



Oracle数据库管理与应用

主编 宋 芳

北京希望电子出版社

CX-8830

责任编辑：付寒冰

封面设计：黄燕美

Oracle数据库 管理与应用

Oracle SHUJUKU
GUANLI YU YINGYONG

Oracle数据库 管理与应用

主编 宋 芳

Oracle SHUJUKU
GUANLI YU YINGYONG



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8830，获取更多资源

微信公众号



扫码下载资料包



定价：53.00元

北京希望电子出版社网址：www.bhp.com.cn

电话：010-82626270

传真：010-62543892

投稿：xiaohuijun@bhp.com.cn



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

Oracle数据库 管理与应用

主编 宋 芳



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

内容简介

本书面向数据库管理人员和数据库开发人员，从实用角度出发，系统地介绍了数据库的相关概念和原理、Oracle 数据库管理及 Oracle 数据库应用开发基础。对于 Oracle 数据库初学者，本书是一本很好的入门教程，对 Oracle 的管理员和应用程序开发人员，也有很好的学习和参考价值。

本书可作为数据库管理与应用课程的教材，也可作为 Oracle 数据库管理和开发人员的参考用书。

图书在版编目（C I P）数据

Oracle 数据库管理与应用 / 宋芳主编. — 北京：北京希望电子出版社, 2023.9

ISBN 978-7-83002-854-1

I. ①O… II. ①宋… III. ①关系数据库系统 IV.①TP311.132.3

中国国家版本馆 CIP 数据核字(2023)第 156359 号

出版：北京希望电子出版社

地址：北京市海淀区中关村大街 22 号
中科大厦 A 座 10 层

邮编：100190

网址：www.bhp.com.cn

电话：010-82620818（总机）转发行部
010-82626237（邮购）

传真：010-62543892

经销：各地新华书店

封面：黄燕美

编辑：付寒冰

校对：石文涛

开本：787mm×1092mm 1/16

印张：16

字数：379 千字

印刷：三河市骏杰印刷有限公司

版次：2023 年 9 月 1 版 2 次印刷

定价：53.00 元



前

言

PREFACE

党的二十大报告指出，教育、科技、人才是全面建设社会主义现代化国家的基础性、战略性支撑，要坚持教育优先发展、科技自立自强、人才引领驱动，加快建设教育强国、科技强国、人才强国，坚持为党育人、为国育才，全面提高人才自主培养质量。

计算机科学技术的飞速发展大大推动了社会的进步，也逐渐改变了人们的生活、工作和学习方式。其中数据库技术是计算机科学技术中发展最迅速的领域之一，也是应用最广泛的技术之一。因此，数据库系统已成为计算机信息系统与应用系统的核心技术和重要基础。Oracle数据库是数据库领域优秀的数据库管理系统之一，随着版本的不断升级，功能越来越强大。最新的版本软件可以更好地为各类用户提供完整的数据库解决方案，帮助用户建立自己的电子商务体系，从而增强了用户对外界变化的敏捷反应能力，提高了用户的市场竞争力。

本书从“统筹职业教育、高等教育、继续教育协同创新，推进职普融通、产教配合、科教融汇”的思路出发，围绕计算机相关专业的人才培养目标，按照注重基础、突出实用的原则进行内容设计，由浅入深、循序渐进地对数据库原理、Oracle基础知识、核心技术、应用方法等进行了详细介绍，能够帮助初学者快速上手，迅速提高并掌握数据库技术。本书根据不同的理论和知识点，配备了示例演示，让读者在实验环境中快速掌握知识点，以达到融会贯通、举一反三的目标。本书与实际应用紧密结合，可以使读者快速将学到的知识应用到实际工作中。

本书共10章，各章节的内容安排如下：

章节	内容概述	难点指数
第1章	主要介绍数据库的基本概念、数据模型、数据库系统的结构、数据库的规范化、数据库设计等	★☆☆
第1章	主要介绍数据库的基本概念、数据模型、数据库系统的结构、数据库的规范化、数据库设计等	★☆☆
第2章	主要介绍Oracle的发展历史及版本、Oracle的体系结构、Oracle数据库的安装、配置、监听及服务	★★☆
第3章	主要介绍SQL语言的发展历史与SQL语言的基础知识，包括数据定义、数据查询、数据操纵、数据控制等	★★★
第4章	主要介绍Oracle PL/SQL语言及编程，包括PL/SQL的程序结构、程序控制语句、游标、过程、函数、包、触发器、同义词、序列等	★★☆
第5章	主要介绍Oracle对象的基本操作，包括启动和关闭Oracle、表操作、视图操作、索引操作、数据查询与数据操纵等	★★★
第6章	主要介绍数据库的安全性、用户管理、权限管理、角色管理、概要文件管理等	★★☆
第7章	主要介绍数据库存储管理操作，包括数据文件、表空间与数据文件、控制文件、重做日志文件、归档重做日志文件等	★★★
第8章	主要介绍数据库的备份与恢复操作，包括备份与恢复的概念、逻辑备份和恢复、脱机备份与恢复、联机备份与恢复等	★★★
第9章	主要介绍闪回技术，包括闪回技术的概念、闪回查询技术、闪回错误恢复技术等	★★☆
第10章	综合应用案例	★★★

本书由河南物流职业学院宋芳主编，在编写过程中力求严谨细致，但由于时间与精力有限，疏漏之处在所难免，望广大读者批评指正。

编者
2023年8月

目录

CONTENTS

第1章 数据库技术概述

1.1 数据库的基本概念	2	1.4.1 数据依赖	22
1.1.1 数据管理的发展	2	1.4.2 相关概念	22
1.1.2 数据库与数据库管理系统	6	1.4.3 范式	23
1.1.3 数据库系统	7	1.5 数据库设计	25
1.2 数据模型	8	1.5.1 需求分析	26
1.2.1 E-R模型	8	1.5.2 概念结构设计	27
1.2.2 关系模型	11	1.5.3 逻辑结构设计	27
1.3 数据库系统的结构	16	1.5.4 物理结构设计	28
1.3.1 数据库的三级模式结构	16	1.5.5 数据库的实施	28
1.3.2 数据库的体系结构	18	1.5.6 数据库的运行和维护	29
1.3.3 数据库的连接	21		
1.4 数据库的规范化	22		



课后作业

30

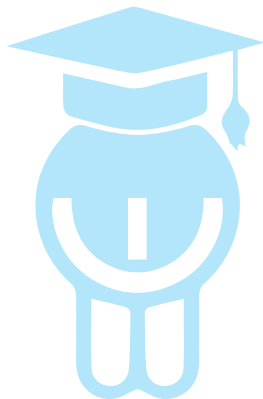
第2章 初识Oracle数据库

2.1 Oracle的发展历史及版本介绍	32	2.3 安装Oracle 19c数据库	43
2.1.1 Oracle的发展历史	32	2.3.1 Oracle 19c的安装条件	43
2.1.2 Oracle 19c版本介绍	33	2.3.2 Oracle 19c数据库的安装过程	43
2.2 Oracle体系结构概述	34	2.4 配置Oracle监听及服务	48
2.2.1 存储结构	34	2.4.1 配置Oracle监听程序	48
2.2.2 内存结构	39	2.4.2 启动与停止Oracle服务	50
2.2.3 进程结构	40		
2.2.4 数据字典	42		



课后作业

51





第3章 SQL语言基础



3.1 SQL语言简介	53	3.3.5 HAVING子句	73
3.1.1 发展历史	53	3.3.6 多表连接查询	74
3.1.2 语言特点	53	3.3.7 集合操作	80
3.1.3 Oracle示例数据库简介	55	3.3.8 子查询	82
3.2 数据定义	57	3.4 数据操纵	85
3.2.1 创建操作	57	3.4.1 插入数据	85
3.2.2 删除操作	61	3.4.2 修改数据	88
3.2.3 修改操作	62	3.4.3 删除数据	89
3.3 数据查询	63	3.5 数据控制	90
3.3.1 简单查询	64	3.5.1 授权语句	90
3.3.2 WHERE子句	67	3.5.2 授权收回语句	93
3.3.3 ORDER BY子句	69		
3.3.4 GROUP BY子句	70	课后作业	94

第4章 Oracle PL/SQL语言及编程



4.1 PL/SQL简介	96	4.5.1 基本原理	115
4.1.1 程序结构	96	4.5.2 包的创建	116
4.1.2 注释	97	4.5.3 包的调用	118
4.1.3 数据类型	98	4.5.4 包的删除	118
4.1.4 变量和常量	100	4.6 触发器	118
4.1.5 结构控制语句	100	4.6.1 基本原理	118
4.1.6 表达式	103	4.6.2 触发器的创建	120
4.2 游标	104	4.6.3 触发器的执行	120
4.2.1 游标的概念	104	4.6.4 触发器的删除	121
4.2.2 游标的处理	105	4.7 同义词	121
4.2.3 游标的属性	107	4.7.1 同义词的创建	121
4.2.4 游标变量	109	4.7.2 同义词的使用	122
4.3 过程	110	4.7.3 同义词的删除	122
4.3.1 过程的创建	110	4.7.4 同义词的查看	122
4.3.2 过程的调用	111	4.8 序列	123
4.3.3 过程的删除	112	4.8.1 序列的创建	123
4.3.4 参数类型及传递	112	4.8.2 序列的使用	124
4.4 函数	113	4.8.3 序列的修改	124
4.4.1 函数的创建	113	4.8.4 序列的删除	124
4.4.2 函数的调用	114	4.8.5 序列的查看	124
4.4.3 函数的删除	115		
4.5 包	115	课后作业	125

第5章 Oracle对象的操作基础



5.1 启动和关闭Oracle	127
5.1.1 Oracle数据库的启动	127
5.1.2 Oracle数据库的关闭	134
5.2 表	137
5.2.1 设计表	137
5.2.2 创建表	140
5.2.3 修改表	141
5.3 视图	142
5.3.1 创建视图	142
5.3.2 修改视图	145

5.3.3 删除视图	145
------------	-----

5.4 索引	145
--------	-----

5.4.1 创建索引	146
------------	-----

5.4.2 删除索引	146
------------	-----

5.5 数据查询与数据操纵	147
---------------	-----

5.5.1 对数据表进行查询	147
----------------	-----

5.5.2 新建表并批量插入记录	150
------------------	-----

5.5.3 通过视图操纵数据	150
----------------	-----



课后作业

152

第6章 数据库安全管理



6.1 数据库安全性概述	154
6.2 用户管理	154
6.2.1 创建用户	156
6.2.2 修改用户	157
6.2.3 删除用户	158
6.2.4 查询用户信息	158
6.3 权限管理	159
6.3.1 授予权限	159
6.3.2 回收权限	167
6.4 角色管理	169
6.4.1 创建角色	170

6.4.2 角色权限的授予与回收	171
------------------	-----

6.4.3 修改角色口令	172
--------------	-----

6.4.4 角色的生效与失效	173
----------------	-----

6.4.5 删除角色	173
------------	-----

6.4.6 使用角色进行权限管理	174
------------------	-----

6.4.7 查询角色信息	175
--------------	-----

6.5 概要文件管理	176
------------	-----

6.5.1 概要文件中的参数	176
----------------	-----

6.5.2 概要文件的管理	177
---------------	-----



课后作业

179

第7章 数据库存储管理



7.1 数据文件	181
7.1.1 数据文件概述	181
7.1.2 数据文件的管理	181
7.2 表空间与数据文件	184
7.2.1 表空间概述	185
7.2.2 创建表空间	187
7.2.3 修改表空间	190
7.2.4 删除表空间	192
7.2.5 表空间信息的查询	193
7.3 控制文件	194

