

责任编辑：付寒冰  
封面设计：黄燕美

# SQL Server 数据库开发与实践

SQL Server  
SHUJUKU KAIFA  
YU SHIJIAN

SQL Server 数据库开发与实践

主编 方柯 王艳荣

北京希望电子出版社

CX-8786

# SQL Server 数据库开发与实践

主编 方柯 王艳荣

SQL Server  
SHUJUKU KAIFA  
YU SHIJIAN



扫码关注

“北京希望电子出版社”微信公众号  
微信公众号回复8786，获取更多资源

微信公众号



扫码下载资料包

ISBN 978-7-83002-890-9



9 787830 028909 >

定价：58.00元

北京希望电子出版社网址：www.bhp.com.cn

电话：010-82626270

投稿：xiaohuijun@bhp.com.cn



北京希望电子出版社  
Beijing Hope Electronic Press  
www.bhp.com.cn



# SQL Server

## 数据库开发与实践

主 编 方 柯 王艳荣

副主编 姚 政

**SQL Server**  
**SHUJUKU KAIFA**  
**YU SHIJIAN**



北京希望电子出版社  
Beijing Hope Electronic Press  
[www.bhp.com.cn](http://www.bhp.com.cn)

## 内 容 简 介

本书以任务驱动的方式组织内容,突出实践性与可操作性,结合 SSMS 图形化工具,引导学生从零基础逐步掌握数据库开发的核心技术。全书共分为 8 个任务,主要内容有数据库基础、数据库设计、SQL Server 数据库管理系统、编程语言基础、SQL Server 数据库/表操作、SQL Server 数据操作、数据库编程对象以及数据库安全管理等。

本书适合作为“数据库基础与应用”课程的教材,也可供从事数据库管理与开发工作的技术人员学习参考。

---

### 图书在版编目(CIP)数据

SQL Server 数据库开发与实践 / 方柯, 王艳荣主编.

北京: 北京希望电子出版社, 2025. 12 (2026.2 重印).

ISBN 978-7-83002-890-9

I. TP311.132.3

中国国家版本馆 CIP 数据核字第 2025ZP5281 号

---

出版: 北京希望电子出版社

封面: 黄燕美

地址: 北京市海淀区中关村大街 22 号

编辑: 付寒冰

中科大厦 A 座 10 层

校对: 毕明燕

邮编: 100190

开本: 787 mm × 1 092 mm 1/16

网址: [www.bhp.com.cn](http://www.bhp.com.cn)

印张: 16.5

电话: 010-82620818 (总机) 转发行部

字数: 391 千字

010-82626237 (邮购)

印刷: 三河市骏杰印刷有限公司

经销: 各地新华书店

版次: 2026 年 2 月 1 版 2 次印刷

定价: 58.00 元

随着信息技术的迅猛发展，数据库技术已经成为现代信息社会不可或缺的一部分。无论是企业级应用还是个人项目，高效的数据管理和处理能力都是成功的关键因素之一。SQL Server 作为一款功能强大、易于使用的数据库管理系统，因其出色的性能、高可靠性和丰富的功能集，被广泛应用于各类数据管理场景中。

本书旨在帮助学生掌握 SQL Server 的基础知识和高级应用技巧，培养学生在数据库管理方面的实际操作能力和解决复杂问题的能力。本书注重理论与实践相结合，通过一系列精心设计的任务和案例，引导学生从基础概念逐步深入到复杂的数据库编程对象及安全管理等领域。

本书主要内容如下：

任 务	内 容 概 述
任务 1	介绍数据库系统的基本概念、数据模型及关系型数据库的核心理论，包括关系代数、关系运算及完整性约束，帮助学生建立数据库理论基础，为后续学习奠定基础
任务 2	讲述从需求分析到数据库运行维护的全流程，重点讲解 E-R 模型转换、关系模式规范化及物理设计要点，指导学生掌握科学规范的数据库设计方法
任务 3	概述 SQL Server 数据库的发展历程、体系结构及存储机制，详细介绍 SQL Server 2022 版本的安装配置与 SSMS 工具的使用，通过学习使学生能快速搭建开发环境
任务 4	介绍 T-SQL 语言的核心语法，包括数据类型、流程控制及错误处理机制等，以及与标准 SQL 语言的区别，为数据库编程提供必要的语言技能支撑
任务 5	讲解使用 T-SQL 管理数据库与数据表的实操方法，涵盖创建、修改、删除等关键操作，帮助学生掌握数据库的基础运维能力
任务 6	阐述数据增、删、改、查操作，重点解析单表查询、多表连接及嵌套查询技术，提升学生操作数据的效率与复杂查询的解决能力
任务 7	详细介绍索引、视图、存储过程等编程对象的特性与操作方法，结合事务与游标的应用，培养学生的高级数据库开发能力
任务 8	分析数据库安全风险与防护策略，介绍用户权限管理、数据备份恢复机制，使学生具备数据库系统安全保障能力



本书具有以下特色：

（1）任务驱动教学，强化实践能力培养。全书以“任务”为主线组织内容，采用“任务目标→任务描述→知识储备→任务实施→任务评价”的结构模式，引导学生通过完成具体任务，掌握数据库开发与管理的核心技能，提升解决实际问题的能力。

（2）内容系统全面，构建完整的知识体系。本书从数据库基础到高级应用，内容由浅入深、层层递进，涵盖数据库设计、SQL Server 系统架构、T-SQL 编程语言基础、数据库/表操作、数据操作、常用数据库编程对象及安全管理等关键知识点，形成完整的数据库知识链条，满足教学与实践的需求。

（3）工具结合语言，突出可操作性。本书紧密融合 SQL Server Management Studio (SSMS) 图形化工具与 T-SQL 语言，引导学生在图形界面与命令行双重环境下进行数据库操作，增强学习的直观性和实用性，提升学生的实际开发与调试能力。

（4）融入课程思政，落实立德树人根本任务。本书在知识传授过程中有机融入思政教育元素。例如，在讲解数据库安全时强调信息安全意识与职业道德；在任务实施中倡导严谨求实、精益求精的工匠精神；通过真实项目案例激发学生的社会责任感和科技报国情怀，实现专业教育与价值引领的统一，助力高素质技术技能人才的培养。

（5）紧跟技术前沿，体现与时俱进的特点。本书基于 SQL Server 2022 版本编写，反映当前主流数据库技术发展趋势，涵盖现代数据库管理系统的功能特性和应用场景，帮助学生掌握符合行业需求的新技术、新工具和新理念，提升就业竞争力。

本书适合作为“数据库基础与应用”课程的教材，也可供从事数据库管理与开发工作的技术人员学习参考。对于希望深入学习 SQL Server 并提升数据库应用能力的读者而言，本书同样是一本实用且难得的学习资料。

本书由方柯和王艳荣担任主编，由姚政担任副主编。由于编者水平有限，书中疏漏之处在所难免，恳请读者朋友批评指正。

编 者

2025 年 8 月



# 目录

CONTENTS

## 任务 1 数据库基础 ..... 1

任务描述 .....	2
知识储备 .....	2
1.1 数据库系统 .....	2
1.1.1 信息、数据和数据库 .....	2
1.1.2 数据库系统 .....	3
1.2 数据模型 .....	5
1.2.1 数据模型的概念 .....	5
1.2.2 数据模型的组成要素 .....	6
1.2.3 数据模型的应用层次 .....	6
1.3 关系型数据库 .....	11
1.3.1 关系模型 .....	11
1.3.2 关系型数据库的概念和特点 .....	16
1.3.3 关系型数据库管理系统 .....	17
任务实施 .....	20
任务评价 .....	22
课后习题 .....	23

## 任务 2 数据库设计 ..... 25

任务描述 .....	26
知识储备 .....	26
2.1 需求分析 .....	27
2.1.1 需求分析的内容 .....	27
2.1.2 需求分析的方法 .....	28
2.2 概念设计 .....	29
2.2.1 概念设计的方法 .....	29
2.2.2 概念设计的步骤 .....	30
2.3 逻辑设计 .....	34
2.3.1 E-R 模型到关系模型的转换 .....	35
2.3.2 关系模式的规范化 .....	36
2.3.3 关系模式的评价与改进 .....	39
2.4 物理设计 .....	39
2.4.1 物理设计的任务 .....	40



2.4.2 物理设计的评价	42
2.5 数据库实施	42
2.6 数据库运行与维护	44
2.6.1 数据库的运行	44
2.6.2 数据库的维护	44
任务实施	46
任务评价	50
课后习题	51

### 任务3 SQL Server 数据库管理系统 53

任务描述	54
知识储备	54
3.1 SQL Server 数据库概述	54
3.1.1 SQL Server 的发展历程	54
3.1.2 SQL Server 2022 版本介绍	56
3.1.3 SQL Server 的管理工具	56
3.1.4 SQL Server 2022 的运行环境要求	58
3.2 SQL Server 数据库的体系结构	59
3.2.1 SQL Server 2022 的组成	59
3.2.2 SQL Server 的数据库种类	60
3.2.3 SQL Server 数据库的存储结构	61
任务实施	63
任务评价	91
课后习题	92

