

第 9 章 MySQL 数据库服务器配置

学习引导

【本章概述】

Linux 系统中可以安装多种关系数据库管理系统(RDBMS)。如 Oracle、MySQL、PostSQL 和 DB2 等。其中 MySQL 数据库是最具有竞争力的数据库,因为它是一个高性能、多线程、多用户、建立在客户/服务器结构上的 RDBMS,并且他是可以免费获取的。MySQL 数据库是专门为速度和稳定性而设计的小型数据库。现在,已经有越来越多的网站使用并且配置了基于 MySQL 的应用程序,MySQL 已经在商业、教育、科学和工程领域得到了广泛的应用。

本章围绕三个项目主要介绍 MySQL 数据库在 Linux 系统中的安装、配置、管理和维护方法。

【教学重点与难点】

重点:MySQL 服务器的安装方法、安装过程以及测试方法。

难点:MySQL 数据库的权限管理,MySQL 数据库的备份与恢复方法。

9.1 项目一:MySQL 的安装与启动

【项目描述】

公司设计部需要在服务器上部署一个用于发布项目考核信息的网站,在该网站上实现员工对考核内容的选择和考核成绩的查询等基本操作。为此,除了需要部署前台相关 Web 服务外,还需要部署后台的数据库服务。管理员选取目前比较常用的 MySQL 数据库,因此需要对 MySQL 数据库进行安装并测试。

【构思设计】

该项目是搭建后台 MySQL 数据库的第 1 步,进行服务的安装与测试。MySQL 数据库的

安装可以选择使用 RPM 软件包安装或者源代码版本安装。RPM 软件包安装比源代码安装要容易些,但是不灵活,而源代码安装要比 RPM 安装有更好的灵活性。安装后启动 MySQL 服务器并进行测试,相关知识点如表 9.1 所示。

表 9.1 本项目知识点分析

序号	知识点	详见章节
1	了解 MySQL 数据库	9.1.1
2	安装 MySQL 数据库的过程	9.1.2
3	管理 mysqld 服务	9.1.3
4	测试 MySQL 数据库	9.1.4

【实施运行】

操作步骤:

```
# rpm -qmysql //检查是否安装 MySQL 软件
```

如果没有安装则下载 MySQL 软件包或者从安装光盘中查找。

```
# rpm -ivh mysql-server-5.0.77-4.el5_4.2.i386.rpm //安装 MySQL 服务器,详见 9.1.2
```

```
# rpm -ivh mysql-5.0.77-4.el5_4.2.i386.rpm //安装客户端程序和共享库,详见 9.1.2
```

```
# service mysqld start //开启 MySQL 服务器,注意服务名为 mysqld,详见 9.1.3
```

```
# chkconfig --level 235 mysqld on //设置 MySQL 服务器在 235 级别自启动,详见 9.1.3
```

```
# /usr/bin/mysqladmin version //查看服务器状态,详见 9.1.4
```

/usr/bin/mysql //测试是否能连接到 MySQL 服务器。出现:mysql>表示可以连接 MySQL 服务器了,详见 9.1.4

```
# /usr/bin/mysqladmin -u root -h localhost password '123456' //修改 root 用户登录 MySQL 服务器的口令,详见 9.1.4
```

9.1.1 MySQL 数据库简介

MySQL 由瑞典 MySQL AB 公司开发,是一个开放源码的小型关系型数据库管理系统,关系型数据库将数据保存在不同的表中,而不是将所有数据放在一个大仓库内,这样就增加了对数据库的访问速度并提高了灵活性。由于 MySQL 数据库具有体积小、速度快、总体拥有成本低,尤其是开放源码等特点,使许多中小型网站为了降低网站总体成本而选择 MySQL 作为网站数据库。

在 Linux 操作系统中,Apache 和 Nginx 作为 Web 服务器,MySQL 作为数据库,PHP/Perl/Python 作为服务器端脚本解释器。由于这几个软件都是免费或开放源码软件,因此使用这组软件不用花一分钱就可以建立起一个稳定、免费的网站系统,被业界称为“LAMP”组合。

9.1.2 安装 MySQL 数据库

MySQL 安装可以选取 RPM 软件包安装方式或者源代码版本方式。相比而言,RPM 安装方式简单,而源代码安装灵活,通过对源代码重新裁剪和编译可以根据具体需要对安装目录进行重新设置,也可以编译一个更小型、更轻松的 MySQL 版本,去掉一些特性,节省开销,还可以

选择编译安装 MySQL 最新的测试功能。

(1)RPM 软件包安装方式:可以从 <http://www.mysql.com> 上下载需要的 MySQL 的软件包,也可以在安装光盘中找到需要安装的 RPM 软件包。至少需要以下两个软件包。

MySQL 客户端程序和共享库:mysql-5.0.77-4.el5_4.2.i386

MySQL 数据库服务器,管理数据库和表、控制用户访问和处理 SQL 查询:mysql-server-5.0.77-4.el5_4.2.i386

安装命令如下:

```
# rpm -ivh mysql-server-5.0.77-4.el5_4.2.i386.rpm
# rpm -ivh mysql-5.0.77-4.el5_4.2.i386.rpm
```

(2)当用户进行定制安装,就需要安装源代码版本的 MySQL。可以从 Internet 上下载所需要的版本,以 mysql-5.0.56.tar.gz 为例,步骤如下:

```
# tar -zxvf mysql-5.0.56.tar.gz //以 root 用户切换到源代码包所在的目录,使用 tar 命令释放文件
```

```
# cd mysql-5.0.56 //进入包含源代码的目录
# make //使用 make 编译程序
# make install //使用 make install 继续安装
# groupadd mysql //创建 mysql 用户组
# useradd -g mysql mysql //创建 mysql 用户
# /scripts/mysql_install_db //运行程序中包含的初始化脚本 mysql_install_db,为运行 MySQL
```