7.3.1 控制文件概述	194
--------------	-----

7.3.2 控制文件的管理	195
---------------	-----

7.4 重做日志文件	198
------------	-----

7.4.1 重做日志文件概述	198
----------------	-----

7.4.2 重做日志文件的管理	198
-----------------	-----

7.5 归档重做日志文件	199
--------------	-----

7.5.1 归档重做日志文件概述	199
------------------	-----

7.5.2 归档重做日志文件的管理	200
-------------------	-----



课后作业

202



第8章 数据库的备份与恢复



8.1 备份与恢复概述	204	8.4 联机备份与恢复	211
8.2 逻辑备份与恢复	205	8.4.1 使用RMAN程序进行联机备份	211
8.2.1 使用expdp导出数据	206	8.4.2 使用RMAN程序进行联机恢复	216
8.2.2 使用impdp导入数据	207	8.5 各种备份与恢复方法的比较	217
8.3 脱机备份与恢复	209		
8.3.1 脱机备份	209		
8.3.2 脱机恢复	211		



课后作业

218

第9章 闪回技术



9.1 闪回技术概述	220	9.3 闪回错误恢复技术	223
9.2 闪回查询技术	220	9.3.1 闪回数据库	224
9.2.1 闪回查询	221	9.3.2 闪回表	224
9.2.2 闪回版本查询	221	9.3.3 闪回回收站	225
9.2.3 闪回事务查询	223		



课后作业

227

第10章 综合应用案例



10.1 项目概述	229	10.3.3 创建数据库对象	232
10.2 系统架构	229	10.4 应用系统设计	236
10.3 数据库设计	230	10.4.1 持久层设计	237
10.3.1 E-R图	230	10.4.2 业务逻辑层设计	241
10.3.2 表结构设计	231	10.4.3 表示层设计	244

附录

课后作业参考答案



参考文献





第 1 章

数据库技术概述



内容概要

数据库技术是实现数据管理的有效技术，是计算机科学的重要分支。许多信息系统都是以数据库为基础建立的。本章将介绍数据库的基本概念、数据模型、数据库系统的结构、数据库的规范化、数据库设计等。本章是学习后面各章节内容的预备和基础。



1.1 数据库的基本概念

数据库技术是计算机技术中发展最为迅速的领域之一，已成为人们存储数据、管理信息和共享资源时最常用的一种计算机技术。数据库技术在科学、经济、文化和军事等各个领域都发挥着重要的作用。

1.1.1 数据管理的发展

自计算机产生以来，人类社会便进入信息时代，数据处理在速度和规模上的需求已远远超出过去人工或机械方式的能力范畴，计算机以其快速准确的计算能力和海量的数据存储能力在数据处理领域得到了广泛的应用。随着数据处理的工作量呈几何级数不断增加，数据管理技术便应运而生，其演变过程随着计算机硬件和软件的发展速度及计算机应用领域的不断拓宽而不断变化。总的来说，数据管理的发展经历了人工管理、文件系统和数据库系统3个阶段。

1. 人工管理阶段

计算机没有应用到数据管理领域之前，数据管理的工作是由人工完成的。这种数据处理方式经历了很长一段时间。

20世纪50年代中期以前，计算机主要用于科学计算。当时作为外存使用的只有纸带、卡片、磁带等设备，并没有磁盘等可以直接存取的存储设备；而计算机系统软件的状况是没有操作系统，没有管理数据的软件。在这种情况下，数据管理方式为人工管理。

人工管理数据具有如下特点。

(1) 数据不被保存。

当时的计算机主要用于科学计算，一般不需要将数据长期保存，只是在计算某一课题时将数据输入，用完就撤走。

(2) 应用程序管理数据。

数据需要由应用程序自行管理，没有相应的软件系统负责数据的管理工作。应用程序中不仅要规定数据的逻辑结构，而且要设计数据的物理结构，包括存储结构、存取方法、输入方式等，因此程序员的负担很重。

(3) 数据不能共享。

数据是面向应用的，一组数据只能对应一个程序。当多个应用程序涉及某些相同的数据时，由于必须各自定义，无法互相利用、互相参照，即数据组中可能存在需要分别被不同程序处理的相同数据，也就是说存在冗余数据。

(4) 数据不具有独立性。

如果数据的逻辑结构或物理结构改变，则必须对应用程序做相应的修改，即数据是不能独立于其逻辑结构和物理结构的。这将导致进一步加重程序员的负担。

在人工管理阶段，应用程序与数据之间的对应关系如图1-1所示。

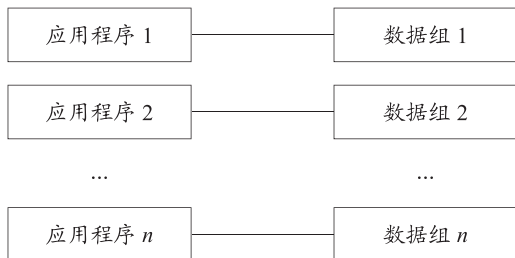


图 1-1 人工管理阶段应用程序与数据之间的对应关系

2. 文件系统阶段

20世纪50年代后期到60年代中期，硬件方面已有了磁盘、磁鼓等直接存储设备；在软件方面，不同类型操作系统的出现极大地增强了计算机系统的功能。操作系统中用来进行数据管理的部分是文件系统。用户可以把相关的数据组织成一个文件存放在计算机中，在需要时由计算机通过文件系统找出所要的文件，再对其进行处理。

文件系统管理数据具有如下优点。

(1) 数据可以长期保存。

数据可以组织成文件长期保存在计算机中反复使用。

(2) 由文件系统管理数据。

文件系统把数据组织成内部有结构的记录，实现“按文件名访问，按记录进行存取”的管理技术。文件系统使应用程序与数据之间有了初步的独立性，程序员可不必过多地考虑数据存储的物理细节。例如，文件系统中可以有顺序结构文件、索引结构文件、Hash文件等。数据在存储上的不同不会影响程序的处理逻辑。如果数据的存储结构发生改变，应用程序的改变很小，这将大大节省程序的维护工作量。

但是，文件系统仍存在以下缺点。

(1) 数据共享性差，冗余度大。

在文件系统中，一个（或一组）文件基本上对应于一个应用（程序），即文件是面向应用的。当不同的应用（程序）使用部分相同的数据时，也必须建立各自的文件，而不能共享相同的数据，因此，数据的冗余度大，浪费存储空间；同时，由于相同数据的重复存储、各自管理，容易造成数据的不一致性，给数据的修改和维护带来困难。

(2) 数据独立性差。

文件系统上的文件是为某一特定应用服务的，文件的逻辑结构对该应用来说是优化的，因此，如果要想对现有的数据再增加一些新的应用就会很困难，系统不容易扩充。一旦数据的逻辑结构发生改变，就必须修改应用程序，修改文件结构的定义，因此，数据与程序之间仍缺乏独立性。

文件系统阶段应用程序与数据之间的关系如图1-2所示。

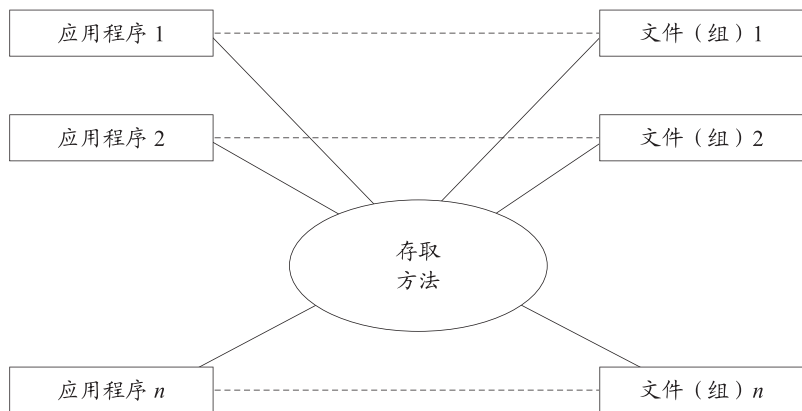


图 1-2 文件系统阶段应用程序与数据之间的关系

3. 数据库系统阶段

20世纪60年代后期，计算机用于管理的规模越来越大，应用也越来越广泛，所管理的数据量急剧增长，同时多种应用、多种语言互相覆盖的共享数据集合的要求也越来越强烈。

这一时期，硬件方面已有大容量磁盘，硬件价格也在下降，而软件价格则在上升，因而为编制和维护系统软件及应用程序所需的成本相对增加。在这种背景下，以文件系统作为数据管理手段已经不能满足应用的需要。为解决多用户、多应用共享数据的需求，使数据为尽可能多的应用服务，数据库技术便应运而生，出现了统一管理数据的专用软件系统——数据库管理系统（database management system, DBMS）。

用数据库系统管理数据相比用文件系统管理数据具有明显的优点：数据库以数据为中心组织数据，减少了数据的冗余，提供了更高的数据共享能力；同时，程序和数据具有较高的独立性，当数据的逻辑结构改变时，不涉及数据的物理结构，也不影响应用程序，从而降低了应用程序开发与维护的费用。从文件系统到数据库系统，标志着数据管理技术的飞跃。

在数据库系统阶段，应用程序与数据之间的对应关系如图1-3所示。

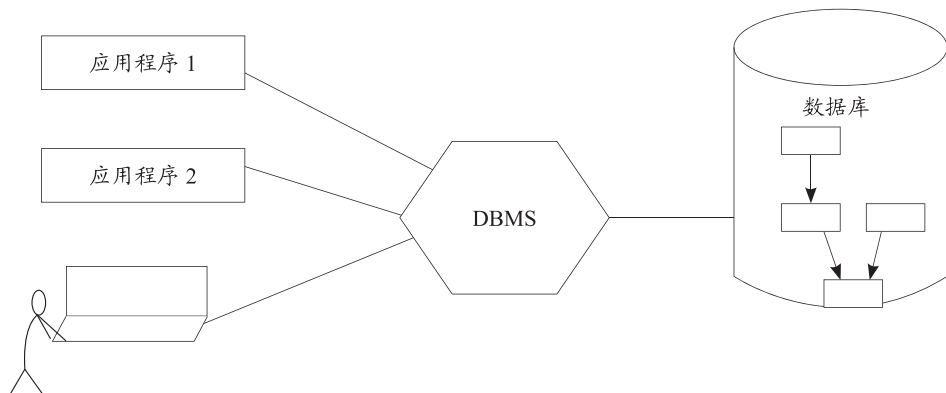


图 1-3 数据库系统阶段应用程序与数据之间的对应关系

随着计算机应用的进一步发展和网络的出现，有人提出数据管理的高级数据库阶段，这一阶段的主要标志是20世纪80年代的分布式数据库系统、90年代的对象数据库系统和21世纪初的

网络数据库系统的出现。

(1) 分布式数据库系统。

20世纪80年代以前的数据库系统是集中式的。集中式数据库把数据集中在一个数据库中进行管理,减少了数据冗余和不一致性,数据联系比文件系统更好(在文件系统中,数据分散在各个文件中,文件之间缺乏联系)。但集中式系统也有弱点:一是随着数据量增加,系统变得相当庞大,操作复杂,开销大;二是数据集中存储,大量的通信都要通过主机,会造成主机的拥堵现象。随着小型计算机和微型计算机的普及、计算机网络软件和远程通信的发展,分布式数据库系统崛起了。