### 任务4 T-SQL 编程语言基础 94

任务描述	95
知识储备	95
4.1 标准 SQL 语言与 T-SQL	95
4.1.1 标准 SQL 语言基础	95
4.1.2 T-SQL 与标准 SQL 的区别与联系	98
4.2 T-SQL 语言基础	99
4.2.1 数据类型	99
4.2.2 常量与变量	100
4.2.3 表达式与运算符	101
4.2.4 系统内置函数	102
4.2.5 流程控制语句	104
4.2.6 错误处理	108
4.2.7 延迟执行	108
任务实施	110
任务评价	118
课后习题	119

**任务 5 SQL Server 数据库 / 表操作 ..... 121**

任务描述 .....	122
知识储备 .....	122
5.1 使用 T-SQL 管理数据库 .....	122
5.1.1 创建数据库 .....	122
5.1.2 查看数据库 .....	123
5.1.3 选择数据库 .....	124
5.1.4 修改数据库 .....	125
5.1.5 删除数据库 .....	126
5.2 使用 T-SQL 管理数据表 .....	126
5.2.1 创建数据表 .....	126
5.2.2 修改数据表 .....	128
5.2.3 删除数据表 .....	133
任务实施 .....	135
任务评价 .....	152
课后习题 .....	153

**任务 6 SQL Server 数据操作 ..... 155**

任务描述 .....	156
知识储备 .....	156
6.1 使用 T-SQL 管理数据 .....	156
6.1.1 插入数据 .....	156
6.1.2 更新数据 .....	158
6.1.3 删除数据 .....	159
6.2 使用 T-SQL 查询数据 .....	161
6.2.1 基本查询语句 .....	161
6.2.2 无数据源查询 .....	162
6.2.3 单表查询 .....	163
6.2.4 多表查询 .....	172
6.2.5 嵌套查询 .....	175
任务实施 .....	177
任务评价 .....	185
课后习题 .....	186

**任务 7 数据库编程对象 ..... 189**

任务描述 .....	190
知识储备 .....	190
7.1 索引 .....	190
7.1.1 索引概述 .....	190
7.1.2 索引操作 .....	193
7.2 视图 .....	194



7.2.1 视图概述 .....	195
7.2.2 视图操作 .....	195
7.3 存储过程 .....	198
7.3.1 存储过程概述 .....	198
7.3.2 存储过程操作 .....	199
7.4 触发器 .....	202
7.4.1 触发器概述 .....	202
7.4.2 触发器操作 .....	203
7.5 游标 .....	206
7.5.1 游标概述 .....	206
7.5.2 游标操作 .....	206
7.6 批处理与脚本 .....	208
7.6.1 批处理 .....	208
7.6.2 脚本 .....	212
7.7 事务 .....	212
7.7.1 事务概述 .....	212
7.7.2 事务操作 .....	214
任务实施 .....	216
任务评价 .....	229
课后习题 .....	230

## 任务 8 数据库安全管理 ..... 233

任务描述 .....	234
知识储备 .....	234
8.1 数据库安全 .....	234
8.1.1 数据库安全概述 .....	234
8.1.2 数据库安全技术、策略与机制 .....	236
8.2 用户与权限管理 .....	237
8.2.1 用户管理 .....	237
8.2.2 权限管理 .....	240
8.3 数据备份与恢复 .....	241
8.3.1 数据备份 .....	241
8.3.2 数据恢复 .....	243
任务实施 .....	246
任务评价 .....	250
课后习题 .....	251

## 课后习题答案 ..... 254

## 参考文献 ..... 256



## 任务 1

# 数据库基础

### 任务目标

#### 知识目标

- 掌握数据库系统的基本概念、组成要素与体系结构,理解数据库与传统文件管理方式的区别。
- 理解数据模型的 3 个层次及其在数据库设计中的作用。
- 熟悉关系型数据库的核心理论,包括关系模型、关系运算、完整性约束以及主键、外键等关键概念。

#### 技能目标

- 能够根据实际应用场景分析数据库系统的构成要素,并能描述其运行机制。
- 具备将现实世界信息抽象为概念模型的能力,初步掌握 E-R 图的绘制方法。
- 能通过不同关系型数据库管理系统的对比,理解其各自的适用场景与技术差异。

#### 素养目标

- 树立信息安全意识,在学习数据库系统原理的同时强化职业道德观念,特别是在涉及数据安全和隐私保护时保持高度责任感。
- 培养严谨求实的学习态度,注重理论联系实际,追求建模与设计的科学性与规范性,体现精益求精的工匠精神。
- 提升团队协作与沟通能力,在任务实施过程中注重逻辑表达与问题分析,助力高素质技术技能人才的成长路径。



## 任务描述

某企业计划开发一个名为 EB-Shop 的线上书城系统。公司技术团队已经确定了基础技术路线，即服务器操作系统为 Windows Server，核心开发框架为 .NET Core。项目预期上线后，日均独立访客数 (UV) 可达 5 000+，日订单量 200+。

作为开发团队的一员，你需要在上述技术背景和业务规模下，为 EB-Shop 线上书城选用一款合适的数据库管理系统，以满足以下业务与功能需求：

(1) 能够与企业现有的 Windows Server 和 .NET Core 开发框架高效协同工作，以简化开发流程、降低集成复杂度。

(2) 能够确保交易过程的可靠性，特别是在高并发场景下（如用户下单、支付、库存扣减等关键操作）保障数据准确与一致。

(3) 能够高效完成业务中的各类数据查询与分析，例如商品搜索、订单查询、销售统计等，并在预期访问压力下保持稳定、快速的响应。

(4) 能够提供便于数据库管理员进行日常运维的工具，降低系统在监控、优化和故障排查等方面的管理难度。

(5) 在满足所有功能与性能需求的前提下，总体成本应合理可控，具备良好的成本效益。

## 知识储备

### 1.1 数据库系统

数据库技术是一种综合性的软件技术，具有存储、管理大量数据和高效检索的优势，被广泛应用于人们日常生活中的各个领域。随着计算机应用的不断发展，数据处理变得越来越重要，数据库技术的应用范围也越来越广泛。

#### 1.1.1 信息、数据和数据库

自计算机问世以来，人类社会便进入了信息时代。

信息 (information) 是现实世界事物的存在方式或运动状态在人们头脑中的反映，是对客观世界的认识。它具有可感知、可存储、可加工、可传递和可再生等自然特性。

数据 (data) 是用来记录信息的符号，是信息的具体表现形式。数据可以用型和值来表示，其中，型是指数据内容存储在媒体上的具体形式，值是指所描述的客观事物的具体特性。同一信息可以用不同的数据形式表示，但数据本身不会因为数据形式的不同而改变。例如，一个人的身高可以用两种不同的方式来表示，如“1.60 米”或“1 米 6”，

这两种方式的数据的值是相同的，但是数据的型不同。除了数字、文字形式，数据还包括图形、图像、声音、动画等多媒体数据形式。

数据库（database, DB）是长期存放在计算机内、有组织的、可共享的数据集合。数据库中的数据是按照一定的数据模型进行组织、描述和储存的，具有较小的冗余度、较高的数据独立性和易扩展性，并可为各种用户所共享，可以形象地将数据库理解为存储数据的仓库。当人们在社交媒体上分享动态信息时，体现动态信息的数据实际上被储存在数据库中；当人们在购物网站上搜索商品时，网站界面展现出来的商品列表其实就是人们在数据库中查询的结果。数据库就像一座图书馆，将各种数据保存起来，便于人们根据需求进行查询和检索。

### 1.1.2 数据库系统

数据库系统是现代信息管理的核心工具，它允许用户高效地存储、管理和检索数据。数据库系统能够确保数据的完整性、安全性和可用性，同时支持多用户并发访问，具有处理大量数据的能力。

为了更好地理解数据库系统的重要性及其运作方式，需要了解数据库系统的基本概念、组成要素及其系统架构。

#### 1. 数据库系统的概念

一个完整的数据库系统结构如图 1-1 所示，它主要由数据库、支持数据库运行的硬件及操作系统、数据库管理系统、应用系统、数据库管理员和用户构成。

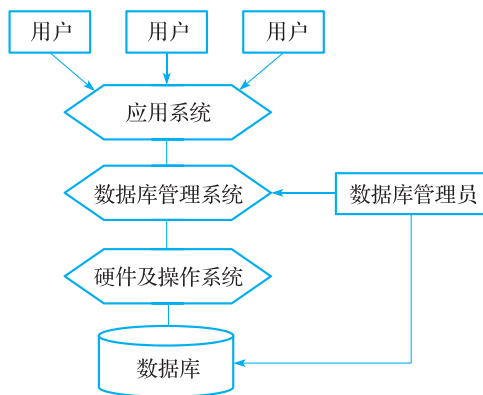


图 1-1 数据库系统结构

#### 2. 数据库系统的组成要素

数据库系统的组成要素包括：硬件、软件、人员和数据。

##### （1）硬件。

硬件是指系统所有的物理设备。鉴于数据库应用系统的需求，通常会要求数据库主机或数据库服务器的外存要足够大，I/O 存取效率要高；要求主机的吞吐量大、作业处理能力。对于分布式数据库而言，计算机网络也是基础硬件环境。具体要求如下：



