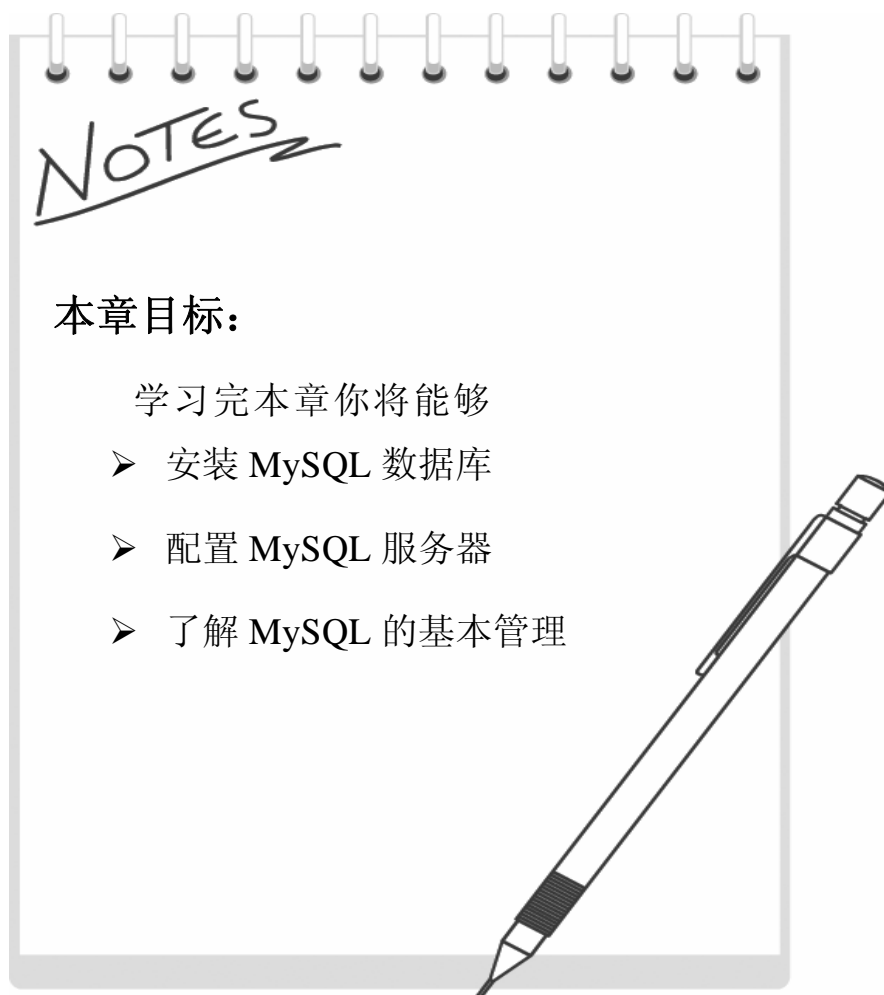


## 第九章 MySQL 的安装与配置

### 本章导读

MySQL 是目前使用最多的开放源代码的免费的数据库软件。无论是在 Linux 平台，还是在 Windows 平台，都有很多中小企业用它来存储和管理公司的数据。要使用 MySQL 数据库，首先要安装和配置它。本章将讲解 MySQL 的安装与配置。



# 1. MySQL 简介

MySQL 是最流行的开放源码 SQL 数据库管理系统，它是由 MySQL AB 公司开发、发布并支持的。MySQL AB 是由多名 MySQL 开发人创办的一家商业公司。它是一家第二代开放源码公司，结合了开放源码价值取向、方法和成功的商业模型。

在 MySQL 的网站 (<http://www.mysql.com/>) 上，给出了关于 MySQL 和 MySQL 的最新信息。

## 1.1 MySQL 的主要特性

MySQL 虽然是免费的，但同 Oracle, Sybase, Informix, Db2 等商业数据库一样，具有数据库系统的通用性：

1) 数据库管理系统。我们知道，所谓的数据库就是一些结构化的数据的联合体，要提供对这些数据的存取、增加、修改、删除或更加复杂的数据抽取等操作，需要有一个支撑系统，这就是数据库管理系统 (DBMS)，MySQL 完全具有这方面的功能。

