



认识数据库

项目导读

在大数据时代，数据已成为一种宝贵的资源，其价值和影响力日益凸显。数据库作为数据资源的主要存储工具，其重要性不言而喻。数据库不仅是一个简单的数据存储仓库，更是一个能够将数据按照特定的规律、结构和方式进行组织和存储的系统。数据库的设计、建立和维护都需要遵循一系列严格的规则和原则，以确保数据的完整性、准确性和高效性。为充分利用这一资源并发挥数据库的最大效用，深入了解数据库技术至关重要。而要实现这一目标，需要先从认识数据库开始。

学习目标

- 掌握数据库的基本概念与原理。
- 熟悉结构化查询语言（SQL）的核心特性。
- 熟练安装与配置 MySQL 数据库系统。
- 熟悉与 MySQL 服务器建立连接、启动并运行的多种途径。

任务 1.1 数据库概述

任务描述

- (1) 掌握数据库的基本概念。
- (2) 了解数据库的发展历程。
- (3) 了解数据模型的概念。
- (4) 掌握关系数据库的基础知识。



任务目标

- (1) 理解和掌握数据库的基本概念和基本理论。
- (2) 通过学习数据库的基本概念, 了解数据安全的意义, 培养学生保护数据安全、防范潜在风险的能力, 增强其社会责任感和职业道德意识。



知识储备

知识点1 数据库的基本概念

1. 数据

数据是客观世界被记录下来的可辨识物理符号, 用于反映真实情况。如今, 数据的定义已不再仅限于数值型数据, 而是扩展到文字、声音、图像等一切计算机能接收并处理的符号。

2. 数据库

数据库 (database, DB) 是指按照数据结构组织、存储和管理数据的仓库。它是一个长期存储在计算机内、有组织、可共享、可统一管理的大量数据的集合。数据库能以最佳方式、最少重复, 为多种应用服务。

3. 数据库管理系统

数据库管理系统 (database management system, DBMS) 是一种操纵和管理数据库的大型软件, 用于建立、使用和维护数据库。它对数据库进行统一的管理和控制, 以保证数据库的安全性和完整性。用户通过 DBMS 访问数据库中的数据, 数据库管理员也通过 DBMS 进行数据库的维护。DBMS 提供数据定义语言 (data definition language, DDL) 和数据操纵语言 (data manipulation language, DML), 供用户定义数据库的模式结构与权限约束, 实现对数据的追加、删除等操作。

知识点2 数据库发展史

1. 人工管理阶段

20 世纪 50 年代中期之前, 数据量相对较小, 数据主要由人工管理, 没有专门的数据库管理系统, 计算机主要用于科学计算。数据存储于磁带、卡片等外部存储设备上, 且主要以批处理方式进行处理。数据由应用程序自己管理, 无法保存, 且数据不共享, 不具有独立性, 冗余大。这个阶段的数据管理效率较低, 缺乏统一的数据结构和标准。

2. 文件系统阶段

20 世纪 50 年代后期至 60 年代中期, 随着计算机硬件和软件技术的发展, 计算机开始被用于信息管理。数据的存储、检索和维护变得日益重要, 因此数据结构和数据管

理技术迅速发展起来。在这个阶段，数据可以长期保存，由文件系统来管理，并以文件的形式存储在计算机上。虽然文件系统提供了一定的数据管理功能，但仍然存在数据共享性差、数据冗余大、数据独立性差等问题。

3. 数据库系统阶段

20 世纪 60 年代后期开始，数据库系统逐渐崭露头角，克服了文件系统的缺陷，提供了对数据更高级、更有效的管理。在这个阶段，数据库管理系统的出现使得程序和数据的联系得以通过 DBMS 实现。数据库系统提供了数据结构化、数据共享、数据独立和数据安全等核心特性，大大提高了数据的管理效率和使用便利性。随着数据库技术的发展，关系数据库、面向对象数据库和非结构化数据库等不同类型的数据库系统相继出现，满足了不同领域和应用场景的需求。

知识点3 数据库系统

数据库系统 (database system, DBS) 是一个集成系统，主要由数据库、数据库管理系统和相关应用程序组成。它的主要目标是提供数据存储、管理和访问的功能，以支持各种数据处理和应用程序。