① 要有足够大的内存，能存放操作系统和数据库管理系统的核心模块、数据库缓存区和应用程序。

② 要有足够大的硬盘等直接存取设备存放数据库，有足够的存储空间作为数据备份介质。

③ 要求连接系统的网络有较高的数据传输速度。

④ 要有性能强劲的中央处理器（CPU）以保证数据处理效率。

## （2）软件。

软件是指计算机系统包含的各种程序及其文档。数据库系统的软件主要包括操作系统（如 Linux 或 Windows 等）、数据库管理系统（如 Oracle、MySQL、DB2、SQL Server 或 Access 等）、数据库系统开发工具（如 Oracle SQL Developer、IBM Data Studio、Microsoft SQL Server Management Studio 等）。

其中，数据库管理系统是介于用户与操作系统之间的数据管理软件，它为用户或应用程序提供了访问数据的方法，包括建立数据库、操纵数据、检索数据和进行数据控制等。

## （3）人员。

人员指数据库系统的所有用户，一般把数据库系统中的用户分为 4 类：数据库管理员、系统分析员和数据库设计人员、应用程序开发人员，以及终端用户。

① 数据库管理员。数据库管理员的主要职责是负责数据库的规划、设计、维护和监控，需要对各个应用的数据需求进行全面规划、设计和集成，负责对数据库中数据的安全性、完整性以及系统恢复进行实施与维护，并且不断调整数据库的内部结构，以保持系统的最佳状态和最高效率。

② 系统分析员和数据库设计人员。系统分析员的主要任务是编写应用系统的需求分析、确定数据库系统的软硬件配置，并参与数据库的设计和程序开发工作。数据库设计人员主要负责设计数据库的结构，实际上他们是数据库的建筑师。

③ 应用程序开发人员。应用程序开发人员是负责设计、开发应用系统功能模块的软件编程人员，他们根据数据库结构编写特定的应用程序，并进行调试和安装。

④ 终端用户。终端用户是使用应用程序的人员，包括日常业务操作人员 and 高级用户（如主管、经理、董事等）。高级用户可以利用从数据库获取的信息做战略和战术上的决策。

## （4）数据。

数据是指存储在数据库中的事实集合，它是一个数据库系统的“质量”基础。数据库系统中的数据种类包括永久性数据（persistent data）、索引数据（indexes）、数据字典（data dictionary）和事务日志（transaction log）等。

## 3. 数据库系统架构

根据数据存储和处理方式的不同，数据库系统架构可以分为集中式数据库系统和分布式数据库系统。

### （1）集中式数据库系统。

集中式数据库系统是指数据库系统运行在一台计算机上，不与其他计算机系统交互，

如个人计算机上的单用户数据库系统或高性能单机数据库系统等。这种系统的优点是管理和维护比较简单，数据的安全性和一致性较高。缺点在于数据量过大时，计算机的性能可能会受到影响，而且存在单点故障风险，一旦主机发生故障，整个系统将无法正常运行。

在应用访问层面，集中式数据库系统的常见实现模式有客户端－服务器（C/S）模式与浏览器－服务器（B/S）模式。

① C/S 模式。C/S 模式将数据库的功能分为前台客户端系统和后台服务器系统。其中，客户端系统主要负责图形用户界面工具、表格和报表生成等；服务器系统负责数据存取和控制，包括故障恢复和并发控制等。客户端和服务器之间通过网络通信，客户端可以通过客户端软件与服务器进行交互。这种模式可以将数据处理和存储分离，提高了系统的可扩展性和可靠性，但是需要进行网络通信，可能会受到网络延迟和安全风险的影响。

② B/S 模式。B/S 模式是 C/S 模式的一种扩展形式，将客户端中的应用层从客户端中分离出来，应用程序的用户界面以 Web 界面方式实现，将系统功能实现的核心部分集中到服务器上，所有的业务逻辑和数据处理都在服务器上进行，简化了系统的开发和维护，用户只需要安装一个浏览器即可访问数据库。Web 服务器充当了客户端与数据库服务器的中介，架起了用户界面与数据库之间的桥梁。这种模式使得用户可以通过任何设备和浏览器访问数据库，无须安装客户端软件，而且系统的维护和管理也比较简单。但是，由于所有的数据处理和存储都在服务器上进行，可能会对服务器的性能产生影响，而且系统的安全性也需要考虑。

## （2）分布式数据库系统。

分布式数据库系统是将数据分散存储在多个节点上，每个节点都可以处理一部分数据和业务逻辑，通过网络同时存取和处理多个异地数据库中的数据。它由多台计算机组成，每台计算机都配有各自的本地数据库，大部分处理任务由本地计算机完成，是计算机网络技术发展的必然产物。这种系统的优点是可以实现数据的高可用性和可扩展性，提高了系统的性能和可靠性。缺点是系统的设计和实现比较复杂，需要考虑数据一致性和分布式事务等问题。

## 1.2 数据模型

模型是对现实世界的抽象。在数据库技术中，以数据模型的概念描述数据库的结构和语义，并对现实世界的数据进行抽象。从现实世界的信息到数据库存储的数据，再到用户使用的数据，这是一个逐步抽象的过程。

### 1.2.1 数据模型的概念

数据库中的数据是结构化的，建立数据库时需要考虑如何组织数据，如何表示数据之间的联系，并将数据合理地存放在计算机中，这样才能便于对数据进行有效的处理。

数据模型是对现实世界的一种抽象和描述，它可以用来模拟现实世界中的事物、概念和关系，并在计算机中进行存储、管理和查询。

良好的数据模型应该具备以下 3 个特点。

- 能够比较真实地描述现实世界，即数据模型应该能够反映现实世界的本质特征和规律。
- 易于被用户理解，即数据模型应该具有清晰、简洁、易于理解的表达方式，能够帮助用户快速理解数据结构和关系。
- 易于在计算机上实现，即数据模型应该能够被计算机程序理解和处理，能够方便地进行存储、管理和查询。

### 1.2.2 数据模型的组成要素

数据模型由数据结构、数据操作和数据完整性约束 3 个要素组成。

#### 1. 数据结构

数据结构用于对数据系统的静态特征进行描述，研究的是数据本身的类型、内容和性质，以及数据之间的关系。

#### 2. 数据操作

数据操作用于对数据系统的动态特征进行描述，是对数据库中对象实例允许执行的操作的集合，包括对对象实例的检索、插入、删除和修改等操作。数据模型必须定义这些操作的确切含义、操作符号、操作规则（如优先级）和实现操作的语言。

#### 3. 数据完整性约束

数据完整性约束是一组完整性规则的集合，用以规定数据库状态及状态变化所应满足的条件，以保证数据的正确性、有效性和一致性。

### 1.2.3 数据模型的应用层次

在实际应用中，数据模型通常会被分为 3 个层次：概念数据模型、逻辑数据模型和物理数据模型。通常需要根据具体的需求和应用场景选择合适的数据模型层次，以达到数据存储和管理的最佳效果。

#### 1. 概念数据模型

（1）概念数据模型的定义。

概念数据模型（conceptual data model）是对现实世界中的数据结构和关系进行抽象和描述的一种高层次的数据模型。它通常是基于业务需求和用户需求进行设计和建立的，用于描述业务实体和实体间的关系。概念数据模型通常是一个高层次的概念模型，它并不涉及具体的实现细节，只描述了数据的结构和关系。

（2）E-R 模型。

概念模型的表示方法很多，其中最常用的是实体－联系模型（entity-relationship model），

简称为 E-R 模型。在 E-R 模型中，信息用实体、实体属性和实体间的联系 3 种概念单元来表示。

① 实体：指建立概念模型的对象。在 E-R 模型图中用矩形框表示，在框内写上实体名。例如，学生、课程就是两个不同的实体。

② 实体属性：指实体的说明。在 E-R 模型图中用椭圆框表示，并用无向边把实体与其属性连接起来。例如，学生实体有学号、姓名、性别、出生日期、手机号等属性。

③ 实体间的联系：指两个或两个以上实体类型之间的关联关系，需指定名称。在 E-R 模型图中，实体间的联系用菱形框表示，菱形框内要有联系名，并用无向边把菱形框分别与有关实体相连接，在无向边的旁边标上联系的类型。例如，用 E-R 图表示某学校学生选课情况的概念模型，如图 1-2 所示。一个学生可以选修多门课程，一门课程也可以被多个学生选修，因此，学生和课程之间具有多对多的联系。

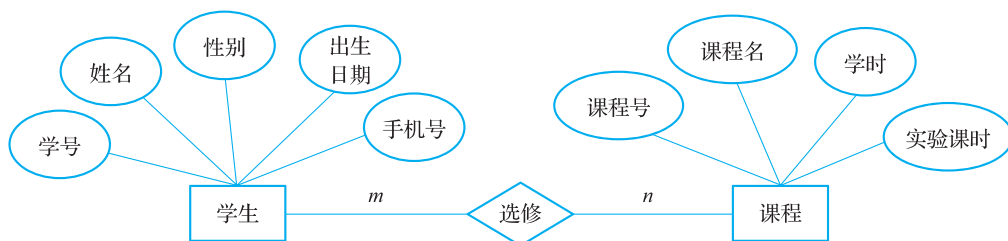


图 1-2 实体、实体属性及实体联系模型

## 2. 逻辑数据模型

### （1）逻辑数据模型的定义。

逻辑数据模型（logical data model）是对概念数据模型进一步抽象和细化的一种数据模型。它主要描述了数据在计算机内部的存储和组织方式，包括实体、属性、关系等的具体实现方式。逻辑数据模型通常是一个中间层次的数据模型，它将概念数据模型转化为计算机可理解的数据结构和关系。

