



高等教育立体化精品教材

# MySQL数据库基础

MySQL SHUJUKU JICHU

罗印主编

汕头大学出版社

## 图书在版编目 (CIP) 数据

MySQL 数据库基础/罗印主编. — 汕头: 汕头大学出版社, 2020. 8

ISBN 978-7-5658-4100-2

I. ①M… II. ①罗… III. ①SQL 语言—程序设计  
IV. ①TP311.132.3

中国版本图书馆 CIP 数据核字 (2020) 第 145328 号

MySQL 数据库基础

MySQL SHUJUKU JICHU

主 编: 罗 印

责任编辑: 汪艳蕾

责任技编: 黄东生

封面设计: 易 帅

出版发行: 汕头大学出版社

广东省汕头市大学路 243 号汕头大学校园内 邮政编码: 515063

电 话: 0754-82904613

印 刷: 天津市蓟县宏图印务有限公司

开 本: 787mm×1092mm 1/16

印 张: 14.5

字 数: 302 千字

版 次: 2020 年 8 月第 1 版

印 次: 2020 年 8 月第 1 次印刷

定 价: 42.00 元

ISBN 978-7-5658-4100-2

版权所有, 翻版必究

如发现印装质量问题, 请与承印厂联系退换

# INTRODUCTION 内 容 简 介

本书介绍了数据库及 MySQL 的基础知识，以案例为驱动，循序渐进地介绍了 MySQL 数据库的基本操作及在实践中的具体应用。全书共分为 10 章，内容包括数据库基础知识、数据库设计、数据库及数据表的基本操作、数据更新、数据查询、数据视图、索引、数据库编程、数据安全、综合案例。

本书配有大量的演示案例和课后练习，注重实际操作，具有很强的实用性。本书既可作为高等院校相关专业的教材，也可作为 MySQL 的初学人员及程序开发人员的学习参考用书。







# PREFACE 前言

本书主要针对大学计算机软件相关专业学生及希望学习 MySQL 技术的读者而编写。在编写的过程中，充分突出了“理论够用实践为主”的特点，以案例驱动，通过小案例、课堂案例、综合案例贯穿全书内容。对每个知识点，按照理论讲解、案例演示、案例分析、应用技巧进行组织编写，符合高校学生的学习特点。教材注重启发和培养学生的编程思想，将理论知识应用到实践分析和解决问题中。

本书介绍了数据库及 MySQL 的基础知识，以案例为驱动，循序渐进地介绍了 MySQL 数据库的基本操作及在实践中的具体应用。全书共分为 10 章，内容包括数据库基础知识、数据库设计、数据库及数据表的基本操作、数据更新、数据查询、数据视图、索引、数据库编程、数据安全、综合案例。

本书配有大量的演示案例和课后练习，注重实际操作，具有很强的实用性。本书既可作为高等院校相关专业的教材，也可作为 MySQL 的初学人员及程序开发人员的学习参考用书。

本书由罗印担任主编，并负责编写第 1 章和第 3 章的内容，由严新巧负责编写第 2 章和第 4 章的内容，由张受亭负责编写第 5 章和第 6 章的内容，由赵勇负责编写第 7 章和第 9 章的内容，由刘雯雯负责编写第 8 章和第 10 章的内容。罗印负责全书的目录制定、内容优化、统稿及审稿等工作。

由于作者水平有限，难免有不妥之处，恳请广大读者批评指正，并提出宝贵的意见和建议。

编 者





# CONTENTS 目录

## 第 1 章 数据库基础知识

|                  |    |
|------------------|----|
| 1.1 基本概念         | 2  |
| 1.2 关系数据库        | 5  |
| 1.3 MySQL 的安装与配置 | 8  |
| 本章小结             | 24 |
| 课后习题             | 24 |

## 第 2 章 数据库设计

|              |    |
|--------------|----|
| 2.1 数据库设计概述  | 26 |
| 2.2 概念结构设计   | 27 |
| 2.3 逻辑结构设计   | 31 |
| 2.4 物理结构设计   | 41 |
| 2.5 数据库实施与维护 | 43 |
| 本章小结         | 45 |
| 课后习题         | 45 |

## 第 3 章 数据库及数据表的基本操作

|                |    |
|----------------|----|
| 3.1 数据库的基本操作   | 48 |
| 3.2 MySQL 存储引擎 | 54 |
| 3.3 数据类型       | 60 |
| 3.4 数据表的基本操作   | 64 |
| 本章小结           | 71 |
| 课后习题           | 71 |

## 第 4 章 数据更新

|          |    |
|----------|----|
| 4.1 插入数据 | 74 |
| 4.2 数据修改 | 82 |
| 4.3 数据删除 | 84 |
| 本章小结     | 87 |
| 课后习题     | 87 |





## 第 5 章 数据查询

|                        |     |
|------------------------|-----|
| 5.1 MySQL 数据基本查询 ..... | 90  |
| 5.2 MySQL 查询条件选择 ..... | 101 |
| 5.3 MySQL 多表查询 .....   | 113 |
| 5.4 行浏览查询 .....        | 126 |
| 本章小结 .....             | 128 |
| 课后习题 .....             | 129 |

## 第 6 章 数据视图

|                   |     |
|-------------------|-----|
| 6.1 视图的基本概念 ..... | 132 |
| 6.2 视图的操作 .....   | 133 |
| 6.3 对性能的影响 .....  | 139 |
| 本章小结 .....        | 140 |
| 课后习题 .....        | 140 |

## 第 7 章 索引

|                |     |
|----------------|-----|
| 7.1 索引简介 ..... | 142 |
| 7.2 创建索引 ..... | 143 |
| 7.3 删除索引 ..... | 145 |
| 本章小结 .....     | 145 |
| 课后习题 .....     | 145 |

## 第 8 章 数据库编程

|                      |     |
|----------------------|-----|
| 8.1 存储过程的基本操作 .....  | 148 |
| 8.2 MySQL 存储函数 ..... | 167 |
| 8.3 触发器 .....        | 174 |
| 本章小结 .....           | 182 |
| 课后习题 .....           | 182 |

## 第 9 章 数据安全

|                    |     |
|--------------------|-----|
| 9.1 用户与权限 .....    | 184 |
| 9.2 数据的备份与恢复 ..... | 193 |
| 本章小结 .....         | 204 |
| 课后习题 .....         | 204 |





---

## 第 10 章 综合案例

---

|                   |     |
|-------------------|-----|
| 10.1 功能需求概述 ..... | 206 |
| 10.2 概要结构设计 ..... | 206 |
| 10.3 逻辑结构设计 ..... | 211 |
| 10.4 物理结构设计 ..... | 215 |

---

## 参考文献

---





# 第 1 章 数据库基础知识

随着社会的不断发展，计算机技术已深入人们生产、生活的各个方面。数据对于人们来说越来越重要，要管理好各种数据就离不开数据库管理系统。开源数据库管理系统由于其性能高、成本低、可靠性好等特点，越来越受到开发者的青睐。MySQL数据库管理系统就是开源数据管理系统库中的杰出代表。本章将对数据库的基本概念、关系数据库，以及MySQL的安装和配置等进行详细的介绍。

## 学习目标

- 了解数据库的相关概念。
- 掌握关系数据模型及关系完整性的概念。
- 掌握MySQL数据库的安装与配置方法。



## 1.1 基本概念

20世纪50年代中期以前，数据主要依赖于人工管理。20世纪50年代后期到20世纪60年代中期，随着计算机硬件和软件技术的进一步发展，有了存储设备和文件系统。到20世纪60年代后期，出现了数据库系统，满足了多用户、多应用共享数据的需求。本节将对数据库中的基本概念进行介绍。

### 1.1.1 数据与数据库

数据(Data)是对客观事物的符号记录，是按照一定规则排列组合的物理符号，可以是数字、文字、图像等。在计算机中，数据是指经过数字化后能被计算机处理的符号记录。数据是描述和管理信息的有效载体。

例如，要描述一个学生的基本信息可以这样来记录：(2020010201, 张三, 男, 18, 计算机, 1班)。这样的一条记录就是数据。对这条数据可以解释为：学号为2020010201的学生，姓名为张三，性别为男，年龄为18岁，就读于计算机专业1班。通过解释明确了数据的含义，也称为数据的语义，即信息。数据是对客观事物的记录，是对信息的符号表示，而信息是对数据的语义解释，是数据的内涵。

信息与数据既有区别，又紧密联系。一般情况下，信息与数据无须严格区分。例如，数据处理与信息处理、数据资源与信息资源，表达的是同一个意思。

数据库(Database)可以理解存储数据的地方。在计算机中，数据库是指存储在计算机内，有组织、可共享的数据集合。在数据库中保存的是以一定方式存储在一起的相互关联的数据整体。数据库既保存了数据，也保存了数据与数据之间的相互关系。通过数据库对数据进行存储，可以方便地对数据进行查询、更新、添加和删除等操作。

例如，将一个班的学生信息进行存储，可以使用如表1-1所示的结构。

表 1-1 学生信息表

| 学号         | 姓名 | 性别 | 专业  | 班级 |
|------------|----|----|-----|----|
| 2020010201 | 张三 | 男  | 计算机 | 1班 |
| 2020010202 | 李四 | 男  | 计算机 | 1班 |
| 2020010203 | 王五 | 男  | 计算机 | 1班 |

一个或多个数据表可以看作一个数据库。每个表中的数据之间存在一定的关系，表与表之间也可能存在一定的关系。

数据库实现了数据的独立性和共享性，减少了数据冗余度，实现了数据的集中控制，提高了数据的可恢复性，也维护了数据的完整性。

### 1.1.2 数据库系统

数据库系统(Database System, DBS)通常由硬件、数据库、管理软件、用户4个部







分组成，如图 1-1 所示。

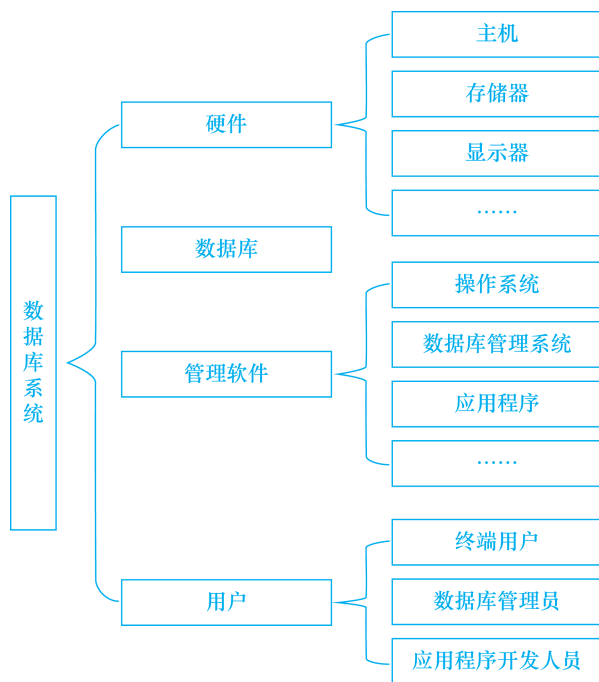


图 1-1 数据库系统的组成

硬件是构成计算机系统的各种物理设备，包括主机、存储器、显示器等。由于数据库系统一般存放和处理的数据量比较大，且对数据处理性能有较高的要求，所以对计算机硬件有较高的要求，包括足够大的内存、足够大的存储容量、较快的处理速度和数据传输速率等。