此外，数据库系统还包括相关的应用程序和数据库管理员。应用程序用于与数据库进行交互，实现数据的增、删、改、查等操作。而数据库管理员 (database administrator, DBA) 是负责管理数据库系统的专业人员，负责数据库的设计、安装、配置、维护和优化等工作，以确保数据库的高效、安全和可靠运行。图 1.1 所示为数据库系统的结构。

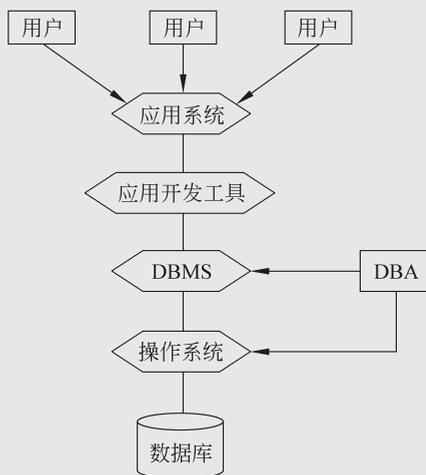


图 1.1 数据库系统的结构

知识点4 数据模型

数据模型 (data model) 是对现实世界数据特征的抽象表示,描述了系统的静态特征、动态行为和约束条件,并为数据库系统的信息表示与操作提供了抽象的框架。数据模型主要包含数据结构、数据操作和数据约束三大部分。

数据模型是数据库设计中的一个重要概念,作为对现实世界的抽象工具,为数据库提供了一个统一的、结构化的信息表示和操作框架。数据模型可以根据其抽象层次的不同,被分为三类:最上面的概念模型、中间的逻辑模型和最底层的物理模型。

在数据库设计过程中,从概念模型到逻辑模型的转换主要由数据库设计师完成,他们需要根据实际需求和分析结果,设计出合适的逻辑模型。而从逻辑模型到物理模型的转换则主要由数据库管理系统完成,数据库管理系统会根据逻辑模型的要求,自动选择合适的物理存储和操作方式,以实现数据的高效管理。



1. 概念模型

概念模型是数据库设计中的一个核心概念，它是对现实世界的事物及其关系的抽象表示，也称信息模型。这个概念模型的核心在于它提供了一个高层次的、与特定数据库管理系统无关的视角，使设计师能够专注于理解业务需求和数据的本质。其主要目的是利用一系列的概念和关系描述现实世界的事物以及它们之间的相互联系。在概念模型中，通常使用实体和关系描述数据。实体代表现实世界中的具体事物，如人、地点或事物本身，而关系则描述了这些实体之间的连接和互动。这种模型通常通过图形化的方式呈现，如实体—关系图（entity relationship diagram, E-R 图），其中实体以矩形表示，关系以菱形表示，而连接实体和关系的线则显示了数据流动的方向。概念模型的主要优势在于其简单性和通用性，它能够帮助设计师更好地理解业务流程和数据流动，为后续的数据库设计提供坚实的基础。

2. 逻辑模型

逻辑模型是面向数据库的逻辑结构，提供了表示和组织数据的方法，使得数据可以在数据库中以一种有序、结构化的方式进行存储和查询。数据库系统中常用的逻辑模型有三种：层次模型、网状模型和关系模型，其中关系模型应用最为广泛。

1) 层次模型

层次模型是一种树状结构，每个节点代表一个记录类型，节点间的连线表示记录间的从属关系。这种模型直观且易于理解，特别适用于表示具有明显层次结构的数据，如组织架构、家族关系。然而，层次模型在处理多对多关系时较为困难，且查询效率较低。图 1.2 所示为一个系的层次模型。

