

 高等教育规划教材

MySQL 数据库 基础与实践

夏辉 白萍 李晋 屈巍 编著



免费提供电子教案、源代码、习题答案

下载网址 <http://www.cmpedu.com>



机械工业出版社
CHINA MACHINE PRESS



高等教育规划教材

MySQL 数据库基础与实践

夏辉 白萍 李晋 屈巍 编著



机械工业出版社

本书从实用的角度出发,全面讲解 MySQL 数据库技术。在内容安排上由浅入深,让读者循序渐进地掌握编程技术;在内容形式上附有大量的注解、说明等栏目,以提高读者的编程技术,丰富读者的编程经验。全书共分四大部分,第1部分为数据库设计基础部分;第2部分介绍数据库设计,包括 MySQL 数据库管理表记录、检索表记录、数据库设计视图和触发器、以及常见函数等;第3部分介绍 MySQL 数据库的一些高级特性,主要包括事务管理,以及 MySQL 连接器 JDBC 和连接池;第4部分介绍 Hibernate 框架。每章均配有习题,最后一章还有一个综合案例,以指导读者深入地进行学习。

本书附有所有程序的源代码、多媒体教学PPT、程序开发资源库和课后习题答案。其中,源代码全部经过精心测试,能够在 Windows XP、Windows 7 和 Windows 10 操作系统上编译和运行。

本书既可作为高等学校计算机软件技术课程的教材,也可作为管理信息系统开发人员的技术参考书。

本书配套授课电子课件,需要的教师可登录 www.cmpedu.com 免费注册,审核通过后下载,或联系编辑索取(QQ: 2850823885, 电话: 010-88379739)。

图书在版编目(CIP)数据

MySQL 数据库基础与实践 / 夏辉等编著. —北京: 机械工业出版社, 2017.3
高等教育规划教材

ISBN 978-7-111-56699-1

I. ①M… II. ①夏… III. ①SQL 语言—高等学校—教材
IV. ①TP311.132.3

中国版本图书馆 CIP 数据核字(2017)第 091293 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

责任编辑: 郝建伟 责任校对: 张艳霞

责任印制: 常天培

涿州市京南印刷厂印刷

2017 年 6 月第 1 版·第 1 次印刷

184mm×260mm·19.5 印张·474 千字

0001—3000 册

标准书号: ISBN 978-7-111-56699-1

定价: 55.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务

网络服务

服务咨询热线: (010) 88379833

机工官网: www.cmpbook.com

读者购书热线: (010) 88379649

机工官博: weibo.com/cmp1952

教育服务网: www.cmpedu.com

封面无防伪标均为盗版

金书网: www.golden-book.com

出版说明

当前,我国正处在加快转变经济发展方式、推动产业转型升级的关键时期。为经济转型升级提供高层次人才,是高等院校最重要的历史使命和战略任务之一。高等教育要培养基础性、学术型人才,但更重要的是加大力度培养多规格、多样化的应用型、复合型人才。

为顺应高等教育迅猛发展的趋势,配合高等院校的教学改革,满足高质量高校教材的迫切需求,机械工业出版社邀请了全国多所高等院校的专家、一线教师及教务部门,通过充分的调研和讨论,针对相关课程的特点,总结教学中的实践经验,组织出版了这套“高等教育规划教材”。

本套教材具有以下特点:

1) 符合高等院校各专业人才的培养目标及课程体系的设置,注重培养学生的应用能力,加大案例篇幅或实训内容,强调知识、能力与素质的综合训练。

2) 针对多数学生的学习特点,采用通俗易懂的方法讲解知识,逻辑性强、层次分明、叙述准确而精炼、图文并茂,使学生可以快速掌握,学以致用。

3) 凝结一线骨干教师的课程改革和教学研究成果,融合先进的教学理念,在教学内容和方法上做出创新。

4) 为了体现建设“立体化”精品教材的宗旨,本套教材为主干课程配备了电子教案、学习与上机指导、习题解答、源代码或源程序、教学大纲、课程设计和毕业设计指导等资源。

5) 注重教材的实用性、通用性,适合各类高等院校、高等职业学校及相关院校的教学,也可作为各类培训班教材和自学用书。

欢迎教育界的专家和老师提出宝贵的意见和建议。衷心感谢广大教育工作者和读者的支持与帮助!

机械工业出版社

前 言

MySQL 数据库是世界上最流行的数据库之一。MySQL 是一款非常优秀的免费软件，由瑞士的 MySQL AB 公司开发，是一款真正的快速、多用户、多线程的 SQL 数据库。全球最大的网络搜索引擎公司——Google 使用的数据库就是 MySQL，并且国内很多大型网络公司也选择 MySQL 数据库，如百度、网易和新浪等。据统计，世界上一流的互联网公司中，排名前 20 位的有 80% 是 MySQL 的忠实用户。学习和掌握 MySQL 数据库技术语言已经成为计算机相关专业学生的迫切需求。

