



目录

第 1 章 认识数据库	1
1.1 数据库的基本概念	2
1.2 数据库技术的发展	6
1.3 数据模型	7
1.4 三级模式和二级映像	10
1.5 关系数据库	12
1.6 常用数据库管理系统	16
任务实践.....	17
思考与练习.....	18
第 2 章 MySQL 的安装与配置	20
2.1 MySQL 的获取途径	21
2.2 MySQL 的安装与配置	23
2.3 关闭与启动 MySQL 服务	27
2.4 用户登录与密码设置	27
2.5 基本功能测试	28
2.6 MySQL 客户端的相关命令	28
任务实践.....	29
思考与练习.....	29
第 3 章 创建和管理数据库	31
3.1 创建数据库	32
3.2 查看数据库	33
3.3 修改数据库	36
3.4 删除数据库	37



3.5 选择数据库	38
任务实践	39
思考与练习	40
第4章 创建和管理数据表	41
4.1 数据类型	43
4.2 创建数据表	52
4.3 查看数据表	53
4.4 修改数据表	56
4.5 删除数据表	63
4.6 数据完整性	64
任务实践	74
思考与练习	76
第5章 数据操作	79
5.1 插入数据	80
5.2 修改数据	85
5.3 删除数据	87
任务实践	89
思考与练习	89
第6章 单表数据查询	91
6.1 基础查询	92
6.2 条件查询	96
6.3 聚合函数	106
6.4 分组查询	109
6.5 排序查询	111
任务实践	112
思考与练习	113
第7章 多表数据查询	115
7.1 多表连接查询	116
7.2 子查询	121
任务实践	127

思考与练习127

第 8 章 索引 129

8.1 索引简介130

8.2 索引的分类130

8.3 索引的优缺点132

8.4 创建索引133

8.5 查看索引136

8.6 删除索引138

任务实践140

思考与练习140

第 9 章 视图 142

9.1 视图简介143

9.2 视图的优点143

9.3 创建视图144

9.4 查看视图147

9.5 修改视图148

9.6 删除视图149

9.7 利用视图操作数据150

任务实践152

思考与练习153

第 10 章 数据库编程 154

10.1 运算符155

10.2 变量159

10.3 函数164

10.4 存储过程180

10.5 流程控制语句190

10.6 触发器200

10.7 游标204

任务实践205

思考与练习207

第 11 章 安全性管理	209
11.1 创建用户	211
11.2 修改用户	212
11.3 删除用户	214
11.4 用户授权	215
11.5 删除用户授权	218
任务实践	219
思考与练习	221
第 12 章 事务管理	223
12.1 事务的特性	224
12.2 事务管理操作	224
12.3 事务隔离级别	228
任务实践	230
思考与练习	233
第 13 章 数据库备份与还原	235
13.1 数据库备份	236
13.2 数据库恢复	241
13.3 数据库迁移	245
任务实践	246
思考与练习	247
第 14 章 图书管理系统的设计与开发	249
14.1 系统数据流图	250
14.2 系统功能设计	251
14.3 数据库设计	251
14.4 系统开发与实现	254
参考文献	264

第 1 章

认识数据库

学习成果目标

- 1 理解数据库、数据库管理系统、数据库系统的基本概念。
- 2 了解数据库技术的发展阶段及其特点。
- 3 理解数据库的三级模式结构和二级映像。
- 4 了解数据模型的概念和分类。
- 5 熟悉关系模型的特点，掌握关系数据库的基本知识。
- 6 掌握 ER 模型到关系模型的转换方法和规则。
- 7 熟悉常用的数据库管理系统。
- 8 提升信息素养，激发数据库技术学习兴趣。

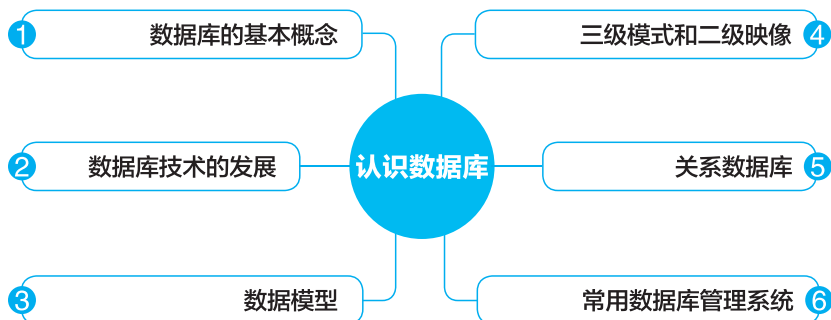
学习任务

随着信息化技术的不断发展，数字化校园建设对打造高水平大学具有广泛深远的意义。为此，学校准备开发一个“教务管理系统”，以提升学校日常教务的工作效率和管理水平。“教务管理系统”可以实现学生信息管理、课程信息管理、教师信息管理、班级管理、成绩信息管理，同时提供在线选课、教学任务安排、综合信息查询、系统管理等功能。请结合上面所描述的功能场景，抽象“教务管理系统”的概念模型，绘制 ER（Entity Relationship，实体联系）图，并将概念模型转换为逻辑数据模型，为数据库的创建做好准备。

学习内容

数据库技术是计算机应用领域中非常重要的技术，是软件技术的一个重要分支。为了能深刻理解和学习 MySQL，首先我们来介绍数据库的基础知识，学习数据库的基础原理。

本章的内容结构如下：



1.1 数据库的基本概念

1.1.1 数据

数据是指对客观事件进行记录并可以鉴别的符号，是对客观事物的性质、状态以及相互关系等进行记载的物理符号或这些物理符号的组合。数据是可识别的、抽象的符号。

数据不仅指狭义上的数字，还可以是具有一定意义的文字、字母、数字符号的组合、图形、图像、视频、音频等，也是客观事物的属性、数量、位置及其相互关系的抽象表示。例如，“0、1、2…”“阴、雨、下降、气温”“学生的档案记录、货物的运输情况”等都是数据。数据经过加工后就成了信息。

在日常生活中，人们直接用语言来描述事物，例如在描述一门课程时，人们会用“Java 编程——入门篇，是由编程达人郎老师讲授的编程语言课，本课程面向青少年编程爱好者，以浅显易懂的语言，带领大家逐步进入计算机编程的世界”。在计算机科学中，数据是所有能输入计算机并被计算机程序处理的符号介质的总称，是用于输入电子计算机进行处理，具有一定意义的数字、字母、符号和模拟量等的通称。例如，将课程事务抽象为数据项记录（课程名称、授课老师、类别、学生对象、课程简介），课程事务的一条数据可以描述为（Java 编程——入门篇，郎老师，编程语言，高职学生，课程以浅显易懂的语言带领大家进入计算机编程世界）。计算机存储和处理的对象十分广泛，数据也随之变得越来越复杂。

数据和信息是有区别的。数据是指对某一事物定性或定量描述的原始资料，包括数字、文字、符号、图形、图像以及它们能够转换成的数据形式。信息是关于现实世界的事实的知识，是数据、消息中所包含的意义。数据与信息的关系如图 1-1 所示，主要表现为信息与数据是不可分离的。数据是客观对象的表示，是信息的载体，而信息则是数据内涵的意义。即信息是有用的数据，数据是信息的表现形式。

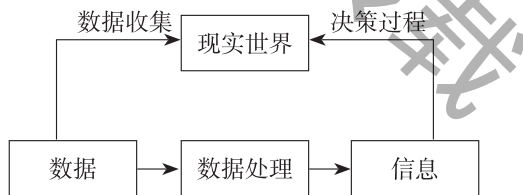


图 1-1 数据与信息的关系

数据与信息又有着本质上的区别，主要表现为以下三点：

(1) 数据是记录下来的某种可以识别的符号，可因载体的不同表现出不同的形式，而信息不随载体的改变而改变。

(2) 信息可以离开信息系统而独立存在，也可以离开信息系统的各个组成和阶段而独立存在；而数据的格式往往与计算机系统有关，并随载荷它的物理设备形式的改变而改变。

(3) 数据不能减少人的不确定性，而信息可以减少人的不确定性，这是数据与信息之间的本质区别。

1.1.2 数据库



数据库 (database, DB) 是长期存储在计算机内、有组织的、可共享的、统一管理的相关数据的集合。简单来说, 数据库可以理解为存储数据的仓库, 每个数据项在数据仓库中都有编号, 通过编号就可以找到该数据项。例如, 图书馆就是存储图书的仓库, 在图书馆存储的每本图书都有一个编号, 编号表示了书的类别和顺序号, 同类别的书放在一个书架上, 便于图书管理员和读者查找图书。图书馆存储的是图书, 数据库存储的是数据, 每条数据称为数据项, 当需要存储多个数据项时, 就构成了数据库, 数据库为多种应用提供数据服务。

数据库是一个按数据结构来存储和管理数据的计算机软件系统, 其主要特点如下:

1. 数据按一定的数据模型组织、描述和存储

数据结构化是数据库与文件系统的根本区别。在文件系统中, 相互独立的文件的记录, 其内部是有结构的。传统文件的最简单形式是等长同格式的记录集合。例如, 一个学生的人事记录文件, 每个记录都有如图 1-2 所示的记录格式。

学生人事记录:

学号	姓名	性别	政治面貌	年龄	籍贯	系别	家庭成员	奖惩情况
----	----	----	------	----	----	----	------	------

图 1-2 学生人事记录

在文件系统中, 尽管记录内部已有了某些结构, 但记录之间没有联系。数据库系统实现整体数据的数据结构化, 是数据库系统的主要特征之一, 也是数据库系统与文件系统的本质区别。在数据库系统中, 数据不再针对某一应用, 而是面向全组织, 具有整体的结构化特点。不仅数据是结构化的, 而且存取数据的方式也很灵活, 可以读、写数据库中的某一个数据项、一组数据项、一个记录或一组记录。

2. 可为各种用户共享

数据库系统从整体角度描述数据, 数据不再面向某个应用而是面向整个系统, 因此数据可以被多个用户、多个应用共享使用。数据共享可以大大减少数据冗余, 节约存储空间, 还能够避免数据之间的不相容性与不一致性。所谓数据的不一致性, 是指同一数据不同拷贝的值不一样。采用人工管理或文件系统管理时, 由于数据被重复存储, 当不同的应用使用和修改不同的拷贝时就很容易造成数据的不一致。在数据库中进行数据共享, 减少了由于数据冗余造成的一致现象。由于数据面向整个系统, 是有结构的, 数据不仅可以被多个应用共享使用, 而且容易增加新的应用, 这就使得数据库系统弹性大, 易于扩充, 可以适应用户的各种要求。用户可以取整体数据的各种子集用于不同的应用系统, 当应用需求改变或增加时, 只要重新选取不同的子集或加上一部分数据便可以满足新的需求。

