

巍巍文大 百年书香  
www.jiaodapress.com.cn  
bookinfo@sjtu.edu.cn



策划编辑 李 勇  
责任编辑 胡思佳  
封面设计 黄燕美



校企“双元”合作新形态教材

数据库技术基础

主编 王立力

上海交通大学出版社

免费提供  
精品教学资料包  
服务热线: 400-615-1233  
www.xinsijiaocai.com



ISBN 978-7-313-33895-2  
9 787313 338952  
定价: 45.00元





校企“双元”合作新形态教材

# 数据库技术基础

MySQL版

主 编 王立力

副主编 田晓光



上海交通大学出版社  
SHANGHAI JIAO TONG UNIVERSITY PRESS



## 内容提要

本书共分7个模块,结构合理,内容涵盖数据库基础、MySQL的安装与配置、数据库和表的基本操作、数据查询、数据更新与数据定义、事务管理与存储过程、数据库的高级操作等方面,旨在为学生提供一套完整的数据库学习体系。全书内容从数据库的基本原理入手,逐步深入介绍数据库设计、优化、查询等实操内容,力求通俗易懂并注重实际应用,通过丰富的实例和实践环节,帮助学生灵活运用数据库技术,培养学生解决实际问题的能力。

本书既可作为职业院校数据库技术课程的教材,也可供相关技术人员参考。

## 图书在版编目(CIP)数据

数据库技术基础 / 王立力主编. -- 上海 : 上海交通大学出版社, 2025. 11. -- ISBN 978-7-313-33895-2

I. TP311.13

中国国家版本馆 CIP 数据核字第 2025XR2516 号

## 数据库技术基础

SHUJUKU JISHU JICHU

主 编:王立力

出版发行:上海交通大学出版社

邮政编码:200030

印 制:三河市龙大印装有限公司

开 本:787 mm×1 092 mm 1/16

字 数:266千字

版 次:2025年11月第1版

书 号:ISBN 978-7-313-33895-2

定 价:45.00元

地 址:上海市番禺路951号

电 话:021-64071208

经 销:全国新华书店

印 张:12.5

印 次:2025年11月第1次印刷

版权所有 侵权必究

告读者:如您发现本书有印装质量问题请与印刷厂质量科联系

联系电话:0316-3655788



# 前言

在信息技术快速发展的时代，数据已成为各行各业的重要资产。数据库作为组织、存储和管理数据的核心技术，是信息系统的核心组成部分，掌握数据库技术已成为计算机及相关专业学生必备的基本技能之一。本书旨在为学生提供一本系统、实用、易学的数据库技术入门教材，帮助学生掌握数据库的基本概念、技术方法，提升实际操作能力。

本书以 MySQL 这一广泛应用的关系数据库管理系统为载体，围绕数据库的设计、操作与管理展开讲解。我们结合多年教学经验与项目实战，帮助学生更好地理解数据库基本概念，掌握数据库使用和管理技能，提升职业发展竞争力。

全书内容从数据库的基本原理入手，逐步深入介绍数据库设计、优化、查询等实操内容，力求通俗易懂并注重实际应用。本书通过丰富的实例和实践环节，帮助学生灵活运用数据库技术，培养学生解决实际问题的能力。

本书结构合理，内容涵盖数据库基础、MySQL 的安装与配置、数据库和表的基本操作、数据查询、数据更新与数据定义、事务管理与存储过程、数据库的高级操作等方面，旨在为学生提供一套完整的数据库学习体系。希望本书能够成为学生学习数据库技术的重要参考书，激发学生对数据库技术的热爱，为学生未来的职业生涯奠定坚实的基础。

本书在编写过程中力求语言简洁、示例丰富、步骤清晰，每个模块均配有“练习与思考题”，方便学生巩固所学内容。书中大量实例均来源于真实应用场景或典型教学案例，注重培养学生的实际操作能力。

本书由沈阳北软信息职业技术学院王立力担任主编，田晓光担任副主编。

尽管我们总结多年教学实践精心编写本书，但计算机技术日新月异，书中难免存在不足之处。我们诚恳地欢迎读者提出宝贵意见与建议。

编 者





# 目录

<b>模块 1</b>	<b>数据库基础</b>	<b>1</b>
1.1	数据库系统概述	2
1.2	数据模型	8
1.3	数据库的体系结构	23
1.4	练习与思考题	26
<b>模块 2</b>	<b>MySQL 的安装与配置</b>	<b>28</b>
2.1	在 Windows 平台下安装与配置 MySQL	29
2.2	MySQL 常用图形化管理工具	39
2.3	在 Linux 平台下安装与配置 MySQL	46
2.4	练习与思考题	50
<b>模块 3</b>	<b>数据库和表的基本操作</b>	<b>52</b>
3.1	数据库基础知识	53
3.2	数据类型	55
3.3	数据表的基本操作	60
3.4	表的约束	62
3.5	设置表的字段值自动增加	64
3.6	索引	66
3.7	练习与思考题	68
<b>模块 4</b>	<b>数据查询</b>	<b>71</b>
4.1	了解 SQL	72
4.2	检索数据	76
4.3	排序检索数据	85



4.4	过滤数据	88
4.5	高级数据过滤	90
4.6	用通配符进行过滤	92
4.7	创建计算字段	94
4.8	使用函数处理数据	96
4.9	汇总数据	101
4.10	数据分组	104
4.11	使用子查询	107
4.12	使用联结	112
4.13	组合查询	117
4.14	练习与思考题	119

## 模块 5 数据更新与数据定义 122

5.1	插入数据	123
5.2	更新和删除数据	127
5.3	使用视图	132
5.4	练习与思考题	142

## 模块 6 事务管理与存储过程 144

6.1	事务管理	145
6.2	存储过程的基础知识	146
6.3	自定义函数的使用	151
6.4	存储过程的使用	154
6.5	触发器的使用	158
6.6	练习与思考题	163

## 模块 7 数据库的高级操作 165

7.1	数据备份与还原	166
7.2	用户管理	169
7.3	权限管理	172
7.4	练习与思考题	175

附录 1	常用函数	177
------	------	-----

附录 2	练习与思考题参考答案	189
------	------------	-----





## 模块

# 1



# 数据库基础



## 模块导读

在现代信息系统中，数据库技术扮演着关键角色，提供有效的数据存储、管理、检索与维护手段。本模块深入探讨数据库基础，包括其核心概念、发展历程和主要组成，旨在构建坚实的学习基础。从数据库技术的发展讲起，介绍关系型数据库的特性和优势。本模块详解数据库系统的组成要素，如软件、硬件和数据。关于数据模型，本模块不仅阐释其定义和分类，还专注于关系模型，探讨其规范化和设计原则，以及实体与关系的重要性，帮助读者掌握高效、避免冗余的数据库设计方法。最后，讨论数据库三级模式结构和 ER 模型向关系模式的转换，增进对数据库内部运作和数据抽象的理解。

