中的语句看看是否已更正过来。

4、选择特定行

上面修改了 tom 的出生日期,我们能选择 tom 这一行来看看是否已有了变化:

```
mysql> select * from mytable where name = "tom";
```



```
+-----+-----+-----+-----+
| name | sex | birth | birthaddr |
+-----+-----+-----+-----+
| tom | m | 1973-09-02 | usa |
+-----+-----+-----+-----+
1 row in set (0.06 sec)
```

上面 WHERE 的参数指定了检索条件。我们还能用组合条件来进行查询：

```
mysql> SELECT * FROM mytable WHERE sex = "f" AND birthaddr = "china";
```

```
+-----+-----+-----+-----+
| name | sex | birth | birthaddr |
+-----+-----+-----+-----+
| abccs | f | 1977-07-07 | china |
+-----+-----+-----+-----+
1 row in set (0.06 sec)
```

5、 选择特定列

如果你想查看表中的所有人的姓名，则能这样操作：

```
mysql> SELECT name FROM mytable;
```

```
+-----+
| name |
+-----+
| abccs |
| mary |
| tom |
+-----+
```

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

如果想列出姓名和性别两列，则能用逗号将关键词 name 和 birth 分开：

```
mysql> select name,birth from mytable;
```

6、对行进行排序

我们能对表中的记录按生日大小进行排序：

```
mysql> SELECT name, birth FROM mytable ORDER BY birth;
```

```
+-----+-----+
| name | birth |
+-----+-----+
| tom | 1973-09-02 |
| abccs | 1977-07-07 |
| mary | 1978-12-12 |
+-----+-----+
3 row in set (0.00 sec)
```

我们能用 DESC 来进行逆序排序：

```
mysql> SELECT name, birth FROM mytable ORDER BY birth DESC;
```

```
+-----+-----+
| name | birth |
+-----+-----+
| mary | 1978-12-12 |
| abccs| 1977-07-07 |
| tom  | 1973-09-02 |
+-----+-----+
3 row in set (0.00 sec)
```

7、 行计数

数据库经常要统计一些数据，如表中员工的数目，我们就要用到行计数函数 **COUNT()**。
COUNT()函数用于对非 **NULL** 结果的记录进行计数：

```
mysql> SELECT COUNT(*) FROM mytable;
```

```
+-----+
| COUNT(*) |
+-----+
| 3 |
+-----+
1 row in set (0.06 sec)
```

员工中男女数量：

```
mysql> SELECT sex, COUNT(*) FROM mytable GROUP BY sex;
```

```
+-----+-----+
| sex | COUNT(*) |
+-----+-----+
| f|2 |
| m|1 |
+-----+-----+
2 row in set (0.00 sec)
```

注意我们使用了 **GROUP BY** 对 **SEX** 进行了分组。

前面我们熟悉了数据库和数据库表的基本操作，目前我们再来看看怎么操作多个表。

在一个数据库中，可能存在多个表，这些表都是相互关联的。我们继续使用前面的例子。前面建立的表中包含了员工的一些基本信息，如姓名、性别、出生日期、出生地。我们再创建一个表，该表用于描述员工所发表的文章，内容包括作者姓名、文章标题、发表日期。

1、查看第一个表 **mytable** 的内容：

```
mysql> select * from mytable;
```

```
+-----+-----+-----+-----+
| name | sex | birth | birthaddr |
```

```
+-----+-----+-----+-----+
| abccs|f | 1977-07-07 | china |
| mary |f | 1978-12-12 | usa |
| tom |m | 1970-09-02 | usa |
+-----+-----+-----+-----+
```

2、创建第二个表 **title**（包括作者、文章标题、发表日期）：

```
mysql> create table title(writer varchar(20) not null,
-> title varchar(40) not null,
-> senddate date);
```

向该表中填加记录，最后表的内容如下：

```
mysql> select * from title;
+-----+-----+-----+
| writer | title | senddate |
+-----+-----+-----+
| abccs | a1| 2000-01-23 |
| mary | b1| 1998-03-21 |
| abccs | a2| 2000-12-04 |
| tom| c1| 1992-05-16 |
| tom| c2| 1999-12-12 |
+-----+-----+-----+
5 rows in set (0.00sec)
```

