



责任编辑：付寒冰

封面设计：黄燕美

中文版 SQL Server 项目教程

ZHONGWENBAN SQL Server
XIANGMU JIAOCHENG

中文版 SQL Server 项目教程

主编 胡 钢 邬俊阳

中文版 SQL Server 项目教程

主编 胡 钢 邬俊阳

ZHONGWENBAN SQL Server
XIANGMU JIAOCHENG

北京希望电子出版社



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8812，获取更多资源

微信公众号



扫码下载资料包



定价：58.00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

中文版 SQL Server 项目教程

主 编 胡 钢 邬俊阳

副主编 向文欣 俞天均



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

内容简介

本书采用图文结合的方式,通过大量案例由浅入深、循序渐进地介绍了数据库原理、SQL Server 数据库的基础知识、核心技术、应用方法等,能够帮助初学者快速上手、迅速提高并掌握数据库技术。全书共分 12 个项目,包括数据库基础、SQL Server 的安装、数据库与数据表、数据查询、视图、索引、SQL 语言编程基础、存储过程、触发器、SQL 数据库维护、数据库的安全机制、企业进销存管理系统综合案例等。

本书适合作为 SQL Server 数据库原理与应用课程的教材,也可作为数据库管理人员、数据处理与分析人员、程序开发人员、数据库开发人员以及对数据库知识感兴趣的读者的学习参考书。

图书在版编目(CIP)数据

中文版 SQL Server 项目教程 / 胡钢, 邬俊阳主编. -- 北京: 北京希望电子出版社, 2024.5

ISBN 978-7-83002-866-4

I. ①中… II. ①胡… ②邬… III. ①SQL 语言—关系数据库系统—教材 IV. ①TP311.132.3

中国国家版本馆 CIP 数据核字(2024)第 110298 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: 18
字数: 427 千字
印刷: 三河市骏杰印刷有限公司
版次: 2024 年 6 月 1 版 2 次印刷

定价: 58.00 元



PREFACE

数据库的应用已经涉及我们生活的方方面面，如云计算、人工智能等。数据库的管理和使用，需要专业的人员和管理系统软件，因此，对数据库专业人才的需求也越来越大。Microsoft SQL Server是微软推出的一种关系型数据库管理系统，它具有良好的数据库设计、管理与网络功能，又与Windows、Linux、Docker及Azure云紧密集成，因而成为数据库开发的首选软件。对于新手来说，从SQL Server入手，快速掌握一门数据库的使用技巧非常重要，这也是编程人员、数据库开发人员 and 数据库管理人员所必需具备的技能。

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、产教融合、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SQL Server数据库原理、基础知识、核心技术及应用方法进行了详细介绍，能够帮助初学者快速上手，迅速掌握并提高数据库技术。本书根据不同的理论和知识点，配备了大量的案例，让读者能在实验环境中快速掌握知识点，达到融会贯通，举一反三的目的。文中的案例、操作均附有相对应的图片及代码展示，版面清晰，阅读方便。本书内容与实际应用紧密连接，读者可以快速将学到的知识应用到工作中。

全书共设12个项目，建议总课时为72课时，具体安排如下。

序 号	内 容	理论教学	上机实训
项目 1	数据库基础	2 课时	2 课时
项目 2	SQL Server 的安装	2 课时	2 课时
项目 3	数据库与数据表	4 课时	4 课时
项目 4	数据查询	4 课时	4 课时
项目 5	视图	3 课时	3 课时
项目 6	索引	4 课时	4 课时
项目 7	SQL 语言编程基础	4 课时	4 课时
项目 8	存储过程	2 课时	2 课时
项目 9	触发器	2 课时	2 课时
项目 10	SQL 数据库维护	4 课时	4 课时
项目 11	数据库的安全机制	2 课时	2 课时
项目 12	企业进销存管理系统综合案例	—	6 课时

本书由四川信息职业技术学院胡钢和四川工业科技学院邬俊阳担任主编，四川信息职业技术学院向文欣和俞天均担任副主编。编写分工如下：项目1至项目3由胡钢编写，项目4至项目6由邬俊阳编写，项目7至项目9由向文欣编写，项目10至项目12由俞天均编写。他们在长期工作中积累了大量经验，在写作过程中始终坚持严谨、细致的态度，力求精益求精。

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
2024年5月

目录

CONTENTS

项目1 数据库基础



1.1 数据库系统概述2	1.4 关系数据库的设计19
1.1.1 数据管理技术的起源与发展2	1.4.1 需求分析20
1.1.2 数据库系统的组成6	1.4.2 概念结构设计21
1.1.3 数据库管理系统8	1.4.3 逻辑结构设计21
1.2 数据库的体系结构9	1.4.4 物理结构设计21
1.2.1 数据库系统的内部结构9	1.4.5 数据库实施21
1.2.2 数据库系统的外部结构11	1.4.6 数据库运行和维护22
1.3 数据模型13	1.5 常见的关系型数据库管理系统22
1.3.1 数据模型简介13	
1.3.2 E-R模型14	课后作业24
1.3.3 关系模型16	

项目2 SQL Server的安装



2.1 SQL Server 2019简介26	2.3.5 管理网络协议45
2.1.1 SQL Server 2019的新特性26	2.4 服务器的注册46
2.1.2 SQL Server 2019的版本27	2.4.1 注册服务器46
2.2 SQL Server 2019的安装与卸载28	2.4.2 删除服务器47
2.2.1 SQL Server 2019的安装要求28	2.5 SQL Server Management Studio
2.2.2 SQL Server 2019的安装28	的管理和操作47
2.2.3 安装SQL Server Management Studio33	2.5.1 启动SQL Server Management Studio47
2.2.4 SQL Server 2019的卸载35	2.5.2 使用对象资源管理器49
2.3 SQL Server 2019的服务和管理工具35	2.5.3 在Transact SQL编辑器中编写和执行查询语句49
2.3.1 SQL Server 2019的服务组件35	2.5.4 使用模板资源管理器降低编码难度49
2.3.2 SQL Server的管理工具37	
2.3.3 SQL Server服务的启动39	课后作业50
2.3.4 SQL Server服务的管理39	

项目3 数据库与数据表



3.1 SQL Server 数据库简介	52
3.2 创建数据库	53
3.2.1 数据库文件简介	53
3.2.2 使用SSMS创建数据库	55
3.2.3 使用SQL命令创建数据库	56
3.3 修改数据库	59
3.3.1 查看数据库的属性	59
3.3.2 修改数据库	61
3.3.3 收缩数据库	64
3.4 删除数据库	65

3.4.1 使用SSMS删除数据库	65
3.4.2 使用命令删除数据库	66
3.5 操作数据表	66
3.5.1 数据表与数据类型	66
3.5.2 创建数据表	71
3.5.3 管理数据表	74
3.5.4 管理约束	77
3.5.5 管理数据表数据	84

课后作业

89

项目4 数据查询



4.1 查询工具的使用	91
4.1.1 编辑查询	91
4.1.2 查询结果的显示方法	92
4.2 SELECT查询语句	92
4.2.1 选择列	93
4.2.2 选择行	95
4.2.3 WHERE子句	96
4.2.4 排序查询	100
4.2.5 分组查询	101
4.3 嵌套查询	103
4.3.1 带IN的嵌套查询	104
4.3.2 带比较运算符的嵌套查询	105
4.3.3 带ANY或ALL的嵌套查询	106