分布式数据库系统主要有以下3个特点:

- 数据库的数据在物理上分布于各个场地,但在逻辑上是一个整体。
- 各个场地既可以执行局部应用(访问本地数据库),又可以执行全局应用(访问异地数据库)。
- 分布于各地的计算机通过数据通信网络相互联系。本地计算机不能单独胜任的处理任务,可以通过通信网络取得其他计算机和数据库的支持。

分布式数据库系统兼顾了集中管理和分布处理两个方面,因而具有良好的性能。

(2) 对象数据库系统。

在数据处理领域,关系数据库是相当出色的,使用也相当普遍。但是现实世界存在着许多具有复杂数据结构的应用领域,如多媒体数据、多维表格数据等的应用,已有的层次、网状和关系3种数据模型(data model)在处理这些应用领域的问题时都显得力不从心。这就需要更高级的数据库技术来表达,以便管理、构造与维护大容量的持久数据,并使这些数据能与大型的复杂程序紧密结合。对象数据库正是适应这种形势发展起来的,它是面向对象的程序设计技术与数据技术结合的产物。

对象数据库系统主要有以下两个特点:

- 对象数据库模型能完整地描述现实世界的数据结构,表达数据间嵌套、递归的关系。
- 具有面向对象技术的封装性(把数据与操作定义在一起)和继承性(继承数据结构和操作)的特点,提高了软件的可重用性。

(3) 网络数据库系统。

随着客户机/服务器(client/server, C/S)结构的出现,人们可以更有效地使用计算机资源。C/S结构也大大促进了数据库系统向网络化方面的发展。如今,计算机网络在实现通信交往、资源共享或协调工作等方面发挥着越来越大的作用,已经成为信息化社会中十分重要的一类基础设施,

网络数据库系统就是一种基于计算机网络的数据库管理系统,它允许多个用户通过网络访问和共享同一数据库[用户可以通过各种客户端设备(如PC、笔记本电脑、智能手机等)连接到服务器,并使用专门的数据库管理软件(如SQL查询工具)来访问和操作数据]。这种数据库系统的主要特点是数据的集中存储和管理,以及数据的高效访问和处理。

网络数据库系统已被广泛应用于各个领域,如企业管理、电子商务、在线教育、科研、在



线医疗健康管理等。

数据管理技术所经历的3个阶段的比较如表1-1所示。

表 1-1 数据管理技术所经历的 3 个阶段的比较

		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、管理	大规模管理
	硬件背景	无直接存储设备	磁盘、磁鼓	大容量磁盘
	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	数据库的管理者	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	一个企业（部门）的数字化信息
	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
	数据的独立性	不独立，完全依赖于程序	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	应用程序自行控制	应用程序自行控制	由数据库管理系统提供数据安全性、完整性检查，并发控制和恢复能力

■ 1.1.2 数据库与数据库管理系统

数据库、数据库管理系统是密切相关的两个基本概念，一般可以简单地将数据库理解为存放数据的仓库，而数据库管理系统是用来管理和控制数据库文件的组织、存储和访问方式的工具。

1. 数据库

顾名思义，数据库就是存放数据的仓库，只不过这个仓库在计算机的存储设备上。而数据是按照一定的数据模型组织并存放在外存上的一组相关数据的集合，通常这些数据是面向一个组织、企业或部门的。例如，在学生成绩管理系统中，学生的基本信息、课程信息、成绩信息等也都存储在学生成绩管理数据库中。人们收集并抽取出一个应用所需要的大量数据之后，应将其保存起来供进一步查询或加工处理，以获得更多有用的信息。现在，人们借助数据建模、数据库等计算机技术，不仅能科学地保存和管理大量复杂的数据，还能充分地利用这些宝贵的信息资源。

严格地讲，数据库是长期存储在计算机内的、有组织的、大量的、可共享的数据集合。数据库中的数据按一定的数据模型组织、描述和存储，具有较小的冗余度、较高的数据独立性和

易扩展性，并可为用户共享。简单来说，数据库数据具有永久存储、有组织和可共享3个基本特点。

2. 数据库管理系统

在建立了数据库之后，下一个问题就是如何科学地组织和存储数据，如何高效地获取和维护数据。而实现这一任务的，就是数据库管理系统（DBMS）。

DBMS是指数据库系统中对数据进行管理的软件系统，它是数据库系统的核心组成部分，数据库系统的一切操作，包括查询、更新及各种控制，都是通过DBMS进行的。DBMS是基于数据模型的，因此可以把它看成是某种数据模型在计算机系统上的具体实现。根据所采用数据模型的不同，DBMS可以分为网状型、层次型、关系型、面向对象型等。

如果用户要对数据库进行操作，需要由DBMS把操作从应用程序带到外部级、概念级，再导向内部级，进而操纵存储器中的数据。DBMS的主要目标是使数据成为一种可管理的资源，DBMS应使数据易于被各种不同的用户所共享，并增进数据的安全性、完整性、可用性，提供高度的数据独立性。

1.1.3 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）、应用系统和数据库管理员构成。应当指出的是，数据库的建立、使用和维护等工作只靠一个DBMS是远远不够的，还要有专门的人员来管理，这些人被称为数据库管理员（database administrator，DBA）。

在不引起混淆的情况下，可以把数据库系统简称为数据库。数据库系统组成如图1-4所示。

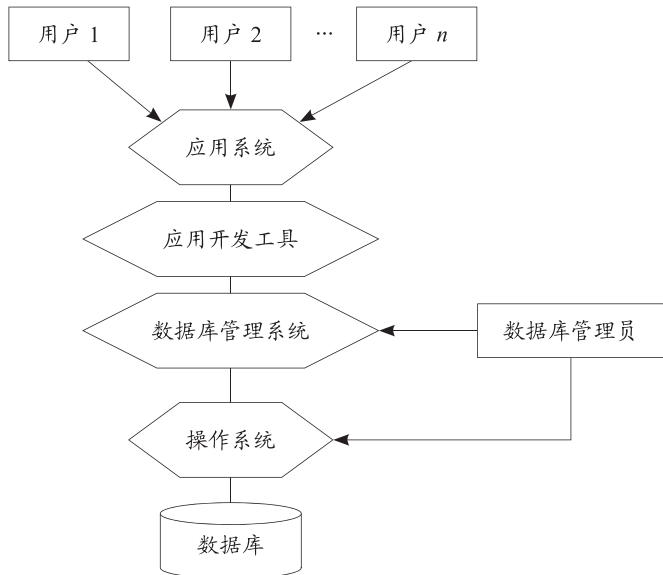


图 1-4 数据库系统组成



1.2 数据模型

模型，是对现实世界特征的模拟与抽象。例如，一组建筑规划沙盘、精致逼真的航模等，都是对现实生活中事物的描述和抽象，会让人们联想到现实世界中对应的实物。数据模型也是一种模型，它是对现实世界数据特征的抽象。由于计算机不可能直接处理现实世界中的具体事物，因此必须事先把具体事物转换成计算机能够处理的数据，即首先要数字化，要把现实世界中的人、事、物、概念用数据模型这个工具来抽象、表示和加工处理。数据模型是数据库中用来对现实世界进行抽象的工具，是数据库中用于提供信息表示和操作手段的形式构架，是现实世界的一种抽象模型。

数据模型按不同的应用层次分为3种类型，分别是概念数据模型（conceptual data model）、逻辑数据模型（logical data model）和物理数据模型（physical data model）。

1. 概念数据模型

概念数据模型又称概念模型，是一种面向客观世界、面向用户的模型，与具体的数据库管理系统无关，与具体的计算机平台无关。人们通常先将现实世界中的事物抽象到信息世界，建立所谓的“概念模型”，然后再将信息世界的模型映射到机器世界，将概念模型转换为计算机世界中的模型。因此，概念模型是从现实世界到机器世界的一个中间层次。

2. 逻辑数据模型

逻辑数据模型又称逻辑模型，是一种面向数据库系统的模型，它是概念模型到计算机之间的中间层次。概念模型只有在转换成逻辑模型之后才能在数据库中得以表示。目前，逻辑模型的种类很多，其中比较成熟的有层次模型、网状模型、关系模型、面向对象模型等。

这4种逻辑数据模型的根本区别在于数据结构不同，即数据之间联系的表示方式不同。

- (1) 层次模型用“树结构”表示数据之间的联系。
- (2) 网状模型用“图结构”表示数据之间的联系。
- (3) 关系模型用“二维表”表示数据之间的联系。
- (4) 面向对象模型用“对象”表示数据之间的联系。

3. 物理数据模型

物理数据模型又称物理模型，它是一种面向计算机物理表示的模型，此模型是数据模型在计算机上的物理结构表示。

数据模型通常由三部分组成，分别是数据结构、数据操纵和完整性约束，也称为数据模型三要素。

■ 1.2.1 E-R模型

概念模型中最著名的是实体-联系模型（entity-relationship model，E-R模型）。E-R模型是陈品山（Peter P.Chen）于1976年提出的。这个模型直接从现实世界中抽象出实体型内部及实体型之间的联系，然后用实体-联系图（E-R图）表示数据模型。设计E-R图的方法称为E-R方法。

E-R图是设计概念模型的有力工具，其中涉及很多术语。

1. 相关术语

(1) 实体 (entity)。

现实世界中客观存在并可相互区分的事物叫作实体。实体可以是一个具体的人或物，如王伟、汽车等；也可以是抽象的事件或概念，如购买一本图书、学生与课程之间的联系等。

(2) 属性 (attribute)。

实体的某一特性称为属性，如学生实体有学号、姓名、年龄、性别、系等属性。属性有“型”和“值”之分：“型”即为属性名，如姓名、年龄、性别是属性的“型”；“值”即为属性的具体内容，如“990001,张立,20,男,计算机”，这些属性值的集合表示一个学生实体。

(3) 实体型 (entity type)。

若干个属性的型组成的集合可以表示一个实体的类型，简称实体型，如“学生(学号,姓名,年龄,性别,系)”就是一个实体型。

(4) 实体集 (entity set)。

同型实体的集合称为实体集，如所有的学生、所有的课程等。

(5) 码 (key)。

能唯一标识一个实体的属性或属性集称为实体的码。例如，学生的学号可作为码，而学生的姓名不能作为学生实体的码，因为学生有可能重名，姓名不能唯一标识出学生。

(6) 域 (domain)。

属性值的取值范围称为该属性的域。例如，学号的域为6位整数，姓名的域为字符串集合，年龄的域为小于40的整数，性别的域为“男,女”。

(7) 联系 (relationship)。

在现实世界中，事物内部以及事物之间是有联系的，这些联系同样也要抽象并反映到信息世界中来。在信息世界中，事务内部以及事务之间的联系将被抽象为实体型内部的联系和实体型之间的联系。实体型内部的联系通常是指组成实体的各属性之间的联系，实体型之间的联系通常是指不同实体集之间的联系。

两个实体型之间的联系有如下3种类型。

(1) 一对一联系 (1:1)。

实体集A中的一个实体至多与实体集B中的一个实体相对应，反之亦然，则称实体集A与实体集B为一对一的联系，记作1:1，如班级与班长、观众与座位、病人与床位之间的联系。

