

第19章 学会使用简单的MySQL操作

学习Linux请加QQ群： 群1(163262181) 群2(148412746) 群3(246401509) 群4(173884211)

跟阿铭学Linux邀请函 (<http://www.aminglinux.com>)，猿课已上线，请加微信aminglinux84索要配套视频教程。

在前面两个章节中已经介绍过MySQL的安装了，但是光会安装还不够，你还需要会一些基本的相关操作。当然了，关于MySQL的内容也是非常多的，只不过对于linux系统管理员来讲，一些基本的操作已经可以应付日常的管理工作了，至于更高深的那是DBA（专门管理数据库的技术人员）的事情了。

更改mysql数据库root的密码

首次进入数据库是不用密码的：

```
[root@localhost ~]# /usr/local/mysql/bin/mysql -uroot
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.1.40-log MySQL Community Server (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

退出的话，直接输入quit或者exit即可。细心的读者也许会发现，阿铭在上一条命令中，使用的是绝对路径，这样不方便，但是单独只是输入一个“mysql”命令是不行的，因为“/usr/local/mysql/bin”没有 在 PATH 这个环境变量里。如何把它加入环境变量PATH中？之前阿铭介绍过：

```
[root@localhost ~]# PATH=$PATH:/usr/local/mysql/bin
```

这样就可以了，但重启Linux后还会失效，所以需要让它开机加载：

```
[root@localhost ~]# echo "PATH=$PATH:/usr/local/mysql/bin" >> /etc/profile
[root@localhost ~]# source /etc/profile
[root@localhost ~]# mysql -uroot
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.1.40-log MySQL Community Server (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

阿铭再来解释一下上一条命令 -u 的含义，它用来指定要登录的用户，后边可以有空格，也可以无空格，root 用户是mysql自带的管理员账户，默认没有密码的，那么如何给root用户设定密码？按如下操作：

```
[root@localhost ~]# mysqladmin -uroot password 'yourpassword'
```

这样就设置了‘root’账号的密码了，不妨再来用上面的命令登陆一下试试看：

```
[root@localhost ~]# mysql -uroot
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password:NO)
```

报错了，这是在提示我们，root账号是需要密码登陆的。

```
[root@localhost ~]# mysql -uroot -p'yourpassword'
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 7
Server version: 5.1.40-log MySQL Community Server (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

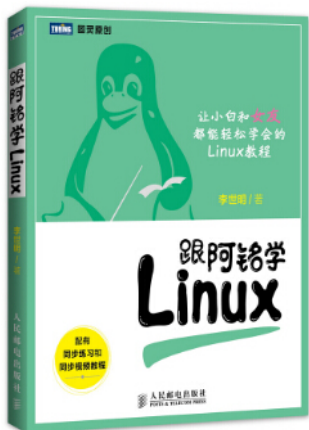
需要加一个 -p 选项，它后面可以直接跟密码，后面不可以有空格，不过密码最好用单引号括起来，不括也可以，但是密码中如果有特殊字符就会有问题了，所以最好是括起来吧。当然， -p后面也是可以不加密码，而是和用户交互的方式，让我们输入密码：

```
[root@localhost ~]# mysql -uroot -p
Enter password:
```

目录列表

第1章 前言
第2章 关于Linux的历史
第3章 对Linux系统管理员的建议
第4章 安装Linux操作系统
第5章 初步认识Linux
第6章 Linux系统的远程登陆
第7章 Linux文件与目录管理
第8章 Linux系统用户及用户组管理
第9章 Linux磁盘管理
第10章 文本编辑工具vim
第11章 文档的压缩与打包
第12章 安装RPM包或者安装源码包
第13章 学习 shell脚本之前的基础知识
第14章 正则表达式
第15章 shell脚本
第16章 linux系统日常管理
第17章 LAMP环境搭建
第18章 LNMP环境搭建
第19章 学会使用简单的MySQL操作
第20章 NFS服务配置
第21章 配置FTP服务
第22章 配置Squid服务
第23章 配置Tomcat
第24章 配置Samba服务器
第25章 MySQL replication(主从)配置
结语

阿铭著作：



微信扫码获取最新版linux电子书和视频

刚刚讲过通过使用 `mysql -u root -p` 就可以连接数据库了，但这只是连接的本地的数据库 “localhost”，可是有很多时候都是去连接网络中的某一个主机上的mysql。