4.3.4 带EXISTS的嵌套查询	107
4.4 集合查询	108
4.4.1 并运算 (UNION)	108
4.4.2 交运算 (INTERSECT)	109
4.4.3 差运算 (EXCEPT)	109
4.5 连接查询	110
4.5.1 交叉连接查询	111
4.5.2 内连接查询	111
4.5.3 外连接查询	113
4.6 使用聚合函数统计汇总查询	115

课后作业

117

项目5 视图



5.1 视图简介	119
5.2 创建视图	120
5.2.1 使用图形界面创建视图	120
5.2.2 使用CREATE VIEW语句创建视图	122
5.3 查看视图	125
5.3.1 查看视图中的数据信息	125
5.3.2 查看视图的定义信息	126
5.4 修改视图	129
5.4.1 使用图形界面修改视图	129
5.4.2 使用ALTER VIEW语句修改视图	129

5.4.3 使用存储过程sp_rename修改视图	130
5.5 删除视图	130
5.5.1 使用图形界面删除视图	130
5.5.2 使用DROP VIEW语句删除视图	131
5.6 通过视图更新数据	131
5.6.1 向视图中添加数据	131
5.6.2 修改视图中的数据	131
5.6.3 删除视图中的数据	132

课后作业

133

项目6 索引

6.1 认识索引	135	6.4 索引的分析与维护	144
6.2 索引的分类	135	6.5 全文索引	146
6.2.1 聚集索引	135	6.5.1 使用CREATE FULLTEXT INDEX 创建全文索引	146
6.2.2 非聚集索引	136	6.5.2 使用ALTER FULLTEXT INDEX 更改全文索引	148
6.2.3 其他索引	136	6.5.3 使用DROP FULLTEXT INDEX 删除全文索引	148
6.3 索引的操作	137	6.5.4 全文目录	149
6.3.1 创建索引	137	课后作业	150
6.3.2 查看索引信息	141		
6.3.3 索引的修改	141		
6.3.4 删除索引	143		
6.3.5 设置索引选项	143		

项目7 SQL语言编程基础

7.1 T-SQL简介	152	7.4.5 RETURN	164
7.2 数据类型、常量和变量	152	7.4.6 批处理	164
7.2.1 常量	153	7.4.7 其他命令	165
7.2.2 变量	153	7.5 SQL函数的使用	165
7.3 注释符、运算符和通配符	155	7.5.1 聚合函数	165
7.3.1 注释符	155	7.5.2 日期和时间函数	168
7.3.2 运算符	156	7.5.3 字符串函数	171
7.3.3 通配符	159	7.5.4 数学函数	174
7.3.4 表达式	159	7.5.5 数据类型转换函数	176
7.4 流程控制语句和批处理	160	7.5.6 其他函数类型	176
7.4.1 IF...ELSE语句	160	7.5.7 用户自定义函数	177
7.4.2 BEGIN...END	161	课后作业	179
7.4.3 WHILE...CONTINUE...BREAK	161		
7.4.4 CASE	162		

项目8 存储过程

8.1 存储过程简介	181	8.3 管理存储过程	186
8.1.1 存储过程的概念	181	8.3.1 执行存储过程	186
8.1.2 存储过程的优点	182	8.3.2 查看存储过程	187
8.2 创建存储过程	182	8.3.3 修改存储过程	188
8.2.1 使用向导创建存储过程	182	8.3.4 重命名存储过程	188
8.2.2 用CREATE PROC语句创建 存储过程	183	8.3.5 删除存储过程	189
		课后作业	190

项目9 触发器

9.1 触发器简介	192	9.3.2 修改触发器	203
9.2 创建触发器	193	9.3.3 重命名触发器	204
9.2.1 创建DML触发器	193	9.3.4 禁用和启用触发器	204
9.2.2 创建DDL触发器	199	9.3.5 删除触发器	204
9.2.3 创建登录触发器	201		
9.3 管理触发器	202	课后作业	205
9.3.1 查看触发器	202		

项目10 SQL数据库维护

10.1 导入/导出数据表	207	10.3 数据库状态与分离/附加操作	215
10.1.1 导出SQL Server数据表	207	10.3.1 查看数据库状态	215
10.1.2 导入SQL Server数据表	209	10.3.2 脱机数据库	216
10.2 备份/恢复数据库	211	10.3.3 联机数据库	216
10.2.1 备份类型	211	10.3.4 分离数据库	217
10.2.2 恢复模式	212	10.3.5 附加数据库	218
10.2.3 备份数据库	212		
10.2.4 恢复数据库	214	课后作业	219

项目11 数据库的安全机制

11.1 SQL Server 安全性简介	221	11.4.1 服务器角色管理	229
11.2 安全验证方式	222	11.4.2 数据库角色管理	233
11.2.1 身份验证简介	222	11.5 权限管理	237
11.2.2 验证模式的修改	222	11.5.1 权限管理简介	237
11.3 用户管理	223	11.5.2 使用SSMS管理权限	238
11.3.1 登录用户管理	224	11.5.3 使用T-SQL管理权限	239
11.3.2 数据库用户管理	227		
11.4 角色管理	229	课后作业	241

项目12 企业进销存管理系统综合案例



12.1 系统分析243	12.7.3 “登录”按钮的事件处理262
12.1.1 需求描述243	12.8 系统主窗体设计262
12.1.2 用例图243	12.8.1 设计菜单栏263
12.2 系统设计248	12.8.2 子菜单事件处理263
12.2.1 系统目标248	12.8.3 权限管理265
12.2.2 系统功能结构248	12.9 进货单模块设计266
12.3 数据库与数据表设计248	12.9.1 设计进货单窗体266
12.3.1 数据库系统的概念设计249	12.9.2 添加进货商品267
12.3.2 数据库系统的物理设计251	12.9.3 商品入库270
12.4 创建项目255	12.10 销售单模块设计270
12.5 系统文件夹组织结构255	12.10.1 设计销售单窗体271
12.6 公共类设计255	12.10.2 添加销售商品271
12.6.1 SQLHelper类255	12.10.3 商品出库273
12.6.2 User类258	12.11 库存盘点模块设计273
12.6.3 Jhd类259	12.11.1 设计库存盘点窗体274
12.7 系统登录模块设计261	12.11.2 添加盘点数据275
12.7.1 设计登录窗体261	12.12 运行项目276
12.7.2 “密码”文本框的回车事件261	

附录 课后作业参考答案



参考文献





项目 1

数据库基础



内容概要

随着计算机技术、网络技术和数据存储技术的发展，数据库的应用已经涉及生活的方方面面。本项目将介绍数据库的基本概念、体系结构、数据模型等数据库方面的基础知识，为后续学习打下坚实的基础。



知识要点

- 数据管理技术的起源与发展。
- 数据库系统的组成。
- 数据库的体系结构。
- 数据模型。
- 关系数据库的设计。
- 常见的关系型数据库管理系统。

1.1 数据库系统概述

在数据库中，数据按照标准进行存储和使用。要保持数据库的功能性和稳定性，需要有一套完备的数据库系统。为此需要先了解数据库系统的一些基本概念。

1.1.1 数据管理技术的起源与发展

数据是现实世界中实体（或客体）在计算机中的符号表示，数据不仅可以是数字，还可以是文字、图形、图像、音频和视频等。现实生活中需要管理大量的数据，例如，在学校有学生、教工、课程和成绩等方面的数据，在医院有病历、药品、医生、处方等方面的数据，在银行有存款、贷款、信用卡和投资理财业务等方面的数据。因此，对各种数据实现有效的管理具有重要意义。