通过学习本模块，读者将全面理解数据库系统的基本理论，为深入学习数据库设计和应用奠定基础。



## 学习目标

- 掌握数据库的基本概念及发展历程：了解数据库系统的组成和发展。
- 理解数据模型：学习不同数据模型及关系数据库的规范化和设计。
- 学习数据库体系结构：熟悉数据库三级模式结构和 ER（实体关系）图向关系模式的转换。
- 培养数据库设计能力：通过实体与关系的概念，提高数据库设计的有效性和效率。



## 1.1

## 数据库系统概述

在深入探讨数据库的基础知识之前，先简要了解一些基本且广泛使用的数据库术语。

### 1.1.1 数据库技术的产生和发展

数据库技术的产生旨在满足数据管理的需求。

数据管理涵盖了数据的分类、组织、编码、存储、检索及维护等关键环节，这些环节构成了数据处理的核心。数据处理则包括数据的收集、存储、加工与传播等一系列活动。随着计算机技术的进步和应用需求的增长，数据管理技术逐步从人工管理发展到文件系统，再到数据库系统。这三个阶段的特点及比较如表 1-1 所示。

表 1-1 数据管理三个阶段比较

背景及特点		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	没有操作系统	有文件系统	有数据库管理系统
	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
特点	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	现实世界（一个部门、企业、跨国组织等）
	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，冗余度小

(续表)

学习笔记

背景及特点		人工管理阶段	文件系统阶段	数据库系统阶段
特点	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
	数据控制能力	应用程序自己控制	应用程序自己控制	由数据库管理系统提供数据安全性、完整性、并发控制和恢复能力

1. 人工管理阶段（20 世纪 50 年代中期以前）

在这个时期，计算机的主要应用领域是科学计算。硬件设备仅限于纸带、卡片和磁带等，缺乏磁盘等直接存取的存储介质；在软件方面，既没有操作系统也缺少管理数据的专门软件，数据处理仅限于批处理模式。

2. 文件系统阶段（20 世纪 50 年代后期至 60 年代中期）

随着磁盘和磁鼓等直接存取存储设备的出现，硬件条件得到显著改善。软件方面，操作系统开始集成专门的数据管理软件，即文件系统。数据处理方式也有所进步，不仅保留了批处理，还引入了联机实时处理。

3. 数据库系统阶段（20 世纪 60 年代后期至今）

进入这一阶段，计算机管理的数据规模和应用范围不断扩大，数据量暴增。应用的多样性和对数据共享的需求日益增强。这个时期的硬件已经能够提供大容量磁盘，硬件成本下降，而软件成本相对上升，因为系统软件 and 应用程序的开发及维护成本不断增加。在处理方式方面，对联机实时处理的需求更加迫切，并开始考虑分布式处理的可能性。由于文件系统已不能满足日益增长的应用需求，数据库技术随之诞生，数据库管理系统（database management system, DBMS）作为一种统一管理数据的专门软件系统，为解决多用户、多应用数据共享需求的关键技术。



## 学习 笔记

数据库管理系统不仅满足了数据共享和多样化应用的需求，还标志着数据管理技术的一大飞跃。下面详细探讨数据库系统的特性及其带来的显著优势。

### 1.1.2 数据库系统的组成

数据库系统是指在计算机系统中引入数据库后的系统，一般由数据库、数据库管理系统（及其开发工具）、应用系统、数据库管理员构成。首先，了解数据、数据库、数据库管理系统和数据库系统这四个与数据库技术密切相关的基本概念。

#### 1. 数据

数据（data）是数据库存储的核心元素，不限于数字，还包括文本（text）、图形（graph）、图像（image）、音频（audio）、视频（video）等多种类型。它们可以是关于学生记录、货物运输等的信息，通过数字化存入计算机。在计算机科学中，数据经常被转换和编码为可由计算机程序识别和处理的格式。简而言之，数据是对现实世界事物的客观反映，以便于被人们理解、使用和交流。

对数据做如下定义：对事物属性或状态的符号性记录。数据的解释是指对数据含义的说明，数据的含义称为数据的语义，数据与其语义是不可分的。

在日常生活中，人们通常使用自然语言（如汉语）来描述事物。例如，描述一名学生的基本信息可能会说：王五，男，2006 年 12 月出生，来自辽宁省沈阳市，2024 年入学的计算机系学生。在计算机系统中，这段描述可以被整理为一个有结构的记录：（王五，男，200612，辽宁省沈阳市，计算机系，2024），其中包括了学生的姓名、性别、出生年月、家乡、院系和入学时间等信息。这种记录方式显示了如何通过计算机来组织和存储有结构的数据。

#### 2. 数据库

数据库（database，DB）是长期存储在计算机存储设备内，有组织的、共享的数据集合。它旨在具有较小的冗余度（redundancy）、较高的数据独立性（data independency）、易扩展性（scalability）

和用户共享性 (shareability) 等特点, 在数据库设计和管理中, 这些特性是至关重要的。

随着科技的迅猛发展, 数据收集和处理需求不断增长, 数据量也急剧上升。为了有效利用这些数据, 人们从传统的文件柜存储转向使用计算机和数据库技术进行科学管理。这不仅便于保存和加工处理大量复杂数据, 还能充分挖掘其中的有价值信息, 优化资源利用。

### 3. 数据库管理系统

数据库管理系统 (DBMS) 是核心系统软件, 负责有效地组织、存储、管理和处理数据, 保障数据的持久性、结构化和共享性。它通过数据定义语言 (data definition language, DDL) 和数据操纵语言 (data manipulation language, DML) 提供了一种方式, 让用户可以灵活地定义数据结构并执行数据查询、更新等操作。DBMS 的主要职责还包括事务管理、系统维护和保障数据的安全与完整性。

作为用户和操作系统的中间层, DBMS 有以下多项功能。

(1) 数据定义功能: 允许用户通过 DDL 定义数据库中的数据对象。

(2) 数据组织、存储和管理: 负责各类数据 (数据字典、用户数据、数据的存取路径等) 的组织、存储和管理, 提供多种存取方法 (如索引查找、Hash 查找、顺序查找等), 目的是提高存储空间利用率和存取效率。