(2) 一对多联系 (1:n)。

实体集A中的一个实体与实体集B中的多个实体相对应，反之，实体集B中的一个实体至多与实体集A中的一个实体相对应，记作1:n，如班级与学生、公司与职员之间的联系。

(3) 多对多联系 (m:n)。

实体集A中的一个实体与实体集B中的多个实体相对应，反之，实体集B中的一个实体与实体集A中的多个实体相对应，记作m:n，如教师与学生、学生与课程之间的联系。

实际上，一对一联系是一对多联系的特例，而一对多联系又是多对多联系的特例。通常可以用图形来表示两个实体型之间的这3种联系，如图1-5所示。

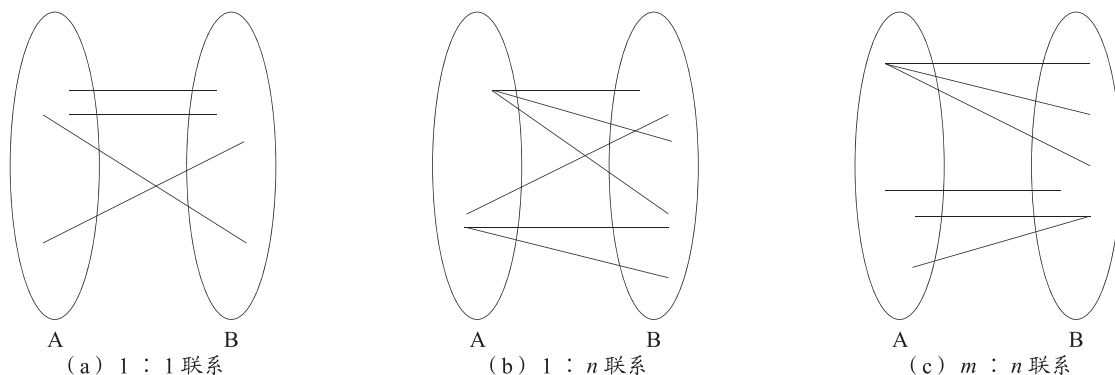


图 1-5 3 种联系示意图

2. E-R图

在E-R图中有如下4个基本成分。

- (1) 矩形框，表示实体类型（研究问题的对象），相应的命名记入框中。
- (2) 菱形框，表示联系类型（实体间的联系），相应的命名记入框中。
- (3) 椭圆形框，表示实体类型和联系类型的属性，相应的命名记入框中。
- (4) 直线，联系类型与其涉及的实体类型之间以直线连接，用来表示它们之间的联系，并在直线端部标注联系的种类（1 : 1、1 : n或m : n）。

下面以图书管理设计E-R模型为例来说明设计E-R图的过程。在图书管理系统中，读者从图书馆借书，图书馆从出版社购书，E-R图的具体建立过程如下：

- (1) 首先确定实体类型。本问题有3个实体类型：读者、书、出版社。
- (2) 确定联系类型。读者和书之间是m : n联系，起名为“借阅”；出版社和书之间是1 : n联系，起名为“订购”。
- (3) 把实体类型和联系类型组合成E-R图。
- (4) 确定实体类型和联系类型的属性。实体类型“读者”的属性有读者编号、姓名、年龄、性别、系别等，实体类型“书”的属性有书号、书名、作者、价格等，实体类型“出版社”的属性有出版社编号、出版社名、地址等，联系类型“借阅”的属性有借阅日期、归还日期等。
- (5) 确定实体类型的码，在E-R图属于码的属性名下画一条横线。具体的E-R图如图1-6所示。

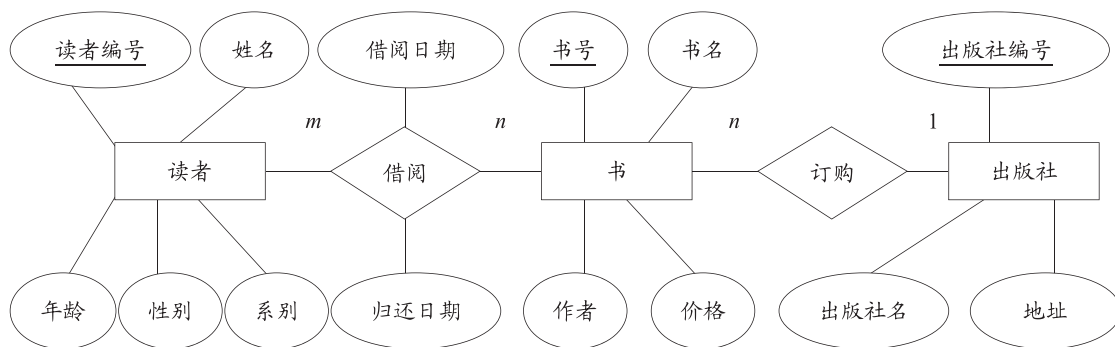


图 1-6 图书管理 E-R 图

E-R模型有两个明显的优点：一是接近于人的思维，容易理解；二是与计算机无关，用户容易接受。因此，E-R模型已成为软件工程中一个重要的设计方法。但是，E-R模型只能说明实体间语义的联系，还不能进一步说明详细的数据结构。一般遇到一个实际问题，总是先设计一个E-R模型，然后再把E-R模型转换成计算机能实现的数据模型。

1.2.2 关系模型

目前，数据库领域中最常用的逻辑数据模型有4种，分别是：

- 层次模型（hierarchical model）。
- 网状模型（network model）。
- 关系模型（relational model）。
- 面向对象模型（object-oriented model）。

其中，层次模型和网状模型统称为非关系模型。非关系模型的数据库系统在20世纪70年代至80年代初非常流行，在当时的数据库系统产品中占据主导地位，现在已逐渐被关系型数据库系统取代。

1970年美国IBM公司San Jose研究室的研究员E.F.Codd首次提出了数据库系统的关系模型，开创了数据库关系方法和关系数据理论的研究，为数据库技术奠定了理论基础。20世纪80年代以来，计算机厂商新推出的数据库管理系统几乎都支持关系模型，很多非关系系统的产品也都加上了对关系模型的接口。面向对象的方法和技术在计算机各个领域，包括程序设计语言、软件工程、信息系统设计、计算机硬件设计等各方面都产生了深远的影响，也促进了数据库面向对象数据模型的研究和发展。随着因特网的迅速发展，目前也出现了支持各种半结构化和非结构化数据的新一代数据模型，如XML数据模型等。截至目前，在数据库应用领域中，仍是以基于关系模型的数据库为主。

关系模型是目前主流数据库采用的一种数据模型。

1. 关系模型的基本术语

(1) 二维表。

在关系模型中，数据结构只有单一的二维表结构，用于表示实体与实体间的关系。也就是说，一个关系对应一张二维表，二维表名就是关系名。如图1-8中包含两个二维表，即两个关系：学生信息关系和选课信息关系。

(2) 属性及值域。

二维表中的列（字段）称为关系的属性。属性的个数称为关系的元数，又称为度。度为1的关系称为一元关系，度为 n 的关系称为 n 元关系。关系的属性包括属性名和属性值两部分，其列名即为属性名，列值即为属性值。属性值的取值范围称为值域，每一个属性对应一个值域，不同属性的值域可以相同。

例如，在图1-7中，“学生信息表”中有学号、姓名、性别、年龄4个属性，是四元关系。其中，性别属性的值域是“男”和“女”，年龄属性的值域是18~65。“选课信息表”



中有学号、课程号、成绩3个属性，是三元关系。其中，“101001”是“学号”属性的一个值，“001”是“课程号”属性的一个值。

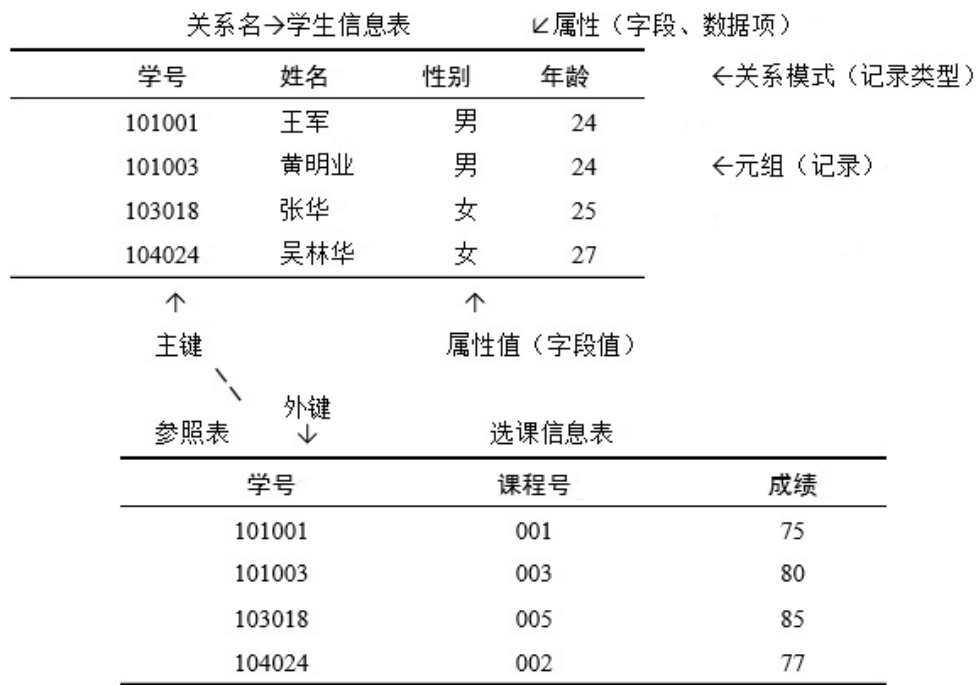


图 1-7 关系模型

(3) 关系模式。

二维表中的表头(记录类型)，即对关系的描述称为关系模式，关系模式的一般形式为：

关系名(属性1,属性2,...,属性n)

图1-7中的两个关系模式可表示为：

学生信息表(学号,姓名,性别,年龄)

选课信息表(学号,课程号,成绩)

(4) 元组。

二维表中的一行，即每一条记录的值得称为关系的一个元组。其中，每一个属性的值称为元组的分量。关系由关系模式和元组的集合组成。

图1-7中学生信息表由以下元组组成。

(101001,王军,男,24)

(101003,黄明业,男,24)

(103018,张华,女,25)

(104024,吴林华,女,27)

图1-7中选课信息表由以下元组组成。

```
(101001,001,75)
(101003,003,80)
(103018,005,85)
(104024,002,77)
```

(5) 键（码）。

由一个或多个属性组成。在实际使用中，有下列几种键：

- **候选键**（candidate key）：若关系中的某一属性组的值能唯一地标识一个元组，则称该属性组为候选键。
- **主键**（primary key）：若一个关系有多个候选键，则选定其中一个为主键。主键中包含的属性称为主属性。不包含在任何候选键中的属性称为非键（码）属性（non-key attribute）。关系模型的所有属性组是这个关系模式的候选键，称为全键（all-key）。
- **外键**（foreign key）：设F是关系R的一个或一组属性，但不是关系R的码。如果F与关系S的主码相对应，则称F是关系R的外码，关系R称为参照关系，关系S称为被参照关系或目标关系。