随着计算机技术和网络技术的发展，尤其是云计算的广泛应用，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应用。实际上，在数据库技术出现前，人们就已经开始研究数据管理技术。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三个发展阶段。

1. 人工管理阶段

20世纪50年代中期以前，计算机主要用于科学计算。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，在这样的情况下，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己管理，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，而且要设计物理结构，包括存储结构、存取方法、输入方式等，因此程序员负担很重。

（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，因此程序与程序之间有大量的数据冗余。

知识点拨

数据冗余

数据冗余是指数据之间的重复，也可以说是同一数据存储在不同数据文件中的现象。

（4）数据不具有独立性。

数据的逻辑结构或物理结构改变后，必须对应用程序做相应的修改，这就进一步加重了程

序员的负担。

在人工管理阶段，程序与数据之间的一一对应关系，如图1-1所示。

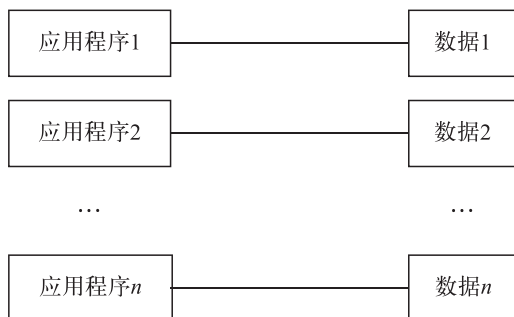


图 1-1 人工管理阶段程序与数据的对应关系

2. 文件系统阶段

20世纪50年代后期到60年代中期已有了磁盘、磁鼓等直接存储设备；而在计算机系统方面，不同类型的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。这时可以把相关的数据组织成一个文件存放在计算机中，在需要的时候只要提供文件名，计算机就能从文件系统中找出所要的文件，把文件中存储的数据提供给用户进行处理。但是，由于这时数据的组织仍然是面向程序的，所以仍然存在大量的数据冗余，也无法有效地进行数据共享。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，那么应用程序的改变很小，节省了程序的维护工作量。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，给数据的修改和维护带来了困难。

(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，而要想对现有的数据再增加一些新的应用会很困难，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，还要修改文件结构，所以，数据与程序之间是缺乏独立性的。文件系统阶段程序与数据之间的关系，如图1-2所示。

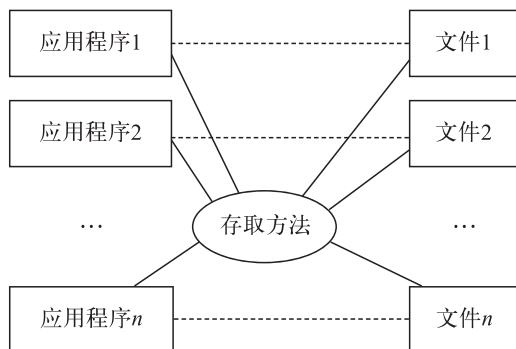


图 1-2 文件系统中程序与数据的关系

知识点拨

突发事件的应对不足

在文件系统阶段还有一项重要的缺点就是对于各种突发事件，如文件误删除、磁盘故障等情况，无法有效地应对，这对数据安全来说是非常大的弊端。

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，应用越来越广泛，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。这时已有大容量磁盘，硬件价格下降，软件价格则上升，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求，于是为满足多用户、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了，出现了统一管理数据的专用软件系统——数据库管理系统。

用数据库系统来管理数据比文件系统具有明显的优点，从文件系统到数据库系统，标志着数据管理技术的飞跃。数据库系统具有如下特点。

(1) 数据结构化。

数据库系统实现了整体数据的结构化，这是数据库最主要的特征之一。“整体”结构化是指在数据库中的数据不再仅针对某个应用，而是面向全组织；不仅数据内部结构化，而且整体结构化，数据之间有联系。

(2) 数据的共享性高，冗余度小，易扩充。

因为数据是面向整体的，所以数据可以被多个用户、多个应用程序共享使用，可以大大减少数据冗余，节约存储空间，避免数据之间的不相容性与不一致性。

(3) 数据独立性高。

数据独立性包括数据的物理独立性和逻辑独立性。物理独立性是指数据在磁盘上的数据库中如何存储是由数据库管理系统（Database Management System, DBMS）管理的，用户程序不需要了解，应用程序要处理的只是数据的逻辑结构，这样一来当数据的物理存储结构改变时，用户的程序不用改变。逻辑独立性是指用户的应用程序与数据库的逻辑结构是相互独立的，也就是说，数据的逻辑结构改变了，用户程序可以不改变。

数据与程序的独立，把数据的定义从程序中分离出去，加上存取数据是由DBMS负责的，

从而简化了应用程序的编制，大大减少了应用程序的维护和修改。

(4) 数据由DBMS统一管理和控制。

数据库系统中的数据由DBMS来进行统一管理和控制，所有应用程序对数据的访问都要交给DBMS来完成。

知识点拨

DBMS 提供的控制功能

DBMS 主要提供以下控制功能：数据的安全性保护（security）；数据的完整性检查（integrity）；数据库的并发访问控制（concurrency）；数据库的故障恢复（recovery）。

在数据库系统阶段，程序与数据之间的对应关系，如图1-3所示。

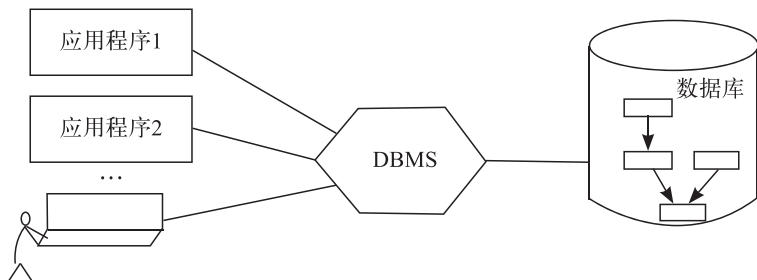


图 1-3 数据库系统中程序与数据的关系

上述数据管理技术发展的三个阶段的比较，如表1-1所示。

表 1-1 数据管理阶段对比

		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	无直接存取设备	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
	数据的共享程度	某一应用程序	某一应用	现实世界
	数据面向的对象	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

1.1.2 数据库系统的组成

数据库系统（Database System, DBS），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完成，这些人被称为数据库管理员（Date Base Administrator, DBA）。