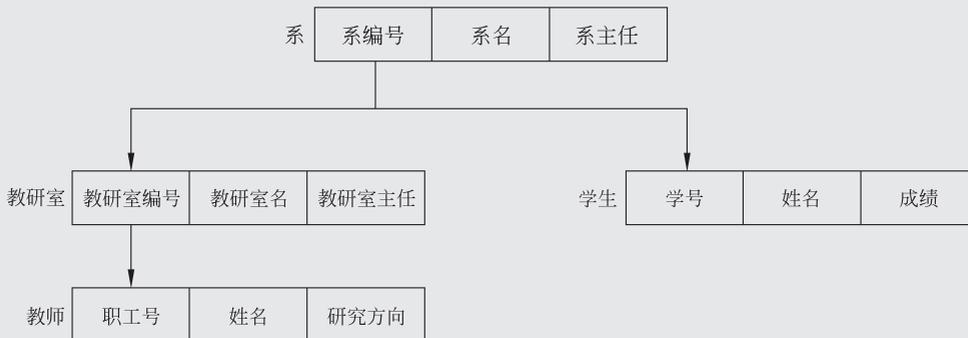


图 1.2 层次模型示例

2) 网状模型

网状模型在层次模型的基础上增加了灵活性，允许节点拥有多个父节点和子节点，从而能够表示更复杂的数据关系，尤其是多对多关系。这种模型在图书馆系统或社交网络等场景中非常有用。但网状模型的查询效率较低，且结构相对复杂，管理和维护起来较困难。图 1.3 所示为网状模型示例。

3) 关系模型

关系模型使用二维表组织数据，每个表由行和列组成，表之间的关系通过“键”建立。

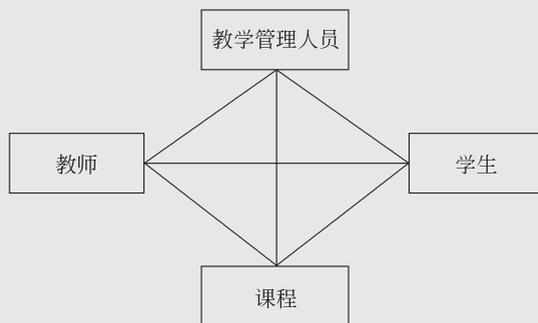


图 1.3 网状模型示例

这种模型提供了规范的数据结构和丰富的查询语言，如结构化查询语言（structured query language, SQL），使数据的查询和管理变得高效且统一。关系模型广泛应用于各种数据库系统中，尽管对于某些特定类型的数据（如图形、图像）的表示不够直观，但它仍然是当前数据库设计的主流选择。表 1.1 所示为关系模型示例。

表 1.1 关系模型示例

c_no	c_name	credit	cr_hours	semester
11001	计算机基础	3	48	1
11002	Office 应用	2	32	2
11003	Java 语言程序设计	4	64	3
21001	会计学	3	48	2
21002	就业指导	2	32	2
11005	数据库	4	64	4
11006	操作系统	4	64	5

3. 物理模型

物理模型是数据模型的最底层，是对数据的最底层抽象，主要描述了数据在计算机系统内部的表示方式和存取方法。它涉及数据的物理存储、索引结构、磁盘操作等细节，是面向计算机系统的。物理模型的实现主要由数据库管理系统完成，数据库管理系统会根据逻辑模型的要求，选择合适的物理存储结构和操作方式，以实现数据的高效访问和管理。

知识点5 关系型数据库

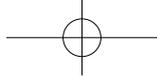
基于关系模型建立的数据库称为关系型数据库，是由若干张二维表组成的集合，广泛应用于企业级应用、电子商务、金融系统等领域，是目前最成熟、最稳定的数据库技术之一。常见的关系型数据库管理系统有 Oracle、MySQL、SQL Server、PostgreSQL 等。



微课：理解关系型数据库

1. 关系

关系是满足关系模型基本性质的二维表格，一个关系就是一张二维表，是关系型数据库存储数据的基本单位。每个关系都有一个唯一的名称（或称为表名），用于标识



该关系。例如，表 1.2 所示为 student 表。

表 1.2 student 表

s_no	s_name	sex	birthday	nat_place	nation	class_no
20221090501	汪燕	女	2003/12/9	江西	汉	JSJ2201
20221090502	李强	男	2004/7/10	江西	汉	JSJ2201
20221090803	陈海	男	2004/3/27	江西	汉	JSJ2202
20221090602	程鸿	男	2003/11/12	广西	壮	JSJ2202
20221090701	谢婷	女	2004/8/13	河南	汉	JD2301
20221090976	王菲	女	2003/12/29	山西	汉	KJ2201
20232090807	陆优	女	2004/8/15	安徽	汉	KJ2302

2. 属性

属性是关系中的列，也叫作字段。每个属性都有一个名字，称为属性名。属性表示实体的一个特征或数据项，具有相同的数据类型。例如，student 表中，属性包括学号 (s_no)、姓名 (s_name)、性别 (sex) 等。