数据库是数据的集合，是数据库系统管理的对象，也是用户操作的对象。

数据库系统中的管理软件包括操作系统、数据库管理系统和各种应用程序。数据库管理系统是数据库系统中的核心软件，在操作系统的支持下，负责组织 and 存储数据，并能高效地获取和维护数据。

用户主要包括终端用户、应用程序开发人员和数据库管理员。终端用户通过应用程序提供的操作界面对数据库进行操作。应用程序开发人员负责根据用户需求设计和开发应用程序，通过应用程序对数据库进行数据获取和维护等，并向终端用户提供友好的操作界面。数据库管理员负责对数据库系统进行日常的管理和维护，保障数据库系统的正常运行。数据库管理员处于终端用户和开发人员之间。

### 1.1.3 数据库管理系统

数据库管理系统 (Database Management System, DBMS) 是位于用户与操作系统之间的一种数据管理软件。使用数据库管理系统可以创建、使用和维护数据库。数据库管理系统建立在操作系统的基础上，对数据库进行统一的管理和控制，以保证数据库的安全和完整性。





数据库管理系统的主要功能包括以下几点。

(1) 数据定义。数据库管理系统提供了数据定义语言 (Data Definition Language, DDL), 用于定义数据库的结构、完整性约束条件和触发机制等。

(2) 数据操纵。数据操纵语言 (Data Manipulation Language, DML) 可以实现对数据的操作, 包括对数据库中数据的查询、添加、修改、删除等。

(3) 数据库的建立与维护。数据库管理系统提供了一些管理维护程序, 可以实现对数据库初始数据的输入, 对工作日志进行记录, 对数据库性能进行监视, 重新组织数据库, 修改和更新数据库及数据恢复等。

(4) 数据库控制。数据库管理系统还提供了一系列运行控制程序, 可以实现对运行中的数据库进行管理和控制。对数据库的管理和控制主要体现为以下几点。

① 协调多用户同时访问数据库时的访问进程。

② 核对用户名和口令, 检验用户访问的合法性。

③ 检查和执行数据库完整性约束条件。在数据库操作前或操作后, 核对完整性约束条件, 从而决定执行数据库操作或清除执行操作后的影响。

④ 自动维护数据库的索引、数据字典等。

数据库的所有操作都需要在数据库管理系统的统一管理下运行, 以保证数据的正确有效。

#### 1.1.4 结构化查询语言

在与数据库进行交互时通常需要一种语言来完成对数据库的操作。目前, 被广泛使用的关系数据库标准语言就是结构化查询语言 (Structured Query Language, SQL)。使用 SQL 可以实现对关系数据库中数据的查询、插入、更新、删除等, 也可以对关系数据库进行管理。

SQL 是一种高级的非过程化编程语言, 允许用户在高层数据结构上工作, 既不要求用户指定数据的存放方法, 也不需要知道数据的具体存放方式, 只需要指出“做什么”即可。由于 SQL 简单易学, 且功能丰富, 因此它已被定为关系数据库语言的国际标准。

SQL 中的语句主要分为以下 4 类。

(1) 数据定义。数据定义语言主要负责定义数据库对象及数据结构。例如, 对数据库对象的创建、修改和删除。常使用的数据定义语句包括 CREATE、ALTER、DROP 等。例如, 创建数据库 cjgl 可以使用语句: CREATE DATABASE cjgl。

(2) 数据操纵。数据操纵语言主要负责对数据库中的数据进行访问, 包括数据的插入、修改、删除和查询等。常用的数据操纵语句包括: INSERT、UPDATE、DELETE、SELECT 等。例如, 查询 student 表的数据可以使用语句: SELECT \* from student。

(3) 数据控制。数据控制语言 (Data Control Language, DCL) 主要负责对数据的访问权限进行控制, 可以实现用户对数据表、预存程序和函数等的使用权限的管理。

(4) 事务处理。数据库中的事务指的是一组不可分割的操作, 要么都执行, 要么都不执行。事务处理语句负责对事务中的操作进行控制。例如, 可以通过 BEGIN TRANSACTION 语句开启事务; 通过 COMMIT TRANSACTION 语句实现事务的提交; 通过 ROLLBACK TRANSACTION 语句实现事务的回滚。





## 1.2 关系数据库

不同数据库系统采用的数据模型有所不同，数据库管理系统也是根据数据模型有针对性地设计出来的。根据数据模型的不同，可以将数据库系统分为：层次模型数据库、网状模型数据库和关系数据库。层次模型数据库采用“树结构”表示数据之间的联系，是最早使用的一种数据库系统。网状模型数据库采用“网结构”表示数据之间的联系。关系数据库以“二维表”表示数据之间的联系。本节将详细介绍关系数据库的基础知识。

### 1.2.1 关系数据模型

数据模型（Data Model）是对现实世界数据的模拟和抽象，是数据库的核心与基础，提供了信息表示和操作手段的形式架构。数据模型的主要作用是确定系统中数据的定义和格式，使人更容易理解，以便在数据库系统上得以更好的实现。

数据模型通常由数据结构、数据操作和完整性约束 3 个部分组成。

(1) 数据结构：是对数据静态特征的描述，包括数据的类型、内容、性质和数据之间的相互关系等。

(2) 数据操作：是对数据库对象动态特征的描述，主要包括操作类型和操作方式。

(3) 完整性约束：是一组完整性规则的集合，主要用于描述数据之间的语法、数据之间的相互关系和制约关系等，并规定了数据库状态及状态变化时所应满足的条件，以保证数据的正确性、有效性和相容性。

关系数据模型是以二维表来描述数据的。一个关系数据库中可以有多个二维表。每个二维表由行和列构成。表中一行也称为一条记录，每一行可以有一个或多个列，称为字段，每个字段有具体的数据类型，包括数字、字符串和日期等。表中行与列交叉的单元就是具体的数据。

例如，班级和学生之间的关系可以通过图 1-2 表示。

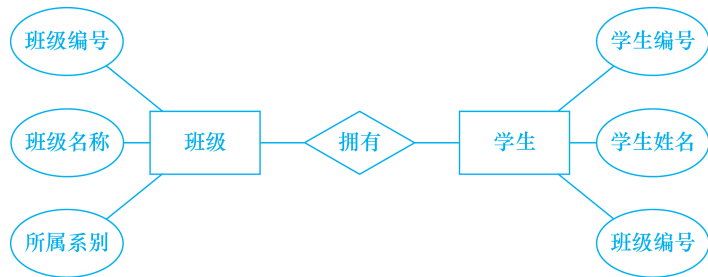


图 1-2 班级与学生之间的关系

一个班级可以有多个学生，一个学生只能属于一个班级。将班级与学生的这种关系转换为关系表，可以通过班级编号进行关联，如表 1-2 和表 1-3 所示。





表 1-2 班级表

| 班级编号  | 班级名称      | 所属系别 |
|-------|-----------|------|
| c_101 | 软件技术 1 班  | 计算机系 |
| c_102 | 软件技术 2 班  | 计算机系 |
| c_201 | 计算机应用 1 班 | 计算机系 |

表 1-3 学生表

| 学生编号       | 学生姓名 | 班级编号  |
|------------|------|-------|
| 2020010101 | 张三   | c_101 |
| 2020010102 | 李四   | c_101 |
| 2020010103 | 王五   | c_101 |
| 2020010201 | 赵六   | c_201 |

班级表与学生表通过班级编号建立了一对多的联系。

## 1.2.2 关系数据库管理系统

针对关系数据模型设计的数据库管理系统称为关系数据库管理系统（Relational Database Management System, RDBMS）。常见的关系数据库管理系统包括 SQL Server、DB2、Oracle 和 MySQL 等。

### 1 SQL Server

SQL Server 是 Microsoft 公司推出的关系数据库管理系统。该系统拥有图形化的操作界面，采用客户机/服务器的体系结构，使数据库的管理更加直观、简单和方便，具有可伸缩性好、集成度高等优点。SQL Server 可以运行在 Windows 平台的不同版本上。

### 2 DB2

DB2 是 IBM 公司开发的关系数据库管理系统，主要运行在 UNIX、Linux 等服务器操作系统上。DB2 主要应用于大型应用系统，具有较好的可伸缩性，可以支持从大型机到单用户环境。DB2 提供了高层次的数据利用性、完整性、安全性、可恢复性，以及小规模到大规模应用程序的执行能力，具有与平台无关的基本功能和 SQL 命令。

### 3 Oracle

Oracle 是甲骨文公司开发的关系数据库管理系统。Oracle 具有可移植性好、使用方便、功能强等特点，适用于各类大、中、小、微型机环境。Oracle 一直在数据库领域处于领先地位，是目前世界上最流行的关系数据库管理系统。

### 4 MySQL

MySQL 由瑞典的 MySQL AB 公司开发，目前是 Oracle 公司旗下产品。MySQL 是非常流行的关系数据库管理系统之一。由于其性能高、成本低、可靠性好等特点，被广泛应用于各类中小型网站。随着 MySQL 的不断发展和完善，它也逐渐被应用于各类大型网站