本书讲解了 MySQL 开发基础和数据库编程技巧，在内容的编排上力争体现新的教学思想和方法。本书的内容编写遵循“从简单到复杂”“从抽象到具体”的原则。书中通过各个章节穿插了很多实例，提供了 MySQL 从入门到实际应用所必备的知识。数据库设计既是一门理论课，也是一门实践课。学生除了要在课堂上学习程序设计的理论方法，掌握编程语言的语法知识和编程技巧外，还要进行大量的课外练习和实践操作。为此，本书每章都配有课后习题，并且每章都有一个综合案例，除此之外，每章还安排了实验的题目，可供教师实验教学使用。

本书共分 10 章。第 1 章介绍数据库设计基础，主要介绍数据库开发的基本概念及专用术语。第 2 章为 MySQL 数据库概述，主要介绍 MySQL 数据库安装、数据库的操作，以及数据表结构的操作。第 3 章介绍 MySQL 管理表记录，主要包括基本数据类型、运算符、字符集和数据表的操作。第 4 章介绍检索表记录，主要讲解利用各种不同方式进行条件查询表记录。第 5 章介绍视图和触发器。第 6 章介绍事务管理。第 7 章介绍 MySQL 连接器 JDBC 和连接池。第 8 章介绍 Hibernate 框架。第 9 章介绍常见函数和数据管理。第 10 章介绍了一个综合案例，通过这个综合案例可以加深读者对 MySQL 数据库的认识。

本书内容全面，案例新颖，针对性强。本书所介绍的实例都是在 Windows 10 操作系统下调试运行通过的。每章都配有与本章知识点相关的案例和实验，以帮助读者顺利地完成任务。从应用程序的设计到应用程序的发布，读者都可以按照书中所讲述的内容来实施。作为教材，每章后面均附有习题。

本书由夏辉负责全书的整体策划，夏辉、白萍、李晋和屈巍负责全书的编写，并且最终完成书稿的修订、完善、统稿和定稿工作，由王晓薇教授、吴鹏博士负责主审。刘杰教授、李航教授为本书的策划和编写提供了有益的帮助和支持，并且对本书初稿在教学过程中存在的问题提出了宝贵的意见。本书也借鉴了中外参考文献中的原理知识和资料，在此一并表示感谢。

本书配有电子课件、课后习题答案、每章节案例代码和实验代码，以方便教学和自学参考使用，如有需要请到 <http://www.scse.sdu.edu.cn> 网络中下载。

由于时间仓促，书中难免存在不妥之处，敬请广大读者原谅，并提出宝贵意见。

编 者

目 录

出版说明

前言

第1章 数据库设计基础	1	2.4.1 创建数据表	30
1.1 数据库设计概述	1	2.4.2 查看数据库表结构	33
1.1.1 关系数据库概述	1	2.4.3 修改表结构	33
1.1.2 结构化查询语言 SQL	2	2.4.4 删除数据库表	36
1.1.3 数据库设计的基本步骤	3	2.5 MySQL 存储引擎	37
1.2 关系模型	5	2.5.1 InnoDB 存储引擎	37
1.2.1 数据库和表	5	2.5.2 MyISAM 存储引擎	38
1.2.2 列和行	6	2.5.3 存储引擎的选择	38
1.2.3 主键与外键	6	2.6 案例：网上书店系统	39
1.2.4 约束	7	本章总结	42
1.3 E-R 图	9	实践与练习	42
1.3.1 实体和属性	9	实验指导：学生选课系统数据库设计	43
1.3.2 实体与属性之间的关系	10	第3章 MySQL 管理表记录	45
1.3.3 E-R 图的设计原则	11	3.1 MySQL 的基本数据类型	45
本章总结	12	3.1.1 整数类型	45
实践与练习	12	3.1.2 小数类型	47
实验指导：E-R 图的设计与画法	13	3.1.3 字符串类型	48
第2章 MySQL 数据库概述	15	3.1.4 日期时间类型	50
2.1 认识 MySQL 数据库	15	3.1.5 复合类型	52
2.1.1 MySQL 简介	15	3.1.6 二进制类型	53
2.1.2 MySQL 体系结构	15	3.2 MySQL 运算符	54
2.2 MySQL 数据库的安装和配置	16	3.2.1 算术运算符	54
2.2.1 MySQL 的安装和配置服务	16	3.2.2 比较运算符	55
2.2.2 启动和停止服务	23	3.2.3 逻辑运算符	56
2.2.3 MySQL 加入环境变量	24	3.2.4 位运算符	57
2.2.4 连接 MySQL 服务器	25	3.2.5 运算符的优先级	58
2.2.5 MySQL 可视化操作工具	26	3.3 字符集设置	59
2.3 MySQL 数据库的基本操作	28	3.3.1 MySQL 字符集与字符排序规则	59
2.3.1 创建数据库	28	3.3.2 MySQL 字符集的设置	60
2.3.2 查看数据库	29	3.4 增添表记录	61
2.3.3 显示数据库	29	3.4.1 INSERT 语句	61
2.3.4 选择当前数据库	30	3.4.2 REPLACE 语句	65
2.3.5 删除数据库	30	3.5 修改表记录	66
2.4 MySQL 数据库表结构的操作	30	3.6 删除表记录	66