2) 关系型数据库管理系统。在数据库的发展历程中，曾出现过多种不同形式的数据库系统，但关系型数据库管理系统 (RDBMS) 以其优越性而被广为采用，象现在几种广泛使用的数据库全为关系型数据库。同样，MySQL 也是关系型的数据库系统，支持标准的结构化查询语言 (Structured Query Language)。

3) 开放源码数据库。同商业性的数据库相比，这是 MySQL 最大的特点。MySQL 的源码是公开的，这就意味着任何人，只要遵守 GPL 的规则都可以对 MySQL 的源码使用、修改以符合自己特殊的需求。

4) 技术特点。MySQL 是 C/S 架构的服务器，服务器端是多线程的，为客户端提供了不同的程序接口和链接库，如 C、C++、Java、Perl、PHP、Tcl 等，也提供了简单的管理工具，如 mysqladmin, mysql 等。

正因为 MySQL 的源码是公开的，使得我们有机会从源码级，更深入的了解 MySQL。从这一层面上可以探求、了解 MySQL 数据库的特性：

- 1) 开发语言为 C/C++，支持多种编译器，如 gcc, cc, xlc, aCC 等
- 2) 良好的内存管理机制，尤其是内存泄漏的管理，通过了 Purify 的严格测试
- 3) 支持多种平台，如 AIX、Solaris、HP、FreeBSD、Tru64、SGI、Windows 等
- 4) 提供多种形式的 API，为我们开发工具的选择提供了便利

5) 使用多线程的技术,可充分发挥系统的特点,避免在 SMP 系统中出现仅使用单 CPU 的现象

6) 对磁盘表的管理采用 B 树加密及索引的技术,为我们快速访问数据提供了可能

7) 高性能的内存申请技术

8) 临时表采用在内存中以哈希表实现

9) 提供多种数据类型,如 signed/unsigned integer, double, blob 等

10) 支持定长的和变长的纪录

11) 所有的列都有缺省值,为我们的某些“粗心”提供了方便

12) 灵活、安全的权限和密码系统,密码在传输中加密传送,允许主机端验证密码

13) 客户端可以通过 TCP/IP、Unix 套接字、命名管道 (NT) 连接到 MySQL 数据库服务器

14) 提供 Windows 端的 ODBC 连接

15) 对多种字符集的完全支持

MySQL 有如此多的特点,又由于其免费的特点,这就给许多的中小应用提供了不错的选择。

尤其是对一些中小企业,无论是从降低成本,还是从性能方面,采用 MySQL 作为其数据支撑系统,都是一种可行的方案。但也应当注意,采用 MySQL 作为应用数据库,就意味着所有的问题都需要自己解决,要承担一定的风险。

## 1.2 MySQL 的兼容性

MySQL 支持 ANSI/ISO SQL 标准。SQL 标准自 1986 年以来不断演化发展,有数种版本。“SQL-92”指的是 1992 年发布的标准,“SQL:1999”指的是 1999 年发布的标准,“SQL:2003”指得是标准的当前版本。我们采用术语“SQL 标准”标示 SQL 标准的当前版本。

当然,MySQL 除了支持 SQL 标准外,为了提高速度和效益,也多 SQL 标准进行了扩展。

MySQL 服务器能够工作在不同的 SQL 模式下,并能针对不同的客户端以不同的方式应用这些模式。这样,应用程序就能对服务器操作进行量身定制以满足自己的需求。

这类模式定义了 MySQL 应支持的 SQL 语法,以及应该在数据上执行何种确认

检查。这样，就能在众多不同的环境下、与其他数据库服务器一起更容易地使用 MySQL。

可以使用 “--sql-mode="modes"” 选项，通过启动 `mysqld` 来设置默认的 SQL 模式。从 MySQL 4.1 开始，也能在启动之后，使用 `ET [SESSION|GLOBAL] sql_mode='modes'` 语句，通过设置 `sql_mode` 变量更改模式。

关于 MySQL 的更多特性或使用，请参考科卡在线 (<http://www.koorka.com/>) 的另一课程《MySQL 数据库管理》。在本课程中仅介绍 MySQL 的安装与配置，以及它的基本使用。

## 2. 安装 MySQL 数据库

MySQL 可以安装在 Windows、Linux、FreeBSD 和 Sun Solaris 等系统上。在本课程中，如果不特别说明，凡是涉及到 MySQL 的部分都是在 Linux 平台下。