通常，在不引起混淆的情况下，一般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。

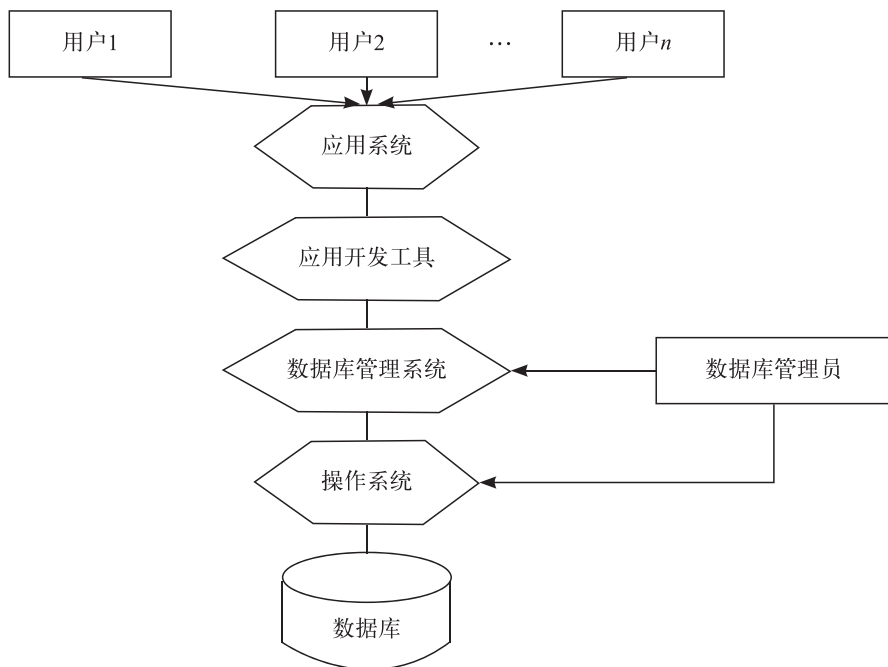


图 1-4 数据库系统组成

数据库系统在计算机系统中的地位，如图1-5所示。

数据库系统的主要组成部分如下所述。

1. 硬件平台

硬件系统主要指计算机各个硬件组成部分。鉴于数据库应用系统的需求，往往特别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、作业处理能力强。对于分布式数据库，计算机网络也是基础环境。硬件平台的具体要求如下。

（1）要有足够大的内存，能存放操作系统和DBMS的核心模块、数据库缓冲区和应用程序。

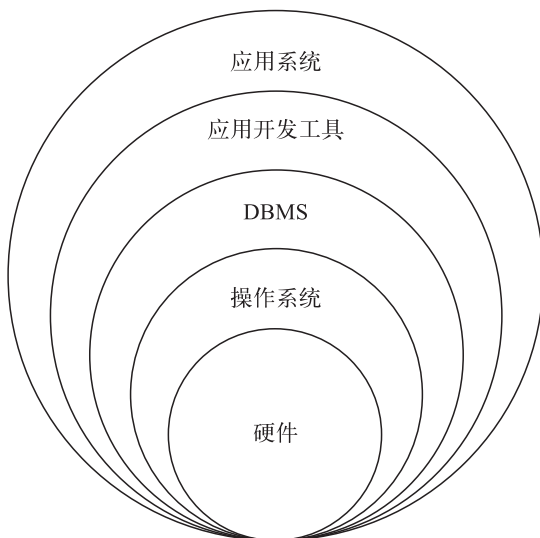


图 1-5 计算机系统

- (2) 有足够大的硬盘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) 要有较强处理能力的中央处理器（CPU）以保证数据处理的速度。

2. 数据库

数据库（Database），就是存放数据的仓库。只不过这个仓库是在计算机的存储设备上，而且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，学生成绩管理系统中，学生的基本信息、课程信息、成绩信息等都是从学生成绩管理数据库。

严格地讲，数据库是长期存储在计算机内、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，并可为各种用户共享。简单来说，数据库数据具有永久存储、有组织及可共享三个基本特点。

3. 软件

数据库搭建完毕后，需要对数据库进行维护、信息录入、读取、删除等操作，这些都需要各种管理软件的支持。数据库的软件系统主要包括以下几个部分。

(1) DBMS。

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，完成这个任务的是一个系统软件——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。

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作为一种可管理的资源来处理。DBMS应使数据易于为各种不同的用户所共享，并增进数据的安全性、完整性及可用性，提供高度的数据独立性。

知识点拨

DBMS 的主要功能

DBMS 的主要功能有：数据的定义功能，数据的操纵功能，数据的控制功能等。

- (2) 支持DBMS运行的操作系统。
- (3) 与数据库通信的高级程序语言及编译系统。
- (4) 为特定应用环境开发的数据库应用系统。

4. 数据库管理员及其他相关人员

所有的软硬件及数据库都需要有人工负责的部分，有些还必须由人工完成。所以在数据库系统的人员方面需要有：数据库管理员（DBA）、系统分析员、应用程序员和普通用户。

(1) 数据库管理员。

数据库管理员负责管理和监控数据库系统，负责为用户解决应用中出现的系统问题。为了

保证数据库能够高效正常地运行，大型数据库系统都设有专人负责数据库系统的管理和维护。其主要职责如下。

- 决定数据库中的信息内容和结构。数据库中要存放哪些信息，DBA要参与决策。因此DBA必须参加数据库设计的全过程，并与用户、应用程序员、系统分析员密切合作，共同协商，做好数据库设计工作。
- 决定数据库的存储结构和存取策略。
- 监控数据库的运行（系统运行是否正常，系统效率如何），及时处理数据库系统运行过程中出现的问题。例如，系统发生故障时，数据库若因此遭到破坏，DBA必须在最短的时间内把数据库恢复到正确状态。
- 安全性管理，通过对系统的权限设置、完整性控制设置来保证系统的安全性。DBA要负责确定各个用户对数据库的存取权限、数据的保密级别和完整性约束条件等。
- 日常维护，如定期对数据库中的数据进行备份、维护日志文件等。
- 对数据库有关文档进行管理。

总之，数据库管理员在数据库管理系统的正常运行中起着非常重要的作用。

(2) 系统分析员。

系统分析员负责应用系统的需求分析和规范说明，和用户及DBA相配合，确定系统的硬件、软件配置，并参与数据库系统概要设计。

(3) 应用程序员。

应用程序员是负责设计、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。

■ 1.1.3 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，完成这个任务的是一个系统软件—数据库管理系统（database management system,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，在数据库系统中占据着举足轻重的地位，它是应用软件与底层操作系统软件之间的桥梁，在整个数据库系统中起着关键的作用。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。

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作为一种可管理的资源，成为易于为各种不同用户所共享的资源，并增进数据的安全性、完整性、可用性，以及提供高度的数据独立性。

DBMS 的主要功能如下。

- 数据的定义功能。
- 数据的操纵功能。
- 数据的控制功能。
- 其他功能。

1.2 数据库的体系结构

考察数据库系统的结构可以有多种不同的层次或角度。从系统组成角度来看,数据库系统的结构主要分为内部结构和外部结构。内部结构是从数据库管理系统角度看的,数据库系统通常采用三级模式结构;外部结构是从数据库最终用户角度看的,数据库系统可分为单用户结构、主从式结构、客户端/服务器结构(C/S结构)、浏览器/服务器结构(B/S结构)和分布式结构等。

本节分别从内部结构和外部结构介绍数据库系统的结构。

1.2.1 数据库系统的内部结构

虽然实际的数据库系统软件的产品种类很多,并且支持不同的数据模型,使用不同的数据库管理语言,可应用在不同的操作系统中。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1-6所示。