(3) 数据操纵功能: 通过数据操纵语言 (DML), 实现对数据库的基本操作, 如查询、插入、删除和修改等。

(4) 数据库的事务管理和运行管理: 数据库在建立、运用和维护时由数据库管理系统统一管理和控制, 以保证数据的一致性、安全性、完整性、多用户并发使用及发生故障时的数据恢复。

(5) 数据库的建立和维护功能: 数据备份、恢复、重组及性能监控等, 确保数据库长期、稳定地运行。

此外, DBMS 还支持与网络中其他系统的数据通信和转换, 以及具备异构数据库间的互操作性。它是任何数据库系统不可或缺的

#### 学习 笔记





## 学习 笔记

一部分,使数据能够被科学地组织和高效地访问。

## 4. 数据库系统

数据库系统(database system, DBS)是指集成了数据库(DB)、数据库管理系统(DBMS)、应用程序(Application, App)及数据库管理员(database administrator, DBA)的完整计算机系统。DBA 扮演关键角色,负责数据库的建立、使用和维护。

数据库系统的特点如下。

## 1) 数据结构化

数据结构化,即考虑到整个组织的数据结构而不仅仅是单个应用的需求。这是数据库的主要特征之一,也是数据库系统与文件系统的本质区别。

在关系数据库中,表之间的关系可以通过参照完整性来确保,这将在1.2节进行详细解释。例如,若尝试为不在学生表(xsxx\_t)的学生添加考试成绩到成绩表(xsxk\_t),关系数据库管理系统(relational database management system, RDBMS)将拒绝这种插入操作,以此保障数据的准确性。相比之下,在文件系统中实现同样的约束需要程序员手动编写额外代码。

例如,一个学校的信息系统中不仅要考虑教务处的学生学籍管理、选课管理,还要考虑学生处的学生人事管理等。因此,学校信息系统中的学生数据就要面向各个处室的应用而不仅仅是教务处的一个学生选课应用。可以按照图1-1所示的信息系统组织其中的学生数据。

学生基本信息

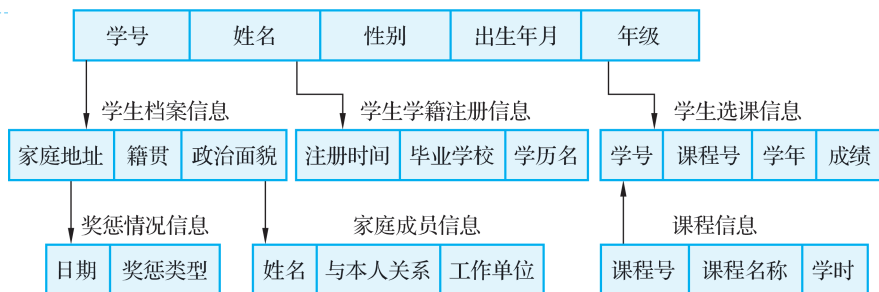


图 1-1 某学校信息系统中的学生数据

这种数据组织方法为各部门的应用提供了所需的记录,实现了

数据的整体结构化，并且在描述数据时，不仅阐明了数据本身，还明确了数据之间的关系。在数据库系统中，数据不只是被整体结构化，数据访问方式也显得额外灵活，能够针对单个数据项、一组数据项、单条记录或多条记录进行存取。相比之下，在文件系统中，数据访问仅限于记录层面，无法精细至单个数据项。

### 2) 数据的共享性高，冗余度低，易扩充

数据库系统以整体视角管理数据，支持多用户和应用的数据共享，显著减少数据冗余和不一致问题，提高存储效率。数据共享还促进了系统的灵活性和扩展性，便于加入新应用和满足变化的用户需求，通过选择数据子集以适应不同应用的变化。

### 3) 数据独立性高

高数据独立性是数据库的核心优势，包括物理独立性和逻辑独立性。物理独立性意味着应用程序与数据的存储无关，由 DBMS 管理，确保了物理存储变化不影响应用程序。逻辑独立性指应用程序与数据库的逻辑结构无关，改变数据结构不需要修改应用程序。这种独立性简化了应用开发和维护流程，由 DBMS 通过层次映射机制保证。

### 4) 数据由 DBMS 统一管理和控制

在统一管理和控制下，数据库支持并发访问，多用户可以同时操作相同的数据项。DBMS 负责维护数据的完整性、安全性，并在多用户环境中管理并发访问，以及在系统故障时恢复数据。

总之，数据库是一个组织良好、长期存储在计算机内的共享数据集合，它可以最小化数据冗余，保持高度的数据独立性。DBMS 的存在确保了数据库的有效建立、运用和维护，是计算机科学中快速发展的领域之一，其进展主要围绕数据模型的演化。



#### 温馨提示

在一般不引起混淆的情况下常常把数据库系统简称为数据库。

#### 学习笔



## 1.2

## 数据模型

人们对模型，特别是具体模型并不陌生。一辆汽车、一组数字校园的元宇宙模型、一架精致的航模飞机等，都是具体模型的例子。这些模型一目了然，能够让人们立刻想到它们所代表的现实世界中的对象。模型是对现实世界对象特征的一种模拟和抽象。以模型飞机为例，它模拟了飞机飞行的各个阶段，如起飞、飞行和降落，同时概括了飞机的基本特征，包括机头、机身、机翼和机尾。

### 1.2.1 数据模型的概念

数据模型 (data model) 是一种对现实世界中的数据特征进行抽象的工具，它用于描述数据的结构、组织方式，以及如何对数据进行操作。因为计算机无法直接处理现实世界的具体事物，人们需要将这些事物数字化，即通过数据模型将现实世界中的对象、事件和概念转化为计算机可以处理的形式。

通俗地讲，数据模型是对现实世界的一种抽象和模拟。所有的数据库系统都建立在特定的数据模型之上，这使得数据模型成为了解和学习数据库技术的基础。数据模型构成了数据库系统的核心框架，是数据库设计和实现的基石。