```
[root@localhost ~]# mysql -uroot -p -h192.168.137.10 -P3306
Enter password:
```

其中后边的 -P(大写) 用来指定远程主机mysql的绑定端口，默认都是3306, -h 用来指定远程主机的IP.

一些基本的MySQL操作命令

1. 查询当前的库

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| test |
+-----+
3 rows in set (0.06 sec)
```

mysql的命令，结尾处需要加一个分号。

2. 查询某个库的表

首先需要切换到某个库里去：

```
mysql> use mysql;
Database changed
```

然后再把表列出来：

```
mysql> show tables;
+-----+
| Tables_in_mysql |
+-----+
| columns_priv |
| db |
| event |
| func |
| general_log |
| help_category |
| help_keyword |
| help_relation |
| help_topic |
| host |
| ndb_binlog_index |
| plugin |
| proc |
| procs_priv |
| servers |
| slow_log |
| tables_priv |
| time_zone |
| time_zone_leap_second |
| time_zone_name |
| time_zone_transition |
| time_zone_transition_type |
| user |
+-----+
23 rows in set (0.06 sec)
```

3. 查看某个表的全部字段

```
mysql> desc slow_log;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| start_time | timestamp | NO | | CURRENT_TIMESTAMP | on update CURRENT_TIMESTAMP |
| user_host | mediumtext | NO | | NULL | |
| query_time | time | NO | | NULL | |
| lock_time | time | NO | | NULL | |
| rows_sent | int(11) | NO | | NULL | |
```

Go

Enter search terms or a module, class or function name.

rows_examined	int(11)	NO		NULL
db	varchar(512)	NO		NULL
last_insert_id	int(11)	NO		NULL
insert_id	int(11)	NO		NULL
server_id	int(10) unsigned	NO		NULL
sql_text	mediumtext	NO		NULL

11 rows in set (0.04 sec)

也可以使用两一条命令，显示比这个更详细，而且可以把建表语句全部列出来：

```
mysql> show create table slow_log\G;
***** 1. row *****

      Table: slow_log
Create Table: CREATE TABLE `slow_log` (
  `start_time` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
  `user_host` mediumtext NOT NULL,
  `query_time` time NOT NULL,
  `lock_time` time NOT NULL,
  `rows_sent` int(11) NOT NULL,
  `rows_examined` int(11) NOT NULL,
  `db` varchar(512) NOT NULL,
  `last_insert_id` int(11) NOT NULL,
  `insert_id` int(11) NOT NULL,
  `server_id` int(10) unsigned NOT NULL,
  `sql_text` mediumtext NOT NULL
) ENGINE=CSV DEFAULT CHARSET=utf8 COMMENT='Slow log'
1 row in set (0.01 sec)
```

4. 查看当前是哪个用户

```
mysql> select user();
+-----+
| user() |
+-----+
| root@localhost |
+-----+
1 row in set (0.00 sec)
```

5. 查看当前所使用数据库

```
mysql> select database();
+-----+
| database() |
+-----+
| mysql      |
+-----+
1 row in set (0.01 sec)
```

6. 创建一个新库

```
mysql> create database db1;
Query OK, 1 row affected (0.05 sec)
```

7. 创建一个新表

```
mysql> use db1;
Database changed
mysql> create table t1 (`id` int(4), `name` char(40));
Query OK, 0 rows affected (0.02 sec)
```

要注意的是，字段名需要用反引号括起来。

8. 查看当前数据库版本

```
mysql> select version();
+-----+
| version() |
+-----+
| 5.1.40-log |
+-----+
1 row in set (0.01 sec)
```

9. 查看当前mysql状态

```
mysql> show status;
+-----+
```

Variable_name	Value	
Aborted_clients	0	
Aborted_connects	5	
Binlog_cache_disk_use	0	
Binlog_cache_use	0	
Bytes_received	303	
Bytes_sent	7001	

由于内容太长，阿铭没有全部列出来，如果有兴趣可以网上找资料查一下每一行的含义。

10. 查看mysql的参数

```
mysql> show variables;
```

Variable_name	Value	
auto_increment_increment	1	
auto_increment_offset	1	
autocommit	ON	
automatic_sp_privileges	ON	
back_log	50	
basedir	/usr/local/mysql/	

限于篇幅，阿铭省略了很多参数没有显示，其中很多参数都是可以在/etc/my.cnf中定义的，并且有部分参数是可以在线编辑的。

11. 修改mysql的参数

```
mysql> show variables like 'max_connect%';
```

Variable_name	Value	
max_connect_errors	10	
max_connections	151	

2 rows in set (0.00 sec)

```
mysql> set global max_connect_errors = 1000;
```

Query OK, 0 rows affected (0.01 sec)

```
mysql> show variables like 'max_connect_errors';
```

Variable_name	Value	
max_connect_errors	1000	

1 row in set (0.01 sec)