### （2）逻辑数据模型的分类。

逻辑数据模型包括层次模型、网状模型、关系模型、面向对象模型和对象关系模型等。

① 层次模型。层次模型（hierarchical model）是一种树状结构的数据模型，它由多个实体和实体之间的关系组成，每个实体可以有多个属性。在层次模型中，实体和属性被组织成一个树状结构，其中，根节点代表整个数据模型，每个节点代表一个实体，每个实体又可以包含多个子节点和属性。层次模型通常使用层次数据库来实现，它的优点是对于处理具有明确定义的层级关系的数据非常有用（如组织结构、文件系统等），缺点是缺乏灵活性和适应性，数据库中的数据冗余较高，且难以处理复杂查询。

② 网状模型（network model）是一种用网状结构表示实体及其关系的数据模型。它由记录类型（实体）和系（关系）组成，允许一个记录类型参与多个系，从而直接表示复杂的多对多关系。网状模型通常由专门的网状数据库管理系统（如 IDMS）实现，是数据库技术发展早期的主流模型之一。它的主要优点是查询性能高，能高效处理复杂的关系；



主要缺点是结构僵化，数据独立性差，难以维护和修改，这也是它后来被更灵活的关系模型所取代的主要原因。

③ 关系模型。关系模型（relational model）是一种二维表格结构的数据模型，它由多个实体和它们之间的关系组成，每个实体对应一张表格，而关系通过外键在表格之间建立关联。在关系模型中，每个表格包含若干列，每列代表一个属性。关系模型通常使用关系型数据库实现，其优点是结构清晰、易于维护和修改，缺点是难以处理大规模数据和复杂的关系。

④ 面向对象模型。面向对象模型（object-oriented model）是一种基于对象的数据模型，它将现实世界中的事物和概念抽象成对象，每个对象都具有属性和方法。在面向对象模型中，数据被组织成一个对象图，每个对象代表一个实体或概念，它们之间的关系通过对象之间的方法和属性来表示。面向对象模型通常使用面向对象数据库或 NoSQL 数据库实现，其优点是可以处理大规模数据和复杂的关系，并具有良好的可扩展性和灵活性，缺点是需要对面向对象编程有一定程度的理解和掌握。

⑤ 对象关系模型。对象关系模型（object relational model）是一种将关系型数据库中的数据映射到面向对象编程语言中的对象上的技术。对象关系模型框架通常提供了一种将数据库表 and 对象之间进行映射的方式，使得开发人员可以使用面向对象的方式访问和操作数据库。

### 3. 物理数据模型

物理数据模型是对逻辑数据模型进行最终实现的一种数据模型。它描述了数据在实际存储和组织中的具体实现方式，包括数据在磁盘等存储介质中的存储方式、索引方式、备份和恢复方式等。物理数据模型通常是一个低层次的数据模型，它将逻辑数据模型转化为具体的存储结构和实现方式。

## 知识延伸

### 三级模式和二级映像

美国国家标准学会（American National Standards Institute, ANSI）所属的标准计划与需求委员会（Standards Planning and Requirements Committee, SPARC）在 1971 年公布的研究报告中提出了 ANSI-SPARC 体系结构，即三级模式结构（或称三层体系结构）。ANSI-SPARC 最终没有成为正式标准，但它仍然是理解数据库管理系统的基础。

三级模式是指数据库管理系统从 3 个层次来管理数据，分别是外部层（external level）、概念层（conceptual level）和内部层（internal level）。这 3 个层次分别对应 3 种不同类型的模式，分别是外模式（external schema）、概念模式（conceptual schema）和内模式（internal schema）。在外模式与概念模式之间，以及概念模式与内模式之间，还存在外模式 / 概念模式映像和概念模式 / 内模式映像，即二级映像，具体如图 1-3 所示。

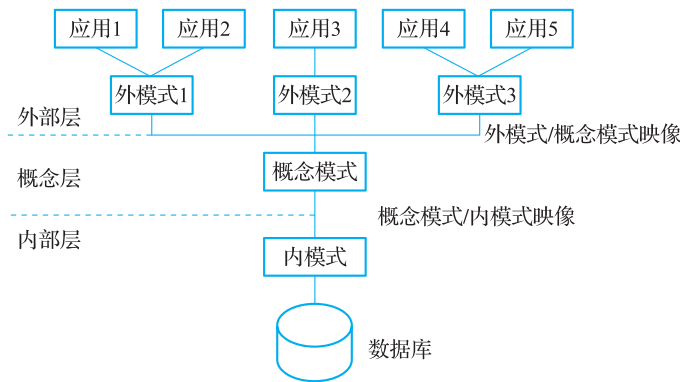


图 1-3 三级模式和二级映像

在图 1-3 中，外模式面向应用程序，描述用户的数据视图（view）；内模式（又称物理模式、存储模式）面向物理上的数据库，描述数据在磁盘中如何存储；概念模式（又称模式、逻辑模式）面向数据库设计人员，描述数据的整体逻辑结构。由于三级模式比较抽象，为了更好地理解，下面将计算机中常用的 Excel 电子表格类比成数据库，并假设有一个商城使用电子表格来保存商品信息。

（1）概念模式。概念模式类似于表格的列标题，它描述了商品表中包含哪些信息，如表 1-1 所示。

表 1-1 商品信息表格

编号	商品名称	商品分类	商品价格	库存	销量

在表 1-1 中，表的横向称为行，纵向称为列，第 1 行就是列标题，用来描述该列的数据表示什么含义。实际上，概念模式在数据库中描述的信息还有很多，如多张表之间的联系、表中每一列的数据类型和长度等，读者在后面的学习中就会接触到这些内容。

（2）内模式。在将 Excel 表格另存为文件时，可以选择保存的文件路径、文件类型（如 XLS、XLSX、CSV 等格式）等，这些与存储相关的描述信息相当于内模式。在数据库中，内模式描述数据的物理结构和存储方式，如堆文件、索引文件、散列（hash）文件等。

（3）外模式。在打开一个电子表格后，默认会显示表格中所有的数据，这个表格即基本表（base table）。在将数据提供给其他用户时，出于权限、安全控制等因素的考虑，只允许用户看到一部分数据，或不同用户看到不同的数据，这样的需求就可以用视图（view）来实现。视图是从一个或几个基本表导出的表，是一个虚拟表（virtual table）。它本身不独立存在于数据库中，数据库中只存放视图的定义而不存放视图对应的数据，这些数据仍存放在导出视图的基本表中。当基本表中的数据发生变化时，从视图中查询出来的数据也随之改变。

图 1-4 演示了视图和基本表之间的关系。

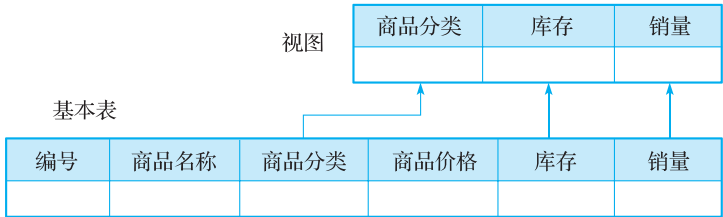


图 1-4 视图和基本表之间的关系

在图 1-4 中，基本表中的数据是实际存储在数据库中的，而视图中的数据是查询或计算出来的。由此可见，外模式可以为不同用户的需求创建不同的视图，且由于不同用户的需求不同，数据的显示方式也会多种多样。因此，一个数据库中会有多个外模式，而概念模式和内模式则只有一个。

通过前面的分析可知，三级模式是数据的 3 个抽象级别，每个级别关心的重点不同。为了使三级模式之间产生关联，数据库管理系统在三级模式之间提供了二级映像功能。二级映像是一种规则，它规定了映像双方如何进行转换。通过二级映像，体现了逻辑和物理两个层面的数据独立性。具体解释如下：

（1）逻辑独立性。外模式 / 概念模式映像体现了逻辑独立性。逻辑独立性是指当修改了概念模式，不影响其上一层的外模式。例如，将图 1-4 中基本表的“库存”和“销量”拆分到另一张表中，此时概念模式发生了变化，但可以通过改变外模式 / 概念模式映像，继续为用户提供原有的视图，如图 1-5 所示。

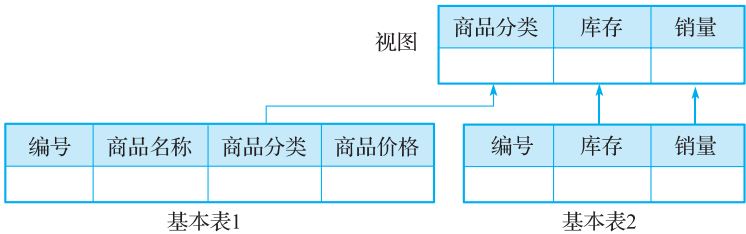


图 1-5 拆分基本表后的视图与基本表之间的关系

由此可见，逻辑独立性能够让使用视图的用户感觉不到基本表的改变。其实，逻辑独立性带来的好处还有很多，随着后面的学习，读者会有更深入的体会。

（2）物理独立性。概念模式 / 内模式映像体现了物理独立性。物理独立性是指修改了内模式，不影响其上层的概念模式和外模式。例如，在 Excel 中将 XLS 文件另存为 XLSX 文件，虽然更换了文件格式，但是打开文件后显示的表格内容并不会发生改变。