3.6.1 DELETE 删除表记录	66	实验指导：学生选课系统数据库检索	102
3.6.2 TRUNCATE 清空表记录	67	第5章 视图和触发器	104
3.7 案例：图书管理系统中表记录的操作	69	5.1 视图	104
本章总结	73	5.1.1 创建视图	105
实践与练习	73	5.1.2 查看视图	110
实验指导	74	5.1.3 管理视图	112
实验1 MySQL中字符集的设置	75	5.1.4 使用视图	113
实验2 数据表中记录的插入、修改和删除	76	5.2 触发器的使用	116
第4章 检索表记录	78	5.2.1 创建并使用触发器	117
4.1 SELECT基本查询	78	5.2.2 查看触发器	119
4.1.1 SELECT··FROM查询语句	78	5.2.3 删除触发器	120
4.1.2 查询指定字段信息	79	5.2.4 触发器的应用	120
4.1.3 关键字 DISTINCT 的使用	80	5.3 案例：在删除分类时自动删除分类对应的消息记录	124
4.1.4 ORDER BY子句的使用	81	本章总结	128
4.1.5 LIMIT子句的使用	81	实践与练习	128
4.2 条件查询	82	实验指导：视图、触发器的创建与管理	129
4.2.1 使用关系表达式查询	83	第6章 事务管理	131
4.2.2 使用逻辑表达式查询	83	6.1 事务机制概述	131
4.2.3 设置取值范围的查询	84	6.2 事务的提交和回滚	132
4.2.4 空值查询	84	6.2.1 事务的提交	132
4.2.5 模糊查询	85	6.2.2 事务的回滚	134
4.3 分组查询	86	6.3 事务的四大特性和隔离级别	136
4.3.1 GROUP BY子句	87	6.3.1 事务的四大特性	136
4.3.2 HAVING子句	88	6.3.2 事务的隔离级别	139
4.4 表的连接	88	6.4 解决多用户使用问题	139
4.4.1 内连接	89	6.4.1 脏读	139
4.4.2 外连接	91	6.4.2 不可重复读	141
4.4.3 自连接	92	6.4.3 幻读	142
4.4.4 交叉连接	93	6.5 案例：银行转账业务的事务处理	143
4.5 子查询	93	本章总结	146
4.5.1 返回单行的子查询	93	实践与练习	146
4.5.2 返回多行的子查询	94	实验指导：MySQL中的事务管理	147
4.5.3 子查询与数据更新	96	第7章 MySQL连接器JDBC和连接池	148
4.6 联合查询	98	7.1 JDBC	148
4.7 案例：网上书店系统综合查询	99	7.2 JDBC连接过程	149
本章总结	101		
实践与练习	101		

7.3 JDBC 数据库操作	155	第 9 章 常见函数和数据管理	213
7.3.1 增加数据	155	9.1 常见函数	213
7.3.2 修改数据	156	9.1.1 数学函数	213
7.3.3 删除数据	157	9.1.2 字符串函数	218
7.3.4 查询数据	157	9.1.3 时间日期函数	223
7.3.5 批处理	159	9.1.4 数据类型转换函数	227
7.4 数据源	162	9.1.5 控制流程函数	227
7.5 案例: 分页查询大型数据库	164	9.1.6 系统信息函数	228
本章总结	167	9.2 数据库备份与还原	229
实践与练习	168	9.2.1 数据的备份	229
实验指导: 学生选课系统数据库		9.2.2 数据的还原	232
操作	169	9.3 MySQL 的用户管理	234
第 8 章 Hibernate 框架介绍	172	9.3.1 数据库用户管理	234
8.1 Hibernate 简介	172	9.3.2 用户权限设置	238
8.2 Hibernate 原理	172	9.4 案例: 数据库备份与恢复	242
8.3 Hibernate 的工作流程	174	本章总结	251
8.4 Hibernate 的核心组件	175	实践与练习	251
8.4.1 Configuration 接口	175	实验指导: 数据库安全管理	252
8.4.2 SessionFactory 接口	175	第 10 章 综合案例——图书管理系统	254
8.4.3 Session 接口	176	10.1 系统需求分析	254
8.4.4 Transaction 接口	178	10.2 数据库设计	254
8.4.5 Query 接口	178	10.3 数据库表的创建	256
8.4.6 Criteria 接口	180	10.4 系统实现	259
8.5 Hibernate 框架的配置过程	182	10.4.1 使用 JDBC 访问 MySQL	
8.5.1 导入相关 jar 包	182	数据库	259
8.5.2 创建数据库及表	183	10.4.2 管理员登录	261
8.5.3 创建实体类 (持久化类)	184	10.4.3 系统参数设置	265
8.5.4 配置映射文件	185	10.4.4 图书基本信息管理	271
8.5.5 配置主配置文件	186	10.4.5 图书馆藏信息管理	277
8.5.6 编写数据库	187	10.4.6 图书借阅管理	280
8.6 Hibernate 的关系映射	196	10.4.7 图书归还管理	287
8.7 案例: 人事管理系统数据库	201	10.4.8 读者登录	292
本章总结	209	10.4.9 读者信息查询	294
实践与练习	209	10.4.10 读者图书查询	298
实验指导: Hibernate 框架的持久		本章总结	303
层数据操作	210	参考文献	304