做准备

```
# chown -R mysql /usr/local/mysql //递归修改/usr/local/mysql 目录的拥有者为 mysql 用户
# chgrp -R mysql /usr/local/mysql //递归修改/usr/local/mysql 目录的所属群组为 mysql 组
```

9.1.3 启动与停止 MySQL 数据库

MySQL 服务的管理可以采用命令和图形界面两种方式。

1. 命令方式启动或停止 MySQL 服务

启动 MySQL 服务,如图 9.1 所示。

```
[root@localhost ~]# service mysqld start
启动 MySQL: [确定]
```

图 9.1 启动 MySQL 服务

停止 MySQL 服务,如图 9.2 所示。

```
[root@localhost ~]# service mysqld stop
停止 MySQL: [确定]
```

图 9.2 停止 MySQL 服务

重新启动 MySQL 服务,操作过程如图 9.3 所示。

```
[root@localhost ~]# service mysqld restart
停止 MySQL: [确定]
启动 MySQL: [确定]
```

图 9.3 重新启动 MySQL 服务

检查 MySQL 服务状态,操作过程如图 9.4 所示。

```
[root@localhost ~]# service mysqld status
mysqld (pid 8392) 正在运行...
```

图 9.4 查询 MySQL 服务

注意:如果是采用源代码版安装的 MySQL 数据库,可通过命令启动 MySQL 服务,操作过程如图 9.5 所示。“&”表示在后台启动服务。

```
[root@localhost ~]# /usr/local/mysql/bin/mysql_safe --user=mysql&
[1] 8456
[root@localhost ~]# Starting mysqld daemon with databases from /var/lib/mysql
```

图 9.5 源代码安装时启动 MySQL 服务

停止 MySQL 服务,操作过程如图 9.6 所示。

```
[root@localhost ~]# /usr/local/mysql/mysqldadmin shutdown
```

图 9.6 停止 MySQL 服务

2. 图形界面方式启动或停止 MySQL 服务

如果安装系统时同时安装了 MySQL 组件,可以在终端下执行 `ntsysv` 命令,GUI 界面弹出“服务配置”窗口,找到 `mysqld` 服务在其前面选择“*”保存退出即可,如图 9.7 所示。或者选择【系统】→【管理】→【服务器设置】→【服务】,弹出“服务配置”窗口,选择 `mysqld`,并点击【启动】,如图 9.8 所示。



图 9.7 使用 ntsysv 命令启动 MySQL 服务



图 9.8 通过“服务配置”窗口启动 MySQL 服务

如果是通过安装源代码包安装,可以把启动 MySQL 服务的命令编辑到/etc/rc.local 文件里。这样,主机在启动后就可以自动启动 MySQL 服务。

9.1.4 测试 MySQL 数据库

不管通过何种方式安装 MySQL,安装完成后都要进行测试,以确定安装正确可以进行下一步的数据库建立。

(1)使用 mysqladmin 命令查看服务器状态,它位于/usr/bin 目录下。使用这个命令可以看到与图 9.9 类似的输出结果。

```
[root@localhost ~]# /usr/bin/mysqladmin version
/usr/bin/mysqladmin Ver 8.41 Distrib 5.0.77, for redhat-linux-gnu on i386
Copyright (C) 2000-2006 MySQL AB
This software comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to modify and redistribute it under the GPL license

Server version          5.0.77
Protocol version        10
Connection              Localhost via UNIX socket
UNIX socket             /var/lib/mysql/mysql.sock
Uptime:                 4 hours 12 min 22 sec

Threads: 1  Questions: 3  Slow queries: 0  Opens: 12  Flush tables: 1  Open
tables: 6  Queries per second avg: 0.000
```

图 9.9 命令 mysqladmin 的输出结果

(2)使用 MySQL 客户端连接服务器

可以使用/usr/bin/mysql 来测试是否能连接到 MySQL 服务器。如果出现 mysql> 提示符表示 MySQL 客户端可以连接到 MySQL 服务器,如图 9.10 所示。

```
[root@localhost ~]# /usr/bin/mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.0.77 Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> █
```

图 9.10 mysql 命令显示的输出结果

(3)修改 MySQL 的 root 口令

当 MySQL 第一次安装后,MySQL 的 root 用户不同于系统的 root 用户。MySQL 的 root 用户对数据库进行访问是受到限制的。默认情况下,口令为空,这是非常不安全的事情。需要使用 mysqladmin 来更改口令,如图 9.11 所示。

```
[root@localhost ~]# /usr/bin/mysqladmin -u root password 'admin'
```

图 9.11 设置 root 用户密码

密码会立即生效。当下次连接 MySQL 服务器时,就需要上面设置的密码“admin”,退出使用 quit,如图 9.12 所示。

```
[root@localhost ~]# /usr/bin/mysql -h localhost -root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 6
Server version: 5.0.77 Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> quit
Bye
```

图 9.12 使用 root 用户登录 MySQL 数据库

如果 root 用户已经设置了密码“admin”，需要更改 MySQL 的 root 用户密码为“mysql”时，操作如图 9.13 所示。

```
[root@localhost ~]# /usr/bin/mysqladmin -u root -p password 'mysql'
Enter password:
```

图 9.13 修改 root 用户登录 MySQL 数据库的密码

需要注意的是在 password 后面输入的是新密码“mysql”。当终端提示输入密码时，应输入原始密码“admin”。确认后，密码就改为用户重新定义的密码。

9.2 项目二：新建数据库和数据表

【项目描述】

在 9.1 项目基础上，管理员准备在 MySQL 数据库中创建项目考核信息发布网站需要的数据库(couman)和相关表。数据表包括：员工信息表(employee)、考核内容表(exam)、员工考核信息表(score)，各个表的结构如下：