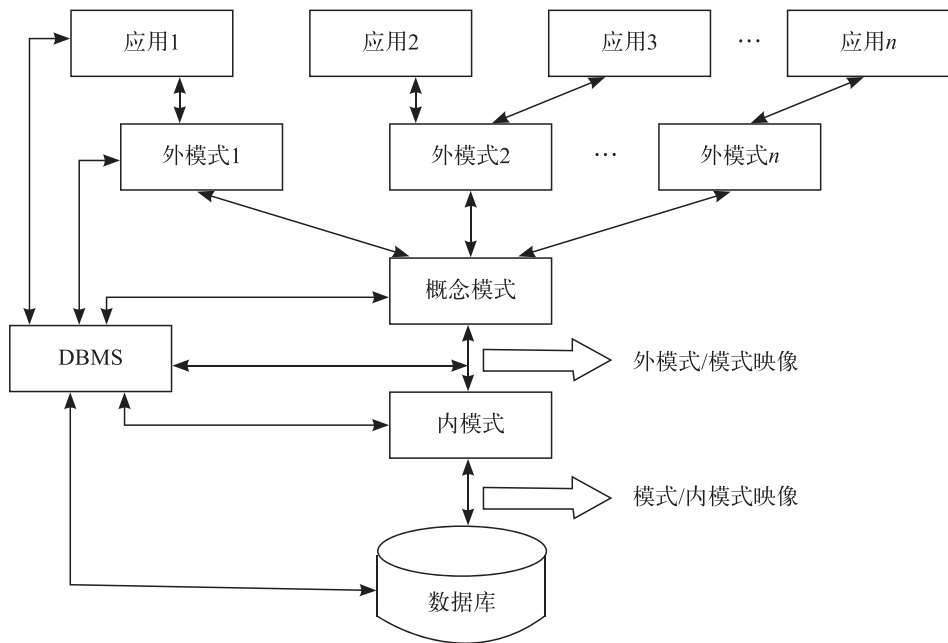


图 1-6 数据库系统的模式结构

(1) 模式。

模式 (Schema) 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,是所有用户的公共视图,它仅仅涉及类型 (Type) 的描述,不涉及具体的值 (Value)。模式的定义中主要包含数据的逻辑结构 (数据项的名字、类型、取值范围等)、数据之间的联系以及数据有关的安全性要求等方面的描述。一个数据库只有一个模式。

(2) 外模式。

外模式也称子模式 (Subschema) 或用户模式, 它是数据库用户 (包括应用程序员和最终用户) 能够看见和使用的局部数据的逻辑结构和特征的描述, 是数据库用户的数据视图, 是与某一应用有关的数据的逻辑表示。

外模式通常是模式的子集。一个模式可以有多个外模式。由于它是各个用户的数据视图, 如果不同的用户在不同应用需求、看待数据的方式、对数据保密的要求等方面存在差异, 则其外模式描述就可能不同。即使是模式中的同一数据, 在外模式中的结构、类型、长度、保密级别等都可以不同。另外, 同一外模式也可以为某一用户的多个应用系统所使用, 但一个应用程序只能使用一个外模式。

外模式是保证数据库安全性的一个有力措施。每个用户只能看到和访问所对应的外模式中的数据, 数据库中的其他数据是看不到的。

设立外模式的好处如下。

- 方便了用户的使用, 简化了用户的接口。用户只要依照模式编写应用程序或在终端输入命令, 无须了解数据的存储结构。
- 保证数据的独立性。由于在三级模式之间存在两级映像, 使得物理模式和概念模式的变化, 都反映不到子模式一层, 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子模式, 减少了数据的冗余度, 有利于为多种应用服务。
- 有利于数据的安全和保密。用户程序只能操作其子模式范围内的数据, 从而把其与数据库中的其余数据隔离开来, 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。

内模式 (Internal Schema) 也称存储模式 (Storage Schema), 一个数据库只有一个内模式。它是数据物理结构和存储方式的描述, 定义所有的内部记录类型、索引和文件的组织形式, 以及数据控制方面的细节。

内部记录并不涉及物理记录, 也不涉及设备的约束。比内模式更接近于物理存储和访问的那些软件机制是操作系统的一部分 (即文件系统), 例如, 从磁盘读数据或写数据到磁盘上的操作等。

2. 数据库系统的二级映像和数据独立性

为了让用户不必考虑存取路径等细节, 减少应用程序的维护和修改工作, 需要保证程序和数据之间的独立性, 即当数据改变时程序不需要改变; 反之, 程序改变时数据也不需要改变。为了使程序和数据之间具有一定的独立性, DBMS提供了两层映像, 外模式/模式映像和模式/内模式映像。

映像实质上是一种对应关系, 是指映像双方如何进行数据转换, 并定义转换规则。有了这两层映像, 用户在处理数据时不必关心数据在计算机中的具体表示方式与存储方式。正是这两层映像保证了数据库系统中的数据能够具有较高的逻辑独立性和物理独立性。

(1) 外模式/模式映像。

外模式/模式映像定义了外模式与模式之间的对应关系。如果模式需要进行修改, 例如, 数

据重新定义，增加新的关系、新的属性、改变属性的数据类型等，那么只需对各个外模式/模式的映像做相应的修改，使外模式尽量保持不变；而应用程序一般是依据外模式编写的，因此应用程序也不必修改，从而保证了数据与程序的逻辑独立性，这就是数据的逻辑独立性。

(2) 模式/内模式映像。

模式/内模式映像定义了模式和内模式之间的对应关系，即数据全局逻辑结构与存储结构之间的对应关系。模式/内模式映像一般是在模式中描述的。当数据库的存储结构改变时（如采用了另外一种存储结构），由数据库管理员对模式/内模式映像做相应改变，可以使模式保持不变，因此应用程序也不必改变。这就保证了数据与程序的物理独立性，简称数据的物理独立性。

1.2.2 数据库系统的外部结构

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，数据库系统的结构分为单用户结构、主从式结构、客户端/服务器结构（C/S结构）、浏览器/服务器结构（B/S结构）和分布式结构等。

1. 单用户数据库系统

单用户数据库系统是指早期最简单的数据库系统，如图1-7所示。在单用户系统中，整个数据库系统包括应用程序、DBMS和数据等，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。

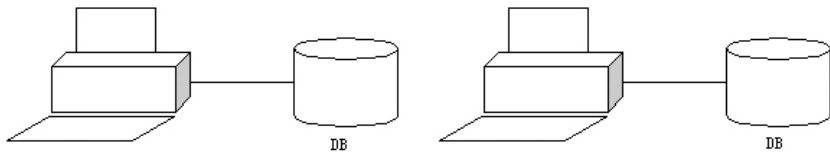


图 1-7 单用户数据库系统

例如，一个企业中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。由于不同部门之间不能共享数据，因此企业内部存在大量的数据冗余。

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和数据等，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，共享数据资源，如图1-8所示。

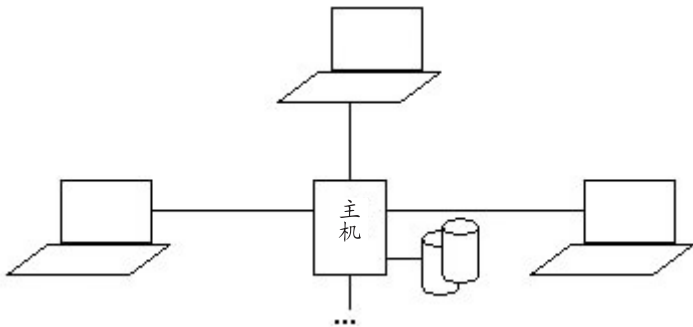


图 1-8 主从式结构数据库系统

主从式结构数据库系统的优点是结构简单，数据易于维护和管理。其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，从而会使系统性能大幅度下降。另外，当主机出现故障后，整个系统不能运行，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个整体，但物理分布在计算机网络的不同节点上，如图1-9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；也可以同时存取和处理多个异地数据库中的数据，执行全局应用。

