

第1章 数据库概述

简而言之，数据库（DataBase）就是一个存储数据的仓库。为了方便数据的存储和管理，它将数据按照特定的规律存储在磁盘上。通过数据库管理系统，可以有效的组织和管理存储在数据库中的数据。如今，已经存在了Oracle、SQL Server、MySQL等诸多优秀的数据库。在这一章中将讲解的内容包括。

数据存储方式

数据库在开发中作用

数据库访问技术

MySQL数据库的介绍

数据库泛型

SQL语言

常见数据库系统

如何学习数据库

1.1 数据库理论基础

数据库能够将数据按照特定的规律组织起来。那么，数据是如何存储的、数据库要遵守什么规则、数据库在什么地方使用，这些都是首先要了解的问题。本节将为读者介绍这些问题。

1.1.1 数据存储方式

如今数据库已经无处不在了。一个网站需要有数据库来存储数据。一个学校需要用数据库来存储学生和教师的信息。一个公司需要用数据库来存储员工的信息和公司的资料。要学习数据库，必须先要了解数据库是如何存储数据的。本小节将为读者介绍数据的存储方式。数据存储分为三个阶段：人工管理阶段、文件系统阶段和数据库系统阶段。

- 1. 人工管理阶段**
- 2. 文件系统阶段**
- 3. 数据库系统阶段**

1.1.2 数据库泛型

数据库泛型就是数据库应该遵循的规则。数据库泛型也称为范式。目前关系数据库最常用的四种范式分别是：第一范式（1NF）、第二范式（2NF）、第三范式（3NF）和BCN范式（BCNF）。本小节将为读者简单的介绍一下范式的内容。

在设计数据库时，数据库需要满足的最低要求的范式是第一范式。第一范式的要求表中不能有重复字段，并且每个字段不能再拆分。如果一个数据库连第一范式都不满足的话，那就不能称之为关系数据库了。只有满足了第一范式的要求，才可以在这个关系数据库中创建表。

在满足第一范式的基础上，可以将数据库中进行进一步的细化。细化后可以使数据库满足第二范式的要求。依次进行细化，可以得到第三范式、BCN范式。

1.1.3 数据库在开发中作用

现在大部分的管理系统和软件都需要使用数据库来存储数据。在开发过程中，数据库已经成为必不可少的一部分。本小节将为读者简单介绍一下数据库在开发中的作用。

在软件开发过程中，经常需要使用数据库来存储数据。比如，一个学校的学生管理系统就需要数据库来存储学生的学籍信息、考试信息、教师的信息、课程信息等。再比如，银行的管理系统也需要数据库来存储信息。用户的银行账户、存款量、存款和取款的记录等信息都是存储在数据库中的。当用户向自己的账户里存款时，管理系统会在数据库中更新该用户的存款量。

1.2 数据库技术构成

数据库技术的出现是为了更加有效地管理和存取大量的数据资源。简单的讲，数据库技术包括数据库系统、SQL语言、数据库访问技术等。本节将为读者介绍数据库技术的内容。

1.2.1 数据库系统

很多读者认为数据库就是数据库系统（ DataBase System，简称为DBS）。其实，数据库系统的范围比数据库大很多。数据库系统不是一个单纯的数据库，而是由数据库、数据库管理系统、应用开发工具构成。很多时候，数据库管理员和用户可以当成数据库系统的一份子。本小节将为读者介绍数据库系统的内容。

前面的章节对数据库已经进行了简单的介绍，数据库就是存储数据的地方。数据库管理系统（ DataBase Management System，简称为DBMS）是用来定义数据、管理和维护数据的软件。它是数据库系统的一个重要的组成部分。应用系统是需要使用数据库的软件。比如学员管理系统就是一个应用系统，这个应用系统需要数据库来管理它的数据。应用开发工具就是用来开发应用系统的。

1.2.2 SQL语言

SQL (Structured Query Language) 语言的全称是结构化查询语言。数据库管理系统通过SQL语言来管理数据库中的数据。本小节将为读者介绍SQL语言的知识。

SQL语言分为三个部分：数据定义语言 (Data Definition Language , 简称为DDL)、数据操作语言 (Data Manipulation Language , 简称为DML) 和数据控制语言 (Data Control Language , 简称为DCL)。

- **DDL语句：**
- **DML语句：**
- **DCL语句：**

1.2.3 数据库访问技术

应用系统中，程序语言需要使用数据库访问技术来访问数据库。只有使用了数据库访问技术，程序中嵌入的SQL语句才会起作用。不同程序语言访问数据库的方式是不一样的。本小节将为读者简单讲解各种数据库访问技术。

JDBC是一种用于执行SQL语句的Java API。通过JDBC可以访问多种关系数据库。JDBC由一组用Java语言编写的类和接口组成。使用JDBC时，必须要安装驱动程序Connector/J。

PHP中为程序员提供了MySQL功能模块，PHP 5以后开始提供mysqli接口。PHP可以通过MySQL功能模块和mysqli接口来访问MySQL数据库。

1.3 MySQL基础

现在数据库的版本很多，Oracle、DB2、SQL Server都是很优秀的商业数据库。同时，还有MySQL、PostgreSQL都是很优秀的开源数据库。MySQL数据库的使用已经非常广泛了。本节将为读者介绍常见的数据库系统、以及MySQL数据库的基本知识。

1.3.1 常见数据库系统

如今已经存在很多优秀的商业数据库，如甲骨文（Oracle）公司的Oracle数据库、IBM公司的DB2数据库、微软公司的SQL Server数据库和Access数据库。同时，还有很多优秀的开源数据库，如MySQL数据库、PostgreSQL数据库等。本小节将为读者介绍这些常见的数据库。

1. 甲骨文的Oracle
2. IBM的DB2
3. 微软的Access和SQL Server
4. 开源PostgreSQL
5. 开源MySQL

1.3.2 为什么要使用MySQL

如今很多大型网站已经选择MySQL数据库来存储数据。那么，MySQL到底有什么优势呢？本小节将为读者介绍选择MySQL数据库的原因。MySQL数据库的使用以及非常广泛，尤其是在Web应用方面。由于MySQL数据库发展势头迅猛，Sun公司于2008年收购了MySQL数据库。这笔交易的收购价格高达10亿美元。这足以说明MySQL数据库的价值。MySQL数据库有很多的优势，下面总结了其中几个优势：

1. MySQL是开放源代码的数据库
2. MySQL的跨平台性
3. 价格优势
4. 功能强大且使用方便

1.3.3 MySQL版本和获取

MySQL数据库可以在Windows、UNIX、Linux和Mac OS等操作系统上运行。因此，MySQL有不同操作系统的版本。而且，根据发布的先后顺序，现在已经在开发MySQL的6.0版了。本小节将为读者介绍MySQL的版本和如何下载MySQL。

根据MySQL的开发情况，可以将MySQL分为Alpha、Beta、Gamma和Generally Available (GA) 等版本。这几种版本的说明如下：

- Alpha
- Beta
- Gamma
- Generally Available (GA)

1.4 如何学习数据库

数据库已经成为软件系统的一部分，那么学好数据库将是软件开发的一个必要条件。如何才能学好数据库，这个问题没有确切的答案。笔者在本节跟大家分享一下自己学习的经验。

学好数据库，最主要的是要多练习。笔者将自己学习数据库的方法总结如下：

- 1. 多上机实践**
- 2. 多编写SQL语句**
- 3. 通过Java等程序语言来操作数据库**
- 4. 数据库理论知识不能丢**

1.5 常见问题及解答

1. 如何选择数据库？
2. 如何选择MySQL版本？

1.6 小结

本章介绍了数据库和MySQL的基础知识。通过本章的学习，希望读者对数据库、MySQL数据库和SQL语言等知识有所了解。而且，希望读者能够了解常用的数据库系统。关于数据库泛型的知识难度比较大，读者只要能够了解相关知识就行了。下一章将介绍在Windows操作系统下安装和配置MySQL。如果读者是Linux用户，可以跳过下一章直接学习第3章。

1.7 本章习题

- 1.数据存储的发展过程经历了哪几个阶段？
- 2.常用数据库系统有哪些？
- 3.MySQL数据库如何分类？

第2章 Windows平台下安装与配置MySQL

在Windows系列的操作系统下，MySQL数据库的安装包分为图形化界面安装和免安装（noinstall）这两种安装包。这两种安装包的安装方式不同，而且配置方式也不同。图形化界面安装包有完整的安装向导，安装和配置很方便。免安装的安装包直接解压即可使用，但是配置起来很不方便。在这一章中将讲解的内容包括。

- 安装MySQL数据库
- 配置MySQL数据库
- 常用图形管理工具介绍
- 配置和使用免安装的MySQL数据库

2.1 安装与配置MySQL

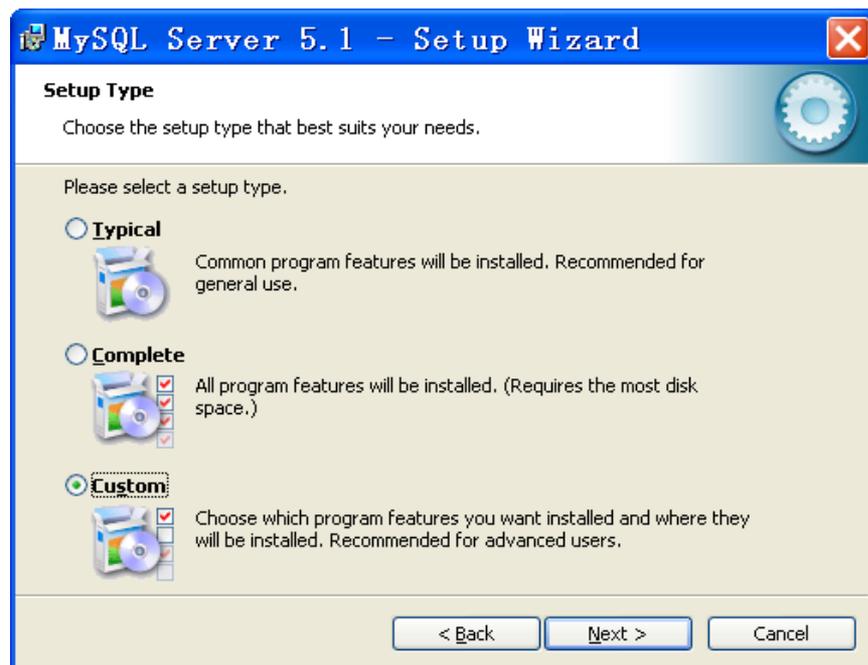
Windows操作系统下，可以通过图形化方式安装MySQL，也可以使用免安装的MySQL软件包。图形化方式有很完善的安装向导，根据安装向导的说明安装即可。本节将为读者介绍通过安装向导来安装和配置MySQL的方法。

2.1.1 安装MySQL

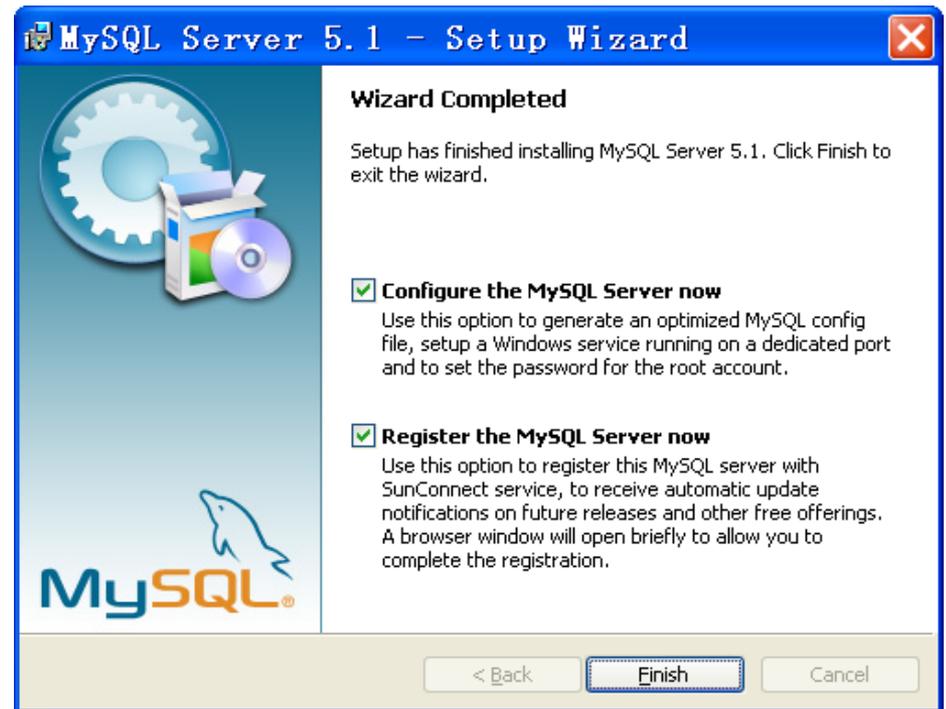
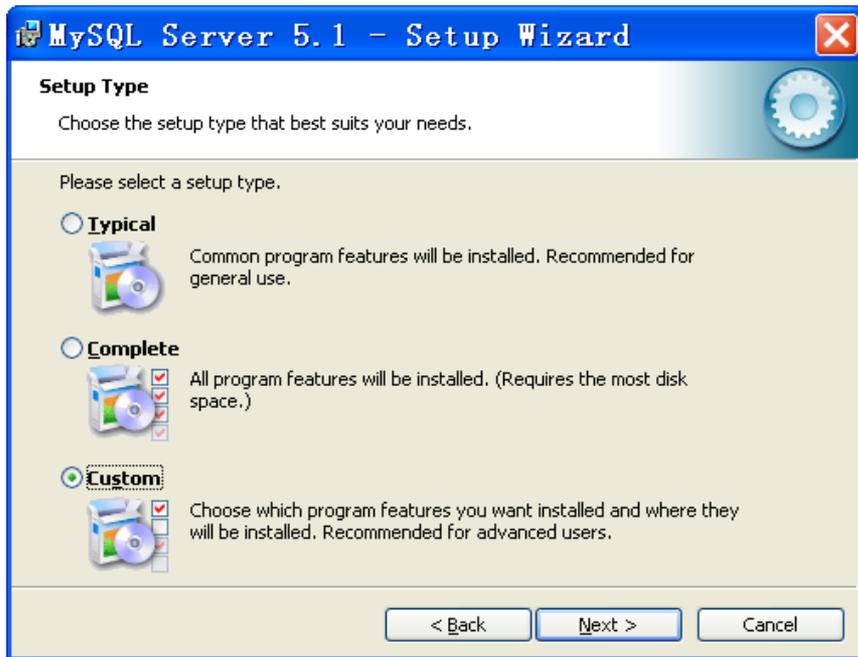
MySQL图形化安装包有一个完整的安装向导，根据安装向导可以很方便的安装MySQL数据库。在Windows操作系统下，有两种MySQL图形化安装包。这两种安装包分别是“Windows Essentials”和“Windows MSI Installer”。前者包含了MySQL中最主要和最常用的功能，但是不包含一些不常用的功能。后者包含了MySQL全部功能，包括不常用的功能。本小节将介绍MySQL的图形化安装包的安装过程。

MySQL下载完成后，在软件下载目录下进行安装。

2.1.1 安装MySQL

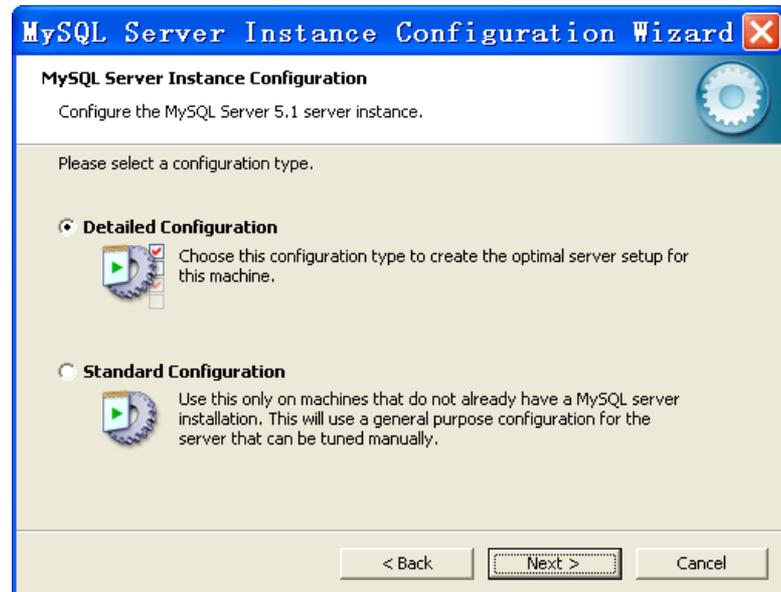
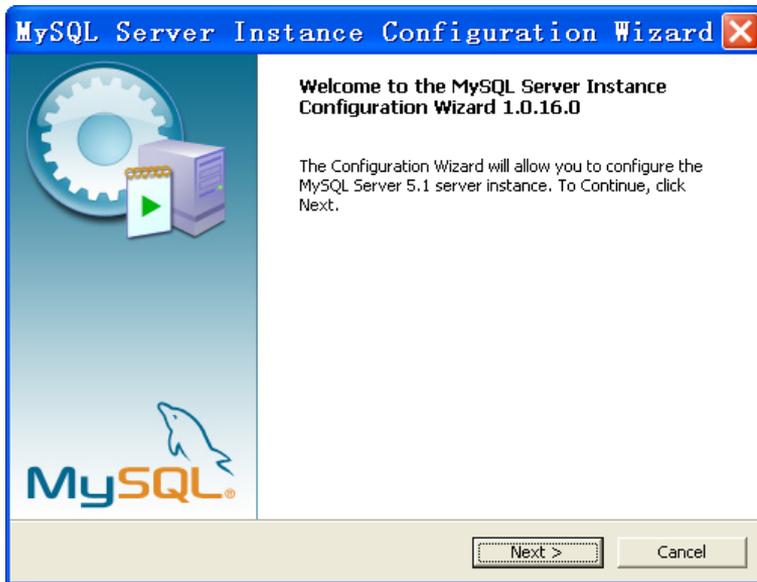


2.1.1 安装MySQL

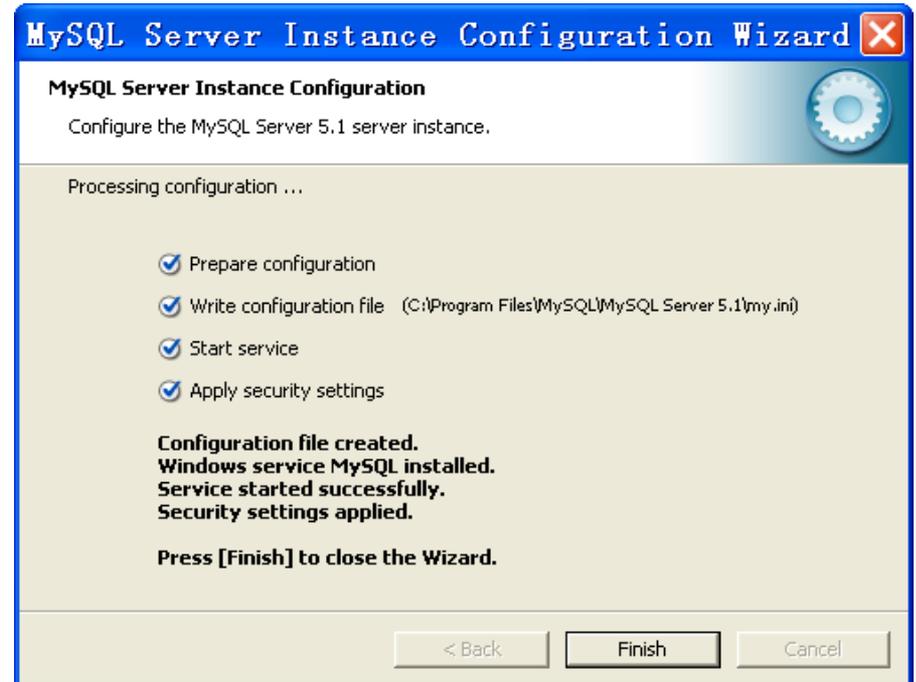
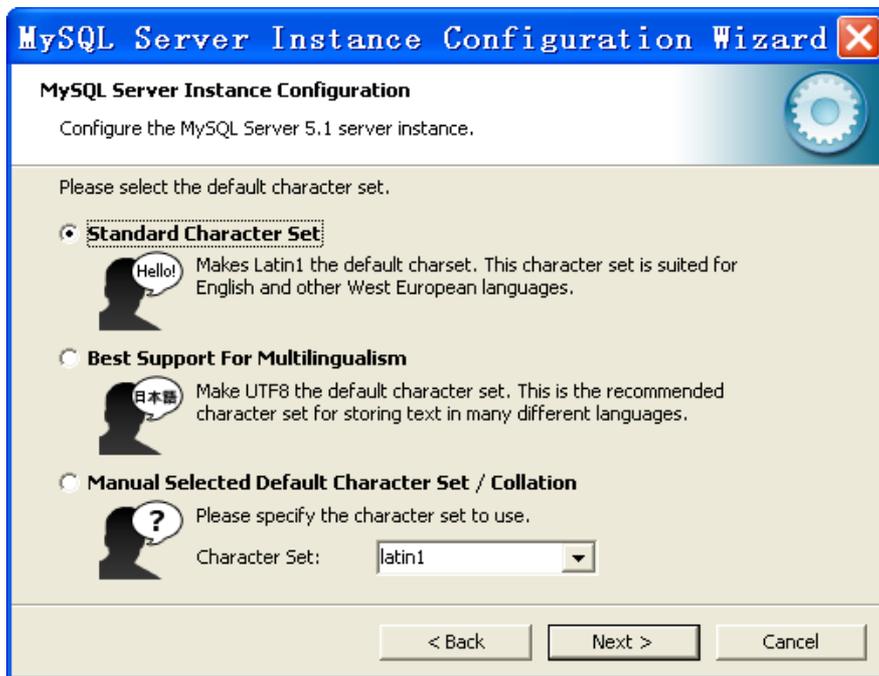


2.1.2 配置MySQL

安装完成时，选上了Configure the MySQL Server now 选项，图形化安装向导将进入MySQL配置欢迎界面。通过配置向导，可以设置MySQL数据库的各种参数。本小节将为读者介绍配置MySQL的内容。



2.1.2 配置MySQL



2.2 启动服务并登录MySQL数据库

MySQL数据库是分为服务器端（Server）和客户端（Client）两部分。只有服务器端的服务开启以后，才可以通
过客户端来登录到MySQL数据库。本节将为读者介绍启动
服务和登录MySQL数据库的方法。

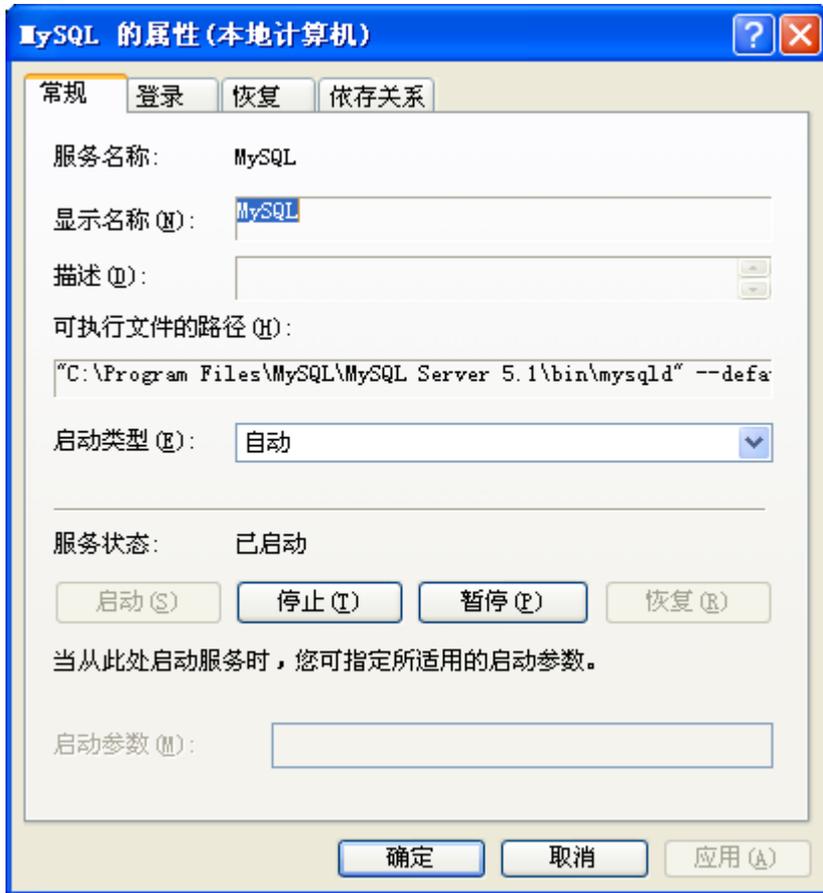
2.2.1 启动MySQL服务

只有启动MySQL服务，客户端才可以登录到MySQL数据库。在Windows操作系统上，可以设置自动启动MySQL服务，也可以手动来启动MySQL服务。本小节将为读者介绍启动MySQL服务的方法。

可以在【MySQL的属性】中设置服务状态，可以将服务状态设置为【启动】、【停止】、【暂停】和【恢复】。而且还可以设置启动类型，在启动类型处的下拉菜单中可以选择【自动】、【手动】和【已禁用】。这三种启动类型的说明如下：

- **【自动】**
- **【手动】**
- **【已禁用】**

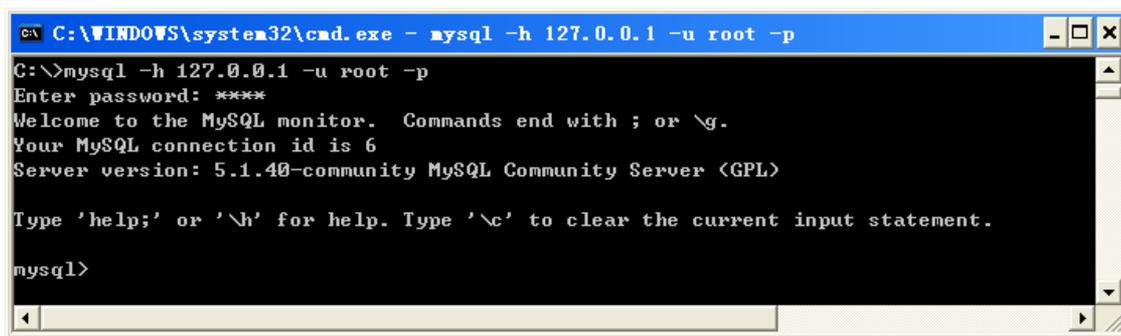
2.2.1 启动MySQL服务



2.2.2 登录MySQL数据库

当MySQL服务开启后，用户可以通过客户端来登录MySQL数据库。Windows操作系统下可以在DOS窗口中登录MySQL数据库。登录可以通过DOS命令完成。本小节将为读者介绍用命令方式登录MySQL数据库的方法。

Windows操作系统下要使用DOS窗口来执行命令，读者可以在【开始】|【运行】中打开一个【运行】对话框。



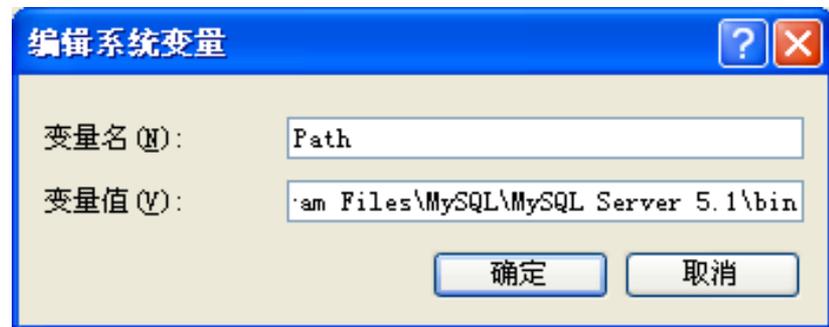
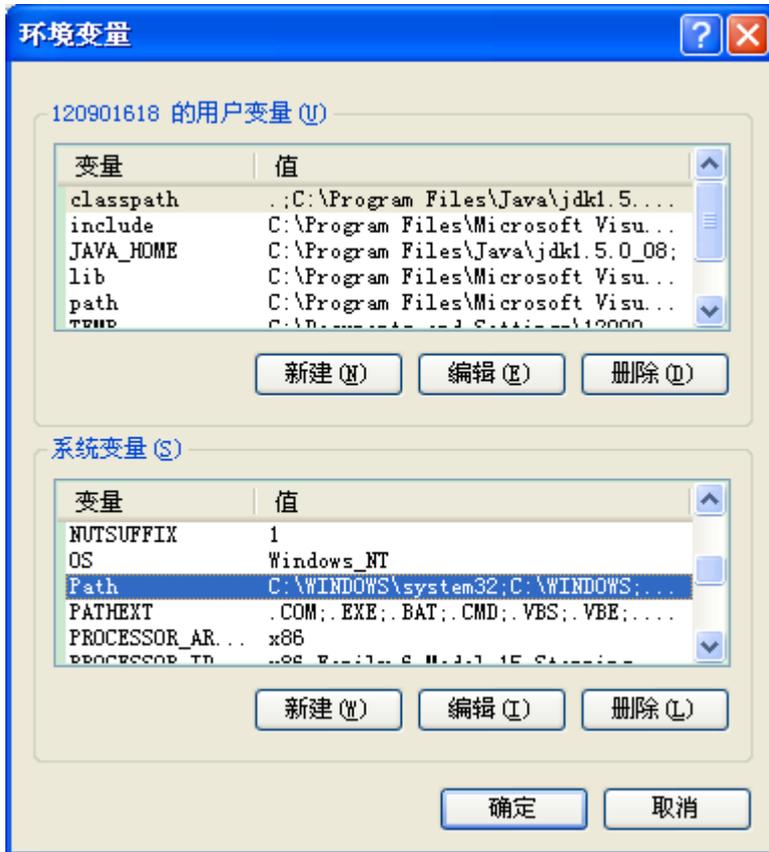
2.2.3 配置Path变量

如果MySQL的应用程序的目录没有添加到Windows系统的Path中，可以手工的将MySQL的目录添加到Path中。本小节将介绍配置Path变量的方法。

将MySQL的应用程序的目录添加到Windows系统的Path中，可以使以后的操作更加方便。例如，可以直接从运行对话框中输入MySQL数据库的命令。而且，以后在编程时也会更加方便。配置Path路径很简单，只要将MySQL的应用程序的目录添加到系统的Path变量中就可以了。步骤如下：

- (1) 右击【我的电脑】图标，选择【属性】命令。
- (2) 在系统变量中选中Path变量，然后单击【编辑】按钮进入编辑环境变量的对话框。
- (3) 可以在【变量值】中添加MySQL的应用程序的目录。已经存在的目录用分号隔开。

2.2.3 配置Path变量



2.3 更改MySQL的配置

MySQL数据库安装好了以后，可能根据实际情况更改MySQL的某些配置。一般可以通过两种方式来更改。一种是通过配置向导来更改配置，另一种是手工来更改配置。本节将为读者详细介绍更改MySQL配置的方法。