第1章 数据库设计基础

当今的时代是一个信息化的时代，信息成为一种重要的战略资源，对信息的占有和利用成为衡量一个国家、地区、组织或企业综合实力的一项重要指标。随着社会各行各业信息化的飞速发展，人们的知识也以惊人的速度增长，如何有效地组织和利用如此庞大的知识，以及如何合理地管理和维护如此海量的信息，都要依靠数据库。数据库技术是数据管理的核心技术，主要研究如何科学地组织、存储和管理数据库中的数据，以提供可共享、安全、可靠的数据。

1.1 数据库设计概述

数据库 (Database, DB) 是“按照某种数据结构对数据进行组织、存储和管理的容器”，简单地说就是用来存储和管理数据的容器。数据库系统 (Database System, DBS) 是指在计算机中引入数据库后的系统，一般由数据库、数据库管理系统、应用程序和数据库管理员组成。数据库管理系统 (Database Management System, DBMS) 是一个管理、控制数据库容器中各种数据库对象的系统软件。数据库用户无法直接通过操作系统获取数据库文件中的具体内容，数据库管理系统则可以通过调用操作系统的进程管理、内存管理、设备管理及文件管理等服务，为数据库用户提供管理、控制数据库中各种数据库对象和数据库文件的接口，从而实现对数据库中具体内容的获取。数据库管理系统按照一定的数据模型组织数据，常用的模型包括“层次模型”“网状模型”“关系模型”及“面向对象模型”等，基于“关系模型”的数据库管理系统称为关系数据库管理系统 (Relational Database Management System, RDBMS)。

1.1.1 关系数据库概述

关系数据库的概念是由 E.F.Codd 博士于 1976 年发表的《关于大型共享数据库数据的关系模型》论文中提出的，论文中阐述了关系数据库模型及其原理，并将其用于数据库系统。

使用关系模型对数据进行组织、存储和管理的数据库称为关系数据库，关系数据库系统是支持关系数据模型的数据库系统。在关系数据库中，所谓的“关系”，实际上是一张二维表，表是逻辑结构而不是物理结构，系统在物理层可以使用任何有效的存储结构来存储数据，如顺序文件、索引、哈希表和指针等，因此，表是对物理存储数据的一种抽象表示，即对许多存储细节的抽象，如存储记录的位置、记录的顺序、数据值的表示，以及记录的访问结构（如索引）等。

关系数据库要求将每个具有相同属性的数据独立存放在一张表中，克服了层次数据库横向关联不足的缺点，也避免了网状数据库关联过于复杂的问题，因此被广泛应用。

1.1.2 结构化查询语言 SQL

结构化查询语言 (Structured Query Language, SQL) 是一种专门用来与数据库通信的语言, 其利用一些简单的句子构成基本的语法来存取数据库中的内容, 便于用户从数据库中获得和操作所需数据。例如, 删除“选课系统”中课程表 (course) 的所有记录, 一条 `delete from course` 语句就可以做到。

SQL 语言具有以下特点。

1) SQL 语言是非过程化语言。

SQL 语言允许用户在高层的数据结构上工作, 而不对单个记录进行操作, 可以操作记录集。SQL 语句接受集合作为输入, 返回集合作为输出。在 SQL 语言中, 用户只需要在程序中说明“做什么”, 无须说明“怎样做”, 即无须用户指定对数据存放的方法。

2) SQL 语言是统一的语言。

SQL 语言适用于所有用户的数据活动类型, 即 SQL 语言可用于所有用户, 包括系统管理员、数据库管理员、应用程序员、决策支持系统人员, 以及许多其他类型的终端用户对数据库等数据对象的定义、操作和控制活动。

3) SQL 语言是关系数据库的公共语言。

用户可将使用 SQL 的应用从一个关系型数据库管理系统转移到另一系统。

SQL 语言由 4 部分组成。

1) 数据定义语言 (Data Definition Language, DDL)。

DDL 用来定义数据的结构, 是对数据的格式和形态进行定义的语言, 主要用于创建、修改和删除数据库对象、表、索引、视图及角色等, 常用的数据定义语言有 CREATE、ALTER 和 DROP 等。

每个数据库要建立时首先要面对一些问题, 例如, 数据与哪些表有关、表内有什么栏目主键, 以及表与表之间互相参照的关系等, 这些都需要在设计开始时就预先规划好, 所以, DDL 是数据库管理员和数据库所有者才有权操作的用于生成与改变存储结构的命令语句。

2) 数据操纵语言 (Data Manipulation Language, DML)。

DML 用于读取和操纵数据, 数据定义完成后接下来就是对数据的操作。数据的操作主要有插入数据 (insert)、查询数据 (query)、更改数据 (update) 和删除数据 (delete) 4 种方式, 即数据操纵主要用于数据的更新、插入等操作。