3. 数据冗余度较小

数据冗余是指数据在存储器中不必要的多次重复存储, 即在一个数据集合中重复的数据。数据冗余会妨碍数据库中数据的完整性 (integrality), 也会造成存储空间的浪费。尽可能地降低数据冗余度, 是数据库设计的主要目标之一。关系模式的规范化理论 (NF 理论) 的主要思想之一就是最小冗余原则, 即规范化的关系模式在某种意义上冗余度应该



最小。

4. 数据独立性较高

数据的独立性是数据库领域中的一个常用术语，包括数据的物理独立性和数据的逻辑独立性。物理独立性是指用户的应用程序与存储在磁盘上的数据库中的数据是相互独立的。也就是说，数据在磁盘上的数据库中的存储是由 DBMS 管理的，应用程序要处理的只是数据的逻辑结构，这样当数据的物理存储改变时，应用程序却不用改变。逻辑独立性是指用户的应用程序与数据库的逻辑结构是相互独立的，也就是说，数据的逻辑结构改变了，用户程序也可以不变。数据与程序的独立，把数据的定义从程序中分离出去，加上数据的存取又由 DBMS 负责，从而简化了应用程序的编制，大大减少了对应用程序的维护和修改。

5. 可扩展性好

可扩展性 (scalability) 是一种对数据库系统处理能力的设计指标，高可扩展性代表一种弹性，在系统扩展成长过程中，数据库能够保证旺盛的生命力，通过很少的改动甚至只是硬件设备的添置，就能实现整个系统处理能力的线性增长，实现高吞吐量和低延迟高性能。

可扩展性和纯粹地性能调优有本质区别。可扩展性是高性能、低成本和可维护性等诸多因素的综合考量和平衡，可扩展性讲究平滑线性的性能提升，更侧重于系统的水平伸缩，通过廉价的服务器实现分布式计算；而普通性能优化只是单台机器的性能指标优化。他们的共同点都是根据应用系统特点在吞吐量和延迟之间进行侧重选择。

1.1.3 数据库管理系统

数据库管理系统 (database management system, DBMS) 是为管理数据库而设计的一种大型软件，用于建立、使用和维护数据库。数据库管理系统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 进行数据库的维护工作。DBMS 可使多个应用程序和用户用不同的方法在相同或不同时刻去建立、修改和询问数据库。

数据库管理系统是数据库系统的核心，是管理数据库的软件。数据库管理系统就是把用户意义下抽象的逻辑数据，转换成为计算机中具体的物理数据的软件。有了数据库管理系统，用户就可以在抽象意义下处理数据，而不必顾及这些数据在计算机中的布局和物理位置。

数据库语言是提供给用户并帮助用户实现数据库管理的语言。结构化查询语言 (structured query language, SQL) 是一种集数据定义和数据操纵为一体的典型数据库语言，主要用于存取数据以及查询、更新和管理关系数据库系统。SQL 语言由 Boyce 和 Chamberlin 于 1974 年提出，具有功能丰富、使用方便、语言简洁等优点。20 世纪 80 年代，经美国国家标准学会 (American National Standards Institute, ANSI) 的数据库委员会 X3H2 批准，将 SQL 作为关系数据库语言的美国标准，不久后，国际标准化组织 (International Organization for Standardization, ISO) 将其采纳为关系数据库语言的国际标准，后面出现的关系数据库产品几乎都将 SQL 语言作为标准数据库语言。SQL 语言的主

要功能如下：

(1) 数据定义。数据定义语言 (data definition language, DDL) 是 DBMS 的基础功能之一, 供用户定义数据库的三级模式结构、两级映像以及完整性约束和权限约束等。数据定义包括数据库模式定义和数据库存储结构与存取方法定义两方面, 数据库模式定义处理程序接收的、用数据定义语言表示的数据库外模式、模式、存储模式及它们之间的映射的定义, 通过各种模式翻译程序将它们翻译成相应的内部表示形式, 存储到数据库系统的数据字典 (data dictionary) 中, 作为数据库管理系统存取和管理数据的基本依据; 而数据库存储结构和存取方法定义处理程序接收的用数据定义语言表示的数据库存储结构和存取方法定义, 在存储设备上创建相关的数据库文件, 建立起相应物理数据库, 给出数据库的框架并存放在数据字典中。数据定义的常用语句包括 CREATE、ALTER 和 DROP 等。

(2) 数据操作。数据操作语言 (data manipulation language, DML) 是 DBMS 的另一基础功能, 用来表示用户对数据库的操作请求, 是用户与 DBMS 之间的接口。DML 一般供用户实现对数据的添加、删除、更新、查询等操作, 常用语句包括 SELECT、INSERT、UPDATE 和 DELETE 等。数据操作语言通常又分为两类: 一类是嵌入主语言, 由于这种语言本身不能独立使用, 故称为宿主型的语言; 另一类是交互式命令语言, 由于这种语言本身能独立使用, 故又称为自主型或自含型的语言。

(3) 数据控制。DBMS 提供的数据库控制语言 (data control language, DCL), 主要用于系统运行控制、管理, 包括多用户环境下的并发控制、安全性检查和存取限制控制、完整性检查和执行、运行日志的组织管理、事务的管理和自动恢复, 保证事务的原子性。这些功能保证了数据库系统的正常运行。数据控制的常用语句包括 GRANT 或 REVOKE 等。

(4) 数据组织、存储与管理。DBMS 要分类组织、存储和管理各种数据, 包括数据字典、用户数据、存取路径等, 需确定以何种文件结构和存取方式在存储级上组织这些数据, 如何实现数据之间的联系。数据组织和存储的基本目标是提高存储空间利用率, 选择合适的存取方法提高存取效率。

(5) 数据库的保护。数据库中的数据是信息社会的战略资源, 所以数据的保护至关重要。DBMS 对数据库的保护通过 4 个方面来实现: 数据库的恢复、数据库的并发控制、数据库的完整性控制、数据库的安全性控制。DBMS 的其他保护功能还包括系统缓冲区的管理以及数据存储的某些自适应调节机制等。

(6) 数据库的维护。这一部分包括数据库的数据载入、转换、转储、数据库的组合重构以及性能监控等功能, 这些功能分别由各个应用程序来完成。

(7) 数据通信。DBMS 具有与操作系统的联机处理、分时系统及远程作业输入的相关接口, 负责处理数据的传送。对网络环境下的数据库系统, 还应该包括 DBMS 与网络中其他软件系统的通信功能以及数据库之间的互操作功能。

1.1.4 数据库系统

数据库系统是指在计算机系统中引入数据库后构成的系统, 一般由数据、数据库、数据库管理系统 (及其开发工具)、操作数据库的应用系统、支撑的硬件平台、支撑的软件



平台、数据库管理员和用户等构成的一个完整的系统。数据库是数据库系统的核心和管理对象。因此，数据库系统的含义已经不仅仅是一个对数据进行管理的软件，也不仅仅是一个数据库，而是一个实际运行的，按照数据库方式存储、维护和向应用系统提供数据支持的系统，其构成如图 1-3 所示。

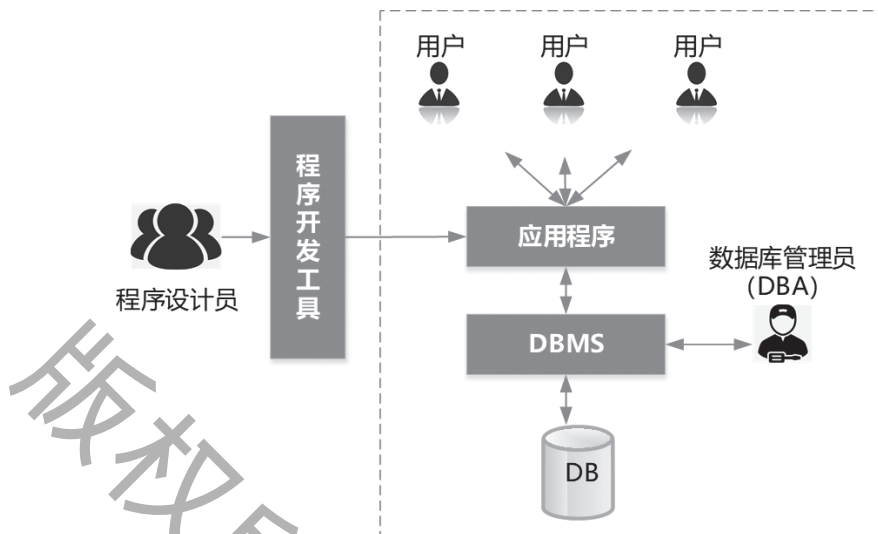


图 1-3 数据库系统的构成

1.2 数据库技术的发展

数据管理技术的发展过程经历了四个不同的阶段：人工管理阶段、文件系统阶段、数据库系统阶段、高级数据库阶段。

1. 人工管理阶段

20 世纪 50 年代中期以前，计算机主要用于科学计算，对数据的管理主要为人工管理。外部存储器只有磁带、卡片和纸带等，还没有磁盘等直接存取存储的设备。软件只有汇编语言，尚无数据管理方面的软件，数据处理方式基本是批处理。该阶段的特点是数据不保存、没有对数据进行管理的软件系统、没有文件概念、数据不共享、数据不具有独立性等。

2. 文件系统阶段

20 世纪 50 年代后期至 60 年代中期为文件系统阶段，计算机不仅用于科学计算，还用于信息管理。随着数据量的增加，数据的存储、检索和维护问题日益紧迫，数据结构和数据管理技术迅速发展起来，硬件方面，外部存储有了磁盘、磁鼓等直接存取的存储设备；软件方面，操作系统中有了专门用于管理数据的软件，即文件系统。文件系统能够提供目录结构简单的文件组织形式，并将其作为操作系统的基本用户界面。该阶段的数据可以长期保存，由文件系统管理数据，文件的形式已经多样化，数据具有一定的独立性。

3. 数据库系统阶段

20 世纪 60 年代后期至 80 年代，数据管理技术进入数据库系统阶段。数据库系统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。这个阶段的程序和数据的联系通过数据库管理系统（DBMS）来实现。该阶段的特点是采用复杂的结构化数据模型，

具有较高的数据独立性和较低的冗余度。

4. 高级数据库阶段

20世纪80年代,随着科学技术的不断进步,各个行业都对数据库技术提出了更多的需求,关系型数据库已经不能完全满足这些需求,于是数据库管理技术进入了高级数据库阶段。该阶段的主要特征是支持数据管理、对象管理和知识管理,支持数据库语言标准和标准网络协议,对其他系统开放,具有良好的可移植性、可连续性、可扩展性和可操作性。数据库技术和诸多新技术结合,如分布处理技术、并行处理技术、人工智能技术、多媒体技术、模糊技术等,广泛应用于商业管理、GIS、统计分析等多个领域,并由此衍生出多种新的数据库技术。不同数据库技术阶段比较如表1-1所示。