2.3.1 通过配置向导来更改配置

MySQL提供了一个人性化的配置向导，通过配置向导可以很方便进行配置。对于初级用户而言，这种配置的方式很容易使用。本小节将为读者介绍使用配置向导来更改配置的方法。

。

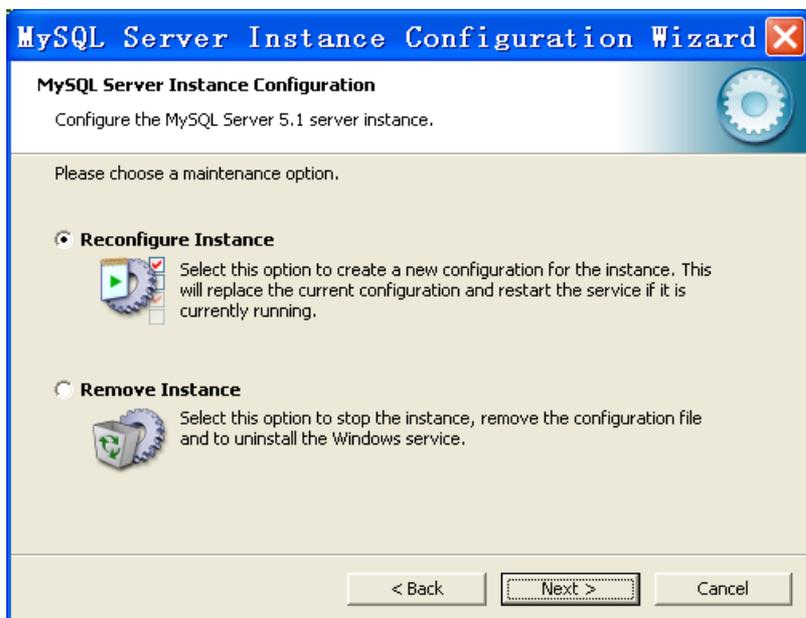
MySQL的配置向导在【开始】|【所有程序】|MySQL|MySQL Server 5.1中。在该位置可以看到MySQL Command Line Client、MySQL Server Instance Config Wizard和SunInventory Registration。这三个内容的介绍如下：

MySQL Command Line Client是MySQL客户端的命令行，通过该命令行可以登录的到MySQL数据库中。然后可以在该命令行中执行SQL语句、操作数据库等。

MySQL Server Instance Config Wizard是配置向导。通过该向导可以配置MySQL数据库的各种配置。

SunInventory Registration是注册的网页链接。

2.3.1 通过配置向导来更改配置



2.3.2 手工更改配置

用户可以通过修改MySQL配置文件方式来进行配置。这种配置方式更加灵活，但是相对来说比较难。初级用户可以通过手工配置的方式来学习MySQL的配置。这样可以了解的更加透彻。本小节将向读者介绍手工更改配置的方法。

在进行手工配置之前，读者需要对MySQL的文件有所了解。前面已经介绍过，MySQL的文件安装在C:\Program Files\MySQL\MySQL Server 5.1这个目录下。而MySQL数据库的数据文件安装在C:\Documents and Settings\All Users\Application Data\MySQL\MySQL Server 5.1\data目录下。

2.4 MySQL常用图形管理工具

MySQL图形管理工具可以在图形界面上操作MySQL数据库。在命令行中操作MySQL数据库时，需要使用很多的命令。而图像管理工具则只是使用鼠标点击即可，这使MySQL数据库的操作更加简单。本节将介绍一些常用的MySQL图形管理工具。

MySQL的图形管理工具很多。常用的有MySQL GUI Tools、phpMyAdmin、Navicat等。通过这些图像管理工具，可以使MySQL的管理更加的方便。每种图形管理工具各有特点，下面分别进行简单的介绍。

- 1 . MySQL GUI Tools**
- 2 . phpMyAdmin**
- 3 . Navicat**
- 4 . SQLyog**

2.5 使用免安装的MySQL

Windows操作系统下有免安装的MySQL软件包。用户直接解压这个软件包，进行简单的配置就可以使用了。免安装包省略了安装过程，使用起来也很方便。本小节将为读者介绍一下免安装的MySQL的使用。

读者可以在MySQL官方网站上下载免安装的MySQL软件包。现在的最新的稳定版本是5.1.40。下载后进行设置的操作如下：

- 1. 解压软件包**
- 2. 创建my.ini文件**
- 3. 修改my.ini文件**
- 4. 设置MySQL服务**
- 5. 配置系统Path变量**
- 6. 启动和关闭服务**

2.6 上机操作

1. 通过图形化方式安装MySQL数据库
2. 配置免安装的MySQL

2.7 常见问题及解答

1. 如何选择字符集？
2. 如何删除MySQL数据库？

2.8 小结

本章主要介绍了在Window操作系统上安装和配置MySQL数据库的方法。通过本章的学习，读者需要掌握下载MySQL数据库、使用图形化方式安装MySQL数据库、配置MySQL数据库、启动MySQL服务和登录MySQL数据库等内容。使用免安装的MySQL软件包和手动配置MySQL数据库是本章的难点。读者在学习本章时一定要结合实践，只有在安装与配置的过程中才会真正掌握本章的内容。下一章将为读者介绍在Linux操作系统下安装和配置MySQL数据库的方法。

2.9 本章习题

- 1.练习使用图形化方式安装MySQL数据库。
- 2.练习使用配置向导配置MySQL数据库。
- 3.练习使用免安装的MySQL软件包安装。
- 4.练习通过手工修改my.ini文件的方式更改配置。

第3章 Linux平台下安装与配置MySQL

在Linux系列的操作系统下，一般都使用命令来安装MySQL数据库。因为Linux操作系统的发行版比较多，所以相应的MySQL版本也比较多。相同Linux发行版也有不同的MySQL软件包。读者需要根据自己的操作系统的版本来下载和安装不同的MySQL版本。本章将向读者介绍的内容如下：

- Linux操作系统下的MySQL版本介绍
- 安装和配置MySQL的RPM包
- 安装和配置MySQL的二进制包
- 安装和配置MySQL的源码包

3.1 Linux操作系统下的MySQL版本介绍

Linux操作系统的发行版很多，不同Linux发行版的MySQL版本是不同的。MySQL数据库主要支持的Linux版本是Red Hat Enterprise Linux和SUSE Linux Enterprise Server。这两个版本下也有不同的安装包。本小节将为读者介绍一些Linux操作系统下的MySQL版本的内容。

Linux操作系统的MySQL软包一般分为三类，分别是RPM软件包，二进制软件包和源码包。这三类MySQL软件包的介绍如下：

RPM软件包：

二进制包：

源码包：

1 . Red Hat Enterprise Linux发行版

2 . SUSE Linux Enterprise Server发行版

3.1 Linux操作系统下的MySQL版本介绍

Linux版本	推荐安装的MySQL版本
Red Hat Enterprise Linux	RPM软件包
SUSE Linux Enterprise Server	RPM软件包
Fedora	RPM软件包
Ubuntu	源码包
CentOS	RPM软件包

3.2 安装和配置MySQL的RPM包

大部分Linux版本的MySQL软件是以RPM包的形式发布的。RPM包的安装和卸载都很方便，通过简单的命令就可以实现RPM包的安装与卸载。可以通过手工修改配置文件的方式来进行配置。本小节将介绍MySQL的RPM包的安装和配置。

先必须到MySQL官方网站下载RPM包，网址为<http://dev.mysql.com/downloads/mysql/5.1.html>。在该网站上有很多种类的RPM安装包，读者必须根据自己的操作系统和处理器类型来选择。本书选择Red Hat Enterprise Linux 5 RPM (x86)来安装。

3.2 安装和配置MySQL的RPM包

Red Hat Enterprise Linux 5 RPM (x86) downloads

Server	5.1.40-0	20.3M	Download Pick a mirror MD5: 897095124067232129d776515046434e
Client	5.1.40-0	432.8K	Download Pick a mirror MD5: 470a563f1d3607865e569143ebcc4198
Shared libraries	5.1.40-0	1.8M	Download Pick a mirror MD5: 9496d5218533bd426580f1c9beb6ee87
Shared compatibility libraries (3.23, 4.x, 5.x libs in same package)	5.1.40-0	5.1M	Download Pick a mirror MD5: b82f326e234a3b3a069b660003873386
Headers and libraries	5.1.40-0	10.9M	Download Pick a mirror MD5: d46395024ed7a7c59e439afae1c26f57
Embedded server	5.1.40-0	30.8M	Download Pick a mirror MD5: eb870f104212750af585937637dd4777
Test suite	5.1.40-0	12.1M	Download Pick a mirror MD5: 8a9c39cd50e597fbd6ff51eff8129c8c
Debug information	5.1.40-0	55.2M	Download Pick a mirror MD5: b35631bc366528c957f4334ecbe55e71

3.3 安装和配置MySQL的二进制包

Linux操作系统下有以二进制的形式发布的MySQL软件包。这些二进制的MySQL软件包比RPM包要灵活，但是安装没有RPM包那么容易。本小节将为读者介绍MySQL的二进制包的安装与配置。

先必须到MySQL官方网站下载二进制包，网址为<http://dev.mysql.com/downloads/mysql/5.1.html>。在该网站上单击Linux (non RPM packages)链接，就可以跳转到下载二进制包的位置。

3.3 安装和配置MySQL的二进制包

Linux (non RPM packages) downloads [\(platform notes\)](#)

Linux (x86)	5.1.40	120.8M	Download Pick a mirror MD5: cab41b122e8f1f558aade44d44eb0679 Signature
Linux (AMD64 / Intel EM64T)	5.1.40	125.6M	Download Pick a mirror MD5: 31a82a9938c02a3929c47cb016bd5784 Signature
Linux (IA64)	5.1.40	126.0M	Download Pick a mirror MD5: 96bf35f72357dde331db83981572eade Signature
Linux (POWER / PowerPC, 32-bit)	5.1.40	187.8M	Download Pick a mirror MD5: 7dc8360802e37f746a8ea60b39fa8c82 Signature
Linux (S/390X)	5.1.40	123.6M	Download Pick a mirror MD5: 02cf1dc5c542487ad863bb5751611ede Signature

3.4 安装和配置MySQL的源码包

Linux操作系统下有以源码的形式发布的MySQL软件包。这样的软件包中是MySQL的源代码，需要用户自己进行编译。这种MySQL软件包的灵活性最高，用户可以根据自己的需要进行定制。而且，感兴趣的用户可以查看MySQL的源代码。但是编译代码耗费的时间比较长。本小节将为读者介绍MySQL的源码包的安装与配置。

先必须到MySQL官方网站下载源码包，网址为<http://dev.mysql.com/downloads/mysql/5.1.html>。在该网站上单击Source链接，就可以跳转到下载源码包的位置。

3.4 安装和配置MySQL的源码包

Source downloads

Note that in the more recent MySQL 5.0 and 5.1 releases, Windows binaries are built from the same source as the Unix/Linux source TAR.

Compressed GNU TAR archive (tar.gz)	5.1.40	21.1M	Download Pick a mirror MD5: 32e7373c16271806007374396e6742ad Signature
Generic Source RPM	5.1.40-0	20.6M	Download Pick a mirror MD5: 9e93e44b30f273c69987353f8b8e6b4a
Red Hat Enterprise Linux 4	5.1.40-0	20.6M	Download Pick a mirror MD5: 2eee37c6cce9550caac75ea29100a437
Red Hat Enterprise Linux 5	5.1.40-0	20.6M	Download Pick a mirror MD5: 1237fa37e57b311ffc75df2f4c297624
Red Hat Enterprise Linux 3	5.1.40-0	20.6M	Download Pick a mirror MD5: 3d2f244622341bb6266cf03d2e1e82b4
SuSE Linux Enterprise Server 9	5.1.40-0	20.6M	Download Pick a mirror MD5: 0cf01f1693b151ea8c1a76a94125f76a
SuSE Linux Enterprise Server 10	5.1.40-0	20.6M	Download Pick a mirror MD5: 77ea0018b18c9739e5d2ca7017acb7b3
Source (zip)	5.1.40	25.3M	Download Pick a mirror MD5: 2f997c4aa3d6c04698873d4f7cf43064 Signature

3.5 上机实践

1. 在Linux操作系统下用RPM包来安装MySQL数据库
2. 在Linux操作系统下用二进制包来安装MySQL数据库
3. 在Linux操作系统下用源码包来安装MySQL数据库

3.6 常见问题及解答

- 1 . 如何选择Linux操作系统下的MySQL数据库？
- 2 . Linux下如何卸载MySQL数据库？

3.7 小结

本章主要介绍Linux操作系统下安装和配置MySQL数据库的方法。通过本章的学习，读者需要掌握在Linux操作系统下安装MySQL的RPM软件包、二进制软件包、源码包的方法。并且，应该掌握手动配置MySQL的方法。如果读者的Linux发行版能够使用RPM软件包，推荐首选RPM软件包来安装MySQL数据库。源码包安装的难度比较大，需要有熟练使用Linux操作系统的能力。因此，读者可以选择性的学习。读者在学习本章时一定要结合实践，只有在安装与配置的过程中才会真正掌握本章的内容。下一章将介绍MySQL的数据类型。

3.8 本章习题

- 1.练习使用RPM软件包安装MySQL数据库。
- 2.练习使用二进制软件包安装MySQL数据库。
- 3.练习使用源码包安装MySQL数据库。
- 4.练习手动配置MySQL的配置文件。

第4章 MySQL数据类型

数据类型是数据的一种属性，其可以决定数据的存储格式、有效范围和相应的限制。MySQL的数据类型包括整数类型、浮点数类型、定点数类型、日期和时间类型、字符串类型和二进制数据类型。在这一章中将讲解的内容包括。

- **整数类型、浮点数类型和定点数类型**
- **日期与时间类型**
- **字符串类型**
- **二进制类型**
- **如何选择数据类型**

4.1 MySQL数据类型介绍

MySQL数据库提供了多种数据类型。其中包括整数类型、浮点数类型、定点数类型、日期和时间类型、字符串类型和二进制数据类型。不同的数据类型有各自的特点，使用范围不相同。而且，存储方式也不一样。本节将详细讲解各种数据类型。

4.1.1 整数类型

整数类型是数据库中最基本的数据类型。标准SQL中支持INTEGER和SMALLINT这两类整数类型。MySQL数据库除了支持这两种类型以外，还扩展支持了TINYINT、MEDIUMINT和BIGINT。本小节将讲解各种整数类型的取值范围、存储的字节数、特点等内容。

整数类型	字节数	无符合数的取值范围	有符合数的取值范围
TINYINT	1	0~255	-128~127
SMALLINT	2	0~65535	-32768~32767
MEDIUMINT	3	0~16777215	-8388608~8388607
INT	4	0~4294967295	-2147483648~2147483647
INTEGER	4	0~4294967295	-2147483648~2147483647
BIGINT	8	0~ 18446744073709551615	-9223372036854775808~ 9223372036854775807

4.1.2 浮点数类型和定点数类型

MySQL中使用浮点数类型和定点数类型来表示小数。浮点数类型包括单精度浮点数（FLOAT型）和双精度浮点数（DOUBLE型）。定点数类型就是DECIMAL型。本小节中将讲解FLOAT型、DOUBLE型和DECIMAL型的取值范围、存储的字节数、特点等内容。

整数类型	字节数	负数的取值范围	非负数的取值范围
FLOAT	4	-3.402823466E+38～ -1.175494351E-38	0和1.175494351E-38～ 3.402823466E+38
DOUBLE	8	- 1.7976931348623157E+308 ～ -2.2250738585072014E- 308	0和2.2250738585072014E-308～ 1.7976931348623157E+308
DECIMAL(M,D) 或者DEC(M,D)	M+2	同DOUBLE型	同DOUBLE型

4.1.3 日期与时间类型

日期与时间类型是为了方便在数据库中存储日期和时间而设计的。MySQL中有多种表示日期和时间的数据类型。其中，YEAR类型表示年份；DATE类型表示日期；TIME类型表示时间；DATETIME和TIMESTAMP表示日期和时间。本小节将介绍上述类型的存储的字节数、取值范围和特点。

- 1 . YEAR类型**
- 2 . TIME类型**
- 3 . DATE类型**
- 4 . DATETIME类型**
- 5 . TIMESTAMP类型**

4.1.4 字符串类型

字符串类型是在数据库中存储字符串的数据类型。字符串类型包括CHAR、VARCHAR、BLOB、TEXT、ENUM和SET。本小节将讲解各种字符串类型的特点和差异。

- 1 . CHAR类型和VARCHAR类型**
- 2 . TEXT类型**
- 3 . ENUM类型**
- 4 . SET类型**

4.1.5 二进制类型

二进制类型是在数据库中存储二进制数据的数据类型。二进制类型包括BINARY、VARBINARY、BIT、TINYBLOB、BLOB、MEDIUMBLOB和LONGBLOB。本小节将讲解各种二进制类型的特点和差异。

- 1 . BINARY和VARBINARY类型**
- 2 . BIT类型**
- 3 . BLOB类型**

4.2 如何选择数据类型

在MySQL中创建表时，需要考虑为字段选择哪种数据类型是最合适的。选择了合适的数据类型，会提高数据库的效率。本小节将讲解选择数据类型的原则。

- 1. 整数类型和浮点数类型**
- 2. 浮点数类型和定点数类型**
- 3. CHAR类型和VARCHAR类型**
- 4. 时间和日期类型**
- 5. ENUM类型和SET类型**
- 6. TEXT类型和BLOB类型**

4.3 常见问题及解答

- 1 . MySQL中什么数据类型能够储存路径？
- 2 . MySQL中如何使用布尔类型？
- 3 . MySQL中如何存储JPG图片和MP3音乐？

4.4 小结

本章介绍了**MySQL**数据库常见的数据类型。整数类型、浮点数类型、日期和时间类型和字符串类型是数据库中使用最频繁的数据类型。定点数类型、二进制数据类型使用相对比较少。因此，读者应该重点掌握前面那几种数据类型。选择数据库类型是本章的难点。读者应该考虑各种数据类型的特点，根据不同的需要选择相应的数据类型。下一章将介绍创建和删除数据库的基本方法，同时还介绍各种**MySQL**存储引擎的特点。

4.5 本章习题

1.浮点数类型和定点数类型的区别是什么？

2.DATETIME类型和TIMESTAMP类型的相同点和不同点是什么？

3.如果一篇新闻中包含文字和图片，应该选择哪种数据类型进行存储？

4.举例说明哪种情况下用ENUM类型，哪种情况下用SET类型。

第5章 操作数据库

数据库是指长期存储在计算机内、有组织的、可共享的数据集合。简而言之，数据库就是一个存储数据的地方。只是，其存储方式有特定的规律。这样可以方便处理数据。数据库的操作包括创建数据库和删除数据库。这些操作都是数据库管理的基础。在这一章中将讲解的内容包括。

- **创建数据库**
- **删除数据库**
- **数据库的存储引擎**
- **如何选择存储引擎**

5.1 创建数据库

创建数据库是指在数据库系统中划分一块空间，用来存储相应的数据。这是进行表操作的基础，也是进行数据库管理的基础。本节主要讲解如何创建数据库。MySQL中，创建数据库是通过SQL语句CREATE DATABASE实现的。其语法形式如下：

```
CREATE DATABASE 数据库名 ；
```

5.2 删除数据库

删除数据库是指在数据库系统中删除已经存在的数据库。删除数据库之后，原来分配的空间将被收回。值得注意的是，删除数据库会删除该数据库中所有的表和所有数据。因此，应该特别小心。本节主要讲解如何删除数据库。

MySQL中，删除数据库是通过SQL语句DROP DATABASE实现的。其语法形式如下：

DROP DATABASE 数据库名；

5.3 数据库存储引擎

MySQL中提到了存储引擎的概念。简而言之，存储引擎就是指表的类型。数据库的存储引擎决定了表在计算机中的存储方式。本小节将讲解存储引擎的内容和分类，以及如何选择合适的存储引擎。

5.3.1 MySQL存储引擎简介

存储引擎的概念是MySQL的特点，而且是一种插入式的存储引擎概念。这决定了MySQL数据库中的表可以用不同的方式存储。用户可以根据自己的不同要求，选择不同的存储方式、是否进行事务处理等。

【示例5-3】下面使用SHOW ENGINES语句可以查看MySQL数据库支持的存储引擎类型，查询方法如下：

```
SHOW ENGINES ;
```

5.3.2 InnoDB存储引擎

InnoDB 是MySQL数据库的一中存储引擎。InnoDB给MySQL的表提供了事务、回滚、崩溃修复能力、多版本并发控制的事务安全。在MySQL从3.23.34a开始包含InnoDB存储引擎。InnoDB是MySQL上第一个提供外键约束的表引擎。而且InnoDB对事务处理的能力，也是MySQL其他存储引擎所无法与之比拟的。笔者安装的MySQL的默认存储引擎就是InnoDB。下文中将讲解InnoDB存储引擎的特点及其优缺点。

InnoDB存储引擎中支持自动增长列AUTO_INCREMENT。自动增长列的值不能为空，且值必须惟一。MySQL中规定自增列必须为主键。在插入值时，如果自动增长列不输入值，则插入的值为自动增长后的值；如果输入的值0或者空（NULL），则插入的值也为自动增长后的值；如果插入某个确定的值，且该值在前面没有出现过，则可以直接插入。

5.3.3 MyISAM存储引擎

MyISAM存储引擎是MySQL中常见的存储引擎，曾是MySQL的默认存储引擎。MyISAM存储引擎是基于ISAM存储引擎发展起来的。MyISAM增加了很多有用的扩展。本小节将讲解MyISAM存储引擎的文件类型、存储格式和优缺点。

MyISAM存储引擎的表存储成三个文件。文件的名字与表名相同。扩展名包括frm、MYD和MYI。其中，frm为扩展名的文件存储表的结构；MYD为扩展名的文件存储数据，其是MYData的缩写；MYI为扩展名的文件存储索引，其是MYIndex的缩写。

5.3.4 MEMORY存储引擎

MEMORY存储引擎是MySQL中的一类特殊的存储引擎。其使用存储在内存中的内容来创建表，而且所有数据也放在内存中。这些特性都与InnoDB存储引擎、MyISAM存储引擎不同。下面将讲解MEMORY存储引擎的文件存储形式、索引类型、存储周期和优缺点。

每个基于MEMORY存储引擎的表实际对应一个磁盘文件。该文件的文件名与表名相同，类型为frm类型。该文件中只存储表的结构。而其数据文件，都是存储在内存中。这样有利于对数据的快速的处理，提高整个表的处理效率。值得注意的是，服务器需要有足够的内存来维持MEMORY存储引擎的表的使用。如果不需要使用了，可以释放这些内存，甚至可以删除不需要的表。

5.3.5 存储引擎的选择

在实际工作中，选择一个合适的存储引擎是一个很复杂的问题。每种存储引擎都有各自的优势，不能笼统的说谁比谁更好。本小节将对各个存储引擎的特点进行对比，给出不同情况下选择存储引擎的建议。

5.4 本章实例

学校需要建立信息化的管理，必须要建立一个信息完备的数据库系统。这个数据库系统中存储着学校的教师、学生、课程安排、考试成绩等各种信息。本章实例将和读者一起建立一个名为school的数据库。

- 1. 登录数据库系统**
- 2. 查看已存在的数据库**
- 3. 查看默认存储引擎**
- 4. 创建数据库**
- 5. 删除数据库**

5.5 上机实践

题目要求：登录数据库系统以后，创建student数据库和teacher数据库。都创建成功后，删除teacher数据库。然后查看数据库系统中还存在哪些数据库。

主要实现过程如下所示：

- (1) 登录数据库
- (2) 查看数据库系统中已存在的数据库。
- (3) 查看该数据库系统支持的存储引擎的类型。
- (4) 创建student数据库和teacher数据库。
- (5) 再次查看数据库系统中已经存在的数据库，确保student和teacher数据库已经存在。
- (6) 删除teacher数据库。
- (7) 再次查看数据库系统中已经存在的数据库，确保teacher数据库已经删除。

5.6 常见问题及解答

1. 如何修改默认存储引擎？
2. 如何选择存储引擎？

5.7 小结

本章主要介绍了创建数据库、删除数据库、MySQL存储引擎的知识。创建和删除数据库是本章的重点。读者应该在计算机上练习创建和删除数据库的方法，这样可以更加透彻的理解这部分的内容。存储引擎的知识比较难，读者只要了解相应的知识即可。读者应该特别注意，安装MySQL数据库的方式不同造成默认存储引擎不同。因此，读者一定要了解自己的MySQL数据库默认使用哪一个存储引擎。下一章将介绍创建表、修改表和删除表的基本方法。

5.8 本章习题

- 1.练习在MySQL数据库系统中创建一个名为worker的数据库。创建成功后，删除该数据库。
- 2.练习用三种不同的方法找出所使用的MySQL数据库的默认存储引擎。
- 3.存储引擎InnoDB、MyISAM和MEMORY各有什么优缺点？

第6章 创建、修改和删除表

表是数据库存储数据的基本单位。一个表包含若干个字段或记录。表的操作包括创建新表、修改表和删除表。这些操作都是数据库管理中最基本，也是最重要的操作。在这一章中将讲解如何在数据库中操作表，内容包括。

- **创建表的方法**
- **表的完整性约束条件**
- **查看表结构的方法**
- **修改表的方法**
- **删除表的方法**

6.1 创建表

创建表是指在已存在的数据库中建立新表。这是建立数据库最重要的一步，是进行其他表操作的基础。本节主要讲解如何创建表。

6.1.1 创建表的语法形式

MySQL中，创建表是通过SQL语句CREATE TABLE实现的。其语法形式如下：

```
CREATE TABLE 表名 ( 属性名 数据类型 [完整性约束  
条件],  
属性名 数据类型 [完整性约束条件],  
.....  
属性名 数据类型  
);
```

6.1.2 设置表的主键

主键是表的一个特殊字段。该字段能唯一地标识该表中的每条信息。主键和记录的关系，如同身份证和人的关系。主键用来标识每个记录，每个记录的主键值都不同。身份证是用来标明人的身份，每个人都具有唯一的身份证号。设置表的主键指在创建表时设置表的某个字段为该表的主键。

主键的主要目的是帮助MySQL以最快的速度查找到表中的某一条信息。主键必须满足的条件：主键必须是唯一的，表中任意两条记录的主键字段的值不能相同；主键的值是非空值。主键可以是单一的字段，也可以是多个字段的组合。

。

- 1. 单字段主键**
- 2. 多字段主键**

6.1.3 设置表的外键

外键是表的一个特殊字段。如果字段sno是一个表A的属性，且依赖于表B的主键。那么，称表B为父表，表A为子表，sno为表A的外键。通过sno字段将父表B和子表A建立关联关系。设置表的外键指在创建表设置某个字段为外键。本小节主要讲解外键设置的原则、外键的作用、设置外键的方法。

设置外键的原则：必须依赖于数据库中已存在的父表的主键；外键可以为空值。

外键的作用是建立该表与其父表的关联关系。父表中删除某条信息时，子表中与之对应的信息也必须有相应的改变。例如，stu_id是student表的主键，stu_id是grade表的外键。当stu_id为‘123’同学退学了，需要从student表中删除该学生的信息。那么，grade表中stu_id为‘123’的所有信息也应该同时删除。

6.1.4 设置表的非空约束

非空性是指字段的值不能为空值（NULL）。非空约束将保证所有记录中该字段都有值。如果用户新插入的记录中，该字段为空值，则数据库系统会报错。例如，在id字段加上非空约束，id字段的值就不能为空值。如果插入记录的id字段的值为空，该记录将不能插入。设置表的非空约束是指在创建表时为表的某些特殊字段加上NOT NULL约束条件。设置非空约束的基本语法规则如下：

属性名 数据类型 NOT NULL

6.1.5 设置表的惟一性约束

惟一性是指所有记录中该字段的值不能重复出现。设置表的惟一性约束是指在创建表时为表的某些特殊字段加上UNIQUE约束条件。惟一性约束将保证所有记录中该字段的值不能重复出现。例如，在id字段加上惟一性约束，所以记录中id字段上不能出现相同的值。例如，在表的id字段加上惟一性约束，那么每条记录的id值都是惟一的，不能出现重复的情况。如果一条的记录为 '0001'，那么该表中就不能出现另一条记录的id为 '0001'。设置惟一性约束的基本语法规则如下：

属性名 数据类型 UNIQUE

6.1.6 设置表的属性值自动增加

AUTO_INCREMENT是MySQL数据库中一个特殊的约束条件。其主要用于为表中插入的新记录自动生成惟一的ID。一个表只能有一个字段使用AUTO_INCREMENT约束，且该字段必须为主键的一部分。AUTO_INCREMENT约束的字段可以是任何整数类型（TINYINT、SMALLINT、INT、BIGINT等）。默认的情况下，该字段的值是从1开始自增。

设置属性值字段增加的基本语法规则如下：

属性名 数据类型 AUTO_INCREMENT

6.1.7 设置表的属性的默认值

在创建表时可以指定表中字段的默认值。如果插入一条新的记录时没有为这个字段赋值，那么数据库系统会自动为这个字段插入默认值。默认值是通过DEFAULT关键字来设置的。设置默认值的基本语法规则如下：

属性名 数据类型 DEFAULT 默认值

6.2 查看表结构

查看表结构是指查看数据库中已存在的表的定义。查看表结构的语句包括DESCRIBE语句和SHOW CREATE TABLE语句。通过这两个语句，可以查看表的字段名、字段的数据类型、完整性约束条件等。本节将详细讲解查看表结构的方法。

6.2.1 查看表基本结构语句DESCRIBE

MySQL中，DESCRIBE语句可以查看表的基本定义。其中包括，字段名、字段数据类型、是否为主键和默认值等。DESCRIBE语句的语法形式如下：

```
DESCRIBE 表名;
```

6.2.2 查看表详细结构语句SHOW CREATE TABLE

MySQL中，SHOW CREATE TABLE语句可以查看表的详细定义。该语句可以查看表的字段名、字段的数据类型、完整性约束条件等信息。除此之外，还可以查看表默认的存储引擎和字符编码。SHOW CREATE TABLE语句的语法形式如下：

```
SHOW CREATE TABLE 表名;
```

6.3 修改表

修改表是指修改数据库中已存在的表的定义。修改表比重新定义表简单，不需要重新加载数据，也不会影响正在进行的服务。MySQL中通过ALTER TABLE语句来修改表。修改表包括修改表名、修改字段数据类型、修改字段名、增加字段、删除字段、修改字段的排列位置、更改默认存储引擎和删除表的外键约束等。本节将详细讲解上述几种修改表的方式。

6.3.1 修改表名

表名可以在一个数据库中唯一的确定一张表。数据库系统通过表名来区分不同的表。例如，数据库school中有student表。那么，student表就是唯一的。在数据库school中不可能存在另一个名为“student”的表。MySQL中，修改表名是通过SQL语句ALTER TABLE实现的。其语法形式如下：

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

6.3.2 修改字段的数据类型

字段的数据类型包括整数型、浮点型、字符串型、二进制类型、日期和时间类型等。数据类型决定了数据的存储格式、约束条件和有效范围。表中的每个字段都有数据类型。有关数据类型的详细内容见第4章。MySQL中，ALTER TABLE语句也可以修改字段的数据类型。其基本语法如下：