在图1-7所示的学生信息表中，“学号”就是主键；在选课信息表中，“(学号,课程号)”为主键，而“学号”为外键。

(6) 主属性与非主属性。

关系中包含在任何一个候选键中的属性称为主属性，不包含在任何一个候选键中的属性称为非主属性。在图1-7中的“学生信息表”中，“学号”和“姓名”都是候选键，所以“学号”和“姓名”是主属性，其他属性是非主属性。

2. 关系的性质

一般用集合的观点定义关系，关系是多个属性值域的笛卡儿积的子集。也就是说，把关系看成一个集合，集合中的元素是元组，每个元组的属性个数均相同。如果一个关系的元组个数是无限的，称为无限关系；反之，称为有限关系。

在关系模型中对关系做了一些规范性的限制，可通过二维表格形象地理解关系的性质。

(1) 关系中每个属性值都是不可分解的，即关系的每个元组分量必须是原子的。从二维表的角度讲，即不允许表中嵌套表。表1-2(a)就出现了这种表中嵌套表的情况，在“学时”下嵌套“理论”和“实验”。因为关系是从域出发定义的，每个元组分量都是不可再分的。遇到这种情况，可对表格进行简单的等价变换，使之成为符合规范的关系。例如，可将表1-2(a)改为表1-2(b)。



表 1-2 关系规范举例

课程	学时	
	理论	实验
数据库原理	54	10
编译原理	40	10

(a) 不符合规范, 非关系

课程	理论学时	实验学时
数据库原理	54	10
编译原理	40	10
操作系统	50	12

(b) 符合规范的关系

(2) 关系中不允许出现相同的元组。从语义角度看, 二维表中的一行即一个元组, 代表着一个实体。现实生活中不可能出现完全一样、无法区分的两个实体, 因此, 二维表中不允许出现相同的两行。同一关系中不能有两个相同的元组存在, 否则将使关系中的元组失去唯一性, 这一性质在关系模型中很重要。

(3) 在定义一个关系模式时, 属性的排列次序可任意交换, 因为交换属性排序的先后, 并不改变关系的实际意义。

(4) 在一个关系中, 元组的排列次序可任意交换, 并不改变关系的实际意义。判断两个关系是否相等, 是从集合的角度来考虑的, 与属性及元组的次序无关, 与关系的命名也无关。如果两个关系仅仅是上述差别, 在其余各方面完全相同, 就认为这两个关系相等。

(5) 关系模式相对稳定, 关系却随着时间的推移不断变化。这是由数据库的更新操作(包括插入、删除、修改)引起的。

3. 关系模式

关系模式是对关系的描述。关系模式是“型”, 而关系是“值”。定义关系模式必须指明:

- (1) 元组集合的结构, 包括属性构成、属性来自的域、属性与域之间的映像关系。
- (2) 元组语义以及完整性约束条件。
- (3) 属性间的数据依赖关系集合。

关系模式可以形式化地表示为 $R(U, D, \text{dom}, F)$, 其中各字符的含义如下:

- **R**: 关系名。
- **U**: 组成该关系的属性名集合, 即属性组。
- **D**: 属性组U中属性所属的域。
- **dom**: 属性对应域的映像集合。
- **F**: 属性间的数据依赖关系集合。

关系模式通常可以简记为 $R(U)$ 或 $R(A_1, A_2, \dots, A_n)$, 其中, R为关系名, $A_1, A_2, \dots, A_n$ 为属性名。

4. 关系的数据操作和完整性

关系数据模型的操作主要包括查询、插入、删除和修改数据。关系模型中的数据操作是集合操作, 操作对象和操作结果都是关系, 即若干元组的集合, 而不像非关系模型中那样是单记录的操作方式。另一方面, 关系模型把存取路径向用户隐藏起来, 用户只要指出“做什么”或

“找什么”，不必详细说明“怎么做”或“怎么找”，从而大大地提高了数据的独立性，提高了用户生产率。

关系的数据操作必须满足关系的完整性约束条件。关系的完整性约束条件有3类：实体完整性、参照完整性和用户定义的完整性。

(1) 实体完整性 (entity integrity)。

一个基本关系通常对应现实世界的一个实体集，如学生关系对应于学生的集合。现实世界中的实体是可区分的，即它们具有某种唯一性标识。相应地，关系模型中以主键作为唯一性标识。主键中的属性即主属性，不能取空值。所谓空值就是“不知道”或“无意义”的值。如果主属性取空值，就说明存在某个不可标识的实体，即存在不可区分的实体，这与现实世界的应用环境相矛盾，因此这个实体一定不是一个完整的实体。

实体完整性规则是指若属性A是基本关系R的主属性，则属性A不能取空值。

(2) 参照完整性 (referential integrity)。

现实世界中的实体之间往往存在某种联系，在关系模型中实体与实体间的联系都是用关系来描述的，这样就自然存在着关系与关系间的引用。

参照完整性规则是指若属性（或属性组）F是基本关系R的外键，它与基本关系S的主键Ks相对应（基本关系R和S不一定是不同的关系），则对于R中每个元组在F上的值必须为：

- 或者取空值（F的每个属性值均为空值）。
- 或者等于S中某个元组的主键值。

参照完整性规则就是定义外键与主键之间的引用规则。

下面以具体实例来说明参照完整性规则在关系中的实现。

在关系数据库中有两个关系模式：

- **学生关系模式：**S(学号,姓名,性别,年龄,班级号,系别), PK(学号)。
- **学习关系模式：**SC(学号,课程号,成绩), PK(学号,课程号), FK1(学号), FK2(课程号)。

根据规则要求，关系SC中的“学号”值应该在关系S中出现。如果关系SC中有一个元组“(S07,C04,80)”，而学号S07却在关系S中找不到，那么就认为在关系SC中引用了一个不存在的学生实体，这就违反了参照完整性规则。另外，在关系SC中“学号”不仅是外键，也是主键的组成部分，因此这里“学号”值不允许为空。

(3) 用户定义的完整性 (user-defined integrity)。

实体完整性和参照完整性适用于任何关系数据库系统。除此之外，不同的关系数据库系统根据其应用环境的不同，往往还需要一些特殊的约束条件。

用户定义的完整性就是针对某一具体关系数据库的约束条件，它反映某一具体应用所涉及的数据必须满足的语义要求。关系模型应提供定义和检验这类完整性的机制，以使用统一系统的方法处理它们，而不是由应用程序承担这一功能。

上述的学生关系S中，学生的年龄定义为两位整数，但范围较大，为此可以写出如下规则把年龄限制在15~30岁之间。



```
CHECK (age BETWEEN 15 AND 30)
```

5. 关系数据模型的存储结构

在关系数据模型中，实体与实体间的联系都是用二维表表示的。在数据库的物理组织中，表以文件形式存储，有的系统一张表对应一个操作系统文件，有的系统则自己设计文件结构。

6. 关系数据模型的优缺点

关系数据模型具有下列优点：

- 关系模型与非关系模型不同，它是建立在严格的数学概念的基础上的。
- 关系模型的概念单一，无论实体还是实体之间的联系都用关系表示，对数据的检索结果也是关系（即表）。因此其数据结构简单、清晰，用户易懂易用。
- 关系模型的存取路径对用户透明，因而关系模型具有更高的数据独立性、更好的安全保密性，同时也简化了程序员的工作及数据库开发建立的工作。

正是由于上述优点的存在，关系数据模型诞生以后得以快速发展。但是，关系数据模型也有缺点。其中最主要的缺点是，由于存取路径对用户透明，查询效率往往不如非关系数据模型。因此为了提高性能，必须对用户的查询请求进行优化，这就增加了开发数据库管理系统的难度。

1.3 数据库系统的结构

数据库系统的结构可以从多种不同的层次或不同的角度进行解读。从数据库管理系统角度看，数据库系统通常采用三级模式结构，这是数据库管理系统内部的系统结构；从数据库最终用户角度看，数据库系统的结构分为集中式结构（包括单用户结构和主从式结构）、分布式结构、客户机/服务器结构和并行结构，这是数据库系统外部的体系结构。

本节分别从以上两个方面介绍数据库的系统结构。

1.3.1 数据库的三级模式结构

模式（schema）是对数据库中全体数据的逻辑结构和特征的描述，它仅仅涉及型的描述，不涉及具体的值。模式的一个具体值称为模式的一个实例（instance），同一个模式可以有很多实例。模式是相对稳定的，而实例是相对变动的，这是因为数据库中的数据是在不断更新的。模式反映的是数据的结构及其联系，而实例反映的是数据库某一时刻的状态。

虽然实际的数据库系统软件的产品种类很多，它们支持不同的数据模型，使用不同的数据库语言，建立在不同的操作系统之上。但从数据库管理系统的角度看，它们的体系结构都具有相同的特征，即采用三级模式结构。

1. 数据库系统的三级模式结构

数据库系统的三级模式结构是指数据库系统是由外模式、模式和内模式三级构成的，如图1-8所示。

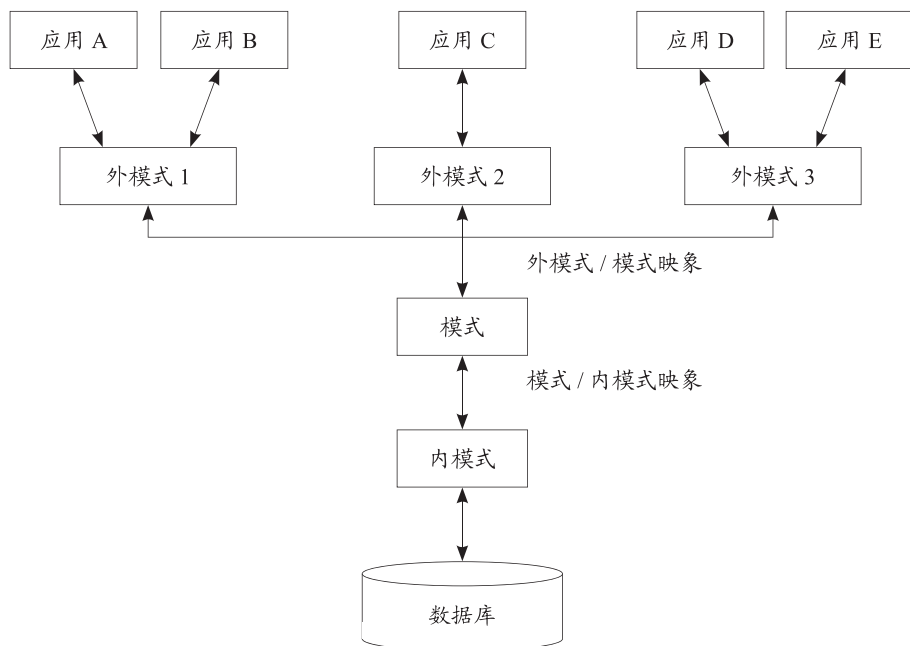


图 1-8 数据库系统的三级模式结构图

(1) 模式。

模式也称为逻辑模式，是数据库中全体数据的逻辑结构和特征的描述，是所有用户数据的公共视图。在模式的定义中，通过某个具体数据模型描述了数据库的完整逻辑结构，同时还给出了实体和属性的名称及其约束条件等完整性控制定义，是一个可以放进数据项值的框架。