员工信息表(employee)字段：员工号(eno)、姓名(ename)、性别(sex)、项目组(groups)；

考核内容表(exam)字段：内容序号(exid)、考核内容(cname)；

员工考核信息表(score)字段：序号(scid)、员工号(eno)、内容序号(exid)、成绩(score)。

数据表创建成功后，管理员将对数据进行增删改查的测试。

【构思设计】

本项目主要是完成在 MySQL 数据库中基本的数据库操作，首先需要创建数据库(couman)，创建数据库表：学生信息表(student)、课程描述表(course)、学生选课表(stucou)，在创建数据表时要完成对数据完整性的约束。然后使用 SQL 语言对数据信息进行测试。知识点主要涉及 MySQL 数据库的使用和 SQL 语言，如表 9.2 所示。

表 9.2 本项目知识点分析

序号	知识点	详见章节
1	数据库创建	9.2.1
2	数据表的创建	9.2.2
3	对数据的增删改查操作	9.2.3

【实施运行】

操作步骤:

```
# /usr/bin/mysql -h localhost -u root -p //登录 mysql 服务器
mysql> create database couman; //创建 couman 数据库, 详见 9.2.1
mysql> show databases; //显示目前所有数据库的列表, 详见 9.2.1
mysql> use couman; //选择 couman 数据库, 详见 9.2.1
mysql> create table employee (eno varchar(10) not null, ename varchar(30) not null, sex int(5)
default 0, groups varchar(20), primary key(eid)); //创建 employee 表, eno、ename 字段不为空, sex 字段默
认为 0, 详见 9.2.2
mysql> describe employee; //查看 employee 表结构, 详见 9.2.2
mysql> create table exam(exid varchar(10) not null, exname varchar(50) not null, primary key
(exid)); //创建 exam 表, exid、exname 字段不为空, 详见 9.2.2
mysql> describe exam; //查看 exam 表结构, 详见 9.2.2
mysql> create table score(scid int(10) not null auto_increment, eno varchar(10) not null, exid
varchar(10) not null, score int(5), primary key (scid), constraint foreign key (eno) references exam(eno)
, constraint foreignkey (eno) references employee(eno)); //创建 score 表, 设置 scid 字段为自动递增主键,
eno、exid 字段为外键不为空, 详见 9.2.2
mysql> describe score; //查看 score 表结构, 详见 9.2.2
mysql> show tables; //显示所有的表, 详见 9.2.2
mysql> insert into employee (eno, ename, sex, groups) values (1001, 'wangli', 1, 'dgroup01'); //向
employee 表插入数据, 详见 9.2.3
mysql> insert into employee (eno, ename, sex, groups) values (1002, 'liuning', 1, 'dgroup01');
mysql> insert into employee (eno, ename, sex, groups) values (1003, 'zhangqi', 0, 'dgroup02');
mysql> select * from employee; //查看 employee 表中内容, 详见 9.2.3
mysql> insert into exam (exid, exname) values (10001, 'test'); //向 exam 表插入数据
mysql> insert into exam (exid, exname) values (10002, 'program');
mysql> insert into exam (exid, exname) values (10003, 'english');
mysql> select * from exam; //查看 exam 表中内容
mysql> insert into score (eno, exid, score) values (1001, 10001, 87); //向 score 表插入数据
mysql> insert into score (eno, exid, score) values (1001, 10002, 94);
mysql> insert into score (eno, exid, score) values (1002, 10003, 72);
mysql> insert into score (eno, exid, score) values (1003, 10001, 85);
mysql> select * from score; //查看 score 表中内容
```

9.2.1 数据库操作

在使用数据库存储数据之前首先要完成数据库的创建, 然后在数据库中创建数据表和表中的数据, 然后通过对数据的增删改查来完成对数据库信息的使用, 同时作为数据库管理员还需要完成数据库权限、备份等管理。

对数据库的常用操作包括: 创建数据库、选择数据库和删除数据库。

1. 创建数据库

登录数据库后,使用“create database 数据库名;”完成对数据库的创建。然后可以使用“show databases;”命令来查看目前所有数据库信息。Red Hat Enterprise Linux 5 系统安装了 MySQL 服务后,默认自带 3 个数据库,information_schema 数据库存储相关信息,mysql 数据库存储授权表,test 数据库提供给用户练习使用,如图 9.14 所示。

```
[root@localhost mysql]# /usr/bin/mysql -h localhost -u root -p
Enter password:
mysql> create database couman;
Query OK, 1 row affected (0.00 sec)
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| couman |
| mysql |
| test |
+-----+
4 rows in set (0.00 sec)
```

图 9.14 创建并显示 couman 数据库

2. 选择数据库

当需要对某一数据库进行操作时,首先需要选择该数据库才能操作,使用“use 数据库名字;”命令来完成,如图 9.15 所示。

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

图 9.15 选择 couman 数据库

3. 删除数据库

使用“delete database 数据库名;”命令删除指定数据库,如图 9.16 所示,删除数据库时,连同数据库里的所有表(包括其中的数据)和数据库目录都将被删除。

```
mysql> drop database couman;
Query OK, 0 rows affected (0.00 sec)
```

图 9.16 删除 couman 数据库

当删除不存在的数据库时,会提示 MySQL 错误信息,为了避免这种情况的发生,可以在命令中添加 if exists 子句,如图 9.17 所示。

```
mysql> drop database teacher;
ERROR 1008 (HY000): Can't drop database 'teacher'; database doesn't exist
mysql> drop database if exists teacher;
Query OK, 0 rows affected, 1 warning (0.00 sec)
```