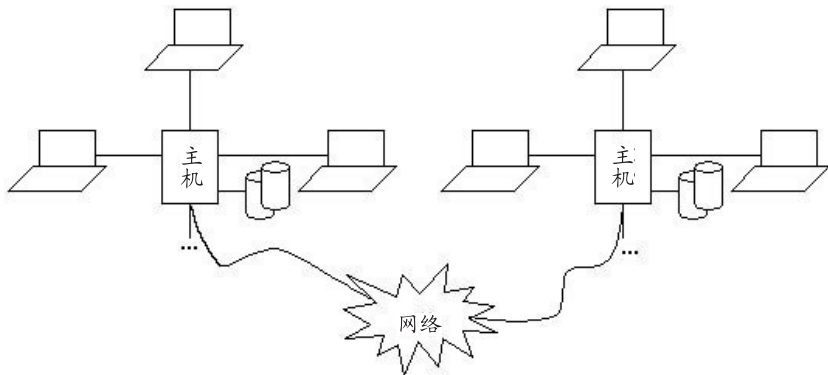


图 1-9 分布式结构数据库系统

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，系统效率会明显地受网络通信的制约。

4. 客户端/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的功能和应用分开。网络中某些节点上的计算机专门执行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安装DBMS外围应用开发工具，支持用户的应用，称为客户机；这就是客户端/服务器（Client/Server）结构，简称为C/S结构，如图1-10所示。

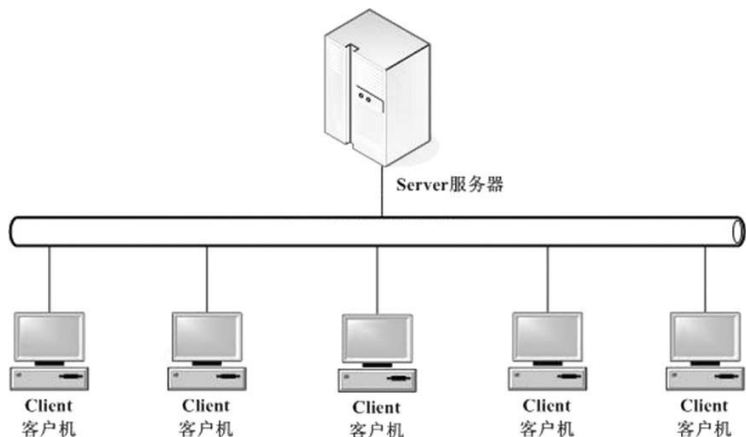


图 1-10 C/S 结构数据库系统

在客户端/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。

5. 浏览器/服务器结构的数据库系统

随着互联网的飞速发展，移动办公和分布式办公越来越普及，而C/S结构的缺点就逐渐暴露出来，特别是客户端需要安装专用的客户端软件。一旦客户端软件升级，那么所有的客户计算机上的客户端软件均需要更新，因此需要对C/S结构进行改进，浏览器/服务器结构（Browser/Server）便应运而生，简称为B/S结构，如图1-11所示。

在浏览器/服务器结构下，用户工作界面是通过浏览器实现的，主要的事务逻辑都在服务器端（Server）实现，只有极少部分的事务逻辑在前端（Browser）实现。这种模式统一了客户端，将系统功能实现的核心部分集中到服务器上，简化了系统的开发、维护和使用。客户机上只要安装一个浏览器（Browser），浏览器通过Web Server同数据库进行数据交互，大大减轻了客户端机器的载荷，减少了系统维护与升级的成本和工作量，降低了用户的总体成本。

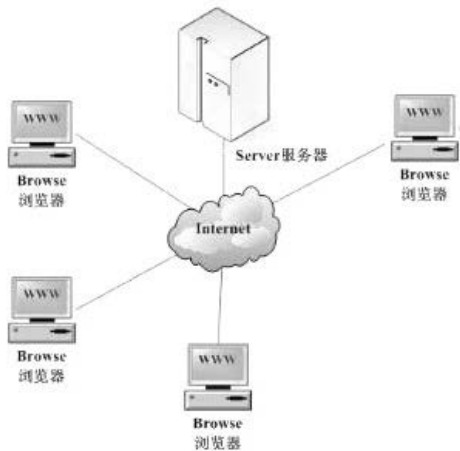


图 1-11 B/S 结构数据库系统

1.3 数据模型

数据在数据库中是按照标准进行存储和使用的。为了保持数据库的功能性和稳定性，需要一整套完备的数据库系统进行运作。

1.3.1 数据模型简介

数据模型（Data Model）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、概念等信息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（Logic Data Model）和物理数据模型（Physical Data Model）。

（1）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、关系模型和面向对象模型等。

❶ 注意事项：4种逻辑模型的区别

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。

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

(3) 物理数据模型又称物理模型，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，也称为数据模型的三大要素。概念数据模型非常多，在这里介绍最经典的概念模型——ER模型；同样，在几种逻辑数据模型中，这里只介绍目前主流的逻辑数据模型——关系模型。

■ 1.3.2 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P. 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类型及实体间联系，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。

1. E-R模型术语与E-R图

首先介绍有关的名词术语及E-R图，具体内容如下。

(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。

(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、系等方面的属性。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“(220001,肖敏,19,女,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被称为实体的码。例如，学生的学号可作为码，但因为学生的姓名可能有重名，所以不能作为学生实体的码。

(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，在信息世界中将被抽象为实体型内部的联系和实体型之间的联系。

2. 实体型联系类型

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。

(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、省与市。

(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，工厂与产品。

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。可以用图形来表示两个实体型之间的这三类联系，如图1-12所示。

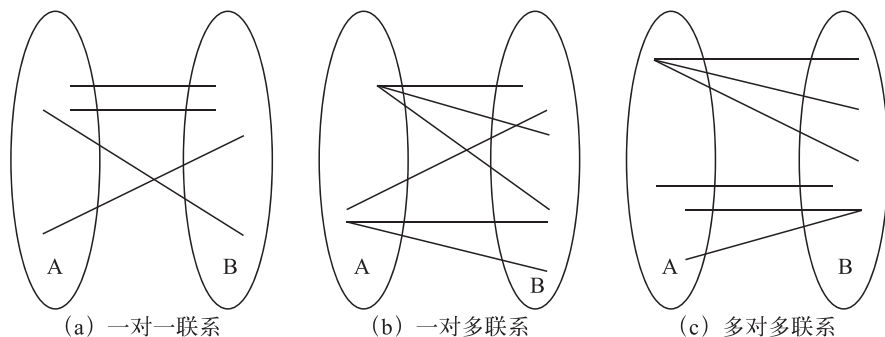


图 1-12 实体间的联系

知识点拨

E-R 图中的 4 个基本成分

(1) 矩形框，表示实体类型（研究问题的对象）。

(2) 菱形框，表示联系类型（实体间的联系）。

(3) 椭圆形框，表示实体类型和联系类型的属性。

相应的命名均记入各种框中。对于实体标识符的属性，在属性名下面画一条横线。

(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）。

3. E-R模型的优点

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模型转换成计算机已能实现的数据模型。

■ 1.3.3 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：层次模型（Hierarchical Model）、网状模型（Network Model）、关系模型（Relational Model）和面向对象模型（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了主导地位，20世纪80年代后就逐渐被关系模型的数据库系统取代。但在美国等一些国家，由于早期开发的应用系统都是基于层次或网状模型的数据库系统，因此目前仍有不少层次或网状数据库系统在继续使用。