在mysql命令行，“%”类似于shell下的 `*`，表示万能匹配。使用“set global”可以临时修改某些参数，但是重启mysqld服务后还会变为原来的，所以要想恒久生效，需要在配置文件 my.cnf 中定义。

12. 查看当前mysql服务器的队列

这个在日常的管理工作中使用最为频繁，因为使用它可以查看当前mysql在干什么，可以发现是否有锁表：

```
mysql> show processlist;
```

Id	User	Host	db	Command	Time	State	Info
13	root	localhost	db1	Query	0	NULL	show processlist

1 row in set (0.01 sec)

13. 创建一个普通用户并授权

```
mysql> grant all on *.* to user1 identified by '123456';
```

Query OK, 0 rows affected (0.01 sec)

all 表示所有的权限（读、写、查询、删除等等操作），*.* 前面的 * 表示所有的数据库，后面的 * 表示所有的表，identified by 后面跟密码，用单引号括起来。这里的user1指的是localhost上的user1，如果是给网络上的其他机器上的某个用户授权则这样：

```
mysql> grant all on db1.* to 'user2'@'10.0.2.100' identified by '111222';
```

Query OK, 0 rows affected (0.01 sec)

用户和主机的IP之间有一个@，另外主机IP那里可以用%替代，表示所有主机，例如：

```
mysql> grant all on db1.* to 'user3'@'%' identified by '231222';
```

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

一些常用的sql

1. 查询语句

```
mysql> select count(*) from mysql.user;
+-----+
| count(*) |
+-----+
|          8 |
+-----+
1 row in set (0.00 sec)
```

mysql.user表示mysql库的user表；count(*)表示表中共有多少行。

```
mysql> select * from mysql.db;
```

这个用来表示查询mysql库的db表中的所有数据，也可以查询单个字段或者多个字段：

```
mysql> select db from mysql.db;
mysql> select db,user from mysql.db;
```

同样，在查询语句中可以使用万能匹配 “%”

```
mysql> select * from mysql.db where host like '10.0.%';
```

2. 插入一行

```
mysql> insert into db1.t1 values (1, 'abc');
Query OK, 1 row affected (0.02 sec)
```

```
mysql> select * from db1.t1;
+-----+-----+
| id    | name |
+-----+-----+
| 1     | abc  |
+-----+-----+
1 row in set (0.00 sec)
```

3. 更改表的某一行

```
mysql> update db1.t1 set name='aaa' where id=1;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql> select * from db1.t1;
+-----+-----+
| id    | name |
+-----+-----+
| 1     | aaa  |
+-----+-----+
1 row in set (0.00 sec)
```

4. 清空表数据

```
mysql> truncate table db1.t1;
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> select count(*) from db1.t1;
+-----+
| count(*) |
+-----+
|          0 |
+-----+
1 row in set (0.00 sec)
```

5. 删除表

```
mysql> drop table db1.t1;
Query OK, 0 rows affected (0.00 sec)
```

6. 删除数据库

```
mysql> drop database db1;
Query OK, 0 rows affected (0.02 sec)
```

mysql数据库的备份与恢复

备份:

```
[root@localhost ~]# mysqldump -uroot -p'yourpassword' mysql >/tmp/mysql.sql
```

使用 `mysqldump` 命令备份数据库, `-u` 和 `-p` 两个选项使用方法和前面说的 `mysql` 同样, 而后面的“mysql”指的是库名, 然后重定向到一个文本文档里。备份完后, 你可以查看 `/tmp/mysql.sql` 这个文件里的内容。

恢复和备份正好相反:

```
[root@localhost ~]# mysql -uroot -p'yourpassword' mysql </tmp/mysql.sql
```

关于MySQL的基本操作阿铭就介绍这么多, 当然学会了这些还远远不够, 希望你能够在工作中学习到更多的知识, 如果你对MySQL有很大兴趣, 不妨深入研究一下, 毕竟多学点总没有坏处。如果想学跟多的东西请去查看MySQL官方中文参考手册(5.1) <http://dev.mysql.com/doc/refman/5.1/zh/index.html>

阿铭建议你最好再扩展学习一下: <http://www.aminglinux.com/bbs/thread-5443-1-1.html>

教程答疑: [请移步这里](#).

欢迎你加入 [阿铭学院](#) 和阿铭一起学习Linux, 让阿铭成为你Linux生涯中永远的朋友吧!

[PREVIOUS](#) | [NEXT](#) | [INDEX](#)

© Copyright 2013, lishiming.net. Created using [Sphinx 1.3b1](#)[网站统计](#).