图 9.17 删除 teacher 数据库

9.2.2 数据表操作

创建数据库后,需要进一步创建和管理数据表。每个表由行和列组成,每一行是一条记录,每个记录包含多个列(字段)。对数据库表的常用操作包括:创建表、修改表、复制表和删除表。

1. 创建表

创建表的 SQL 语句:

create table 表名(字段名 1 字段类型[字段约束], 字段名 2 字段类型[字段约束],..., 字

段名 n 字段类型[字段约束],[表约束]) [Type|Engine=表类型|存储引擎];

其中字段类型,也称作列类型,规定了某个字段所允许输入的数据类型,常用字段类型如表 9.3 所示。

表 9.3 字段类型

INT	整形,4 个字节
FLOAT	数值类型,支持浮点数或小数
DOUBLE	数值类型,支持双精度浮点数
TIME	HH:MM:SS 格式的时间字段
DATE	YYYYMMDD 格式的日期字段
CHAR	字符型,最大长度为 255
VARCHAR	字符串类型,最大长度为 255
YEAR	YYYY 或 YY 格式的年字段
BLOB	可变数据的二进制类型
ENUM	枚举型

字段约束用于进一步约束某个字段允许输入的数据,常见的字段约束如表 9.4 所示。

表 9.4 字段约束

Null 或 Not Null	允许字段为空或不为空,默认为 Null
Default	指定字段的默认值
Auto_Increment	设置 INT 型字段能够自动生成递增 1 的整数

表约束用于确定表的主键、外键和索引等,如表 9.5 所示。

表 9.5 表级约束

Primary Key	指定主键
Foreign Key	指定外键
Index	指定索引
Unique	指定唯一索引
Fulltext	指定全文索引

表类型则指明了表中数据的存储格式,MySQL 支持多个存储引擎作为不同类型的处理器。其中默认的为“MyISAM”,对应的存储引擎为“MyISAM”。表 9.6 列出了常用的表存储引擎。

表 9.6 表存储引擎

ARCHIVE	档案存储引擎
BDB	带页面锁定的事物安全表
CSV	值之间用逗号隔开的表
EXAMPLE	示例引擎
MyISAM	二进制存储引擎

一旦数据库创建成功会在 `/var/lib/mysql` 下生成一个与数据库同名的目录。表创建成功后会在该目录中生成“表名. frm 文件”表示新建的每个表,“表名. MYD 文件”表示数据信息,“表名. MYI 文件”表示索引,如表 9.7 所示。

表 9.7 MyISAM 表文件

表的名字. frm	表格式(定义)文件
表的名字. MYD	数据文件
表的名字. MYI	索引文件

在 9.2 项目中创建员工信息表(employee)字段:员工号(eno)、姓名(ename)、性别(sex)、项目组(groups);

考核内容表(exam)字段:内容序号(exid)、考核内容(cname);

员工考核信息表(score)字段:序号(scid)、员工号(eno)、内容序号(exid)、成绩(score)。

操作过程如图 9.18 所示。

```
mysql> create table employee (eno varchar(10) not null,ename varchar(30) not null, sex int(5)
default 0 , groups varchar(20),primary key(eno));
Query OK, 0 rows affected (0.00 sec)

mysql> describe employee ;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| eno   | varchar(10)   | NO   | PRI | NULL    |       |
| ename | varchar(30)   | NO   |     | NULL    |       |
| sex   | int(5)        | YES  |     | 0       |       |
| groups | varchar(20)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> create table exam(exid varchar(10) not null,exname varchar(50) not null ,primary key (exid));
Query OK, 0 rows affected (0.01 sec)
mysql> describe exam;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| exid  | varchar(10)   | NO   | PRI | NULL    |       |
| exname | varchar(50)   | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> create table score (scid int(10) not null auto_increment,eno varchar(10) not null,exid
varchar(10) not null,score int(5),primary key (scid), constraint foreign key (eno) references
exam(eno) , constraint foreign key (eno) references employee(eno));
Query OK, 0 rows affected (0.01 sec)
mysql> describe score;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| scid  | int(10)       | NO   | PRI | NULL    | auto_increment |
| eno   | varchar(10)   | NO   | MUL | NULL    |       |
| exid  | varchar(10)   | NO   |     | NULL    |       |
| score | int(5)        | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> show tables;
+-----+
| Tables_in_couman |
+-----+
| employee          |
| exam              |
| score             |
+-----+
3 rows in set (0.00 sec)
```

图 9.18 创建 employee、exam、score 表

其中“describe 表名”用于查看所创建的表的结构，“show tables”用于列出已经创建的表。

2. 修改表

修改表的语句：

Alter table 表名 操作 1[,操作 2,...];

可以对表的结构进行修改,包括添加、删除或修改字段,更改表名或类型等。操作包括 add、change、modify、drop 和 rename 等。

例如:在 employee 表中增加一个字段 email,可以使用 add,如图 9.19 所示。

```
mysql> alter table employee add email varchar(20);
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> describe employee;
```

Field	Type	Null	Key	Default	Extra
eno	varchar(10)	NO	PRI	NULL	
ename	varchar(30)	NO		NULL	
sex	int(5)	YES		0	
groups	varchar(20)	YES		NULL	
email	varchar(20)	YES		NULL	

```
5 rows in set (0.01 sec)
```

图 9.19 在 employee 表中增加一个字段 email

使用 change 把 employee 表中的字段 email 改为 mail,如图 9.20 所示。

```
mysql> alter table employee change email mail varchar(20);
Query OK, 0 rows affected (0.01 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> describe employee;
```

Field	Type	Null	Key	Default	Extra
eno	varchar(10)	NO	PRI	NULL	
ename	varchar(30)	NO		NULL	
sex	int(5)	YES		0	
groups	varchar(20)	YES		NULL	
mail	varchar(20)	YES		NULL	