(2) 外模式。

外模式也称子模式或用户模式，从逻辑上看，外模式是模式的一个逻辑子集，一个模式可以推导出多个不同的外模式。外模式是数据库用户看见和使用的对局部数据的逻辑结构和特征的描述，通常是与某个具体应用相关的数据子集的逻辑表示。多个应用（用户）可以共享一个外模式，一个外模式可以为多个应用（用户）使用。设立外模式的好处有以下几点：

- 方便了用户的使用，简化了用户的接口。用户只需依照外模式编写应用程序或在终端输入命令，而无须了解数据的逻辑结构和存储结构。
- 保证数据的独立性。由于在三级模式之间存在两级映象，使得物理模式和逻辑模式的变化都反映不到外模式一层，从而不用修改应用程序，提高了数据的独立性。
- 有利于数据共享。从同一模式产生不同的外模式，减少了数据的冗余度，有利于为多种应用服务。
- 用户程序只能操作对应的外模式范围内的数据，所以会与数据库中的其他数据相互隔离，这样便缩小了程序错误传播的范围，保证了其他数据的安全。

(3) 内模式。

描述数据的物理存储的模式叫内模式，也称物理模式。它是对数据的物理结构和存储结构的描述，是数据库的内部表示方式。它规定数据项、记录、数据集、索引和存取路径在内的一切物理组织方式，以及优化性能、响应时间和存储空间的需求。它还规定记录的位置、块的大小与溢出区等。一个数据库只有一个内模式。

无论哪一级模式都只能处理数据的一个框架，按这些框架组织和存储的数据才是数据库的内容。但要注意的是框架和数据是两回事，它们放在不同的地方。所以，模型、模式、具体值是3个不同的概念。

2. 数据库系统的二级映象功能和数据独立性

数据库系统的三级模式对应数据的3个抽象级别，它把数据的具体组织交由DBMS管理，使用户能逻辑、抽象地处理数据，而不必关心数据在计算机中的具体表示与存储方式。

对于每一个外模式，数据库系统都有一个外模式/模式映象，它定义了该数据库外模式与模式之间的对应关系。这些映象定义通常包含在各自外模式的描述中。当模式改变时（如增加新的数据类型、新的数据项、新的关系等），由数据库管理员对各个外模式/模式映象做相应改变，可以使外模式不变，从而不必修改应用程序，保证了数据的逻辑独立性。

数据库只有一个模式，也只有一个内模式，所以模式/内模式映象是唯一的，它定义了数据的全局逻辑结构与存储结构之间的对应关系，即说明逻辑记录在内部是如何表示的。该映象定义通常包含在模式描述中。当数据库的存储结构改变时（如采用了更先进的存储结构等），由数据库管理员对模式/内模式映象做相应的改变，可以使模式保持不变，从而保证了数据的物理独立性。

■ 1.3.2 数据库的体系结构

从数据库管理系统的角度看，数据库系统是一个三级模式结构，但数据库的这种模式结构对最终用户和程序员是透明的，他们见到的仅是数据库的外模式和应用程序。从最终用户角度来看，数据库系统分为单用户结构、主从式结构、分布式结构和客户机/服务器结构。

1. 单用户结构的数据库系统

单用户数据库系统是最早期的、最简单的数据库系统，如图1-9所示。在单用户系统中，整个数据库系统（包括应用程序、DBMS、数据等）都装在一台计算机上，由一个用户独占，不同的机器间不能共享数据。

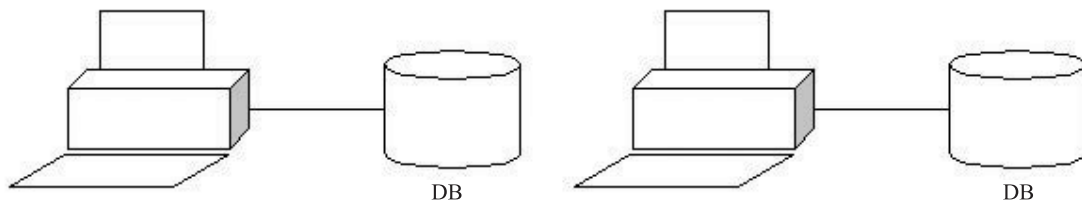


图 1-9 单用户结构的数据库系统

例如，一个企业的各个部门都使用本部门的机器来管理本部门的数据，各个部门的机器是独立的。因为不同部门之间不能共享数据，所以企业内部存在大量的冗余数据。

2. 主从式结构的数据库系统

主从式结构是指一个主机带多个终端的多用户结构。在这种结构中，数据库系统（包括应用程序、DBMS、数据等）集中存放在主机上，所有任务都由主机完成，各个用户通过主机的终端并发地存取数据库，共享数据资源，如图1-10所示。

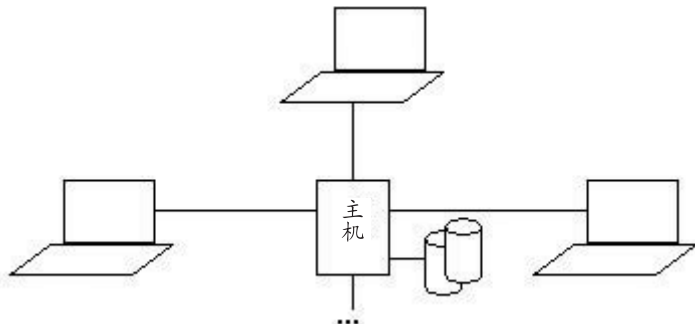


图 1-10 主从式结构的数据库系统

主从式结构的优点是结构简单，数据易于维护和管理。缺点是当终端用户增加到一定程度后，主机的任务过于繁重，成为瓶颈，从而使系统性能大幅度下降。另外，当主机出现故障，可能导致整个系统不能使用，因而系统的可靠性不高。

3. 分布式结构的数据库系统

分布式结构的数据库系统是指数据库中的数据在逻辑上是一个整体，但物理分布在计算机网络的不同节点上，如图1-11所示。网络的每一个节点都可以独立处理本地数据库中的数据，执行局部应用；同时也可以存取和处理多个异地数据库中的数据，执行全局应用。

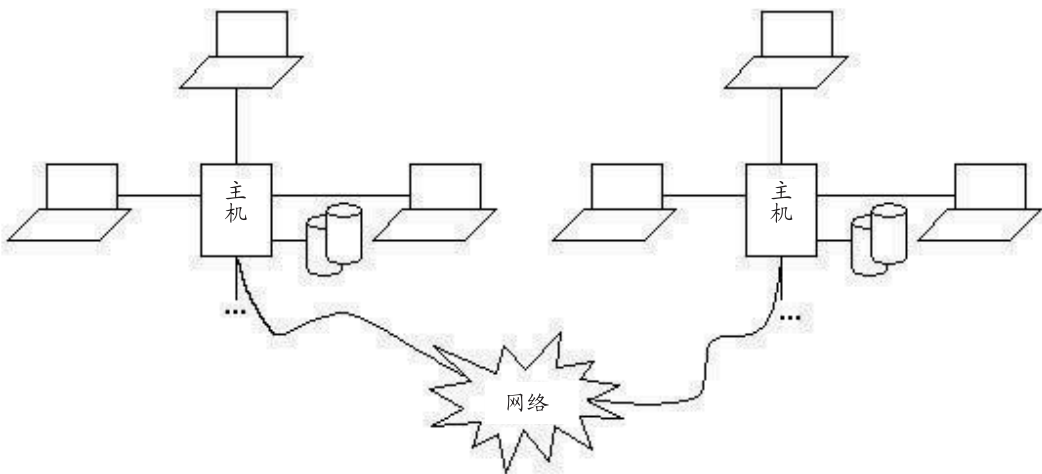


图 1-11 分布式结构的数据库系统

分布式结构的数据库系统是计算机网络发展的必然产物，它适应了地理上分散的公司、团体和组织对于数据库应用的需求，但数据的分布存放会给数据的管理、维护带来困难。此外，当用户需要经常访问远程数据时，系统效率将受网络环境的制约。

4. 客户机/服务器结构的数据库系统

主从式数据库系统中的主机和分布式数据库系统中的每个节点机一样，通常是一个通用计算机，既执行DBMS功能，又执行应用程序。随着工作站功能的增强和广泛使用，人们开始把DBMS功能和应用分开，即网络中某些节点上的计算机专门用于执行DBMS功能，称为数据库服务器（简称服务器），其他节点上的计算机则安装DBMS外围的应用开发工具，支持用户的应用，称为客户机，这就是客户机/服务器结构的数据库系统。

在客户机/服务器结构中，客户端的用户请求被传送到数据库服务器，数据库服务器进行处理后只将结果返回给用户（而不是整个数据），从而显著减少了网络数据的传输量，提高了系统的性能、吞吐量和负载能力。

另外，客户机/服务器结构的数据库往往更加开放，它们一般都能在多种不同的硬件和软件平台上运行，可以使用不同厂商的数据库应用开发工具，应用程序具有更强的可移植性，同时减少了软件维护的开销。

客户机/服务器数据库系统可以分为集中式服务器结构（图1-12）和分布式服务器结构（图1-13）。前者在网络中仅有一台数据库服务器，而有多台客户机。后者在网络中有多台数据库服务器。分布式服务器结构是客户机/服务器结构与分布式结构的结合。

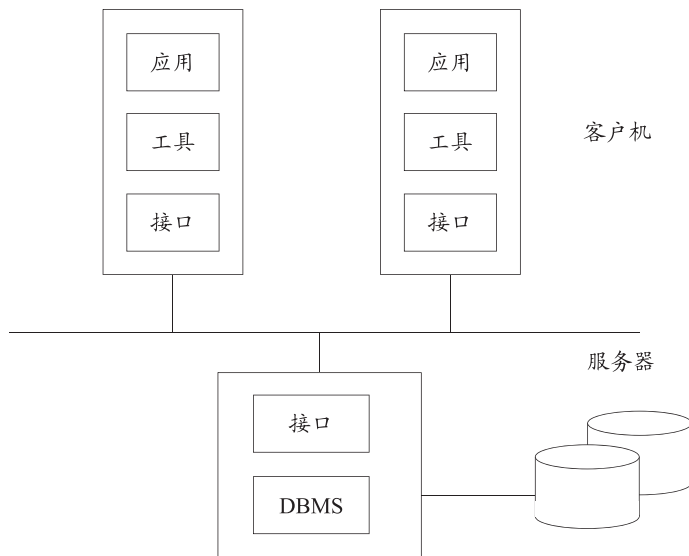


图 1-12 集中式服务器结构

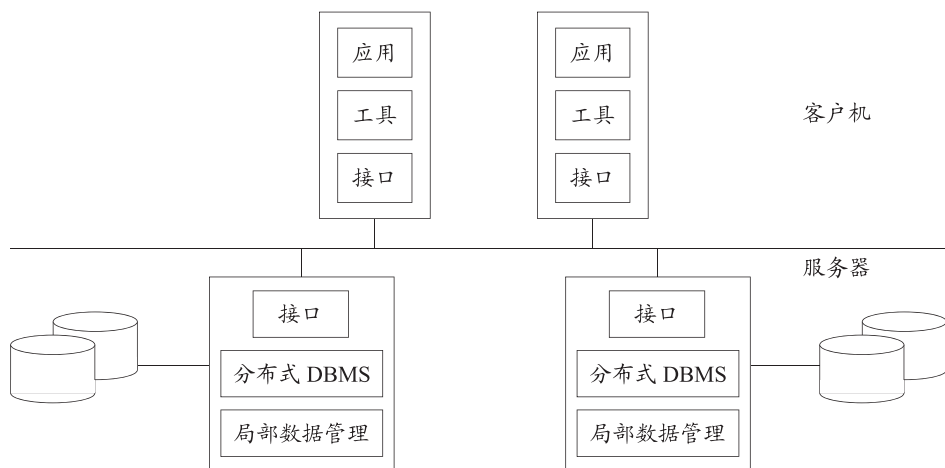


图 1-13 分布式服务器结构

与主从式结构相似，在集中式服务器结构中，一个数据库服务器要为众多的客户机服务，往往容易构成瓶颈，制约系统的性能。与分布式结构相似，在分布式服务器结构中，数据分布在不同的服务器上，会给数据的处理、管理和维护带来困难。