ALTER TABLE 表名 MODIFY 属性名 数据类型；

6.3.3 修改字段名

字段名可以在一张表中唯一的确定一个字段。数据库系统通过字段名来区分表中的不同字段。例如，student表中包含id字段。那么，id字段在student表中是唯一的。student表中不可能存在另一个名为“id”的字段。MySQL中，ALTER TABLE语句也可以修改表的字段名。其基本语法如下：

**ALTER TABLE 表名 CHANGE 旧属性名 新属性名
新数据类型；**

其中，“旧属性名”参数指修改前的字段名；“新属性名”参数指修改后的字段名；“新数据类型”参数修改后的数据类型，如不需要修改，则将新数据类型设置成与原来一样。

1. 只修改字段名
2. 修改字段名和字段数据类型

6.3.4 增加字段

在创建表时，表中的字段就已经定义完成。如果要增加新的字段，可以通过ALTER TABLE语句进行增加。MySQL中，ALTER TABLE语句增加字段的基本语法如下：

ALTER TABLE 表名 ADD 属性名1 数据类型 [完整性约束条件] [FIRST | AFTER 属性名2]；

- 1．增加无完整性约束条件的字段**
- 2．增加有完整性约束条件的字段**
- 3．表的第一个位置增加字段**
- 4．表的指定位置之后增加字段**

6.3.5 删除字段

删除字段是指删除已经定义好的表中的某个字段。在表创建好之后，如果发现某个字段需要删除。可以采用将整个表都删除，然后重新创建一张表的做法。这样做是可以达到目的，但必然会影响到表中的数据。而且，操作比较麻烦。MySQL中，ALTER TABLE语句也可以删除表中的字段。其基本语法如下：

ALTER TABLE 表名 DROP 属性名；

6.3.6 修改字段的排列位置

创建表的时候，字段在表中的排列位置就已经确定了。如果要改变字段在表中的排列位置，则需要ALTER TABLE语句来处理。MySQL中，修改字段排列位置的ALTER TABLE语句的基本语法如下：

```
ALTER TABLE 表名 MODIFY 属性名1 数据类型  
FIRST | AFTER 属性名2;
```

其中，“属性名1”参数指需要修改位置的字段的名称；“数据类型”参数指“属性名1”的数据类型；“FIRST”参数指定位置为表的第一个位置；“AFTER 属性名2”参数指定“属性名1”插入在“属性名2”之后。

1. 字段修改到第一个位置
2. 字段修改到指定位置

6.3.7 更改表的存储引擎

MySQL存储引擎是指MySQL数据库中表的存储类型。MySQL存储引擎包括InnoDB、MyISAM、MEMORY等。不同的表类型有着不同的优缺点，在第5章有详细的介绍。在创建表的时候，存储引擎就已经设定好了。如果要改变，可以通过重新创建一张表来实现。这样做是可以达到目的，但必然会影响到表中的数据。而且，操作比较麻烦。MySQL中，ALTER TABLE语句也可以更改表的存储引擎的类型。其基本语法如下：

ALTER TABLE 表名 ENGINE=存储引擎名；

6.3.8 删除表的外键约束

外键是一个特殊字段，其将某一表与其父表建立关联关系。在创建表的时候，外键约束就已经设定好了。由于特殊需要，与父表之间的关联关系需要去除，要求删除外键约束。MySQL中，ALTER TABLE语句也可以删除表的外键约束。其基本语法如下：

ALTER TABLE 表名 DROP FOREIGN KEY 外键别名；

6.4 删除表

删除表是指删除数据库中已存在的表。删除表时，会删除表中的所有数据。因此，在删除表时要特别注意。MySQL中通过DROP TABLE语句来删除表。由于创建表时可能存在外键约束，一些表成为了与之关联的表的父表。要删除这些父表，情况比较复杂。本节将详细讲解删除没有被关联的普通表和被其他表关联的父表的方法。

6.4.1 删除没有被关联的普通表

MySQL中，直接使用DROP TABLE语句可以删除没有被其他关联的普通表。其基本语法如下：

DROP TABLE 表名；

其中，“表名”参数为要删除的表的名称。【示例6-25】下面将删除example5表，SQL代码如下：

DROP TABLE example5；

如果代码运行成功，将从数据库中删除example5表。在执行代码之前，先用DESC语句查看是否存在example5表，以便与删除后进行对比。

6.4.2 删除被其他表关联的父表

在6.1.3节中讲解了创建表时设置表的外键。这样就使数据库中的某些表之间建立了关联关系。一些表成为了父表，这些表被其子表关联着。要删除这些父表，情况不像上一节那么简单。

【示例6-26】下面将要删除6.1节中创建的example1表。SQL代码如下：

```
DROP TABLE example1 ;
```

代码执行后，结果显示为：

```
mysql> DROP TABLE example1;
```

```
ERROR 1217 (23000): Cannot delete or update a parent  
row: a foreign key constraint fails
```

6.5 本章实例

在本小节中将在example数据库创建一个student表和一个grade表。

- 1. 登录数据库系统**
- 2. 选择example数据库**
- 3. 创建student表和grade表**
- 4. 将grade表的course字段的数据类型改为VARCHAR(20)**
- 5. 将s_num字段的位置改到course字段的前面**
- 6. 将grade字段改名为score**
- 7. 删除grade表的外键约束**
- 8. 将grade表的存储引擎更改为MyISAM类型**
- 9. 将student表的address字段删除**
- 10. 在student表中增加名为phone的字段**
- 11. 将grade表改名为gradeInfo**
- 12. 删除student表**

6.6 上机实践

1 . 操作teacher表

题目要求：本题将在school数据库创建一个teacher表。

2 . 操作department表和worker表

题目要求：本题将在example数据库创建一个department表和一个work表。

6.7 常见问题及解答

1. 字段改名后，为什么会有部分约束条件丢失？
2. 如何设置外键？
3. 为什么自增字段不能设置默认值？
4. 如何删除父表？

6.8 小结

本章介绍了创建表、查看表结构、修改表和删除表的方法。创建表、修改表是本章最重要的内容。创建表和修改表的内容比较多，难度也非常大。这两个部分需要不断的练习。只有通过实践练习，才会对这两部分了解得更加透彻。而且，这两部分很容易出现语法错误，必须在练习中掌握正确的语法规则。创建表和修改表后一定要查看表的结构，这样可以确认操作是否正确。本章中的完整性约束条件是难点，希望读者在以后的学习和实践中多思考，以便对完整性约束条件了解的更加透彻。删除表的时候一定要特别小心，因为删除表的同时会删除表中的所有记录。下一章将介绍索引的相关知识。

第7章 索引

索引是一种特殊的数据库结构，可以用来快速查询数据库表中的特定记录。索引是提高数据库性能的重要方式。MySQL中，所有的数据类型都可以被索引。MySQL的索引包括普通索引、惟一性索引、全文索引、单列索引、多列索引和空间索引等。在这一章中将讲解的内容包括：

- **索引的含义和特点**
- **索引的分类**
- **如何设计索引**
- **如何创建索引**
- **如何删除索引**

7.1 索引简介

索引由数据库表中一列或多列组合而成，其作用是提高对表中数据的查询速度。本节将详细讲解索引的含义、作用、分类和设计索引的原则。

7.1.1 索引的含义和特点

索引是创建在表上的，是对数据库表中一列或多列的值进行排序的一种结构。索引可以提高查询的速度。本小节将详细讲解索引的含义、作用、优缺点。索引有起明显的优势，也有其不可避免的缺点。

索引的优点

索引的缺点

7.1.2 索引的分类

MySQL的索引包括普通索引、唯一性索引、全文索引、单列索引、多列索引和空间索引等。本小节将详细讲解这几种索引的含义和特点。

- 1 . 普通索引**
- 2 . 唯一性索引**
- 3 . 全文索引**
- 4 . 单列索引**
- 5 . 多列索引**
- 6 . 空间索引**

7.1.3 索引的设计原则

为了使索引的使用效率更高，在创建索引的时候必须考虑在哪些字段上创建索引和创建什么类型的索引。本小节将向读者介绍一些索引的设计原则。

- 1. 选择惟一性索引**
- 2. 为经常需要排序、分组和联合操作的字段建立索引**
- 3. 为常作为查询条件的字段建立索引**
- 4. 限制索引的数目**
- 5. 尽量使用数据量少的索引**
- 6. 尽量使用前缀来索引**
- 7. 删除不再使用或者很少使用的索引**

7.2 创建索引

创建索引是指在某个表的一列或多列上建立一个索引，以便提高对表的访问速度。创建索引有三种方式，这三种方式分别是创建表的时候创建索引、在已经存在的表上创建索引和使用ALTER TABLE语句来创建索引。本节将详细讲解这三种创建索引的方法。

7.2.1 创建表的时候创建索引

创建表的时候可以直接创建索引，这种方式最简单、方便。其基本形式如下：

CREATE TABLE 表名 (属性名 数据类型 [完整性约束条件],

属性名 数据类型 [完整性约束条件],

- 1 . 创建普通索引**
- 2 . 创建唯一性索引**
- 3 . 创建全文索引**
- 4 . 创建单列索引**
- 5 . 创建多列索引**
- 6 . 创建空间索引**

7.2.2 在已经存在的表上创建索引

在已经存在的表上，可以直接为表上的一个或几个字段创建索引。基本形式如下：

```
CREATE [ UNIQUE | FULLTEXT | SPATIAL ]  
INDEX 索引名  
ON 表名 (属性名 [(长度)] [ ASC | DESC]);
```

1. 创建普通索引
2. 创建唯一性索引
3. 创建全文索引
4. 创建单列索引
5. 创建多列索引
6. 创建空间索引

7.2.3 用ALTER TABLE语句来创建索引

在已经存在的表上，可以通过ALTER TABLE语句直接为表上的一个或几个字段创建索引。基本形式如下：

```
ALTER TABLE 表名 ADD [ UNIQUE | FULLTEXT  
| SPATIAL ] INDEX
```

索引名 (属性名 [(长度)] [ASC | DESC]) ;

其中的参数与上面的两种方式的参数是一样的。

1. 创建普通索引
2. 创建唯一性索引
3. 创建全文索引
4. 创建单列索引
5. 创建多列索引
6. 创建空间索引

7.3 删除索引

删除索引是指将表中已经存在的索引删除掉。一些不再使用的索引会降低表的更新速度，影响数据库的性能。对于这样的索引，应该将其删除。本节将详细讲解删除索引的方法。

对应已经存在的索引，可以通过DROP语句来删除索引。基本形式如下：

DROP INDEX 索引名 ON 表名；

7.4 本章实例

在本小节中将在job数据库创建一个user表和一个information表。

1. 登录数据库系统并创建job数据库
2. 创建user表
3. 创建information表
4. 在name字段创建名为index_name的索引
5. 创建名为index_bir的多列索引
6. 用ALTER TABLE语句创建名为index_id的惟一性索引
7. 删除user表上的index_user索引
8. 删除information表上的index_name索引

7.5 上机实践

题目要求：

(1) 在数据库job下创建workInfo表。创建表的同时在id字段上创建名为index_id的惟一性索引，而且也降序的格式排列。

字段名	字段描述	数据类型	主键	外键	非空	唯一	自增
id	编号	INT(10)	是	否	是	是	是
name	职位名称	VARCHAR(20)	否	否	是	否	否
type	职位类别	VARCHAR(10)	否	否	否	否	否
address	工作地址	VARCHAR(50)	否	否	否	否	否
wages	工资	INT	否	否	否	否	否
contents	工作内容	TINYTEXT	否	否	否	否	否
extra	附加信息	TEXT	否	否	否	否	否

7.6 常见问题及解答

1. MySQL中索引、主键和惟一性的区别是什么？
2. 表中建立了索引以后，导入大量数据为什么会很慢

?

7.7 小结

本章介绍了MySQL数据库的索引的基础知识、创建索引的方法、删除索引的方法。创建索引的内容是本章的重点。读者应该重点掌握创建索引的三种方法。这三种方法分别是创建表的时候创建索引、使用CREATE INDEX语句来创建索引和使用ALTER TABLE语句来创建索引。设计索引的基本原则是本章的难点。读者应该根据本章介绍的基本原则，结合表的实际情况进行设计。下一章将介绍视图的定义、视图的作用、创建视图、删除视图、查询视图和更新视图等内容。

第8章 视图

视图是从一个或多个表中导出来的表，是一种虚拟存在的表。视图就像一个窗口，通过这个窗口可以看到系统专门提供的数据。这样，用户可以不用看到整个数据库表中的数据，而只关心对自己有用的数据。视图可以使用户的操作更方便，而且可以保障数据库系统的安全性。在这一章中将讲解的内容包括：

- **视图的含义和作用**
- **如何创建视图**
- **如何修改视图**
- **如何查看视图**
- **如何删除视图**

8.1 视图简介

视图由数据库中的一个表或多个表导出的虚拟表。其作用是方便用户对数据的操作。本节将详细讲解视图的含义、作用。

8.1.1 视图的含义

视图是一种虚拟的表。视图从数据库中的一个或多个表导出来的表。视图还可以从已经存在的视图的基础上定义。数据库中只存放了视图的定义，而并没有存放视图中的数据。这些数据存放在原来的表中。使用视图查询数据时，数据库系统会从原来的表中取出对应的数据。因此，视图中的数据是依赖于原来的表中的数据的。一旦表中的数据发生改变，显示在视图中的数据也会发生改变。

8.1.2 视图的作用

视图是在原有的表或者视图的基础上重新定义的虚拟表，这可以从原有的表上选取对用户有用的信息。那些对用户没有用，或者用户没有权限了解的信息，都可以直接屏蔽掉。这样做既使应用简单化，也保证了系统的安全。视图起着类似于筛选的作用。视图的作用归纳为如下几点：

- 1．使操作简单化**
- 2．增加数据的安全性**
- 3．提高表的逻辑独立性**

8.2 创建视图

创建视图是指在已存在的数据库表上建立视图。视图可以建立在一张表上，也可以建立在多张表上。本节主要讲解创建视图的方法。

8.2.1 创建视图的语法形式

MySQL中，创建视图是通过SQL语句CREATE VIEW实现的。其语法形式如下：

```
CREATE [ ALGORITHM = { UNDEFINED |  
MERGE | TEMPTABLE } ]  
VIEW 视图名 [( 属性清单 )]  
AS SELECT语句  
[ WITH [ CASCADED | LOCAL ] CHECK OPTION  
];
```

8.2.2 在单表上创建视图

MySQL中可以在单个表上创建视图。【示例8-2】下面在department表上创建一个简单的视图，视图名称为department_view1。创建视图的代码如下：

```
CREATE VIEW department_view1  
AS SELECT * FROM department;
```

代码执行如下：

```
mysql> CREATE VIEW department_view1  
-> AS SELECT * FROM department;  
Query OK, 0 rows affected (0.03 sec)
```

8.2.3 在多表上创建视图

MySQL中也可以在两个或两个以上的表上创建视图，也是使用CREATE VIEW语句实现的。【示例8-4】下面在department表和worker表上创建一个名为worker_view1的视图。创建视图的代码如下：

```
CREATE ALGORITHM=MERGE VIEW
worker_view1 ( name, department,sex, age,address )
AS SELECT name, department.d_name, sex, 2009-
birthday, address
FROM worker , department WHERE worker.d_id=
department.d_id
WITH LOCAL CHECK OPTION;
```

8.3 查看视图

查看视图是指查看数据库中已存在的视图的定义。查看视图必须要有SHOW VIEW的权限，mysql数据库下的user表中保存着这个信息。查看视图的方法包括DESCRIBE语句、SHOW TABLE STATUS语句、SHOW CREATE VIEW语句和查询information_schema数据库下的views表等。本节将详细讲解查看视图的方法。

8.3.1 DESCRIBE语句查看视图基本信息

在6.2.1小节中已经详细讲解过使用DESCRIBE语句来查看表的基本定义。因为，视图也是一张表。只是这张表比较特殊，是一张虚拟的表。因此，同样可以使用DESCRIBE语句可以用来查看视图的基本定义。DESCRIBE语句查看视图的基本形式与查看表的形式是一样的。基本形式如下：

```
DESCRIBE 视图名；
```

8.3.2 SHOW TABLE STATUS语句查看视图 基本信息

MySQL中，可以使用SHOW TABLE STATUS语句来查看视图的信息。其语法形式如下：

SHOW TABLE STATUS LIKE ‘视图名’ ；

其中，“LIKE”表示后面匹配的是字符串；“视图名”参数指要查看的视图的名程，需要用单引号引起了。

8.3.3 SHOW CREATE VIEW语句查看视图详细信息

MySQL中，SHOW CREATE VIEW语句可以查看视图的详细定义。其语法形式如下：

```
SHOW CREATE VIEW 视图名
```

8.3.4 在views表中查看视图详细信息

MySQL中，所有视图的定义都存在information_schema数据库下的views表中。查询views表，可以查看到数据库中所有视图的详细信息。查询的语句如下：

```
SELECT * FROM information_schema.views ;
```

其中，“*”表示查询所有的列的信息；

“information_schema.views”表示information_schema数据库下面的views表。

8.4 修改视图

修改视图是指修改数据库中已存在的表的定义。当基本表的某些字段发生改变时，可以通过修改视图来保持视图和基本表之间一致。MySQL中通过CREATE OR REPLACE VIEW语句和ALTER语句来修改视图。本节将详细讲解修改视图的方式。

8.4.1 CREATE OR REPLACE VIEW语句修改视图

MySQL中，CREATE OR REPLACE VIEW语句可以用来修改视图。该语句的使用非常灵活。在视图已经存在的情况下，对视图进行修改；视图不存在时，可以创建视图。CREATE OR REPLACE VIEW语句的语法形式如下：

```
CREATE OR REPLACE [ ALGORITHM = {  
UNDEFINED | MERGE | TEMPTABLE } ]  
VIEW 视图名 [ ( 属性清单 ) ]  
AS SELECT语句  
[ WITH [ CASCADED | LOCAL ] CHECK OPTION  
];
```

8.4.2 ALTER语句修改视图

在MySQL中，ALTER语句可以修改表的定义，可以创建索引。不仅如此，ALTER语句还可以用来修改视图。ALTER语句修改视图的语法格式如下：

```
ALTER [ ALGORITHM = { UNDEFINED | MERGE |  
TEMPTABLE } ]  
VIEW 视图名 [ ( 属性清单 ) ]  
AS SELECT语句  
[ WITH [ CASCADED | LOCAL ] CHECK OPTION  
];
```

8.5 更新视图

更新视图是指通过视图来插入（INSERT）、更新（UPDATE）和删除（DELETE）表中的数据。因为是视图是一个虚拟表，其中没有数据。通过视图更新时，都是转换到基本表来更新。更新视图时，只能更新权限范围内的数据。超出了范围，就不能更新。本小节将重点讲解更新视图的方法和更新视图的限制。

8.6 删除视图

删除视图是指删除数据库中已存在的视图。删除视图时，只能删除视图的定义，不会删除数据。MySQL中，使用DROP VIEW语句来删除视图。但是，用户必须拥有DROP权限。本节将详细讲解删除视图的方法。

对需要删除的视图，使用DROP VIEW语句进行删除。基本形式如下：

**DROP VIEW [IF EXISTS] 视图名列表 [RESTRICT
| CASCADE]**

8.7 本章实例

在本小节中将在test数据库中work_info表上进行视图操作。

- 1 . 在test数据库中work_info表
- 2 . 插入记录
- 3 . 创建视图info_view
- 4 . 查看视图info_view的基本结构和详细结构
- 5 . 查看视图info_view的所有记录
- 6 . 修改视图info_view
- 7 . 更新视图
- 8 . 删除视图

8.8 上机实践

题目要求：

(1) 在数据库example下创建college表。

(2) 在student表上创建视图college_view。视图的字段包括student_num、student_name、student_age和department。ALGORITHM设置为UNDEFINED类型。加上WITH LOCAL CHECK OPTION条件。

(3) 查看视图college_view的详细结构。

(4) 更新视图。向视图中插入三条记录。

(5) 修改视图，使其显示专业为“计算机”的信息，其他条件不变。

(6) 删除视图college_view。

8.9 常见问题及解答

- 1 . MySQL中视图和表的区别和联系是什么？
- 2 . 为什么视图更新不了？

8.10 小结

本章介绍了**MySQL**数据库的视图的含义和作用，并且讲解了创建视图、修改视图和删除视图的方法。创建视图和修改视图是本章的重点内容。这两部分的内容比较多，而且比较复杂。希望读者能够认真学习这两部分的内容，并且需要在计算机上实际操作。读者在创建视图和修改视图后，一定要查看视图的结构，以确保创建和修改的操作是否正确。更新视图是本章的一个难点。因为实际中存在一些造成视图不能更新的因素。本章中介绍了一些造成视图不能更新的因素，希望读者在练习中认真分析。下一章将介绍触发器的基本内容，包括创建触发器、删除触发器和查询触发器。

第9章 触发器

触发器 (TRIGGER) 是由事件来触发某个操作。这些事件包括INSERT语句、UPDATE语句和DELETE语句。当数据库系统执行这些事件时，就会激活触发器执行相应的操作。MySQL从5.0.2版本开始支持触发器。在这一章中将讲解的内容包括：

触发器的含义和作用

如何创建触发器

如何查看触发器

如何删除触发器

9.1 创建触发器

触发器是由**INSERT**、**UPDATE**和**DELETE**等事件来触发某种特定操作。满足触发器的触发条件时，数据库系统就会执行触发器中定义的程序语句。这样做可以保证某些操作之间的一致性。例如，当学生表中增加了一个学生的信息时，学生的总数就必须同时改变。可以在这里创建一个触发器，每次增加一个学生的记录，就执行一次计算学生总数的操作。这样就可以保证每次增加学生的记录后，学生总数是与记录数是一致的。触发器触发的执行语句可能只有一个，也可能有多个。本节将详细讲解创建触发器的方法。

9.1.1 创建只有一个执行语句的触发器

MySQL中，创建只有一个执行语句的触发器的基本形式如下：

```
CREATE TRIGGER 触发器名 BEFORE | AFTER  
触发事件  
ON 表名 FOR EACH ROW 执行语句
```

9.1.2 创建有多个执行语句的触发器

MySQL中，触发器触发的执行语句可能多个。创建有多个执行语句的触发器的基本形式如下：

```
CREATE TRIGGER 触发器名 BEFORE | AFTER  
触发事件  
ON 表名 FOR EACH ROW  
BEGIN  
执行语句列表  
END
```

9.2 查看触发器

查看触发器是指查看数据库中已存在的触发器的定义、状态和语法等信息。查看触发器的方法包括SHOW TRIGGERS语句和查询information_schema数据库下的triggers表等。本节将详细讲解查看触发器的方法。

9.2.1 SHOW TRIGGERS语句查看触发器信息

MySQL中，可以执行SHOW TRIGGERS语句来查看触发器的基本信息。其基本形式如下：

```
SHOW TRIGGERS ;
```

9.2.2 在triggers表中查看触发器信息

MySQL中，所有触发器的定义都存在于information_schema数据库下的triggers表中。查询triggers表，可以查看到数据库中所有触发器的详细信息。查询的语句如下：

```
SELECT * FROM information_schema.triggers ;
```

9.3 触发器的使用

MySQL中，触发器执行的顺序是BEFORE触发器、表操作（INSERT、UPDATE和DELETE）、AFTER触发器。下面通过一个示例演示这三者的执行顺序。

【示例9-6】下面在department表上创建BEFORE INSERT和AFTER INSERT这两个触发器。在向department表中插入数据时，观察这两个触发器的触发顺序。创建触发器的代码如下：

```
CREATE TRIGGER before_insert BEFORE INSERT
ON department FOR EACH ROW
INSERT INTO trigger_test VALUES(null, " before_insert ");
CREATE TRIGGER after_insert AFTER INSERT
ON department FOR EACH ROW
INSERT INTO trigger_test VALUES(null, " after_insert ");
```

9.4 删除触发器

删除触发器指删除数据库中已经存在的触发器。MySQL中使用DROP TRIGGER语句来删除触发器。其基本形式如下：

DROP TRIGGER 触发器名；

9.5 本章实例

在本小节中将在product表上创建三个触发器。每次激活触发器后，都会更新operate表。

- 1 . 创建product表和operate表**
- 2 . 创建product_bf_insert触发器**
- 3 . 创建product_af_update触发器**
- 4 . 创建product_af_del触发器**
- 5 . 对product表进行操作**
- 6 . 删除触发器**

9.6 上机实践

题目要求：

(1) 9.5小节中的在product表上分别创建AFTER INSERT、BEFORE UPDATE和BEFORE DELETE三个触发器，触发器的名称分别为product_af_insert、product_bf_update和product_bf_del。执行语句部分都是向operate表插入操作方法和操作时间。

(2) 查看product_bf_del触发器的基本结构。

(3) 分别执行INSERT、UPDATE和DELETE操作来触发这三个触发器。

(4) 删除product_bf_update和product_bf_del这两个触发器。

9.7 常见问题及解答

1. MySQL中创建多条执行语句的触发器总是遇到分号就结束创建，然后报错？

MySQL中，创建多条执行语句的触发器时，需要用到BEGIN...END的形式。每个执行语句都必须是以分号结束。但是，这样就会出问题。因为，系统默认分号是SQL程序结束的标志，遇到分号整个程序就结束了。要解决这个问题，就需要DELIMITER语句来改变程序的结束符号。如“DELIMITER &&”，可以将程序的结束符号变成“&&”。如果要把结束符号变回分号，只要执行“DELIMITER ;”即可。

9.8 小结

本章介绍了MySQL数据库的触发器的定义和作用、创建触发器、查看触发器、使用触发器、删除触发器等内容。创建触发器和使用触发器是本章的重点内容。读者在创建触发器后，一定要查看触发器的结构。使用触发器时，触发器执行的顺序是BEFORE触发器、表操作（INSERT、UPDATE 和DELETE）、AFTER触发器。创建触发器是本章的难点。读者需要将本章的知识结合实际需要来设计触发器。下一章将介绍查询语句的使用。

9.9 本章习题

1. 各种触发器的触发顺序是什么？
2. 触发器执行的语句的限制条件是什么？
3. 在9.5小节上的product表上创建BEFORE INSERT和AFTER INSERT两种触发器。这两个触发器触发后都是更新test_trigger表。

这两个触发器创建成功后，执行如下操作：

(1) 查看两个触发器的基本结构。

(2) 向product表中插入记录，插入语句如下所示：

```
INSERT INTO product VALUES(10, 'ccc', '治疗头痛', '北京ccc制药厂', '北京市昌平区');
```

(3) 查看test_trigger表的记录，分析触发器的触发顺序。

(4) 删除触发器test_bf_insert和test_af_insert。

第10章 查询数据

查询数据指从数据库中获取所需要的数据。查询数据是数据库操作中最常用，也是最重要的操作。用户可以根据自己对数据的需求，使用不同的查询方式。通过不同的查询方式，可以获得不同的数据。MySQL中是使用SELECT语句来查询数据的。在这一章中将讲解的内容包括。

- **查询语句的基本语法**
- **在单表上查询数据**
- **使用聚合函数查询数据**
- **多表上联合查询**
- **子查询**
- **合并查询结果**
- **为表和字段取别名**
- **使用正则表达式查询**

10.1 基本查询语句

查询数据是数据库操作中最常用的操作。通过对数据库的查询，用户可以从数据库中获取需要的数据。数据库中可能包含着无数的表，表中可能包含着无数的记录。因此，要获得所需的数据并非易事。MySQL中可以使用SELECT语句来查询数据。根据查询的条件不同，数据库系统会找到不同的数据。通过SELECT语句可以很方便的获取所需的信息。

MySQL中，SELECT的基本语法形式如下：

SELECT 属性列表

FROM 表名和视图列表

[WHERE 条件表达式1]

[GROUP BY 属性名1 [HAVING 条件表达式2]]

[ORDER BY 属性名2 [ASC | DESC]]

10.2 单表查询

单表查询是指从一张表中查询所需要的数据。查询数据时，可以从一张表中查询数据，也可以从多张表中同时查询数据。两者的查询方式上有一定的区别。因为单表查询只在一张表上进行操作，所以查询比较简单。本小节将讲解在单表上查询所有的字段、查询指定的字段、查询指定的行、多条件查询、查询结果不重复、给查询结果排序、分组查询和用LIMIT限制查询结果的数量等内容。

10.2.1 查询所有字段

查询所有字段是指查询表中所有字段的数据。这种方式可以将表中所有字段的数据都查询出来。MySQL中有两种方式可以查询表中所有的字段。本小节将详细讲解这两种方法。

- 1. 列出表的所有字段**
- 2. 使用 “*” 查询所有字段**

10.2.2 查询指定字段

查询数据时，可以在SELECT语句的“属性列表”中列出所要查询的字段。这种方式可以指定需要查询的字段，而不需要查询出所有的字段。【示例10-5】下面查询employee表中num、name、sex和homeaddr这四个字段的数据。

SELECT语句的代码如下：

```
SELECT num, name, sex,homeaddr FROM employee;
```

10.2.3 查询指定记录

**SELECT语句中可以设置查询条件。用户可以根据自己
的需要来设置查询条件，按条件进行查询。查询的结果必须
满足查询条件。例如，用户需要查找d_id为1001的记录，那
么可以设置“d_id=1001”为查询条件。这样查询结果中的记
录就都会满足“d_id=1001”这个条件。WHERE子句可以用
来指定查询条件。其语法规则如下：**

WHERE 条件表达式

10.2.4 带IN关键字的查询

IN关键字可以判断某个字段的值是否在指定的集合中。如果字段的值在集合中，则满足查询条件，该纪录将被查询出来。如果不在集合中，则不满足查询条件。其语法规则如下：

[NOT] IN (元素1, 元素2, ..., 元素n)

10.2.5 带BETWEEN AND的范围查询

BETWEEN AND关键字可以判读某个字段的值是否在指定的范围内。如果字段的值在指定范围内，则满足查询条件，该纪录将被查询出来。如果不在指定范围内，则不满足查询条件。其语法规则如下：

[NOT] BETWEEN 取值1 AND 取值2

10.2.6 带LIKE的字符匹配查询

LIKE关键字可以匹配字符串是否相等。如果字段的值与指定的字符串相匹配，则满足查询条件，该纪录将被查询出来。如果与指定的字符串不匹配，则不满足查询条件。其语法规则如下：

[NOT] LIKE '字符串'

10.2.7 查询空值

IS NULL关键字可以用来判断字段的值是否为空值（NULL）。如果字段的值是空值，则满足查询条件，该记录将被查询出来。如果字段的值不是空值，则不满足查询条件。其语法规则如下：

IS [NOT] NULL

其中，“NOT”是可选参数，加上NOT表示字段不是空值时满足条件。

10.2.8 带AND的多条件查询

AND关键字可以用来联合多个条件进行查询。使用AND关键字时，只有同时满足所有查询条件的记录会被查询出来。如果不满足这些查询条件的其中一个，这样的记录将被排除掉。AND关键字的语法规则如下：

条件表达式1 AND 条件表达式2 [... AND 条件表达式n]

其中，AND可以连接两个条件表达式。而且，可以同时使用多个AND关键字，这样可以连接更多的条件表达式。

10.2.9 带OR的多条件查询

OR关键字也可以用来联合多个条件进行查询，但是与AND关键字不同。使用OR关键字时，只要满足这几个查询条件的其中一个，这样的记录将会被查询出来。如果不满足这些查询条件中的任何一个，这样的记录将被排除掉。OR关键字的语法规则如下：

条件表达式1 OR 条件表达式2 [...OR 条件表达式n]