3、多表查询

目前我们有了两个表: **mytable** 和 **title**。利用这两个表我们能进行组合查询：

例如我们要查询作者 **abccs** 的姓名、性别、文章：

```
mysql> SELECT name,sex,title FROM mytable,title
-> WHERE name=writer AND name='abccs';
+-----+-----+-----+
| name | sex | title |
+-----+-----+-----+
| abccs | f| a1|
| abccs | f| a2|
+-----+-----+-----+
```

上面例子中，由于作者姓名、性别、文章记录在两个不同表内，因此必须使用组合来进行查询。必须要指定一个表中的记录怎么和其他表中的记录进行匹配。

注意：如果第二个表 **title** 中的 **writer** 列也取名为 **name**（和 **mytable** 表中的 **name** 列相同）而不是 **writer** 时，就必须用 **mytable.name** 和 **title.name** 表示，以示差别。

再举一个例子，用于查询文章 **a2** 的作者、出生地和出生日期：

```
mysql> select title,writer,birthaddr,birth from mytable,title
-> where mytable.name=title.writer and title='a2';
```

```
+-----+-----+-----+-----+
| title | writer | birthaddr | birth |
+-----+-----+-----+-----+
| a2| abccs | china | 1977-07-07 |
+-----+-----+-----+-----+
```

有时我们要对数据库表和数据库进行修改和删除，能用如下方法实现：

1、增加一列：

如在前面例子中的 **mytable** 表中增加一列表示是否单身 **single**：

```
mysql> alter table mytable add column single char(1);
```

2、修改记录

将 **abccs** 的 **single** 记录修改为“y”：

```
mysql> update mytable set single='y' where name='abccs';
```

目前来看看发生了什么：

```
mysql> select * from mytable;
+-----+-----+-----+-----+-----+
| name | sex | birth | birthaddr | single |
+-----+-----+-----+-----+-----+
| abccs|f | 1977-07-07 | china | y |
| mary |f | 1978-12-12 | usa | NULL |
| tom  |m | 1970-09-02 | usa | NULL |
+-----+-----+-----+-----+-----+
```

3、增加记录

前面已讲过怎么增加一条记录，为便于查看，重复和此：

```
mysql> insert into mytable
-> values ('abc','f','1966-08-17','china','n');
```

Query OK, 1 row affected (0.05 sec)

查看一下：

```
mysql> select * from mytable;
+-----+-----+-----+-----+-----+
| name | sex | birth | birthaddr | single |
+-----+-----+-----+-----+-----+
| abccs|f | 1977-07-07 | china | y |
| mary |f | 1978-12-12 | usa | NULL |
| tom  |m | 1970-09-02 | usa | NULL |
| abc  |f | 1966-08-17 | china | n |
+-----+-----+-----+-----+-----+
```

3、删除记录

用如下命令删除表中的一条记录：

```
mysql> delete from mytable where name='abc';
```

DELETE 从表中删除满足由 where 给出的条件的一条记录。

再显示一下结果：

```
mysql> select * from mytable;
```

```
+-----+-----+-----+-----+-----+
| name | sex | birth | birthaddr | single |
+-----+-----+-----+-----+-----+
| abccs|f | 1977-07-07 | china | y |
| mary |f | 1978-12-12 | usa | NULL |
| tom  |m | 1970-09-02 | usa | NULL |
+-----+-----+-----+-----+-----+
```

4、删除表：

```
mysql> drop table ****(表 1 的名字), ***表 2 的名字;
```

能删除一个或多个表，小心使用。

5、数据库的删除：

```
mysql> drop database 数据库名;
```

小心使用。

6、数据库的备份：

退回到 DOS：

```
mysql> quit
```

```
d:mysqlbin
```

使用如下命令对数据库 abccs 进行备份：

```
mysqldump --opt abccs>abccs.dbb
```

abccs.dbb 就是你的数据库 abccs 的备份文件。

7、用批处理方式使用 MySQL：

首先建立一个批处理文件 mytest.sql,内容如下：

```
use abccs;
```

```
select * from mytable;
```

```
select name,sex from mytable where name='abccs';
```

在 DOS 下运行如下命令：

```
d:mysqlbin mysql < mytest.sql
```

在屏幕上会显示执行结果。

如果想看结果，而输出结果非常多，则能用这样的命令：

```
mysql < mytest.sql | more
```

我们还能将结果输出到一个文件中：

```
mysql < mytest.sql > mytest.out
```