### 2.1 MySQL 的版本

准备安装 MySQL 时，你应确定使用哪个版本。MySQL 的开发有几个发布系列，可以选择最适合你要求的一个版本。确定了安装的版本后，便可以选择分发版。有二进制或源码发布格式。

在 MySQL 开发过程中，同时存在多个发布系列，每个发布处在成熟度的不同阶段：

- MySQL 5.1 是最新开发的发布系列，是将执行新功能的系列。不久的将来可以使用 Alpha 发行，以便感兴趣的用户进行广泛的测试。
- MySQL 5.0 是当前稳定(产品质量)发布系列。只针对漏洞修复重新发布；没有增加会影响稳定性的新功能。
- MySQL 4.0 和 3.23 是旧的稳定(产品质量)发布系列。该版本不再使用，新的发布只用来修复特别严重的漏洞（以前的安全问题）。

通常，如果你是第一次开始使用 MySQL 或想要将它移植到一些还没有二进制分发版的系统上，我们推荐使用最终的稳定版本。目前是 MySQL 5.0。

对于早期版本和当前版本来讲，主要是内部机制和 SQL 方面的扩展。在本课程中介绍的方法和内容，对 3.23 之后的版本通用。

MySQL 的命名机制使用 3 个数字和一个后缀组成的版本号。例如 `mysql-5.0.9-beta` 的版本号这样解释：

- 第 1 个数字(5)是主版本号，描述了文件格式。所有版本 5 的发行都有相同的

文件格式。

- 第 2 个数字(0)是发行级别。主版本号和发行级别组合到一起便构成了发行序列号。

- 第 3 个数字(9)是在此发行系列的版本号，随每个新分发版递增。通常你需要已经选择的发行(release)的最新版本(版本)。

后缀显示发行的稳定性级别。通过一系列后缀显示如何改进稳定性。可能的后缀有：

- **alpha** 表明发行包含大量未被彻底测试的新代码。已知的缺陷应该在新闻小节被记录。请参见附录 D：MySQL 变更史。在大多数 **alpha** 版本中也有新的命令和扩展。**alpha** 版本也可能有主要代码更改等开发。但我们在发布前一定对其进行测试。

- **beta** 意味着该版本功能是完整的，并且所有的新代码被测试了，没有增加重要的新特征，应该没有已知的缺陷。当 **alpha** 版本至少一个月没有出现报导的致命漏洞，并且没有计划增加导致已经实施的功能不稳定的新功能时，版本则从 **alpha** 版变为 **beta** 版。在以后的 **beta** 版、发布版或产品发布中，所有 **API**、外部可视结构和 **SQL** 命令列均不再更改。

- **rc** 是发布代表；是一个发行了一段时间的 **beta** 版本，看起来应该运行正常。只增加了很小的修复。(发布代表即以前所称的 **gamma** 版)

- 如果没有后缀，这意味着该版本已经在很多地方运行一段时间了，而且没有非平台特定的缺陷报告。只增加了关键漏洞修复修复。这就是我们称为一个产品（稳定）或“通用”版本的东西。

## 2.2 从源代码安装 MySQL

1. 下载软件：

创建存放软件的目录：

`mkdir /backup/software` (可以存放在任何自己想存放的目录)

到 <http://dev.mysql.com/Downloads/> 下载最新稳定版的源代码，放到 `/backup/software`。

本文中下载的是：`mysql-5.0.28.tar.gz`

2. 解压软件：

```
tar -zxvf mysql-5.0.25.tar.gz
```

3.进入源代码目录:

```
cd mysql-5.0.25
```

4.创建 MySQL 的运行用户

```
groupadd mysql  
adduser -g mysql mysql
```

5.配置编译选项:

```
./configure \  
--prefix=/mnt/software/mysql \  
--with-unix-socket-path=/mnt/software/mysql/tmp/mysql.sock \  
--localstatedir=/mnt/database/mysql_data \  
--enable-asm  
--enable-asm  
--with-mysqld-ldflags=-all-static \  
--with-mysqld-user=mysql
```

编译选项说明:

--prefix=/mnt/software/mysql: 将所有文件安装到/mnt/software/mysql 目录下

--with-unix-socket-path=/mnt/software/mysql/tmp/mysql.sock: 指定 MySQL 的 Unix socket 文件存放的目录。

--localstatedir=/mnt/database/mysql\_data 指定 mysql 的日志存放位置和数据库位置。

--enable-asm: 允许使用汇编模式(优化性能)