其中，OR可以用来连接两个条件表达式。而且，可以同时使用多个OR关键字，这样可以连接更多的条件表达式

。

10.2.10 查询结果不重复

如果表中的某些字段上没有惟一性约束，这些字段可能存在着重复的值。例如，employee表中的d_id字段就存在着重复的情况。

```
+-----+  
| d_id |  
+-----+  
| 1001 |  
| 1001 |  
| 1002 |  
| 1004 |  
+-----+
```

10.2.11 对查询结果排序

从表中查询出来的数据可能是无序的，或者其排列顺序不是用户所期望的顺序。为了使查询结果的顺序满足用户的要求，可以使用ORDER BY关键字对记录进行排序。其语法规则如下：

ORDER BY 属性名 [ASC | DESC]

10.2.12 分组查询

GROUP BY关键字可以将查询结果按某个字段或多个字段进行分组。字段中值相等的为一组。其语法规则如下：

GROUP BY 属性名 [HAVING 条件表达式] [WITH ROLLUP]

- 1 . 单独使用GROUP BY关键字来分组**
- 2 . GROUP BY关键字与GROUP_CONCAT()函数一起使用**
- 3 . GROUP BY关键字与集合函数一起使用**
- 4 . GROUP BY关键与HAVING一起使用**
- 5 . 按多个字段进行分组**
- 6 . GROUP BY关键与WITH ROLLUP一起使用**

10.2.13 用LIMIT限制查询结果的数量

查询数据时，可能会查询出很多的记录。而用户需要的记录可能只是很少的一部。这样就需要来限制查询结果的数量。LIMIT是MySQL中的一个特殊关键字。其可以用来指定查询结果从哪条记录开始显示。还可以指定一共显示多少条记录。LIMIT关键字有两种使用方式。这两种方式分别是不指定初始位置和指定初始位置。

1. 不指定初始位置
2. 指定初始位置

10.3 使用集合函数查询

集合函数包括COUNT()、SUM()、AVG()、MAX()和MIN()。其中，COUNT()用来统计记录的条数；SUM()用来计算字段的值的总和；AVG()用来计算字段的值的平均值；MAX()用来查询字段的最大值；MIN()用来查询字段的最小值。当需要对表中的记录求和、求平均值、查询最大值、查询最小值等操作时，可以使用集合函数。例如，需要计算学生成绩表中的平均成绩，可以使用AVG()函数。GROUP BY关键字通常需要与集合函数一起使用。本节中将详细讲解各种集合函数。

10.3.1 COUNT()函数

COUNT()函数用来统计记录的条数。如果要统计employee表中有多少条记录，可以使用COUNT()函数。如果要统计employee表中不同部门的人数，也可以使用COUNT()函数。【示例10-40】下面使用COUNT()函数统计employee表的记录数。SELECT语句如下：

```
SELECT COUNT(*) FROM employee ;
```

执行结果如下：

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

10.3.2 SUM()函数

SUM()函数是求和函数。使用SUM()函数可以求出表中某个字段取值的总和。例如，可以用SUM()函数来求学生的总成绩。

10.3.3 AVG()函数

AVG()函数是求平均值的函数。使用AVG()函数可以求出表中某个字段取值的平均值。例如，可以用AVG()函数来求平均年龄，也可以使用AVG()函数来求学生的平均成绩。

10.3.4 MAX()函数

MAX()函数是求最大值的函数。使用MAX()函数可以求出表中某个字段取值的最大值。例如，可以用MAX()函数来查询最大年龄，也可以使用MAX()函数来求各科的最高成绩。

10.3.5 MIN()函数

MIN()函数是求最小值的函数。使用MIN()函数可以求出表中某个字段取值的最小值。例如，可以用MIN()函数来查询最小年龄，也可以使用MIN()函数来求各科的最低成绩。

。

10.4 连接查询

连接查询是将两个或两个以上的表按某个条件连接起来，从中选取需要的数据。连接查询是同时查询两个或两个以上的表时使用的。当不同的表中存在表示相同意义的字段时，可以通过该字段来连接这几个表。例如，学生表中有course_id字段来表示所学课程的课程号，课程表中有num字段来表示课程号。那么，可以通过学生表中的course_id字段与课程表中的num字段来进行连接查询。连接查询包括内连接查询和外连接查询。本小节将详细讲解内连接查询和外连接查询。同时，还会讲解多个条件结合在一起进行复合连接查询。

10.4.1 内连接查询

内连接查询是一种最常用的连接查询。内连接查询可以查询两个或两个以上的表。为了读者更好的理解，暂时只讲解两个表的连接查询。当两个表中存在表示相同意义的字段时，可以通过该字段来连接这两个表。当该字段的值相等时，就查询出该记录。

10.4.2 外连接查询

外连接查询可以查询两个或两个以上的表。外连接查询也需要通过指定字段来进行连接。当该字段取值相等时，可以查询出该记录。而且，该字段取值不相等的记录也可以查询出来。外连接查询包括左连接查询和右连接查询。其基本语法如下：

SELECT 属性名列表

FROM 表名1 LEFT | RIGHT JOIN 表名2

ON 表名1.属性名1=表名2.属性名2；

1 . 左连接查询

2 . 右连接查询

10.4.3 复合条件连接查询

在连接查询时，也可以增加其他的限制条件。通过多个条件的复合查询，可以使查询结果更加准确。例如，employee表和department表进行连接查询时，可以限制age字段的取值必须大于24。这样，可以更加准确的查询出年龄大于24岁的员工的信息。【示例10-54】下面使用内连接查询的方式查询employee表和department。并且employee表中的age字段的值必须大于24。右连接的SELECT语句如下：

```
SELECT
num,name,employee.d_id,age,sex,d_name,function
FROM employee, department
WHERE employee.d_id=department.d_id
AND age>24;
```

10.5 子查询

子查询是将一个查询语句嵌套在另一个查询语句中。内层查询语句的查询结果，可以为外层查询语句提供查询条件。因为在特定情况下，一个查询语句的条件需要另一个查询语句来获取。例如，现在需要从学生成绩表中查询计算机系学生的各科成绩。那么，首先就必须知道哪些课程是计算机系学生选修的。因此，必须先查询计算机系学生选修的课程，然后根据这些课程来查询计算机系学生的各科成绩。通过子查询，可以实现多表之间的查询。子查询中可能包括IN、NOT IN、ANY、ALL、EXISTS、NOT EXISTS等关键字。子查询中还可能包含比较运算符，如“=”、“!=”、“>”和“<”等。本小节将详细讲解子查询的知识。

10.5.1 带IN关键字的子查询

一个查询语句的条件可能落在另一个SELECT语句的查询结果中。这可以通过IN关键字来判断。例如，要查询哪些同学选择了计算机系开设的课程。先必须从课程表中查询出计算机系开设了哪些课程。然后再从学生表中进行查询。如果学生选修的课程在前面查询出来的课程中，则查询出该同学的信息。这可以用带IN关键字的子查询来实现。

10.5.2 带比较运算符的子查询

子查询可以使用比较运算符。这些比较运算符包括=、!=、>、>=、<、<=、<>等。其中，<>与!=是等价的。比较运算符在子查询时使用的非常广泛。如查询分数、年龄、价格、收入等。

10.5.3 带EXISTS关键字的子查询

EXISTS关键字表示存在。使用EXISTS关键字时，内层查询语句不返回查询的记录。而是返回一个真假值。如果内层查询语句查询到满足条件的记录，就返回一个真值（True）。否则，将返回一个假值（False）。当返回的值是真值时，外层查询语句将进行查询。当返回值是假值时，外层查询语句不进行查询或者查询不出任何记录。

10.5.4 带ANY关键字的子查询

ANY关键字表示满足其中任一条件。使用ANY关键字时，只要满足内层查询语句返回的结果中的任何一个，就可以通过该条件来执行外层查询语句。例如，需要查询哪些同学能够获得奖学金。那么，首先必须从奖学金表中查询出各种奖学金要求的最低分。只要一个同学的成绩高于不同奖学金最低分的任何一个，这个同学就可以获得奖学金。ANY关键字通常与比较运算符一起使用。

10.5.5 带ALL关键字的子查询

ALL关键字表示满足所有条件。使用ALL关键字时，只有满足内层查询语句返回的所有结果，才可以执行外层查询语句。例如，需要查询哪些同学能够获得一等奖学金。首先必须从奖学金表中查询出各种奖学金要求的最低分。因为一等奖学金要求的分数最高。只有当同学的成绩高于所有奖学金最低分时，这个同学才可能获得一等奖学金。ALL关键字也经常与比较运算符一起使用。

10.6 合并查询结果

合并查询结果是将多个SELECT语句的查询结果合并到一起。因为某种情况下，需要将几个SELECT语句查询出来的结果合并起来显示。例如，现在需要查询公司甲和公司乙这两个公司所有员工的信息。这就需要从公司甲中查询出所有员工的信息，再从公司乙中查询出所有员工的信息。然后将两次的查询结果合并到一起。进行合并操作使用UNION和UNION ALL关键字。本小节将详细讲解合并查询结果的方法。

使用UNION关键字时，数据库系统会将所有的查询结果合并到一起，然后去除掉相同的记录。而UNION ALL关键字则只是简单的合并到一起。其语法规则如下：

```
SELECT语句1
UNION | UNION ALL
SELECT语句2
UNION | UNION ALL ....
SELECT语句n;
```

10.7 为表和字段取别名

在查询时，可以为表和字段取一个别名。这个别名可以代替其指定的表和字段。本小节将详细的讲解怎样为表和字段取别名。

10.7.1 为表取别名

当表的名称特别长时，在查询中直接使用表名很不方便。这时可以为表取一个别名。用这个别名来代替表的名称。例如，电力软件中的变压器表的名称为 `power_system_transform`。如果要使用该表下面的字段 `id`，但同时查询的其他表中也有 `id` 字段。这样就必须指明是哪个表下的 `id` 字段，如 `power_system_transform.id`。因为变压器表的表名太长，使用起来不是很方便。为了解决这个问题，可以将变压器表取一个别名。如将 `power_system_transform` 取个别名为 `t`，那么 `t` 就代表了变压器表。 `t.id` 就与 `power_system_transform.id` 表示的意思相同了。本小节中将讲解怎么样为表取一个别名，以及查询时如何使用别名。

MySQL中为表取别名的基本形式如下：

表名 表的别名

10.7.2 为字段取别名

当查询数据时，MySQL会显示每个输出列的名词。默认的情况下，显示的列名是创建表是定义的列名。例如，department表的列名分别是d_id、d_name、function和address。当查询department表时，就会相应显示这几个列名。有时为了显示结果更加直观，需要一个更加直观的名字来表示这一列。如department_name可以很直接的知道是部门名称。这时就需要将d_name字段取别名为department_name。本小节将详细讲解如何为字段取别名。

MySQL中为字段取别名的基本形式如下：

属性名 [AS] 别名

10.8 使用正则表达式查询

正则表达式是用某种模式去匹配一类字符串的一个方式。例如，使用正则表达式可以查询出包含A、B、C其中任一字母的字符串。正则表达式的查询能力比通配字符的查询能力更强大，而且更加的灵活。正则表达式可以应用于非常复杂查询。本节将详细讲解如何使用正则表达式来查询。

MySQL中，使用REGEXP关键字来匹配查询正则表达式。其基本形式如下：

属性名 REGEXP '匹配方式'

10.8.1 查询以特定字符或字符串开头的记录

使用字符 “^”可以匹配以特定字符或字符串开头的记录。【示例10-72】下面从info表name字段中查询以字母“L”开头的记录。SQL代码如下：

```
SELECT * FROM info WHERE name REGEXP  
'^L';
```

10.8.2 查询以特定字符或字符串结尾的记录

使用字符 “\$”可以匹配以特定字符或字符串结尾的记录。**【示例10-74】**下面从info表name字段中查询以字母 “c”结尾的记录。SQL代码如下：

```
SELECT * FROM info WHERE name REGEXP  
'c$';
```

10.8.3 用符号 “.”来替代字符串中的任意一个字符

用正则表达式来查询时，可以用 “.”来替代字符串中的任意一个字符。【示例10-76】下面从info表name字段中查询以字母 “L”开头，以字母 “y”结尾，中间有两个任意字符的记录。SQL代码如下：

```
SELECT * FROM info WHERE name REGEXP  
'^L..y$';
```

10.8.4 匹配指定字符中的任意一个

使用方括号 ([]) 可以将需要查询字符组成一个字符集。只要记录中包含方括号中的任意字符，该记录将会被查询出来。例如，通过 “[abc]” 可以查询包含 a、b、c 这三个字母中任何一个的记录。

10.8.5 匹配指定字符以外的字符

使用 “[^字符集合]”可以匹配指定字符以外的字符。【**示例10-80**】下面从info表name字段中查询包含a到w字母和数字以外的字符的记录。SQL代码如下：

```
SELECT * FROM info WHERE name REGEXP  
'[^a-w0-9]';
```

10.8.6 匹配指定字符串

正则表达式可以匹配字符串。当表中的记录包含这个字符串时，就可以将该记录查询出来。如果指定多个字符串时，需要用符号“|”隔开。只要匹配这些字符串中的任意一个即可。

10.8.7 使用 “*” 和 “+” 来匹配多个字符

正则表达式中，“*” 和 “+” 都可以匹配多个该符号之前的字符。但是，“+” 至少表示一个字符，而 “*” 可以表示零个字符。

10.8.8 使用{M}或者{M,N}来指定字符串连续出现的次数

正则表达式中，“字符串{M}”表示字符串连续出现M次；“字符串{M,N}”表示字符串连续出现至少M次，最多N次。例如，“ab{2}”表示字符串“ab”连续出现两次。“ab{2,4}”表示字符串“ab”连续出现至少两次，最多四次。

10.9 本章实例

在本小节中将在student表和score表上进行查询。

10.10 上机实践

在**department**表和**employee**表上进行信息查询。

10.11 常见问题及解答

1. MySQL中通配符与正则表达式的区别？
2. 什么情况下使用LIMIT来限制查询结果的数量？
3. 集合函数必须要用GROUP BY关键字吗？
4. 给表和字段取别名有什么用？

10.12 小结

本章介绍了MySQL数据库常见的查询方法。查询指定字段、查询指定记录、使用LIKE关键字和通配符查询、使用AND和OR来实现多条件查询、分组查询、连接查询、子查询、查询结果排序是本章的重点内容。这些查询方式实际中使用最频繁。分组查询经常和集合函数一起使用，而且使用方法非常灵活。使用LIMIT关键字来限制查询结果的条数是MySQL数据库的特色。本章的难点是使用正则表达式来查询。正则表达式的功能很强大，使用起来很灵活。希望读者能够阅读有关正则表达式的相关知识，能够对正则表达式了解得更加透彻。由于字符串函数、日期和时间函数都还没有讲解，那么对字符串、日期和时间的查询将放在第13章详细讲解。下一章将为读者讲解数据的插入、更新和删除。

10.13 本章习题

1. 10.10小节中的department表和employee表中进行如下查询：

(1) 计算employee表所有员工的出生年份，并且显示结果中字段的别名为birth_year。

(2) 用LIMIT关键字来查询工资最低的员工的信息。

(3) 计算男性员工和女性员工的平均工资。

(4) 查询在上海工作的员工的姓名、性别、年龄和部门名称。

(5) 用右连接的方式查询department表和employee表。

(6) 查询名字以字母T开头的员工的姓名、性别、年龄、部门、工作地点。

(7) 查询年龄小于25或者大于30的员工的信息。

(8) 查询家庭住址是永州市的员工的信息

2. 习正则表达式的使用。

第11章 插入、更新与删除数据

数据库通过插入、更新和删除等方式来改变表中的记录。插入数据是向表中插入新的记录，通过INSERT语句来实现。更新数据是改变表中已经存在的数据，使用UPDATE语句来实现。删除数据是删除表中不再使用的数据，通过DELETE语句来实现。在这一章中将讲解的内容包括：

- **插入新纪录**
- **更新数据**
- **删除记录**

11.1 插入数据

插入数据是向表中插入新的记录。通过这种方式可以为表中增加新的数据。MySQL中，通过INSERT语句来插入新的数据。使用INSERT语句可以同时为表的所有字段插入数据，也可以为表的指定字段插入数据。INSERT语句可以同时插入多条记录，还可以将一个表中查询出来的数据插入到另一个表中。本小节将详细讲解这些内容。

11.1.1 为表的所有字段插入数据

通常情况下，插入的新记录要包含表的所有字段。

INSERT语句有两种方式可以同时为表的所有字段插入数据。第一种方式是不指定具体的字段名。第二种方式是列出表的所有字段。下面为读者详细讲解这两种方法。

- 1 . INSERT语句中不指定具体的字段名**
- 2 . INSERT语句中列出所有字段**

11.1.2 为表的指定字段插入数据

如果上一节中讲解的INSERT语句只是指定部分字段，这就可以为表中的部分字段插入数据了。其基本语句形式如下：

```
INSERT INTO 表名(属性1, 属性2, ..., 属性m)  
VALUES(值1,值2, ..., 值m);
```

11.1.3 同时插入多条记录

同时插入多条记录是指一个INSERT语句插入多条记录。当用户需要插入好几条记录，用户可以使用上面两个小节中的方法逐条插入记录。但是，每次都要写一个新的INSERT语句。这样比较麻烦。MySQL中，一个INSERT语句可以同时插入多条记录。其基本语法形式如下：

```
INSERT INTO 表名 [ (属性列表) ]  
VALUES(取值列表1),(取值列表2)  
... ,  
(取值列表n) ;
```

11.1.4 将查询结果插入到表中

INSERT语句可以将一个表中查询出来的数据插入到另一表中。这样，可以方便不同表之间进行数据交换。其基本语法形式如下：

```
INSERT INTO 表名1 (属性列表1)  
SELECT 属性列表2 FROM 表名2 WHERE 条件表达  
式;
```

11.2 更新数据

更新数据是更新表中已经存在的记录。通过这种方式可以改变表中已经存在的数据。例如，学生表中某个学生的家庭住址改变了，这就需要在学生表中修改该同学的家庭地址。MySQL中，通过UPDATE语句来更新数据。本小节将详细讲解这些内容。

MySQL中，UPDATE语句的基本语法形式如下：

UPDATE 表名

SET 属性名1=取值1, 属性名2=取值2,

…,

属性名n=取值n

WHERE 条件表达式;

11.3 删除数据

删除数据是删除表中已经存在的记录。通过这种方式可以删除表中不再使用的记录。例如，学生表中某个学生退学了，这就需要从学生表中删除该同学的信息。MySQL中，通过DELETE语句来删除数据。MySQL中，DELETE语句的基本语法形式如下：

DELETE FROM 表名 [WHERE 条件表达式] ;

11.4 本章实例

在本小节中将在food表上插入数据、更新数据和删除数据。

字段名	字段描述	数据类型	主键	外键	非空	唯一	自增
id	编号	INT(10)	是	否	是	是	是
name	食品名称	VARCHAR(20)	否	否	是	否	否
company	生产厂商	VARCHAR(30)	否	否	是	否	否
price	价格（单位：圆）	FLOAT	否	否	否	否	否
produce_time	生产年份	YEAR	否	否	否	否	否
validity_time	保质期（单位：年）	INT(4)	否	否	否	否	否
address	厂址	VARCHAR(50)	否	否	否	否	否

11.5 常见问题及解答

1. 插入记录时，哪种情况不需要在INSERT语句中指定字段名？
2. 如何为自增字段（AUTO_INCREMENT）赋值？
3. 如何进行联表删除？

11.6 小结

本章介绍了如何向表中插入数据，如何更新表中已经存在的记录，以及如何删除数据。这些内容都是本章的重点内容。INSERT语句使用非常灵活，读者需要多练习。更新语句和删除语句需要设置查询条件。查询条件一定要合理设置，否则会造成数据丢失。如果没有设置查询条件，更新语句将更新所有数据，删除语句将删除所有数据。学习本章时一定要多练习，在实际操作中掌握本章的内容。下一章将为读者讲解MySQL数据库的运算符。

第12章 MySQL运算符

运算符是用来连接表达式中各个操作数的符号，其作用是用来指明对操作数所进行的运算。MySQL数据库支持使用运算符。通过运算符，可以使数据库的功能更加强大。而且，可以更加灵活的使用表中的数据。MySQL运算符包括四类，分别是算术运算符、比较运算符、逻辑运算符和位运算符。在这一章中将讲解的内容包括：

- **算术运算符**
- **比较运算符**
- **逻辑运算符**
- **位运算符**
- **运算符的优先级**

12.1 运算符简介

当数据库中的表定义好了以后，表中的数据代表的意义就已经定下来了。通过使用运算符进行运算，可以得到包含另一层意义的数据。例如，学生表中存在一个birth字段，这个字段是表示学生的出生年份。如果，用户现在希望查找这个学生的年龄。而学生表中只有出生年份，没有字段表示年龄。这就需要进行运算，用当前的年份减去学生的出生年份，这样就可以计算出学生的年龄了。

从上面可以知道，MySQL运算符可以指明对表中数据所进行的运算，以便得到用户希望得到的数据。这样可以使MySQL数据库更加灵活。MySQL运算符包括算术运算符、比较运算符、逻辑运算符和位运算符这四类。

算术运算符：

比较运算符：

逻辑运算符：

位运算符：

12.2 算术运算符

算术运算符是MySQL中最常用的一类运算符。MySQL支持的算术运算符包括加、减、乘、除、求余。

符号	表达式的形式	作用
+	$x1+x2+\dots+xn$	加法运算
-	$x1-x2-\dots-xn$	减法运算
*	$x1*x2*\dots*xn$	乘法运算
/	$x1/x2$	除法运算，返回x1除以x2的商
DIV	$x1 \text{ DIV } x2$	除法运算，返回商。同“/”
%	$x1\%x2$	求余运算，返回x1除以x2的余数
MOD	$\text{MOD}(x1,x2)$	求余运算，返回余数。同“%”

12.3 比较运算符

比较运算符是查询数据时最常用的一类运算符。SELECT语句中的条件语句经常要使用比较运算符。通过这些比较运算符，可以判断表中的哪些记录是符合条件的。

1. 运算符 “=”
2. 运算符 “<”和 “!=”
3. 运算符 “<=>”
4. 运算符 “>”
5. 运算符 “>=”
6. 运算符 “<”
7. 运算符 “<=”
8. 运算符 “IS NULL”
9. 运算符 “BETWEEN AND”
10. 运算符 “IN”
11. 运算符 “LIKE”
12. 运算符 “REGEXP”

12.4 逻辑运算符

逻辑运算符用来判断表达式的真假。逻辑运算符的返回结果只有1和0。如果表达式是真，结果返回1。如果表达式是假，结果返回0。逻辑运算符又称为布尔运算符。MySQL中支持四种逻辑运算符。这四种逻辑运算符分别是与、或、非和异或。

- 1 . 与运算**
- 2 . 或运算**
- 3 . 非运算**
- 4 . 异或运算**

12.5 位运算符

位运算符是在二进制数上进行计算的运算符。位运算会先将操作数变成二进制数，然后进行位运算。然后再将计算结果从二进制数变回十进制数。MySQL中支持六种位运算符。这六种位运算符分别是按位与、按位或、按位取反、按位异或、按位左移和按位右移。

- 1. 按位与**
- 2. 按位或**
- 3. 按位取反**
- 4. 按位异或**
- 5. 按位左移与按位右移**

12.6 运算符的优先级

由于在实际应用中可能需要同时使用多个运算符。这就必须考虑运算符的运算顺序。到底谁先运算，谁后运算。本小节将给读者讲解运算符的优先级。MySQL的表达式都是从左到右开始运算，哪个运算符的优先级高，哪个运算符先进行计算。

12.7 本章实例

1. 在t表上使用算术运算符和比较运算符进行运算
2. 将数字2, 0和NULL之间的任意两个进行逻辑运算
3. 按下列要求进行位运算

12.8 上机实践

题目要求：

(1) 在MySQL中执行下面的表达式： $4+3-1$ ， $3*2+7$ ， $8/3$ ， $9\%2$ 。

(2) 在MySQL中执行下面的表达式： $30>28$ ， $17\geq 16$ ， $30<28$ ， $17\leq 16$ ， $17=17$ ， $16<>17$ ， $7<=>NULL$ ， $NULL<=>NULL$ 。

(3) 判断字符串“mybook”是否为空，是否以字母m开头，以字母k结尾。

(4) 在MySQL中执行下列逻辑运算： $2\&\&0\&\&NULL$ ， $1.5\&\&2$ ， $3\|\|NULL$ ， $NOT\ NULL$ ， $3\ XOR\ 2$ ， $0\ XOR\ NULL$ 。

(5) 在MySQL中执行下列位运算： $3\&5$ ， $3\|5$ ， $3\^5$ ， ~ 5 。

(6) 将12左移两位，将9右移3三位。

12.9 常见问题及解答

1. 比较运算符的运算结果只能是0和1吗？
2. 哪种运算符的优先级最高？
3. 十进制的数也可以直接使用位运算符吗？

12.10 小结

本章介绍了**MySQL**中的运算符。**MySQL**中包括四类运算符，分别是算术运算符、比较运算符、逻辑运算符、位运算符。前三种运算符在实际操作中使用比较频繁，也是本章中重点讲述的内容。因此，读者需要认真学习这部分的内容。位运算符是本章的难点。因为，位运算符需要将操作数转换为二进制数，然后进行位运算。这要求读者能够掌握二进制运算的相关知识。位运算符在实际操作中使用的频率比较低。下一章将为读者讲解**MySQL**的函数。

12.11 本章习题

1. 在MySQL中执行如下算术表达式： $5*2-4$ ， $(2+7)/3$ ， $9 \text{ DIV } 2$ ， $\text{MOD}(9,2)$ 。
2. 在MySQL中执行下面的比较运算的表达式： $40 \geq 30$ ， $40 \leq 30$ ， $\text{NULL} \leq \text{NULL}$ ， $7 \leq 7$ 。
3. 在MySQL中执行下面的逻辑运算的表达式： $-1 \& 2$ ， $-2 \|\text{NULL}$ ， $\text{NULL} \text{ XOR } 0$ ， $1 \text{ XOR } 0$ ， $!-1$ 。
4. 在MySQL中执行下列位运算： $11 \& 15$ ， $11 | 15$ ， $13 \wedge 15$ ， ~ 15 。

第13章 MySQL函数

MySQL数据库中提供了很丰富的函数。MySQL函数包括数学函数、字符串函数、日期和时间函数、条件判断函数、系统信息函数、加密函数、格式化函数等。通过这些函数，可以简化用户的操作。例如，字符串连接函数可以很方便的将多个字符串连接在一起。在这一章中将讲解的内容包括：

- 数学函数
- 字符串函数
- 日期和时间函数
- 条件判断函数
- 系统信息函数
- 加密函数
- 格式化函数

13.1 MySQL函数简介

MySQL函数是MySQL数据库提供的内部函数。这些内部函数可以帮助用户更加方便的处理表中的数据。本小节中将简单介绍MySQL中包含哪几类函数，以及这几类函数的使用范围和作用。

MySQL函数包括数学函数、字符串函数、日期和时间函数、条件判断函数、系统信息函数、加密函数等。SELECT语句及其条件表达式都可以使用这些函数。同时，INSERT、UPDATE、DELETE语句及其条件表达式也可以使用这些函数。例如，表中的某个数据是负数，现在需要将这个数据显示为正数。这就可以使用绝对值函数。从上面可以知道，MySQL函数可以对表中数据进行相应的处理，以便得到用户希望得到的数据。这些函数可以使MySQL数据库的功能更加强大。

13.2 数学函数

数学函数是MySQL中常用的一类函数。主要用于处理数字，包括整型、浮点数等。数学函数包括绝对值函数、正弦函数、余弦函数、获取随机数的函数等。

13.2.1 绝对值函数ABS(x)和返回圆周率的函数PI()

ABS(x)用来求绝对值；PI()用来返回圆周率。

13.2.2 平方根函数SQRT(x)和求余函数MOD(x,y)

SQRT(x)用来求平方根；MOD(x,y)用来求余数。【示例13-2】下面将演示SQRT(x)、MOD(x,y)这两个函数的使用

。

```
mysql> SELECT SQRT(16), SQRT(2), MOD(5,2);
```

```
+-----+-----+-----+
| SQRT(16) | SQRT(2)      | MOD(5,2) |
+-----+-----+-----+
|      4 | 1.4142135623731 |      1 |
+-----+-----+-----+
```

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

13.2.3 获取整数的函数**CEIL(x)**、**CEILING(x)** 和**FLOOR(x)**

CEIL(x)和**CEILING(x)**这两个函数返回大于或等于x的最小整数；**FLOOR(x)**函数返回小于或等于x的最大整数。

13.2.4 获取随机数的函数RAND()和RAND(x)

RAND()和RAND(x)这两个函数都是返回0 ~ 1的随机数。但是RAND()返回的数是完全随机的，而RAND(x)函数的x相同时返回的值是相同的。

13.2.5 四舍五入函数ROUND(x)、 ROUND(x,y)和TRUNCATE(x,y)

ROUND(x)函数返回离x最近的整数，也就是对x进行四舍五入处理；ROUND(x,y)函数返回x保留到小数点后y位的值，截断时需要进行四舍五入处理；TRUNCATE(x,y)函数返回x保留到小数点后y位的值。

13.2.6 符号函数SIGN(x)

SIGN(x)函数返回x的符号，x是负数、0、正数分别返回-1、0、1。

13.2.7 幂运算函数POW(x,y)、POWER(x,y)和EXP(x)

POW(x,y)和POWER(x,y)这两个函数计算x的y次方，即 x^y ；EXP(x)函数计算e的x次方，即 e^x 。

13.2.8 对数运算函数LOG(x)和LOG10(x)函数

LOG(x)函数计算x的自然对数；LOG10(x)函数计算以10为底的对数。其中，EXP(x)和LOG(x)这两个函数互为反函数。

13.2.9 角度与弧度相互转换的函数 **RADIANS(x)和DEGREES(x)**

RADIANS(x)函数将角度转换为弧度；DEGREES(x)函数将弧度转换为角度。这两个函数互为反函数。

13.2.10 正弦函数SIN(x)和反正弦函数ASIN(x)

SIN(x)函数用来求正弦值，其中x是弧度；ASIN(x)函数用来求反正弦值。ASIN(x)中x的取值必须在-1到1之间。否则返回的结果将会是NULL。

13.2.11 余弦函数**COS(x)**和反余弦函数 **ACOS(x)**

COS(x)函数用来求余弦值，其中x是弧度；**ACOS(x)**函数用来求反余弦值。**COS(x)**和**ACOS(x)**互为反函数。并且，**ACOS(x)**中x的取值必须在-1到1之间。否则返回的结果将会是**NULL**。

13.2.12 正切函数、反正切函数和余切函数

TAN(x)函数用来求正切值，其中x是弧度；ATAN(x)和ATAN2(x)用来求反正切值；COT(x)函数用来求余切值。TAN(x)与ATAN(x)、ATAN2(x)互为反函数。而且TAN(x)返回值是COT(x)返回值的倒数。

13.3 字符串函数

字符串函数是MySQL中最常用的一类函数。字符串函数主要用于处理表中的字符串。字符串函数包括求字符串长度、合并字符串、在字符串中插入子串、大小字母之间切换等函数。

13.3.1 计算字符串字符数的函数和字符串长度的函数

**CHAR_LENGTH(s)函数计算字符串s的字符数；
LENGTH(s)函数计算字符串s的长度。**

13.3.2 合并字符串的函数CONCAT(s1,s2,...) 和CONCAT_WS(x,s1,s2,...)