表 1-1 不同数据库技术阶段比较

比较项目		人工管理阶段	文件系统阶段	数据库系统阶段
背景	应用背景	科学计算	科学计算、数据管理	大规模数据管理
	硬件背景	无直接存取存储设备	磁盘、磁鼓	大容量磁盘、磁盘阵列
	软件背景	无操作系统	有文件系统	有数据管理系统
特点	数据的处理方式	批处理	联机实时处理、批处理	联机实时处理、分布处理、批处理
	数据的管理者	用户(程序员)	文件系统	数据库管理系统
	数据面向的对象	某一应用程序	某一应用程序	现实世界(部门、企业、组织等)
	数据的共享程度	无共享,冗余度极大	共享性差,冗余度大	共享性高,冗余度低
	数据的独立性	不独立,完全依赖程序	独立性差	具有高度的物理独立性和一定的逻辑独立性
	数据的结构化	无结构	记录内有结构,整体无结构	整体结构化,用数据模型描述
数据控制	由应用程序控制	由应用程序控制	由数据库管理系统保证数据的安全性、完整性、并发控制和恢复能力	

1.3 数据模型

数据模型是数据库的基础,是数据库系统的一个关键概念。数据是对客观事物的符号表示,模型是现实世界的抽象,数据模型是对数据特征的抽象,用来描述记录内的数据项和数据项之间关系的数据结构形式。数据模型通常需要满足三方面的要求:能较真实地模拟现实世界;易于被人理解;便于计算机实现。

目前成熟地应用在数据库系统中的数据模型有层次模型、网状模型和关系模型。它们之间的根本区别在于数据之间联系的表示方式不同。

1.3.1 层次模型

层次模型是数据库系统最早使用的一种模型。层次模型将数据组织成一对多关系的结构,并表示为一棵“有向树”,根节点在树的最上端,层次最高,每个节点的上方节点称



笔记

为该节点的父节点，而下方节点称为该节点的子节点，没有子节点的节点称为叶子节点。层次结构采用关键字来访问其中每一层次的每一部分。最有影响的层次模型的 DBS 是 20 世纪 60 年代末 IBM 公司推出的 IMS 层次模型数据库系统，作为第一个大型的商用数据库管理系统曾被广泛应用。

层次模型的特征：有且仅有一个节点没有父节点，即根节点；其他节点有且仅有一个父节点。图 1-4 是一个层次数据模型示例。

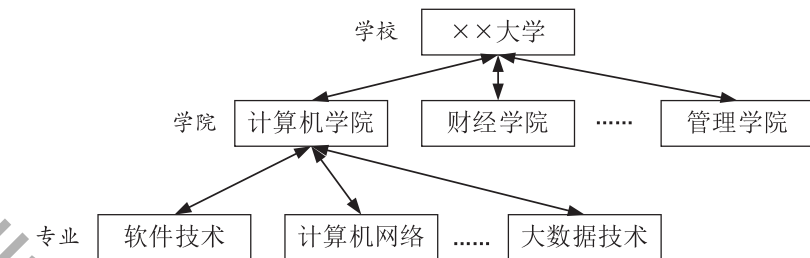


图 1-4 层次数据模型示例

层次模型的优点是存取方便且速度快、结构清晰、容易理解、数据修改和数据库扩展容易实现、检索关键属性十分方便；缺点是结构呆板、缺乏灵活性、同一属性数据要存储多次、数据冗余大、不适合于拓扑空间数据的组织。

1.3.2 网状模型

网状模型以网状结构表示实体与实体之间的联系，并表示为一张“图”。网中的每一个节点代表一个记录类型，节点之间的联系用链接指针来实现。网状模型可以表示多个从属关系的联系，也可以表示数据间的交叉关系，即数据间的横向关系与纵向关系，是层次模型的扩展。

网状模型的特征：允许节点有多于一个父节点，也可以有一个以上的节点没有父节点，两个节点之间可以有一对多联系。图 1-5 所示是一个网状数据模型示例。

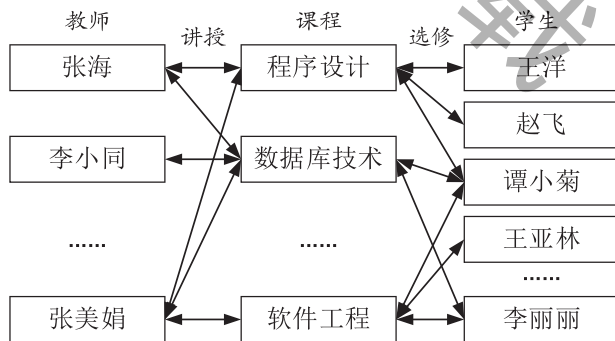


图 1-5 网状数据模型示例

网状模型的优点是可以方便地表示数据间的复杂关系，数据冗余小；缺点是网状结构复杂，实现的算法难以规范化，增加了用户查询和定位的困难，需要存储数据间联系的指针，使得数据量增大，数据修改不方便。

1.3.3 关系模型



关系模型由 IBM 公司的高级研究员 E.F.Codd 于 1970 年提出，是目前描述现实世界主要的抽象化方法，以关系数学理论为基础，以“二维表”结构来表示实体与实体之间的联系，每个二维表又可称为关系表。在关系模型中，操作的对象和结果都是关系表。支持关系模型的数据库管理系统称为关系数据库管理系统，20 世纪 80 年代以来推出的数据库管理系统都支持关系数据模型，MySQL 就是一种关系数据库管理系统。

关系模型由关系数据结构、关系操作集合和关系完整性约束三部分组成。

(1) 关系数据结构：关系模型采用二维表结构来表示实体以及实体间的联系，一个关系对应一个二维表，二维表名就是关系名。

(2) 关系操作集合：关系操作集合是建立在关系上的数据操纵，一般有数据查询（基本单位是元组分量）、数据删除（基本单位是元组）、数据插入（基本单位是元组）和数据修改（基本单位是元组分量）四种操作。

(3) 关系完整性约束：关系模型提供实体完整性约束、参照完整性约束和用户完整性约束三种数据约束。

以“学生选课系统”为例，假设系统涉及“学生表”“课程表”和“成绩表”3 个关系表，如表 1-2 至表 1-4 所示。

表 1-2 学生表

学号	姓名	性别	出生日期	专业	籍贯	电话
20191001	王洋	男	2000.10	软件技术	广州市	13500112233
20191002	赵飞	男	2000.08	软件技术	深圳市	13511223344
20191003	谭小菊	女	1999.04	软件技术	北海市	15811223344
20191004	王亚林	男	2000.06	软件技术	南通市	15822334455
20191005	李丽丽	女	2001.07	软件技术	成都市	15833445566

表 1-3 课程表

课程编号	课程名称	学时	学分	开设学期
J0001	程序设计	64	4	1
J0002	数据库技术	64	4	2
J0003	软件工程	48	3	3

表 1-4 成绩表

学号	课程编号	成绩
20191001	J0001	83
20191002	J0001	78
20191003	J0001	90
20191003	J0002	79
20191003	J0003	86

(续表)

笔记 

学号	课程编号	成绩
20191004	J0003	80
20191005	J0002	74
20191005	J0003	58

在“学生选课系统”中，关系表“学生表”和“课程表”用于描述实体本身，关系表“成绩表”用于描述实体之间的联系，表现了描述的一致性。关系模型可以表示“一对一”“一对多”“多对多”等不同类型的联系，同时数据间的关系必须是规范化的，即每个属性是不可分的数据项，不允许出现表中嵌套表的情况。

关系模型的优点在于结构灵活，概念单一，可以满足所有布尔逻辑运算和数学运算规则形成的查询要求，能搜索、组合和比较不同类型的数据，增加和删除数据非常方便，具有更高的数据独立性、更好的安全保密性。其缺点是数据库大时，查找满足特定关系的数据费时，对空间关系无法满足。

1.4 三级模式和二级映像

数据库模式 (schema) 是对现实世界的抽象，是对数据库中全部数据的逻辑结构和特征的描述，是所有用户的公共数据视图。模式反映的是数据的结果及其联系，数据库系统在其内部具有三级模式和二级映像。三级模式分别是外模式、模式和内模式，二级映像则是外模式 / 模式映像、模式 / 内模式映像，如图 1-6 所示。

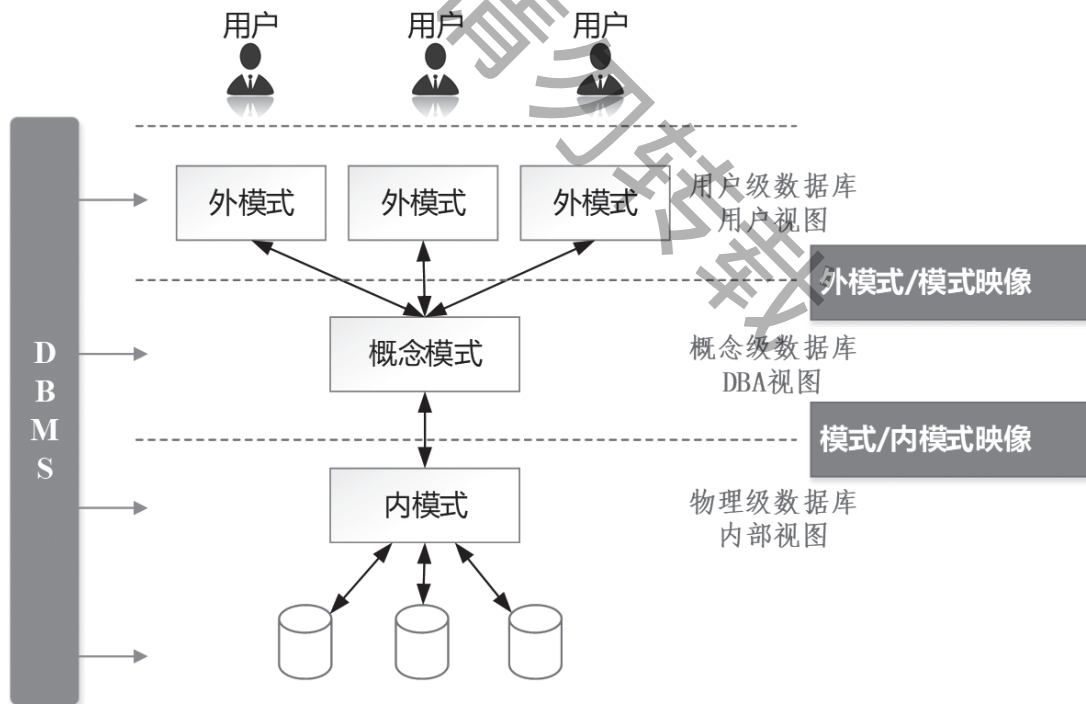


图 1-6 数据库的三级模式和二级映像



1.4.1 三级模式

美国国家标准学会 (ANSI) 的数据库管理学系统研究小组于 1978 年提出了标准化的建议, 将数据库结果分为三级: 面向用户或程序员的用户级、面向建立和维护数据库人员的概念级和面向系统程序员的物理级。用户级对应外模式, 概念级对应模式, 物理级对应内模式。