和应用中，如百度、腾讯、新浪、谷歌、阿里巴巴等公司都用到了 MySQL。可以说，MySQL 是 Web 应用中最好的关系数据库管理系统之一。

MySQL 的主要优势包括以下几点。

- (1) 开源，对于大多数个人和公司来说，其开发成本极低。
- (2) 运行速度快，优化 SQL 查询算法，具有高效的查询速度。
- (3) 跨平台，支持在多种操作系统上运行，包括 Windows、Linux、Mac OS 等。
- (4) 为大多编程语言提供了支持，如 C、C++、Java、PHP、Perl、C# 等。
- (5) 支持多线程，充分利用 CPU 资源。
- (6) 支持多存储引擎，可以根据实际情况，选择不同的存储引擎，以发挥更好的性能和实现不同的功。
- (7) 提供 TCP/IP、ODBC 及 JDBC 等多种数据库连接方式。

### 1.2.3 关系的完整性约束

关系的完整性约束是为保证数据库中数据的正确性和相容性，对关系数据模型提出的某种约束条件或规则。完整性通常包括实体完整性、参照完整性和用户定义完整性。其中，实体完整性和参照完整性是关系数据模型必须满足的完整性约束条件。

#### 1 实体完整性

实体完整性也称为表中的完整性，指的是能唯一标示一个实体且属性不能为空值，也不能有重复的主码。在关系数据模型中，将能标示一条记录的属性或属性组称为候选码，一般选中其中的一个作为该关系的主码。包含在任何一个候选码中的属性都称为主属性。不包含在任何候选码中的属性称为非主属性。实体完整性规则是指关系中的主属性不能为空值，也不能重复。

例如，关系模式：学生（学号，姓名，班级，出生日期），“学号”为该关系的主码，主码对应的属性只有学号，所以学号不能为空值，也不能重复。这和现实世界的理解是一样的。如果一个学生的学号为空值，那么这条记录就没有意义了，每个学生的学号是不能重复的。又如，关系模式：选修（学号，课程号，成绩），通过“学号”和“课程号”的组合标示一条选修成绩，该关系的主码为“学号”和“课程号”构成的属性组，所以“学号”和“课程号”均不能为空，且“学号”和“课程号”的组合是唯一的。

#### 2 参照完整性

参照完整性是定义建立关系之间联系的主关键字与外部关键字引用的约束条件。在关系数据模型中，实体与实体之间的关系可以通过引用对应的实体关系模式的主码来表示。例如，有以下关系模式：

班级（班级编号，班级名称，所属系别）；

学生（学生编号，学生姓名，学生性别，所属班级编号）。

在上面的关系中，学生关系中的“所属班级编号”对应于班级关系中的“班级编号”，而“班级编号”是班级关系的主码，则“所属班级编号”是学生关系的外码。

参照完整性约束规则为：若属性 F 是基本关系 R 的外码，且与基本关系 S 的主码 K





相对应，则 R 中某条记录的 F 值必须与 S 中某条记录的主码值相同，或者取空值。

在学生关系中，“所属班级编号”可以取空值，表示该学生未分配班级，也可以取班级关系中的某个“班级编号”值，表示该学生属于这个班级。但如果学生的“所属班级编号”不为空，且取值为一个不存在的“班级编号”，则表明该学生被分配到了一个不存在的班级，这是不合理的，违背了参照完整性约束规则。

### 3 用户定义完整性

用户定义完整性是根据应用环境的要求和实际的需要，对某一具体应用所涉及的数据提出约束性条件。这一约束机制一般不应由应用程序提供，而应由关系数据模型提供定义并检验，用户定义完整性主要包括字段有效性约束和记录有效性约束。

例如，在学生（学号，姓名，性别，班级，年龄）关系中，性别只能取“男”或“女”，年龄必须是 16~60 的值。

## 1.3 MySQL 的安装与配置

在使用 MySQL 之前需要先安装和配置 MySQL 服务器。MySQL 目前分为社区版和企业版，社区版是完全免费的，但官方不提供任何技术支持；企业版是收费的，官方提供技术支持和更多功能。本节将详细介绍在 Windows 操作系统下安装和配置 MySQL 社区版的详细过程，MySQL 的基本操作，以及图形化界面管理工具的使用。

### 1.3.1 MySQL 服务器的安装与配置

MySQL 服务器的安装包可以到 MySQL 官网下载。下载 MySQL 安装包文件的具体步骤如下。

(1) 在浏览器地址栏中输入：<https://dev.mysql.com/downloads/>，按“Enter”键进入下载页面，如图 1-3 所示。

(2) 单击超链接“MySQL Installer for Windows”，跳转到 MySQL 版本选择页面。单击“Looking for the latest GA version?”，在“Select Version”下拉列表框中选择“5.7.29”，在“Select Operating System”下拉列表框中选择“Microsoft Windows”，如图 1-4 所示。



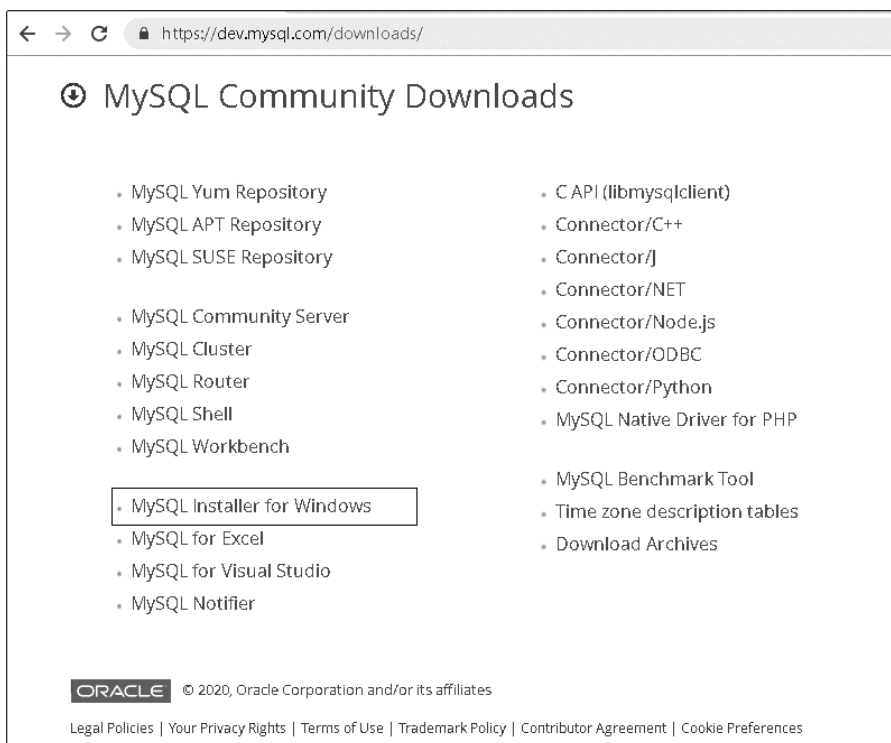


图 1-3 MySQL 下载页面

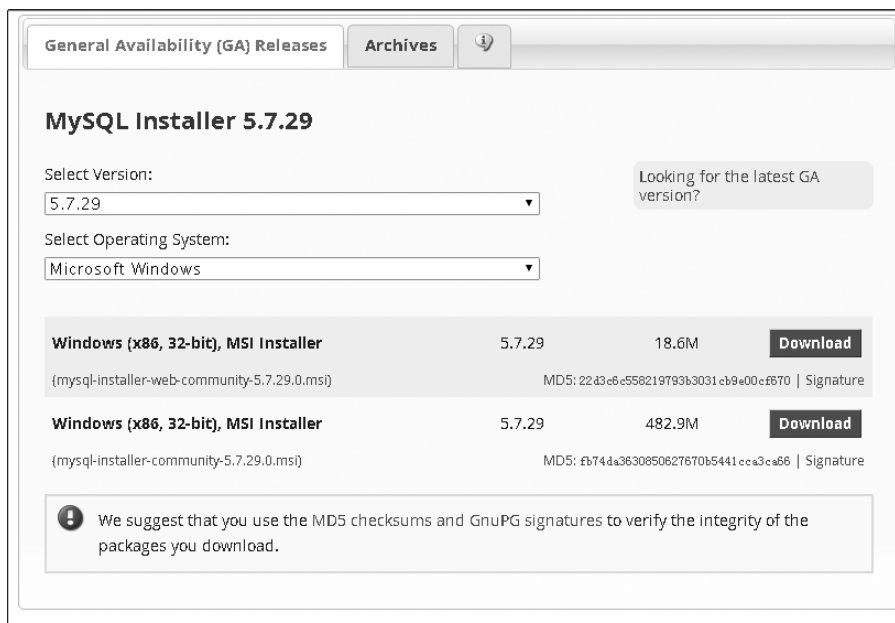


图 1-4 MySQL 版本选择页面

(3) 单击“Windows (x86, 32-bit), MSI Installer”右边的“Download”按钮，跳转到用户注册页面，单击页面下方的“No thanks, just start my download.”超链接，开始下载，如图1-5所示。



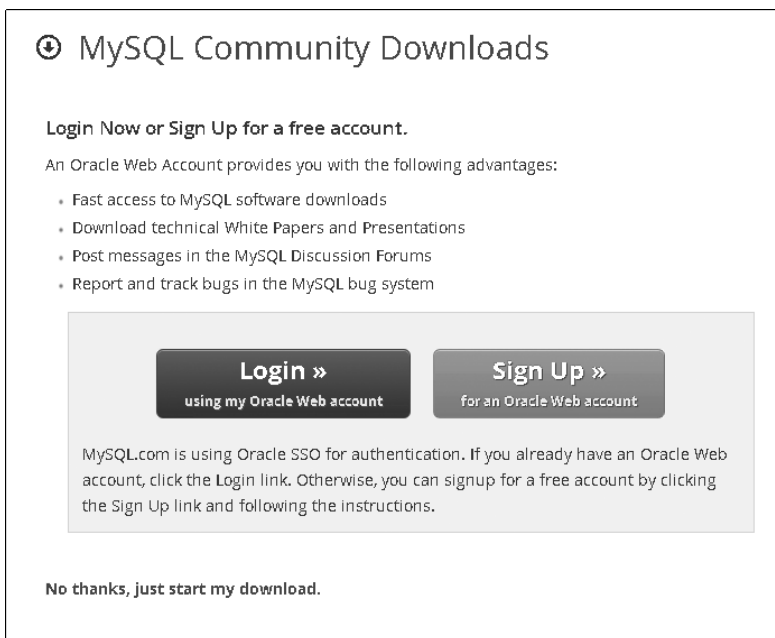


图 1-5 开始下载 MySQL

MySQL 安装包文件下载成功后，便可以安装 MySQL 服务器。安装 MySQL 的具体步骤如下。

(1) 双击下载的 mysql-installer-community-5.7.29.0.msi 文件，进入安装向导。首先需要选择一种安装类型。安装类型包括 5 种：“Developer Default”（开发者默认）、“Server only”（仅服务器）、“Client only”（仅客户端）、“Full”（完全）、“Custom”（自定义），如图 1-6 所示。