3) 数据控制语言 (Data Control Language, DCL)。

DCL 用于安全性控制, 如权限管理、定义数据访问权限、进行完整性规则描述及事务控制等, 其主要内容包括以下 3 方面。

- 用来授予或回收操作数据库的某种特权。
- 控制数据库操纵事务发生的时间及效果。
- 对数据库实行监视。

4) 嵌入式 SQL 语言的使用规定。

嵌入式 SQL 语言 (Embed SQL) 主要涉及 SQL 语句嵌入在宿主语言程序中的规则。SQL 通常有两种使用方式: 练级交互使用方式 (命令方式) 和嵌入某种高级程序设计语言的程序中 (嵌入方式), 两种使用方法虽然不同, 但是 SQL 语言的语法结构一致。

根据 SQL 语言的 4 个组成部分可以得到 SQL 的数据定义、数据查询、数据操纵及数据控制的 4 个基本功能，表 1-1 列出了实现其功能的主要动词。

表 1-1 SQL 功能及包含的主要动词

SQL 功能	动 词
数据定义	CREATE、DROP、ALTER
数据查询	SELECT
数据操纵	INSERT、UPDATE、DELETE
数据控制	GRANT、REVOKE

经过多年发展，SQL 成为一种应用最广泛的关系数据库语言，并定义了操作关系数据库的标准语法。为了实现更强大的功能，各关系数据库管理系统通过增加语句或指令的方式对 SQL 标准进行了各自的扩展，如 Oracle 的 PL/SQL、SQL Server 的 T-SQL 等。MySQL 也对 SQL 标准进行了扩展，如 MySQL 命令“show database;”用于查询当前 MySQL 服务实例所有的数据库名。

为了区分 SQL 扩展与 SQL 标准，本书将符合 SQL 标准的代码称为“SQL 语句”，如 delete from course，把 MySQL 对 SQL 标准进行的扩展代码称为“MySQL 命令”，如“show database;”。

1.1.3 数据库设计的基本步骤

按照规范设计的方法，同时考虑数据库及其应用系统开发的全过程，可以将数据库设计分为以下 6 个阶段。

1. 需求分析阶段

需求分析是数据库设计的第一步，也是整个设计过程的基础，本阶段的主要任务是对现实世界要处理的对象（公司、部门及企业）进行详细调查，在了解现行系统的概况、确定新系统功能的过程中，收集支持系统目标的基础数据及其处理方法。

需求分析是在用户调查的基础上，通过分析，逐步明确用户对系统的需求，包括数据需求和围绕这些数据的业务处理需求。用户调查的重点是“数据”和“处理”。通过调查要从用户处获得对数据库的下列要求。

- 信息需求。定义所设计数据库系统用到的所有信息，明确用户将向数据库中输入什么样的数据，从数据库中要求获得什么样的内容，将要输出什么信息。即明确在数据库中需要存储什么数据，对这些数据将做什么处理等，同时还需要描述数据之间的联系。
- 处理需求。定义系统数据处理的操作功能，描述操作的优先次序，包括操作的执行频率和场合，操作与数据间的联系。要明确用户需要完成哪些处理功能，每种处理的执行频度，用户需求的响应时间，以及处理方式等。
- 安全性与完整性要求。安全性要求描述系统中不同用户对数据库的使用和操作情况，完整性要求描述数据之间的管理关系及数据的取值范围要求。

在数据分析阶段不必确定数据的具体存储方式，这些问题留待进行物理结构设计时再考虑。需求分析是整个数据库设计中最重要的一步，为后续的各个阶段提供充足的信息，如

果把整个数据库设计看做一个系统工程，那么需求分析就是该系统工程最原始的输入信息，需求分析不充分，会导致整个数据库重新返工。

2. 概念结构设计阶段

概念结构设计阶段是整个数据库设计的关键。通过对用户需求进行综合、归纳与抽象，形成一个独立于具体 DBMS 的概念模型。

概念结构设计的策略主要有以下几种。

- 自底向上。先定义每个局部应用的概念结构，然后按一定的规则把它们集成起来，从而得到全局概念结构。
- 自顶向下。先定义全局概念结构，然后逐步细化。
- 自内向外。先定义最重要的核心结构，然后逐步向往扩展。
- 混合策略。先用自顶向下的方法设计一个概念结构的框架，然后以它为框架再用自底向上策略设计局部概念结构，最后集成。

3. 逻辑结构设计阶段

逻辑结构设计阶段将概念结构转换为某个 DBMS 所支持的数据模型，并将其性能进行优化。

逻辑结构设计一般包含两步。

- 将概念结构转换为某种组织层数据模型。
- 对组织层数据模型进行优化。

4. 数据库物理结构设计阶段

数据库物理结构设计阶段是利用数据库管理系统提供的方法和技术，对已经确定的数据库逻辑结构，以较优的存储结构、数据存取路径、合理的数据存储位置及存储分配，设计出一个高效的、可实现的物理数据库结构。

