MySQL 常用命令大全

1. mysql：连接数据库

mysql 命令用户连接数据库。

mysql 命令格式： 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. mysqladmin：修改用户密码

mysqladmin 命令用于修改用户密码。

mysqladmin 命令格式：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. grant on：新增用户

grant on 命令用于增加新用户并控制其权限。

grant on 命令格式：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，解决办法如下。

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. create：创建数据库

create 命令用于创建数据库。

create 命令格式：create database <数据库名>;

注意：创建数据库之前要先连接 Mysql 服务器。

1) 建立一个名为 xhkdb 的数据库：

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

2) 创建数据库并分配用户：

a：CREATE DATABASE 数据库名;

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

c:SET 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。

5. show databases：显示数据库

show databases 命令用于显示所有数据库。

show databases 命令格式：show databases; （注意：最后有个 s）

例如：mysql> show databases;

6. drop database：删除数据库

drop 命令用于删除数据库。
drop 命令格式：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;   // 创建一个数据库
 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)
```

7. use：使用数据库

use 命令可以让我们来使用数据库。
use 命令格式： 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
1) 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，或者直到出现下一个不同的 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
```

2) 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 author 表，并从 db2 数据库访问 editor 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。

8. select：当前连接的数据库

select 命令表示当前选择（连接）的数据库。
select 命令格式：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 命令的特殊功能呢？
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。

9. create table：创建数据表

数据表属于数据库，在创建数据表之前，应该使用语句“USE <数据库名>”指定操作是在哪个数据库中进行，如果没有选择数据库，会抛出“No database selected”的错误。

创建数据表的语句为 CREATE TABLE，语法规则如下：

```
CREATE   TABLE <表名>
(
    字段名 1，数据类型 [列级别约束条件] [默认值],
    字段名 2，数据类型 [列级别约束条件] [默认值],
    ……
    [表级别约束条件]
);
```

使用 CREATE TABLE 创建表时，必须指定以下信息：

（1）要创建的表的名称，不区分大小写，不能使用 SQL 语言中的关键字，如 DROP、ALTER、INSERT 等。

（2）数据表中每一个列（字段）的名称和数据类型，如果创建多个列，要用逗号隔开。

创建员工表 tb_emp1，结构如下表所示。

表　tb_emp1 表结构

字段名称	数据类型	备注
id	INT(11)	员工编号
name	VARCHAR(25)	员工名称
deptId	INT(11)	所在部门编号
salary	FLOAT	工资

首先创建数据库，SQL 语句如下：

```
CREATE   DATABASE test_db;
```

选择创建表的数据库，SQL 语句如下：

```
USE test_db;
```

创建 tb_emp1 表，SQL 语句为：

```
CREATE TABLE tb_emp1
(
```

```
    id          INT(11),
    name        VARCHAR(25),
    deptId      INT(11),
    salary      FLOAT
);
```

语句执行后，便创建了一个名称为 tb_emp1 的数据表，使用 SHOW TABLES；语句查看数据表是否创建成功，SQL 语句如下：

```
mysql> SHOW TABLES;
+----------------------+
| Tables_in_ test_db |
+----------------------+
| tb_emp1              |
+----------------------+
1 row in set (0.00 sec)
```

可以看到，test_db 数据库中已经有了数据表 tb_tmp1，数据表创建成功。

10. desc：获取表结构

在 MySQL 中，查看表结构可以使用 DESCRIBE 和 SHOW CREATE TABLE 语句。

DESCRIBE/DESC 语句可以查看表的字段信息，其中包括：字段名、字段数据类型、是否为主键、是否有默认值等。语法规则如下：

```
DESCRIBE 表名;
```

或者简写为：

```
DESC 表名;
```

SHOW CREATE TABLE 语句可以用来显示创建表时的 CREATE TABLE 语句，语法格式如下：

```
SHOW CREATE TABLE <表名\G>;
```

使用 SHOW CREATE TABLE 语句，不仅可以查看表创建时候的详细语句，而且还可以查看存储引擎和字符编码。

如果不加'\G'参数，显示的结果可能非常混乱，加上参数'\G'之后，可使显示结果更加直观，易于查看。

使用 SHOW CREATE TABLE 查看表 tb_emp1 的详细信息，SQL 语句如下：

```
mysql> SHOW CREATE TABLE tb_emp1;
```

11. drop table：删除数据表

在 MySQL 中，使用 DROP TABLE 可以一次删除一个或多个没有被其他表关联的数据表。语法格式如下：

```
DROP TABLE [IF EXISTS]表 1, 表 2, . . . 表 n;
```