### 1.2.2 常见的数据模型

在数据库系统中，数据模型是用来定义数据结构、存储方式和操作方法的框架。常见的数据模型可以分为关系模型和非关系模型两大类。每种模型都有其特定的应用场景和优缺点。

#### 1. 关系模型

##### 1) 定义

关系模型 (relational model) 是数据库系统中最传统也是使用最广泛的数据模型之一。它通过二维表格 (关系) 来组织数据，每

张表由行（记录）和列（属性）组成。表与表之间通过键（主键和外键）建立关系。

### 学习笔记

#### 2) 特点

(1) 表格结构：数据以表格形式存储，每行表示一条记录，每列表示一个属性。

(2) SQL：使用结构化查询语言（structured query language, SQL）进行数据操作和查询。

(3) 数据独立性：数据的物理存储与逻辑结构分离，允许对数据结构进行修改而不影响应用程序。

(4) 约束和完整性：支持主键、外键等数据完整性约束，确保数据的一致性和准确性。

#### 3) 优点

(1) 灵活性高：可以通过 SQL 进行复杂查询和操作。

(2) 数据一致性：数据完整性约束确保数据的一致性和准确性。

(3) 易于理解：表格结构直观，容易理解和操作。

#### 4) 缺点

(1) 性能问题：在处理大规模数据和高并发访问时，性能可能成为瓶颈。

(2) 复杂查询：复杂的查询可能导致性能下降，需要优化。

#### 5) 应用场景

关系模型适用于事务处理和复杂查询的应用，如金融系统、企业资源计划（enterprise resource planning, ERP）系统等。

## 2. 非关系模型

#### 1) 定义

非关系模型（non-relational model 或 NoSQL）指不基于关系模型的数据库系统，通常用于处理大规模数据和高并发需求。它包括多种类型的数据库，每种类型都有其独特的存储方式和操作模型。

#### 2) 主要类别

(1) 键值存储（key-value store）。



## 学习 笔记

定义：数据以键值对的形式存储，键是唯一的，值可以是简单的数据类型或复杂的数据结构。

特点：操作简单，读写速度快，适合快速检索。

应用场景：适用于缓存、会话存储等场景，如 Redis。

(2) 文档存储 (document store)。

定义：数据以文档格式存储，文档通常是 JSON、XML 或 BSON 格式，内部可以包含复杂的嵌套数据结构。

特点：灵活的数据结构，支持动态模式，适合处理结构化和半结构化数据。

应用场景：适用于内容管理、电子商务等场景，如 MongoDB。

(3) 列族存储 (column-family store)。

定义：数据以列族的形式存储，每列族包含多列，适合存储大规模数据。

特点：高效的读写性能，适合大规模数据分析。

应用场景：适用于大数据分析、日志存储等场景，如 Cassandra 和 HBase。

(4) 图形存储 (graph store)。

定义：数据以图形结构存储，节点表示实体，边表示实体之间的关系。

特点：擅长处理复杂的关系和连接查询。

应用场景：适用于社交网络、推荐系统等场景，如 Neo4j。

### 3) 非关系型数据库的优点

扩展性强：大多数非关系型数据库支持水平扩展，适合处理大规模数据。

灵活性高：数据模型灵活，能够处理多种数据结构。

高性能：适合高并发访问和大数据存储。

### 4) 非关系型数据库的缺点

一致性问题：通常不提供强一致性保障，可能需要应用层进行协调。

查询能力：相比关系型数据库，查询功能可能较为有限。



### 5) 非关系型数据库的应用场景

非关系型数据库适用于需要高扩展性、高性能和灵活数据模型的应用，如社交网络、物联网、大数据分析等。

关系模型和非关系模型各有特点，选择合适的数据库模型需要考虑具体的应用场景和需求。关系模型适合处理结构化数据和复杂查询，而非关系模型则在处理大规模数据、高并发需求和灵活数据结构方面表现突出。



上文所提及的数据模型是逻辑层面的，即用户所见的数据结构。这些模型能够通过特定语言描述，以便计算机系统理解，并得到数据库管理系统（DBMS）的支持，形成数据库中的数据视图。实际上，这些数据按照 DBMS 的物理存储模型在数据库系统中进行存储，这一过程完全由 DBMS 负责。

## 1.2.3 关系型数据库的规范化——数据模型的组成要素

数据模型是定义数据库的核心框架，共性为组成要素，精确描述了系统的静态特性、动态特性和完整性约束条件。一个完整的数据模型通常包括数据结构、数据操作和完整性约束条件三个主要部分。

### 1. 数据结构

数据结构描述了数据库中对象的类型、内容和这些对象之间的关系。

对象类型：定义数据的基本组成单位。例如，关系模型中的“域”“属性”和“关系”，以及非关系模型中的“数据项”和“记录”。

对象之间的联系：描述数据之间的关联。例如，非关系模型中的图存储类型（set type）。

数据结构是数据模型的核心，决定了模型的静态特性。

### 2. 数据操作

数据操作是指对数据库中的各种对象实例允许执行的操作的集合。

### 学习笔记



## 学习 笔记

查询：获取数据库中的数据。

更新：包括插入、删除和修改操作。

数据模型需要定义这些操作的具体含义、操作符号和规则，以及用于实现这些操作的语言。

数据操作描述了系统的动态特性。

### 3. 完整性约束条件

数据的完整性约束条件是指在数据模型中定义的一组规则，用于确保数据的准确性、一致性和有效性。这些约束规定了数据及其关系的合法状态和变化规则，从而维持数据库的完整性。

## 1.2.4 关系型数据库的设计原则——概念模型

概念模型是把现实世界转换为信息世界的第一步，它帮助设计数据库并促进设计人员与用户的沟通。概念模型要求：一方面是清晰的语义表达，即直观地展示应用中的信息，让数据库准确地反映现实世界；另一方面是简洁易懂，即结构简单明了，易于理解和使用，方便用户参与设计。