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

图 9.20 把 employee 表中的字段 email 改为 mail

使用 modify 把 employee 表中字段 mail 的类型改为 char,如图 9.21 所示。

```
mysql> alter table employee modify mail char(50);
Query OK, 0 rows affected (0.01 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> describe employee;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| eno   | varchar(10)  | NO   | PRI | NULL    |       |
| ename | varchar(30)  | NO   |     | NULL    |       |
| sex   | int(5)       | YES  |     | 0       |       |
| groups | varchar(20)  | YES  |     | NULL    |       |
| mail  | char(50)     | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> alter table employee drop mail ;
Query OK, 0 rows affected (0.00 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

图 9.21 把 employee 表中字段 mail 的类型改为 char

使用 drop 删除表 employee 中的 mail 字段,如图 9.22 所示。

```
mysql> alter table employee drop mail ;
Query OK, 0 rows affected (0.00 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> describe employee;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| eno   | varchar(10)  | NO   | PRI | NULL    |       |
| ename | varchar(30)  | NO   |     | NULL    |       |
| sex   | int(5)       | YES  |     | 0       |       |
| groups | varchar(20)  | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

图 9.22 删除表 employee 中的 mail 字段

使用 rename 更改数据库中的表名,将 employee 表名改为 employee_info,如图 9.23 所示。

3. 复制表

复制表语句:

create table 新表名 like 原表名;

复制的内容包括表结构、表中的数据 and 约束。

```
mysql> alter table employee rename to employee_info ;
Query OK, 0 rows affected (0.00 sec)

mysql> show tables;
+-----+
| Tables_in_couman |
+-----+
| employee_info    |
| exam             |
| score            |
+-----+
3 rows in set (0.00 sec)
```

图 9.23 将 employee 表名改为 employee_info

例如:将表 employee_info 复制为另一个表 employee,如图 9.24 所示。

```
mysql> create table employee like employee_info;
Query OK, 0 rows affected (0.00 sec)

mysql> show tables;
+-----+
| Tables_in_couman |
+-----+
| employee          |
| employee_info     |
| exam              |
| score             |
+-----+
4 rows in set (0.00 sec)
```

图 9.24 将表 employee_info 复制为另一个表 employee

4. 删除表

删除表语句:

```
Drop table 表名1[,表名2...];
```

用于删除一个或多个已存在的表,表结构和表中的数据都将被删除。

例如:删除 employee_info 表,如图 9.25 所示。

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

mysql> show tables;
+-----+
| Tables_in_couman |
+-----+
| employee          |
| exam              |
| score             |
+-----+
3 rows in set (0.00 sec)
```

图 9.25 删除 employee_info 表

9.2.3 处理表数据

创建数据库和表后,下一步是处理数据。一般情况下,需要使用 SQL 语言来增加、更新、查询和删除数据库中的记录数据。处理数据使用 SQL 语言,在这里仅做简单介绍。

1. 增加记录

增加记录的语句:

insert into 表名 (字段 1, 字段 2, ..., 字段 n) values (字段 1 的值, 字段 2 的值, ..., 字段 n 的值);

例如:向 employee、exam、score 表中插入记录,如图 9.26 所示。

```
mysql> insert into employee (eno,ename,sex,groups) values(1001,'wangli',1,'dgroup01');
Query OK, 1 row affected (0.00 sec)
mysql> insert into employee (eno,ename,sex,groups) values(1002,'liuning',1,'dgroup01');
Query OK, 1 row affected (0.00 sec)
mysql> insert into employee (eno,ename,sex,groups) values(1003,'zhangqi',0,'dgroup02');
Query OK, 1 row affected (0.00 sec)
mysql> select * from employee;
+-----+-----+-----+-----+
| eno | ename | sex | groups |
+-----+-----+-----+-----+
| 1002 | liuning | 1 | dgroup01 |
| 1001 | wangli | 1 | dgroup01 |
| 1003 | zhangqi | 0 | dgroup02 |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)
mysql> insert into exam (exid,exname) values(10001,'test');
Query OK, 1 row affected (0.00 sec)
mysql> insert into exam (exid,exname) values(10002,'program');
Query OK, 1 row affected (0.00 sec)
mysql> insert into exam (exid,exname) values(10003,'english');
Query OK, 1 row affected (0.00 sec)
mysql> select * from exam;
+-----+-----+
| exid | exname |
+-----+-----+
| 10001 | test |
| 10002 | program |
| 10003 | english |
+-----+-----+
3 rows in set (0.00 sec)
mysql> insert into score (eno,exid,score) values(1001,10001,87);
Query OK, 1 row affected (0.00 sec)
mysql> insert into score (eno,exid,score) values(1001,10002,94);
Query OK, 1 row affected (0.00 sec)
mysql> insert into score (eno,exid,score) values(1002,10003,72);
Query OK, 1 row affected (0.00 sec)
mysql> insert into score (eno,exid,score) values(1003,10001,85);
Query OK, 1 row affected (0.01 sec)
mysql> select * from score;
+-----+-----+-----+-----+
| scid | eno | exid | score |
+-----+-----+-----+-----+
| 1 | 1001 | 10001 | 87 |
| 2 | 1001 | 10002 | 94 |
| 3 | 1002 | 10003 | 72 |
| 4 | 1003 | 10001 | 85 |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

图 9.26 向 employee、exam、score 表中插入记录

其中“select”语句查询出 employee 表中所有记录数据。

2. 更新记录

更新记录的语句：

Update 表名 set 字段名 1=值, 字段名 2=值, ..., 字段名 n=值, where 匹配条件;