数据库物理结构设计通常分两步。

- 确定数据库的物理结构，在关系数据库中主要指存取方法和存储结构。
- 对物理结构进行评价，评价的重点是时间和空间效率。

如果评价的结果满足原设计要求，则可以进入数据库实施阶段，否则，需要重新设计或修改物理结构，有时甚至需要返回到逻辑设计阶段修改数据模式。

若物理数据库设计得合理，可以使事务的响应时间短，存储空间利用率高，事务吞吐量大。在设计数据库时，首先要对经常用到的查询和对数据进行更新的事务进行详细的分析，获得物理结构设计所需的各种参数。其次，要充分了解所使用的 DBMS 的内部特征，特别是系统提供的存取方法和存储结构。

通常关系数据库的物理结构设计主要包括以下内容。

- 确定数据的存取方法。
- 确定数据的存储结构。

5. 数据库实施阶段

在数据库实施阶段运用 DBMS 提供的数据库语言（如 SQL）及宿主语言（如 C），根据逻辑设计和物理设计的结果建立数据库，编制与调试应用程序，组织数据入库，并进行试运行。

6. 数据库运行与维护阶段

数据库应用系统经过试运行后即可投入正式运行，在运行过程中需要不断对其进行调

整、修改与完善。

图 1-1 给出了各阶段的设计内容及各阶段的设计描述。


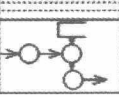

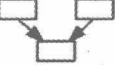
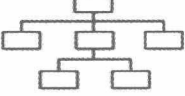
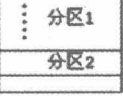
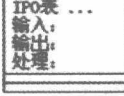
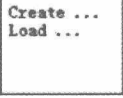
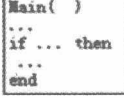
设计阶段	设计描述	
	数据	处理
需求分析	数据字典、全系统中数据项、数据流、数据存储的描述	数据流图和判定表（判定树）、数据字典中处理过程的描述
概念结构设计	概念模型（E-R图）  数据字典	系统说明书包括： ①新系统要求、方案和概图 ②反映新系统信息流的数据流图 
逻辑结构设计	某种数据模型 关系  非关系 	系统结构图（模块结构） 
物理设计	存储安排 方法选择 存取路径建立 	模块设计 IPO表 
实施阶段	编写模式 装入数据 数据库试运行 	程序编码 编译联结 测试 
运行维护	性能监测、转储、恢复 数据库重组和重构	新旧系统转换、运行、维护（修正性、适应性、改善性维护）

图 1-1 数据库设计阶段及内容描述

设计一个完善的数据库应用系统需要上述 6 个阶段的不断反复。在设计过程中，应把数据库的结构设计和数据处理的操作紧密结合起来，这两个方面的需求分析、数据抽象、系统设计及实现等各阶段应同时进行，互相参照和互相补充。

1.2 关系模型

关系模型是目前最重要的也是应用最广泛的数据模型。简而言之，关系就是一张二维表，由行和列组成。关系模型将数据模型组织成表格的形式，这种表格在数学上称为关系。

1.2.1 数据库和表

关系数据库是由多个表和其他数据库对象组成的，表是一种最基本的数据库对象，由行和列组成，类似电子表格。一个关系数据库通常包含多个二维表（称为数据库表或表），从而实现所设计的应用中各类信息的存储和维护。在关系数据库中，如果存在多个表，则表与表之间也会因为字段的关系产生关联，关联性由主键和外键所体现的参照关系实现。关系数据库不仅包含表，还包含其他数据库对象，如关系图、视图、存储过程和索引等，所以，通常提到的关系数据库就是指一些相关的表和其他数据库对象的集合。例如，表 1-2 所示的课程表中收集了教师申报课程的相关信息，包括课程名、课程编号、人数上限、授课教师、课程性质及课程状态信息，构成了一张二维表。

表 1-2 课程表（二维表实例）

课程名	课程编号	人数上限	授课教师	课程性质	状态
C 语言程序设计	16209020	60	孙老师	必修	未审核
MySQL 数据库设计	16309620	90	李老师	必修	未审核
物联网导论	16309490	40	王老师	选修	未审核
专业外语	16209101	70	田老师	必修	未审核

1.2.2 列和行

数据表中的列也称为字段，用一个列名（也称为字段名）标记。除了字段名行，表中每一行都称为一条记录。例如，表 1-2 中共有 4 个字段、4 条记录。如果想查找“MySQL 数据库设计”这门课程的授课教师，则可以查找“MySQL 数据库设计”所在的行与字段“授课教师”所在的列关联相交处获得。初看上去，关系数据库中的一个数据表与一个不存在“合并单元”的 Excel 相似，但是同一个数据表的字段名不允许重复，而且为了优化存储空间，便于数据排序，数据库表的每一列要求指定数据类型。

1.2.3 主键与外键

关系型数据库中的一个表由行和列组成，并且要求表中的每行记录必须唯一。在设计表时，可以通过定义主键（primary key）来保证记录（实体）的唯一性。一个表的主键由一个或多个字段组成，值具有唯一性，且不允许去控制，主键的作用是唯一地标识表中的每一条记录。例如，在表 1-3 中，可以用“学号”字段作为主键，但是不能使用“姓名”字段作为主键，因为存在同名现象，无法保证唯一性，有时候表中也有可能没有一个字段具有唯一性，即没有任何字段可以作为主键，这时可以考虑使用两个或两个以上字段的组合作为主键。