在数据库中，更换更先进的存储结构，或者创建索引以加快查询速度，会使内模式发生改变。此时，只需改变概念模式 / 内模式映像，就不会影响到原有的概念模式。

另外，物理独立性使得用户不必了解数据库内部的存储原理即可正常使用数据库来保存数据。数据库管理系统会自动将用户的操作转换成物理级数据库的操作。

## 1.3 关系型数据库

关系型数据库是现代数据管理中最常用的数据库类型之一，它基于关系模型来组织数据。理解关系型数据库的基本理论对于有效地设计、开发和管理数据库系统至关重要，因此需要介绍关系模型的几个核心概念。

### 1.3.1 关系模型

关系模型是一种基于关系代数的数据模型，它使用表格组织和存储数据，并使用关系描述表格之间的联系。关系代数是关系数据库查询语言的基础，它提供了用于操作和检索关系中数据的一系列操作。

#### 1. 关系代数

关系代数是一种基于关系运算的代数系统，可用于描述和操作关系型数据库中的数据。关系代数提供一组操作，它们接受一个或两个关系作为输入，并生成一个新的关系作为输出。关系代数提供的用于关系型数据库的操作主要由基本运算、函数、关系模式和查询等组成。

##### （1）基本运算。

基本运算包括选择（selection）、投影（projection）、连接（join）、笛卡儿积（cartesian product）、并（union）、交（intersection）、差（difference）等。这些运算还可以组合、变换，用于操作关系模式。

##### （2）函数。

函数是用来计算或操作数据的一种特殊映射，如求和、求平均值、求最大值等。函数可以用来对数据进行聚合计算，以便从多个关系模式中获取有用的信息。

##### （3）关系模式。

关系模式是用来描述关系型数据库中实体和实体之间联系的数据结构。一个关系模式通常由一个关系名和若干个属性组成。关系模式可以用来表示实体和实体之间的联系，也可以用来存储和组织数据。

##### （4）查询。

查询是从关系模式中获取数据的一种操作。查询通常由一个或多个选择语句组成，每个选择语句指定要查询的表和要返回的列。查询可以用来获取满足特定条件的数据，也可以用来对数据进行聚合计算和分析。在关系代数中，查询被视为一个表达式，它描述了对关系型数据库中的数据进行操作的方式。

#### 2. 关系的类型

实体间的联系是错综复杂的。在常见的数据库环境中，实体之间有 3 种类型的关系：一对一、一对多和多对多，如图 1-6 所示。



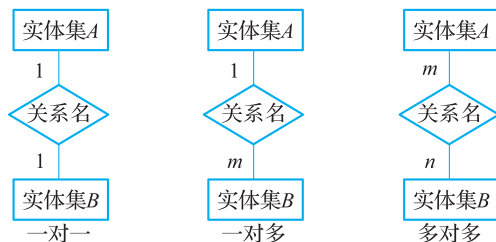


图 1-6 关系的类型

### （1）一对一关系。

对于实体集  $A$  中的每一个实体，实体集  $B$  中至多有一个实体与之联系，反之亦然，则称实体集  $A$  与实体集  $B$  具有一对一关系，记为  $1:1$ 。例如，通常一个班级内只有一个班长，一个班长也只能是某一个班级的班长，所以班级和班长之间具有一对一关系。

### （2）一对多关系。

对于实体集  $A$  中的每一个实体，实体集  $B$  中有  $m$  个实体 ( $m>2$ ) 与之联系；反过来，对于实体集  $B$  中的每一个实体，实体集  $A$  中至多有一个实体与之联系，则称实体集  $A$  与实体集  $B$  具有一对多关系，记为  $1:m$ 。例如，一个班级内有多名学生，一名学生只能属于一个班级，即班级与学生之间具有一对多关系。

### （3）多对多关系。

对于实体集  $A$  中的每一个实体，实体集  $B$  中有  $n$  个实体 ( $n>0$ ) 与之联系；反过来，对于实体集  $B$  中的每一个实体，实体集  $A$  中也有  $m$  个实体 ( $m>0$ ) 与之联系，则称实体集  $A$  与实体集  $B$  具有多对多关系，记为  $m:n$ 。例如，在选课时，一名学生可以选修多门课程，一门课程也可以被多名学生选修，学生和课程之间具有多对多关系。

## 3. 关系运算

关系运算用于从一个或多个关系中按特定条件选择、组合或投影出所需的数据，并形成新的关系。

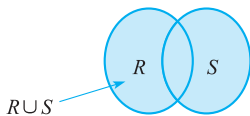
### （1）集合运算。

传统的集合运算包括 4 种运算：并 ( $\cup$ )、交 ( $\cap$ )、差 ( $-$ ) 和广义笛卡儿积 ( $\times$ )。注意：进行并、交、差集合运算的两个关系必须具有相同的模式，即结构相同。

① 并。如果  $R$  和  $S$  都是关系，那么由所有属于  $R$  或属于  $S$  的元组（记录）组成的新关系称为  $R$  和  $S$  的并，记作  $R \cup S$ ，即：

$$R \cup S = \{t \mid t \in R \vee t \in S\}$$

$R \cup S$  的属性与关系  $R$  或关系  $S$  相同，其值由所有属于  $R$  或  $S$  的元组组成。图 1-7 中的深色部分表示了  $R \cup S$  的运算结果。

图 1-7  $R \cup S$  的运算结果

② 交。如果  $R$  和  $S$  都是关系，那么由同属于  $R$  和  $S$  的元组组成的新关系称为  $R$  和  $S$  的交，记为  $R \cap S$ ，即：

$$R \cap S = \{t \mid t \in R \wedge t \in S\}$$

$R \cap S$  的属性与关系  $R$  或关系  $S$  相同，其值由既属于  $R$  又属于  $S$  的元组组成。图 1-8 中的深色部分表示了  $R \cap S$  的运算结果。

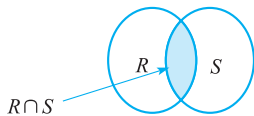


图 1-8  $R \cap S$  的运算结果

③ 差。如果  $R$  和  $S$  都是关系，那么由所有属于  $R$  但不属于  $S$  的元组组成的新关系称为  $R$  和  $S$  的差，记为  $R - S$ ，即：

$$R - S = \{t \mid t \in R \wedge t \notin S\}$$

$R - S$  的属性与关系  $R$  或关系  $S$  相同，其值由属于  $R$  但不属于  $S$  的所有元组组成。图 1-9 中的深色部分表示了  $R - S$  的运算结果。

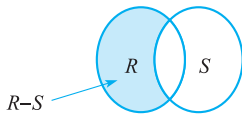


图 1-9  $R - S$  的运算结果

④ 广义笛卡儿积。两个分别有  $n$  个属性和  $m$  个属性的关系  $R$  和  $S$  的广义笛卡儿积是一个有  $n+m$  个属性的元组的集合，其中，该元组的前  $n$  个属性是关系  $R$  的属性，后  $m$  个属性是关系  $S$  的属性。若  $R$  有  $k_1$  个元组， $S$  有  $k_2$  个元组，则关系  $R$  和  $S$  的广义笛卡儿积有  $k_1 \times k_2$  个元组，记作  $R \times S$ ，即：

$$R \times S = \{(t_R, t_S) \mid t_R \in R \wedge t_S \in S\}$$

例如，若  $R$  和  $S$  两个关系如图 1-10 和图 1-11 所示，则  $R \times S$  关系如图 1-12 所示。

$A$	$B$	$C$
$a_1$	$b_2$	$c_2$
$a_1$	$b_2$	$c_3$
$a_2$	$b_1$	$c_1$

图 1-10 关系  $R$

$A$	$B$	$C$
$a_1$	$b_2$	$c_2$
$a_2$	$b_4$	$c_1$
$a_3$	$b_3$	$c_2$
$a_4$	$b_1$	$c_1$