3. 元组

元组是关系中的一行数据，也称记录。它是关系的一个具体实例，即实体的一个具体属性值集合。例如表 1.2 中，(20221090501, 汪燕, 女, 2003/12/9, 江西, 汉, JSJ2201) 为一条记录。在关系型数据库中，元组是数据的基本操作单位，如插入、更新和删除等操作都是以元组为单位的。

4. 分量

元组中的一个字段值称为一个分量。关系数据库要求每一个分量都必须是不可分的数据项，即不允许表中还有表。

5. 值域

值域是指属性可以取值的范围或集合，它定义了属性可以拥有的合法值。例如，性别 (sex) 属性的值域为“男”和“女”。

6. 关系模式

对关系的描述称为关系模式，一个关系模式对应一个关系的结构。关系模式的一般结构如下：

关系名 (属性名 1, 属性名 2, 属性名 3, ..., 属性名 n)

例如，表 1.2 中的关系模式为 student (s_no, s_name, sex, birthday, nat_place, nation, class_no)。

7. 主关键字

主关键字是关系型数据库中的一个重要概念，也称主键或主码，是指在一个关系 (即表) 中能够唯一标识每一个元组的一个或一组属性。其作用是确保数据的唯一性和完整性，即在一个关系中，主关键字的值必须是唯一的，不允许有重复值。同时，主关键字的值也不能为 NULL，即不能为空值。

例如，在 `student` 表中，学生的学号 (`s_no`) 通常被设为主关键字，因为每个学生的学号都是唯一的，可以唯一标识一个学生。这样，通过学号就可以准确地查询到对应的学生信息，保证了数据的唯一性和准确性。

8. 外部关键字

外部关键字是关系型数据库中另一个重要的概念，也称外键。它是指在一个关系（即表）中，一个或一组属性不是该关系的主关键字，而是另一个关系的主关键字。外部关键字用于建立两个关系之间的联系，实现数据的关联和引用完整性。

假设除 `student` 表外，还有一张 `class` 表，其主关键字是班级编号 (`class_no`)。这样，通过班级编号这个外部关键字，可以将 `student` 表和 `class` 表关联起来，以查询每个学生所在的班级信息。

知识点6 常见的数据库管理系统

数据库作为关键的基础设施软件，在企业架构中扮演着不可或缺且难以替代的角色。事实上，高达 90% 的企业业务应用系统都是基于数据库构建的。即便在大数据和云计算的时代浪潮中，数据库服务仍然是云计算行业巨头竞相争夺的核心领域。随着技术的进步和应用场景的不断演变，数据库领域已经从关系数据库的“一统天下”逐渐转变为多种类型数据库的“百花齐放”。这些数据库根据所使用的语言可以划分为三大类：SQL、NoSQL 和 NewSQL。

SQL 类数据库、NoSQL 类数据库和 NewSQL 类数据库之间的区别主要体现在数据结构、查询语言、一致性模型、应用场景以及扩展性等方面。

1. SQL 类数据库

数据结构：SQL 类数据库使用二维表存储数据，每个二维表都由行和列组成，行代表元组或记录，列代表属性或字段。

查询语言：通过结构化查询语言（SQL）进行数据操作，包括增、删、改、查等。

一致性模型：遵循 ACID 原则（原子性、一致性、隔离性、持久性），确保数据在事务处理中的完整性和一致性。

应用场景：适用于需要复杂查询、事务处理和数据一致性的场景，如企业资源规划（enterprise resource planning, ERP）、客户关系管理（customer relationship management, CRM）等。