--with-mysqld-ldflags=-all-static: 服务器使用静态库(优化性能)

--with-mysqld-user=mysql: 指定 MySQL 的运行用户。

其它常用编译选项说明:

--without-server: 仅编译安装客户端工具。

--with-client-ldflags=-all-static: 客户端使用静态库。

--with-charset=CHARSET: 指定 MySQL 使用的默认字符集。*CHARSET* 可以是: big5、cp1251、cp1257、czech、danish、dec8、dos、euc\_kr、gb2312、gbk、german1、hebrew、hp8、hungarian、koi8\_ru、koi8\_ukr、latin1、latin2、sjis、swe7、tis620、ujis、usa7 或 win1251ukr。如果不指定, 默认使用 latin1。

1 如果你需要其它选项，请使用 `./configure --help` 查看或参看文档  
<http://dev.mysql.com/doc/mysql/en/configure-options.html>

6.编译:

```
make
```

7.安装:

```
make install
```

8.拷贝 MySQL 的配置文件到/etc 目录下

```
cp support-files/my-medium.cnf /etc/my.cnf
```

9.初始化数据库:

```
cd /mnt/software/mysql
mkdir -p /mnt/database/mysql_data
bin/mysql_install_db --user=mysql
chown -R root /mnt/software/mysql(将软件的安装目录所有者改为 root 用户)
chown -R mysql /mnt/database/mysql_data(将存放数据库的目录所有者改为 mysql 用户)
chgrp -R mysql /mnt/software/mysql(将软件的安装目录属组改为 mysql 组)
```

10.启动数据库服务:

```
/mnt/software/mysql/bin/mysqld_safe --user=mysql &
```

11.设置环境变量:

如果要使用本地的客户端工具，并且不想每次都输入完整路径，就需要设置 PATH 变量。

```
export PATH=$PATH:/mnt/software/mysql/bin
```

12.测试:

执行 `/mnt/software/mysql/bin/mysql`，出现如下提示符

```
[root@koorka mysql]# /mnt/software/mysql/bin/mysql
```

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.0.28-log Source distribution
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
mysql>
```

执行 `show databases`，显示当前系统内的数据库，出现如下提示：

```
mysql> show databases;
+-----+
| Database          |
+-----+
| information_schema |
| mysql              |
| test               |
+-----+
3 rows in set (0.01 sec)
```

13.启动脚本(使系统启动时自动启动 MySQL 数据库服务):

在 Linux 系统下：

在 `/etc/rc.local` 文件内添加下面的行：

```
/mnt/software/mysql/bin/mysqld_safe --user=mysql&
```

14.MySQL 的启动和停止脚本

如果不想使用 13 步中的方法来在系统启动时自动启动 MySQL 服务，也可以使用 MySQL 提供的脚本：

```
[root@koorka mysql-5.0.28]# cp support-files/mysql.server /etc/init.d/MySQL
[root@koorka mysql-5.0.28]# chmod 755 /etc/init.d/MySQL
[root@koorka mysql-5.0.28]# chkconfig --add MySQL
```

## 3. MySQL 的基本管理

### 3.1 登录服务器

为了连接服务器，当调用 `mysql` 时，通常需要提供一个 MySQL 用户名并且很可能需要一个 密码。如果服务器运行在登录服务器之外的其它机器上，还需要指定



主机名。联系管理员以找出进行连接所使用的参数（即，连接的主机、用户名和使用的密码）。知道正确的参数后，可以按照以下方式进行连接：

```
shell> mysql -h host -u user -p
```

host 和 user 分别代表 MySQL 服务器运行的主机名和 MySQL 账户用户名。设置时替换为正确的值。\*\*\*\*\* 代表你的密码；当 mysql 显示 Enter password:提示时输入它。

如果有效，你应该看见 mysql>提示符后的一些介绍信息：

```
shell> mysql -h host -u user -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4 to server version: 5.0.28-log
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
mysql>
```

mysql> 提示符告诉你 mysql 准备为你输入命令。

连接上服务器并不代表选择了任何数据库，但是，这个时候就可以执行一些查询和管理操作了，例如，要求服务器告诉它的版本号和当前日期：