例如: 修改 employee 表中 wangli 的姓名为 wanglili, 项目组改为 dgroup02, 如图 9.27 所示。

```
mysql> update employee set ename='wanglili' , groups='dgroup02' where ename='wangli';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> select * from employee;
+-----+-----+-----+-----+
| eno | ename | sex | groups |
+-----+-----+-----+-----+
| 1002 | liuning | 1 | dgroup01 |
| 1001 | wanglili | 1 | dgroup02 |
| 1003 | zhangqi | 0 | dgroup02 |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

图 9.27 修改记录

3. 查询记录

查询记录的语句：

Select 字段名 from 表名 where 匹配条件;

例如: 查询 employee 表中所有员工的姓名, 如图 9.28 所示。

```
mysql> select ename from employee;
+-----+
| ename |
+-----+
| liuning |
| wanglili |
| zhangqi |
+-----+
3 rows in set (0.00 sec)
```

图 9.28 查询记录

4. 删除记录

删除记录的语句：

Delete from 表名 where 匹配条件;

例如: 删除 employee 表中 wanglili 员工的所有信息, 如图 9.29 所示。

```
mysql> delete from employee where ename='wanglili';
Query OK, 1 row affected (0.00 sec)

mysql> select * from employee;
+-----+-----+-----+-----+
| eno | ename | sex | groups |
+-----+-----+-----+-----+
| 1002 | liuning | 1 | dgroup01 |
| 1003 | zhangqi | 0 | dgroup02 |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

图 9.29 删除记录

9.3 项目三:数据库的权限管理与备份

【项目描述】

管理员在创建完成 couman 数据库后,为了数据库的使用安全,对数据库的使用权限进行了授权,具体授权如下:

赋予 dpuser0101 用户对数据库的远程登录、数据查询(select)权限;

赋予 dpuser0102 用户对数据库的远程登录、数据库(couman)的所有权限;

赋予 dpuser0201 用户对数据库的远程登录、数据的增删改查(insert, delete, update select)权限;

赋予 dpuser0202 用户对数据库的所有权限。

同时为了保障数据库的意外恢复,管理员对 couman 数据库做了备份操作。

【构思设计】

在现实工作中,对数据库进行访问权限管理和定期备份是一项极为重要的事情。恰当的权限管理可以保障数据库运行的安全性和稳定性,而对数据库的定期备份也为数据库突发状况提供了保障。本项目主要涉及数据库管理和维护方面的内容,主要知识点如表 9.8 所示。

表 9.8 本项目知识点分析

序号	知识点	详见章节
1	数据库的权限管理	9.3.1
2	数据库的备份与恢复	9.3.2

【实施运行】

操作步骤:

```
# /usr/bin/mysql -h localhost -u root -p //登录 mysql 服务器
```

```
mysql>grant select on *.* to 'dpuser0101'@'%' identified by '123456' with grant option; //赋予 dpuser0101 用户对数据库的远程登录、数据查询(select)权限,详见 9.3.1
```

```
mysql>grant all privileges on couman.* to 'dpuser0102'@'%' identified by '123456' with grant option; //赋予 dpuser0102 用户对数据库的远程登录、数据库(couman)的所有权限,详见 9.3.1
```

```
mysql>grant select,insert,update,delete on *.* to 'dpuser0201'@'%' identified by '123456' with grant option; //赋予 dpuser0201 用户对数据库的远程登录、数据的增删改查(insert, delete, update select)权限,详见 9.3.1
```

```
mysql>grant all privileges on *.* to 'dpuser0202'@'%' identified by 'mysql' with grant option; //赋予 dpuser0202 用户对数据库的所有权限,详见 9.3.1
```

```
mysql> /usr/bin/mysqldump couman --user=root --password=mysql >db01.sql //将 couman 数据库备份为 db01.sql,详见 9.3.2
```


9.3.1 数据库的权限管理

1. 数据库权限介绍与设置

当首次安装 MySQL 时,MySQL 的安装程序会自动创建一个名为 mysql 的数据库,该数据库中包含 5 个与数据库授权相关的表(user、db、host、tables_priv、columns_priv),这 5 个授权表决定用户连接服务器及操作的权限。表 9.9 说明了 mysql 数据库中这 5 个授权表的功能。

表 9.9 mysql 数据库中授权表

表名	说明
user	定义了允许连接数据库服务器的用户名、口令和可以连接的主机
db	定义了连接到数据库服务器的用户可以使用的数据库以及在数据库中能进行的操作
host	如果 db 表中 Host 字段为空,由 host 表定义可以连接到数据库服务器的主机,并且不同的主机对指定的数据库可以有不同的权限
tables_priv	定义了连接到数据库服务器的用户可以访问具体的表,对表中可执行的操作
columns_priv	定义了连接到数据库服务器的用户可以访问表中的具体字段,对字段可执行的操作

当用户登录数据库时,会根据这几张表中的内容来决定用户的访问权限,决定过程如下:首先从 user 表中的 host、user、password 这 3 个字段中判断连接的 IP、用户名、密码是否存在表中,存在则通过身份验证;然后通过权限验证。进行权限分配时,按照 user、db、tables_priv、columns_priv 的顺序进行分配。即先检查全局权限表 user,如果 user 中对应的操作权限为 Y,则此用户对所有数据库的该操作权限都为 Y,将不再检查 db、tables_priv 和 columns_priv 中的权限;如果为 N,则到 db 表中检查此用户对应的具体数据库,并得到 db 中为 Y 的权限;如果 db 中为 N,则检查 tables_priv 中此数据库对应的具体表,取得表中的权限 Y,依此类推。

对于表 host、db 和 user,除了可以用于绑定用户、口令和主机,还有若干字段用于对数据库访问权限的设置,表 9.10 描述了可设置权限的字段。

表 9.10 可设置权限的字段

字段	权限名	说明
Select_priv	SELECT	查询操作
Insert_priv	INSERT	插入操作
Update_priv	UPDATE	更新操作
Delete_priv	DELETE	删除操作
Create_priv	CREATE	创建操作
Drop_priv	DROP	删除数据库和表
Reload_priv	RELOAD	重载/刷新 MySQL 服务器
Shutdown_priv	SHUTDOWN	关闭运行的 MySQL 服务器