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，为关系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，非关系系统的产品也大都加上了对关系模型的接口。数据库领域当前的研究工作也基本上是以关系方法为基础的。

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，同样也促进了数据库中面向对象数据模型的研究和发展。

关系模型是目前最重要的一种数据模型。关系数据库系统采用关系模型作为数据的组织方式。关系数据库系统与非关系数据库系统的区别：关系数据库系统中只有“表”这一种数据结构，而非关系数据库系统中还有其他类型数据结构，对这些数据结构还有其他的操作。下面主要介绍关系模型的相关知识。

1. 关系模型的基本术语

在关系模型中，用单一的二维表结构来表示实体及实体间的关系。

(1) 关系（Relationship）：一个关系对应一个二维表，二维表名就是关系名。

(2) 关系模式（Relationship Schema）：二维表中的行（表头）定义记录的类型，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 ）。例如，学生关系模式：学生（学号,姓名,年龄,性别,院系）。

(3) 属性（Attribute）及值域（Domain）：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 n 的关系称为 n 元关系，度为1的关系称为一元关系，度为2的关系称为二元关系。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。

(4) 元组（Tuple）：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。

(5) 键 (Key): 键也称为码, 由一个或多个属性组成。

知识点拨

几种实际使用的键

在实际使用中, 有下列几种键。

(1) 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。

(2) 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)。

(3) 外键 (Foreign Key): 设 F 是关系 R 的一个或一组属性, 但不是关系 R 的键。如果 F 与关系 S 的主键相对应, 则称 F 是关系 R 的外键, 关系 R 称为参照关系, 关系 S 称为被参照关系。

(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。

2. 关系的性质

一般用集合的观点定义关系, 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, 不允许表中嵌套表。如表1-2所示中就出现了这种表中再嵌套表的情况, 在“成绩”下嵌套“平时”和“期末”。虽然类似的表在实际生活中司空见惯, 但是却不符合关系的基本定义。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, 所以不可能出现表中套表的现象。遇到这种情况, 可以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, 可把表1-2改成表1-3。这里把“成绩”分成两列, “平时成绩”和“期末成绩”, 两个属性都取自同一个域“成绩”。

表 1-2 二维表 1

学号	姓名	成绩	
		平时	期末
201804001	陈家乐	90	88
201804002	王静茹	98	96

表 1-3 二维表 2

学号	姓名	平时成绩	期末成绩
201804001	陈家乐	90	88
201804002	王静茹	98	96

(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，可随意指定属性的排列次序，因为交换属性顺序的先后，并不改变关系的实际意义。例如，在定义表1-3所示的关系模式时，可以指定属性的次序为（学号,姓名,平时成绩,期末成绩），也可以指定属性的次序为（学号,姓名,期末成绩,平时成绩）。

(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。由于关系是一个集合，因此不考虑元组间的顺序问题。在实际应用中，常常对关系中的元组排序，这样做仅仅是为了加快检索数据的速度，提高数据处理的效率。

❶ 注意事项：判断关系是否相等

判断两个关系是否相等，是从集合的角度来考虑的，与属性的次序无关，与元组次序无关，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. 关系的完整性

关系模型的完整性规则是对关系的某种约束条件。关系模型中可以有3类完整性约束：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，如银行关系对应于银行的集合。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：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，在关系模型中实体及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。

设F是基本关系R的一个或一组属性，但不是关系R的键，如果F与基本关系S的主键Ks相对应，则称F是基本关系R的外键 (Foreign Key)，并称基本关系R为参照关系 (Referencing Relation)，基本关系S为被参照关系 (Referenced Relation) 或目标关系 (Target Relation)。关系R和S不一定是不同的关系。参照完整性规则就是定义外键与主键之间的引用规则。

参照完整性规则：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。

(3) 用户定义的完整性 (User-defined Integrity)。

实体完整性和参照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，以便用统一系统的方法处理，而不是由应用程序承担这一功能。

1.4 关系数据库的设计

要开发管理信息系统，数据库设计的好坏是关键。数据库设计是指在给定的环境下，创建一个性能良好、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。从本质上讲，数据库设计是将数据库系统与现实世界相结合的一种过程。

人们总是力求设计出的数据库好用，但是设计数据库时既要考虑数据库的框架和数据结构，又要考虑应用程序存取数据库和处理数据。因此，最佳设计不可能一蹴而就，只能是一个反复探寻的过程。

大体上可以把数据库设计划分成以下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1-13所示。

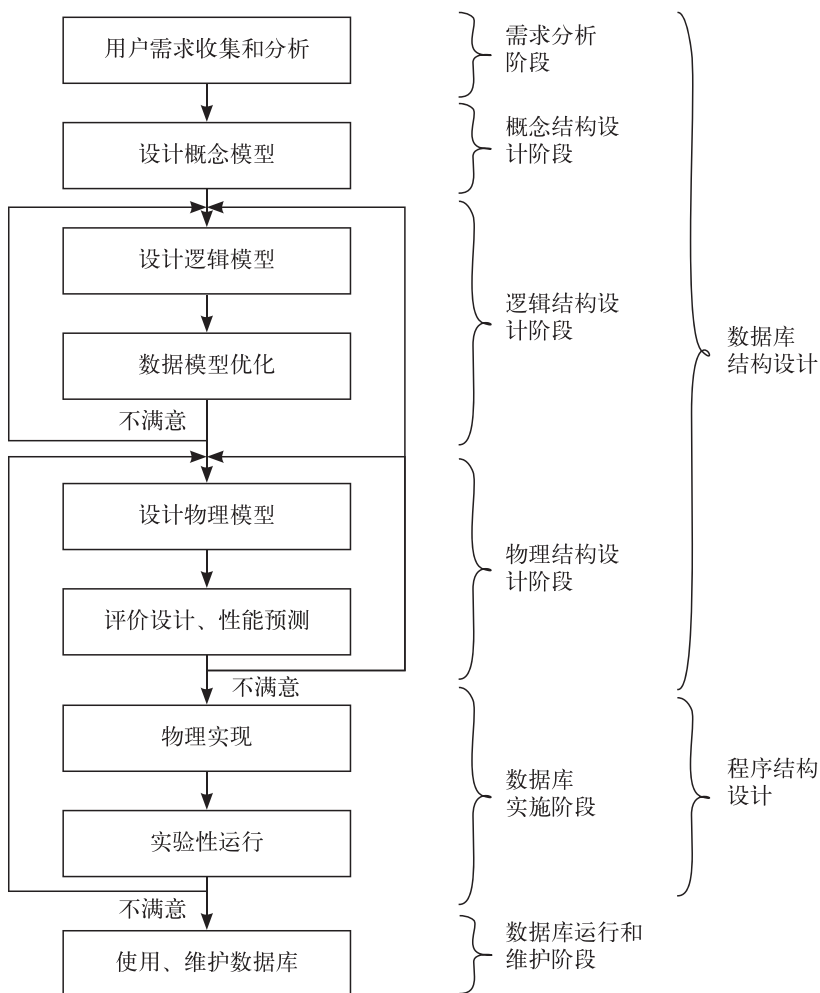


图 1-13 数据库设计阶段划分

■ 1.4.1 需求分析

准确地搞清楚用户需求，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。

1. 需求分析的成果

需求分析阶段的成果是系统需求说明书，主要包括数据流图、数据字典、各种说明性表格、统计输出表、系统功能结构图等。系统需求说明书是以后设计、开发、测试和验收等过程的重要依据。