```
mysql> SELECT VERSION(), CURRENT_DATE;
+-----+-----+
| VERSION() | CURRENT_DATE |
+-----+-----+
| 5.0.28-log | 2006-10-27    |
+-----+-----+
1 row in set (0.00 sec)
mysql>
```

在 mysql>提示符下，要退出该程序，执行 quit 命令。要获取当前可执行的命令，执行 “\h” 或 help，要获取某个 SQL 关键字的帮助，执行 “help 关键字” 例如：

```
mysql> help SELECT
Name: 'SELECT'
Description:
Syntax:
SELECT
    [ALL | DISTINCT | DISTINCTROW ]
      [HIGH_PRIORITY]
    [STRAIGHT_JOIN]
    [SQL_SMALL_RESULT] [SQL_BIG_RESULT] [SQL_BUFFER_RESULT]
    [SQL_CACHE | SQL_NO_CACHE] [SQL_CALC_FOUND_ROWS]
  select_expr, ...
  [FROM table_references
```

```
[WHERE where_condition]
[GROUP BY {col_name | expr | position}
  [ASC | DESC], ... [WITH ROLLUP]]
[HAVING where_condition]
[ORDER BY {col_name | expr | position}
  [ASC | DESC], ...]
[LIMIT {[offset,] row_count | row_count OFFSET offset}]
[PROCEDURE procedure_name(argument_list)]
[INTO OUTFILE 'file_name' export_options
  | INTO DUMPFILE 'file_name'
  | INTO @var_name [, @var_name]]
[FOR UPDATE | LOCK IN SHARE MODE]]
.....
```

在帮助中会列出该 SQL 语句的详细语法和注意事项。

## 3.2 创建并使用数据库

要查看服务器上当前存在什么数据库，可以执行 `show databases` 命令（刚安装完成的服务器只有两个数据库）：

```
mysql>SHOW DATABASES
```

要创建数据库，可以使用 `CREATE DATABASE` 命令，例如，下面的命令创建了一个数据库，名称叫 `beijing`，用于存储北京分公司的数据：

```
mysql> CREATE DATABASE beijing;
Query OK, 1 row affected (0.00 sec)
```

需要特别注意的是，在 `Unix` 下，数据库名称是区分大小写的（不像 `SQL` 关键字），因此你必须总是以 `beijing` 访问数据库，而不能用 `BeiJing`、`BEIJING` 或其它一些变量。对表名也是这样的。（在 `Windows` 下，该限制不适用，尽管你必须在一个给定的查询中使用同样的大小写来引用数据库和表。但是，由于多种原因，作为最好的惯例，一定要使用与数据库创建时的同样的大小写。）

要使用刚创建的数据库，必须要先使用 `USE` 命令来选择该数据库：

```
mysql> USE beijing
Database changed
```

## 3.3 创建表格

创建数据库是很容易的部分，当使用 `SHOW TABLES` 来显示数据库所包含的表时，在这时它是空的：

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

使用数据库，较难的部分是决定你的数据库结构应该是什么：你需要什么数据库表，各数据库表中有什么样的列。

公司需要一个包含公司每个员工的记录的表。它可称为 `employees` 表，并且它应该包含最少每个员工的名字。因为只有名字本身不是很有用，表应该包含另外的信息。例如，我们可能还想列出每个员工的联系电话、住址、工资、年龄等一些基本的描述信息。

比较有趣的是年龄。年龄随着时间流逝而变化，如果直接将年龄存储到数据库，这意味着你将要不断地更新你的记录。相反，存储一个固定值例如生日比较好，那么，无论何时你需要年龄，可以以当前日期和出生日期之间的差来计算它。`MySQL` 提供了日期运算函数，因此这并不困难。存储出生日期而非年龄还有其它优点：

- 你可以使用数据库完成这样的任务，例如生成即将到来的员工生日的提示。（如果你认为这类查询有点蠢，注意，这与从商务数据库来识别出不久要发给生日祝贺的客户是同一个问题，因为计算机帮助私人联络。）
- 你可以相对于日期而不止是当前日期来计算年龄。

下面使用一个 `CREATE TABLE` 语句指定你的数据库表的布局，该表格用于存储员工信息，包括员工姓名、生日、住址、联系电话、工资：

```
mysql> CREATE TABLE employees (name VARCHAR(20), birthday DATE,  
address VARCHAR(50), phone VARCHAR(20), salary FLOAT);
```

员工姓名、地址是字符串，而且其长度不定，所以用 `VARCHAR` 来指定其数据类型（20 表示最长只能有 20 个字符），而电话号码有手机和固定电话，长度也不顾定；生日是日期，用 `DATE`；工资可能有小数点，用 `FLOAT`。

创建了数据库表后，`SHOW TABLES` 应该产生一些输出：