1. 模式

模式对应着概念级, 也称为逻辑模式, 它是由数据库设计者综合所有用户的数据, 按照统一的观点构造的全局逻辑结构, 是对数据库中全部数据的逻辑结构和特征的总体描述, 是所有用户的公共数据视图。一个数据库只有一个模式, 由数据库管理系统提供的数据库定义语言 (DDL) 来描述定义, 不仅要定义数据的逻辑结构 (如数据记录由哪些数据项构成, 数据项的名字、类型、取值范围等), 而且要定义与数据有关的安全性、完整性要求, 定义这些数据之间的联系, 这体现并反映了数据库系统的整体观。

2. 外模式

外模式 (external schema) 对应于用户级, 也称用户模式, 它是某个或某几个用户看见和使用的局部数据的逻辑结构和特征的描述, 是数据库用户的数据视图, 是与某个应用或用户具体的需求相关的数据逻辑表示。外模式是从模式导出的一个子集, 包含模式中允许特定用户使用的那部分数据, 一个数据库可以有多个外模式。用户可以通过外模式描述语言来描述、定义对应于用户的数据记录 (外模式), 也可以利用数据操纵语言 (DML) 对这些数据记录进行操作。

3. 内模式

内模式 (internal schema) 对应于物理级, 也称存储模式, 它是数据库中全部数据的内部表示或底层描述, 是数据库最低一级的逻辑描述, 它描述了数据在存储介质上的存储方式的物理结构, 对应着实际存储在存储介质上的数据库。例如, 内模式可以描述记录的存储方式是顺序存储、B 树结构存储还是 Hash 方法存储, 索引按照什么方式组织, 数据是否压缩、是否加密, 数据的存储记录结构有何规定等。一个数据库只有一个内模式。

1.4.2 二级映像

数据库系统的三级模式是对数据的 3 个抽象, 它把数据的具体组织留给 DBMS 管理, 使用户能抽象地处理数据, 而不必关心数据在计算机中的具体表示和存储。为了能够在内部实现这 3 个抽象层次的联系和转换, DBMS 在这 3 个级别之间提供了两层映像: “外模式 / 模式映像” 和 “模式 / 内模式映像”。

1. 外模式 / 模式映像

“外模式 / 模式映像” 定义了外模式和模式之间的对应关系, 使数据具有较高的逻辑独立性。一个模式可以有多个外模式, 每一个外模式都有一个外模式 / 模式映像。当模式改变时, DBA 要求相关的外模式 / 模式映像做相应的改变, 以使外模式保持不变。例如, 若要在关系数据库系统中增加新的关系、改变关系的属性或数据类型等, 则只需更改外模

式 / 模式的映射关系以保证面向用户的外模式不变，因为应用程序是依据实际的外模式编写的，所以外模式不变应用程序就没必要修改。所以“外模式 / 模式映像”功能保证了数据与程序的逻辑独立性，映像定义包含在外模式描述中。

2. 模式 / 内模式映像

“模式 / 内模式映像”定义了数据的逻辑结构形式与物理存储结构之间的对应关系，使数据具有较高的物理独立性。模式 / 内模式映像是唯一的，当存储结构发生改变时，DBA 对模式 / 内模式映像做相应的改变，以保持模式不变，从而使外模式无须发生改变，应用程序也无须发生改变。所以“模式 / 内模式映像”功能保证了数据与程序的物理独立性，其映像定义包含在模式描述里。

数据库的三级模式和二级映像是数据库管理系统（DBMS）对数据库中数据进行有效组织和管理的办法，可以有效减少数据冗余，提高存取效率，实现数据共享，改善系统性能。

1.5 关系数据库

关系数据库是目前各类数据库中最重要、最流行的数据库，是采用关系模型作为数据组织方式的数据库，它将用户数据的逻辑结构归纳为满足一定条件的二维表的形式。关系数据库的特点在于它将每个具有相同属性的数据独立地存储在一个表中，对任一表而言，用户可以新增、删除和修改表中的数据，而不会影响表中的其他数据。

1.5.1 基本概念

关系数据库中的基本术语如下：

1. 表

一个关系模型的逻辑结构是一张二维表，由行和列组成，这个二维表就叫关系。关系数据库的表采用二维表格来存储数据，通俗地说，一个关系对应一张表。一个数据库可以包含任意多个数据表。

2. 记录（元组）

表中的一行称为一条记录，或称为一个元组。一个记录的内容是描述一类事物中的一个具体事物的一组数据，如一个雇员的编号、姓名、工资数目，一次商品交易过程中的订单编号、商品名称、客户名称、单价、数量等。一般地，一个记录由多个数据项（字段）构成，记录中的字段结构由表的标题（关系模式）决定。

记录的集合（元组集合）称为表的内容，表的行数称为表的基数。值得注意的是，表名以及表的标题是相对固定的，而表中记录的数量和多少则是经常变化的。

3. 字段（属性）

数据表中的每一列称为一个字段，表是由其包含的各种字段定义的，每个字段表示表中所描述的对象的一个属性，如产品名称、单价、订购量等。每个字段都有相应的描述信息，如字段名、数据类型、数据宽度、数值型数据的小数位数等。由于每个字段都包含了数据类型相同的一批数据，因此，字段名相当于一种多值变量。字段是数据库操纵的最小

单位，字段的个数称为关系的元数。

表定义的过程就是指定每个字段的字段名、数据类型及长度（占用的字节数）。表中每个字段都只接受所定义的数据类型的数据。

4. 属性值

行和列的交叉位置表示某个属性值，如“数据库原理”就是课程名称的属性值。

5. 域

每个属性的取值范围对应一个值的集合，称为该属性的域。例如，性别属性的域为“男”“女”。

6. 主码（主键、主关键字）

主码是表中用于唯一地确定一个元组的数据。关键字用来确保表中记录的唯一性，可以是一个字段或多个字段，常用作一个表的索引字段。每条记录的关键字都是不同的，因而关键字可以唯一地标识一个记录。关键字也称为主关键字或者主键。例如，在“学生选课系统”的“学生表”（见表 1-2）中，学生的学号具有唯一性，学号可以作为学生的主键；“课程表”（见表 1-3）中，课程编号具有唯一性，课程编号可以作为课程的主键；“成绩表”（见表 1-4）中，学号、课程编号和成绩都可能存在重复值，不适合使用单一字段作为主键，因此，由学号和课程编号共同组成组合主键。

7. 关系模式

关系的描述称为关系模式。对关系的描述，一般表示为“关系名（属性 1，属性 2……属性 n ）”，例如表 1-3 中的关系可描述为“课程表（课程号、课程名称、学时、学分、开设学期）”。关系模式这种简单的数据结构能够表达丰富的语义，描述现实世界的实体以及实体间的各种联系。

8. 外码

当关系中的某个属性（或属性组）虽然不是该关系的主码（键）或者只是主码（键）的一部分，但却是另一个关系的主码（键）时，则称该属性（或属性组）为这个关系的外码（键）。例如，在“学生选课系统”中，“学生表”的主键是“学号”，“成绩表”的主键是“学号+课程编号”，“学号”在“成绩表”中称为外键。通过学号，两个表建立了一对多的关系，即一个学生可以有多门课程的成绩。

1.5.2 关系数据库设计

设计关系数据库系统需要将现实世界抽象到数据世界，即将现实世界用数据进行描述，得到一个现实世界的数据库模型。将客观事物抽象为数据库模型是一个逐步转化的过程，会经历现实世界、信息世界和计算机世界这三个不同的世界，经历两级抽象和转换，如图 1-7 所示。

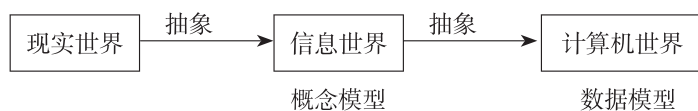


图 1-7 数据抽象过程





1. 现实世界

现实世界是指客观存在的事物及其相互之间的联系，现实世界中的事物有着众多的特征和千丝万缕的联系，人们一般选择事物的基本特征来描述事物。事物可以是具体的，也可以是抽象的，如学生属于具体的事物，通常用学号、姓名、性别、年龄等特征来描述和区分；课程属于抽象的事物，人们通常用课程名称、学时、学分、类别、课程简介等特征来描述和区分。

2. 信息世界

信息世界是对现实世界的一种抽象描述，人们把现实世界中事物的特征和联系通过符号记录下来，并用规范化的语言描述这些事物，从而构成一个基于现实世界的信息世界。在信息世界中，通过筛选、归纳、总结、命名等抽象过程产生出概念模型，用以表示对现实世界的抽象与描述。概念模型主要用来描述现实世界的概念化结构，是现实世界到计算机世界的中间层。概念模型是数据库设计的初始阶段，它摆脱计算机系统及数据库管理系统的具体技术问题，集中精力分析数据以及数据之间的联系。

在概念模型中，最常用的设计模型就是实体联系模型（ER 模型）。ER 模型是描述现实世界概念结构模型的有效方法，是概念数据模型的高层描述所使用的数据库模型，不依赖于具体的计算机实现。ER 模型通常用 ER 图来表示，包含了实体（即数据对象）、属性和关系等 3 种基本成分。

（1）实体（entity）：即数据模型中的数据对象，具有相同的特征和性质。例如学生、商品、图书等都可以作为一个实体。实体在 ER 图中用矩形表示，矩形框内写明实体名。每个实体都有自己的实体成员（entity member）或者说实体对象（entity instance），例如学生实体里包括张三、李四等成员。

（2）属性（attribute）：实体所具有的某一特性，就是数据对象所具有的属性。一个实体可由若干个属性来刻画，比如学生的姓名、学号、性别等都是学生的属性。在 ER 图中属性用椭圆形表示，并用无向边将其与相应的实体连接起来。属性分为唯一属性（unique attribute）和非唯一属性，唯一属性指的是唯一可用来标识该实体实例或者成员的属性，用下划线表示，一般来讲，每个实体都至少有一个唯一属性。

（3）联系（relationship）：数据对象彼此之间相互连接的方式称为联系，也称为关系。ER 图中联系用菱形来表示，并用直线连接该联系和与其关联的实体。联系可分为“一对一”“一对多”“多对多”3 种类型，如图 1-8 所示。