其中“表 n”指要删除的表的名称，后面可以同时删除多个表，只需将要删除的表名依次写在后面，相互之间用逗号隔开即可。如果要删除的数据表不存在，则 MySQL 会提示一条错误信息，“ERROR 1051 (42S02): Unknown table '表名'”。参数“IF EXISTS”用于在删除前判断删除的表是否存在，加上该参数后，再删除表的时候，如果表不存在，SQL 语句可以顺利执行，但是会发出警告（warning）。

在前面的例子中，已经创建了名为 tb_dept2 的数据表。如果没有，读者可输入语句，创建该表，SQL 语句如例 4.8 所示。下面使用删除语句将该表删除。

删除数据表 tb_dept2，SQL 语句如下：

```
DROP TABLE IF EXISTS tb_dept2;
```

12. insert into：向表中插入数据

INSERT INTO 语句用于向表格中插入新的行。

语法如下：

```
INSERT INTO 表名称 VALUES (值 1, 值 2,....)
```

我们也可以指定所要插入数据的列：

```
INSERT INTO table_name (列 1, 列 2,...) VALUES (值 1, 值 2,....)
```

【例】创建数据表 tmp3，定义数据类型为 YEAR 的字段 y，向表中插入值 2010，’2010’，SQL 语句如下：

首先创建表 tmp3：

```
CREATE TABLE tmp3(  y YEAR );
```

向表中插入数据：

```
mysql> INSERT INTO tmp3 values(2010),('2010');
```

13. select from：查询表中数据

MySQL 从数据表中查询数据的基本语句为 SELECT 语句。SELECT 语句的基本格式是：

```
SELECT
        {* | <字段列表>}
        [
            FROM <表 1>,<表 2>...
            [WHERE <表达式>
            [GROUP BY <group by definition>]
            [HAVING <expression> [{<operator> <expression>}...]]
            [ORDER BY <order by definition>]
            [LIMIT [<offset>,] <row count>]
        ]
SELECT [字段 1,字段 2,…,字段 n]
FROM [表或视图]
WHERE [查询条件];
```

其中，各条子句的含义如下：

{* | <字段列表>}包含星号通配符选字段列表，表示查询的字段，其中字段列至少包含一个字段名称，如果要查询多个字段，多个字段之间用逗号隔开，最后一个字段后不要加逗号。

FROM <表 1>,<表 2>...，表 1 和表 2 表示查询数据的来源，可以是单个或者多个。

WHERE 子句是可选项，如果选择该项，将限定查询行必须满足的查询条件。

GROUP BY <字段>，该子句告诉 MySQL 如何显示查询出来的数据，并按照指定的字段分组。

[ORDER BY <字段 >]，该子句告诉 MySQL 按什么样的顺序显示查询出来的数据，可以进行的排序有：升序(ASC)、降序（DESC）。

[LIMIT [<offset>,] <row count>]，该子句告诉 MySQL 每次显示查询出来的数据条数。

14. delete from：删除记录

从数据表中删除数据使用 DELETE 语句，DELETE 语句允许 WHERE 子句指定删除条件。DELETE 语句基本语法格式如下：

```
DELETE FROM table_name [WHERE <condition>];
```

table_name 指定要执行删除操作的表；“[WHERE <condition>]”为可选参数，指定删除条件，如果没有 WHERE 子句，DELETE 语句将删除表中的所有记录。

【例】在 person 表中，删除 id 等于 11 的记录，SQL 语句如下：

```
mysql> DELETE FROM person WHERE id = 11;
Query OK, 1 row affected (0.02 sec)
```

15. update set：修改表中的数据

MySQL 中使用 UPDATE 语句更新表中的记录，可以更新特定的行或者同时更新所有的行。基本语法结构如下：

```
UPDATE table_name
SET column_name1 = value1,column_name2=value2,……,column_namen=valuen
WHERE (condition);
```

column_name1,column_name2,……,column_namen 为指定更新的字段的名称；value1, value2,……valuen 为相对应的指定字段的更新值；condition 指定更新的记录需要满足的条件。更新多个列时，每个“列-值”对之间用逗号隔开，最后一列之后不需要逗号。

【例】在 person 表中，更新 id 值为 11 的记录，将 age 字段值改为 15，将 name 字段值改为 LiMing，SQL 语句如下：

```
UPDATE person SET age = 15, name='LiMing' WHERE id = 11;
```