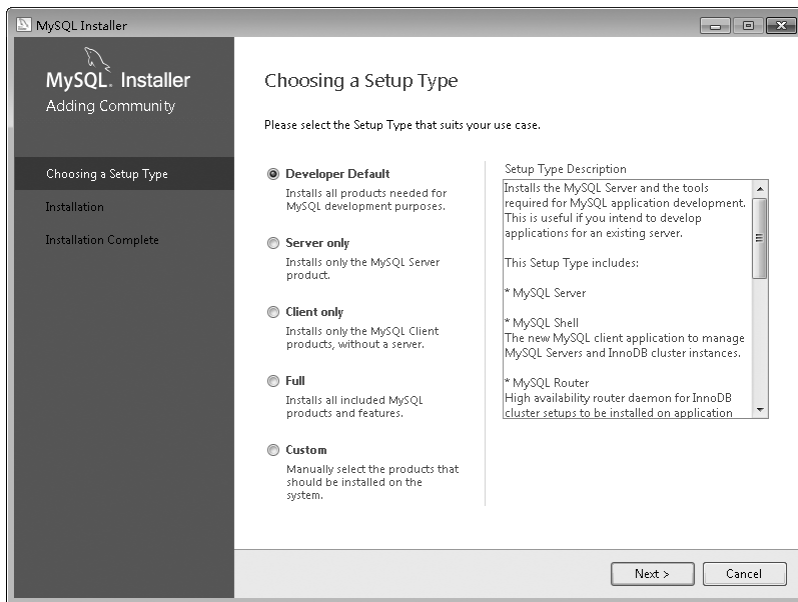


图 1-6 选择安装类型





(2) 选择“Developer Default”单选按钮，然后单击“Next”按钮，进入“Path Conflicts”对话框，如图 1-7 所示。

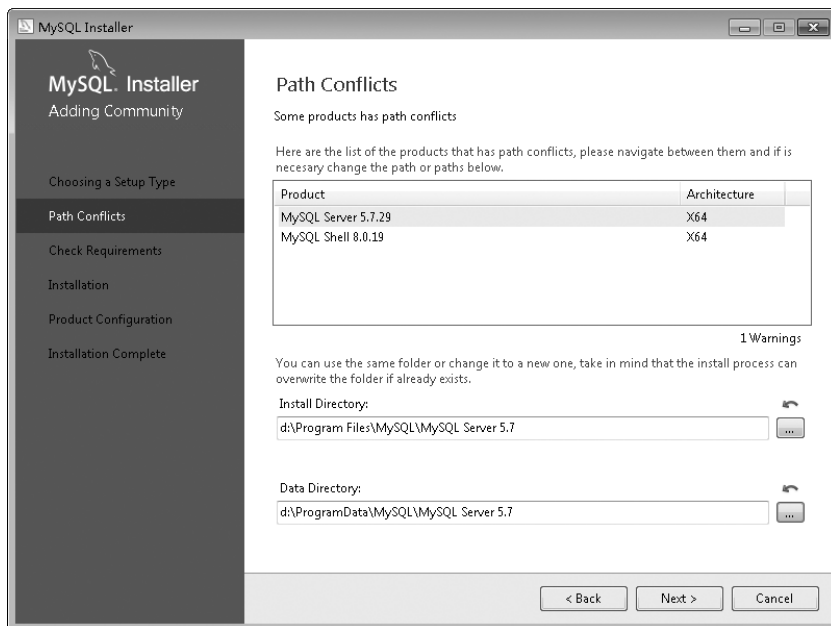


图 1-7 安装路径设置

(3) 在“Install Directory”文本框中设置 MySQL 服务器的安装位置，在“Data Directory”文本框中设置数据的存放位置，然后单击“Next”按钮，进入“Check Requirements”对话框。在该对话框中检查系统是否具备安装 MySQL 所必需的插件，如果不具备，单击“Execute”按钮，在线安装所需的插件，如图 1-8 所示。

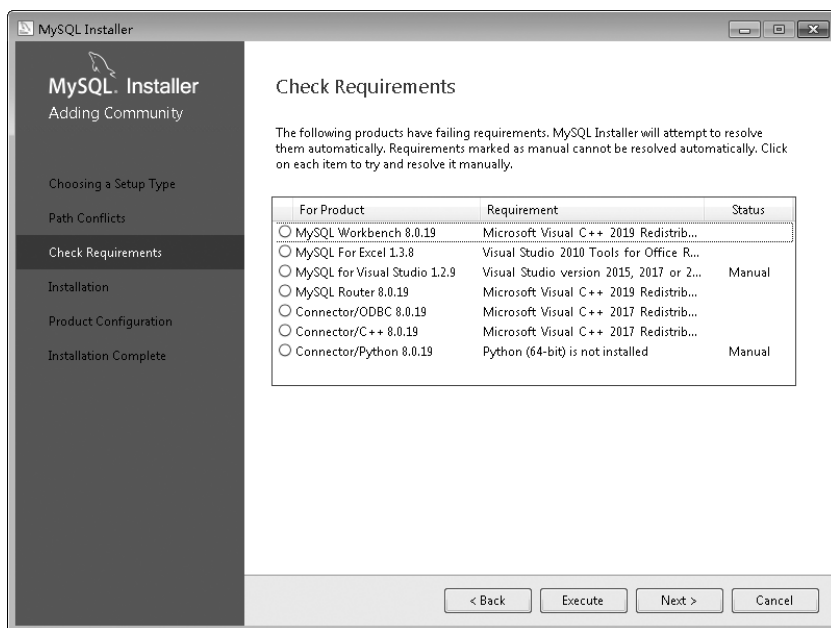


图 1-8 检查安装所需的插件



(4) 插件安装完成以后，单击“Next”按钮，进入“Installation”对话框，如图 1-9 所示。

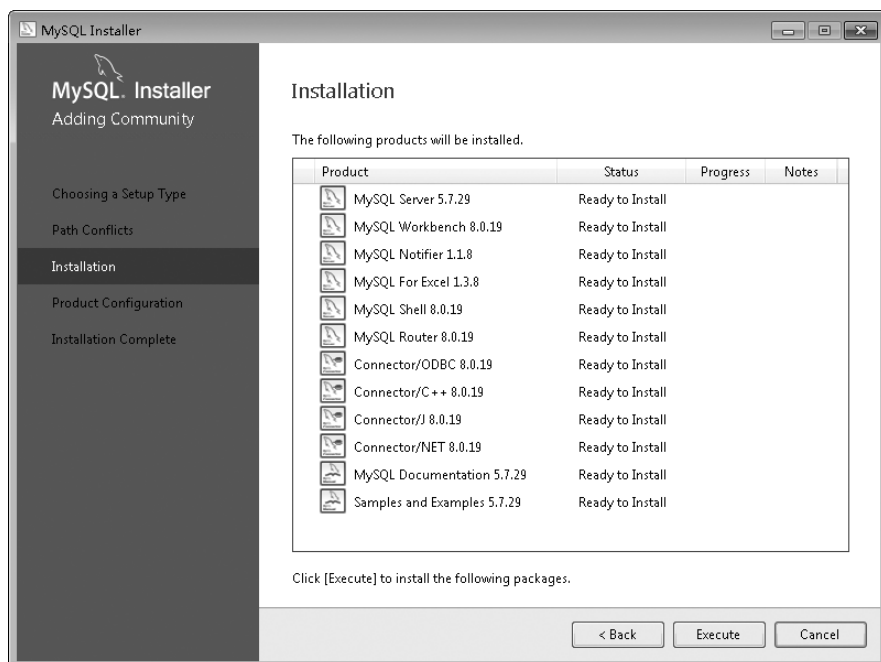


图 1-9 安装 MySQL 组件

(5) 单击“Execute”按钮，开始安装 MySQL 的各项组件。安装完成后单击“Next”按钮，进入“Product Configuration”对话框，如图 1-10 所示。