CONCAT(s1,s2,...)函数和CONCAT_WS(x,s1,s2,...)函数都可以将s1、s2等多个字符串合并成一个字符串。但CONCAT_WS(x,s1,s2,...)可以将各字符串直接用参数x隔开

。

13.3.3 替换字符串的函数

INSERT(s1,x,len,s2)

INSERT(s1,x,len,s2)函数将字符串s1中x位置开始长度为len的字符串用s2替换。

13.3.4 字母大小写转换函数

UPPER(s)函数和UCASE(s)函数将字符串s的所有字母变成大写字母；LOWER(s)函数和LCASE(s)函数将字符串s的所有字母变成小写字母。

13.3.5 获取指定长度的字符串的函数 **LEFT(s,n)和RIGHT(s,n)**

LEFT(s,n)函数返回字符串s的前n个字符；RIGHT(s,n)函数返回字符串s的后n个字符。

13.3.6 填充字符串的函数LPAD(s1,len,s2)和RPAD(s1,len,s2)

LPAD(s1,len,s2)函数将字符串s2填充到s1的开始处，使字符串长度达到len；RPAD(s1,len,s2)函数将字符串s2填充到s1的结尾处，使字符串长度达到len。

13.3.7 删除空格的函数LTRIM(s)、RTRIM(s)和TRIM(s)

**LTRIM(s)函数将去掉字符串s开始处的空格；
RTRIM(s)函数将去掉字符串s结尾处的空格；TRIM(s)函数
将去掉字符串s开始处和结尾处的空格。**

13.3.8 删除指定字符串的函数TRIM(s1 FROM s)

TRIM(s1 FROM s)函数将去掉字符串s中开始处和结尾处的字符串s1。【示例13-20】下面将演示TRIM(s1 FROM s)函数的使用。

```
mysql> SELECT TRIM('ab' FROM 'ababddddabddab');
```

```
+-----+
| TRIM('ab' FROM 'ababddddabddab') |
+-----+
| ddddabdd          |
+-----+
```

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

13.3.9 重复生成字符串的函数REPEAT(s,n)

REPEAT(s,n)函数将字符串s重复n次。【示例13-21】
下面将演示REPEAT(s,n)函数的使用。

```
mysql> SELECT REPEAT('mysql-',2);
```

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

```
| REPEAT('mysql-',2) |
```

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

```
| mysql-mysql-      |
```

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

```
1 row in set (0.05 sec)
```

13.3.10 空格函数SPACE(n)和替换函数 REPLACE(s,s1,s2)

**SPACE(n)函数返回n个空格；REPLACE(s,s1,s2)函数
将字符串s2替代字符串s中的字符串s1。**

13.3.11 比较字符串大小的函数 **STRCMP(s1,s2)**

STRCMP(s1,s2)函数用来比较字符串s1和s2。如果s1大于s2，结果返回1；如果s1等于s2，结果返回0；如果s1小于s2，结果返回-1。

13.3.12 获取子串的函数 SUBSTRING(s,n,len)和MID(s,n,len)

SUBSTRING(s,n,len)函数和MID(s,n,len)函数从字符串s的第n个位置开始获取长度为len的字符串。【示例13-24】下面将演示SUBSTRING(s,n,len)函数和MID(s,n,len)函数的使用。

```
mysql> SELECT s, SUBSTRING(s,4,3), MID(s,4,3)
FROM t2;
```

```
+-----+-----+-----+
| s     | SUBSTRING(s,4,3) | MID(s,4,3) |
+-----+-----+-----+
| beijing | jin             | jin       |
+-----+-----+-----+
```

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

13.3.13 匹配子串开始位置的函数

LOCATE(s1,s)、POSITION(s1 IN s)和INSTR(s,s1)这三个函数从字符串s中获取s1的开始位置。

【示例13-25】下面将演示LOCATE(s1,s)、POSITION(s1 IN s)和INSTR(s,s1)这三个函数的使用。

```
mysql> SELECT s,LOCATE('jin',s), POSITION('jin' IN s), INSTR(s,'jin') FROM t2;
```

```
+-----+-----+-----+-----+
| s      | LOCATE('jin',s) | POSITION('jin' IN s) | INSTR(s,'jin') |
+-----+-----+-----+-----+
| beijing |          4      |          4          |          4      |
+-----+-----+-----+-----+
```

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

13.3.14 字符串逆序的函数REVERSE(s)

REVERSE(s)函数将字符串s的顺序反过来。【示例13-26】下面将演示REVERSE(s)函数的使用。

```
mysql> SELECT s, REVERSE(s) FROM t2;
```

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

```
| s      | REVERSE(s) |
```

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

```
| beijing | gnijieb    |
```

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

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

13.3.15 返回指定位置的字符串的函数

ELT(n,s1,s2,...)函数返回第n个字符串。【示例13-27】
下面将演示ELT(n,s1,s2,...)函数的使用。

```
mysql> SELECT ELT(2,'me','my','he','she');
```

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

```
| ELT(2,'me','my','he','she') |
```

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

```
| my          |
```

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

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

13.3.16 返回指定字符串位置的函数 **FIELD(s,s1,s2,...)**

FIELD(s,s1,s2,...)函数返回第一个与字符串s匹配的字符串的位置。【示例13-28】下面将演示FIELD(s,s1,s2,...)函数的使用。

```
mysql> SELECT FIELD('he','me','my','he','she');
```

```
+-----+  
| FIELD('he','me','my','he','she') |  
+-----+  
|           3 |  
+-----+
```

13.3.17 返回子串位置的函数 **FIND_IN_SET(s1,s2)**

FIND_IN_SET(s1,s2)函数返回在字符串s2中与s1匹配的字符串的位置。其中，字符串s2中包含了若干个用逗号隔开的字符串。【示例13-29】下面将演示FIND_IN_SET(s1,s2)函数的使用。

```
mysql> SELECT FIND_IN_SET('like','i,like,bei,jing');
```

```
+-----+  
| FIND_IN_SET('like','i,like,bei,jing') |  
+-----+  
|                2 |  
+-----+
```

13.3.18 选取字符串的函数 MAKE_SET(x,s1,s2,...)函数

MAKE_SET(x,s1,s2,...)函数按x的二进制数从s1,s2,...,sn中选取字符串。例如12的二进制是1100。这个二进制数从右到左的第三位和第四位是1，所以选取s3和s4。【示例13-30】下面将演示MAKE_SET(x,s1,s2,...)函数的使用

。

```
mysql> SELECT MAKE_SET(11,'a','b','c','d'), MAKE_SET(7,'a','b','c','d');
+-----+-----+
| MAKE_SET(11,'a','b','c','d') | MAKE_SET(7,'a','b','c','d') |
+-----+-----+
| a,b,d           | a,b,c           |
+-----+-----+
1 row in set (0.00 sec)
```

13.4 日期和时间函数

日期和时间函数是MySQL中另一类最常用的函数。日期和时间函数主要用于处理表中的日期和时间数据。日期和时间函数包括获取当前日期的函数、获取当前时间的函数、计算日期的函数、计算时间的函数等。

13.4.1 获取当前日期的函数和获取当前时间的函数

CURDATE()和CURRENT_DATE()函数获取当前日期；CURTIME()和CURRENT_TIME()函数获取当前时间。【示例13-31】下面将演示CURDATE()、CURRENT_DATE()、CURTIME()和CURRENT_TIME()这四个函数的使用。

```
mysql> SELECT
CURDATE(),CURRENT_DATE(),CURTIME(),CURRENT_TIME();
+-----+-----+-----+-----+
| CURDATE() | CURRENT_DATE() | CURTIME() | CURRENT_TIME() |
+-----+-----+-----+-----+
| 2009-10-25 | 2009-10-25      | 15:47:56 | 15:47:56      |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

13.4.2 获取当前日期和时间的函数

NOW()、CURRENT_TIMESTAMP()、LOCALTIME()和SYSDATE()这四个函数都用来获取当前的日期和时间。这四个函数表示相同的含义。【示例13-32】下面将演示NOW()、CURRENT_TIMESTAMP()、LOCALTIME()和SYSDATE()这四个函数的使用。

```
mysql> SELECT NOW(),CURRENT_TIMESTAMP(),LOCALTIME(),SYSDATE();
+-----+-----+-----+-----+
| NOW()          | CURRENT_TIMESTAMP() | LOCALTIME()      | SYSDATE()        |
+-----+-----+-----+-----+
| 2009-10-25 15:53:21 | 2009-10-25 15:53:21 | 2009-10-25 15:53:21 | 2009-10-25
15:53:21 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

13.4.3 UNIX时间戳函数

UNIX_TIMESTAMP()函数以UNIX时间戳的形式返回当前时间；UNIX_TIMESTAMP(d)函数将时间d以UNIX时间戳的形式返回；FROM_UNIXTIME(d)函数把UNIX时间戳的时间转换为普通格式的时间。UNIX_TIMESTAMP(d)函数和FROM_UNIXTIME(d)互为反函数。

13.4.4 返回UTC日期的函数和返回UTC时间的函数

UTC_DATE()函数返回UTC日期；UTC_TIME()函数返回UTC时间。其中，UTC是Universal Coordinated Time的缩写，也就是国际协调时间。

13.4.5 获取月份的函数MONTH(d)和MONTHNAME(d)

MONTH(d)函数返回日期d中的月份值，其取值范围是1~12；MONTHNAME(d)函数返回日期d中的月份的英文名称，如January,February等。其中，参数d可以是日期和时间，也可以是日期。

13.4.6 获取星期的函数DAYNAME(d)、DAYOFWEEK(d)和WEEKDAY(d)

DAYNAME(d)函数返回日期d是星期几，显示其英文名，如Monday,Tuesday等；DAYOFWEEK(d)函数也返回日期d是星期几，1表示星期日，2表示星期一，依次类推；WEEKDAY(d)函数也返回日期d是星期几，0表示星期一，1表示星期二，依次类推。其中，参数d可以是日期和时间，也可以是日期。

13.4.7 获取星期数的函数WEEK(d)和WEEKOFYEAR(d)

WEEK(d)函数和WEEKOFYEAR(d)函数都是计算日期d是本年的第几个星期。返回值的范围是1~53。【示例13-37】下面将演示WEEK(d)函数和WEEKOFYEAR(d)函数的使用。

```
mysql> SELECT d,WEEK(d),WEEKOFYEAR(d),dt,WEEK(dt),WEEKOFYEAR(dt)
FROM t4;
```

```
+-----+-----+-----+-----+-----+-----+
-----+
| d      | WEEK(d) | WEEKOFYEAR(d) | dt          | WEEK(dt) | WEEKOFYEAR(dt) |
+-----+-----+-----+-----+-----+-----+
| 2009-10-25 | 43 | 43 | 2009-10-25 16:15:59 | 43 | 43 |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

13.4.8 获取天数的函数DAYOFYEAR(d)和DAYOFMONTH(d)

**DAYOFYEAR(d)函数日期d是本年的第几天；
DAYOFMONTH(d)函数返回计算日期d是本月的第几天。【
示例13-38】下面将演示DAYOFYEAR(d)函数和
DAYOFMONTH(d)函数的使用。**

```
mysql> SELECT d, DAYOFYEAR(d), DAYOFMONTH(d) FROM t4;
```

```
+-----+-----+-----+
| d      | DAYOFYEAR(d) | DAYOFMONTH(d) |
+-----+-----+-----+
| 2009-10-25 |      298 |      25 |
+-----+-----+-----+
```

13.4.9 获取年份、季度、小时、分钟、秒钟的函数

YEAR(d)函数返回日期d中的年份值；QUARTER(d)函数返回日期d是今年第几季度，值的范围是1~4；HOUR(t)函数返回时间t中的小时值；MINUTE(t)函数返回时间t中的分钟值；SECOND(t)函数返回时间t中的秒钟值。

13.4.10 获取日期的指定值的函数 **EXTRACT(type FROM d)**

EXTRACT(type FROM d)函数从日期d中获取指定的值。这个值是什么由type的值决定。type的取值可以是YEAR、MONTH、DAY、HOUR、MINUTE、SECOND。如果type的值是YEAR，结果返回年份值；MONTH返回月份值；DAY返回是几号；HOUR返回小时值；MINUTE返回分钟值；SECOND返回秒钟值。

13.4.11 时间和秒钟转换的函数

TIME_TO_SEC(t)函数将时间t转换为以秒为单位的时间；SEC_TO_TIME(s)函数将以秒为单位的时间s转换为时分秒的格式。TIME_TO_SEC(t)和SEC_TO_TIME(s)互为反函数。【示例13-41】下面将演示TIME_TO_SEC(t)函数和SEC_TO_TIME(s)函数的使用。

```
mysql> SELECT t,TIME_TO_SEC(t),SEC_TO_TIME(58559) FROM t4;
```

```
+-----+-----+-----+
| t      | TIME_TO_SEC(t) | SEC_TO_TIME(58559) |
+-----+-----+-----+
| 16:15:59 |      58559 | 16:15:59          |
+-----+-----+-----+
```

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

13.4.12 计算日期和时间的函数

1 . TO_DAYS(d)、FROM_DAYS(n)和
DATEDIFF(d1,d2)函数

2 . ADDDATE(d,n)、SUBDATE(d,n)、ADDTIME(t,n)
和SUBTIME(t,n)函数

3 . ADDDATE(d,INTERVAL expr type)和
DATE_ADD(d,INTERVAL expr type)函数

13.4.13 将日期和时间格式化的函数

- 1 . **DATE_FORMAT(d,f)函数**
- 2 . **TIME_FORMAT(t,f)函数**
- 3 . **GET_FORMAT(type,s)函数**

13.5 条件判断函数

条件判断函数用来在SQL语句中进行条件判断。根据是否满足判断条件，SQL语句执行不同的分支。例如，从员工表中查询员工的业绩。如果业绩高于指定值n，则输出“good”。否则，输出“bad”。下面是各种条件判断函数的表达式、作用和使用方法。

13.5.1 IF(expr,v1,v2)函数

IF(expr,v1,v2)函数中，如果表达式expr成立，返回结果v1；否则，返回结果v2。【示例13-51】下面从t6中查询学号（id），分数（grade）。并且，分数大于等于60，显示“PASS”。否则，显示“FAIL”。SELECT语句如下：

```
SELECT id,grade,IF(grade>=60,'PASS','FAIL') from  
t6;
```

13.5.2 IFNULL(v1,v2)函数

IFNULL(v1,v2)函数中，如果v1的不为空，就显示v1的值；否则就显示v2的值。【示例13-52】下面从t6中查询学号（id），分数（grade）。如果分数不为NULL，显示分数。否则，显示“NO GRADE”。SELECT语句如下：

```
SELECT id,IFNULL(grade, 'NO GRADE') FROM t6;
```

13.5.3 CASE函数

1 . CASE WHEN expr1 THEN v1 [WHEN expr2 THEN v2...] [ELSE vn] END

2 . CASE expr WHEN e1 THEN v1 [WHEN e2 THEN v2...] [ELSE vn] END

13.6 系统信息函数

系统信息函数用来查询MySQL数据库的系统信息。例如，查询数据库的版本，查询数据库的当前用户等。本小节将详细讲解系统信息函数的作用和使用方法。

13.6.1 获取MySQL版本号、连接数、数据库名的函数

VERSION()函数返回数据库的版本号；
CONNECTION_ID()函数返回服务器的连接数，也就是到现在为止MySQL服务的连接次数；
DATABASE()和SCHEMA()返回当前数据库名。

13.6.2 获取用户名的函数

USER()、SYSTEM_USER()、SESSION_USER()、CURRENT_USER()和CURRENT_USER这几个函数可以返回当前用户的名称。

13.6.3 获取字符串的字符集和排序方式的函数

CHARSET(str)函数返回字符串str的字符集，一般情况下这个字符集就是系统的默认字符集；COLLATION(str)函数返回字符串str的字符排列方式。

13.6.4 获取最后一个自动生成的ID值的函数

**LAST_INSERT_ID()函数返回最后生成的
AUTO_INCREMENT值。**

13.7 加密函数

加密函数是MySQL中用来对数据进行加密的函数。因为数据库中有些很敏感的信息不希望被其他人看到，就应该通过加密方式来使这些数据变成看似乱码的数据。例如用户的密码，就应该经过加密。本小节将详细讲解加密函数的作用和使用方法。

下面是各种加密函数的名称、作用和使用方法。

13.7.1 加密函数PASSWORD(str)

PASSWORD(str)函数可以对字符串str进行加密。一般情况下，PASSWORD(str)函数主要是用来给用户的密码加密的。【示例13-59】下面使用PASSWORD(str)函数为字符串“abcd”加密。

```
mysql> SELECT PASSWORD('abcd');
```

```
PASSWORD('abcd')
```

```
A154C52565E9E7F94BFC08A1FE702624ED8EFFDA
```

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

13.7.2 加密函数MD5(str)

MD5(str)函数可以对字符串str进行加密。MD5(str)函数主要对普通的数据进行加密。【示例13-60】下面使用MD5(str)函数为字符串“abcd”加密。

```
mysql> SELECT MD5('abcd');
```

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

```
| MD5('abcd') |
```

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

```
| e2fc714c4727ee9395f324cd2e7f331f |
```

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

```
1 row in set (0.05 sec)
```

13.7.3 加密函数ENCODE(str,pswd_str)

ENCODE(str,pswd_str)函数可以使用字符串pswd_str来加密字符串str。加密的结果是一个二进制数，必须使用BLOB类型的字段来保存它。

13.7.4 解密函数

DECODE(encrypt_str,pswd_str)

DECODE(encrypt_str,pswd_str)函数可以使用字符串pswd_str来为encrypt_str解密。encrypt_str是通过ENCODE(str,pswd_str)加密后的二进制数据。字符串pswd_str应该与加密时的字符串pswd_str是相同的。【示例13-62】下面使用DECODE(encrypt_str,pswd_str)为ENCODE(str,pswd_str)加密的数据解密。

```
mysql> SELECT DECODE(ENCODE('abcd','aa'),'aa');
```

```
+-----+
| DECODE(ENCODE('abcd','aa'),'aa') |
+-----+
| abcd          |
+-----+
```

```
1 row in set (0.02 sec)
```

13.8 其它函数

MySQL中除了上述函数以外，还包含了很多函数。例如FORMAT(x,n)函数用来格式化数字x，INET_ATON()函数可以将IP转换为数字。本小节将详细讲解这些函数的作用和使用方法。

13.8.1 格式化函数FORMAT(x,n)

FORMAT(x,n)函数可以将数字x进行格式化，将x保留到小数点后n位。这个过程需要进行四舍五入。例如
FORMAT(2.356,2)返回的结果将会是2.36；
FORMAT(2.353,2)返回的结果将会是2.35。【示例13-63】下面使用FORMAT(x,n)函数来讲235.3456和235.3454进行格式化，都保留到小数点后3位。

```
mysql> SELECT  
FORMAT(235.3456,3),FORMAT(235.3454,3);
```

13.8.2 不同进制的数字进行转换的函数

ASCII(s)返回字符串s的第一个字符的ASCII码；**BIN(x)**返回x的二进制编码；**HEX(x)**返回x的十六进制编码；**OCT(x)**返回x的八进制编码；**CONV(x,f1,f2)**将x从f1进制数变成f2进制数。

13.8.3 IP地址与数字相互转换的函数

INET_ATON(IP)函数可以将IP地址转换为数字表示；INET_NTOA(n)函数可以将数字n转换成IP的形式。其中，INET_ATON(IP)函数中IP值需要加上引号。这两个函数互为反函数。

13.8.4 加锁函数和解锁函数

GET_LOCK(name,time)函数定义一个名称为name、持续时间长度为time秒的锁。如果锁定成功，返回1；如果尝试超时，返回0；如果遇到错误，返回NULL。

RELEASE_LOCK(name)函数解除名称为name的锁。如果解锁成功，返回1；如果尝试超时，返回0；如果解锁失败，返回NULL；**IS_FREE_LOCK(name)**函数判断是否使用名为name的锁。如果使用，返回0；否则，返回1。

13.8.5 重复执行指定操作的函数

BENCHMARK(count,expr)函数将表达式expr重复执行count次，然后返回执行时间。该函数可以用来判断MySQL处理表达式的速度。

13.8.6 改变字符集的函数

CONVERT(s USING cs)函数将字符串s的字符集变成cs。
。【示例13-68】下面将字符串“ABC”的字符集变成gbk。

```
mysql> SELECT  
CHARSET('ABC'),CHARSET(CONVERT('ABC' USING  
gbk));
```

13.8.7 改变字段数据类型的函数

CAST(x AS type)和CONVERT(x,type)这两个函数将x变成type类型。这两个函数只对BINARY、CHAR、DATE、DATETIME、TIME、SIGNED INTEGER、UNSIGNED INTEGER这些类型起作用。但两种方法只是改变了输出值的数据类型，并没有改变表中字段的类型。

13.9 本章实例

1. 生成三个1~100之间的随机整数

2. 计算PI (圆周率) 的余弦值和自然对数值

3. 按如下要求来操作表

4. 先加一个名为 “mybook1”的锁，持续时间为20秒。

然后马上加一个名为 “mybook2”的锁，持续时间为30秒。
然后查询这两个锁的状态。最后解除这两个锁。

13.10 上机实践

题目要求：

(1) 向num_test表中插入记录。

(2) 在字符串 “I love ”和字符串 “beijing”合并为同一个字符串。

(3) 返回字符串 “me”在字符串 “You love me. He love me.”中第一次出现的位置。

(4) 用GET_FORMAT(DATE, 'EUR')返回的格式来显示当前日期；用GET_FORMAT(TIME, 'USA')返回的格式来显示当前时间。

(5) 查看当前数据库的版本号，当前数据库名和当前用户。

(6) 使用字符串 “college”来加密字符串 “university”

13.11 常见问题及解答

1. 表中birth字段存的出生日期，如何来计算年龄？
2. 如何改变字符串的字符集？
3. 用户的密码应该怎么加密？

13.12 小结

本章介绍了MySQL数据库提供的内部函数。这些函数包括数学函数、字符串函数、日期和时间函数、条件判断函数、系统信息函数、加密函数等。字符串函数和日期和时间函数是本章重点介绍的内容。条件判断函数是本章的难点，因为条件判断函数涉及很多条件判断和跳转的语句。这些函数经常与SELECT语句一起使用，用来方便用户的查询。同时INSERT、UPDATE、DELETE语句和条件表达式也可以使用这些函数。读者一定要上机实际操作这些函数，这样可以对函数了解得更加透彻。下一章将为读者讲解如何创建、修改、删除和使用存储过程和函数。

第14章 存储过程和函数

存储过程和函数是在数据库中定义一些SQL语句的集合，然后直接调用这些存储过程和函数来执行已经定义好的SQL语句。存储过程和函数可以避免开发人员重复的编写相同的SQL语句。而且，存储过程和函数是在MySQL服务器中存储和执行的，可以减少客户端和服务端的数据传输。在这一章中将讲解的内容包括：

创建存储过程

变量的使用

光标的使用

调用存储过程和函数

修改存储过程和函数

创建存储函数

定义条件和处理程序

流程控制的使用

查看存储过程和函数

删除存储过程和函数

14.1 创建存储过程和函数

创建存储过程和函数是指将经常使用的一组SQL语句的组合在一起，并将这些SQL语句当作一个整体存储在MySQL服务器中。例如，银行经常需要计算用户的利息。不同类别的用户的利率是不一样的。这就可以将计算利率的SQL代码写成一个存储过程或者存储函数。只要调用这个存储过程或者存储函数，就可以将不同类别用户的利息计算出来。本节将向读者介绍创建存储过程和函数的方法，并且将讲解如何定义变量、如何定义条件和处理、如何使用光标和如何使用流程控制。

14.1.1 创建存储过程

MySQL中，创建存储过程的基本形式如下：

```
CREATE PROCEDURE sp_name  
([proc_parameter[,...]])  
[characteristic ...] routine_body
```

14.1.2 创建存储函数

MySQL中，创建存储函数的基本形式如下：

CREATE FUNCTION sp_name ([func_parameter[,...]])

RETURNS type

[characteristic ...] routine_body

14.1.3 变量的使用

存储过程和函数中可以定义和使用变量。用户可以使用 DECLARE 关键字来定义变量。然后可以为变量赋值。这些变量的作用范围是 BEGIN...END 程序段中。本小节将讲解如何定义变量和为变量赋值。

- 1 . 定义变量**
- 2 . 为变量赋值**

14.1.4 定义条件和处理程序

定义条件和处理程序是事先定义程序执行过程中可能遇到的问题。并且可以在处理程序中定义解决这些问题的办法。这种方式可以提前预测可能出现的问题，并提出解决办法。这样可以增强程序处理问题的能力，避免程序异常停止。MySQL中都是通过DECLARE关键字来定义条件和处理程序。本小节中将详细讲解如何定义条件和处理程序。

- 1. 定义条件**
- 2. 定义处理程序**

14.1.5 光标的使用

查询语句可能查询出多条记录，在存储过程和存储函数中使用光标来逐条读取查询结果集中的记录。有些书上将光标称为游标。光标的使用包括声明光标、打开光标、使用光标和关闭光标。光标必须声明在处理程序之前，并且声明在变量和条件之后。

- 1 . 声明光标**
- 2 . 打开光标**
- 3 . 使用光标**
- 4 . 关闭光标**

14.1.6 流程控制的使用

存储过程和存储函数中可以使用流程控制来控制语句的执行。MySQL中可以使用IF语句、CASE语句、LOOP语句、LEAVE语句、ITERATE语句、REPEAT语句、WHILE语句来进行流程控制。本小节将详细讲解这些流程控制语句

。

- 1 . IF语句
- 2 . CASE语句
- 3 . LOOP语句
- 4 . LEAVE语句
- 5 . ITERATE语句
- 6 . REPEAT语句
- 7 . WHILE语句

14.2 调用存储过程和函数

存储过程和存储函数都是存储在服务器端的SQL语句的集合。要使用这些已经定义好的存储过程和存储函数就必须要通过调用的方式来实现。存储过程是通过CALL语句来调用的。而存储函数的使用方法与MySQL内部函数的使用方法是一样的。执行存储过程和存储函数需要拥有EXECUTE权限。EXECUTE权限的信息存储在information_schema数据库下面的USER_PRIVILEGES表中。本小节将详细讲解如何调用存储过程和存储函数。

14.2.1 调用存储过程

MySQL中使用CALL语句来调用存储过程。调用存储过程后，数据库系统将执行存储过程中的语句。然后，将结果返回给输出值。CALL语句的基本语法形式如下：

```
CALL sp_name([parameter[,...]]);
```

14.2.2 调用存储函数

MySQL中，存储函数的使用方法与MySQL内部函数的使用方法是一样的。换言之，用户自己定义的存储函数与MySQL内部函数是一个性质的。区别在于，存储函数是用户自己定义的，而内部函数是MySQL的开发者定义的。

14.3 查看存储过程和函数

存储过程和函数创建以后，用户可以查看存储过程和函数的状态和定义。用户可以通过SHOW STATUS语句来查看存储过程和函数的状态，也可以通过SHOW CREATE语句来查看存储过程和函数的定义。用户也可以通过查询information_schema数据库下的Routines表来查看存储过程和函数的信息。本小节将详细讲解查看存储过程和函数的状态和定义的方法。

14.3.1 SHOW STATUS语句查看存储过程和函数的状态

MySQL中可以通过SHOW STATUS语句查看存储过程和函数的状态。其基本语法形式如下：

```
SHOW { PROCEDURE | FUNCTION } STATUS [
LIKE ' pattern ' ] ;
```

其中，“PROCEDURE”参数表示查询存储过程；
“FUNCTION”参数表示查询存储函数；“LIKE ' pattern ’”参数用来匹配存储过程或函数的名称。

14.3.2 SHOW CREATE语句查看存储过程和函数的定义

MySQL中可以通过SHOW CREATE语句查看存储过程和函数的状态。其基本语法形式如下：

```
SHOW CREATE { PROCEDURE | FUNCTION }  
sp_name ;
```

其中，“PROCEDURE”参数表示查询存储过程；
“FUNCTION”参数表示查询存储函数；“sp_name”参数表示存储过程或函数的名称。

14.3.3 从information_schema.Routines表中查看存储过程和函数的信息

存储过程和函数的信息存储在information_schema数据库下的Routines表中。可以通过查询该表的记录来查询存储过程和函数的信息。其基本语法形式如下：

```
SELECT * FROM information_schema.Routines  
WHERE ROUTINE_NAME=' sp_name ' ;
```

其中，ROUTINE_NAME字段中存储的是存储过程和函数的名称；“sp_name”参数表示存储过程或函数的名称

。

14.4 修改存储过程和函数

修改存储过程和函数是指修改已经定义好的存储过程和函数。MySQL中通过ALTER PROCEDURE语句来修改存储过程。通过ALTER FUNCTION语句来修改存储函数。本小节将详细讲解修改存储过程和函数的方法。

MySQL中修改存储过程和函数的语句的语法形式如下

:

ALTER {PROCEDURE | FUNCTION} sp_name

[characteristic ...]

characteristic:

**{ CONTAINS SQL | NO SQL | READS SQL DATA |
MODIFIES SQL DATA }**

| SQL SECURITY { DEFINER | INVOKER }

| COMMENT 'string'

14.5 删除存储过程和函数

删除存储过程和函数指删除数据库中已经存在的存储过程和函数。MySQL中使用DROP PROCEDURE语句来删除存储过程。通过DROP FUNCTION语句来删除存储函数。其基本形式如下：