```
mysql> SHOW TABLES;  
+-----+  
| Tables_in_beijing |  
+-----+  
| employees          |  
+-----+  
1 row in set (0.01 sec)
```

为了验证你的表是按你期望的方式创建，使用一个 `DESCRIBE` 语句来列出表格的结构：

```
mysql> DESCRIBE employees;  
+-----+-----+-----+-----+-----+-----+
```

Field	Type	Null	Key	Default	Extra
name	varchar(20)	YES		NULL	
birthday	date	YES		NULL	
address	varchar(20)	YES		NULL	
phone	varchar(20)	YES		NULL	
salary	float	YES		NULL	

5 rows in set (0.01 sec)

将下列数据添加到表格中：

姓名	生日	住址	电话	工资
zhangqiang	1979-07-16	guangzou	13312345678	8000.00
wangkai	1981-11-07	shanghai	13987654321	6000.00
koorka	1976-03-21	beijing	010-51261979	9300.00

要将上面的数据存储在数据库中，需要使用 INSERT INTO 语句：

```
mysql> INSERT INTO employees(name,birthday,address,phone,salary)
values('zhangqiang', '1979-07-16', 'guangzou', '13312345678',8000.00);
```

其它两条数据如法炮制即可。需要注意的是：如果数据是字符串或日期，需要使用单引号引起来，而且日期有规定的格式；如果是数值，则不能使用单引号。

## 3.4 用户及权限管理

MySQL 有先进但非标准的安全/权限系统。本节描述它的工作原理。

### 3.4.1 MySQL 访问权限系统

MySQL 权限系统的主要功能是证实连接到一台给定主机的用户，并且赋予该用户在数据库上的 SELECT、INSERT、UPDATE 和 DELETE 权限。

附加的功能包括有匿名的用户并对于 MySQL 特定的功能例如 LOAD DATA INFILE 进行授权及管理操作的能力。

MySQL 权限系统保证所有的用户只执行允许做的事情。当连接 MySQL 服务器时，你的身份由你从那儿连接的主机和你指定的用户名来决定。连接后发出请求后，系统根据你的身份和你想做什么来授予权限。

MySQL 在认定身份中考虑你的主机名和用户名字，是因为几乎没有原因假定一个给定的用户在因特网上属于同一个人。例如，从 beijing.koorka.com 连接的用户 joe

不一定和从 shanghai.koorka.com 连接的 joe 是同一个人。MySQL 通过允许你区分在不同的主机上碰巧有同样名字的用户来处理它：你可以对 joe 从 beijing.koorka.com 进行的连接授予一个权限集，而为 joe 从 shanghai.koorka.com 的连接授予一个不同的权限集。

MySQL 存取控制包含 2 个阶段：

阶段 1：服务器检查是否允许你连接。

阶段 2：假定你能连接，服务器检查你发出的每个请求。看你是否有足够的权限实施它。例如，如果你从数据库表中选择(select)行或从数据库删除表，服务器确定你对表有 SELECT 权限或对数据库有 DROP 权限。

### 3.4.2 MySQL 用户账户管理

MySQL 使用用户名和客户端或主机定义 MySQL 账户，用户可以根据这些名称来连接服务器。账户也有密码。MySQL 和操作系统使用用户名和密码的方式有几处区别：

- MySQL 用于鉴定目的用户名与 Windows 或 Linux/Unix 使用的用户名(登录名)没有关系。在 Linux/Unix 中，大多数 MySQL 客户端默认试图使用当前 Linux/Unix 的用户名作为 MySQL 用户名来登录，但这样只是为了方便。默认值可以很容易被覆盖，因为客户端程序允许用 -u 或 --user 选项来指定用户名。因为这表示任何人可以试图使用任何用户名来连接服务器，除非所有 MySQL 账户有密码，否则你不能使数据库保持安全。通过为没有密码的账户指定用户名，任何人能够成功连接服务器。

- MySQL 用户名最大可达 16 字符长。这样可以限制 MySQL 服务器和客户端之间的硬编码，并且防止通过修改 mysql 数据库中表的定义来偷窃密码。

*注：应绝对不要以任何方式修改 mysql 数据库中的任何表，只能运行 MySQL 分发中专为此目的提供的脚本。将 MySQL 系统表重新定义为其它方式会导致未定义的(和不支持的!)行为。*

MySQL 的加密密码使用自己的算法。该加密算法不同于 Linux/Unix 登录过程使用的算法。MySQL 密码加密与 PASSWORD() SQL 函数的方法相同。Linux/Unix 密码加密与 ENCRYPT() SQL 函数的方法相同。从版本 4.1 起，MySQL 使用更强的鉴定方法，同以前的版本相比可以在连接过程中提供更好的密码保护。即使 TCP/IP 包被截取或 mysql 数据库被捕获也很安全。(在前面的版本中，即使密码以加密形式保存到 user 表中，仍可以通过加密密码值来连接 MySQL 服务器)。

### 3.4.2.1 向 MySQL 增加、删除用户

可以用两种方式创建 MySQL 账户：

- 使用 GRANT 语句
- 直接操作 MySQL 授权表

最好的方法是使用 GRANT 语句，因为这样更精确，错误少。关于直接操作 MySQL 授权表，此处不进行详细介绍。

下面的示例说明如何使用 MySQL 客户端程序来添加新用户。

首先，使用 MySQL 程序以 MySQL root 用户来连接服务器：