### 1. 信息世界中的基本概念

在构建概念模型时，需要理解和使用一些基本的概念，以便准确地描述和组织数据。

#### 1) 实体

实体 (entity) 是信息世界中的基本构件，代表现实世界中的对象或事物。例如，在一个学生管理系统中，学生、课程和教师都可以被视为实体。

#### 2) 属性

属性 (attribute) 是对实体的描述，用于定义实体的特征或性质。例如，学生实体可以有姓名、学号和出生日期等属性。

#### 3) 关系

关系 (relationship) 描述了实体之间的关联和交互。例如，学生和课程之间可以存在“选修”关系，表示学生选择了某门课程。

#### 4) 实体集

实体集 (entity set) 是相同类型实体的集合。例如，所有学生

组成的学生集，所有课程组成的课程集。

### 5) 关系集

关系集 (relationship set) 是相同类型关系的集合。例如，所有学生选修课程的记录组成的关系集。

### 6) 键

键 (key) 用来唯一标识实体的属性或属性组合。例如，学号可以作为学生实体的主键，唯一标识每名学生。

### 7) 约束

约束 (constraint) 是对数据模型中的数据和关系施加的限制，用于确保数据的完整性和一致性。例如，外键约束用于确保关系的参照完整性。

理解这些基本概念是设计有效概念模型的基础，它们帮助设计人员准确地描述信息世界中的各种数据及其关系，从而构建出符合实际需求的数据库系统。

## 2. 两个实体型之间的联系

两个实体型之间的联系通常可以分为以下三类。

### 1) 一对一联系 (1:1)

当实体集 A 中的每一个实体最多只与实体集 B 中的一个实体 (可以没有) 发生联系时；反之，亦然，则这种关系称为一对一联系，记作 1:1。

例如，在学校中，每个班级只有一个班长，而班长也只有一个班级中担任职务。因此，班级与班长之间存在一对一的联系。

### 2) 一对多联系 (1:n)

当实体集 A 中的每一个实体可以与实体集 B 中的多个实体 ( $n \geq 0$ ) 发生联系；反之，实体集 B 中的每一个实体最多只与实体集 A 中的一个实体发生联系时，这种关系称为一对多联系，记作 1:n。

例如，一个班级中有多名学生，而每名学生只能归属于一个班级。因此，班级与学生之间属于一对多的关系。

### 3) 多对多联系 (m:n)

当实体集 A 中的每一个实体可以与实体集 B 中的多个实体 ( $n \geq 0$ ) 发生联系，同时实体集 B 中的每个实体也可以与实体集 A

## 学习笔记



## 学习笔记

中的多个实体 ( $m \geq 0$ ) 发生联系时, 这种关系称为多对多联系, 记作  $m:n$ 。

例如, 一门课程可能被多名学生选修, 而一名学生也可以同时选修多门课程, 因此, 课程与学生之间存在多对多的联系。

需要注意的是, 一对一联系可以看作一对多联系的特殊情况, 而一对多联系又可以看作多对多联系的特例。

通过精简和结构化, 使内容更加清晰、易读, 重点突出实体关系的层次性, 可以用图形直观地表示这三种实体之间的联系, 如图 1-2 所示。

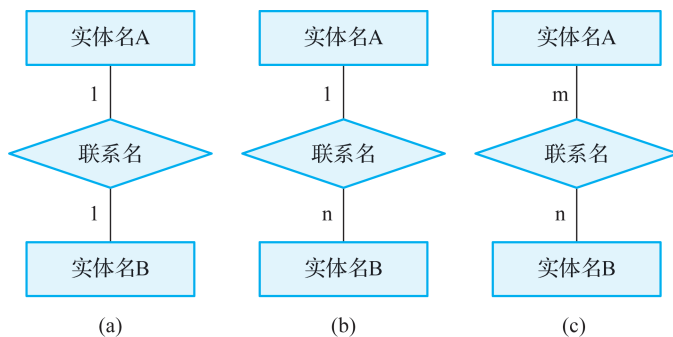


图 1-2 两个实体型之间的三类联系

(a) 1:1 联系; (b) 1:n 联系; (c) m:n 联系

### 3. 单个实体递归多元联系

递归联系是指同一个实体型内部的联系, 它可以是二元、三元甚至多元的。当多个实体型之间的联系是递归的时, 联系模型的设计需要特别考虑。递归多元联系在管理结构、组织层级或分级系统时尤其有用。

例如, 在“学生—班长—班级”场景中, 班长是一个班级的学生, 同时这个班长要管理和负责这个班级。这样的递归联系能够反映复杂的组织结构, 通过递归的方式定义学生与班长的多元联系, 并与班级实体进行关联, 如图 1-3 所示。