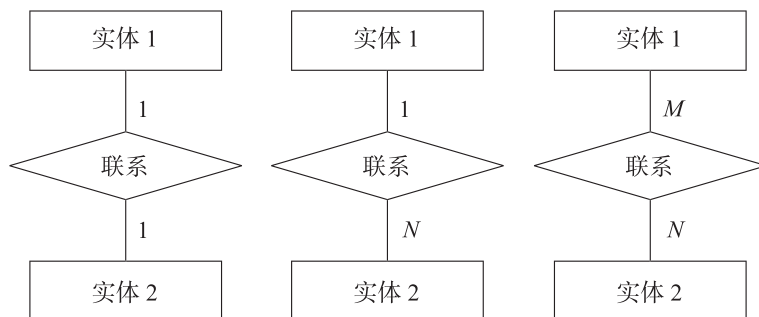


图 1-8 实体联系中三种类型的表示方法



①一对一联系(1:1)是指对于实体A与实体B, A中的每一个实体对象至多与B中一个实体对象有关系;反之,在实体B中的每个实体对象至多与实体A中一个实体对象有关系。例如,一个部门有一个经理,而每个经理只在一个部门任职,则部门与经理的联系是一一对一的。

②一对多联系(1:N)是指实体A中的每一个实体对象与实体B中至少 $N(N>0)$ 个实体对象有关系;并且实体B中每一个实体对象至多与实体A中一个实体对象有关系。例如,某校教师与课程之间存在一对多的联系,每位教师可以教多门课程,但是每门课程只能由一位教师来教。

③多对多联系($M:N$)是指实体A中的每一个实体对象与实体B中至少 $M(M>0)$ 个实体对象有关系,并且实体B中的每一个实体对象与实体A中的至少 $N(N>0)$ 个实体对象有关系。例如,学生与课程间的联系就是多对多,即一个学生可以学多门课程,而每门课程可以有多个学生来学。值得注意的是,联系也可能有属性,例如,学生学某门课程所取得的成绩,既不是学生实体的属性也不是课程实体的属性,而是选修关系的属性,因为“成绩”既依赖于某位特定的学生又依赖于某门特定的课程,所以是学生与课程之间的联系的属性。

3. 计算机世界

计算机世界是将信息世界的内容数据化后的产物。信息世界的概念模型还不能被数据库管理系统直接使用,需要将概念模型进一步转换为逻辑数据模型,形成便于计算机处理的数据形式。

ER图由实体、实体的属性和实体间的联系三个要素组成。关系模型的逻辑结构是一组关系模式的集合,将ER图转换为关系模型,也就是将实体、实体的属性和实体之间的联系转化为关系模式。

1) 实体的转换规则

一个实体转换为一个关系模式,实体的属性转换为关系的属性,实体的主码转换为关系的主码。例如,学生实体可转换为关系模式:学生表(学号,姓名,出生日期,专业,籍贯,联系电话,班级编号),教师、课程都可以分别转换为一个关系模式。

2) 实体间联系的转换规则

(1)“1:1联系”的转换方法。“1:1联系”可以转换为一个独立的关系模式,也可以与任意一端对应的关系模式合并。转换为一个独立的关系模式时,关系的属性包括联系相关联的各实体的码以及联系本身的属性,每个实体的码均是该关系的候选码;与某一端对应的关系模式合并时,合并后关系的属性加入联系所关联实体的码和联系本身的属性,合并后关系的主码不变。

(2)“1:N联系”的转换方法。“1:N联系”可以转换为一个独立的关系模式,也可以与N端对应的关系模式合并。转换为一个独立的关系模式时,关系的属性包括该联系相关联的各实体的码以及联系本身的属性,关系的主码为N端实体的码;与某N端对应的关系模式合并时,合并后在N端关系中加入1端实体的码和联系本身的属性,合并后关系的主码不变。

(3)“M:N联系”的转换方法。“M:N联系”转换为一个独立的关系模式,关系的属



性包括该联系相关联的各实体的码以及联系本身的属性，关系的主码为各实体码的组合。

(4) 多元联系的转换方法。要将三个或三个以上实体间的多元联系转换为关系模式，可根据以下两种情况采用不同的方法处理，对于一对多的多元联系，转换方法是修改 N 端实体集对应的关系，即将与联系相关的 1 端实体的主码和联系自身的属性作为新属性加入 N 端实体中；对于多对多的多元联系，转换方法是新建一个独立的关系模式，该关系的属性为多元联系相关联的各实体的码以及联系本身的属性，关系的主码为各实体码的组合。

3) 关系合并规则

具有相同码的关系模式可合并，目的是减少系统中的关系个数。合并方法是将其中的一个关系模式的全部属性加入另一个关系模式中，然后去掉其中的同义属性（可能同名，也可能不同名），并适当调整属性的次序。

1.6 常用数据库管理系统

随着数据库技术的发展，数据库产品越来越多，如 Oracle、DB2、Sybase、SQL Server、MySQL 等。

1. Oracle

Oracle Database，又名 Oracle RDBMS，或简称 Oracle，是美国甲骨文公司开发的一款关系数据库管理系统。Oracle 数据库系统作为一个通用的关系数据库管理系统，采用标准的 SQL，并经过美国国家标准技术所（NIST）测试，具有完整的数据管理功能，具有良好的兼容性、可移植性和可伸缩性，且使用方便、性能高、安全性强，适用于各类大、中、小、微机环境。作为一个关系数据库，Oracle 数据库是一个具有完备关系的产品；作为分布式数据库，它实现了分布式处理功能，为企业提供高效率、可靠性好的，适应高吞吐量的数据库解决方案，在数据库领域一直处于领先地位，市场占有率很高。

2. DB2

DB2 是美国 IBM 公司开发的一套关系型数据库管理系统，主要的运行环境为 UNIX（包括 IBM 自家的 AIX）、Linux、IBM i、z/OS，以及 Windows 服务器版本。DB2 主要应用于大型应用系统，具有较好的可伸缩性，可支持大型机和单用户环境，应用于所有常见的服务器操作系统平台。DB2 提供了高层次的数据利用性、完整性、安全性、可恢复性，从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，适合于海量数据的存储，但操作比较复杂。

3. Sybase

Sybase 是美国 Sybase 公司研制的一种关系型数据库系统，是一种典型的 UNIX 或 Windows NT 平台上的客户机 / 服务器环境下的大型数据库系统。Sybase 提供了一套应用程序编程接口和库，可以与非 Sybase 数据源及服务器集成，允许在多个数据库之间复制数据，适于创建多层应用。系统具有完备的触发器、存储过程、规则以及完整性定义，支持优化查询，具有较好的数据安全性，在我国大中型系统中具有广泛的应用。

4. SQL Server

SQL Server 是 Microsoft（微软）公司推出的关系型数据库管理系统，具有使用方便、可伸缩性好、与相关软件集成程度高等优点。SQL Server 作为一个全面的数据库平台，易于操作，界面良好，数据库引擎为关系型数据和结构化数据提供了更安全可靠存储功

能，集成商业智能（BI）工具可提供企业级的数据管理，广泛应用于电子商务、银行、保险、电力等行业。SQL Server 早期版本只能在 Windows 平台上运行，2017 版已经支持 Windows 和 Linux 平台，使用户可以构建和管理用于业务的高可用和高性能的数据应用程序。

5.MySQL

MySQL 由瑞典 MySQL AB 公司开发，先后被 Sun 和 Oracle 公司收购，目前属于 Oracle 旗下产品。MySQL 是最流行的关系型数据库管理系统之一，可以在 UNIX、Linux、Mac OS 和 Windows 等平台上使用。MySQL 体积小、速度快，使用更加方便、快捷，并且开放源代码，开发人员可根据需求自由进行修改，一般中小型网站的开发都选择 MySQL 作为网站数据库。MySQL 软件采用了双授权政策，分为社区版和商业版，兼顾免费和付费场景，软件使用成本低。由于其社区版的性能卓越，搭配 PHP 和 Apache 可组成良好的开发环境，在 Web 开发领域，MySQL 占据着举足轻重的地位。

任务实践

根据“学习任务”中描述的“教务管理系统”功能场景和需求，分析学生、课程、班级、教师的特征，可知学生有学号、姓名、性别、出生日期、专业、籍贯、联系电话等基本特征；课程有课程编号、课程名称、学时、学分、开设学期等基本特征；教师有教工号、姓名、性别、出生日期、所在院系、职称、联系电话、电子邮件等基本特征；班级有班级编号、班级名称、人数、班主任等基本特征；学生与课程的主要关系是“选修”，具有成绩属性；教师、课程和班级三者之间的关系是“教授”，具有学年、学期等属性；学生和班级的主要关系是“属于”。经过抽象，画出的 ER 模型如图 1-9 所示，完成从现实世界到概念世界的抽象。

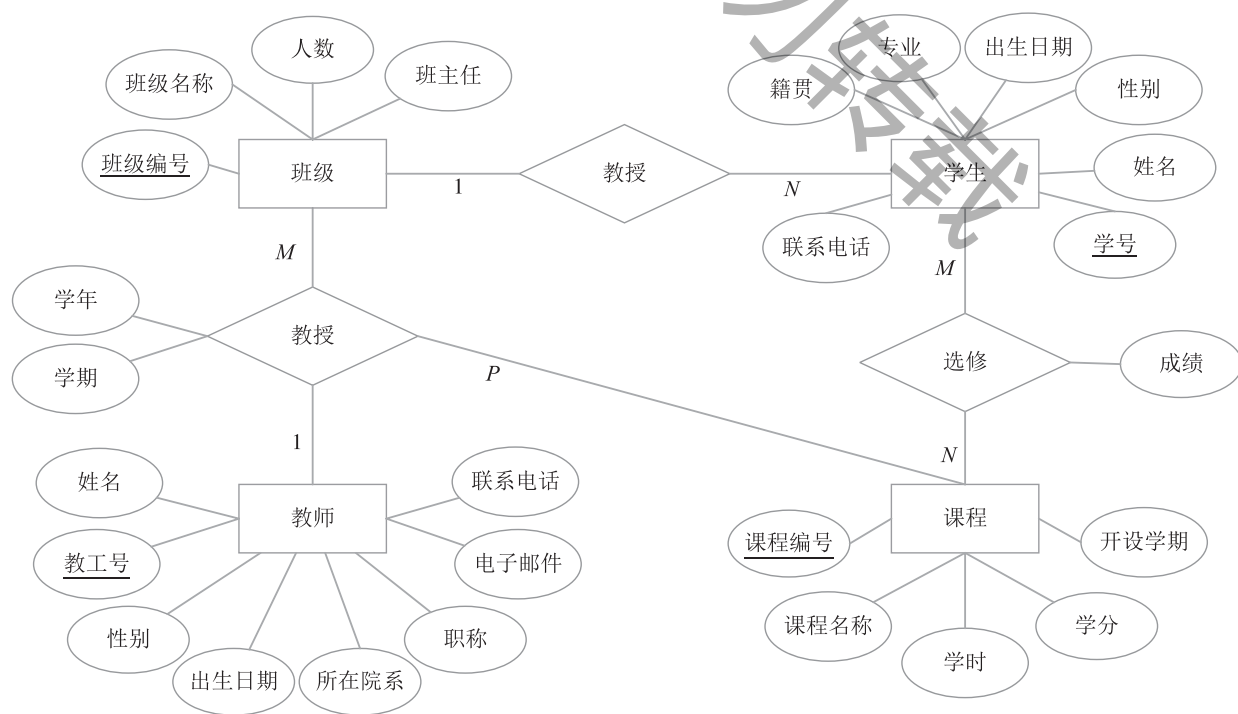


图 1-9 “教务管理系统” ER 模型



笔记



将图 1-9 中描述的 ER 模型转换成逻辑数据模型，结果如下：

- (1) 学生表 (学号, 姓名, 性别, 出生日期, 专业, 籍贯, 联系电话, 班级编号);
 - (2) 课程表 (课程编号, 课程名称, 学时, 学分, 开设学期);
 - (3) 班级表 (班级编号, 班级名称, 人数, 班主任);
 - (4) 教师表 (教工号, 姓名, 性别, 出生日期, 所在院系, 职称, 联系电话, 电子邮件);
 - (5) 选修表 (学号, 课程编号, 成绩);
 - (6) 授课表 (班级编号, 教工号, 课程编号, 学年, 学期)。
- 有了以上关系数据模型结果，在数据库管理系统中就可以创建二维表。