图 1-11 关系  $S$

(2) 选择运算。

选择是指从关系  $R$  中选择满足指定条件的元组，它是一种对表的横向操作。选择运算所形成的新关系的关系模式不变，即不影响原关系的结构，但其元组的数目小于或等

于原关系中的元组数目，因此新关系是原有关系的一个子集。

<i>R.A</i>	<i>R.B</i>	<i>R.C</i>	<i>S.A</i>	<i>S.B</i>	<i>S.C</i>
<i>a</i> <sub>1</sub>	<i>b</i> <sub>2</sub>	<i>c</i> <sub>2</sub>	<i>a</i> <sub>1</sub>	<i>b</i> <sub>2</sub>	<i>c</i> <sub>2</sub>
<i>a</i> <sub>1</sub>	<i>b</i> <sub>2</sub>	<i>c</i> <sub>2</sub>	<i>a</i> <sub>2</sub>	<i>b</i> <sub>4</sub>	<i>c</i> <sub>1</sub>
<i>a</i> <sub>1</sub>	<i>b</i> <sub>2</sub>	<i>c</i> <sub>2</sub>	<i>a</i> <sub>3</sub>	<i>b</i> <sub>3</sub>	<i>c</i> <sub>2</sub>
<i>a</i> <sub>1</sub>	<i>b</i> <sub>2</sub>	<i>c</i> <sub>2</sub>	<i>a</i> <sub>4</sub>	<i>b</i> <sub>1</sub>	<i>c</i> <sub>1</sub>
<i>a</i> <sub>1</sub>	<i>b</i> <sub>2</sub>	<i>c</i> <sub>3</sub>	<i>a</i> <sub>1</sub>	<i>b</i> <sub>2</sub>	<i>c</i> <sub>2</sub>
<i>a</i> <sub>1</sub>	<i>b</i> <sub>2</sub>	<i>c</i> <sub>3</sub>	<i>a</i> <sub>2</sub>	<i>b</i> <sub>4</sub>	<i>c</i> <sub>1</sub>
<i>a</i> <sub>1</sub>	<i>b</i> <sub>2</sub>	<i>c</i> <sub>3</sub>	<i>a</i> <sub>3</sub>	<i>b</i> <sub>3</sub>	<i>c</i> <sub>2</sub>
<i>a</i> <sub>1</sub>	<i>b</i> <sub>2</sub>	<i>c</i> <sub>3</sub>	<i>a</i> <sub>4</sub>	<i>b</i> <sub>1</sub>	<i>c</i> <sub>1</sub>
<i>a</i> <sub>2</sub>	<i>b</i> <sub>1</sub>	<i>c</i> <sub>1</sub>	<i>a</i> <sub>1</sub>	<i>b</i> <sub>2</sub>	<i>c</i> <sub>2</sub>
<i>a</i> <sub>2</sub>	<i>b</i> <sub>1</sub>	<i>c</i> <sub>1</sub>	<i>a</i> <sub>2</sub>	<i>b</i> <sub>4</sub>	<i>c</i> <sub>1</sub>
<i>a</i> <sub>2</sub>	<i>b</i> <sub>1</sub>	<i>c</i> <sub>1</sub>	<i>a</i> <sub>3</sub>	<i>b</i> <sub>3</sub>	<i>c</i> <sub>2</sub>
<i>a</i> <sub>2</sub>	<i>b</i> <sub>1</sub>	<i>c</i> <sub>1</sub>	<i>a</i> <sub>4</sub>	<i>b</i> <sub>1</sub>	<i>c</i> <sub>1</sub>

图 1-12  $R \times S$  关系

选择又被称为限制，其运算操作符为  $\sigma$ ，并用  $c$  代表逻辑表达式，表示选择条件。选择运算由逻辑运算符连接的关系表达式或其他逻辑表达式〔取值为“真（true）”或“假（false）”〕组成，通常把选择运算记作：

$$\sigma_c(R) = \{t \mid t \in R \wedge c(t) = \text{'真'}\}$$

逻辑表达式中的比较运算符有  $>$ （大于）、 $\geq$ （大于或等于）、 $=$ （等于）、 $<$ （小于）、 $\leq$ （小于或等于）、 $\neq$ （不等于），逻辑运算符有  $\wedge$ （与）、 $\vee$ （或）和  $\neg$ （非）。

### （3）投影运算。

投影是从关系  $R$  中选择出若干属性（字段）组成新关系的一种运算，是一种对表的竖向的操作。投影运算是一种针对表内容的列运算。由于在投影后生成的新表中，可能会出现因属性被取消了若干个而导致剩下的元组数据完全相同的情况，因此，投影运算后必须将这些重复元组取消。

关系  $R$  上的投影是从  $R$  中选择出若干属性列组成新的关系，记作：

$$\Pi_A(R) = \{t[A] \mid t \in R\}$$

其中， $A$  为  $R$  中的属性列。

### （4）连接运算。

连接运算可以实现两个关系的横向合并，通过公共属性名连接成一个新的关系，在新的关系中可以反映出原来关系之间的关联。

①  $\theta$  连接。 $\theta$  连接是从关系  $R$  和关系  $S$  的笛卡儿积中选取属性值满足某一  $\theta$  关系的元组，记作：

$$R \bowtie_{A\theta B} S = \{t_R t_S \mid t_R \in R \wedge t_S \in S \wedge t_R[A] \theta t_S[B]\}$$

其中， $A$  和  $B$  分别为  $R$  和  $S$  上度数相等且可比的属性组， $\theta$  是比较运算符。连接运算从  $R$  和  $S$  的广义笛卡儿积  $R \times S$  中选取关系  $R$  在  $A$  属性组上的值与关系  $S$  在  $B$  属性组

上的值满足比较关系  $\theta$  的元组。

② 等值连接。当  $\theta$  为 “=” 时的连接运算称为等值连接。它是从关系  $R$  与  $S$  的广义笛卡儿积中选取  $A$ 、 $B$  属性值相等的那些元组。等值连接可记为：

$$R \bowtie_{A=B} S = \{t_{RS} | t_R \in R \wedge t_S \in S \wedge t_R[A] = t_S[B]\}$$

③ 自然连接。自然连接是一种特殊的等值连接。在连接运算中，以字段值对应相等条件进行的连接操作称为等值连接。自然连接是去掉重复属性的等值连接，即它是对行和列同时进行运算。若关系  $R$  和关系  $S$  具有相同的属性组  $B$ ，则关系  $R$  和  $S$  的自然连接可用下式表示：

$$R \bowtie S = \{t_{RS} | t_R \in R \wedge t_S \in S \wedge t_R[B] = t_S[B]\}$$

具体计算过程如下：

**步骤 01** 计算  $R \times S$ 。

**步骤 02** 设  $R$  和  $S$  的公共属性是  $A_1, A_2, \dots, A_k$ ，挑选满足  $R.A_1 = S.A_1, R.A_2 = S.A_2, \dots, R.A_k = S.A_k$  的那些元组。

**步骤 03** 去掉  $S.A_1, S.A_2, \dots, S.A_k$  的这些列。

注意：如果两个关系中没有公共属性，那么其自然连接就转化为笛卡儿积操作。

#### 4. 关系模型的键

在关系模型中，键是用于唯一标识关系中实体的属性或属性组合。关系模型中的键可以是单个属性，也可以是多个属性的组合。

单个属性键是指一个属性作为关系中实体的唯一标识符。例如，在一名学生和课程之间的关系中，学生的学号可以作为学生的唯一标识符，因此学号可以作为学生的键。

多个属性键是指多个属性作为关系中实体的唯一标识符。例如，在一个订单和订单项之间的关系中，订单号和订单项号可以作为订单的唯一标识符，因此订单号和订单项号可以作为订单的键。

关系模型中的键主要有超键、候选键、主键、外键等 4 种。

(1) 超键。

在关系中，能够唯一标识元组（关系中的一行数据）的属性集被称为关系模式的超键（super key）。一个属性可以作为一个超键，多个属性组合在一起也可以作为一个超键。超键包括候选键和主键。

(2) 候选键。

超键中最小的属性集，也就是没有冗余元素的超键，称为候选键（candidate key）。候选键是可能成为主键的属性集，可以有多个候选键同时存在，这时需要从中选择一个作为主键。候选键必须满足以下两个条件。

① 唯一性：候选键必须能唯一标识关系中的每个实体。

② 非空性：候选键不能为空值。

(3) 主键。

在关系模式中，用户正在使用的候选键被称为主键（primary key）。主键用于在数据表



中唯一和完整地标识数据对象，一个数据表只能有一个主键，且主键的取值不能缺失（即不能为空值）。主键是从候选键中选择的。主键可以结合外键来定义不同数据表之间的关系，并且可以加快数据查询的速度。主键分为两种类型：单字段主键和多字段联合主键。

#### （4）外键。

外键（foreign key）用来在两个表之间建立连接，一个表的外键可以为空值，若不为空值，则该外键的值必须是另一个表中主键的值。外键可以是一列，也可以是多列。一个表可以有一个或者多个外键。外键对应的是参照完整性，其作用是保持数据的一致性和完整性，定义外键后，即不允许删除在另一个表中与其具有关联关系的行。

对于两个具有关联关系的表而言，相关联字段中主键所在的表为主表，外键所在的表为从表。在关系模型中，键的选择是非常重要的，因为它决定了关系中实体的唯一性和完整性。如果键选择不当，可能会导致数据冗余、数据不一致等问题。

### 5. 关系模型的完整性约束

关系模型中有三类完整性约束，即实体完整性、参照完整性和用户自定义完整性。

#### （1）实体完整性。

实体完整性（entity integrity）是指关系模式中每个实体都必须有一个唯一标识符，并且每个标识符只能对应一个实体。这意味着每个实体都必须有一个唯一的主键，并且不能有重复的主键值。实体完整性可以通过定义主键、唯一约束、非空约束等方式来实现。

#### （2）参照完整性。

参照完整性（referential integrity）是指关系模式中的每个实体都必须有一个唯一的参照实体，并且每个实体只能有一个参照实体。这意味着每个实体都必须有一个唯一的外键，并且不能有重复的外键值。参照完整性可以通过定义外键、唯一约束、非空约束等方式来实现。

#### （3）用户自定义完整性。

用户自定义完整性（user-defined integrity）是指在关系型数据库中，用户可以通过定义约束条件限制数据的插入、更新和删除操作。用户自定义完整性可以提高数据的一致性和完整性，避免数据错误或不一致。在关系型数据库中，用户自定义完整性通常通过创建触发器（trigger）实现。触发器是一种特殊的存储程序，当满足特定条件时，自动执行预定义的操作，后续第 7.4 节会对触发器进行详细的讲解。