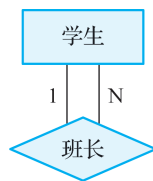


图 1-3 单个实体递归型一对多联系

### 1.2.5 关系型数据库的设计方法——实体关系图

学习笔记

在关系型数据库设计中，实体关系（ER）图是一个非常重要的工具，用于建模和设计数据库的结构。ER 图通过图形化的方式来展示实体、属性和实体之间的关系。

#### 1. 关系型数据库设计和 ER 图设计的主要知识点

实体型：用矩形表示，矩形框内写明实体名。

属性：用椭圆形表示，并用无向边将其与相应的实体型连接起来。

例如，学生实体具有学号、姓名、性别、班级、专业等属性，如图 1-4 所示。

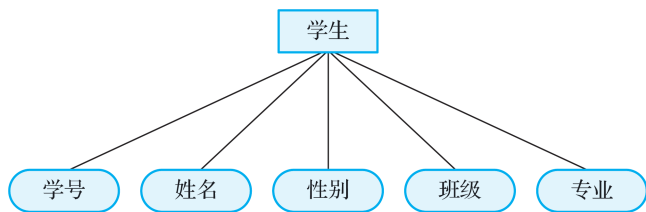


图 1-4 学生实体及属性

联系：用菱形表示，菱形框内写明联系名，并用无向边分别与有关实体型连接起来，同时无向边旁标上联系的类型（1:1、1:n 或 m:n），如图 1-5 所示。

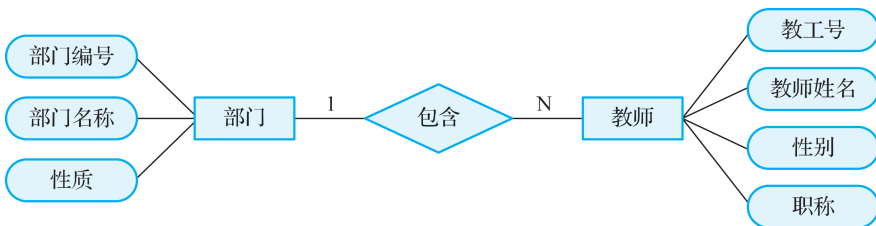


图 1-5 联系的属性

#### 2. ER 图的设计步骤

##### 1) 识别实体

分析业务需求，识别出需要建模的实体。每个实体应代表一个独立的数据对象或概念。





## 学习 笔记

### 2) 识别属性

为每个实体定义合适的属性，考虑必要的描述性信息，同时确定每个实体的主键。

### 3) 识别关系

确定实体之间的关系类型，如一对一 (1:1)、一对多 (1:n) 或多对多 (m:n)。标识关系中的外键 (foreign key)。

### 4) 确定关系的基数和可选性

定义每个关系的基数 (cardinality)。例如，一名学生可以选修多门课程，但每门课程必须至少有一名学生选修。可选性描述关系是不是必需的或可选的。

### 5) 画出 ER 图

使用软件工具或手工绘制 ER 图，确保图形清晰、明了，实体、属性和关系表示准确。

## 3. 常见的 ER 图设计工具

常见的 ER 图设计工具如下。

(1) Microsoft Visio：用于绘制各种图形，包括 ER 图，提供了丰富的模板和符号。

(2) Lucidchart：基于 Web 的设计工具，支持协作和团队工作，适用于 ER 图设计。

(3) Draw.io：免费在线绘图工具，支持 ER 图绘制，界面简单易用。

(4) MySQL Workbench：专为数据库设计而生，提供了可视化的 ER 图设计和反向工程功能。

(5) DB Designer：数据库设计工具，支持 ER 图绘制和数据库导出。

掌握 ER 图设计的基础知识，可以帮助用户更好地进行数据库建模和设计，为实现高效、可靠的数据库系统奠定坚实的基础。

## 4. 示例

下面的 ER 图为某个电商系统的示意概念模型。

电商系统的实体如下。

店铺：店铺 ID、店铺名称、地址。

## 学习笔记

商品：商品 ID、商品名称、单价、库存量、商品描述。

订单：订单 ID、订单日期、总金额。

客户：客户 ID、客户姓名、联系方式、地址号。

这些实体之间的联系如下。

- (1) 一个店铺可以销售多种商品，一个商品可以由一家店铺销售。
- (2) 一个客户可以下多个订单，一个订单只有一个客户下。
- (3) 一个商品可以被多个订单包含，一个订单可以包含多个商品。

例如，给出某个电商系统的示意概念模型 ER 图，分别为实体及属性图（见图 1-6）、实体及关系图（见图 1-7）、完整的 ER 图（见图 1-8）。为了更清晰地展示实体及其相互之间的关系，特意使用了属性图来表示实体的属性。

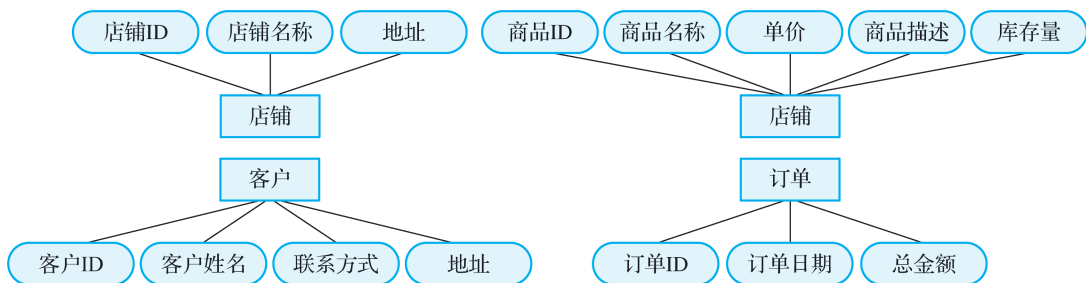


图 1-6 实体及属性图

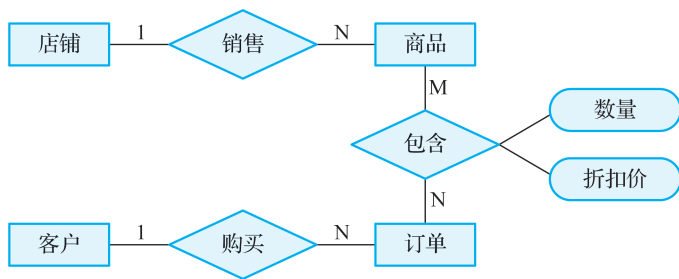


图 1-7 实体及关系图



## 学习笔记

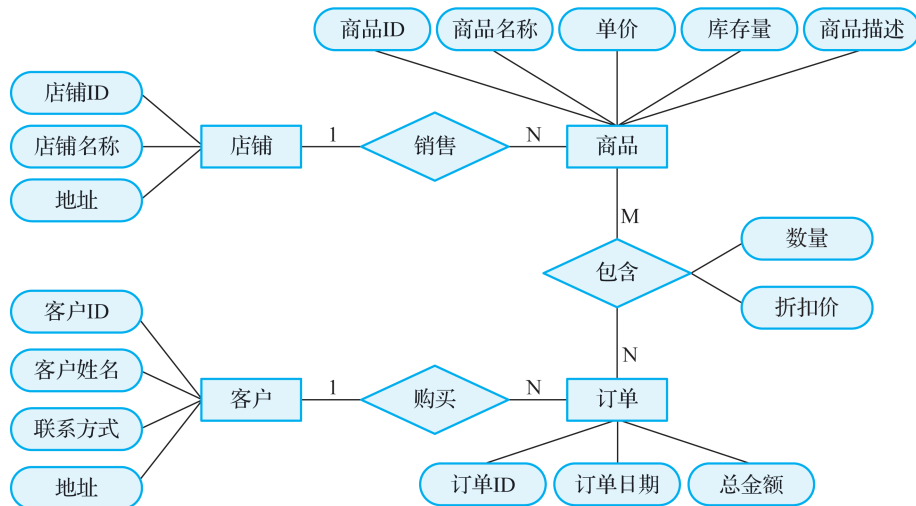


图 1-8 完整的 ER 图

用 ER 图表示的概念模型独立于具体的 DBMS 所支持的数据模型，它是各种数据模型的共同基础，因而比数据模型更一般、更抽象、更接近现实世界。

### 1.2.6 实体与关系——关系模型

在数据库设计中，实体和关系是构建数据库的重要概念，理解它们的联系和表现形式是构建关系型数据库的基础。本小节将重点介绍实体与关系的概念及它们在关系模型中的应用。

#### 1. 关系模型的基本元素

表 (relation)：关系模型中的基本结构，每张表对应一个实体集或关系集。

行 (tuple)：表中的每一行代表一个实体实例，称为元组。

列 (attribute)：表中的每一列代表实体的一个属性。

主键 (primary key)：用于唯一标识表中的每一行，也称为码。

外键 (foreign key)：用于表示不同表之间的关系。

域 (domain)：属性的取值范围，如学生的考试成绩一般为 0 ~ 100 分。

分量：元组中的一个属性值。

关系模式：对关系的描述，一般表示如下。

格式：关系名（属性 1，属性 2，…，属性 n）

## 2. 关系模型的规范化

学生模型的数据结构如表 1-2 所示。

表 1-2 学生模型的数据结构

学号	姓名	性别	班级	年级	专业
1003	王玉明	男	22101	2022	软件技术
2001	陈小峰	男	23102	2023	软件技术
...	...	...	...	...	...