思考与练习

一、选择题

1. 下列四项中，不属于数据库特点的是_____。
 - (A) 数据共享
 - (B) 数据完整性
 - (C) 数据冗余很高
 - (D) 数据独立性高
2. _____是位于用户与操作系统之间的一层数据管理软件，它属于系统软件，为用户或应用程序提供访问数据库的方法。数据库在建立、使用和维护时由其统一管理、统一控制。
 - (A) DBMS
 - (B) DB
 - (C) DBS
 - (D) DBA
3. 下列四项中，不正确的提法是_____。
 - (A) SQL 语言是关系数据库的国际标准语言
 - (B) SQL 语言具有数据定义、查询、操纵和控制功能
 - (C) SQL 语言可以自动实现关系数据库的规范化
 - (D) SQL 语言称为结构查询语言
4. 目前_____数据库系统已逐渐淘汰了网状数据库和层次数据库，成为当今最为流行的商用数据库系统。
 - (A) 关系
 - (B) 面向对象
 - (C) 分布
 - (D) 逻辑
5. 数据库设计中的概念结构设计的主要工具是_____。
 - (A) 数据模型
 - (B) ER 模型
 - (C) 逻辑模型
 - (D) 概念模型
6. 数据库设计中的逻辑结构设计任务是把_____阶段产生的概念数据库模式变换为逻辑结构的数据库模式。
 - (A) 需求分析
 - (B) 物理设计
 - (C) 逻辑结构设计
 - (D) 概念结构设计
7. 数据定义语言的缩写词为_____。
 - (A) DDL
 - (B) DCL
 - (C) DML
 - (D) DBL
8. 公司中有多个部门和多名职员，每个职员只能属于一个部门，一个部门可以有多名职员，从部门到职员的关系类型是_____。

(A) 多对多 (B) 一对一 (C) 多对一 (D) 一对多

9. 数据库的三要素不包括_____。

(A) 完整性规则 (B) 数据结构 (C) 恢复 (D) 数据操作

10. 如果在一个关系中，存在某个属性（或属性组），虽然不是该关系的主码或只是主码的一部分，但却是另一个关系的主码时，称该属性（或属性组）为这个关系的_____。

(A) 候选码 (B) 主码 (C) 外码 (D) 连接码

二、填空题

1. 数据库系统具有数据的_____、_____和内模式三级模式结构。

2. 关系数据模型的逻辑结构是_____，关系中的列称为_____、行称为_____。

3. 数据库系统的三级模式结构和两级映像确保了数据的_____独立性和_____独立性。

4. SQL 缩写词的意义是_____语言。

5. DML 缩写词的意义是_____语言。

三、简答题

1. 如何理解信息与数据的关系？

2. 数据库系统与数据库管理系统的区别是什么？



第 2 章

MySQL 的安装与配置

学习成果目标 >

- 1 了解获取 MySQL 的途径。
- 2 掌握 MySQL 的安装与配置方法。
- 3 掌握 MySQL 服务的启动与关闭。
- 4 掌握登录密码的设置方法。
- 5 培养运用自然科学知识和工具的能力。

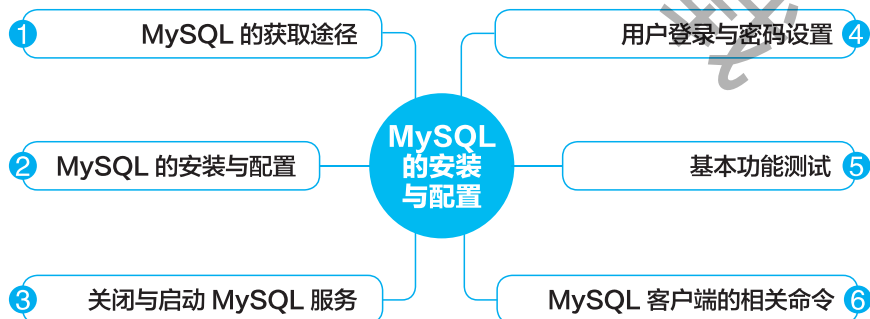
学习任务 >

通过对 MySQL 数据库的获取、安装、配置、启动等基本操作，完成 MySQL 开发环境的准备。

学习内容 >

掌握如何在 Windows 平台下部署 MySQL 5.7.34 版本。并通过对 MySQL 服务器的配置进行简单测试，尝试初次使用 MySQL 的同时确保各项基本参数配置无误，为后续学习做好充分的准备。

本章的内容结构如下：



2.1 MySQL 的获取途径



MySQL 数据库是目前运行速度最快的 SQL 语言数据库之一，是一种完全免费的产品，在实际工作中，可以根据不同需求单击 MySQL 官方网站（<https://www.mysql.com/>）页面中的“DOWNLOADS”按钮获取 MySQL 的各种不同版本。本书中的所有实例均基于 MySQL 5.7.34 版本，因此这里也同样以该版本为例来进行介绍。

首先，打开 MySQL 官方首页并单击“DOWNLOADS”按钮进入如图 2-1 所示页面。



图 2-1 MySQL 官方页面

接着向下拖动浏览器中的滑块，找到“MySQL Community (GPL) Downloads »”链接并单击，如图 2-2 所示。



图 2-2 单击下载链接

打开新页面后，找到并单击页面中的“MySQL Community Server”链接，进入 MySQL Community Server 的版本选择及下载页面。进入该页面后，单击如图 2-3 中所显示的“Archives”标签。

笔记

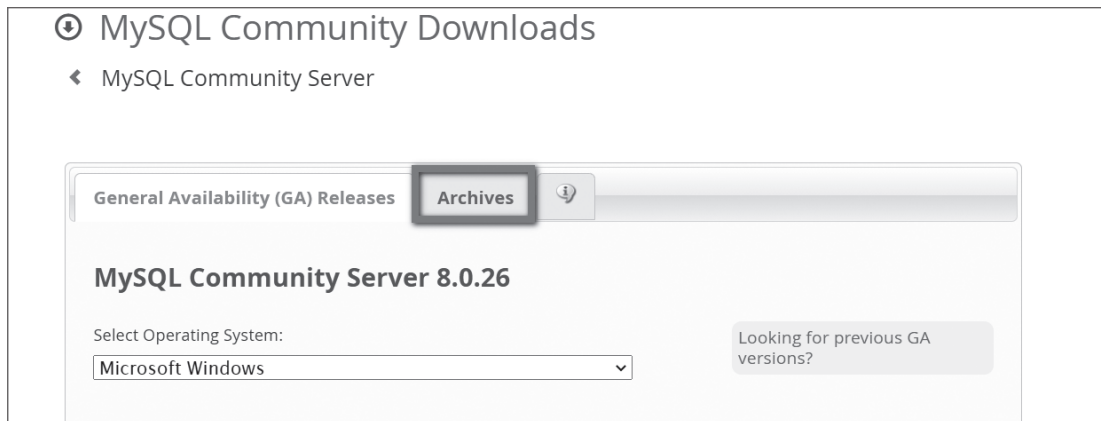


图 2-3 单击“Archives”标签

在“MySQL Product Archives”页面中单击“Product Version”右边的下拉按钮；接着在下拉菜单中找到并选择 5.7.34 版本，如图 2-4 所示。

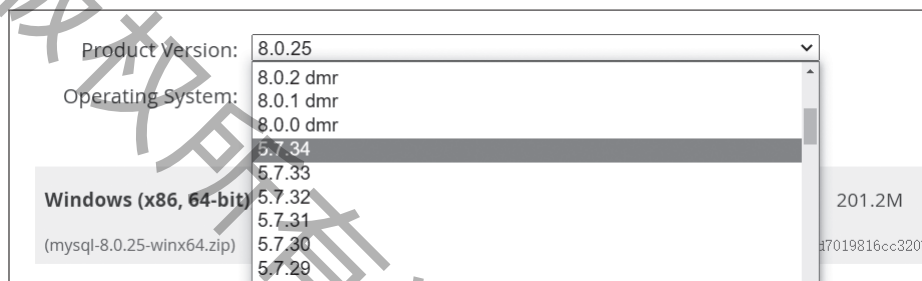


图 2-4 选择版本号

单击后，可以看到页面中的内容发生了改变并出现如图 2-5 所示的 4 项可供下载的内容。此时可以单击所需版本对应的“Download”按钮，即可根据自己的需要获取 Windows 版本下载对应的 MySQL 5.7.34 压缩文件。

Other Downloads:			
Windows (x86, 32-bit), ZIP Archive (mysql-5.7.34-win32.zip)	32位版本	5.7.34	352.3M MD5: a18771b52243df781f4ff91c2363b761 Signature
Windows (x86, 64-bit), ZIP Archive (mysql-5.7.34-winx64.zip)	64位版本	5.7.34	369.2M MD5: 9802dbf9193b5e200b6f7636c1bcb3af Signature
Windows (x86, 32-bit), ZIP Archive Debug Binaries & Test Suite (mysql-5.7.34-win32-debug-test.zip)		5.7.34	446.6M MD5: 85f1279b2648e2fceb43f0a5ab9cc27 Signature
Windows (x86, 64-bit), ZIP Archive Debug Binaries & Test Suite (mysql-5.7.34-winx64-debug-test.zip)		5.7.34	455.5M MD5: 3366373510ed7885a4578441543af73e Signature