16. alter add：增加字段

添加字段的语法格式如下：

```
ALTER TABLE <表名> ADD <新字段名> <数据类型>
    [约束条件] [FIRST | AFTER 已存在字段名];
```

新字段名为需要添加的字段的名称；“FIRST”为可选参数，其作用是将新添加的字段设置为表的第一个字段；“AFTER”为可选参数，其作用是将新添加的字段添加到指定的“已存在字段名”的后面。

【例】在数据表 tb_dept1 中添加一个没有完整性约束的 INT 类型的字段 managerId（部门经理编号），SQL 语句如下：

```
ALTER TABLE tb_dept1 ADD managerId INT(10);
```

17. rename：修改表名

MySQL 是通过 ALTER TABLE 语句来实现表名的修改的，具体的语法规则如下：

```
ALTER TABLE <旧表名> RENAME [TO] <新表名>;
```

其中 TO 为可选参数，使用与否均不影响结果。

【例】将数据表 tb_dept3 改名为 tb_deptment3，SQL 语句如下：

```
ALTER TABLE tb_dept3 RENAME tb_deptment3;
```

18. mysqldump：备份数据库

mysqldump 备份数据库语句的基本语法格式如下：

```
mysqldump   –u user –h host –ppassword dbname[tbname, [tbname...]]> filename.sql
```

user 表示用户名称；host 表示登录用户的主机名称；password 为登录密码；dbname 为需要备份的数据库名称；tbname 为 dbname 数据库中需要备份的数据表，可以指定多个需要备份的表；右箭头符号“>”告诉 mysqldump 将备份数据表的定义和数据写入备份文件；filename.sql 为备份文件

的名称。

【例】使用 mysqldump 命令备份数据库中的所有表，执行过程如下：

打开操作系统命令行输入窗口，输入备份命令如下：

```
C:\ >mysqldump -u root -p booksdb > C:/backup/booksdb_20130301.sql
Enter password: **
```

输入密码之后，MySQL 便对数据库进行了备份，在 C:\backup 文件夹下面查看刚才备份过的文件。

19. mysql 和 source：还原数据库

对于已经备份的包含 CREATE、INSERT 语句的文本文件，可以使用 mysql 命令导入到数据库中。

备份的 sql 文件中包含 CREATE、INSERT 语句（有时也会有 DROP 语句）。mysql 命令可以直接执行文件中的这些语句。其语法如下：

```
mysql –u user –p [dbname] < filename.sql
```

user 是执行 backup.sql 中语句的用户名；-p 表示输入用户密码；dbname 是数据库名。如果 filename.sql 文件为 mysqldump 工具创建的包含创建数据库语句的文件，执行的时候不需要指定数据库名。

【例 1】使用 mysql 命令将 C:\backup\booksdb_20130301.sql 文件中的备份导入到数据库中，输入语句如下：

```
mysql –u root –p booksDB < C:/backup/booksdb_20130301.sql
```

执行该语句前，必须先在 MySQL 服务器中创建 booksDB 数据库，如果不存在恢复过程将会出错。命令执行成功之后 booksdb_20130301.sql 文件中的语句就会在指定的数据库中恢复以前的表。

如果已经登录 MySQL 服务器，还可以使用 source 命令导入 sql 文件。source 语句语法如下：

```
source filename
```

【例 2】使用 root 用户登录到服务器，然后使用 source 导入本地的备份文件 booksdb_20110101.sql，输入语句如下：

```
--选择要恢复到的数据库
mysql> use booksDB;
Database changed
--使用 source 命令导入备份文件
mysql> source C:\backup\booksDB_20130301.sql
```

命令执行后，会列出备份文件 booksDB_20130301.sql 中每一条语句的执行结果。source 命令执行成功后，booksDB_20130301.sql 中的语句会全部导入到现有数据库中。

20. mysqlhotcopy：快速恢复数据库

mysqlhotcopy 备份后的文件也可以用来恢复数据库，在 MySQL 服务器停止运行时，将备份的数据库文件复制到 MySQL 存放数据的位置（MySQL 的 data 文件夹），重新启动 MySQL 服务即可。如果以根用户执行该操作，必须指定数据库文件的所有者，输入语句如下：

```
chown -R mysql.mysql /var/lib/mysql/dbname
```

【例】从 mysqlhotcopy 复制的备份恢复数据库，输入语句如下：

```
cp -R  /usr/backup/test usr/local/mysql/data
```

执行完该语句，重启服务器，MySQL 将恢复到备份状态。