例如，上面的关系可描述如下。

学生信息（学号，姓名，性别，班级，年级，专业）

在关系模型中，实体及实体间的联系都是用关系来表示的。例如，学生、课程、学生与课程之间的多对多联系在关系模型中可以表示如下。

学生信息（学号，姓名，性别，班级，年级，专业）

课程信息（课程编号，课程名称，课程类型，学时）

选课信息（课程编号，学号，成绩）

关系模型要求每个关系必须是规范化的，确保数据结构清晰、无冗余。规范化的一个基本要求是每个属性都必须是不可分的最小数据单元，即原子性。这意味着表中的每个单元格只能包含一个简单的值，不能包含其他表或复杂数据结构。通过这种设计，关系数据库可以确保数据的一致性和完整性，简化查询和更新操作。

正例：如表 1-2 所示，每个属性都是不可分的，符合关系模型要求。

反例：表 1-3 中的成绩是可分的数据项，成绩又分为平时成绩、阶段成绩和期末成绩。因此，表 1-3 不符合关系模型要求。

表 1-3 关系模型的数据结构

学号	姓名	职务	成绩			总评成绩
			平时成绩	阶段成绩	期末成绩	
12086	李力	学委	79	87	90	89
...	...	...	...	...	...	...



## 学习 笔记

### 3. 关系模型的特点

数据独立性：关系模型实现了物理数据和逻辑数据的独立性。即使数据库的底层存储结构发生改变，也不会影响到数据的逻辑结构。

简单易理解：关系模型通过表格形式表示数据，易于理解 and 操作。用户可以通过简单的查询操作来处理数据。

数据完整性：通过主键和外键约束，关系模型可以确保数据的一致性和完整性。例如，在学生和课程的关系中，通过外键约束，可以确保课程号与学生所选的课程号保持一致。

## 1.2.7 ER 图向关系模式的转换

将 ER 图转换为关系模式是从概念模型（ER 图）到逻辑模型（关系模式）的一种过程。该过程确保了数据库能够从抽象的实体关系设计转化为实际可以在关系数据库管理系统中创建和操作的表结构。

### 1. 实体集转换

在 ER 图中，实体集（entity set）表示现实世界中的对象或事物。每个实体集通常包含多个实例（实体），且每个实例有一些属性（attributes）。

转换规则如下。

（1）实体集转换为关系模式中的表：每个实体集转换为关系模式中的一个关系（表）。实体集的每个属性成为关系模式中的一列（属性）。

（2）主键：如果在 ER 图中，实体集有明确的主键码（通常标注为下划线或注明主键），那么这个主键将转换为关系表中的主键。

（3）外键：如果在 ER 图中，实体集或关系中存在明确的外键外码（通常标注为下划线波浪线或注明外键），那么这个外键将转换为关系表中的外键。

### 2. 实体间关系的转换

实体间关系也称为联系集（relationship set），表示实体之间

 学习笔记

21



## 学习笔记

的是将概念模型中的实体、属性和关系映射到具体的关系模式中。通过遵循这些转换规则，能够将抽象的设计转换为实际可用的数据库结构，为后续的数据存储和查询提供支持。

## 1.2.8 实体与关系——三类完整性约束

实体与关系的完整性约束条件包括三大类：实体完整性、参照完整性和用户定义完整性。

(1) 实体完整性 (entity integrity)：确保关系中的每个实体实例具有唯一的标识，即每个表的主键值必须唯一且非空，防止重复和缺失的记录。

例如，在“学生”表中，学号作为主键，必须唯一且非空，两名学生不能有相同的学号。

(2) 参照完整性 (referential integrity)：确保关系中的外键值必须匹配另一个关系中的主键值或为空。它维护了表与表之间的关系，防止孤立的外键引用。

例如，学生实体与专业实体的关系如图 1-10 所示。

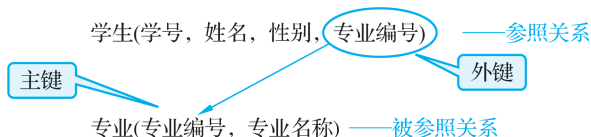


图 1-10 学生实体与专业实体的关系

学生关系引用了专业关系的主键“专业编号”，学生关系中的“专业编号”值必须是专业关系中存在的专业编号，即专业关系中有该专业的信息。

在学生关系中，称“专业编号”为外键，称学生关系为“参照关系”。

在专业关系中，称“专业编号”为主键，称专业关系为“被参照关系”。

(3) 用户定义完整性 (user-defined integrity)：根据具体的应用需求定义的完整性规则，确保数据满足特定的业务逻辑或约束条件。例如：

姓名不能为空。

年龄字段必须在合理范围 0 ~ 150 岁之间。

成绩不及格科目超过 6 门的学生不能获得毕业证。

教授的退休年龄为 65 岁，男性职工为 60 岁，女性职工为 55 岁。

性别的值只有男女。

学习笔记

## 1.3

## 数据库的体系结构

从数据库管理系统 (DBMS) 角度来看，数据库的体系结构主要包括数据模式的概念、三级模式结构及两级映像的定义与作用。这些知识点有助于学生理解数据库是如何组织和管理数据的，以便满足不同用户的需求。

### 1.3.1 数据模式的概念

数据模式是指对数据库中数据结构的描述，包括数据的组织方式、数据之间的关系及数据约束条件。数据模式提供了一种抽象的视图，使得用户能够理解和管理数据库中的数据。

(1) 逻辑模式：描述数据的逻辑结构，包括数据项、数据类型及其相互关系。逻辑模式关注的是数据的内容而非数据的存储方式。

(2) 物理模式：描述数据在存储介质上的实际存储结构，包括文件的存储方式、数据的存取路径等。物理模式与数据库的性能和效率密切相关。

(3) 模式与实例：模式是数据库的设计方案，而实例是某一时刻数据库的实际数据状态。可以将模式视为一张蓝图，而实例则是根据该蓝图建造的建筑。

数据库系统中的模式是数据库结构的抽象描述，定义了数据的类型、组织方式及其之间的关系。通常，数据库模式可以分为逻辑模式和物理模式，逻辑模式关注数据的结构和内容，而物理模式则描述数据在存储设备上的具体存储方式。