图 2-5 下载 MySQL 安装文件

需要注意的是，32 位版本需下载的文件名为“Windows (x86, 32-bit), ZIP Archive”，64 位版本需下载的文件名为“Windows (x86, 64-bit), ZIP Archive”。

2.2 MySQL 的安装与配置



下载成功后双击打开得到的压缩包，并将压缩包中的文件夹解压至任意纯英文目录下（此处以解压至 E 盘根目录下为例）。解压完成后双击打开文件夹（为便于说明，后文中将该文件夹的路径称为“MySQL 的根目录”），如图 2-6 所示。

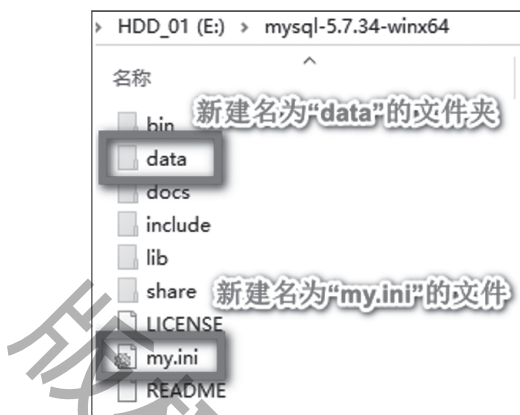


图 2-6 解压文件并创建目录

解压后的安装目录中各文件的含义和作用：

(1) bin 文件夹：用于放置一些可执行文件，如命令行客户端工具 `mysql.exe`、MySQL 服务程序 `mysqld.exe`、显示 MySQL 数据库结构 `mysqlshow.exe` 等。

(2) docx 文件夹：用于存放一些文档，如 `ChangeLog` 等。

(3) include 文件夹：用于放置一些头文件，如 `mysql.h`、`mysql_version.h` 等。

(4) lib 文件夹：用于放置一系列库文件。

(5) share 文件夹：用于存放字符集、语言等信息。

(6) README 文件：介绍版权、版本等信息。

接下来，在“MySQL 的根目录”内新建一个名为“data”的文件夹和一个名为“my.ini”的文件。“data”文件夹用于放置一些日志文件以及数据库。我们创建和保存的数据都存在这个目录里。“my.ini”文件是 MySQL 默认使用的配置文件，一般情况下，只要修改 `my.ini` 配置文件中的内容就可以对 MySQL 进行配置。

创建完毕后使用记事本或 `Sublime_text` 等打开“my.ini”文件，并输入以下内容后进行保存。其中各行代码的含义如表 2-1 所示。

```
[mysql]
default-character-set=gbk
[mysqld]
port = 3306
basedir=E:\mysql-5.7.34-winx64
datadir=E:\mysql-5.7.34-winx64\data
max_connections=200
character-set-server=gbk
default-storage-engine=INNODB
skip-grant-tables
```

表 2-1 “my.ini”文件中各行代码含义

笔记

代码	含义
default-character-set=	设置 mysql 客户端默认字符集
port =	设置端口号
basedir=	设置 mysql 的安装目录
datadir=	设置数据库的数据存放目录
max_connections=	设置允许的最大连接数
character-set-server=	设置服务端使用的默认字符集
default-storage-engine=	设置默认存储引擎
skip-grant-tables	跳过登录数据库时的验证

各行等号后面的内容均为上述各项对应的值。在实际使用时，除端口号 3306 无特殊情况不建议修改外，其他各项值均可以根据实际情况进行修改。但对于初学者来说，建议只根据 MySQL 的根目录修改“basedir=”和“datadir=”两项的值。

填写并保存“my.ini”文件后，在桌面的“此电脑”上单击右键，再单击左键弹出菜单中的“属性”，依次单击“高级系统设置|高级|环境变量”，打开“环境变量”窗口，依次弹出窗口如图 2-7 所示。

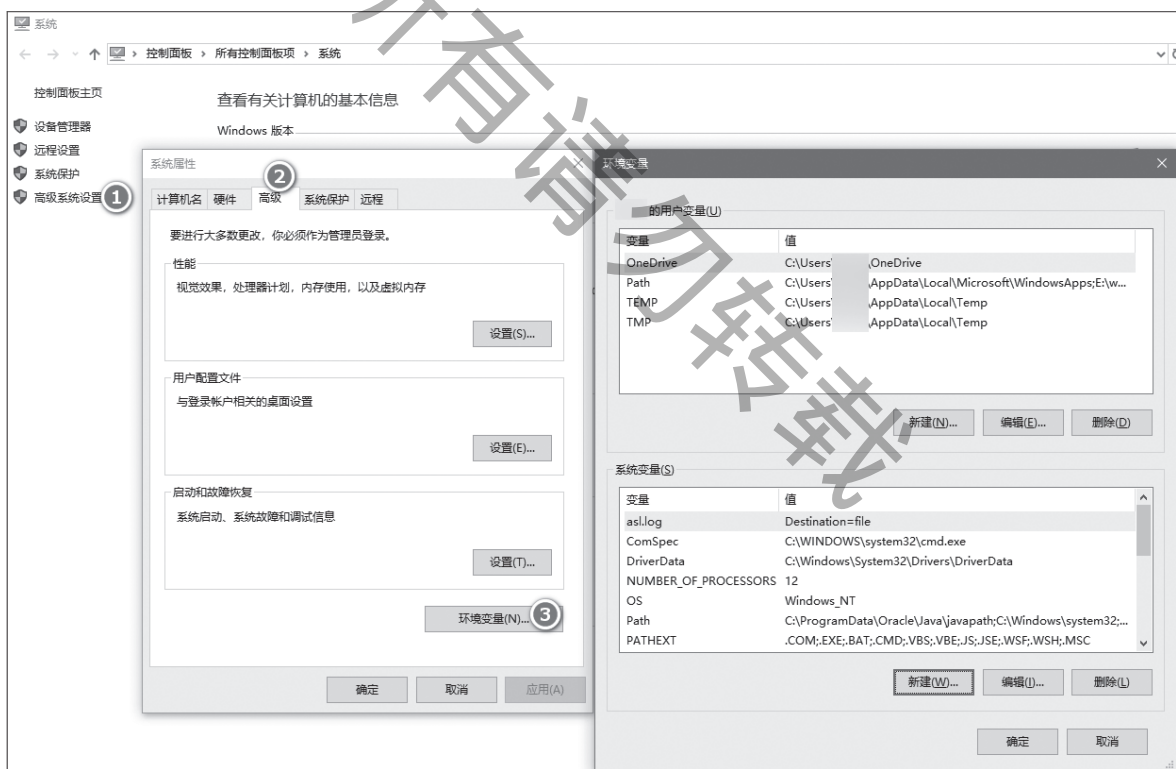


图 2-7 打开“环境变量”窗口

接下来，在“环境变量”窗口中单击“系统变量”下的“新建”按钮，并在弹出窗口中创建新的系统变量，如图 2-8 所示。其中变量名可以根据实际需要进行填写，变量值则为新建文件夹“data”和新建文件“my.ini”所在的目录，填写完毕后单击“确定”按钮。

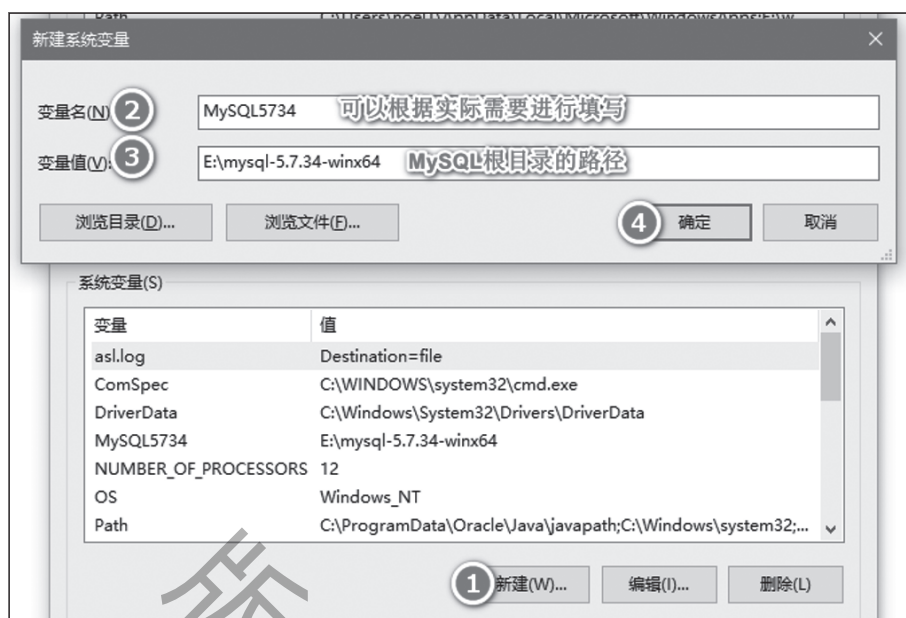


图 2-8 设置新的“环境变量”

继续在“环境变量”窗口中选中名为“Path”的变量，并单击“编辑”按钮，打开“编辑环境变量”窗口，如图 2-9 所示。

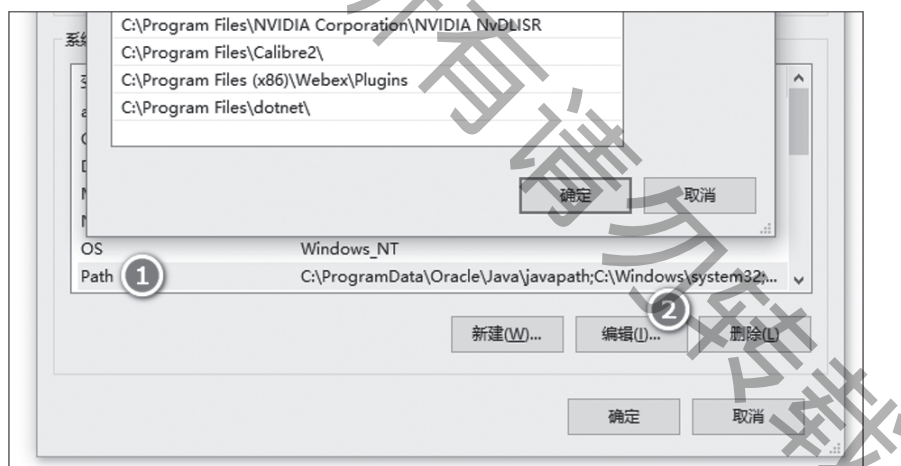


图 2-9 打开“编辑环境变量”窗口

按照如图 2-10 中标出的顺序，首先在“编辑环境变量”窗口中单击“新建”按钮，并新建环境变量“%新系统变量名%\bin”，例如，刚才新建的系统变量名为“MySQL5734”，则此处填写的内容应为“%MySQL5734%\bin”；确认填写无误后单击“确定”按钮关闭“编辑环境变量窗口”；最后单击“环境变量”窗口中的“确定”按钮，关闭“环境变量”窗口。