■ 1.3.3 数据库的连接

1. 关系数据库标准语言SQL

不同的应用程序可以选择不同的DBMS，而对数据库的具体操作都是通过DBMS来实现的，应用程序不可能跳过DBMS直接访问数据本身，所以在应用程序与DBMS之间就需要有一个统一的接口，这就是SQL语言。

结构化查询语言（structure query language，SQL）是关系型数据库管理系统中最流行的数据查询和更新语言，用户可以使用SQL语言对数据库执行各种操作，包括数据定义、数据操纵和数据控制等与数据库有关的全部功能。

SQL语言具有功能丰富、简单易学、使用方式灵活等突出优点，备受计算机工业界和计算机用户的欢迎。尤其自SQL成为国际标准后，各数据库管理系统厂商纷纷推出各自的支持SQL的软件或与SQL接口的软件，这就使得大多数数据库均采用了SQL作为共同的数据存取语言 and 标准接口。

但是，不同的数据库管理系统厂商开发的SQL并不完全相同。这些不同的SQL一方面遵循了标准SQL语言规定的基本操作，另一方面又在标准SQL语言的基础上进行了扩展，增强了一些功能。不同厂商的SQL往往有不同的名称，例如，Oracle产品中的SQL称为PL/SQL（procedural language/SQL，过程化SQL语言），Microsoft SQL Server产品中的SQL称为Transact-SQL。

关于SQL语言的更多介绍将在第3章中给出，这里不展开说明。



2. DBMS的接口

有了统一的SQL语言，就可以在应用程序中通过SQL语言来表达数据访问的需求。由应用程序将SQL语句传递给DBMS，DBMS收到SQL语句后执行相应的操作，再把结果返回给应用程序，完成一次对数据库的访问。负责完成相应传递功能的程序就是DBMS的接口。

在整个数据库系统进行安装部署的时候，DBMS的接口应根据系统要求安装到各个服务器或是客户机上，并进行相应的配置，以保障应用程序能够连接到正确的数据库。一般来说，大型数据库均提供专门的接口供应用程序连接使用，而对于小型数据库，应用程序通常会使用数据库中间件进行连接。Windows系统提供的ODBC就是一款通用的数据库中间件，利用ODBC可以连接文本文件、Excel文件，以及Access数据库等各种不同类型的数据文件，当然，经过配置也可以连接SQL Server或Oracle等大型数据库的数据源。

1.4 数据库的规范化

关系数据库设计的一个重要结果是生成一组关系模式，而一个不规范的关系模式可能会带来数据冗余、插入异常、修改异常和删除异常。因此，需要有一套规范化的理论以判断所设计的关系模式达到了哪种程度的规范化，这是数据库逻辑设计的一个有力工具。

1.4.1 数据依赖

数据依赖是指通过一个关系中属性之间值的相等与否来体现数据间的相互关系，是对现实世界属性之间相互联系的抽象，是数据内在的性质，是语义的体现。

关系模式是用来定义关系的，一个关系数据库包含一组关系，定义这组关系的关系模式集合为 U ，属性间数据的依赖关系集合为 F ，因此，关系模式 R 定义为一个三元组 $R(U, F)$ 。当且仅当 U 上的一个关系 r 满足 F 时， r 称为关系模式 $R(U, F)$ 的一个关系。

由于关系模式经常出现数据冗余量大、数据的插入和删除异常的问题，导致此关系模式不是一个最优关系模式，这主要是因为模式中的某些数据依赖引起的。规范化理论正是用于改造关系模式的，它通过分解关系模式来消除其中不合适或不准确的数据依赖，以解决上述的问题。

1.4.2 相关概念

1. 函数依赖

假设 $R(U)$ 是一个关系模式， U 是 R 的属性集合， X 和 Y 是 U 的子集。对于 $R(U)$ 上的任意一个可能的关系 r ，如果 r 中不存在两个元组，它们在 X 上的属性值相同，而在 Y 上的属性值不同，则称“ X 函数确定 Y ”或“ Y 函数依赖于 X ”，记为 $X \rightarrow Y$ 。

2. 平凡函数依赖和非平凡函数依赖

在关系模式 $R(U)$ 中，对于 U 的子集 X 和 Y ，如果 $X \rightarrow Y$ 且 Y 不是 X 的子集，则 $X \rightarrow Y$ 称为非平凡

函数依赖：若 Y 是 X 的子集，则称其为平凡函数依赖。

3. 完全依赖与部分依赖

在关系模式 $R(U)$ 中，如果 $X \rightarrow Y$ ，并且对 X 的任何一个真子集 X' ，不存在 $X' \rightarrow Y$ ，则称 Y 完全依赖于 X ，否则，可以说 Y 不完全依赖于 X ，或称 Y 部分依赖于 X 。

4. 传递函数依赖

在关系模式 $R(U)$ 中，如果 $X \rightarrow Y$ ， $Y \rightarrow Z$ ，且 Y 不是 X 的子集，也不存在 $Y \rightarrow X$ ，则称 Z 传递依赖于 X 。

5. 键

设 K 为关系模式 $R(U, F)$ 中的属性或属性组合，若 U 完全依赖于 K ，则称 K 为 R 的一个候选键；若关系模式 R 有多个候选键，则选定其中一个作为主键。候选键能够唯一标识关系，是关系模式中的一组最重要的属性。另外，主键和外键一起提供了表示关系之间的联系的手段。

1.4.3 范式

使用规范化理论研究关系模式中各属性之间的依赖关系以及对关系模式性能的影响，可以探讨关系模式应该具备的性质与设计方法。关系必须是规范化的关系，应该满足一定的约束条件。通常把关系的规范化形式称为范式（normal form, NF）。范式表示的是关系模式的规范化程度，即满足某种约束条件的关系模式，根据满足的约束条件的不同可确定6种范式。在6种范式中，本节主要介绍前3种范式。一般来说，在进行数据库设计时，规范化到第三范式已经足够。

1. 第一范式（1NF）

如果一个关系模式 R 的所有属性都是不可分的基本数据项，则称 R 为1NF。

任何一个关系数据库中，1NF是对关系模式最起码的要求，不满足1NF的数据库模式不能称为关系数据库。

由表1-3可以看出，“薪金”是可以分割的数据项，因此不符合1NF的标准，必须对其进行规范化处理，规范化处理后如表1-4所示。但是满足了1NF不一定就是一个好的关系模式。

表 1-3 不符合 1NF 的关系

工作证号	员工姓名	薪金	
		基本工资	奖金
2006001	张天	800	3000
2006002	王耀	1000	4000
2006003	孙东平	1200	5000



表 1-4 符合 1NF 的关系

工作证号	员工姓名	基本工资	奖金
2006001	张天	800	3000
2006002	王耀	1000	4000
2006003	孙东平	1200	5000

2. 第二范式 (2NF)

若关系模式R为1NF，并且每一个非主属性都完全依赖于R的键，则称R为2NF。关系R不仅满足1NF，且R中只存在一个主键，所有非主属性都应该完全依赖于该主键。

2NF不允许关系模式的属性之间有函数依赖 $X \rightarrow Y$ ，其中， X 是键的真子集， Y 是非主属性。

在表1-5中，关系满足1NF，但不满足2NF。

表 1-5 不符合 2NF 的关系

工作证号	员工姓名	项目代号	所在城市
2006001	张天	07001	北京
2006002	王耀	06002	郑州
2006003	孙东平	07001	北京

在表1-5中，主键由“工作证号”和“项目代号”组成，而“员工姓名”依赖于“工作证号”且“所在城市”依赖于“项目代号”，这样会造成数据冗余和更新异常。增加新的项目数据时，没有对应的员工信息；删除员工信息时，有可能同时将项目信息删除。解决的方法是将一个这样的非2NF分解成多个2NF的关系模式，方法如下：

员工关系：工作证号、员工姓名

项目关系：项目代号、所在城市

员工与项目关系：工作证号、项目代号

3. 第三范式 (3NF)

如果关系模式R为2NF， X 是R的候选键， Y 、 Z 是R的非主属性组，如果不存在 $Y \rightarrow Z$ ，亦即不存在属性是通过其他属性（组）传递依赖于键，则称R为3NF。

表1-6给出的关系满足2NF，但不满足3NF。在表1-6中，“项目名称”和“所在城市”依赖于“项目代号”，“邮政编码”也依赖于“项目代号”，但是这个依赖是由于“邮政编码”依赖于“所在城市”，而后者又依赖于“项目代号”，导致“邮政编码”传递依赖于“项目代号”。例如，如果北京的项目很多，那么北京的邮政编码“100000”就会出现大量重复；而另一方面，如果某个城市没有项目，就会造成所在城市和邮政编码对应信息的缺失。也就是说，仍然存在数据冗余和更新异常。解决传递依赖的方法仍然是对关系进行分解，将它分成两个3NF。

表 1-6 不符合 3NF 的关系

项目代号	项目名称	所在城市	邮政编码
07001	调研项目	北京	100000
06002	开发项目	郑州	450000
07002	管理项目	北京	100000

项目关系：项目代号、项目名称、所在城市

城市关系：所在城市、邮政编码

4. 关系模式规范化的步骤

规范化的基本思想是逐步消除数据依赖不合理的部分，使模式中的各关系模式达到某种程度上的分离，尽量减少数据冗余和更新异常的出现，即让一个关系或描述一种实体，或描述实体属性之间的联系，是概念的单一化。关系模式规范化的基本步骤如图1-14所示（仅列出1NF到3NF的步骤）。

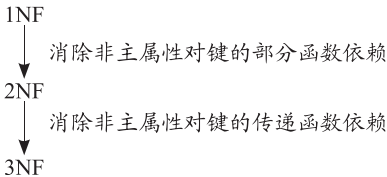


图 1-14 关系模式规范化的步骤

知识点拨

关系数据库中关系模式规划化的6种范式：第一范式（1NF）、第二范式（2NF）、第三范式（3NF）、巴斯-科德范式（BCNF）、第四范式（4NF）和第五范式（5NF，又称完美范式）。范式的等级越高，应满足的约束条件就越严格。范式的每一级别都依赖于它的前一级别，例如，若一个关系模式满足2NF，则一定满足1NF。通常，数据库中的关系模式满足第三范式即可。但是，具体划分到哪种范式，往往要看具体情况。范式越高，数据冗余程度越低，但处理速度相对较慢。所以理论上是第三范式即可，而在实际设计开发过程中，很可能为了速度直接采用第一范式，这需要在数据库设计时根据实际情况进行权衡。