表 1-3 主键外键关系

学号	课程编号	成绩
14180070	16209020	98
14180071	16309620	95
14180083	16309490	87
17180086	16209101	90

为表定义主键时需要注意以下几点。

- 以取值简单的关键字作为主键。例如，如果学生表存在“学号”和“身份证号”两个字段，建议选取“学号”作为主键，对于开发人员来说“学号”的取值比“身份证号”取值简单。
- 不建议使用复合主键。在设计数据库表时，复合主键会给表的维护带来不便，因此不建议使用，对应存在复合主键的表，建议向表中添加一个没有实际意义的字段作为该表的主键。
- 以添加一个没有实际意义的字段作为表的主键的方式来解决无法从已有字段选择主

键或者存在复合主键的问题。例如，在课程表中如果没有包含“课程编号”这个字段，此时因为“课程名”可能重复，课程表就没有关键字，开发人员可以在课程表中添加一个没有实际意义的字段，如“课程号”作为该表的主键。

- 当数据库开发人员向数据库中添加一个没有实际意义的字段作为表的主键时，建议该主键的值由数据库管理系统或者应用程序自动生成，既方便、又避免了人工录入人为操作引入错误的几率。

一个关系型数据库可能包含多个表，可以通过外键（foreign key）使这些表关联起来。如果在表 A 中有一个字段对应表 B 中的主键，那么该字段称为表 A 的外键。该字段出现在表 A 中，但由它所标识的主题的详细信息存储在表 B 中，对表 A 来说这些信息是存储在表的外部的，因此称为外键。

如表 1-3 中所示的学生成绩表中有两个外键，一个是“学号”，其详细信息存储在“学生表”中；一个是“课程编号”，其详细信息存储在“课程表”中。“成绩表”和“学生表”中各有一个“学号”字段，该字段在“成绩表”中是外键，在“学生表”中则是主键，但这两个字段的数据类型及字段宽度必须保持一致，字段的名称可以相同，也可以不同。

1.2.4 约束

设计表时，可对表中的一个字段或多个字段的组合设置约束条件，由数据库管理系统（如 MySQL）自动检测输入的数据是否满足约束条件，不满足约束条件的数据将被数据库管理系统拒绝录入。约束分为表级约束和字段级约束，表级约束是对表中几个字段的约束，字段级约束是对表中一个字段的约束。几种常见的约束形式如下。

1. 主键约束

主键用来保证表中每条记录的唯一性，因此在设计数据库表时，建议为所有的数据库表都定义一个主键，用于保证数据库表中记录的唯一性。一张表只允许设置一个主键，这个主键可以是一个字段，也可以是一个字段组合（不建议使用复合主键）。单个字段作为主键时，使用字段级约束；用字段组合作为主键时，则使用表级约束。在录入数据的过程中，必须在所有主键字段中输入数据，即任何主键字段的值不允许为 null。如果不在主键字段中输入数据，或输入的数据在前面已经输入过，则这条记录将被拒绝。可以在创建表时创建主键，也可以对表已有的主键进行修改或者增加新的主键。

2. 外键约束

外键约束主要用于定义表与表之间的某种关系，对于表 A 来说，外键字段的取值是 null，或者是来自于表 B 的主键字段的取值，表 A 和表 B 必须存放在同一关系型数据库中。外键字段所在的表称为子表，主键字段所在的表称为父表，父表与子表之间通过外键字段建立起了外键约束关系，即表 A 称为表 B 的子表，表 B 称为表 A 的父表。

创建表时建议先创建父表，然后再创建子表，并且建议子表的外键字段与父表的主键字段的数据类型数据长度相似或者可以相互转换（最好相同）。子表与父表之间的外键约束关系如下。

- 如果子表的记录“参照”了父表的某条记录，则父表中该记录的删除（delete）或修改（update）操作可能以失败告终。
- 如果试图直接插入（insert）或者修改（update）子表的“外键值”，子表中的“外键

值”必须是父表中的“主键值”或者 null，否则插入 (insert) 或者修改 (update) 操作失败。

MySQL 的 InnoDB 存储引擎支持外键约束，而 MySQL 的 MyISAM 存储引擎暂时不支持外键约束。对于 MyISAM 存储引擎的表而言，数据库开发人员可以使用触发器“间接地”实现外键约束。

3. 非空约束

如果在一个字段中允许不输入数据，可以将该字段定义为 null，如果在一个字段中必须输入数据，则应当将该字段定义为 not null。如果设置某个字段的非空约束，直接在该字段的数据类型后面加上 not null 关键字即可。当一个字段中出现 null 值时，意味着用户还没有为该字段输入值，非空约束限制该字段的内容不能为空，但可以是空白，所以 null 值既不等价于数值型数据中的 0，也不等价于字符型数据中的空字符串。

4. 唯一性约束

如果一个字段值不允许重复，则应当对该字段添加唯一性 (unique) 约束。与主键约束不同，一张表中可以存在多个唯一性约束，满足唯一性约束的字段可以取 null 值。如果设置某个字段为唯一性约束，直接在该字段的数据类型后面加上 unique 关键字即可。