```
[root@koorka ~]# mysql --user=root mysql
```

如果你为 root 账户指定了密码，还需要为该 MySQL 命令和本节中的其它命令提供 --password 或 -p 选项。

以 root 连接到服务器上后，可以添加新账户。下面的语句使用 GRANT 来设置四个新账户：

```
mysql> GRANT ALL PRIVILEGES ON *.* TO 'monty'@'localhost'  
-> IDENTIFIED BY 'some_pass' WITH GRANT OPTION;  
mysql> GRANT ALL PRIVILEGES ON *.* TO 'monty'@'%'  
-> IDENTIFIED BY 'some_pass' WITH GRANT OPTION;  
mysql> GRANT RELOAD,PROCESS ON *.* TO 'admin'@'localhost';  
mysql> GRANT USAGE ON *.* TO 'dummy'@'localhost';
```

用 GRANT 语句创建的账户有下面的属性：

- 其中两个账户有相同的用户名 monty 和密码 some\_pass。两个账户均为超级用户账户，具有完全的权限可以做任何事情。一个账户 ('monty'@'localhost')只用于从本机连接时。另一个账户('monty'@'%')可用于从其它主机连接。请注意 monty 的两个账户必须能从任何主机以 monty 连接。没有 localhost 账户，当 monty 从本机连接时，mysql\_install\_db 创建的 localhost 的匿名用户账户将占先。结果是，monty 将被视为匿名用户。原因是匿名用户账户的 Host 列值比'monty'@'%'账户更具体，这样在 user 表排序顺序中排在前面。

- 一个账户有用户名 admin，没有密码。该账户只用于从本机连接。授予了 RELOAD 和 PROCESS 管理权限。这些权限允许 admin 用户执行 mysqladmin reload、mysqladmin refresh 和 mysqladmin flush-xxx 命令，以及 mysqladmin processlist。未授予访问数据库的权限。你可以通过 GRANT 语句添加此类权限。

- 一个账户有用户名 dummy，没有密码。该账户只用于从本机连接。未授予权限。通过 GRANT 语句中的 USAGE 权限，你可以创建账户而不授予任何权限。它可以将所有全局权限设为 'N'。假定你将在以后将具体权限授予该账户。

再来看另一个创建用户的实例：

```
mysql> SELECT,INSERT,UPDATE,DELETE
->      ON bankaccount.*
->      TO 'custom'@'localhost'
->      IDENTIFIED BY 'obscure';
```

创建了一个 `custom` 账户，可以访问 `bankaccount` 数据库，但只能从本机访问。在访问数据库时具有 `SELECT,INSERT,UPDATE,DELETE` 权限。

要删除删除用户，使用 `DROP USER` 即可：