```
DROP { PROCEDURE| FUNCTION } sp_name;
```

14.6 本章实例

本小节将在food表上创建名为food_price_count的存储过程。按照11.4小节中表11.1和表11.2来创建food表。存储过程food_price_count有三个参数。输入参数为price_info1和price_info2，输出参数为count。存储过程的作用是查询food表中食品单价高于price_info1且低于price_info2的食品种数，然后由count参数来输出。并且计算满足条件的单价的总和。

14.7 上机实践

题目要求：

本小节将在teacher表上创建名为teacher_info1的存储过程和名为teacher_info2的存储函数。按照11.5小节中表11.3和表11.4来创建teacher表。

(1) 存储过程teacher_info1的要求：

(2) 存储函数teacher_info2的要求：

操作如下：

- 1. 创建并使用存储过程teacher_info1**
- 2. 创建并使用存储函数teacher_info2**

14.8 常见问题及解答

1. 一个存储过程中可以调用其它的存储过程吗？
2. 存储过程和存储函数的区别是什么？
3. 存储函数和MySQL内部函数有什么区别？

14.9 小结

本章介绍了MySQL数据库的存储过程和存储函数。存储过程和存储函数都是用户自己定义的SQL语句的集合。它们都存储在服务器端，只要调用就可以在服务器端执行。本章重点讲解了创建存储过程和存储函数的方法。通过CREATE PROCEDURE语句来创建存储过程，通过CREATE FUNCTION语句来创建存储函数。这两个内容也是本章的难点，尤其是变量、条件、光标、流程控制的使用。这些需要读者将书中的知识点结合实际操作进行练习。下一章将介绍MySQL用户管理。

14.10 本章习题

1.在10.10小节的employee表上创建存储过程employee_info_procedure。该存储过程的输入参数type，输出参数是info。当type的值是1时，计算employee表中所有员工的平均工资，然后通过参数info输出；当type的值是2时，计算employee表中所有员工的平均年龄，然后通过info输出；当type为1和2以外的任何值时，将字符串“Error Input!”赋值给info。

2.创建存储函数employee_info_function来实现习题1的功能。该函数只有一个参数type。通过RETURN语句来将查询结果返回。

3.删除习题1中的存储过程和习题2中的存储函数。

第15章 MySQL用户管理

MySQL用户包括普通用户和root用户。这两种用户的权限是不一样的。root用户是超级管理员，拥有所有的权限。root用户的权限包括创建用户、删除用户、修改普通用户的密码等管理权限。而普通用户只拥有创建该用户时赋予它的权限。用户管理包括管理用户的帐户、权限等。在这一章中将讲解的内容包括。

- 权限表介绍
- 用户登录和退出MySQL服务器
- 创建和删除普通用户
- 普通用户和root用户的密码管理
- 权限管理

15.1 权限表

安装MySQL时会自动安装一个名为mysql的数据库。mysql数据库下面存储的都是权限表。用户登录以后，MySQL数据库系统会根据这些权限表的内容为每个用户赋予相应的权限。这些权限表中最重要的是user表、db表和host表。除此之外，还有tables_priv表、columns_priv表、proc_priv表等。本节将为读者介绍这些表的内容。

15.1.1 user表

user表是MySQL中最重要的一个权限表。读者可以使用DESC语句来查看user表的基本结构。user表有39个字段。这些字段大致可以分为四类，分别是用户列、权限列、安全列和资源控制列。本小节将为读者介绍这些字段含义。

- 1. 用户列**
- 2. 权限列**
- 3. 安全列**
- 4. 资源控制列**

15.1.2 db表和host表

db表和host表也是MySQL数据库中非常重要的权限表。db表中存储了某个用户对一个数据库的权限。db表比较常用，而host表很少会用到。读者可以使用DESC语句来查看这两个表的基本结构。这两个表的表结构差不多。db表和host表的字段大致可以分为两类，分别是用户列和权限列。

- 1. 用户列**
- 2. 权限列**

15.1.3 tables_priv表和columns_priv表

tables_priv表可以对单个表进行权限设置。columns_priv表可以对单个数据列进行权限设置。读者可以使用DESC语句来查看这两个表的基本结构。本小节将介绍这两个表的内容。

tables_priv表包含8个字段，分别是Host、Db、User、Table_name、Table_priv、Column_priv、Timestamp和Grantor。前四个字段分别表示主机名、数据库名、用户名和表名。Table_priv表示对表进行操作的权限。这些权限包括Select、Insert、Update、Delete、Create、Drop、Grant、References、Index和Alter。Column_priv表示对表中的数据列进行操作的权限。这些权限包括Select、Insert、Update和References。Timestamp表示修改权限的时间。Grantor表示权限是谁设置的。

15.1.4 procs_priv表

procs_priv表可以存储过程和存储函数进行权限设置。读者可以使用DESC语句来查看procs_priv表的基本结构。本小节将为读者介绍procs_priv表中各字段的含义。

procs_priv表包含8个字段，分别是Host、Db、User、Routine_name、Routine_type、Proc_priv、Timestamp和Grantor。前三个字段分别表示主机名、数据库名和用户名。Routine_name字段表示存储过程或函数的名称。Routine_type字段表示类型。该字段有两个取值，分别是FUNCTION和PROCEDURE。FUNCTION表示这是一个存储函数。PROCEDURE表示这是一个存储过程。Proc_priv字段表示拥有的权限。权限分为3类，分别是Execute，Alter Routine和Grant。Timestamp字段存储更新的时间。Grantor字段存储权限是谁设置的。

15.2 帐户管理

帐户管理是MySQL用户管理的最基本的内容。帐户管理包括登录和退出MySQL服务器、创建用户、删除用户、密码管理、权限管理等内容。通过帐户管理，可以保证MySQL数据库的安全性。本小节将向读者详细介绍这些内容。

15.2.1 登录和退出MySQL服务器

用户可以通过mysql命令来登录MySQL服务器。在第2章和第3章已经简单介绍过一些登录MySQL服务器的方法，但是有些参数还不全。在本小节中将详细的介绍mysql命令的参数和退出MySQL服务器的方法。启动MySQL服务后，可以通过mysql命令来登录MySQL服务器。命令如下：

```
mysql -h hostname|hostIP -P port -u username -p  
DatabaseName -e "SQL语句"
```

15.2.2 新建普通用户

在MySQL数据库中，可以使用CREATE USER语句来创建新的用户，也可以直接在mysql.user表中添加用户。还可以使用GRANT语句来新建用户。本小节将为读者介绍这三种方法。

1. 用CREATE USER语句来新建普通用户
2. 用INSERT语句来新建普通用户
3. 用GRANT语句来新建普通用户

15.2.3 删除普通用户

在MySQL数据库中，可以使用DROP USER语句来删除普通用户，也可以直接在mysql.user表中删除用户。本小节将为读者介绍这两种方法。

- 1. 用DROP USER语句来删除普通用户**
- 2. 用DELETE语句来删除普通用户**

15.2.4 root用户修改自己的密码

root用户拥有很高的权限，因此必须保证root用户的密码的安全。root用户可以通过多种方式来修改密码。本小节将介绍几种root用户修改自己的密码的方法。

- 1. 使用mysqladmin命令来修改root用户的密码**
- 2. 修改mysql数据库下的user表**
- 3. 使用SET语句来修改root用户的密码**

15.2.5 root用户修改普通用户密码

root用户不仅可以修改自己的密码，还可以修改普通用户的密码。root用户登录MySQL服务器后，可以通过SET语句、修改user表和GRANT语句来修改普通用户的密码。本小节将向读者介绍root用户修改普通用户密码的方法。

- 1．使用SET语句来修改普通用户的密码**
- 2．修改mysql数据库下的user表**
- 3．用GRANT语句来修改普通用户的密码**

15.2.6 普通用户修改密码

普通用户也可以修改自己的密码。这样普通用户就不需要每次需要修改密码时都通知管理员。普通用户登录到MySQL服务器后，可以通过SET语句来设置自己的密码。SET语句的基本形式为：

```
SET PASSWORD=PASSWORD('new_password');
```

15.2.7 root用户密码丢失的解决办法

root用户密码丢失了，会给用户造成很大的麻烦。但是，可以通过某种特殊方法登录到root用户下。然后在root用户下设置新的密码。下面是解决root用户密码丢失的方法，执行步骤如下：

- 1．使用--skip-grant-tables选项启动MySQL服务**
- 2．登录root用户，并且设置新的密码**
- 3．加载权限表**

15.3 权限管理

权限管理主要是对登录到数据库的用户进行权限验证。所有用户的权限都存储在MySQL的权限表中。数据库管理员要对权限进行管理。合理的权限管理能够保证数据库系统的安全。不当的权限设置可能会给数据库系统带来意想不到的危害。本节将为读者介绍权限管理的内容。

15.3.1 MySQL的各种权限

MySQL数据库中有很多种类的权限，这些权限都存储在mysql数据库下的权限表中。其中，user表中的权限种类最多。本小节将为读者介绍MySQL中的各种权限。

15.3.2 授权

授权就是为某个用户赋予某些权限。例如，可以为新建的用户赋予查询所有数据库和表的权限。合理的授权能够保证数据库的安全。不合理的授权会使数据库存在安全隐患。MySQL中使用GRANT关键字来为用户设置权限。本小节将为读者介绍授权的方法。

MySQL中，必须拥有GRANT权限的用户才可以执行GRANT语句。GRANT语句的基本语法如下：

```
GRANT priv_type [(column_list)] ON database.table  
TO user [ IDENTIFIED BY [PASSWORD] 'password' ]  
[, user [ IDENTIFIED BY [PASSWORD] 'password' ] ]
```

...

```
[WITH with_option [with_option] ...]
```

15.3.3 收回权限

收回权限就是取消某个用户的某些权限。例如，如果数据库管理员觉得某个用户不应该拥有DELETE权限，那么就可以将DELETE权限收回。收回权限的方式可以保证数据库的安全。MySQL中使用REVOKE关键字来为用户设置权限。收回指定权限的REVOKE语句的基本语法如下：

```
REVOKE priv_type [(column_list)]...
```

```
ON database.table
```

```
FROM user [, user] ...
```

15.3.4 查看权限

MySQL中，可以使用SELECT语句来查询user表中各个用户的权限，也可以直接使用SHOW GRANTS语句来查看权限。mysql数据库下的user表中存储着用户的基本权限，可以使用SELECT语句来查看。SELECT语句的代码如下：

```
SELECT * FROM mysql.user ;
```

15.4 本章实例

本小节将创建一个名为aric的用户，初始密码设置为“abcdef”。该用户对test数据库下的所有表拥有查询、更新和删除的权限。用户创建成功后进行如下操作：

- (1) 使用root用户将其密码修改为“aaabbb”。**
- (2) 查看aric用户的权限。**
- (3) 收回aric用户的删除权限。**
- (4) 删除aric用户。**

本实例的执行步骤如下：

- 1. 创建aric用户**
- 2. 查看aric用户的权限**
- 3. 收回aric用户的删除权限**
- 4. 删除aric用户**

15.5 上机实践

上机实践的要求如下：

- (1) 将使用root用户创建exam1用户，初始密码为“123456”。让该用户对所有数据库拥有SELECT、CREATE、DROP、SUPER和GRANT权限。**
- (2) 创建用户exam2，该用户没有初始密码。**
- (3) 用exam2登录，将其密码设置为“686868”。**
- (4) 用exam1登录，为exam2设置CREATE和DROP权限。**
- (5) 用root用户登录，收回exam1和exam2的所有权限。**

。

15.6 常见问题及解答

- 1 . **mysqladmin命令不能修改普通用户的密码？**
- 2 . **新创建的MySQL用户不能在其它机器上登录MySQL数据库？**

15.7 小结

本章介绍了MySQL数据库的权限表、账户管理、权限管理的内容。其中，账户管理和权限管理是本章的重点内容。这两部分中的密码管理、授权、收回权限是重中之重，因为这些内容涉及到MySQL数据库的安全。希望读者能够认真学习这部分的内容。取回root用户的密码和授权是本章的难点。取回root用户密码的操作很复杂，需要读者按照本章的内容进行练习。授权时需要确定给用户分配什么权限，这需要根据实际情况来决定。下一章将为读者介绍数据备份与还原。

15.8 本章习题

- 1.用root用户创建一个名为my的用户，密码设置为“mysql”。为该用户设置CREATE和DROP权限。
- 2.修改题1中的my用户的密码，将密码改为“mybook”。分别练习用root用户和my用户的权限来修改。
- 3.假设忘记了root用户的密码，然后为root用户设置新的密码。

第16章 数据备份与还原

为了保证数据的安全，需要定期对数据进行备份。备份的方式有很多种，效果也不一样。如果数据库中的数据出现了错误，就需要使用备份好的数据进行数据还原。这样可以将损失降低到最小。而且，可能还会涉及到数据库之间的数据导入与导出。在这一章中将讲解的内容包括。

- 数据备份
- 数据还原
- 数据库迁移
- 导出和导入文本文件

16.1 数据备份

备份数据是数据库管理中最常用的操作。为了保证数据库中数据的安全，数据库管理员需要定期的进行数据库备份。一旦数据库遭到破坏，可以通过备份的文件来还原数据库。因此，数据备份是很重要的工作。本节将为读者介绍数据备份的方法。

16.1.1 使用mysqldump命令备份

mysqldump命令可以将数据库中的数据备份成一个文本文件。表的结构和表中的数据将存储在生成的文本文件中。本小节将为读者介绍mysqldump命令的工作原理和使用方法。

mysqldump命令的工作原理很简单。它先查出需要备份的表的结构，再在文本文件中生成一个CREATE语句。然后，将表中的所有记录转换成一条INSERT语句。这些CREATE语句和INSERT语句都是还原时使用的。还原数据时就可以使用其中的CREATE语句来创建表。使用其中的INSERT语句来还原数据。

- 1. 备份一个数据库**
- 2. 备份多个数据库**
- 3. 备份所有数据库**

16.1.2 直接复制整个数据库目录

MySQL有一种最简单的备份办法，就是将MySQL中的数据库文件直接复制出来。这种方法最简单，速度也最快。使用这种方法时，最好将服务器先停止。这样，可以保证在复制期间数据库中的数据不会发生变化。如果在复制数据库的过程中还有数据写入，就会造成数据不一致。

这种方法虽然简单快速，但不是最好的备份方法。因为，实际情况可能不允许停止MySQL服务器。而且，这种方法对InnoDB存储引擎的表不适用。对于MyISAM存储引擎的表，这样备份和还原很方便。但是还原时最好是相同版本的MySQL数据库，否则可能会存在文件类型不同的情况。

16.1.3 使用mysqlhotcopy工具快速备份

如果备份时不能停止MySQL服务器，可以采用mysqlhotcopy工具。mysqlhotcopy工具的备份方式比mysqldump命令快。下面为读者介绍mysqlhotcopy工具的工作原理和使用方法。

mysqlhotcopy工具是一个Perl脚本，主要在Linux操作系统下使用。mysqlhotcopy工具使用LOCK TABLES、FLUSH TABLES和cp来进行快速备份。其工作原理是：先将需要备份的数据库加上一个读操作锁，然后用FLUSH TABLES将内存中的数据写回到硬盘上的数据库中，最后把需要备份的数据库文件复制到目标目录。使用mysqlhotcopy的命令如下：

```
[root@localhost ~]# mysqlhotcopy [option] dbname1  
dbname2 ... backupDir/
```

16.2 数据还原

管理员的非法操作和计算机的故障都会破坏数据库文件。当数据库遭到这些意外时，可以通过备份文件将数据库还原到备份时的状态。这样可以将损失降低到最小。本节将为读者介绍数据还原的方法。

16.2.1 使用mysql命令还原

管理员通常使用mysqldump命令将数据库中的数据备份成一个文本文件。通常这个文件的后缀名是sql。需要还原时，可以使用mysql命令来还原备份的数据。本小节将为读者介绍mysql命令的工作原理和使用方法。

备份文件中通常包含CREATE语句和INSERT语句。mysql命令可以执行备份文件中的CREATE语句和INSERT语句。通过CREATE语句来创建数据库和表。通过INSERT语句来插入备份的数据。mysql命令的基本语法如下：

```
mysql -u root -p [dbname] < backup.sql
```

16.2.2 直接复制到数据库目录

之前介绍过一种直接复制数据的备份方法。通过这种方式备份的数据，可以直接复制到MySQL的数据库目录下。通过这种方式还原时，必须保证两个MySQL数据库的主版本号是相同的。因为只有MySQL数据库主版本号相同时，才能保证这两个MySQL数据库的文件类型是相同。而且，这种方式对MyISAM类型的表比较有效。对于InnoDB类型的表则不可用。因为InnoDB表的表空间不能直接复制。

在Windows操作系统下，MySQL的数据库目录通常存放下面三个路径的其中之一。分别是C:\mysql\data、C:\Documents and Settings\All Users\Application Data\MySQL\MySQL Server 5.1\data或者C:\Program Files\MySQL\MySQL Server 5.1\data。在Linux操作系统下，数据库目录通常在/var/lib/mysql/、/usr/local/mysql/data或者/usr/local/mysql/var这三个目录下。上述位置只是数据库目录最常用的位置。具体位置根据读者安装时设置的位置而定。

16.3 数据库迁移

数据库迁移就是指将数据库从一个系统移动到另一个系统上。数据库迁移的原因是多种多样的。可能是因为升级了计算机，或者是部署开发的管理系统，或者升级了MySQL数据库。甚至是换用其他的数据库。根据上述情况，可以将数据迁移大致分为三类。这三类分别是在相同版本的MySQL数据库之间迁移、迁移到其他版本的MySQL数据库中和迁移到其他类型的数据库中。本节将为读者介绍数据库迁移的方法。

16.3.1 相同版本的MySQL数据库之间的迁移

相同版本的MySQL数据库之间的迁移就是在主版本号相同的MySQL数据库之间进行数据库移动。这种迁移的方式最容易实现。本小节将为读者介绍这方面的内容。

相同版本的MySQL数据库之间进行数据库迁移的原因很多。通常的原因是换了新的机器，或者是装了新的操作系统。还有一种常见的原因就是将开发的管理系统部署到工作机器上。因为迁移前后MySQL数据库的主版本号相同，所以可以通过复制数据库目录来实现数据库迁移。但是，只有数据库表都是MyISAM类型的才能使用这种方式。

16.3.2 不同版本的MySQL数据库之间的迁移

不同版本的MySQL数据库之间进行数据迁移通常是MySQL升级的原因。例如，原来很多服务器使用4.0版本的MySQL数据库。5.0的版本推出以后，改进了4.0版本的很多缺陷。因此需要将MySQL数据库升级到5.0版本。这样就需要进行不同版本的MySQL数据库之间进行数据迁移。

高版本的MySQL数据库通常都会兼容低版本，因此可以从低版本的MySQL数据库迁移到高版本的MySQL数据库。对于MySIAM类型的表可以直接复制，也可以使用mysqlhotcopy工具。但是InnoDB类型的表不可以使用这两种方法。最常用的办法是使用mysqldump命令来进行备份，然后通过mysql命令将备份文件还原到目标MySQL数据库中。但是，高版本的MySQL数据库很难迁移到低版本的MySQL数据库。因为高版本的MySQL数据库可能有一些新的特性，这些新特性是低版本MySQL数据库所不具有的。数据库迁移时要特别小心，最好使用mysqldump命令来进行备份，避免迁移时造成数据丢失。

16.3.3 不同数据库之间迁移

不同数据库之间迁移是指从其他类型的数据库迁移到MySQL数据库，或者从MySQL数据库迁移到其他类型的数据库。例如，某个网站原来使用Oracle数据库。因为运营成本太高等诸多原因，希望改用MySQL数据库。或者，某个管理系统原来使用MySQL数据库，因为某种特殊性能的要求，希望改用Oracle数据库。这样的不同数据库之间的迁移也经常会发生。但是这种迁移没有普通适用的解决办法。

MySQL以外的数据库也有类似mysqldump这样的备份工具，可以将数据库中的文件备份成sql文件或普通文本。但是，因为不同数据库厂商没有完全按照SQL标准来设计数据库。这就造成了不同数据库使用的SQL语句的差异。例如，微软的SQL Server软件使用的是T-SQL语言。T-SQL中包含了非标准的SQL语句。这就造成了SQL Server和MySQL的SQL语句不能兼容。

16.4 表的导出和导入

MySQL数据库中的表可以导出成文本文件、XML文件或者HTML文件。相应的文本文件也可以导入MySQL数据库中。在数据库的日常维护中，经常需要进行表的导出和导入的操作。本小节将为读者介绍导出和导入文本文件的方法。

。

16.4.1 用SELECT...INTO OUTFILE导出文本文件

MySQL中，可以使用SELECT...INTO OUTFILE语句将表的内容导出成一个文本文件。其基本语法形式如下：

```
SELECT [列名] FROM table [WHERE语句]  
INTO OUTFILE '目标文件' [OPTION];
```

16.4.2 用mysqldump命令导出文本文件

mysqldump命令可以备份数据库中的数据。但是，备份时是在备份文件中保存了CREATE语句和INSERT语句。不仅如此，mysqldump命令还可以导出文本文件。其基本的语法形式如下：

```
mysqldump -u root -pPassword -T目标目录 dbname  
table [option] ;
```

16.4.3 用mysql命令导出文本文件

mysql命令可以用来登录MySQL服务器，也可以用来还原备份文件。同时，mysql命令也可以导出文本文件。其基本语法形式如下：

```
mysql -u root -pPassword -e "SELECT语句" dbname>  
C:/name.txt ;
```

其中，“Password”表示root用户的密码；使用-e选项就可以执行SQL语句；“SELECT语句”用来查询记录；“C:/name.txt”表示导出文件的路径。

16.4.4 用LOAD DATA INFILE方式导入文本文件

MySQL中，可以使用LOAD DATA INFILE命令将文本文件导入到MySQL数据库中。其基本语法形式如下：

```
LOAD DATA [LOCAL] INFILE file INTO TABLE  
table [OPTION] ;
```

16.4.5 用mysqlimport命令导入文本文件

MySQL中，可以使用mysqlimport命令将文本文件导入到MySQL数据库中。其基本语法形式如下：

```
mysqlimport -u root -pPassword [--LOCAL] dbname  
file [OPTION]
```

16.5 本章实例

本小节将对test数据库中的score表进行备份和还原的操作。score表的结构和记录见10.9小节。本小节要求的操作如下：

(1) 使用mysqldump命令来备份score表。备份文件存储在D:\backup路径下。

(2) 使用mysql命令来还原score表。

(3) 使用SELECT...INTO OUTFILE来导出score表中的记录。记录存储到D:/backup/score.txt中。

(4) 使用mysqldump命令将score表的记录导出到XML文件中。这个XML文件存储在D:\backup中。

16.6 上机实践

上机实践的要求如下：

(1) 根据10.10小节的内容在test数据库中创建department表。使用mysqldump命令将department表的记录导出到C:\目录下。

(2) 删除department表中的所有记录。然后使用LOAD DATA INFILE语句将department.txt中的记录加载到department表中。

(3) 重新删除department表的所有记录，然后使用mysqlimport命令将department.txt中的记录加载到department表中。

(4) 使用mysql命令将department.txt中的记录导出成HTML文件。

16.7 常见问题及解答

1. 如何选择备份数据库的方法？
2. 如何升级MySQL数据库？

16.8 小结

本章介绍了备份数据库、还原数据库、数据库迁移、导出表和导入表的内容。备份数据库和还原数据库是本章的重点内容。在实际应用中，通常使用mysqldump命令备份数据库，使用mysql命令还原数据库。数据库迁移、导出表和导入表是本章的难点。数据库迁移需要考虑数据库的兼容性问题，最好是在相同版本的MySQL数据库之间迁移。导出表和导入表的方法比较多，希望读者能够多练习这些方法的使用。下一章将为读者介绍各种MySQL日志的作用和使用。

16.9 本章习题

- 1.练习mysqldump命令来备份MySQL数据库中的文件，然后用mysql命令来还原备份文件。
- 2.将MySQL 5.1数据库中的文件迁移到MySQL 5.0数据库中。
- 3.练习将数据库中的表导出成XML文件和HTML文件。
。
- 4.练习将数据库中的某个表导出成txt文件，然后删除表中的所有内容。然后将txt文件导入表中。

第17章 MySQL日志

MySQL日志是记录MySQL数据库的日常操作和错误信息的文件。MySQL中，日志可以分为二进制日志、错误日志、通用查询日志和慢查询日志。分析这些日志文件，可以了解MySQL数据库的运行情况、日常操作、错误信息和哪些地方需要进行优化。在这一章中将讲解的内容包括。

- **日志定义、作用和优缺点**
- **二进制日志**
- **错误日志**
- **通用查询日志**
- **慢查询日志**
- **日志管理**

17.1 日志简介

日志是MySQL数据库的重要组成部分。日志文件中记录着MySQL数据库运行期间发生的变化。当数据库遭到意外的损害时，可以通过日志文件来查询出错原因，并且可以通过日志文件进行数据恢复。本节将为读者介绍MySQL日志的含义、作用和优缺点。

MySQL日志可以分为四种，分别是二进制日志、错误日志、通用查询日志和慢查询日志。下面分别简单的介绍这四种日志文件的作用：

- **二进制日志：**
- **错误日志：**
- **通用查询日志：**
- **慢查询日志：**

17.2 二进制日志

二进制日志也叫作变更日志（update log），主要用于记录数据库的变化情况。通过二进制日志可以查询MySQL数据库中进行了哪些改变。本小节将为读者介绍二进制日志的内容。

17.2.1 启动和设置二进制日志

默认情况下，二进制日志功能是关闭的。通过my.cnf或者my.ini文件的log-bin选项可以开启二进制日志。将log-bin选项加入到my.cnf或者my.ini文件的[mysqld]组中，形式如下：

my.cnf (Linux操作系统下) 或者my.ini (Windows操作系统下)

```
[mysqld]
```

```
log-bin [=DIR \ [filename] ]
```

17.2.2 查看二进制日志

使用二进制格式可以存储更多的信息，并且可以使写入二进制日志的效率更高。但是，不能直接打开并查看二进制日志。如果需要查看二进制日志，必须使用mysqlbinlog命令。mysqlbinlog命令的语法形式如下：

```
mysqlbinlog filename.number
```

17.2.3 删除二进制日志

二进制日记会记录大量的信息。如果很长时间不清理二进制日志，将会浪费很多的磁盘空间。删除二进制日志的方法很多，本小节将为读者详细介绍如何删除二进制日志。

- 1. 删除所有二进制日志**
- 2. 根据编号来删除二进制日志**
- 3. 根据创建时间来删除二进制日志**

17.2.4 使用二进制日志还原数据库

二进制日志记录了用户对数据库中数据的改变。如INSERT语句、UPDATE语句、CREATE语句等都会记录到二进制日志中。一旦数据库遭到破坏，可以使用二进制日志来还原数据库。本小节将为读者详细介绍使用二进制日志还原数据库的方法。

如果数据库遭到意外损坏，首先应该使用最近的备份文件来还原数据库。备份之后，数据库可能进行了一些更新。这可以使用二进制日志来还原。因为二进制日志中存储了更新数据库的语句，如UPDATE语句、INSERT语句等。二进制日志还原数据库的命令如下：

```
mysqlbinlog filename.number | mysql -u root -p
```

17.2.5 暂时停止二进制日志功能

在配置文件中设置了log-bin选项以后，MySQL服务器将会一直开启二进制日志功能。删除该选项后就可以停止二进制日志功能。如果需要再次启动这个功能，又需要重新添加log-bin选项。MySQL中提供了暂时停止二进制日志功能的语句。本小节将为读者介绍暂时停止二进制日志功能的方法。

如果用户不希望自己执行的某些SQL语句记录在二进制日志中，那么需要在执行这些SQL语句之前暂停二进制日志功能。用户可以使用SET语句来暂停二进制日志功能，SET语句的代码如下：

```
SET SQL_LOG_BIN=0 ;
```

17.3 错误日志

错误日志是MySQL数据库中最常用的一种日志。错误日志主要用来记录MySQL服务的开启、关闭和错误信息。本小节将为读者介绍错误日志的内容。

17.3.1 启动和设置错误日志

在MySQL数据库中，错误日志功能是默认开启的。而且，错误日志无法被禁止。默认情况下，错误日志存储在MySQL数据库的数据文件夹下。错误日志文件通常的名称为hostname.err。其中，hostname表示MySQL服务器的主机名。错误日志的存储位置可以通过log-error选项来设置。将log-error选项加入到my.ini或者my.cnf文件的[mysqld]组中，形式如下：

```
# my.cnf ( Linux操作系统下 ) 或者my.ini ( Windows操作系统下 )
```

```
[mysqld]
```

```
log-error=DIR / [filename]
```

17.3.2 查看错误日志

错误日志中记录着开启和关闭MySQL服务的时间，以及服务运行过程中出现哪些异常等信息。如果MySQL服务出现异常，可以到错误日志中查找原因。本小节将为读者介绍查看错误日志的方法。

错误日志是以文本文件的形式存储的，可以直接使用普通文本工具就可以查看。Windows操作系统可以使用文本文件查看器查看。Linux操作系统下，可以使用vi工具或者使用gedit工具来查看。

17.3.3 删除错误日志

数据库管理员可以删除很长时间之前的错误日志，以保证MySQL服务器上的硬盘空间。MySQL数据库中，可以使用mysqladmin命令来开启新的错误日志。mysqladmin命令的语法如下：

```
mysqladmin -u root -p flush-logs
```

执行该命令后，数据库系统会自动创建一个新的错误日志。旧的错误日志仍然保留着，只是已经更名为filename.err-old。

17.4 通用查询日志

通用查询日志用来记录用户的所有操作，包括启动和关闭MySQL服务、更新语句、查询语句等。本节将为读者介绍通用查询日志的内容。

17.4.1 启动和设置通用查询日志

默认情况下，通用查询日志功能是关闭的。通过my.cnf或者my.ini文件的log选项可以开启通用查询日志。将log选项加入到my.cnf或者my.ini文件的[mysqld]组中，形式如下：

my.cnf (Linux操作系统下) 或者my.ini (Windows操作系统下)

```
[mysqld]
```

```
log [=DIR \ [filename] ]
```

17.4.2 查看通用查询日志

用户的所有操作都会记录到通用查询日志中。如果希望了解某个用户最近的操作，可以查看通用查询日志。通用查询日志是以文本文件的形式存储的。Windows操作系统可以使用文本文件查看器查看。Linux操作系统下，可以使用vi工具或者使用gedit工具来查看。

17.4.3 删除通用查询日志

通用查询日志会记录用户的所有操作。如果数据库的使用非常频繁，那么通用查询日志将会占用非常大的磁盘空间。数据库管理员可以删除很长时间之前的通用查询日志，以保证MySQL服务器上的硬盘空间。本小节将介绍删除通用查询日志的方法。

MySQL数据库中，也可以使用mysqladmin命令来开启新的通用查询日志。新的通用查询日志会直接覆盖旧的查询日志，不需要再手动删除了。mysqladmin命令的语法如下

:

```
mysqladmin -u root -p flush-logs
```

17.5 慢查询日志

慢查询日志用来记录执行时间超过指定时间的查询语句。通过慢查询日志，可以查找出哪些查询语句的执行效率很低，以便进行优化。本节将为读者介绍慢查询日志的内容。

17.5.1 启动和设置慢查询日志

默认情况下，慢查询日志功能是关闭的。通过my.cnf或者my.ini文件的log-slow-queries选项可以开启慢查询日志。通过long_query_time选项来设置时间值，时间以秒为单位。如果查询时间超过了这个时间值，这个查询语句将被记录到慢查询日志。将log-slow-queries选项和long_query_time选项加入到my.cnf或者my.ini文件的[mysqld]组中，形式如下：

my.cnf (Linux操作系统下) 或者my.ini (Windows操作系统下)

```
[mysqld]
```

```
log-slow-queries [=DIR \ [filename] ]
```

```
long_query_time=n
```

17.5.2 查看慢查询日志

执行时间超过指定时间的查询语句会被记录到慢查询日志中。如果用户希望查询哪些查询语句的执行效率低，可以从慢查询日志中获得想要的信息。慢查询日志也是以文本文件的形式存储的。可以使用普通的文本文件查看工具来查看。

17.5.3 删除慢查询日志

慢查询日志的删除方法与通用查询日志的删除方法是一样的。可以使用mysqladmin命令来删除。也可以使用手工方式来删除。mysqladmin命令的语法如下：