5. 默认约束

默认值字段用于指定一个字段的默认值，当尚未在该字段中输入数据时，该字段中将自动填入这个默认值。例如，可以为课程表 (course) 中的人数上限 (up_limit) 字段设置默认值 90，则当尚未在该字段中输入数据时，该字段会自动填入默认值。如果设置某个字段的默认值约束，直接在该字段的数据类型后面加上“default 默认值”即可。如果对一个字段添加了 not null 约束，但又没有设置默认约束，则必须在该字段中输入一个非 null 值，否则会出现错误。

6. 检查约束

检查 (check) 约束用于检查字段的输入值是否满足指定的条件，在表中输入或者修改记录时，如果不符合检查约束指定的条件，则数据不能写入该字段。例如，课程的人数上限必须在 (90, 100, 120) 整数集中取值；一个人的性别必须在 (‘男’, ‘女’) 字符串集中取值；成绩表中的成绩字段需要满足大于等于 0、并且小于等于 100 的约束条件等。这些约束条件都属于检查约束。

MySQL 暂时不支持检查 (check) 约束，数据库开发人员可以使用 MySQL 复合数据类型或者触发器“间接地”实现检查约束。

7. 自增约束

自增 (AUTO_INCREMENT) 约束是 MySQL 唯一扩展的完整性约束，当向数据库表中插入新记录时，字段上的值会自动生成唯一的 ID。在具体设置自增约束时，一个数据库表中只能有一个字段使用该约束，该字段数据类型必须是整型类型。由于设置自增约束后的字段会生成唯一的 ID，所以该字段也经常被设置为主键。MySQL 中通过 SQL 语句的 AUTO_INCREMENT 来实现。

8. 删除约束

在 MySQL 数据库中，一个字段的所有约束都可以用 alter table 命令删除。

1.3 E-R 图

关系数据库设计一般要从数据模型 E-R 图 (Entity-Relationship Diagram, E-R 图) 设计开始。E-R 图既可以表示现实世界中的事物, 又可以表示事物之间的关系, 它描述了软件系统的数据存储需求, 其中 E 表示实体, R 表示关系, 所以 E-R 图由实体、属性和关系 3 个要素构成, 通过一组与实体、属性和关系相关的概念可以很好地描述信息世界。

1.3.1 实体和属性

1. 实体

E-R 图中的实体表示现实世界具有相同属性描述的事物的集合, 它不是某一个具体事物, 而是一类事物的统称。E-R 图中的实体通常用矩形表示, 如图 1-2 所示, 把实体名写在矩形框内, 实体中的每一个具体的记录值称为该实体的一个实例。在设计 E-R 图时, 一个 E-R 图中通常包含多个实体, 每个实体由实体名唯一标记。开发数据库时每个实体对应于数据库中的一张数据库表, 每个实体的具体取值对应于数据库表中的一条记录。例如, 在“选课系统”中, “课程”是一个实体, “课程”实体对应于“课程”数据库表, 而“课程名”为 MySQL 数据库设计, “人数上限”为 90 的课程是课程实体的具体取值, 对应于“课程”数据库表中的一条记录。

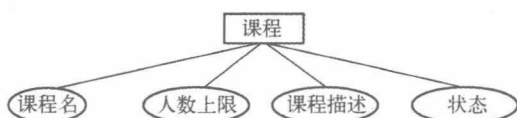


图 1-2 课程实体及属性

2. 属性

E-R 图中的属性通常表示实体的某种特征, 也可以使用属性表示实体间关系的特征。一个实体通常包含多个属性, 每个属性由属性名唯一标记, 画在椭圆内, 如图 1-2 所示, “课程”实体包含“课程名”“人数上限”“课程描述”“状态”4 个属性。再如图 1-3 所示, “学生实体”可以由“学号”“姓名”“性别”“出生年月”“专业”和“联系方式”等属性组成, 而 (14180070, 李天, 男, 1990-08, 计算机科学与技术, 2014) 具体描述了一个名叫李天的学生对应的实例。E-R 图中的实体的属性对应于数据库表的字段, 例如图 1-2 中, “课程”实体具有“课程名”“人数上限”等属性, 对应于课程数据库表的“课程名”字段及“人数上限”字段。在 E-R 图中, 通常来说属性是一个不可再分的最小单元, 如果属性能够再分, 建议将该属性进行细分, 或者将其“升格”为另一实体。例如, 在图 1-3 所示的“学生”实体的“联系方式”属性中, 如果细分为 E-mail、QQ、固定电话和手机等联系方式, 那么可以将“联系方式”属性进一步拆分为 E-mail、QQ、固定电话和手机 4 个属性。

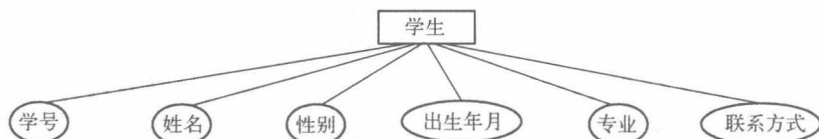


图 1-3 学生实体及属性