扩展性：传统的 SQL 类数据库在垂直扩展方面表现较好，但在水平扩展方面表现较差。

2. NoSQL 类数据库

数据结构：NoSQL 类数据库采用非结构化的数据模型，如键值对、文档、列族、图形等，适用于存储多样化的数据。

查询语言：不依赖于 SQL，而是使用各自的查询语言或 API 进行数据操作。

一致性模型：通常采用最终一致性模型，放宽了 ACID 原则的要求，以提高性能和可用性。



应用场景：适用于需要高并发读写、低延迟、水平扩展以及非结构化数据存储的场景，如社交媒体、实时分析、物联网等。

扩展性：NoSQL 类数据库在水平扩展方面表现出色，可以通过增加节点提高系统的处理能力和存储容量。

3. NewSQL 类数据库

数据结构：NewSQL 类数据库结合了关系型数据库和非关系型数据库的特点，支持结构化和非结构化数据的存储。

查询语言：提供类似 SQL 的查询语言，使用户能够方便地进行数据查询和操作。

一致性模型：在保持高性能和可扩展性的同时，尽量满足 ACID 原则的要求，并提供一定程度的数据一致性保证。

应用场景：适用于需要大规模并发访问、高性能、数据一致性以及支持复杂查询的场景，如在线金融、电商的大数据分析等。

扩展性：NewSQL 类数据库在水平扩展方面表现出色，可以通过分布式架构实现高性能和可扩展性。

总的来说，SQL 类数据库适用于需要复杂查询和事务处理的场景；NoSQL 类数据库适用于需要高并发读写、低延迟和非结构化数据存储的场景；而 NewSQL 类数据库则结合了两者的优点，可提供高性能、可扩展性和一定程度的数据一致性保证。

任务 1.2 MySQL 的安装与配置



任务描述

- (1) 了解 MySQL 数据库的优势。
- (2) 掌握在 Windows 平台安装和配置 MySQL 8.0。
- (3) 掌握 MySQL 常用的图形管理工具。
- (4) 掌握常见的 MySQL 功能。



任务目标

- (1) 掌握 MySQL 数据库的安装与配置方法。
- (2) 通过学习安装与配置 MySQL 数据库，使读者体会到严谨与细致的重要性。数据库的安装涉及多个步骤，包括配置环境变量、设置权限、创建数据库等，每一步都需要认真对待，否则可能导致安装失败或数据库运行不稳定。这种严谨与细致的态度，可以使其在未来的学习和工作中，对待每一项任务都能够认真、负责，追求高质量和高效率。



微课：MySQL 的安装与配置

知识储备

知识点1 MySQL的优势

MySQL 是一个开源的关系型数据库管理系统，由瑞典公司 MySQL AB 开发，后被甲骨文（Oracle）公司收购。由于其性能稳定、易于使用、跨平台兼容性强以及开源免费等特点，MySQL 成了许多 Web 应用程序的首选数据库。

MySQL 数据库主要有以下八个特性。

（1）**开源与免费**：是一个开源项目，用户可以免费地使用、修改和分发其源代码。这使得 MySQL 在开发者社区中非常受欢迎，特别是在需要低成本解决方案的场合。

（2）**稳定可靠**：拥有成熟稳定的架构，支持高并发访问和大量数据处理。通过持续更新和修复，可确保在各种应用场景下的稳定性和可靠性。

（3）**跨平台兼容性**：支持多种操作系统，包括 Windows、Linux、macOS 等，用户可以根据需要选择合适的平台部署 MySQL 数据库。

（4）**高性能**：通过优化查询算法、使用索引、缓存机制等技术手段，提高查询速度和数据处理能力。同时，支持多线程并发访问，能够充分利用多核 CPU 资源。

（5）**易于使用**：提供了丰富的 SQL 接口和图形化管理工具（如 phpMyAdmin、MySQL Workbench 等），使得数据库的管理和维护变得简单直观。

（6）**可扩展性**：支持多种存储引擎，如 InnoDB、MyISAM、Memory 等，用户可以根据需求选择合适的存储引擎以满足不同的性能要求。此外，MySQL 还支持分区、复制、集群等技术，以实现数据库的水平扩展和负载均衡。

（7）**安全性**：提供了访问控制、数据加密、审计日志等安全特性，确保数据库的安全性和数据完整性。用户可以通过配置用户权限、使用 SSL 连接等方式提高数据库的安全性。