```
mysqladmin -u root -p flush-logs
```

执行该命令后，命令行会提示输入密码。输入正确密码后，将执行删除操作。新的慢查询日志会直接覆盖旧的查询日志，不需要再手动删除了。数据库管理员也可以手工删除慢查询日志。删除之后需要重新启动MySQL服务。重启之后就会生成新的慢查询日志。如果希望备份旧的慢查询日志文件，可以将旧的日志文件改名。然后重启MySQL服务。

17.6 本章实例

本小节将对二进制日志进行实际的操作。本小节要求的操作如下：

- (1) 启动二进制日志功能，并且将二进制日志存储到C:\目录下。二进制日志文件命名为binlog。**
- (2) 启动服务后，查看二进制日志。**
- (3) 然后向test数据库下的score表中插入两条记录。**
- (4) 暂停二进制日志功能，然后再次删除score表中的所有记录。**
- (5) 重新开启二进制日志功能。**
- (6) 使用二进制日志来恢复score表。**
- (7) 删除二进制日志。**

17.7 上机实践

上机实践的要求如下：

- (1) 将错误日志的存储位置设置为C:\LOG目录下。**
- (2) 开启通用查询日志，并设置该日志存储在C:\LOG目录下。**
- (3) 开启慢查询日志，并设置该日志存储在C:\LOG目录下。设置时间值为5秒。**
- (4) 查看错误日志、通用查询日志和慢查询日志。**
- (5) 删除错误日志。**
- (6) 删除通用查询日志和慢查询日志。**

17.8 常见问题及解答

1. 平时应该开启什么日志？
2. 如何使用二进制日志？

17.9 小结

本章介绍了日志的含义、作用和优缺点，然后介绍了二进制日志、错误日志、通用查询日志和慢查询日志的内容。本章的重点内容是二进制日志、错误日志和通用查询日志，因为这几种日志的使用频率比较高。二进制日志是本章的难点。二进制日志的查询方法与其它日志不同，需要读者特别注意。而且，二进制日志可以还原数据库。通过本章的学习，读者对MySQL日志会有深入的了解。下一章将为读者介绍MySQL数据库的性能优化。

17.10 本章习题

- 1.练习启动和设置二进制日志、查看二进制日志、暂停二进制日志功能等操作。
- 2.练习使用二进制日志的内容还原数据库。
- 3.练习使用三种方式删除二进制日志。
- 4.练习设置错误日志的存储路径、查看错误日志、删除错误日志。
- 5.练习启动和设置通用查询日志、查看通用查询日志。
- 6.练习启动和设置慢查询日志、查看慢查询日志。
- 7.练习删除通用查询日志和慢查询日志。

第18章 性能优化

性能优化是通过某些有效的方法提高MySQL数据库的性能。性能优化的目的是为了使MySQL数据库运行速度更快、占用的磁盘空间更小。性能优化包括很多方面，例如优化查询速度、优化更新速度、优化MySQL服务器等。本章将为读者介绍的内容包括：****

- **性能优化的介绍**
- **优化查询**
- **优化数据库结构**
- **优化MySQL服务器**

18.1 优化简介

优化MySQL数据库是数据库管理员的必备技能。通过不同的优化方式达到提高MySQL数据库性能的目的。本小节将为读者介绍优化的基本知识。

MySQL数据库的用户和数据非常少的时候，很难判断一个MySQL数据库的性能的好坏。只有当长时间运行，并且有大量用户进行频繁操作时，MySQL数据库的性能就会体现出来了。例如，一个每天有几万用户同时在线的大型网站的数据库性能的优劣就很明显。这么多用户在同时连接MySQL数据库，并且进行查询、插入、更新的操作。如果MySQL数据库的性能很差，很可能无法承受如此多用户同时操作。试想用户查询一条记录需要花费很长时间，用户很难会喜欢这个网站。

18.2 优化查询

查询是数据库中最频繁的操作。提高了查询速度可以有效提高MySQL数据库的性能。本小节将为读者介绍优化查询的方法。

18.2.1 分析查询语句

通过对查询语句的分析，可以了解查询语句的执行情况。MySQL中，可以使用EXPLAIN语句和DESCRIBE语句来分析查询语句。本小节将为读者介绍这两种分析查询语句的方法。

EXPLAIN语句的基本语法如下：

EXPLAIN SELECT语句；

通过EXPLAIN关键字可以分析后面的SELECT语句的执行情况。并且能够分析出所查询的表的一些内容。

18.2.2 索引对查询速度的影响

索引可以快速的定位表中的某条记录。使用索引可以提高数据库查询的速度，从而提高数据库的性能。本小节将为读者介绍索引对查询速度影响。

如果查询时不使用索引，查询语句将查询表中的所有字段。这样查询的速度会很慢。如果使用索引进行查询，查询语句只查询索引字段。这样可以减少查询的记录数，达到提高查询速度的目的。

18.2.3 使用索引查询

索引可以提高查询的速度。但是有些时候即使查询时使用的是索引，但索引并没有起作用。本小节将向读者介绍索引的使用。

- 1. 查询语句中使用LIKE关键字**
- 2. 查询语句中使用多列索引**
- 3. 查询语句中使用OR关键字**

18.2.4 优化子查询

很多查询中需要使用子查询。子查询可以使查询语句很灵活，但子查询的执行效率不高。子查询时，MySQL需要为内层查询语句的查询结果建立一个临时表。然后外层查询语句再临时表中查询记录。查询完毕后，MySQL需要撤销这些临时表。因此，子查询的速度会受到一定的影响。如果查询的数据量比较大，这种影响就会随之增大。在MySQL中可以使用连接查询来替代子查询。连接查询不需要建立临时表，其速度比子查询要快。

18.3 优化数据库结构

数据库结构是否合理，需要考虑是否存在冗余、对表的查询和更新的速度、表中字段的数据类型是否合理等多方面的内容。本节将为读者介绍优化数据库结构的方法。

18.3.1 将字段很多的表分解成多个表

有些表在设计时设置了很多的字段。这个表中有些字段的使用频率很低。当这个表的数据量很大时，查询数据的速度就会很慢。本小节将为读者介绍优化这种表的方法。

对于这种字段特别多且有些字段的使用频率很低的表，可以将其分解成多个表。

18.3.2 增加中间表

有时候需要经常查询某两个表中的几个字段。如果经常进行联表查询，会降低MySQL数据库的查询速度。对于这种情况，可以建立中间表来提高查询速度。本小节将为读者介绍增加中间表的方法。

先分析经常需要同时查询哪几个表中的哪些字段。然后将这些字段建立一个中间表，并从原来那几个表将数据插入到中间表中。之后就可以使用中间表来进行查询和统计了。

18.3.3 增加冗余字段

设计数据库表的时候尽量让表达到三范式。但是，有时候为了提高查询速度，可以有意识的在表中增加冗余字段。本小节将为读者介绍通过增加冗余字段来提高查询速度的方法。

表的规范化程度越高，表与表之间的关系就越多。查询时可能经常需要多个表之间进行连接查询。而进行连接操作会降低查询速度。例如，学生的信息存储在student表中，院系信息存储在department表中。通过student表中的dept_id字段与department表建立关联关系。如果要查询一个学生所在系的名称，必须从student表中查找学生所在院系的编号（dept_id），然后根据这个编号去department查找系的名称。如果经常需要进行这个操作时，连接查询会浪费很多的时间。因此可以在student表中增加一个冗余字段dept_name，该字段用来存储学生所在院系的名称。这样就不用每次都进行连接操作了。

18.3.4 优化插入记录的速度

插入记录时，索引、惟一性校验都会影响到插入记录的速度。而且，一次插入多条记录和多次插入记录所耗费的时间是不一样的。根据这些情况，分别进行不同的优化。本小节将为读者介绍优化插入记录的速度方法。

- 1．禁用索引**
- 2．禁用惟一性检查**
- 3．优化INSERT语句**

18.3.5 分析表、检查表和优化表

分析表主要作用是分析关键字的分布。检查表主要作用是检查表是否存在错误。优化表主要作用是消除删除或者更新造成的空间浪费。本小节将为读者介绍分析表、检查表和优化表的方法。

- 1 . 分析表**
- 2 . 检查表**
- 3 . 优化表**

18.4 优化MySQL服务器

优化MySQL服务器可以从两个方面来理解。一个是从硬件方面来进行优化。另一个是从MySQL服务的参数进行优化。通过这些优化方式，可以提供MySQL的运行速度。但是这部分的内容很难理解，一般只有专业的数据库管理员才能进行这一类的优化。本小节将为读者介绍优化MySQL服务器的方法。

18.4.1 优化服务器硬件

服务器的硬件性能直接决定着MySQL数据库的性能。例如，增加内存和提高硬盘的读写速度，这能够提高MySQL数据库的查询、更新的速度。本小节将为读者介绍优化服务器硬件的方法。

随着硬件技术的成熟，硬件的价格也随之降低。现在普通的个人电脑都已经配置了2G内存，甚至一些个人电脑配置4G内存。因为内存的读写速度比硬盘的读写速度快。可以在内存中为MySQL设置更多的缓冲区，这样可以提高MySQL访问的速度。如果将查询频率很高的记录存储在内存中，那么查询速度就会很快。

如果条件允许，可以将内存提高到4G。并且选择my-innodb-heavy-4G.ini作为MySQL数据库的配置文件。但是，这个配置文件主要支持InnoDB存储引擎的表。如果使用2G内存，可以选择my-huge.ini作为配置文件。而且，MySQL所在的计算机最好是专用数据库服务器。这样数据库可以完全利用该机器的资源。

18.4.2 优化MySQL的参数

内存中会为MySQL保留部分的缓存区。这些缓存区可以提高MySQL数据库的处理速度。缓存区的大小都是在MySQL的配置文件中设置的。本小节将为读者介绍这些配置参数。

MySQL中比较重要的配置参数都在my.cnf或者my.ini文件的[mysqld]组中。

18.5 本章实例

本小节将对MySQL进行优化操作。本小节要求的操作如下：

(1) 查看InnoDB表的查询的记录数和更新的记录数。

(2) 分析查询语句的性能，SELECT语句如下：

```
SELECT * FROM score WHERE stu_id=902 ;
```

(3) 分析score表。

本实例的执行步骤如下：

- 1. 查看InnoDB表的查询次数和更新次数**
- 2. 分析查询语句的性能**
- 3. 分析score表**

18.6 上机实践

本小节将对MySQL进行优化操作。本小节要求的操作如下：

(1) 查看MySQL服务器的连接数、查询次数和慢查询的次数。

(2) 检查score表。

(3) 优化score表。

本实例的执行步骤如下：

1. 查看MySQL服务器的连接数、查询次数和慢查询的次数

2. 检查score表

3. 优化score表

18.7 常见问题及解答

1. 如何使用查询缓存区？
2. 为什么查询语句中的索引没有发挥作用？

18.8 小结

本章介绍了数据库优化的含义和查看数据库性能参数的方法。然后，介绍了优化查询的方法、优化数据库结构的方法和优化MySQL服务器的方法。优化查询的方法和优化数据库结构是本章的重点内容，优化查询部分主要介绍了索引对查询速度的影响。优化数据库结构部分主要介绍了如何对表进行优化。本章的难点是优化MySQL服务器，因为这部分涉及很多MySQL配置文件和配置文件中的参数。下一章将为读者介绍Java语言访问MySQL数据库的方法。

18.9 本章习题

- 1.练习查看MySQL数据库的连接数、上线时间、执行更新操作的次数和执行删除操作的次数。
- 2.练习分析查询语句中是否使用了索引。
- 3.练习分析表、检查表和优化表。
- 4.练习优化MySQL的参数。

第19章 Java访问MySQL数据库

Java是由Sun公司开发的程序设计语言。Java语言是一种面向对象的编程语言，而且具有跨平台性和高效的网络编程特性。Java现在已经是流行的程序语言之一。Java语言可以通过MySQL数据库的接口访问MySQL数据库。本章将为读者介绍的内容包括：

- **Java连接MySQL数据库**
- **Java操纵MySQL数据库**
- **Java备份MySQL数据库**
- **Java还原MySQL数据库**

19.1 Java连接MySQL数据库

Java语言可以通过JDBC（Java Database Connectivity，Java数据库连接）来访问MySQL数据库。JDBC的编程接口提供的接口和类与MySQL数据库建立连接，然后将SQL语句的执行结果进行处理。但这需要一个MySQL数据库的JDBC驱动程序。本小节将为读者介绍Java连接MySQL数据库的方法。

19.1.1 下载JDBC驱动MySQL Connector/J

读者可以在MySQL的官方网站下载JDBC驱动，当前最新的JDBC驱动程序是MySQL Connector/J 5.1。MySQL Connector/J 5.1的下载网址为 <http://dev.mysql.com/downloads/connector/j/5.1.html>。在下载页面有Source and Binaries (tar.gz)和Source and Binaries (zip)两个下载选项。前者主要用于Linux操作系统，后者主要用于Windows操作系统。下载后的文件分别是mysql-connector-java-5.1.10.tar.gz和mysql-connector-java-5.1.10.zip。这里面都包含驱动的源代码和二进制包。源代码可以自行进行编译。二进制包是编译好的驱动，名称为mysql-connector-java-5.1.10-bin.jar。

19.1.2 安装MySQL Connector/J驱动

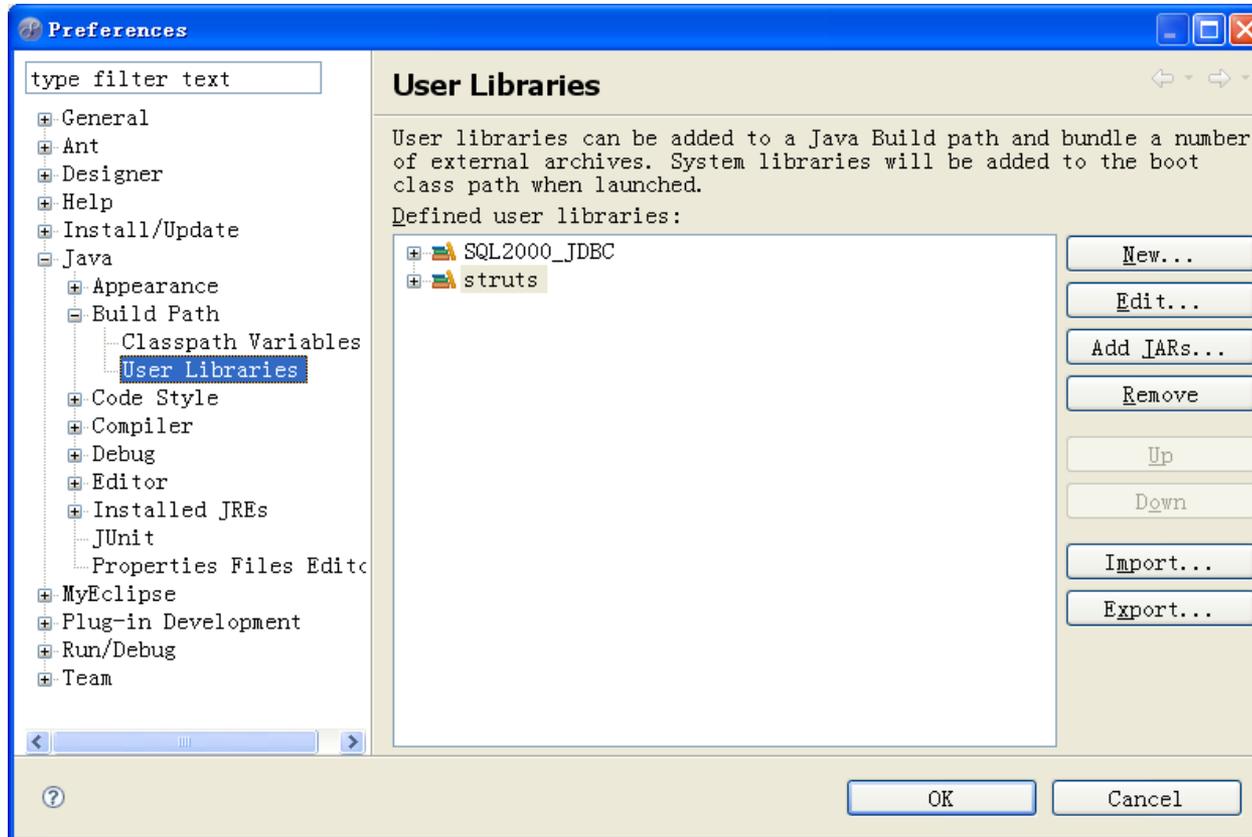
Shell或DOS窗口和Eclipse等编程工具中使用JDBC的方式是不一样的。前者需要将JDBC驱动的路径添加到环境变量中。后者可以直接将JDBC驱动添加到Eclipse等工具中。

Windows操作系统中右击【我的电脑】图标，在下拉菜单中单击【属性】命令，然后单击【高级】|【环境变量】按钮。弹出的窗口中可以看到用户环境变量。在classpath变量中添加mysql-connector-java-5.1.10-bin.jar的路径。在DOS窗口中执行的Java语句中需要调用JDBC驱动时，系统会自动到classpath变量中设置的路径中去查找。

Linux操作系统下先使用tar命令来解压mysql-connector-java-5.1.10.tar.gz，命令如下：

```
tar -xzvf mysql-connector-java-5.1.10.tar.gz
```

19.1.2 安装MySQL Connector/J驱动



19.1.3 连接MySQL数据库

在java.sql包中存在DriverManager类、Connection接口、Statement接口和ResultSet接口。这些类和接口的作用如下：

DriverManager类：

Connection接口：

Statement接口：

ResultSet接口：

19.2 Java操作MySQL数据库

连接MySQL数据库之后，可以对MySQL数据库中的数据进行查询、插入、更新、删除等操作。Statement接口主要用来执行SQL语句，其中定义一些执行SQL语句的方法。SQL语句执行后返回的结果由ResultSet接口管理。通过这两个接口，Java可以方便的操作MySQL数据库。本节将详细的向读者介绍Java操作MySQL数据库的方法。

19.2.1 创建Statement对象

Connection对象调用createStatement()方法来创建Statement对象，其代码如下：

```
Statement statement=connection.createStatement();
```

其中，statement是Statement对象；connection是Connection对象；createStatement()方法返回Statement对象。通过这个Java语句就可以创建Statement对象。Statement对象创建成功后，可以调用其中的方法来执行SQL语句。

19.2.2 使用SELECT语句查询数据

Statement对象可以调用executeQuery()方法执行SELECT语句。SELECT语句的查询结果返回给ResultSet对象。调用executeQuery()方法的代码如下：

```
ResultSet result=statement.executeQuery("SELECT语句");
```

通过该语句可以将查询结果存储到result中。查询结果可能有多条记录，这就需要使用循环语句来读取所有记录，其代码如下：

```
while(result.next()){  
String s=result.getString("字段名");  
System.out.print(s);  
}
```

19.2.3 插入、更新或者删除数据

executeQuery()方法只能执行SELECT语句。如果需要进行插入、更新或者删除操作，则需要Statement对象调用executeUpdate()方法来实现。executeUpdate()方法执行完后，返回影响表的行数。下面是调用executeQuery()方法的代码：

```
int result=statement.executeUpdate(sql);
```

其中，“sql”参数必须是INSERT语句、UPDATE语句或者DELETE语句。该方法返回的结果数字。

19.2.4 执行任意SQL语句

execute()方法可以执行SELECT语句、INSERT语句、UPDATE语句和DELETE语句等。无法确定要执行的SQL语句是查询还是更新时，可以使用execute()函数。该函数的返回结果是boolean类型的值，返回值为true表示执行查询语句，false表示执行更新语句。下面是调用execute()方法的代码：

```
boolean result=statement.execute(sql);
```

19.2.5 关闭创建的对象

当所有SQL语句都执行完毕后，需要关闭所创建的Connection对象、Statement对象和ResultSet对象。关闭对象的顺序与创建对象的顺序相反，关闭的顺序为ResultSet对象、Statement对象、Connection对象。对象调用close()方法来关闭对象，然后将对象的值设为空。

19.3 Java备份与还原MySQL数据库

Java语言中可以执行mysqldump命令来备份MySQL数据库，也可以执行mysql命令来还原MySQL数据库。本小节将为读者介绍Java备份与还原MySQL数据库的方法。

19.3.1 Java备份MySQL数据库

通常使用mysqldump命令来备份MySQL数据库，其语句如下：

```
mysqldump -u username -pPassword dbname table1  
table2 ... > BackupName.sql
```

其中，“username”参数表示登录数据库的用户名；“Password”参数表示用户的密码，其与“-p”之间不能用空格隔开；“dbname”参数表示数据库的名称；“table1”和“table2”参数表示表的名称，没有该参数时将备份整个数据库；“BackupName.sql”参数表示备份文件的名称，文件名前面可以加上一个绝对路径。

19.3.2 Java还原MySQL数据库

通常使用mysql命令来还原MySQL数据库，其语句如下

:

```
mysql -u root -p [dbname] < backup.sql
```

其中，“dbname”参数表示数据库名称。该参数是可选参数，可以指定数据库名，也可以不指定。指定数据库名时，表示还原该数据库下的表。不指定数据库名时，表示还原特定的一个数据库。而备份文件中有创建数据库的语句。

19.4 本章实例

本小节将使用Java语言访问MySQL数据库。本小节要求的操作如下：

- 1 . 编写DB.java类**
- 2 . 编写DB_backup_load.java类**

19.5 上机实践

上机实践的要求如下：

(1) 新建mysql.java类，在该类中封装一些方法。

(2) 新建useMySQL.java类，在其中调用mysql.java中的方法来连接grid数据库。

19.6 常见问题及解答

1. 出现

`java.lang.ClassNotFoundException:com.mysql.jdbc.Driver`错误？

2. 在`executeQuery()`函数中执行SELECT语句后，如何知道查询结果集的记录数？

19.7 小结

本章介绍了Java语言访问MySQL数据库的方法。使用Java语言连接MySQL数据库和操作MySQL数据库是本章的重点内容。这部分重点讲解了Java中执行SELECT语句、INSERT语句、UPDATE语句、DELETE语句的方法。而且，还重点讲解执行SELECT语句后，将查询结果逐条读取出来的方法。Java备份和还原MySQL数据库是本章的难点，因为需要Java语言调用外部命令。通过本章的学习，读者对Java访问MySQL数据库有深入的了解。下一章为读者介绍PHP访问MySQL数据库的方法。

19.8 本章习题

- 1.练习编写查询MySQL数据库中数据的Java代码。
- 2.练习编写更新MySQL数据库中数据的Java代码。
- 3.练习编写向数据库中插入新纪录、从数据库中删除记录的Java代码。
- 4.练习编写备份和还原数据库的Java代码。

第20章 PHP访问MySQL数据库

现在最流行的动态网站开发的软件组合是LAMP。

LAMP是Linux、Apache、MySQL和PHP的缩写。PHP具有简单、易用、功能强大和开放性等特点，这使PHP已经成为了网络世界上最流行的编程语言之一。PHP可以通过mysql接口或者mysqli接口来访问MySQL数据库。本章将为读者介绍的内容包括：

- **PHP连接MySQL数据库**
- **PHP操纵MySQL数据库**
- **PHP备份MySQL数据库**
- **PHP还原MySQL数据库**

20.1 PHP连接MySQL数据库

PHP可以通过mysql接口或者mysqli接口来访问MySQL数据库。如果希望正常的使用PHP，那么需要适当的配置PHP与Apache服务器。同时，PHP中加入了mysql接口和mysqli接口后，才能够顺利的访问MySQL数据库。本小节将为读者介绍PHP连接MySQL数据库的方法。

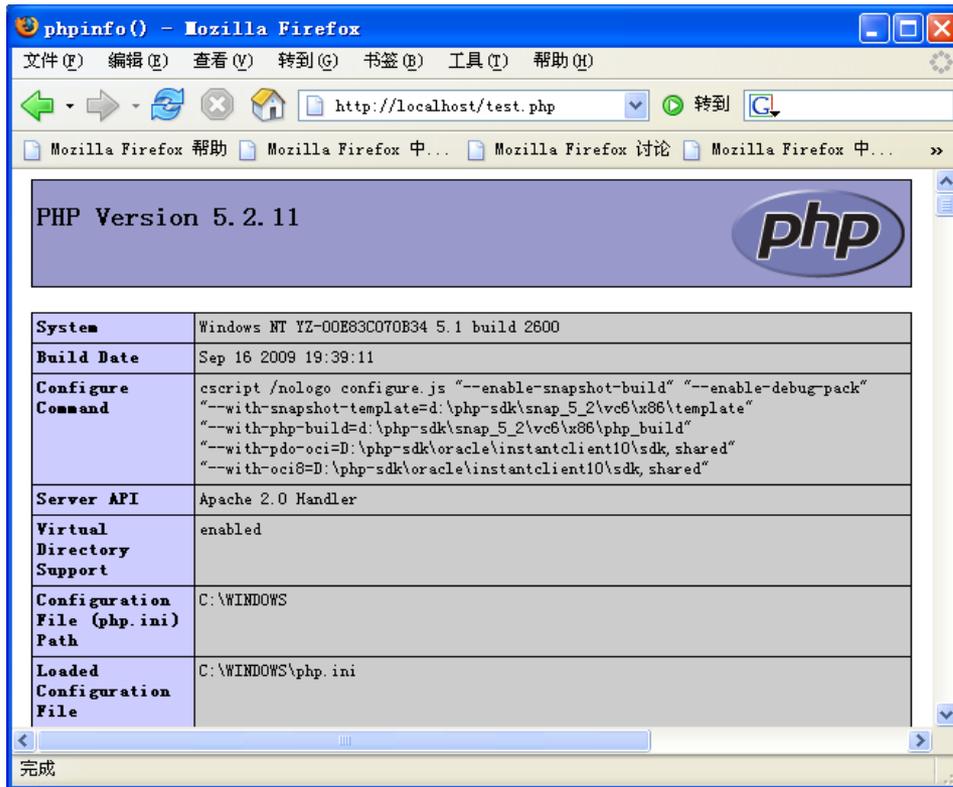
20.1.1 Windows操作系统下配置PHP

如果你还没有安装PHP，你可以在<http://www.php.net/downloads.php>中下载PHP。Windows操作系统下推荐下载PHP 5.2.11 zip package。

在Windows操作系统中，将PHP的软件包解压到C:\php目录下。需要在Apache服务器的配置文件httpd.conf中添加一些信息。Apache服务器的默认路径为C:\Program Files\Apache Software Foundation\Apache2.2。httpd.conf文件在Apache服务器目录下的conf文件夹中。在httpd.conf中加入下面的信息：

```
LoadModule php5_module  
"C:/php/php5apache2_2.dll"  
AddType application/x-httpd-php .php
```

20.1.1 Windows操作系统下配置PHP



phpinfo() - Mozilla Firefox

文件(F) 编辑(E) 查看(V) 转到(G) 书签(B) 工具(T) 帮助(H)

http://localhost/test.php

PHP Version 5.2.11

System	Windows NT YZ-00E83C070B34 5.1 build 2600
Build Date	Sep 16 2009 19:39:11
Configure Command	cscript /nologo configure.js "--enable-snapshot-build" "--enable-debug-pack" "--with-snapshot-template=d:\php-sdk\snap_5_2\vc8\x86\template" "--with-php-build=d:\php-sdk\snap_5_2\vc8\x86\php_build" "--with-pdo-oci=D:\php-sdk\oracle\instantclient10\sdk,shared" "--with-oci8=D:\php-sdk\oracle\instantclient10\sdk,shared"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\WINDOWS
Loaded Configuration File	C:\WINDOWS\php.ini

完成



phpinfo() - Mozilla Firefox

文件(F) 编辑(E) 查看(V) 转到(G) 书签(B) 工具(T) 帮助(H)

http://localhost/test.php

mysql

MySQL Support	enabled
Active Persistent Links	0
Active Links	0

完成

20.1.2 Linux操作系统下配置PHP

Linux操作系统下推荐下载PHP 5.2.11 (tar.gz)。下载网址为<http://www.php.net/downloads.php>。下载完成后将php-5.2.11.tar.gz复制到/usr/local/src目录下，然后在该目录下解压和安装php。假设新下载的php软件包存储在/home/hjh/download目录下。使用下面的语句来安装php：

```
shell> cp /home/hjh/download/php-5.2.11.tar.gz /usr/local/src/
```

```
shell> cd /usr/local/src/
```

```
shell> tar -xzvf php-5.2.11
```

```
shell> cd php-5.2.11
```

```
shell> ./configure --prefix=/usr/local/php --with-mysql=/usr/local --with-mysqli=/usr/bin/mysqli_config
```

```
shell> make
```

```
shell> make install
```

```
shell> make clean
```

20.1.3 连接MySQL数据库

PHP可以通过mysql接口来连接MySQL数据库，也可以通过mysqli接口来连接MySQL数据库。下面对这两种方法都进行简单的介绍。

mysql接口提供mysql_connect()方法来连接MySQL数据库，mysql_connect()函数的使用方法如下：

```
$connection=mysql_connect("host/IP","username","password");
```

20.2 PHP操作MySQL数据库

连接MySQL数据库之后，PHP可以通过query()函数对数据进行查询、插入、更新、删除等操作。但是query()函数一次只能执行一条SQL语句。如果需要一次执行多个SQL语句，需要使用multi_query()函数。PHP通过query()函数和multi_query()函数可以方便的操作MySQL数据库。本小节将为读者介绍PHP操作MySQL数据库的方法。

20.2.1 执行SQL语句

PHP可以通过`query()`函数来执行SQL语句。如果SQL语句是INSERT语句、UPDATE语句、DELETE语句等，语句执行成功`query()`返回`true`，否则返回`false`。并且，可以通过`affected_rows()`函数获取发生变化的记录数。

20.2.2 处理查询结果

query()函数成功的执行SELECT语句后，会返回一个mysqli_result对象\$result。SELECT语句的查询结果都存储在\$result中。mysqli接口中提供了四种方法来读取数据，这四种方法的介绍如下：

\$rs=\$result->fetch_row() :

\$rs=\$result->fetch_array() :

\$rs=\$result->fetch_assoc() :

\$rs=\$result->fetch_object() :

20.2.3 获取查询结果的字段名

通过`fetch_fields()`函数可以获取查询结果的详细信息，这个函数返回对象数组。通过这个对象数组可以获取字段名、表名等信息。例如，`$info=$result->fetch_fields()`可以产生一个对象数组`$info`。然后通过`$info[$n]->name`获取字段名，`$info[$n]->table`获取表名。

20.2.4 一次执行多个SQL语句

query()函数一次只能执行一条SQL语句，而multi_query()函数可以一次执行多个SQL语句。如果第一个SQL语句正确执行，那么multi_query()函数返回true。否则，返回false。PHP中使用store_result()函数获取multi_query()函数执行查询的记录。一次只能获取一个SQL语句的执行结果。可以使用next_result()函数来判断下一个SQL语句的结果是否存在，如果存在，next_result()函数返回true。否则，返回false。

20.2.5 处理带参数的SQL语句

PHP中可以执行带参数的SQL语句。带参数的SQL语句中可以不指定某个字段的值，而使用问号(?)代替。然后在后面的语句中指定值来替换掉问号。通过prepare()函数将带参数的SQL语句进行处理，其语句如下：

```
$stmt=$mysqli->prepare("INSERT INTO table(name1,  
name2) VALUES(?, ?)");
```

20.2.6 关闭创建的对象

对MySQL数据库的访问完成后，必须关闭创建的对象。连接MySQL数据库时创建了\$connection对象，处理SQL语句的执行结果时创建了\$result对象。操作完成后，这些对象都必须使用close()方法来关闭。其基本形式为：

```
$result->close();
```

```
$connection->close();
```

20.3 PHP备份与还原MySQL数据库

PHP语言中可以执行mysqldump命令来备份MySQL数据库，也可以执行mysql命令来还原MySQL数据库。PHP中使用system()函数或者exec()函数来调用mysqldump命令和mysql命令。本小节将为读者介绍PHP备份与还原MySQL数据库的方法。

20.3.1 PHP备份MySQL数据库

PHP可以通过system()函数或者exec()函数来调用mysqldump命令。system()函数的形式如下：