1.5 数据库设计

有人说，一个成功的管理信息系统，是由“50%的业务+50%的软件”所组成，而50%的成功软件又由“25%的数据库+25%的程序”所组成。因此，数据库设计是软件开发的一个关键步骤。数据库设计是指在给定的环境下，创建一个性能良好的、能满足不同用户的使用要求、又能被选定的DBMS所接受的数据模式。

大体上可以把数据库设计划分为6个阶段：需求分析阶段、概念结构设计阶段、逻辑结构设计阶段、物理结构设计阶段、数据库实施阶段、数据库运行和维护阶段，如图1-15所示。

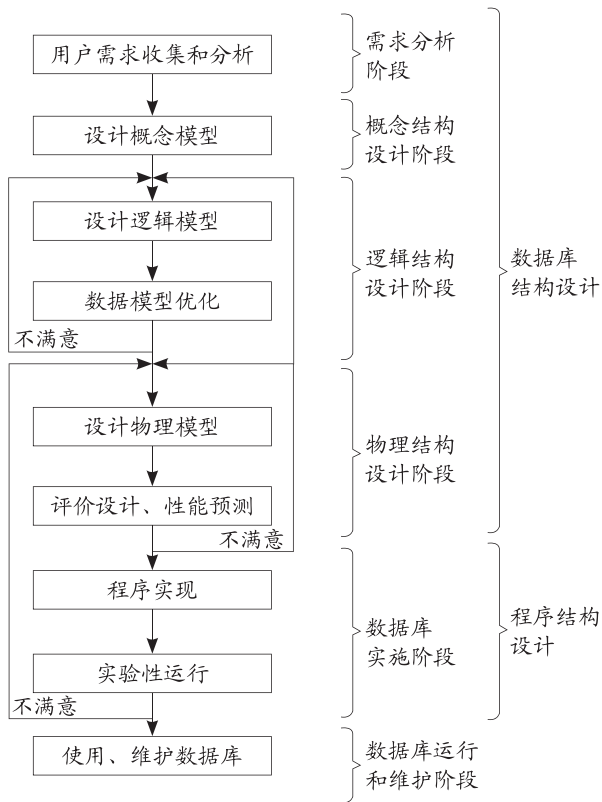


图 1-15 数据库设计阶段划分

1.5.1 需求分析

准确地搞清楚用户要求，是数据库设计的关键。需求分析的好坏，决定了数据库设计的成败。确定用户的最终需求其实是一件很困难的事。一方面用户缺少计算机知识，开始时无法确定计算机究竟能为自己做什么，不能做什么，所以无法准确地表达自己的需求，而且提出的需求往往还不断地变化。另一方面设计人员缺少用户的专业知识，不易理解用户的真正需求，甚至误解用户的需求。此外，新的硬件、软件技术的出现也会使用户需求发生变化。因此，设计人员必须与用户不断深入地进行交流，才能逐步确定用户的实际需求。

需求分析阶段的成果是系统需求规格说明书，主要包括数据流图（data flow diagram，DFD）、数据字典（data dictionary，DD）、各种说明性文档、统计输出表、系统功能结构图等。系统需求规格说明书是后期设计、开发、测试和验收等过程的重要依据。

需求分析的任务是通过详细调查现实世界要处理的对象（如组织、部门、企业等），充分了解原系统（手工系统或计算机系统）的工作概况，明确用户的各种需求，然后在此基础上确定新系统的功能。新系统必须充分考虑今后可能的扩充和改变，不能仅仅按当前应用的需求来设计数据库。需求分析的重点是调查、收集与分析用户在数据管理中的信息要求、处理要求、安全性与完整性要求等。

需求分析阶段的主要任务有以下几方面：

(1) 确认系统的设计范围, 调查信息需求, 收集数据。分析需求调查得到的资料, 明确计算机应当处理和能够处理的范围, 确定新系统应具备的功能。

(2) 综合各种信息包含的数据, 各种数据之间的关系, 以及数据的类型、取值范围和流向。

(3) 建立需求说明文档、数据字典、数据流图。将需求调查文档化, 文档既要为用户所理解, 又要方便数据库的概念结构设计。需求分析的结果应及时与用户进行交流, 反复修改, 直到得到用户的认可。在数据库设计中, 数据需求分析是对有关信息系统现有数据及数据间联系的收集和处理, 当然也要适当考虑系统在将来的需求。一般地, 需求分析包括数据流分析及功能分析。功能分析是指系统如何得到事务活动所需要的数据, 在事务处理中如何使用这些数据进行处理, 以及处理后数据流向的全过程的分析, 一般用数据流图来表示。

在需求分析阶段, 应当用文档形式整理出整个系统所涉及的数据、数据间的依赖关系、事务处理的说明和需要产生的报告, 并且尽量借助于数据字典加以说明。除使用数据流图、数据字典以外, 需求分析还可使用判定表、判定树等工具。

■ 1.5.2 概念结构设计

概念结构设计是数据库设计的第二阶段, 其目标是对需求说明书提供的所有数据和处理要求进行抽象与综合处理, 按一定的方法构造反映用户环境的数据及其相互联系的概念模型, 即用户数据模型或企业数据模型。这种概念数据模型与DBMS无关, 是面向现实世界的的数据模型, 极易为用户所理解。

为了保证所设计的概念数据模型能够完全、正确地反映用户的数据及其相互关系, 便于进行所要求的各种处理, 在本阶段中可吸收用户参与和评议设计。在进行概念结构设计时, 可设计各个应用的视图 (view), 即各个应用所看到的数据及其结构, 然后再进行视图集成 (view integration), 以形成用户的概念数据模型。这样形成的初步数据模型还要经过数据库设计者和用户的审查和修改, 最后形成所需的概念数据模型。

■ 1.5.3 逻辑结构设计

逻辑结构设计阶段的设计目标是把上一阶段得到的与DBMS无关的概念数据模型转换成等价的、为某个特定的DBMS所接受的逻辑模型所表示的概念模式, 同时将概念结构设计阶段得到的应用视图转换成外模式, 即特定DBMS下的应用视图。在转换过程中要进一步落实需求说明, 并满足DBMS的各种限制。逻辑结构设计阶段的结果是用DBMS提供的数据库定义语言写成的数据模式。逻辑结构设计的具体方法与DBMS的逻辑数据模型有关。

逻辑结构设计即是在概念结构设计的基础上进行数据模型设计, 可以采用层次模型、网状模型和关系模型。由于当前绝大多数DBMS都是基于关系模型的, E-R方法又是概念结构设计的主要方法, 如何在全局E-R图基础上进行关系模型的逻辑结构设计成为这一阶段的主要内容。在进行逻辑结构设计时并不考虑数据在某一DBMS下的具体物理实现, 即不考虑数据是如何在计算机中存储的。



■ 1.5.4 物理结构设计

将一个给定的逻辑结构实施到具体的环境中时，逻辑数据模型要选取一个具体的工作环境，这个工作环境提供了数据存储结构与存取方法，这个过程就是数据库的物理结构设计。物理结构设计阶段的任务是把逻辑结构设计阶段得到的逻辑数据库在物理上加以实现，其主要内容是根据DBMS提供的各种手段，设计数据的存储形式和存取路径，如文件结构、索引的设计等，即设计数据库的内模式。数据库的内模式对数据库的性能影响很大，应根据处理需求及DBMS、操作系统和硬件的性能进行精心设计。

数据库的物理设计通常分为两步：第一，确定数据库的物理结构；第二，评价实施的空间效率和时间效率。确定数据库的物理结构包含确定数据的存储结构、设计数据的存取路径、确定数据的存放位置以及确定系统配置等方面的内容。数据库物理设计过程中需要对时间效率、空间效率、维护代价和各种用户要求进行权衡，选择一个优化方案作为最终的数据库物理结构。

■ 1.5.5 数据库的实施

数据库实施主要包括以下工作：

- 用数据描述语言（data description language，DDL）定义数据库结构。
- 组织数据入库。
- 编制与调试应用程序。
- 数据库试运行。

1. 定义数据库结构

确定了数据库的逻辑结构与物理结构后，就可以用所选用的DBMS提供的数据库描述语言来严格描述数据库结构。

2. 数据装载

数据库结构建立好后，就可以向数据库中装载数据（即组织数据入库）了。组织数据入库是数据库实施阶段最主要的工作。对于数据量不是很大的小型系统，可以用人工方式完成数据的入库，其步骤为：

- （1）筛选数据。需要装入数据库中的数据通常都分散在各个部门的数据文件或原始凭证中，所以首先必须把需要入库的数据筛选出来。
- （2）转换数据格式。筛选出的需要入库的数据，其格式往往不符合数据库要求，还需要进行转换。这种转换有时会很复杂。
- （3）输入数据。将转换好的数据输入计算机的数据库中。
- （4）校验数据。检查输入的数据是否有误。

对于中、大型系统，由于数据量很大，用人工方式组织数据入库将会耗费大量的人力物力，而且很难保证数据的正确性。因此还应该设计一个数据输入子系统，以辅助数据的入库工作。

3. 编制与调试应用程序

数据库应用程序的设计应该与数据设计并行进行。在数据库实施阶段，当数据库结构建立好后，就可以开始编制与调试数据库的应用程序，也就是说，编制与调试应用程序是与组织数据入库同步进行的。调试应用程序时可能数据入库尚未完成，可先使用模拟数据。

4. 数据库试运行

应用程序调试完成，并且已有一小部分数据入库后，就可以开始数据库的试运行。数据库试运行也称为联合调试，其主要工作包括：

- (1) 功能测试。即实际运行应用程序，执行对数据库的各种操作，测试应用程序的各种功能。
- (2) 性能测试。即测量系统的性能指标，分析是否符合设计目标。

1.5.6 数据库的运行和维护

数据库试运行结果符合设计目标后，数据库就可以真正投入运行了。数据库投入运行标志着开发任务的基本完成和维护工作的开始，但并不意味着设计过程的终结。由于应用环境在不断变化，数据库运行过程中物理存储也会不断变化，对数据库设计进行评价、调整、修改等维护工作是一个长期的任务，也是设计工作的继续和提高。

在数据库运行阶段，对数据库经常性的维护工作主要是由DBA完成的，其维护工作包括：故障维护，数据库的安全性、完整性控制，数据库性能的监督、分析和改进，数据库的重组和重构等。

拓展阅读

优化数据中心建设布局。在区域数据中心集群间，以及集群和主要城市间建立数据中心直连网络，促进数据中心分级分类布局建设，加快实现集约化、规模化、绿色化发展。

——《“十四五”国家信息化规划》





课后作业



1. 选择题

- (1) 若关系中的某一属性组的值能唯一地标识一个元组，则称之为（ ）。
A. 主键 B. 候选键 C. 外键 D. 联系
- (2) 以下不属于数据模型三要素的是（ ）。
A. 数据结构 B. 数据操纵 C. 数据控制 D. 完整性约束
- (3) 以下对关系性质的描述中，哪个是错误的？（ ）
A. 关系中每个属性值都是不可分解的
B. 关系中允许出现相同的元组
C. 定义关系模式时可随意指定属性的排列次序
D. 关系中元组的排列次序可任意交换

2. 填空题

- (1) 数据管理发展的3个阶段是_____、_____和_____。
- (2) 数据库系统的三级模式包括_____、_____和_____。

3. 思考题

- (1) 用数据库系统管理数据相比用文件系统管理数据的优点有哪些？
- (2) 关系模式规范化的步骤有哪些？
- (3) 数据库设计的阶段包括哪些？