### 1.3.2 关系型数据库的概念和特点

理解关系型数据库的基本概念和特性，对于有效地设计、开发和管理数据存储系统至关重要。接下来介绍关系型数据库的概念和特点。

#### 1. 关系型数据库的概念

根据数据模型的不同，数据库可以分为关系型数据库（relational database）和非关系型数据库（non-relational database）。其中，关系型数据库是一种基于关系模型理论的数据库管理系统。它以表格的形式组织数据，每个表格由多个行和列组成，每行表示一个记录，每列表示一个属性。关系型数据库使用 SQL（structured query language）作为查询

语言，可以对数据进行增、删、改、查等操作。非关系型数据库是一种基于非关系型数据模型的数据库系统，与传统的关系型数据库不同，非关系型数据库不使用表格结构，而是使用键值对、文档、图形等不同的数据模型。非关系型数据库的数据操作和管理方式也不同于关系型数据库，通常使用特定的查询语言或应用程序编程接口进行数据操作。

## 2. 关系型数据库的特点

(1) 关系型数据库的优点。

- 事务性：关系型数据库强调 ACID 规则，即原子性（atomicity）、一致性（consistency）、隔离性（isolation）、持久性（durability），可以满足对事务性要求较高或者需要进行复杂数据查询的数据操作，而且可以充分满足数据库操作的高性能和操作稳定性的要求。关系型数据库可以控制事务原子性细粒度，一旦操作有误或者有需要，可以马上回滚事务。
- 易管理：关系型数据库采用表格的方式来组织数据，数据之间需要建立关联关系。这种数据结构清晰，易于理解和管理。关系型数据库采用 SQL 对数据库进行查询，SQL 已获得了各个数据库厂商的支持，成为数据库行业的标准。
- 可靠性与稳定性高：关系型数据库按照结构化的方法存储数据，每个数据表都必须对各个字段预先定义，再根据表的结构存入数据，从而使整个数据表的可靠性和稳定性都比较高。
- 数据完整性保障：关系型数据库通过各种约束条件（如主键、外键、唯一性约束）确保数据的完整性和正确性。例如，主键约束确保每一行数据的唯一性，外键约束保证数据之间的引用关系正确。此外，关系型数据库还支持检查约束（check constraint）和默认值约束（default constraint），通过检查条件和默认值保证数据的合法性和完整性。

(2) 关系型数据库的缺点。

- 不适合非结构化数据：关系型数据库中存储的是结构化数据，对非结构化数据的存储和查询都不是很方便。
- 数据存储需要预定义数据类型：在关系型数据库中，每个字段必须预先定义数据类型，这限制了数据的灵活性和可扩展性。
- 写入开销大：关系型数据库为了保证数据的完整性和一致性，需要执行大量的写入操作，这增加了写入的开销。

### 1.3.3 关系型数据库管理系统

了解了关系型数据库的基本概念及其优势之后，接下来介绍关系型数据库管理系统。

#### 1. 关系型数据库管理系统的概念

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套数据管理软件，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，包括数据库的建立、查询、更新，以及各种数据控制和操作。数据库管理系统的结构如

图 1-13 所示。

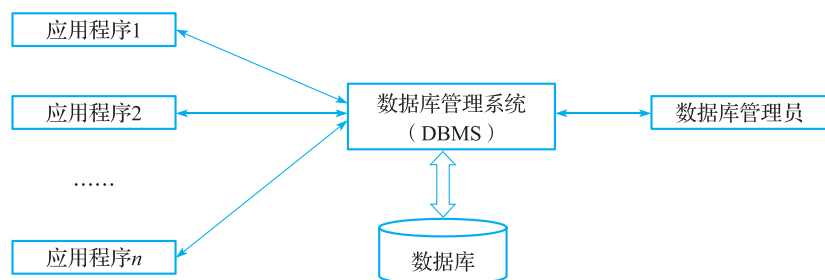


图 1-13 数据库管理系统的结构

数据库管理系统是数据库系统的核心组成部分，总是基于某种数据模型，因此，可以把数据库管理系统看作某种数据模型在计算机系统上的具体实现。基于关系模型的数据库管理系统即为关系型数据库管理系统（relational database management system, RDBMS）。RDBMS 是一种专门设计用于管理关系型数据库的系统软件，它不仅提供了对数据进行存储、管理和检索的功能，还支持数据的安全性、完整性和并发控制等关键特性。理解 RDBMS 的架构和功能对于有效地利用关系型数据库至关重要。

## 2. 关系型数据库管理系统的构成

关系型数据库管理系统的体系结构如图 1-14 所示。

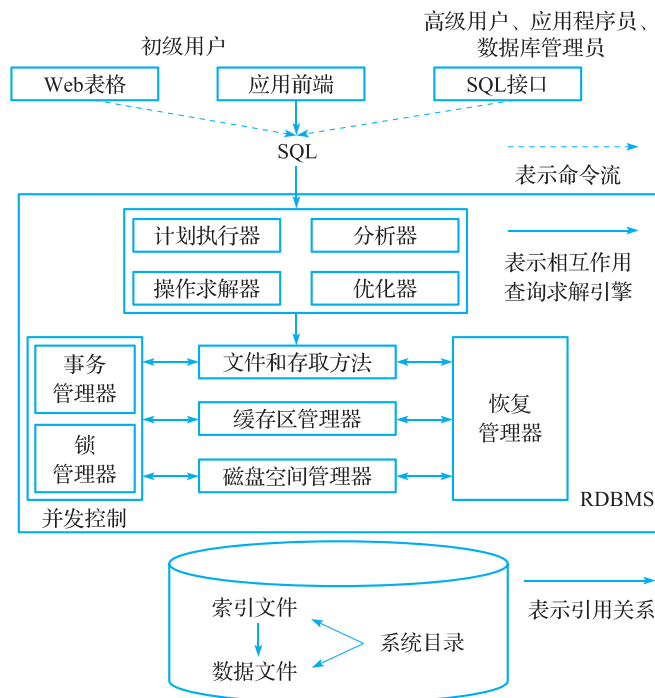


图 1-14 关系型数据库管理系统的体系结构

RDBMS 接收各种用户接口产生的 SQL 命令，生成查询求解计划，然后在数据库中执行这些计划，并返回结果。

当用户提出一个查询，经过语法分析的查询被送至查询优化器。查询优化器借助数据存储的信息生成有效求解查询的执行计划。执行计划通常表示为关系查询树。关系操作的实现代码位于文件和存取方法层之上，这一层包括支持文件概念的各种软件。在 RDBMS 中，文件是页和记录的集合，这一层通常支持堆文件或无顺序页文件，以及索引；除了跟踪文件中的页，这一层还组织页内的信息。文件和存取方法层代码位于缓存区管理器之上，缓存区管理器的责任是把页从磁盘取入主存以满足读的需求。RDBMS 软件的最底层用于实际存储数据的磁盘空间管理，其上层通过这一层执行分配、回收、读和写页的操作，该层称为磁盘空间管理器。

RDBMS 通过仔细调度用户请求和维护记录数据库所有变化的日志来支持并发控制和故障恢复。与并发控制和故障恢复相关的 RDBMS 构件包括事务管理器、锁管理器和恢复管理器：事务管理器用于确保事务依照一个合适的加锁协议来请求和释放锁，并调度执行事务；锁管理器跟踪对锁的需求，并在获得数据库对象时在该对象上授权加锁；恢复管理器负责维护日志，并在系统崩溃后把系统恢复到一个一致性状态。

## 拓展阅读

### 树根互联：工业互联网的数字化转型新基座

在数字经济浪潮席卷全球的今天，中国制造业正面临从“中国制造”向“中国智造”的深刻变革。在这个过程中，树根互联（ROOTCLOUD）应运而生，以构建全球领先的工业互联网操作系统为使命，打造面向未来的数字化转型新基座。作为国家级“跨行业跨领域”工业互联网平台企业，树根互联不仅推动着中国工业的智能化升级，更在全球制造业数字化转型中展现出中国技术的力量与责任。

树根互联的核心产品——根云平台，是一套覆盖设备连接、数据治理、智能应用与产业协同的完整工业互联网操作系统。该平台基于高性能数据库技术，实现对海量设备数据的实时采集、存储和管理，为数据治理提供可靠基础。该平台采用四层架构设计，具备从工厂现场到云端的全链路数字化能力，支持对 1 200 多种工业协议进行解析，连接了全球超过 120 万台工业设备，并为装备制造、汽车制造、钢铁冶金等 30 多个细分行业的近千家产业链企业提供服务。这些企业的设备产生的实时数据通过分布式数据库系统进行持久化存储和高效查询，确保数据一致性和可扩展性，支撑平台对多源异构数据的整合与分析。凭借其深厚的技术积累、广泛的行业覆盖与强大的生态协同能力，树根互联正成为中国智造走向世界的数字底座，书写工业互联网时代的“中国方案”。而数据库技术作为其底层支撑，不仅优化了数据流处理性能，还通过备份恢复机制确保了业务连续性，彰显了工业互联网中数据驱动的核心价值。



任务实施

EB-Shop 线上书城数据库管理系统分析与选型

本任务基于 EB-Shop 线上书城的项目需求，结合数据库理论基础，对数据库管理系统进行综合分析，并完成选型决策。

1. 数据库类型选择

在数据库选型中，首要任务是确定适合该项目的数据库类型。

通过对 EB-Shop 线上书城业务特征的深入分析可以发现，其具备典型的结构化数据特征，即用户信息、图书资料、订单记录等核心数据实体明确，且存在清晰的关联关系，如用户与订单的一对多关系、订单与图书的多对多关系等。同时，电商业务场景对数据一致性要求极高，库存扣减与订单生成等操作必须具备原子性，支付数据必须绝对准确。此外，业务过程中涉及大量的关联查询、聚合统计（如商品搜索、销售分析）和复杂的数据更新操作。