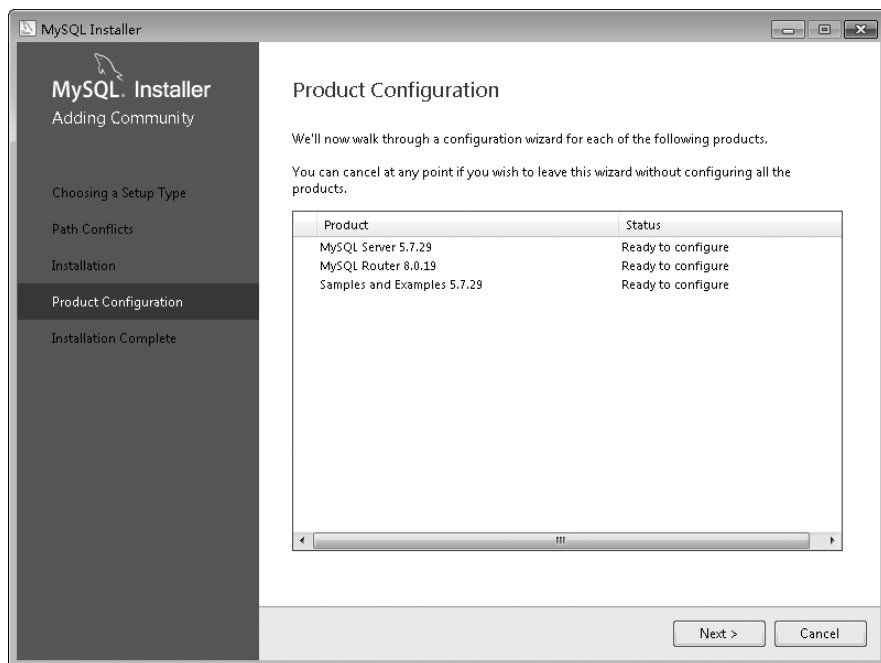


图 1-10 MySQL 配置



(6) 单击“Next”按钮，进入“High Availability”对话框，如图 1-11 所示。

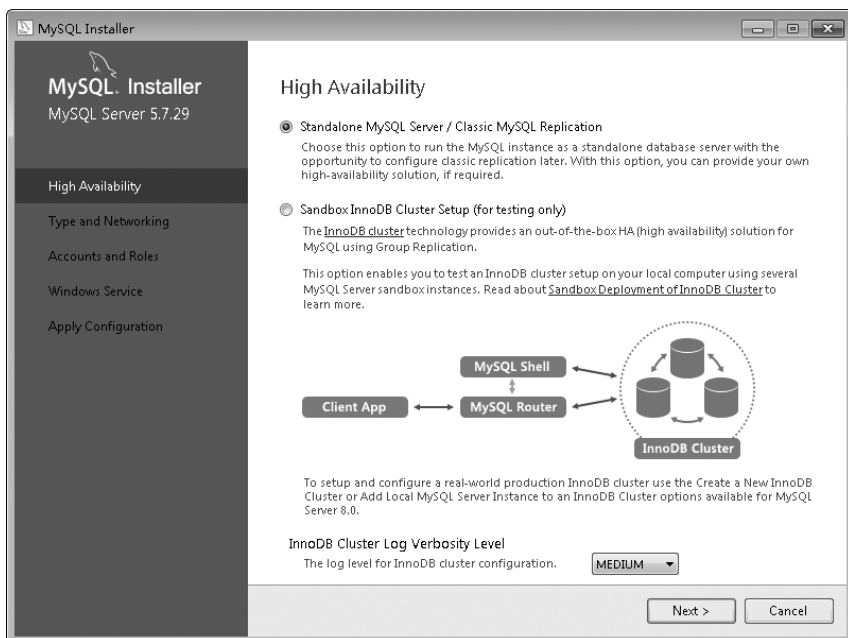


图 1-11 High Availability 配置

(7) 保持默认设置，单击“Next”按钮，进入“Type and Networking”对话框。在该对话框中配置服务器类型和网络选项。服务器类型包括：Development Computer（开发者类型）、Server Computer（服务器类型）、Dedicated Computer（致力于 MySQL 服务类型）3 种。网络选项包括连接方式和端口的设置，如图 1-12 所示。

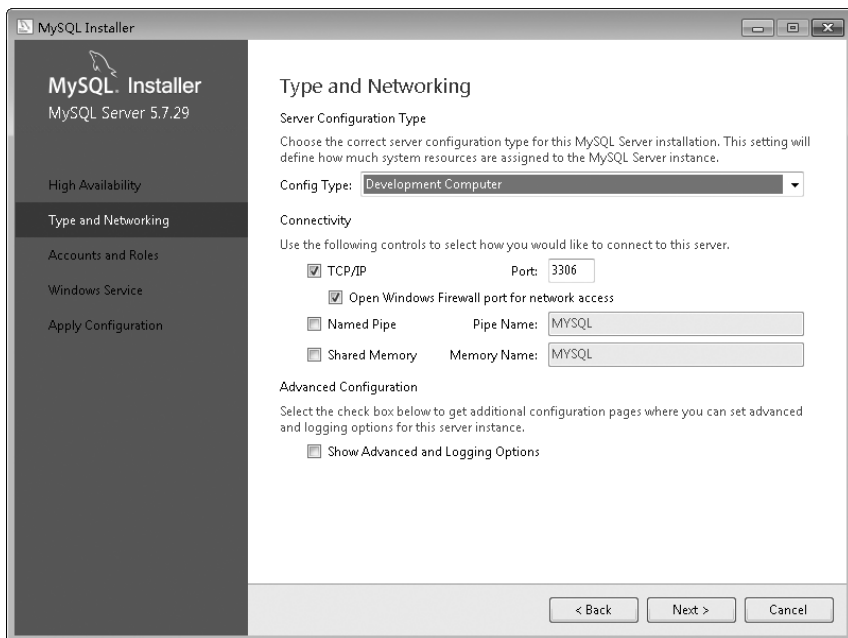


图 1-12 配置服务器类型和网络选项



(8) 选择默认的服务器类型“Development Computer”和默认的网络连接方式“TCP/IP”，并保持默认端口“3306”。单击“Next”按钮，进入“Accounts and Roles”对话框。在该对话框中可以设置默认用户 root 的登录密码，也可以添加新的用户，如图 1-13 所示。

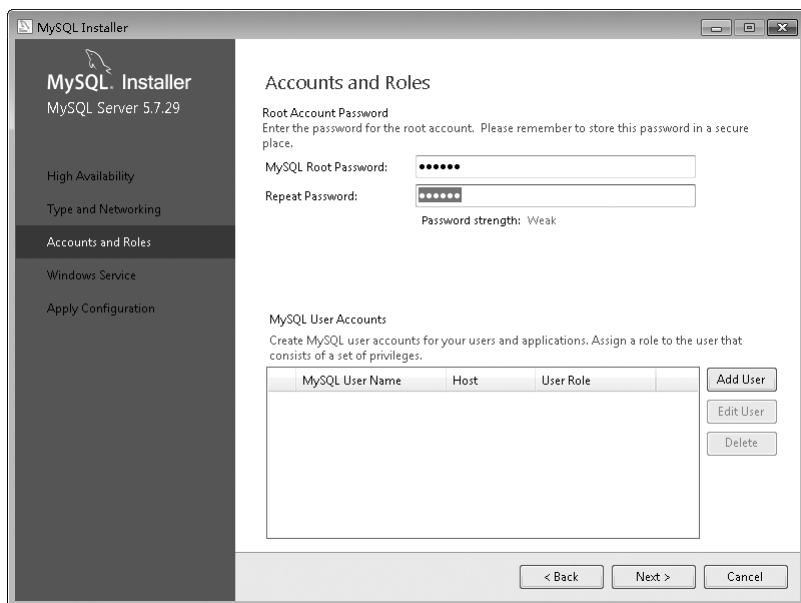


图 1-13 登录账号配置

(9) 在“MySQL Root Password”和“Repeat Password”文本框中输入相同的密码，然后单击“Next”按钮，进入“Windows Service”对话框。在该对话框中，配置 MySQL 服务器的名称，是否开机自启动等，如图 1-14 所示。