```
system("mysqldump -h hostname -u user -pPassword database [table] > dir/backup.sql");
```

exec()函数的使用方法与system()函数是一样的。这里直接将mysqldump命令当作系统命令来调用。这需要将MySQL的应用程序的路径添加到系统变量的Path变量中。添加Path变量的方法请参照2.2.3小节。如果不想把MySQL的应用程序的路径添加到Path变量中，可以使用mysqldump命令的完整路径。假设mysqldump在C:\mysql\bin\目录下，system()函数的形式如下：

```
system("C:/mysql/bin/mysqldump -h hostname -u user -pPassword database [table] > dir/backup.sql");
```

20.3.2 PHP还原MySQL数据库

PHP可以通过system()函数或者exec()函数来调用mysql命令。system()函数的形式如下：

```
system("mysql -h hostname -u user -pPassword  
database [table] < dir/backup.sql");
```

exec()函数的使用方法与system()函数是一样的。mysql命令和mysqldump命令一样，只有MySQL的应用程序的路径添加到系统变量的Path变量中，才可以直接调用mysql命令。否则，需要加上mysql命令的完整路径。假设mysql在C:\mysql\bin\目录下，system()函数的形式如下：

```
system("C:/mysql/bin/mysql -h hostname -u user -  
pPassword database [table] < dir/backup.sql");
```

20.4 本章实例

本小节将使用PHP访问MySQL数据库。本小节要求的操作如下：

(1) 通过mysqli接口连接MySQL中的test数据库。

(2) 使用multi_query()函数同时执行两个SELECT语句。

(3) 在页面上显示查询结果。第一行显示字段名，下面每一行显示一条记录。

(4) 关闭所有对象。

20.5 上机实践

上机实践的要求如下：

(1) 编写mysql.php文件，使用mysqli接口来连接MySQL的test数据库。

(2) 使用query()函数执行UPDATE语句和DELETE语句，语句如下：

UPDATE score SET grade=100 WHERE id=9;

DELETE FROM score WHERE id=10;

语句执行完后，返回更新的记录数和删除的记录数。

(3) 关闭所有对象。

20.6 常见问题及解答

- 1 . 选择mysql接口还是选择mysqli接口来访问MySQL?
?
- 2 . PHP调用mysqldump命令时出错？

20.7 小结

本章介绍了PHP访问MySQL数据库的方法。使用PHP语言连接MySQL数据库和操作MySQL数据库是本章的重点内容。这部分重点讲解了PHP使用mysqli接口连接MySQL数据库，还讲解了PHP中执行SELECT语句、INSERT语句、UPDATE语句、DELETE语句的方法。本章的难点是在PHP中一次执行多个SELECT语句。执行多个SELECT语句需要使用multi_query()函数。PHP备份和还原MySQL数据库也是一大难点，希望读者能够认真学习PHP调用外部命令的方法。下一章为读者介绍C#访问MySQL数据库的方法。

20.8 本章习题

- 1.练习编写查询MySQL数据库中数据的PHP代码。
- 2.练习编写更新MySQL数据库中数据的PHP代码。
- 3.练习编写向数据库中插入新纪录、从数据库中删除记录的PHP代码。
- 4.练习编写备份和还原数据库的PHP代码。

第21章 C#访问MySQL数据库

C#是由微软公司开发的程序设计语言。C#是一种面向对象的、运行于.NET Framework之上的高级程序设计语言。因为，C#语句简单、易用，而且与Windows操作系统有良好的兼容性。所以，C#是Windows操作系统下最流行的程序语言之一。C#语言可以通过MySQL数据库的接口访问MySQL数据库。本章将为读者介绍的内容包括：

- **C#连接MySQL数据库**
- **C#操纵MySQL数据库**
- **C#备份MySQL数据库**
- **C#还原MySQL数据库**

21.1 C#连接MySQL数据库

C#语言可以通过Connector/ODBC和ODBC来访问MySQL数据库，也可以通过Connector/Net来访问MySQL数据库。因为，Connector/Net是执行效率高，而且是MySQL官方推荐的使用的驱动程序。所以，本章主要使用Connector/Net来访问MySQL数据库。本小节将为读者介绍C#连接MySQL数据库的方法。

21.1.1 下载Connector/Net驱动程序

使用C#语言来连接MySQL时，需要安装Connector/Net驱动程序。Connector/Net是MySQL官方网站提供的专业驱动程序。在网址<http://dev.mysql.com/downloads/connector/>中选择Connector/Net链接，就可以跳转到下载Connector/Net的页面。当前的最新版本是Connector/Net 6.1。进入下载页面后，可以看到三种版本。这三种版本介绍如下：

Sources (ZIP) :

Windows Binaries (zipped MSI installer) :

Windows Binaries, no installer (ZIP) :

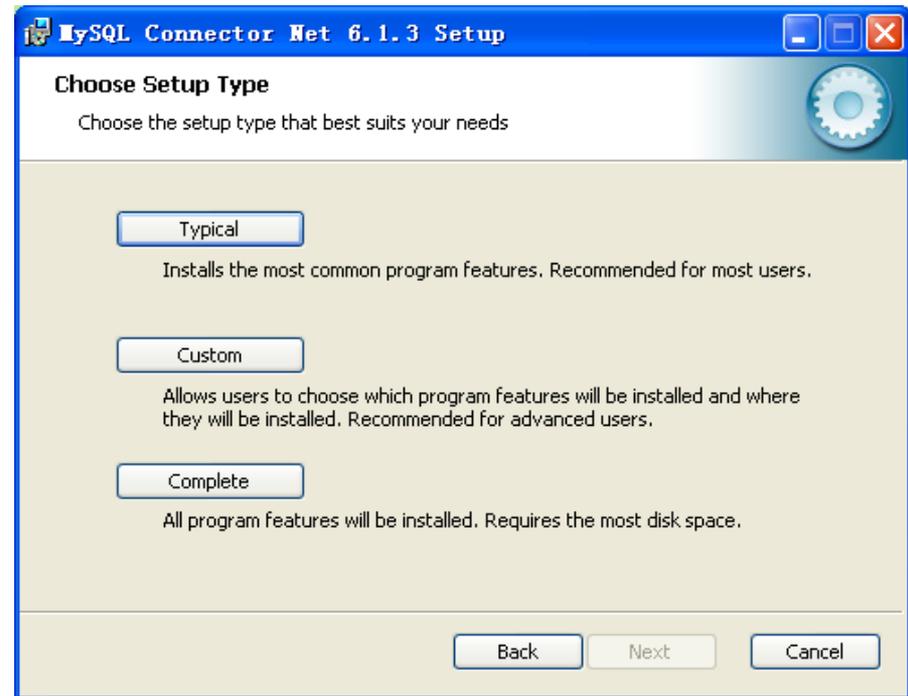
21.1.2 安装Connector/Net驱动程序

Connector/Net驱动程序下载完成后，可以在下载路径看到压缩文件mysql-connector-net-6.1.3.zip。将这个压缩文件解压后，出现安装文件mysql.data.msi。mysql.data.msi的安装过程如下：

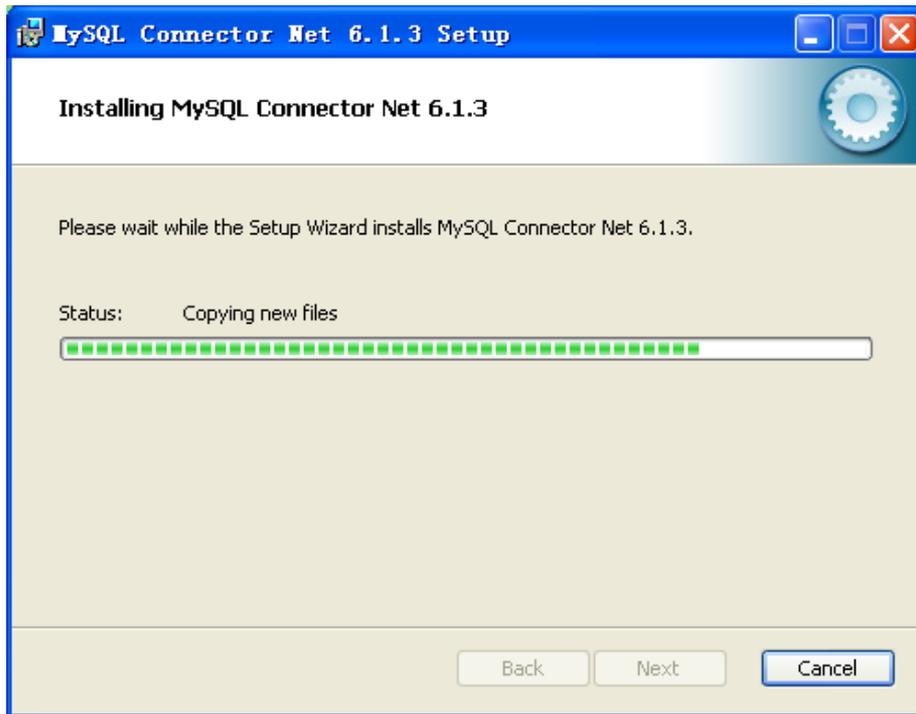
(1) 双击mysql.data.msi，会出现Connector/Net的安装欢迎界面。

(2) 单击Next按钮进入选择安装类型的界面。

21.1.2 安装Connector/Net驱动程序



21.1.2 安装Connector/Net驱动程序



21.1.3 使用Connector/Net驱动程序

推荐使用Microsoft Visual Studio编辑C#程序，因为这个集成开发环境功能非常强大。本章介绍的内容都是基于Microsoft Visual Studio的。

如果需要在项目中使用Connector/Net驱动程序，那么必须将Connector/Net驱动程序引用到项目中。在Microsoft Visual Studio中单击Project（项目）|Add Reference（添加引用）选项。



21.1.4 连接MySQL数据库

使用Connector/Net驱动程序时，通过MySQLConnection对象来连接MySQL数据库。连接MySQL的程序的最前面需要引用MySql.Data.MySqlClient，引用的语句如下：

```
using MySql.Data.MySqlClient;
```

连接MySQL数据库时，需要提供主机名或者IP地址、连接的数据库名、数据库用户名和用户密码等信息。每个信息之间用分号（；）隔开。下面是创建MySQLConnection对象的语句：

```
MySqlConnection conn = null;
```

```
conn = new MySqlConnection("Data Source=主机名或  
者IP地址;Initial Catalog=数据库名;User ID=用户名;  
Password=用户密码");
```

21.2 C#操作MySQL数据库

连接MySQL数据库之后，通过 MySqlCommand 对象来获取 SQL 语句。然后，通过 ExecuteNonQuery() 方法对数据库进行插入、更新和删除等操作。通过 ExecuteRead() 方法查询数据库中的数据，也可以通过 ExecuteScalar() 方法查询数据。通过 MySqlDataReader 对象获取 SELECT 语句的查询结果。除了上述方法操作数据库以外，还可以使用 MySqlDataAdapter 对象、DataSet 对象、DataTable 对象来操作 MySQL 数据库。本小节将详细的向读者介绍 C# 操作 MySQL 数据库的方法。

21.2.1 创建 MySqlCommand 对象

MySqlCommand 对象主要用来管理 MySqlConnection 对象和 SQL 语句。 MySqlCommand 对象的创建方法如下：

```
MySqlCommand com = new MySqlCommand("SQL语句", conn);
```

其中，“SQL语句”可以是INSERT语句、UPDATE语句、DELETE语句、SELECT语句等；“conn”为 MySqlConnection 对象。C#中也可以使用下面的方式来创建 MySqlCommand 对象，语句如下：

```
MySqlCommand com = new MySqlCommand();  
com.Connection = conn;  
com.CommandText = "SQL语句";
```

21.2.2 插入、更新或者删除数据

如果需要对MySQL数据库执行插入、更新和删除等操作，那么需要MySqlCommand对象调用ExecuteNonQuery()方法来实现。ExecuteNonQuery()方法返回一个整型的数字。下面是调用ExecuteNonQuery()方法的代码：

```
int i = com.ExecuteNonQuery();
```

21.2.3 使用**SELECT**语句查询数据

ExecuteNonQuery()方法不能执行SELECT语句。如果需要执行SELECT语句，则需要 MySqlCommand 对象调用 ExecuteReader()方法来实现。ExecuteReader()方法返回一个 MySqlDataReader 对象。调用 ExecuteReader()方法的代码如下：

```
MySqlDataReader dr;  
dr = com.ExecuteReader();
```

21.2.4 一次执行多个SELECT语句

一次执行多个SELECT语句时，ExecuteReader()方法会将所有SELECT语句的执行结果都返回给MySqlDataReader对象。但是MySqlDataReader对象一次只能读取一个SELECT语句的查询结果。如果需要读取下一个SELECT语句的执行结果，MySqlDataReader对象需要调用NextResult()方法。NextResult()方法获取到下一个SELECT语句的查询结果时，该方法返回TRUE。然后MySqlDataReader对象可以通过Read()函数来读取数据。如果NextResult()返回值为FALSE，那说明所有结果已经全都读取出来了。

21.2.5 处理SELECT语句只返回一个值的情况

如果SELECT语句只返回一个值，可以使用ExecuteScalar()方法来执行SELECT语句。因为，ExecuteScalar()方法所使用的资源比ExecuteReader()方法少

。

C#可以直接通过类型转换的方法将ExecuteScalar()方法返回的object对象转换为需要的类型。

21.2.6 处理带参数的SQL语句

C#中可以执行带参数的SQL语句。带参数的SQL语句中可以不指定某个字段的值，而使用问号(?)和变量名代替，例如“?var”就是一个参数。可以通过Add()方法为参数赋值，其代码的基本形式如下：

```
com.CommandText="INSERT INTO table(name1,  
name2) VALUES(?name1, ?name2)";  
com.Parameters.Add("?name1", value1);  
com.Parameters.Add("?name2", value2);  
com.ExecuteNonQuery();
```

21.2.7 使用DataSet对象和DataTable对象

DataSet对象和DataTable对象可以更方便的读取查询结果。而且，可以其中插入、更新、删除数据。一个DataSet对象可以管理一个或者多个DataTable对象。DataSet和DataTable是ADO .NET的类，它们属于命名空间System.Data。因此需要使用using语句引入System.Data，其语句如下：

```
using System.Data;
```

21.2.8 关闭创建的对象

如果不关闭MySQLConnection对象和MySqlDataReader对象，这些对象会一直占用系统资源。如果不需要使用这些对象时，必须将这些对象关闭。这些对象可以调用Close()方法来关闭对象。关闭MySQLConnection对象和MySqlDataReader对象的语句如下：

```
conn.Close();
```

```
dr.Close();
```

关闭MySQLConnection对象和MySqlDataReader对象后，它们所占用的内存资源和其它资源就被释放掉了。

21.3 C#备份与还原MySQL数据库

C#语言中可以执行mysqldump命令来备份MySQL数据库，也可以执行mysql命令来还原MySQL数据库。本小节将为读者介绍C#备份与还原MySQL数据库的方法。

21.3.1 C#备份MySQL数据库

C#中的Process类的Start()方法可以调用外部命令。因此，C#中可以通过调用mysqldump命令来备份MySQL数据库。mysqldump命令通常需要在DOS窗口中执行，所以需要使用Start()方法调用cmd命令来打开DOS窗口。Process类的命名空间为System.Diagnostics，因此需要使用using语句来引用这个命名空间，语句如下：

```
using System.Diagnostics;
```

21.3.2 C#还原MySQL数据库

C#使用Process类的Start()方法调用cmd.exe程序，通过cmd.exe程序打开DOS窗口。然后在DOS窗口中执行mysql命令来还原MySQL数据库。还原MySQL数据库的语句如下：

```
ProcessStartInfo psi = new ProcessStartInfo();  
psi.FileName = "cmd.exe";  
psi.Arguments = "/c mysql -h hostname -u user -  
pPassword database [table] < dir/BackupName.sql";  
Process.Start(psi);
```

21.4 本章实例

本小节将新建“ASP.NET网站”，在程序中完成下面的操作：

(1) 连接MySQL数据库。然后操作shool数据库下的student表和score表。

(2) 向student表中插入两条记录，并且在页面显示插入的记录数。

(3) 从student表和score表中查询出所有记录，并且显示在页面上。

(4) 从student表中删除计算机系学生的信息。

(5) 关闭打开的所有数据库对象。

21.5 上机实践

本小节将新建“控制台应用程序”，在程序完成下面的操作：

(1) 连接MySQL数据库。然后操作test数据库下的score表和student表。

(2) 向score表中插入两条记录，并且在控制台输出插入的记录数。

(3) 在score表中更新id值为31的记录。

(4) 从score表中删除id值为32的记录。

(5) 查询score表的所有记录。

(6) 统计score表中英语的平均成绩。

(7) 关闭打开的所有数据库对象。

21.6 常见问题及解答

1. 一次性执行多条SELECT语句时，只显示了第一个SELECT语句的查询结果？
2. 出现 “There is already an open DataReader”这样的错误？

21.7 小结

本章介绍了C#语言访问MySQL数据库的方法。使用C#语言连接MySQL数据库和操作MySQL数据库是本章的重点内容。这部分重点讲解了C#语言中执行SELECT语句、INSERT语句、UPDATE语句、DELETE语句的方法。而且，还重点讲解执行SELECT语句后，将查询结果逐条读取出来的方法。C#备份和还原MySQL数据库是本章的难点。C#需要使用Process类的Start()方法调用外部命令。通过本章的学习，希望读者对C#语言操作MySQL数据库有深入的了解。

。

21.8 本章习题

- 1.练习编写插入、更新MySQL数据库中数据的C#代码。
- 2.练习编写使用一个SELECT语句查询MySQL数据库的C#代码。
- 3.练习编写使用多个SELECT语句查询MySQL数据库的C#代码。
- 4.练习编写备份和还原数据库的C#代码。

第22章 驾校学员管理系统

MySQL数据库的使用非常广泛，很多的网站和管理系统都使用MySQL数据库存储数据。本章将向读者介绍驾校学员管理系统的开发过程。该管理系统使用Java语言开发，数据库使用MySQL数据库，Web服务器使用Tomcat。本章将为读者介绍的内容包括：

- **系统概述**
- **系统功能**
- **数据库设计**
- **系统实现**

22.1 系统概述

由于计算机技术的飞速发展，数据库技术作为数据管理的一个有效的手段，在各行各业中得到越来越广泛的应用。驾校学员管理系统主要用于管理驾校的各种数据。本小节将介绍本驾校学员管理系统的基本信息。

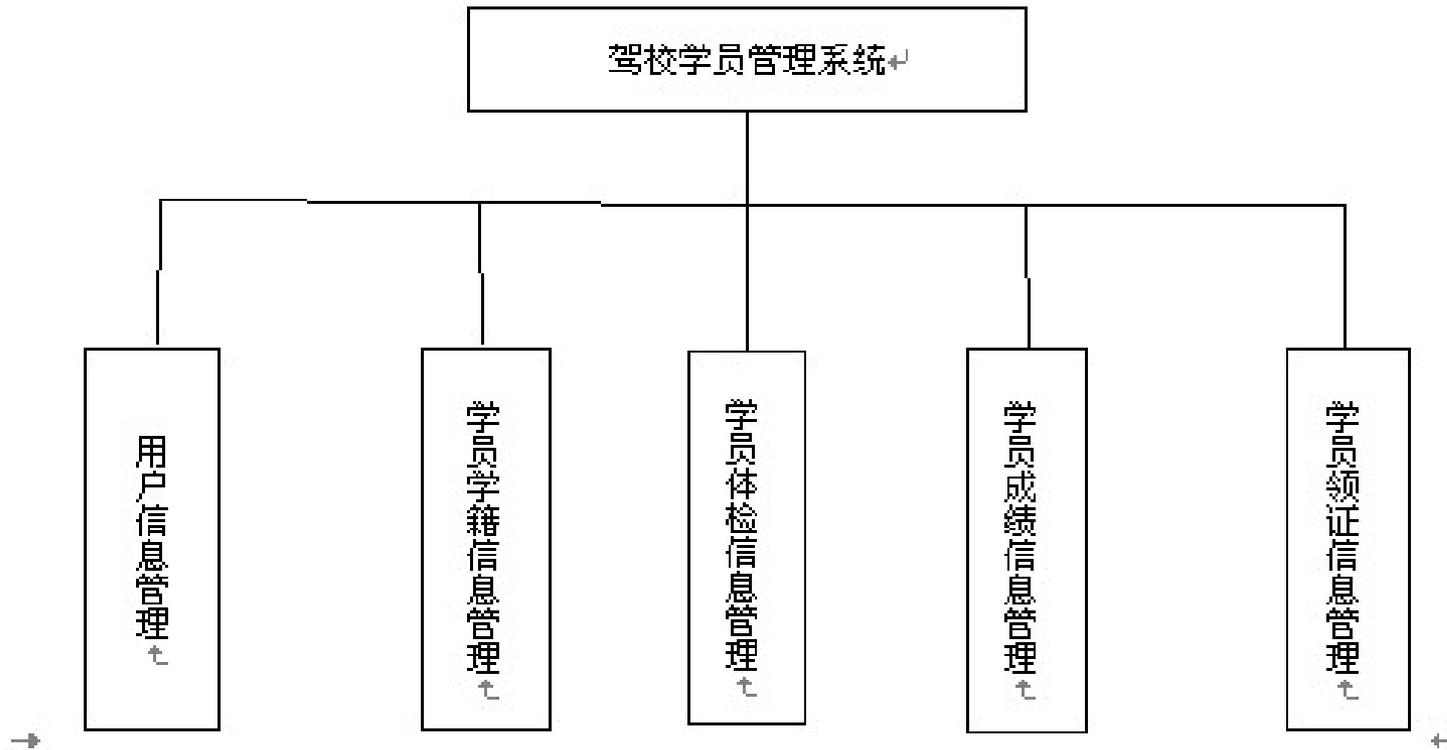
随着驾校学员的增加，就会增加大量的数据。这些数据的增加，给驾校学员管理的管理员在资料的整理，资料的查询，数据的处理上带来很大的不便。建立本系统的基本目标是为了减少管理员的工作强度，使得对学员信息的查询和数据处理的速度得到很大程度的提高，从而提高管理员的工作效率。

22.2 系统功能

驾校学员管理系统的主要功能是管理驾校学员的基本信息。通过本管理系统，可以提高驾校的管理者的工作效率。本小节将详细的介绍本系统的功能。

本驾校学员管理系统分为如下五个管理部分：用户管理、学籍信息管理、体检信息管理、成绩信息管理、领证信息管理。

22.2 系统功能



22.1 系统功能模块图

22.3 数据库设计

数据库设计是开发管理系统的一个重要步骤。如果数据库设计不合理，会给后续的系统开发带来很大的麻烦。本小节为读者介绍驾校管理系统的数据库的设计过程。

数据库设计时要确定创建哪些表、表中有哪些字段、字段的数据类型和长度。本章介绍的驾校学员管理系统选择MySQL数据库。因为本书主要是介绍MySQL数据库的知识，所以在设计数据库时会尽量用到书中介绍过的MySQL数据库的知识点。这样可以让读者对MySQL数据库有一个全面的认识。

22.3.1 设计表

本系统所有的表都放在drivingschool数据库下，创建drivingschool数据库的SQL代码如下：

```
CREATE DATABASE drivingschool;
```

在这个数据库下一共存放六张表，分别是user表、studentInfo表、healthInfo表、courseInfo表、gradeInfo表和licenseInfo表。其中，user表存储管理员的用户名和密码；studentInfo表存储学员的学籍信息；healthInfo表存储学员的体检信息；courseInfo表存储学员的课程信息；gradeInfo表存储学员各科考试信息；licenseInfo表存储领取驾驶证的信息。

1 . user表

2 . studentInfo表

3 . healthInfo表

4 . courseInfo表

5 . gradeInfo表

6 . licenseInfo表

22.3.2 设计索引

索引是创建在表上的，是对数据库表中一列或多列的值进行排序的一种结构。索引可以提高查询的速度。驾校学员管理系统需要查询学员的信息，这就需要在某些特定字段上建立索引，以便提高查询速度。

- 1 . 在studentInfo表上建立索引**
- 2 . 在healthInfo表上建立索引**
- 3 . 在licenseInfo表上建立索引**

22.3.3 设计视图

视图由数据库中的一个表或多个表导出的虚拟表。其作用是方便用户对数据的操作。在这个管理系统中，也设计了一个视图改善查询操作。

在驾校学员管理系统中，如果直接查询gradeInfo表，显示信息时会显示学员的学号和考试的科目号。这种显示并不直观，为了以后查询方便，可以创建一个视图grade_view。这个视图显示编号、学号、姓名、课程名、last_time字段、times字段、grade字段。创建视图grade_view的SQL代码如下：

```
CREATE VIEW grade_view
AS SELECT
g.id,g.sno,s.sname,c.cname,last_time,times,grade
FROM studentInfo s,courseInfo c,gradeInfo g
WHERE g.sno=s.sno AND g.cno=c.cno;
```

22.3.4 设计触发器

触发器是由INSERT、UPDATE和DELETE等事件来触发某种特定操作。满足触发器的触发条件时，数据库系统就会执行触发器中定义的程序语句。这样做可以保证某些操作之间的一致性。为了使驾校学员管理系统的数据更新更加快速、合理，可以在数据库中设计几个触发器。

- 1. 设计INSERT触发器**
- 2. 设计UPDATE触发器**
- 3. 设计DELETE触发器**

22.4 系统实现

本驾校学员管理系统使用Java语言开发，系统开发环境为Eclipse和MyEclipse。本节将向读者介绍本系统的编码实现。

22.4.1 构建工程

首先，在MyEclipse创建一个Web工程，并将这个Web工程取名为DrivingSchool。按照19.1小节的内容将JDBC驱动添加到工程中。然后在工程中的src文件下创建两个包（Package），分别取名为db和servlet。db包下存放连接和处理MySQL数据库的Java类，servlet包下存放着所有的servlet文件。本工程的所有JSP页面都放在WebRoot文件夹下。

22.4.2 访问和操作MySQL数据库的代码

在db包下创建DB.java类。这个Java类中封装了五个方法。这些方法分别是connectMySQL()方法、query()方法、update()方法、execute()方法和closeDB()方法。connectMySQL()方法主要用于连接MySQL数据库。query()方法用于执行SELECT语句。update()方法用于执行INSERT语句、UPDATE语句、DELETE语句。execute()方法可以执行所有的SQL语句。closeDB()方法用于关闭数据库对象。下面分别介绍这几个方法的代码。

- 1 . connectMySQL()方法**
- 2 . query()方法**
- 3 . update()方法**
- 4 . excuteSQL()方法**
- 5 . closeDB()方法**

22.5 用户管理模块

用户管理模块包括两个功能，分别是用户登录功能和修改密码功能。用户登录功能是管理员进入管理系统的入口，只有输入正确的用户名和密码才能够登录成功。修改密码功能能够保证管理员帐号的安全。本节将为读者介绍用户登录功能和修改密码功能的内容。

22.5.1 用户登录功能

用户通过login.jsp页面输入用户名和密码。单击【登录】按钮就可以提交用户名和密码。login.jsp文件有个<form>表单，在<form>表单中通过post方法将用户名和密码提交给servlet文件夹下的userLogin.java文件。userLogin.java中调用DB.java类中的query()方法判断用户名和密码是否正取。

22.5.2 修改密码

用户登录成功后，可以在modifyPasswd.jsp页面修改用户密码。然后将修改后的密码提交给modifyPasswd.java。modifyPasswd.java将新密码更新到user表中。

这里的用户名是登录用户的名称，用户名是不能修改的。页面需要输入旧密码，并且输入两次新密码。如果旧密码不正确或者两次输入的新密码不相同，那么系统会跳转到错误页面。如果输入都正确后，旧密码和新密码被提交到modifyPasswd.java文件中。

22.6 学籍管理模块

学籍管理模块主要管理学员的学籍信息。该模块包括四个功能，分别是添加学员的学籍信息、查询学员的学籍信息、修改学员的学籍信息和删除学员的学籍信息。本节将为读者介绍这四个功能的内容。

22.6.1 添加学员的学籍信息

管理员进入insertStudent.jsp页面，在该页面中添加学员的学籍信息。添加完成后，管理系统会将学籍信息传递给insertStudent.java文件。insertStudent.java文件中调用update()方法，通过该方法将新纪录插入到studentInfo表中。

22.6.2 查询学员的学籍信息

管理员进入queryStudent.jsp页面查询学籍信息，该页面会将查询条件传递给queryStudent.java文件。在queryStudent.java中会根据传递过来的查询条件组合成不同的SELECT语句。然后调用query()方法执行SELECT语句，从studentInfo表中查询出满足条件的记录。

22.6.3 修改学员的学籍信息

管理员进入modifyStudent.jsp页面后，可以修改学员的学籍信息。修改完成后，单击【确定】按钮，修改后的信息就可以提交给modifyStudent.java文件。这个文件调用DB.java中的update()方法将修改的数据写入studentInfo表。

22.6.4 删除学员的学籍信息

管理员在queryStudent.jsp页面查询信息后，可以在每个信息的后面看到【删除】链接。单击【删除】链接后，程序会将学员的学号（sno）值传递给deleteStudent.java文件。deleteStudent.java文件获取sno值后，会生成DELETE语句。然后调用update()方法执行DELETE语句。deleteStudent.java文件中生产DELETE语句的代码如下：

```
sql="DELETE FROM studentInfo WHERE  
sno='"+sno;
```

22.7 体检管理模块

体检管理模块主要管理学员的体检信息。该模块包括四个功能，分别是添加学员的体检信息、查询学员的体检信息、修改学员的体检信息和删除学员的体检信息。本节将为读者介绍这四个功能的内容。

- 1. 添加学员的体检信息**
- 2. 查询学员的体检信息**
- 3. 修改学员的体检信息**
- 4. 删除学员的体检信息**

22.8 成绩管理模块

成绩管理模块主要管理学员的成绩信息。该模块包括四个功能，分别是添加学员的成绩信息、查询学员的成绩信息、修改学员的成绩信息和删除学员的成绩信息。本节将为读者介绍这四个功能的内容。

- 1. 添加学员的成绩信息**
- 2. 查询学员的成绩信息**
- 3. 修改学员的成绩信息**
- 4. 删除学员的成绩信息**

22.9 证书管理模块

证书管理模块主要管理学员的领证信息。该模块包括四个功能，分别是添加领证信息、查询领证信息、修改领证信息和删除领证信息。本节将为读者介绍这四个功能的内容。

- 1 . 添加领证信息**
- 2 . 查询领证信息**
- 3 . 修改领证信息**
- 4 . 删除领证信息**

22.10 小结

本章介绍了开发驾校学员管理系统的方法。本章的重点内容是数据库设计部分。因为本书主要是介绍MySQL数据库的使用，所以数据库设计部分结合了本书前面介绍的知识点。在数据库设计部分，不仅涉及了表和字段的设计，还涉及了索引、视图、触发器等内容。其中，为了提高表的查询速度，有意识的在表中增加了冗余字段，这是数据库的性能优化的内容。系统实现部分是本章的难点，需要读者Java语言和J2EE有相应的了解。通过本章的学习，希望读者对项目开发中如何使用MySQL数据库有一个全新的认识。