## 学习 笔记

假设有一个图书馆管理系统，系统中有三类主要数据：书籍、读者、借阅记录。

逻辑模式定义了系统中的实体和它们的关系。例如：

书籍（实体）属性：书名、作者、出版日期、ISBN 号。

读者（实体）属性：读者姓名、身份证号、联系方式。

借阅记录（实体）表示哪位读者借阅了哪本书，并包含借书日期和还书日期。

在这个逻辑模式中，系统关心的是数据的类型和关系，而不关心这些数据存储在什么设备上或存储的方式。

### 1.3.2 三级模式结构

数据库系统的三级模式结构是指数据库系统由模式、外模式和内模式三级构成，如图 1-11 所示。

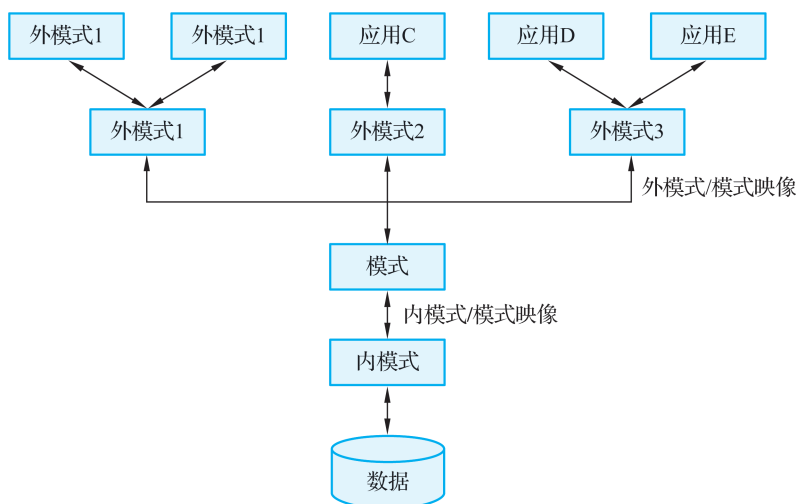


图 1-11 数据库系统的三级模式结构

#### 1. 模式

数据库模式（schema）也称为逻辑模式，是对数据库中所有数据的逻辑结构和特征的描述，是所有用户共享的数据视图。它位于数据库系统的中间层，既不涉及数据的物理存储细节，也与具体的应用程序或编程语言（如 C、COBOL、Fortran 等）无关。简单来说，模式是数据库在逻辑层面的整体视图。

## 学习笔记

一个数据库只有一个逻辑模式，模式通过某一个数据模型将所有用户的需求统一并综合，形成一个整体。定义模式时，不仅要描述数据的逻辑结构（如数据项的名称、类型、取值范围等），还需要定义数据之间的关系及数据的安全性和完整性要求。

数据库管理系统（DBMS）通过模式定义语言（DDL）来严格描述模式，确保数据结构和规则的一致性与完整性。

## 2. 外模式

外模式（external schema）也称为子模式或用户模式，是对数据库用户（包括应用程序员和最终用户）能够访问和使用的局部数据的逻辑结构的描述。它代表了数据库用户对数据的视图，通常与特定应用相关，展示了用户需要的数据及其逻辑关系。外模式一般是数据库模式的一个子集。

一个数据库可以有多个外模式，针对不同用户的需求、数据的表现形式及保密要求，外模式可以有所不同。例如，同一数据在不同外模式中可能具有不同的结构、类型、长度和保密级别。尽管一个外模式可以服务于某一个用户的多个应用系统，但每个应用程序只能使用一个外模式。

外模式还是数据库安全性的重要保障。它确保每个用户只能访问和查看其外模式中的数据，而无法接触数据库中的其他数据。数据库管理系统（DBMS）通过子模式描述语言来严格定义外模式，以确保数据访问的安全性与合规性。

## 3. 内模式

内模式（internal schema）也称为存储模式，是对数据库中数据的物理结构和存储方式的描述。每个数据库只有一个内模式，它定义了数据在数据库内部的存储和表示方式。内模式涉及多个方面。例如，记录的存储方式（堆存储、升降序存储或聚簇存储），索引的组织方式（如 B+ 树索引或哈希索引），数据是否进行压缩或加密存储，数据记录的结构类型（定长或变长结构）等。此外，还包括数据存储时的细节，如记录是否可以跨物理页存储。

数据库管理系统（DBMS）通过内模式描述语言来严格定义内模式，以确保数据的物理存储方式符合系统需求和性能要求。



## 学习 笔记

### 1.3.3 两级映像的定义与作用

数据库系统的三级模式结构定义了数据的三个抽象级别，分别是外模式、模式和内模式。通过这三级模式，数据库将数据的具体组织和存储方式交给数据库管理系统（DBMS）管理，用户可以以逻辑和抽象的方式处理数据，而无须关注数据在计算机中的具体存储形式。

为了实现这三个抽象层次之间的联系和转换，数据库管理系统提供了以下两层映像。

（1）外模式 / 模式映像：将外模式（用户视图）与模式（数据库的逻辑结构）连接起来。

（2）模式 / 内模式映像：将模式（逻辑结构）与内模式（数据的物理存储）连接起来。

这两层映像机制保证了数据库系统的数据能够具有较高的逻辑独立性（用户的视图与数据库的逻辑结构之间可以独立变化）和物理独立性（逻辑结构与物理存储之间的变化不会影响数据库的逻辑设计）。

## 1.4

## 练习与思考题

### 1. 填空题

（1）数据库管理系统是位于\_\_\_\_\_之间的一层数据管理软件。

（2）数据处理技术大致经历了\_\_\_\_\_、\_\_\_\_\_和\_\_\_\_\_三个阶段。

（3）关系的完整性约束条件包括\_\_\_\_\_、\_\_\_\_\_和\_\_\_\_\_三大类。

（4）在关系数据模型中，二维表的列称为\_\_\_\_\_，二维表的行称为\_\_\_\_\_。

（5）数据库系统的核心部分是\_\_\_\_\_，对数据库的

一切操作都是通过它进行的。

## 2. 问答题

(1) 简述数据、数据库、数据库管理系统、数据库系统的概念。

(2) 简述数据库系统的特点。

(3) 试给出三个实际情况的 ER 图，要求实体型之间具有一对一、一对多、多对多各种不同的联系。

(4) 学校中有若干系，每个系有若干班级和教研室，每个教研室有若干教员，其中有的教授和副教授每人各带若干研究生，每个班有若干学生，每个学生选修若干课程，每门课程可由若干学生选修。请用 ER 图画出此学校的概念模型。

(5) 某工厂生产若干产品，每种产品由不同的零件组成，有的零件可用在不同的产品上。这些零件由不同的原材料制成，不同零件所用的材料可以相同。这些零件按所属的不同产品类别分别放在仓库中，原材料按照类别放在若干仓库中。请用 ER 图画出此工厂产品、零件、材料、仓库的概念模型。

学习笔记