❶ 注意事项：需求分析的重点

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。

2. 需求分析的任务

需求分析的任务是通过详细调查现实世界要处理的对象（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，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。需求分析阶段的主要任务有以下几个方面。

（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，各种数据间的关系，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、数据流程图。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（也叫加工），以及处理后数据流向的全过程的分析。换言之，功能分析是对所建数据模型支持的系统事务处理的分析。

知识点拨

数据流分析

数据流分析是对事务处理所需的原始数据的收集以及处理后所得数据及其流向的分析，一般用数据流程图来表示。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、事务处理的说明和所需产生的报告，并且尽量借助于数据字典加以说明。除了使用数据流程图、数据字典，需求分析还可使用判定表、判定树等工具。

■ 1.4.2 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，是面向现实世界的数据库模型，极易为用户所理解。为保证所设计的概念数据模型能正确、完全地反映用户（一个单位）的数据及其相互联系，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），以形成一个单位的概念数据模型。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与修改，最后才能形成所需的数据库模型。

■ 1.4.3 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不被DBMS理解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部模式，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，并使其满足DBMS的各种限制。逻辑结构设计阶段的结果是DBMS提供的数据库定义语言（DDL）写成的数据库模式。逻辑结构设计的具体方法与DBMS的逻辑数据库模型有关。

■ 1.4.4 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或存储模式。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。

■ 1.4.5 数据库实施

数据库实施主要包括以下工作：用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，就可以用选好的DBMS提供的数据库定义语言（DDL）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。组织数据入库是数据库实施阶段最主要的工作。对于数据量不大的小型系统，可以用人工方式完成数据入库，其步骤如下。

（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。筛选出来的需要入库的数据，其格式往往不符合数据库要求，还需要进行转换。这种转换有时可能会很复杂。

(3) 输入数据。将转换好的数据输入计算机中。

(4) 校验数据。检查输入的数据是否有误。

对于大型系统，由于数据量大，用人工方式组织数据入库将会耗费大量人力物力，而且很难保证数据的正确性。因此应该设计一个数据输入子系统，由计算机来辅助数据入库工作。

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入库并行进行。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。调试应用程序时由于数据入库尚未完成，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，其主要工作包括以下两项内容。

(1) 功能测试。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。测量系统的性能指标，分析是否符合设计目标。

1.4.6 数据库运行和维护

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。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。

1.5 常见的关系型数据库管理系统

自20世纪70年代关系模型的概念提出后，由于其突出的优势，迅速被商用关系型数据库管理系统所采用。据统计，在新发展的DBMS系统中，关系型数据库管理系统已占到90%。其中涌现出了大量性价比高、功能强大的关系型数据库管理系统产品。

1. Oracle

Oracle数据库被业界认为是目前比较成功的关系型数据库管理系统。Oracle公司是世界第二大软件供应商，是数据库软件领域第一大厂商（大型机市场除外）。Oracle的数据库产品被认为是运行稳定、功能齐全、性能超群的高端产品。这一方面反映了它在技术方面的领先，另一方面也反映了它在价格定位上更着重于大型的企业数据库领域。对于数据量大、事务处理繁忙、安全性要求高的企业，Oracle无疑是比较理想的选择。当然，用户必须在费用方面做出充足的考虑，因为Oracle数据库在同类产品中是比较贵的。随着因特网的普及，带动了网络经济

的发展, Oracle适时地将自己的产品紧密地和网络计算结合起来, 成为在因特网应用领域数据库厂商的佼佼者。

2. SQL Server

SQL Server是微软公司开发的大型关系型数据库系统。SQL Server的功能比较全面, 效率高, 可以作为大中型企业或单位的数据库平台。SQL Server在可伸缩性与可靠性方面做了许多工作, 近年来在许多企业的高端服务器上得到了广泛的应用。同时, 该产品继承了微软产品界面友好、易学易用的特点, 与其他大型数据库产品相比, 在操作性和交互性方面独树一帜。SQL Server可以与Windows操作系统紧密集成, 这种安排使SQL Server能充分利用操作系统所提供的特性, 无论是应用程序开发速度还是系统事务处理运行速度, 都能得到较大的提升。另外, SQL Server可以借助浏览器实现数据库查询功能, 并支持内容丰富的扩展标记语言(XML), 提供了全面支持Web功能的数据库解决方案。对于在Windows平台上开发的各种企业级信息管理系统来说, 无论是C/S(客户机/服务器)架构还是B/S(浏览器/服务器)架构, SQL Server都是一个很好的选择。

3. Access

Access是微软Office办公套件中一个重要成员。Access简单易学, 任何一个普通的计算机用户都能掌握并使用它。同时, Access的功能也足以应付一般的小型数据管理及处理需要。无论用户是要创建一个个人使用的独立的桌面数据库, 还是部门或中小公司使用的数据库, 在需要管理和共享数据时, 都可以使用Access作为数据库平台, 提高工作效率。例如, 可以使用Access处理公司的客户订单数据, 管理自己的个人通讯录, 科研数据的记录和处理等。但Access只能在Windows系统下运行。

Access最大的特点是界面友好, 简单易用, 和其他Office成员一样, 极易被一般用户所接受。因此, 在许多低端数据库应用程序中, 经常使用Access作为数据库平台。在初次学习数据库系统时, 很多用户也是从Access开始的。

4. MySQL

MySQL是由瑞典MySQL AB公司开发的开源关系型数据库产品, 目前为Oracle公司所有。MySQL是一种关联数据库管理系统, 关联数据库将数据保存在不同的表中, 而不是将所有数据放在一个大仓库内, 这样就增加了速度并提高了灵活性。由于其体积小、速度快、总体拥有成本低, 尤其是开放源码这一点, 一般中小型网站的开发都选择MySQL作为网站数据库。因此, MySQL是目前最流行的关系型数据库管理系统之一, 特别是在Web应用方面, MySQL是最好的关系数据库管理系统(Relational Database Management System, RDBMS)软件。

 课后作业

一、选择题

1. 下列关于数据库的叙述中，（ ）是正确的。
 - A. 数据库减少了数据冗余
 - B. 数据库避免了数据冗余
 - C. 数据库中的一致性是指数据类型是一致的
 - D. 数据库系统比文件系统能管理更多数据
2. 在数据库中存储的是（ ）。
 - A. 信息
 - B. 数据
 - C. 数据结构
 - D. 数据模型
3. Access数据库是（ ）数据管理系统。
 - A. 层状
 - B. 网状
 - C. 关系型
 - D. 树状

二、简答题

1. 简述数据库系统的内部结构和外部结构。
2. 什么是DBMS? DBMS的主要功能有哪些?
3. 简述数据库管理员的主要职责。