(续表)

字段	权限名	说明
Process_priv	PROCESS	跟踪 MySQL 服务器上的一项活动
Grant_priv	GRANT	授权其他用户享有的权限
File_priv	FILE	在服务器上读和写文件
Index_priv	INDEX	创建、编辑和删除索引
Alter_priv	ALTER	修改表
References_priv	REFERENCES	创建、编辑和删除外键
Show_db_priv	SHOW DATABASES	浏览服务器上可使用的数据库
Super_priv	SUPER	执行管理命令
Create_tmp_table_priv	CREATE TEMPORARY TABLES	创建临时表
Lock_tables_priv	LOCK TABLES	锁定创建和删除表权限
Execute_priv	EXECUTE	执行被保存的程序
Repl_slave_priv	REPLICATION SLAVE	在复制内容中读主二进制日志
Repl_client_priv	REPLICATION CLIENT	在复制内容中申请主、副信息

MySQL 数据库提供了两种方法修改授权表中的访问权限。可以使用 Insert、Update 和 Delete 等 SQL 语句手工修改权限表中的信息或者使用 grant 和 revoke 命令。对比两种方法，后一种更加简单而方便。

grant 命令用于授予权限，而 revoke 命令用户撤销权限。

grant 命令的格式：

grant 权限级别 [(字段名)] on 数据库名.表名 to 用户名@域名或 IP 地址 [Identified by 'password'] [with grant option];

revoke 命令的格式：

revoke 权限级别 [(字段名)] on 数据库名.表名 from 用户名@域名或 IP 地址；

其中权限级别如表 9.11 所示，如果针对数据库所有内容或所有访问地址进行设置可以使用“%”。

表 9.11

MySQL 数据库的权限级别

权限级别	操作对象	权限说明
CREATE	数据库、表或索引	创建数据库、表或索引权限
DROP	数据库或表	删除数据库或表权限
GRANT OPTION	数据库、表或保存的程序	赋予权限选项
ALTER	表	更改表，比如添加字段、索引等
DELETE	表	删除数据权限
INDEX	表	索引权限

(续表)

权限级别	操作对象	权限说明
INSERT	表	插入权限
SELECT	表	查询权限
UPDATE	表	更新权限
CREATE VIEW	视图	创建视图权限
SHOW VIEW	视图	查看视图权限
ALTER ROUTINE	存储过程	更改存储过程权限
CREATE ROUTINE	存储过程	创建存储过程权限
EXECUTE	存储过程	执行存储过程权限
FILE	服务器主机上的文件访问	文件访问权限
CREATE TEMPORARY TABLES	服务器管理	创建临时表权限
LOCK TABLES	服务器管理	锁表权限
CREATE USER	服务器管理	创建用户权限
PROCESS	服务器管理	查看进程权限
RELOAD	服务器管理	执行 flush-hosts, flush-logs, flush-privileges, flush-status, flush-tables, flush-threads, refresh, reload 等命令的权限
REPLICATION CLIENT	服务器管理	复制权限
REPLICATION SLAVE	服务器管理	复制权限
SHOW DATABASES	服务器管理	查看数据库权限
SHUTDOWN	服务器管理	关闭数据库权限
SUPER	服务器管理	执行 kill 线程权限

权限的分布用于描述针对表可以设置的权限,针对列可以设置的权限等,如表 9.12 所示。

表 9.12 权限的分布

权限分布	可能设置的权限
表权限	Select, Insert, Update, Delete, Create, Drop, Grant, References, Index, Alter
列权限	Select, Insert, Update, References
过程权限	Execute, Alter Routine, Grant

通常在管理数据库权限时可以基于以下几点考虑来设置用户权限:

- (1) 只赋予能满足用户需要的最小权限,防止用户操作不当。比如用户只是需要查询数据信息,则只赋予 select 权限,不用给用户赋予 update、insert 或 delete 权限。
- (2) 创建用户的时候限制用户的登录主机,一般是限制成指定 IP 或者内网 IP 段。
- (3) 如果安装完数据库自动创建了无密码用户,则最好将这些用户删除。
- (4) 为每个用户设置满足密码复杂度的密码。

(5) 定期清理不需要的用户,回收权限或者删除用户。

2. 设置数据库权限案例

(1) grant 命令使用

创建一个只允许从本地登录的用户 jack,并允许将权限赋予别的用户,密码为:jack.

```
mysql> grant all privileges on *.* to jack@'localhost' identified by "jack" with grant option;
```

说明:

ALL PRIVILEGES 表示所有权限,也可以使用 select、update 等具体权限。

ON 用来指定权限针对的数据库和表。

. 中前面的 * 号用来指定数据库名,后面的 * 号用来指定表名,* 表示所有。

TO 表示将权限赋予某个用户。

jack@'localhost'表示 jack 用户,@后面接限制的主机,可以是 IP、IP 段、域名以及%,% 表示任何地方。

IDENTIFIED BY 指定用户的登录密码。

WITH GRANT OPTION 表示该用户可以将自己拥有的权限授权给别人。

可以使用 grant 命令重复给用户添加权限,权限叠加,比如先给用户添加一个 select 权限,然后又给用户添加一个 insert 权限,那么该用户就拥有了 select 和 insert 权限。

(2) 刷新权限

使用 flush 命令可以使权限生效,尤其是当权限表 user、db、host 等做了 update 或者 delete 更新等操作时。

```
mysql> flush privileges;
```