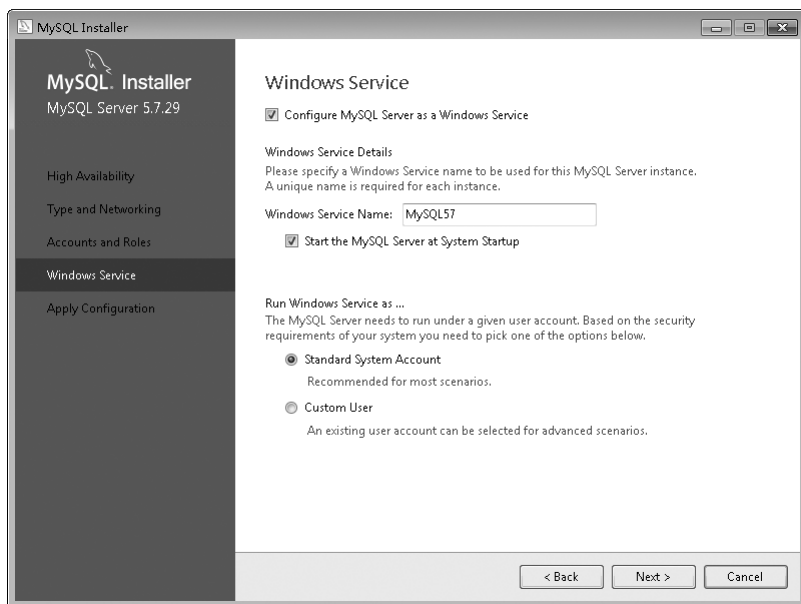


图 1-14 配置 MySQL 服务器



(10) 保持默认设置，单击“Next”按钮，进入“Apply Configuration”对话框，如图 1-15 所示。

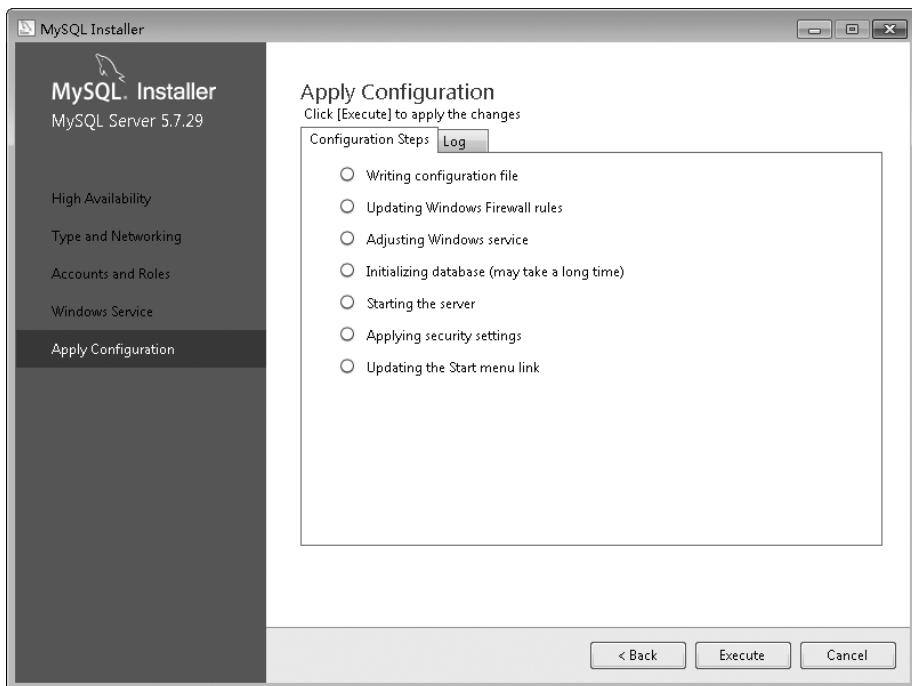


图 1-15 应用 MySQL 服务器配置

(11) 单击“Execute”按钮，完成 MySQL 服务器的配置，如图 1-16 所示。

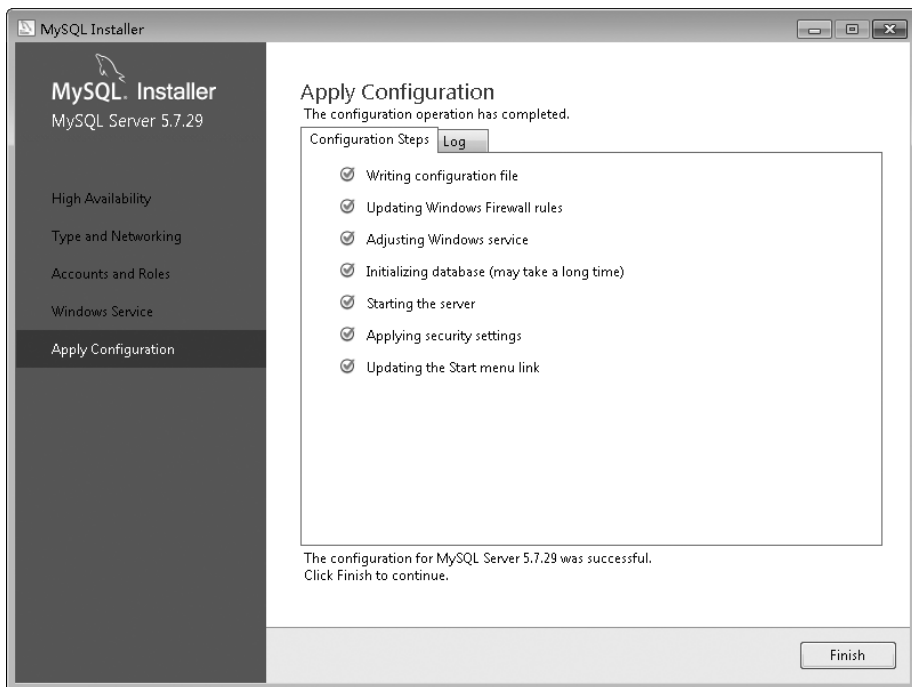


图 1-16 完成 MySQL 服务器的配置



(12) 单击“Finish”按钮，然后在弹出的对话框中单击“Next”按钮，进入“MySQL Router Configuration”对话框，如图 1-17 所示。

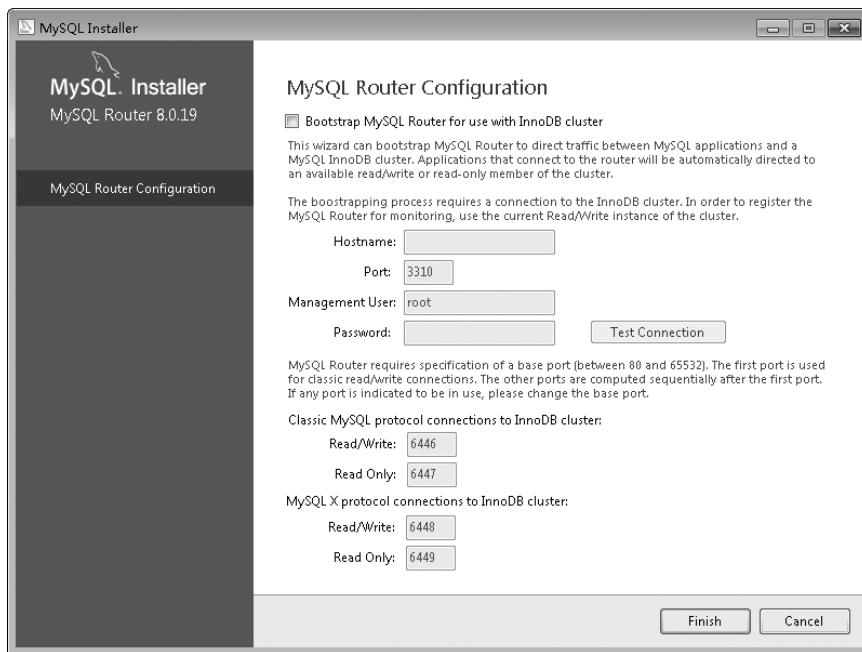


图 1-17 MySQL Router 配置

(13) 保持默认设置，单击“Finish”按钮，然后在弹出的对话框中单击“Next”按钮，进入“Connect To Server”对话框。在“Password”文本框中输入密码，单击“Check”按钮，测试连接，如图 1-18 所示。

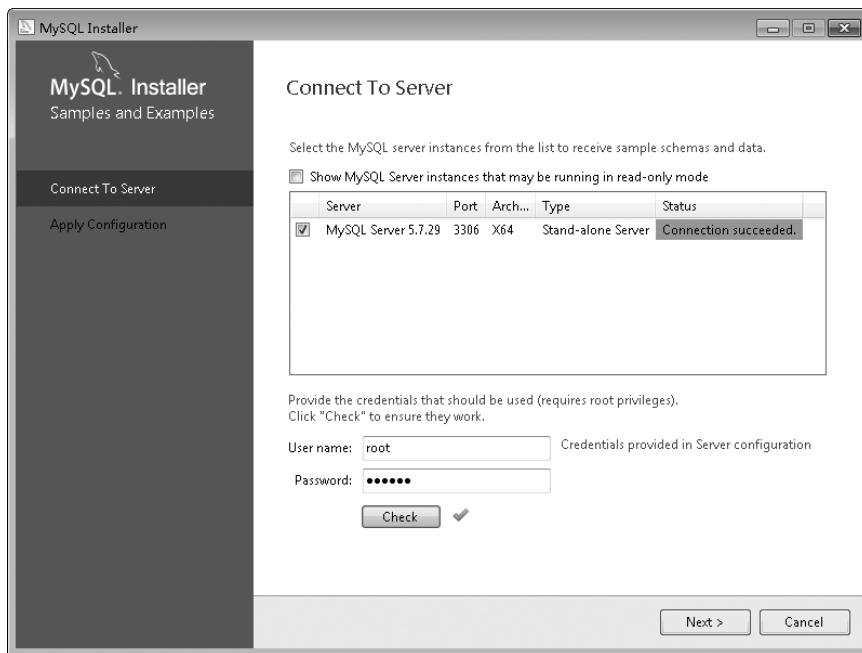


图 1-18 连接测试



(14) 单击“Next”按钮，进入“Apply Configuration”对话框，单击“Execute”按钮，应用配置，如图 1-19 所示。

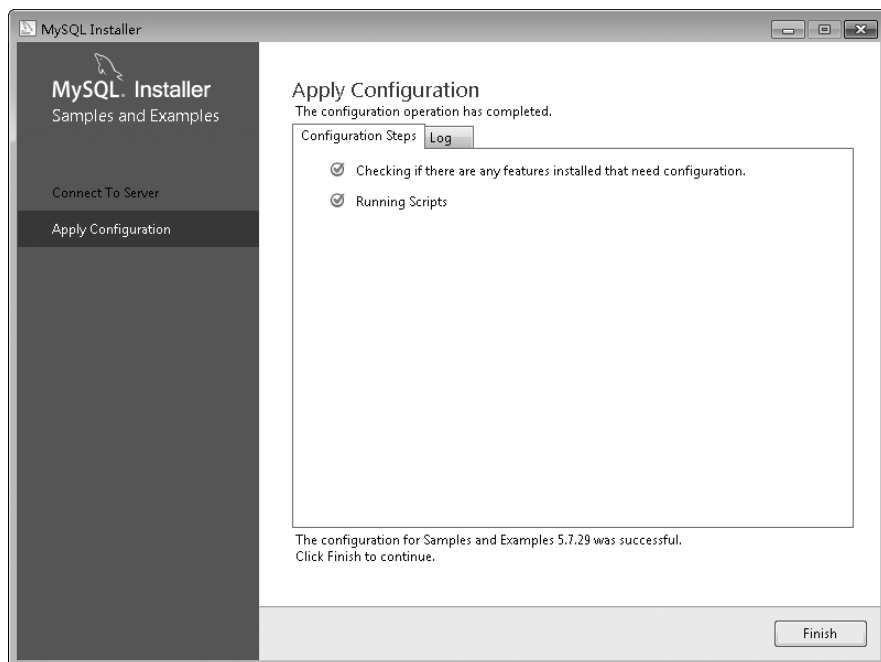


图 1-19 应用 MySQL Router 配置

(15) 单击“Finish”按钮，进入“Installation Complete”对话框，完成安装。在该对话框中，单击“Finish”按钮，完成整个安装过程，如图 1-20 所示。

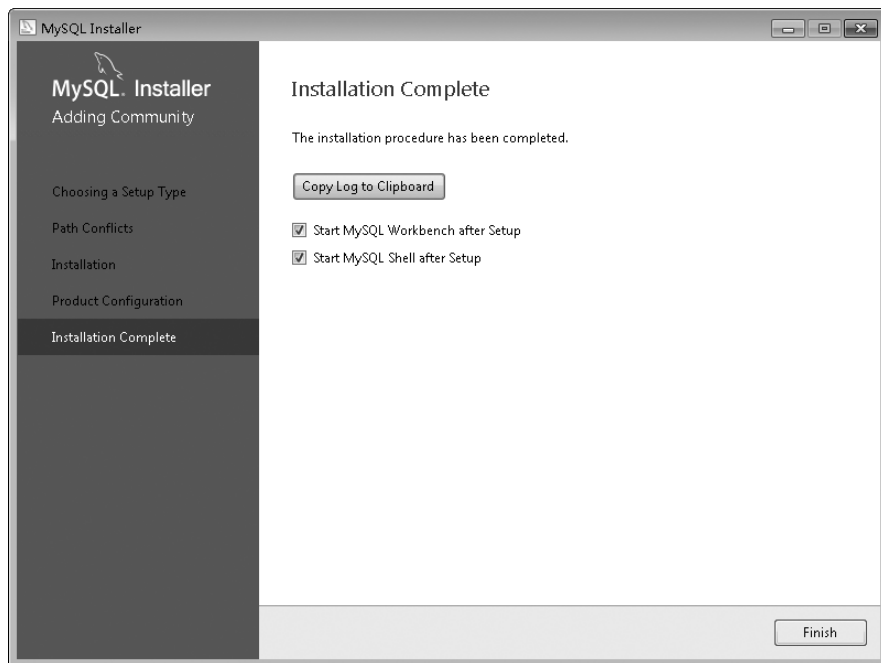


图 1-20 完成安装





### 1.3.2 MySQL 基本操作

成功安装 MySQL 后，可以通过系统服务和命令提示符对 MySQL 服务器进行启动、暂停、停止和重新启动等操作。

#### 1 通过系统服务操作 MySQL 服务器

在 Windows 的系统服务中可以启动、暂停、停止和重新启动 MySQL 服务器，具体方法如下。

执行“开始” | “控制面板” | “管理工具” | “服务”命令，打开 Windows 服务管理器窗口。在该窗口中选择名称为“MySQL57”的服务，可以查看该项服务的状态。在窗口左边单击相应的选项，可实现对 MySQL 服务器的操作。也可以在该项上单击鼠标右键，在弹出的快捷菜单中选择相应的命令，实现对 MySQL 服务的启动、暂停、停止和重新启动等操作，如图 1-21 所示。