（8）**社区支持**：由于 MySQL 是一个开源项目，拥有庞大的用户社区和丰富的文档资源，这意味着用户可以轻松找到解决问题的方案、分享经验以及获取技术支持。

MySQL 广泛应用于各个领域，包括但不限于以下四个领域。

（1）**Web 开发**：MySQL 是许多 Web 应用程序的默认数据库选择，如内容管理系统（CMS）、电子商务网站、社交媒体平台等。

（2）**数据分析**：可以用于存储和查询大量数据，支持数据分析和数据挖掘任务。

（3）**嵌入式系统开发**：其轻量级版本（如 MySQL Embedded）适用于嵌入式系统和设备，如智能家居、物联网设备等。

（4）**企业应用开发**：也适用于企业级应用，如客户关系管理、企业资源规划等应用开发。

知识点2 MySQL图形化管理工具

MySQL 图形化管理工具对于数据库的日常操作和管理至关重要，它们通过直观的用户界面，让数据库管理员和开发者能够更轻松、高效地完成各项任务。在众多图形化



管理工具中,MySQL Workbench、phpMyAdmin、Navicat for MySQL 等工具备受用户青睐。

1. MySQL Workbench

MySQL Workbench 是一款功能全面的数据库管理工具,特别适用于那些需要全面管理和维护 MySQL 数据库的用户。作为官方出品的工具,MySQL Workbench 不仅支持 MySQL 5.0 及以上版本,还提供了丰富的数据库设计、数据建模、SQL 开发以及数据库管理等功能。虽然 4.x 版本以下的用户只能使用社区版,但这也足以满足大部分用户的需求。社区版提供了基本的数据库管理功能,并且完全免费,非常适合个人开发者和小型企业使用。对于需要更多高级功能的企业用户,商业版提供了更加全面的支持和服务。

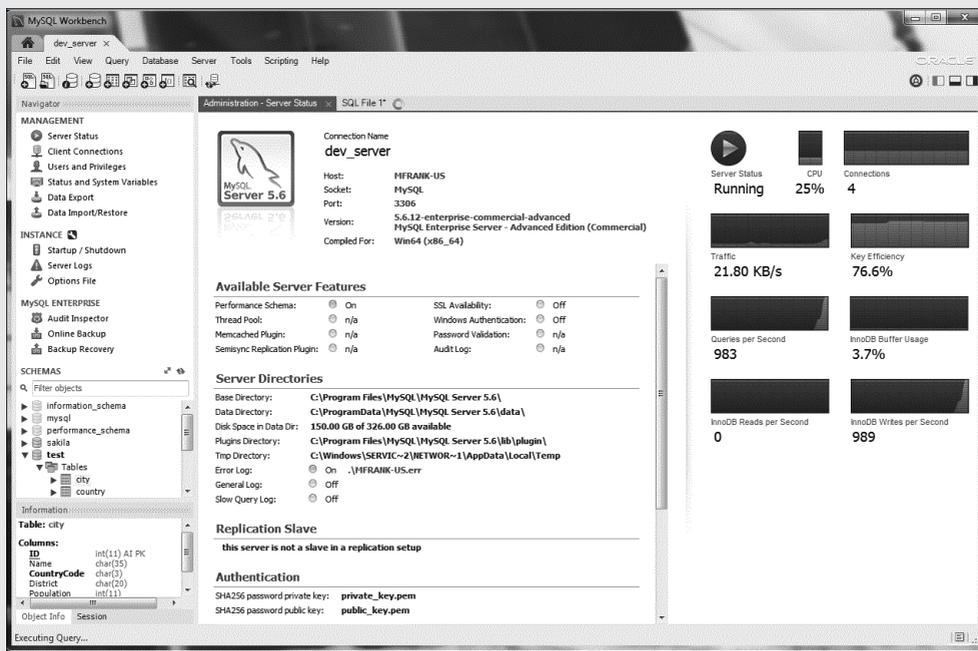


图 1.4 MySQL Workbench 图形化管理工具界面

2. phpMyAdmin

phpMyAdmin 是一个基于 Web 的 MySQL 数据库管理工具,通过直观的界面,用户可以轻松管理数据库、执行 SQL 查询、导入导出数据等,通过简单的配置即可在 Web 服务器上使用。无论是数据库新手还是专业开发者,phpMyAdmin 都提供了强大的功能和易用性,使得数据库管理变得简单高效。图 1.5 所示为 phpMyAdmin 图形化管理工具界面。

3. Navicat for MySQL

Navicat 是一套高效、便捷的数据库管理工具,旨在简化数据库的管理流程并降低系统运维成本,其设计理念充分满足了数据库管理员、开发人员以及中小企业的实际需求。特别是 Navicat for MySQL,它与 MySQL 数据库服务器完美兼容,通过直观易用的图形用户界面 (GUI) 支持 MySQL 的大多数最新功能,同时支持中文操作界面,这使得用户可以安全、快速地创建、组织、访问和共享信息。图 1.6 所示为 Navicat for MySQL 图形化管理工具界面。