(3) 查看权限

查看当前用户权限:

```
mysql> show grants;
```

查看 jack 用户的权限:

```
mysql> show grants for 'jack'@'%';
```

(4) 回收权限

回收 jack 用户的权限:

```
mysql> revoke delete on *.* from 'jack'@'localhost';
```

```
Query OK, 0 rows affected (0.01 sec)
```

(5) 对账户重命名

将数据库用户 jack 修改为 jim:

```
mysql> rename user 'jack'@'%' to 'jim'@'%';
```

(6) 删除用户

删除 MySQL 用户 jack:

```
mysql> drop user 'jack'@'localhost';
```

9.3.2 数据库的备份与恢复

在数据库管理的实际工作中,数据库的定期备份是一件至关重要的事情。通常管理员以批处理的方式进行数据库的自动备份,这样一旦数据库发生故障或者误操作时,可以进行数据库

的恢复来保证服务器的正常运行。

MySQL 自身提供了许多命令行工具,例如 `mysql` 命令可以用来与 MySQL 数据库、数据、用户和配置变量进行交互, `mysqladmin` 命令则可以进行各种管理任务,而下面将要介绍的 `mysqldump` 命令可以用于数据库的备份。`mysqldump` 命令不仅可以对一个服务器上的所有数据库进行地毯式的备份,还可以选择性地备份某个数据库,甚至数据库中指定的某些表。

`mysqldump` 命令格式:

`mysqldump` 备份数据库的名字 `--user=用户名 --password=密码`>备份的文件名

例如备份数据库 `couman`,备份为文件 `db01.sql`。

```
# /usr/bin/mysqldump couman --user=root --password=mysql >db01.sql
```

备份数据库中指定的表:

`mysqldump` 备份数据库的名字 表名 1 (表名 2 ...)--user=用户名 --password=密码 >备份的文件名

从备份文件中恢复数据库和表:

```
mysql -u 用户名 -p 数据库名 <备份文件
```

输入命令后,根据提示输入相应密码就可以完成数据库的恢复。需要注意,在恢复数据库之前首先要创建数据库,然后再恢复。

9.4 习题与项目训练

1. 选择题

(1)修改 MySQL 服务器登录密码的命令是()。

- A. `passwd`
- B. `mysql`
- C. `mysqladmin`
- D. `chmod`

(2)登录 MySQL 服务器使用()命令。

- A. `passwd`
- B. `mysql`
- C. `mysqladmin`
- D. `chmod`

(3)重新启动 MySQL 服务器的命令()。

- A. `service httpd restart`
- B. `service named restart`
- C. `service network restart`
- D. `service mysqld restart`

2. 按要求进行配置:

在 Linux 系统下安装 MySQL 服务器,创建 `news` 数据库,二个数据表,表结构如下:

新闻类型表(`ntype`)字段:序号(`tid`),类型名称(`tname`),其中序号(`tid`)字段为主键;

新闻表(`new`)字段:序号(`nid`)、标题(`head`)、类型序号(`tid`)、发布时间(`sstart`)、有效期(`limit`),其中序号(`nid`)字段为主键,类型序号(`tid`)字段为外键。

向每个表中至少插入三条合法记录。

为 `lucy` 用户赋予对该数据库的查询权限,为 `tom` 用户赋予对该数据库所有数据表的操作权限。

对 `news` 数据库进行备份并恢复成 `news_info` 数据库。

3. 按要求进行配置:

管理员准备在 MySQL 数据库中创建课程管理网站需要的数据库(couman)和相关表。数据表包括:学生信息表(student)、课程描述表(course)、学生选课表(stucou),各个表的结构如下:

学生信息表(student)字段:序号(sid)、学号(sno)、姓名(sname)、性别(sex)、班级(class);

课程描述表(course)字段:序号(cid)、课程号(cno)、课程名(cname);

学生选课表(stucou)字段:序号(scid)、学号(sno)、课程号(cno)、成绩(score)。

数据表创建成功后,向每个表中至少插入三条合法记录。

赋予 lucy 用户对数据库的远程登录、数据查询(select)权限;

赋予 lily 用户对数据库的远程登录、数据库(couman)的所有权限;

赋予 john 用户对数据库的远程登录、数据的增删改查(insert,delete,update select)权限;

赋予 tom 用户对数据库的所有权限。

同时为了保障数据库的意外恢复,管理员对 couman 数据库做了备份操作。

东软电子出版社

第 10 章 Apache 服务器配置

学习引导

【本章概述】

Web 服务器的功能是在网络上发布 Web 站点。Web 服务器的种类很多,其中 Apache 服务器是目前使用最广泛的一款 Web 服务器。

本章以五个项目着重阐述 Apache 服务器的安装过程、配置文件的使用方法、Web 站点的创建和部署过程以及虚拟主机的设置和测试方法。

【教学重点与难点】

重点:使用 Apache 服务器部署 Web 站点。

难点:配置虚拟主机的方法。

10.1 项目一:Apache 服务器的安装与测试

【项目描述】

管理员将根据公司设计部需要发布项目考核信息网站的要求,在服务器上部署该网站。首先要安装 Web 服务器,然后对服务器进行必要测试:服务器 IP 地址为:192.168.137.5,测试网站首页 index.html,其内容是“This is project exam web.”。

测试效果:在浏览器中输入 `http://192.168.137.5`,即可看到 index.html 的内容。

【构思设计】

在本项目中,管理员首先需要安装 Apache 软件,软件的安装可以使用第 5 章中的 rpm 软件包管理来实现。然后对 Apache 进行必要的测试。由于 `/var/www/html` 目录是 Apache 服务器的默认站点根目录,因此可以在 `/var/www/html` 目录下创建 Web 站点的首页 index.html 并使用 vim 编辑器编写内容。最后通过浏览器查看 Web 站点的首页。相关知识点如表 10.1 所示。