```
mysql> DROP USER 'custom'@'localhost';
```

要修改用户的密码，可以直接使用 `SET` 命令，例如：

```
SET PASSWORD FOR 'root'@'localhost' = PASSWORD('MyNewPassword');
```

### 3.4.2.2 添加删除权限

账户权限信息被存储在 `mysql` 数据库的 `user`、`db`、`host`、`tables_priv`、`columns_priv` 和 `procs_priv` 表中。在 `MySQL` 启动时服务器将这些数据库表内容读入内存。

要修改一个用户的权限，可以直接修改上面的几个表，也可以使用 `GRANT` 和 `REVOKE` 语句。推荐使用后者。

`GRANT` 和 `REVOKE` 语句所用的涉及权限的名称如下：

<code>CREATE</code>	#创建数据库、表
<code>DROP</code>	#删除数据库、表
<code>GRANT OPTION</code>	#可以对用户授权的权限
<code>REFERENCES</code>	#可以创建外键（4.1 以上版本才有）
<code>ALTER</code>	#修改数据库、表的属性
<code>DELETE</code>	#在表中删除数据
<code>INDEX</code>	#创建和删除索引
<code>INSERT</code>	#向表中添加数据
<code>SELECT</code>	#从表中查询数据
<code>UPDATE</code>	#修改表的数据
<code>CREATE VIEW</code>	#创建视图（4.1 以上版本才有）
<code>SHOW VIEW</code>	#显示视图的定义（4.1 以上版本才有）
<code>ALTER ROUTINE</code>	#修改存储过程
<code>CREATE ROUTINE</code>	#创建存储过程
<code>EXECUTE</code>	#执行存储过程



FILE	#读、写服务器上的文件
CREATE TEMPORARY TABLES	#创建临时表
LOCK TABLES	#锁定表格
CREATE USER	#创建用户
PROCESS	#管理服务器和客户连接进程
RELOAD	#重载服务
REPLICATION CLIENT	#用于复制
REPLICATION SLAVE	#用于复制
SHOW DATABASES	#显示数据库
SHUTDOWN	#关闭服务器
SUPER	#超级用户

下面通过几个实例来说明：

（1）授予创建数据库的权限：

```
mysql> GRANT CREATE ON *.* TO 'dummy'@'localhost';
```

（2）授予用户在数据 beijing 内具有创建表的权限：

```
mysql> GRANT CREATE ON beijing.* TO 'dummy'@'localhost';
```

（3）授予用户在数据库 beijing 内对表 products,orders 有 SELECT, INSERT, UPDATE 的权限：

```
mysql> GRANT SELECT,INSERT,UPDATE ON beijing.products,beijing.orders TO 'dummy'@'localhost';
```

## 4. MySQL 程序介绍

标准的 MySQL 版本中提供了几种类型的命令行运用程序程序：

（1）MYSQL 服务器和服务器启动脚本：

- o mysqld 是 MySQL 服务器
- o mysqld\_safe、mysql.server 和 mysqld\_multi 是服务器启动脚本
- o mysql\_install\_db 初始化数据目录和初始数据库

（2）访问服务器的客户程序：

- o mysql 是一个命令行客户程序，用于交互式或以批处理模式执行 SQL 语句。



- o       mysqladmin    是用于管理功能的客户程序。
- o       mysqlcheck    执行表维护操作。
- o       mysqldump    和 mysqlhotcopy 负责数据库备份。
- o       mysqlimport   导入数据文件。
- o       mysqlshow    显示信息数据库和表的相关信息。

(3) 独立于服务器操作的工具程序:

- o       myisamchk    执行表维护操作。
- o       myisampack   产生压缩、只读的表。
- o       mysqlbinlog   是处理二进制日志文件的实用工具。
- o       perror        显示错误代码的含义。

除了上面介绍的这些随 MySQL 一起发布的命令行工具外, MySQL AB 公司还提供了 3 个 GUI 客户程序供 MySQL 服务器使用, 这 3 个工具需要单独下载:

- MySQL 管理器: 该工具用于管理 MySQL 服务器、数据库、表以及用户。
- MySQL 查询浏览器: 该图形工具由 MySQL AB 提供, 用于创建、执行以及优化对 MySQL 数据库的查询。
- MySQL 移植工具包: 该工具可以帮助你计划和数据从其它关系数据库管理系统移植到 MySQL。

关于 MySQL 的管理就暂时先介绍这些, 如果需要深入研究 MySQL 数据库, 例如: MySQL 的备份、MySQL 的引擎、复制、主从服务、存储过程、国际化、SQL 优化等知识, 在另一门课程《MySQL 数据库管理》中有详细讲解, 可以登录 <http://www.koorka.com/> 查看详细大纲。