笔记

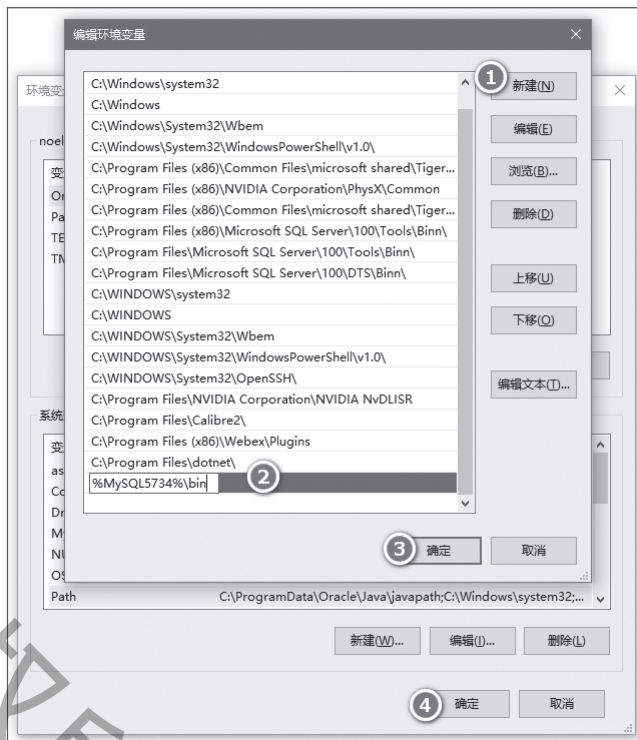


图 2-10 新建环境变量

单击“系统属性”窗口中的“确定”按钮完成系统变量配置。在开始菜单中输入“cmd”，并右键单击“命令提示符”，在弹出菜单中选择“以管理员身份运行”，如图 2-11 所示。

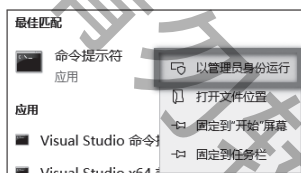


图 2-11 以管理员身份运行命令行窗口

命令行窗口打开后，将盘符切换至 E 盘，如图 2-12 所示。

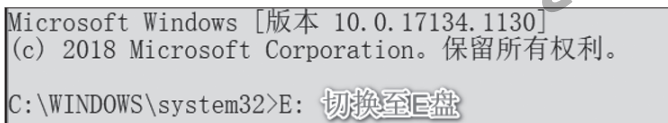


图 2-12 切换盘符至 E 盘

盘符切换成功后，利用 cd 命令将当前目录切换到 MySQL 根目录中的“bin”目录，然后输入“mysqld --initialize”命令进行初始化，如图 2-13 所示。稍作等待后若没有任何提示，则说明初始化成功。

```
E:\>cd E:\mysql-5.7.34-winx64\bin 切换到MySQL根目录中的“bin”目录下
E:\mysql-5.7.34-winx64\bin>mysqld --initialize 初始化
```

图 2-13 初始化

初始化成功后，输入“mysql --install”命令执行安装并稍作等待。当界面中出现“Service successfully installed.”时说明安装成功，如图 2-14 所示。



```
E:\mysql-5.7.34-winx64\bin>mysql --install 执行安装
Service successfully installed.
```

图 2-14 执行安装

至此，MySQL 的安装与基本配置就完成了。

2.3 关闭与启动 MySQL 服务

实际使用时，可以在命令行窗口中通过“net stop mysql”命令关闭 MySQL 服务。想要开启 MySQL 服务则可以使用“net start mysql”命令。接下来，在命令行窗口中输入“net start mysql”命令，启动 MySQL 服务，如图 2-15 所示。

```
E:\mysql-5.7.34-winx64\bin>net start mysql 启动服务
MySQL 服务正在启动。
MySQL 服务已经启动成功。
```

图 2-15 启动 MySQL 服务

2.4 用户登录与密码设置

服务启动成功后，输入“mysql -u root -p”命令进入 MySQL。若安装过程中将 MySQL 的默认端口 3306 修改为其他端口，则应在登录命令中指出修改后的端口号。例如，如果将 MySQL 的端口号改为 3308，则应使用命令“mysql -u root -P 3308 -p”进行登录，如图 2-16 所示。

```
E:\mysql-5.7.34-winx64\bin>mysql -u root -p 进入数据库
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.7.34 MySQL Community Server (GPL)

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql>
```

图 2-16 登录 MySQL

登录成功后，输入“USE MySQL”语句选择名为“MySQL”的数据库，并利用语句“update mysql.user set authentication_string=password('用户密码') where user='root';”设置数据库登录密码（适用于 5.7.XX 版本）。例如图 2-17 中将 root 用户的登录密码设置为“123123”。

```
mysql> USE MySQL; 选择名为“MySQL”的数据库
Database changed
mysql> update mysql.user set authentication_string=password('123123') where user='root';
Query OK, 1 row affected, 1 warning (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 1 设置数据库密码为“123123”
```

图 2-17 设置 MySQL 登录密码

笔记

密码设置成功后，即可使用“exit”命令退出数据库。接下来再次打开 MySQL 根目录中的“my.ini”文件，删除最后一行用于跳过登录数据库时验证的语句（skip-grant-tables），并保存。完成上述操作后，在命令行窗口中先使用“net stop mysql”命令停止服务，服务成功停止后再次使用“net start mysql”命令启动服务。此时，就可以使用刚刚设置的密码正常登录 MySQL 了。

2.5 基本功能测试

目前为止，我们已经完成了 MySQL 的安装与配置，并成功设置了登录密码。接下来，需要对 MySQL 进行简单的输入输出测试，以确保其可用性。

首先在命令行窗口中输入“mysql -u root -p”命令登录 MySQL，然后输入刚才设置的密码，并按回车后进入 MySQL。正常进入后使用如下代码测试数据库创建，数据表创建，数字、英文、中文的输入输出，以及查询等基本功能。

```
mysql> CREATE DATABASE test;
mysql> USE test
mysql> CREATE TABLE test01 (
  -> 编号 int,
  -> Department varchar(255),
  -> 姓名 varchar(255),
  -> Gender enum('男','女')
  ->);
mysql> INSERT INTO test01
  -> (编号, Department, 姓名, Gender)
  -> VALUES
  -> (1, 'Computer Science', '陈宏啸', '男'),
  -> (2, '英语', 'Lucy', '女');
mysql> SELECT * FROM test01;
```

上述代码全部输入完成并执行后，若能看到如下内容，则说明各项基本功能完好。

```
+-----+-----+-----+-----+
|编号| Department | 姓名 | Gender|
+-----+-----+-----+-----+
| 1 | Computer Science | 陈宏啸 | 男 |
| 2 | 英语 | Lucy | 女 |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

2.6 MySQL 客户端的相关命令

使用命令行客户端工具登录 MySQL 数据库后，可以在命令行中输入“help”或者“\h”命令，查看 MySQL 帮助信息，帮助学习者更好地使用 MySQL。表 2-2 列出了 MySQL 中常用的相关命令，这些命令既可以使用一个单词来表示，也可以通过“\字母”的方式来表示。

表 2-2 MySQL 相关命令

命令	缩写形式	具体含义
?	\?	显示所有 MySQL 命令
clear	\c	清除当前输入信息
connect	\r	重新连接服务器, 可选参数是 db 和 host
delimiter	\d	SQL 语句分隔符, 默认是“;”
edit	\e	用 \$EDITOR 编辑器输入命令, 仅 Linux 有效
ego	\G	发送命令到服务器, 垂直显示结果
exit	\q	退出 MySQL, 和 Quit 作用一样
go	\g	发送命令到服务器
help	\h	显示所有 MySQL 命令
nopager	\n	禁止 Pager 输出, 输出到控制台
notee	\t	不将数据输出到文件
pager	\P	设置 Pager 输出命令名称
print	\p	仅输出当前 SQL 语句, 不执行
prompt	\R	修改 MySQL 命令行提示符
quit	\q	退出 MySQL
rehash	\#	表名自动补全
source	\.	执行 SQL 脚本
status	\s	获取服务器状态信息
system	\!	执行系统 shell 命令, 仅 Linux 有效
tee	\T	设置输出结果到文件
use	\u	改变当前数据库
charset	\C	修改字符集
warnings	\W	显示警告
nowarning	\w	不显示警告
resetconnection	\x	清除会话状态信息



笔记

任务实践 >

根据“学习任务”中的实践需求, 参照文中介绍过程, 打开 MySQL 的获取途径, 完成 MySQL 的安装与配置, 并为 Root 用户设置密码, 利用新设置的密码登录 MySQL 服务器。

思考与练习

一、选择题

1. MySQL 安装完毕后为什么要配置环境变量_____。

(A) 为了在任何地方使用 MySQL 命令



- (B) 为了启动 MySQL 服务
 - (C) 为了关闭 MySQL 服务
 - (D) 为了安装连接软件
2. 下列关于 MySQL 的安装和配置正确的是_____。【多选】
- (A) MySQL 在首次启动前必须要初始化数据库
 - (B) MySQL 不仅可以在 Linux 系统上安装，也可以在 Windows 系统上安装
 - (C) MySQL 是最流行的关系型数据库管理系统
 - (D) 在 Centos 系统上我们可以安装 RPM 包，也可以使用源码包安装

二、填空题

1. MySQL 配置文件的文件名是_____。
2. MySQL 配置文件中，_____用于指定数据库文件的保存目录。

版权所有 请勿转载

第 3 章

创建和管理数据库

学习成果目标

- 1 掌握创建数据库的方法。
- 2 熟悉查看数据库的方法。
- 3 掌握修改数据库的方法。
- 4 掌握删除数据库的方法。
- 5 掌握选择数据库的方法。
- 6 培养学生的工程意识和实践能力。

学习任务

随着信息化技术的不断发展，学校准备开发一个“教务管理系统”，以提升学校日常教务管理水平。围绕“教务管理系统”项目，了解数据库的组成，熟悉数据库构建的相关知识点，完成教务管理系统数据库“EduSys”的创建和管理过程。

学习内容

数据库是按照数据结构来组织、存储和管理数据的仓库，是存储在一起的相关数据的集合。数据库是一个专门存储数据对象的容器，这些对象包括表、视图、索引、函数、存储过程、触发器等，其中，表是最基本的数据对象，是存放数据的实体。每一个数据库都有唯一的名称，数据库命名应具有实际意义并且遵循命名规则，这样可以帮使用者清楚地知道每个数据库用来存放什么数据。MySQL 数据库包括系统数据库和自定义数据库，系统数据库是在安装 MySQL 后系统自带的数据库，自定义数据库是由用户定义创建的数据库。

本章的内容结构如下：