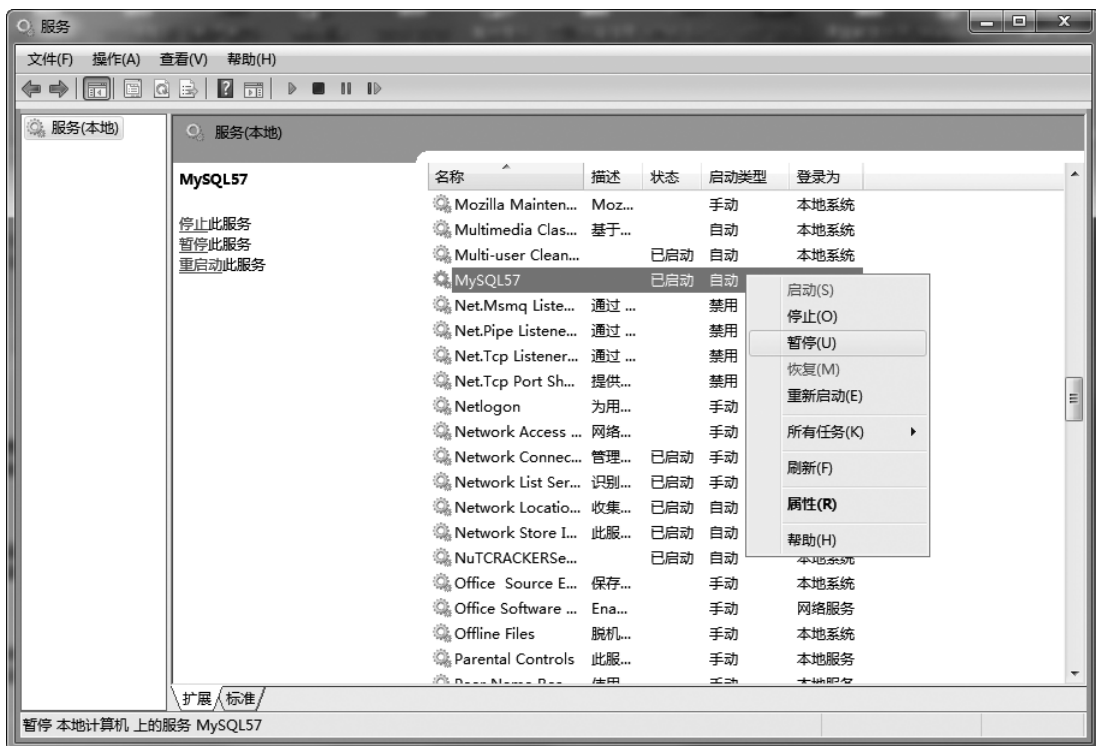


图 1-21 在系统服务中操作 MySQL 服务器

#### 2 通过命令提示符操作 MySQL 服务器

通过在命令提示符窗口中输入命令，也可以实现对 MySQL 服务器的操作，具体方法如下。

单击“开始”菜单，在弹出的输入框中输入“cmd”命令，按“Enter”键，打开命令提示符窗口。在该窗口中输入以下命令。

```
net start mysql57
```







按“Enter”键，即可启动 MySQL 服务器，如图 1-22 所示。



图 1-22 通过命令提示符操作 MySQL 服务器

其中，“mysql57”是 MySQL 服务器的名称，读者应根据各自的设置作相应修改。

```
net stop mysql57
```

按“Enter”键，即可停止 MySQL 服务器。

在命令提示符窗口中输入以下命令。

### 1.3.3 Command Line Client

MySQL 服务器启动后，就可以通过 MySQL 的 Command Line Client 程序来操作 MySQL 数据了，具体步骤如下。

(1) 执行“开始”|“所有程序”|“MySQL”|“MySQL Server 5.7”|“MySQL 5.7 Command Line Client”命令，打开 MySQL 客户端命令行窗口，如图 1-23 所示。



图 1-23 MySQL 客户端命令行窗口





(2) 在该窗口中输入 root 用户的登录密码（该密码为安装过程中配置的密码），登录到 MySQL 服务器，如图 1-24 所示。

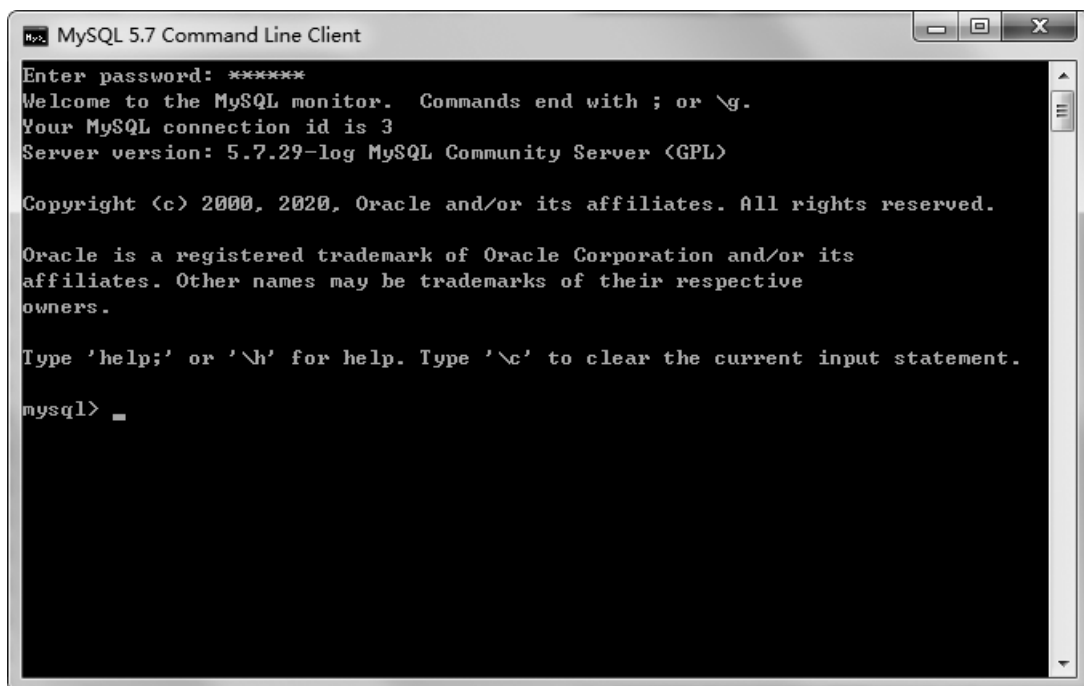


图 1-24 登录到 MySQL 服务器

(3) 输入操作的 SQL 语句，即可实现对数据库的各项操作，如图 1-25 所示。

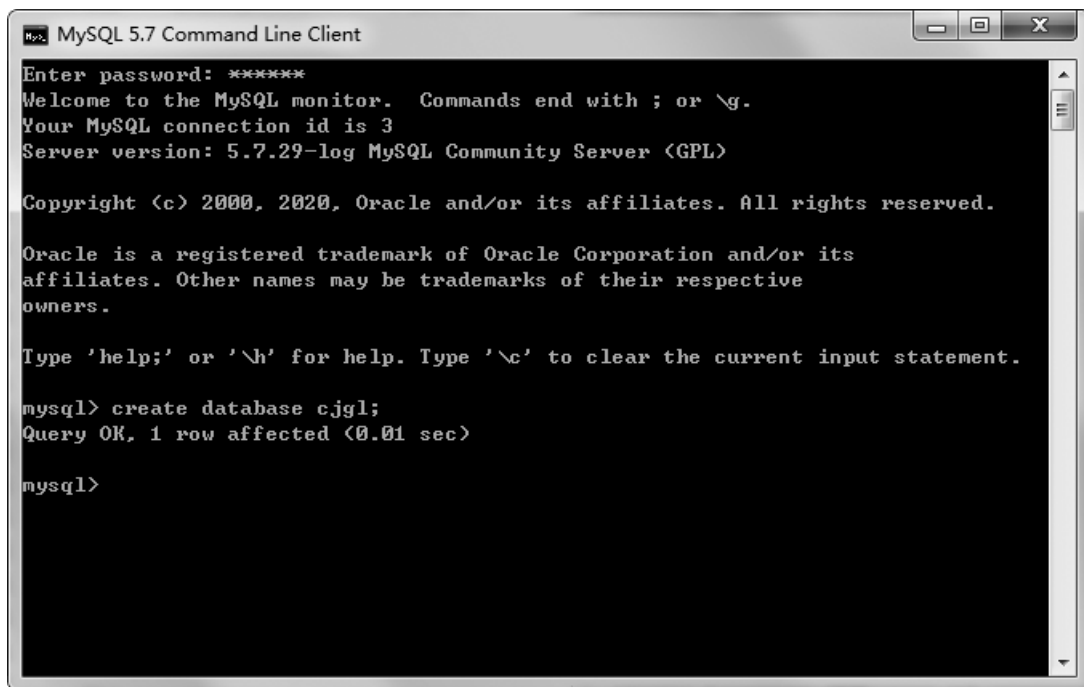


图 1-25 操作数据库





### 1.3.4 MySQL 图形化界面管理工具

MySQL 本身没有提供非常方便的图形化界面管理工具，操作都通过在客户端命令行窗口中进行，需要对操作命令非常熟悉，对于初学者来说增加了学习难度。在学习过程中，可以使用第三方提供的图形化界面管理工具。常用的图形化界面管理工具包括 Navicat、SQLyog、phpMyAdmin、Workbench 等。

下面将详细介绍 Navicat 的基本操作，其余图形化界面管理工具的使用方法与其类似，读者可以自行研究。

Navicat 是目前开发者用得较多的一款 MySQL 图形化界面管理工具。其界面简洁，功能强大，与 Microsoft 公司的 SQL Server 管理器很像，简单易学，支持中文，提供免费版本。