基于以上业务特征，对比不同类型的数据模型可以发现，关系型数据库展现出其独特优势。

- 关系型数据库的二维表结构非常适合表示结构化数据。
- 数据库所支持的 ACID 事务特性能够从根本上保障业务数据的一致性。
- SQL 语言为复杂查询和数据操作提供了成熟、强大的支持。

因此，从数据模型的理论特性与项目实际需求匹配的角度出发，关系型数据库是支撑 EB-Shop 线上书城项目更为合适的技术选择。

2. 主流关系型数据库管理系统对比分析

常见的关系型数据库管理系统包括 Oracle、SQL Server、Access、MySQL、PostgreSQL 和 SQLite 等。在对比分析常见的关系型数据库管理系统时，可以从多个维度进行考量。

表 1-2 是 Oracle、SQL Server、MySQL、PostgreSQL、SQLite、Access 这几种数据库管理系统的详细对比分析。

表 1-2 关系型数据库管理系统对比分析

数据库系统	开发商	主要特点	适用场景	优点	缺点	许可证类型
Oracle	Oracle 公司	功能全面、稳定性高、安全性强	大型企业、金融系统、高并发事务处理	性能卓越、支持海量数据、完善的集群方案	价格昂贵、管理复杂、资源占用高	商业授权
SQL Server	微软公司	与 Windows 集成度高、易用性好	企业级应用、Windows 平台开发	图形界面友好、支持多种开发语言、良好的商业智能支持	主要支持 Windows 平台、企业版价格高	商业授权（有免费 Express 版）
MySQL	Oracle（收购）	开源、轻量级、速度快	中小网站、Web 应用、嵌入式系统	免费开源、跨平台、社区支持强大	功能较商业数据库少、集群方案较弱	GPL 开源协议

(续表)

数据库系统	开发商	主要特点	适用场景	优点	缺点	许可证类型
PostgreSQL	开源社区	功能丰富、标准兼容性好	复杂业务系统、地理信息系统	完全开源、扩展性强、支持 JSON 和 NoSQL 特性	性能优化较复杂、资源占用较高	BSD 开源协议
SQLite	开源项目	零配置、嵌入式	移动应用、桌面应用、嵌入式设备	无须服务器、单文件存储、跨平台	不支持高并发、无用户管理	公有领域
Access	微软公司	桌面级、易用性强	小型应用、个人数据管理	集成 Office 套件、可视化、操作简单	仅支持 Windows、数据量有限	商业授权

图 1-15 为数据库管理系统选型对比示意图。

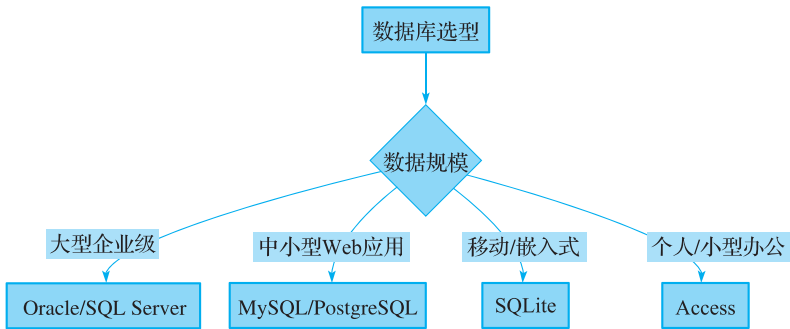


图 1-15 数据库管理系统选型对比示意图

### 3. 推荐选用 Microsoft SQL Server 2022 的充分理由

基于对主流关系型数据库管理系统的综合对比，并结合 EB-Shop 线上书城的项目需求，选择 Microsoft SQL Server 2022 作为 EB-Shop 线上书城的数据库管理系统。

这一选择主要基于以下几个关键因素。

(1) SQL Server 与项目已确定的 Windows Server 操作系统和 .NET Core 开发框架天生一体。这种集成极大地简化了开发、部署和调试流程，提升了开发效率，降低了技术栈不兼容的风险。

(2) SQL Server 完备的 ACID 事务支持，能够确保高并发场景下每一笔交易的数据完整性和一致性，确保交易过程的可靠性；而其强大的查询优化器和分析服务组件，能高效完成业务中的各类数据查询与分析，并在高访问压力下保持稳定。

(3) 同行业应用实例（如当当网、豆瓣阅读）证明了 SQL Server 在电商领域拥有出色的稳定性和性能表现，完全能够满足 EB-Shop 项目的性能要求，并具备良好的纵向扩展能力。

(4) SQL Server Management Studio (SSMS) 提供了功能强大且直观的图形化界面，方便数据库管理员进行日常管理、性能监控和故障排查。

(5) 对于 EB-Shop 这样的项目, 可以选择免费的 SQL Server Developer Edition 用于开发测试, 生产环境则可选用性价比高的 Standard Edition。这种灵活的许可策略, 能够在满足功能需求的同时, 有效控制项目初期的成本。

综合以上分析, Microsoft SQL Server 2022 在技术适配性、功能完备性、可维护性及成本效益方面, 与 EB-Shop 线上书城的项目需求形成了最佳匹配, 因此被选为项目的数据库管理系统。



### 任务评价

根据高等职业教育“德技并修”的目标制定本任务的任务评价表, 如表 1-3 所示。此评价表通过量化评分与质性反馈结合, 旨在帮助学生明确改进方向。

表 1-3 “数据库基础”任务评价表

评价维度	评分标准	评价要点	分值	得分
知识掌握 (40 分)	9 ~ 10 分: 能准确、深刻地阐述核心概念, 逻辑清晰, 无错误 7 ~ 8 分: 能基本正确地阐述概念, 理解无重大偏差, 逻辑基本清晰 5 ~ 6 分: 能复述关键知识点, 但理解较浅或存在个别表述错误 0 ~ 4 分: 关键概念混淆, 或理解存在严重错误	1. 数据库系统基本概念 掌握数据库、数据、信息的定义与区别, 理解数据库系统的组成与作用	10	
		2. 数据模型层次理解 理解概念、逻辑、物理数据模型三个层次的定义、作用与区别	10	
		3. 关系数据库基本理论 熟悉关系模型、关系运算、完整性约束等基本理论	10	
		4. 数据库系统架构 理解集中式、分布式等不同数据库系统架构的特点	10	
能力达成 (40 分)	9 ~ 10 分: 能全面分析问题, 论证充分有力, 结论明确可信 7 ~ 8 分: 分析维度完整, 论证过程清晰, 结论合理 5 ~ 6 分: 能完成主要分析, 但论证缺乏深度或广度 0 ~ 4 分: 无法完成有效分析, 或结论存在严重错误	5. 场景分析与要素识别 能根据 EB-Shop 场景, 分析其数据库系统的构成要素	10	
		6. 概念模型抽象能力 具备将现实世界信息抽象为概念模型的能力, 理解 E-R 图的基本构成	10	
		7. DBMS 对比分析能力 能通过对比主流 DBMS, 理解其技术特点、适用场景与差异	10	
		8. 技术选型与论证能力 能为项目提供合理的 DBMS 选型建议, 并给出充分的技术与成本理由	10	

(续表)

评价维度	评分标准	评价要点	分值	得分
素质体现 (20分)	9~10分:能主动、深入地将安全、协作等素养融入项目实践,表现突出	9.信息安全与责任意识 在分析中体现出对数据安全、隐私保护和职业道德的关注	10	
	7~8分:能清晰体现素养要求,在项目输出中符合规范 5~6分:能被动提及概念,未能在项目实践中有效体现 0~4分:未体现出应有的职业素养意识	10.严谨协作精神 文档规范、逻辑严谨;在团队中能进行有效沟通与协作	10	
总分合计			100	

## 课后习题

### 一、选择题

- 长期存放在计算机内、有组织的、可共享的数据集合称为( )。
  - 数据
  - 数据库
  - 数据库系统
  - 数据库管理系统
- 下列( )选项不是数据库系统架构。
  - 集中式架构
  - 分布式架构
  - B/C
  - B/S
- 下列有关数据库的描述,正确的是( )。
  - 数据库是一个结构化的数据集合
  - 数据库是一个关系
  - 数据库是一个DBF文件
  - 数据库是一组文件
- 下列不属于数据库系统人员的是( )。
  - 数据库管理员
  - 终端用户
  - 数据库营销专员
  - 应用程序开发人员
- 下列不属于数据模型的组成要素的是( )。
  - 数据定义
  - 数据操作
  - 数据结构
  - 数据完整性约束
- ( )表示建立概念模型的对象,是用于唯一标识一个实体的属性或属性组合。
  - 实体
  - 实体型
  - 键
  - 域

- ## 二、填空题

- ### 三、简答题

1. 什么是数据库？什么是数据库系统？简述二者的区别和联系。
2. 组成数据库系统的人员都有哪些？他们的主要任务分别是什么？
3. 数据模型的组成要素都有什么？简要说明各组成要素的功能。
4. 什么是数据库管理系统？它有什么功能？