使用 Navicat 操作 MySQL 数据库的基本步骤如下。

(1) 打开 Navicat，如图 1-26 所示。

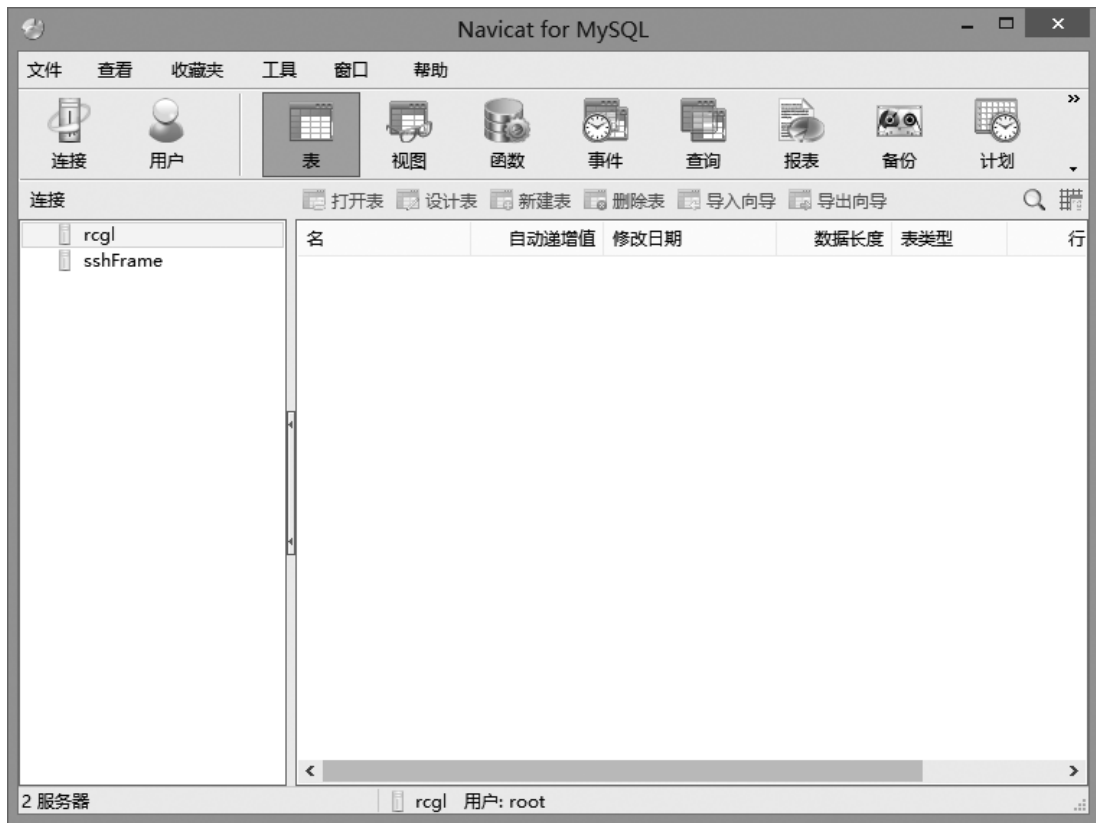


图 1-26 Navicat 窗口

(2) 单击“连接”按钮，在弹出的“新建连接”对话框中输入“连接名”“主机名或 IP 地址”“端口”“用户名”和“密码”等信息，单击“连接测试”按钮。若弹出“连接成功”对话框，则表明连接信息正确，如图 1-27 所示。





图 1-27 新建连接

(3) 单击该对话框中的“确定”按钮，再单击“新建连接”对话框中的“确定”按钮，返回 Navicat 窗口。在窗口左侧显示出刚才建立的连接。双击该连接，即可查看当前 MySQL 服务器中已有的数据库，如图 1-28 所示。

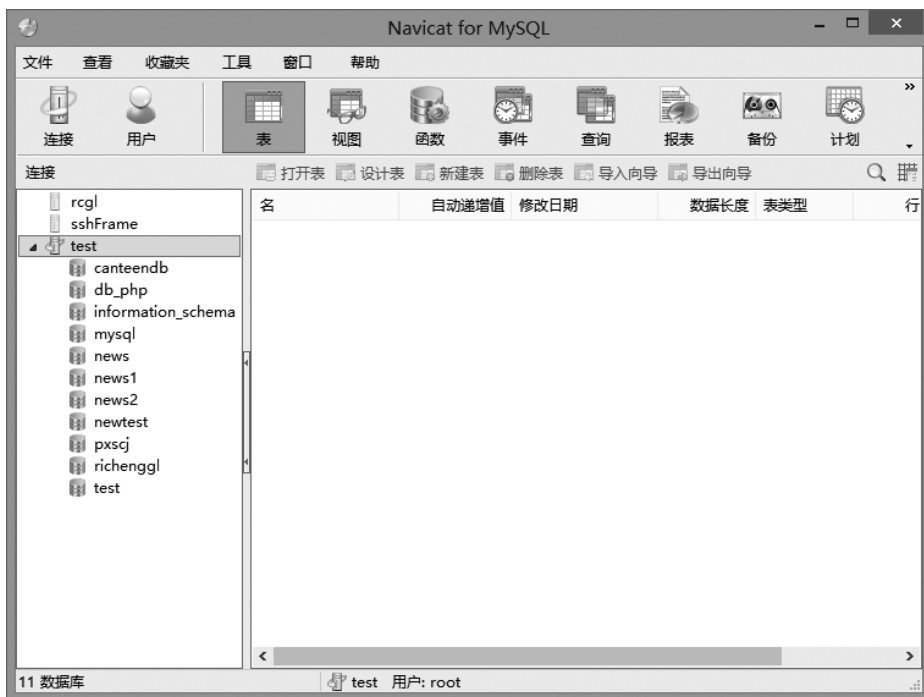


图 1-28 查看连接中的数据库



(4) 在连接名称“test”上单击鼠标右键，在弹出的快捷菜单中选择“新建数据库”命令，弹出“新建数据库”对话框。在该对话框中设置“数据库名”“字符集”和“排序规则”信息，如图 1-29 所示。

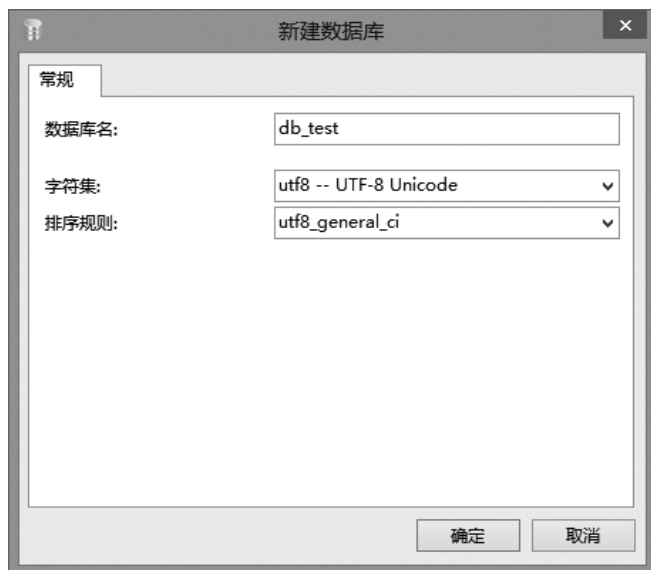


图 1-29 新建数据库

(5) 单击“确定”按钮，返回 Navicat 窗口。在“test”连接下可以看到新建的数据库“db\_test”。双击“db\_test”，打开该数据库。在表上单击鼠标右键，在弹出的快捷菜单中选择“新建表”命令，打开表设计窗口。在该窗口中设置表结构信息，如图 1-30 所示。

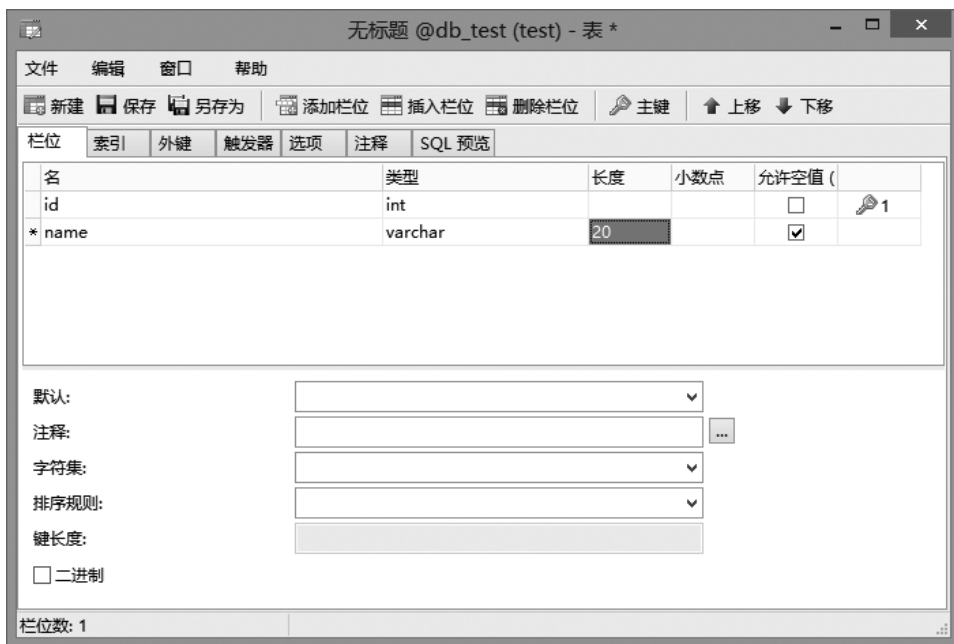


图 1-30 设计数据表





(6) 单击“保存”按钮，弹出“表名”对话框。在该对话框中输入表名信息，单击“确定”按钮，返回 Navicat 窗口。在窗口中双击“db\_test”下的“表”，即可查看刚才创建的数据表“tb\_test”。双击“tb\_test”，打开表，在该表中输入基本数据，如图 1-31 所示。



图 1-31 表数据编辑

(7) 在 Navicat 窗口中对数据库和数据表进行其他操作。由于图形化界面操作相对比较简单，此处不再详细介绍，读者可以在学习的过程中自行研究。

## 本章小结

本章主要介绍了数据库的基础知识，包括数据库相关的基本概念、数据模型及关系数据库、MySQL 的安装与配置、MySQL 服务器的启动与停止等操作、MySQL 客户端命令行与常用图形化界面操作工具的使用等。

掌握好本章所介绍的基本概念和基本操作是学习 MySQL 的基础。

## 课后习题

1. 简述数据与数据库的概念。
2. 简述数据库系统与数据库管理系统的概念。
3. 简述数据模型与关系数据模型的概念。
4. 简述关系的完整性约束。
5. 安装和配置 MySQL 服务器。
6. 安装 Navicat，创建数据库和数据表，并在数据表中录入基本数据。

