



21世纪普通高等教育立体化精品教材

Access数据库及应用

ACCESS SHUJUKU JI YINGYONG

谢鹏 金伟 主编

葛娅 副主编

李洪双 赵偲榆 高淑慧 参编

航空工业出版社

北京

内 容 提 要

本书是基于全国计算机等级考试“Access 数据库程序设计”(二级)的考试大纲编写的,共分为 8 个章节,分别对数据库基础知识、Access 2010 数据库系统概述、表的建立和使用、查询、窗体、报表、宏和库存管理系统的相关内容进行了详细的介绍,通过相关案例的详细操作,降低了理论难度和抽象性,增强了学生的学习兴趣和理解能力。

本书基于“必需、够用”的原则和“以能力为本位”的理念进行编写,具有通俗易懂、内容翔实、深入浅出的特点,并配有相应的实训教材。

本书可作为普通高等院校计算机及相关专业的数据库教材,也可作为全国计算机等级考试“Access 数据库程序设计”的参考书。

图书在版编目(CIP)数据

Access 数据库及应用/谢鹏,金伟主编. —北京:
航空工业出版社,2019.9

ISBN 978-7-5165-2005-5

I. ①A… II. ①谢…②金… III. ①关系数据库系统
IV. ①TP311.138

中国版本图书馆 CIP 数据核字(2019)第 177580 号

Access 数据库及应用 Access Shujuku Ji Yingyong

航空工业出版社出版发行

(北京市朝阳区北苑 2 号院 100012)

发行部电话:010—84936597 010—84936343

河北祥浩印刷有限公司印刷

全国各地新华书店经售

2019 年 9 月第 1 版

2019 年 9 月第 1 次印刷

开本:787×1092 1/16

印张:14.5

字数:334 千字

印数:1—10000

定价:37.50 元

PREFACE 前言

随着信息技术的发展,特别是 20 世纪 90 年代以后,数据管理不再是存储和管理数据,而转变成用户所需要的各种数据管理的方式——数据库系统。Microsoft Office Access 2010 是面向对象的关系型数据库系统,其出现使得数据库的应用和开发变得更加简洁、灵活。

本书介绍了 Microsoft Office Access 数据库的基本理论和各种数据库对象(表、查询、窗体、报表、宏)的基本操作,共分为 8 个章节。

第 1 章主要介绍了数据库管理技术的发展时期、数据库体系的概念、数据库体系的三级模式结构、数据模型的概念及其数据模型的类别划分、关系代数运算等。

第 2 章主要介绍 Microsoft Office Access 2010 的发展过程、软件特点及软件的基本操作。

第 3 章主要介绍表的创立和使用,主要内容包括创建表、数据类型、字段属性、数据有效性、编辑表、设置数据表格式。

第 4 章主要介绍查询的相关内容,包括查询的概念、类型、创建方法以及各种类型的查询操作。

第 5 章主要介绍窗体的相关内容,通过案例的讲解,介绍了窗体的创建方法与设计过程。

第 6 章主要介绍报表的创建,通过案例的讲解,介绍了报表控件的使用以及报表的打印。

第 7 章主要介绍宏的基本概念,宏的分类以及功能、创建方法、调试运行及相关的

应用。

第 8 章是综合案例的介绍,通过将表、报表、窗体、查询、宏等数据库对象结合起来,设计一个“库存管理系统”。

本书以“教、学、做一体化”为思路,以案例为引导,以提高学生实践能力为目的,重点介绍数据库的理论和数据库对象的基本操作,使学生掌握数据库的功能和基本操作,从而能够设计小型的数据库系统。

本书由谢鹏、金伟任主编,由葛娅任副主编,参加编写工作的还有李洪双、刘丽萍、胡悦、张晓潘、高淑慧、赵偲榆、李方方、王赵梦、胡远伟、林丽萍等。王晓琼和梅文等老师对本书的编写工作提出了许多宝贵的意见,在此一并致谢。

由于作者水平有限,书中难免存在有不妥与错漏之处,敬请读者批评指正。

编 者

CONTENTS 目录

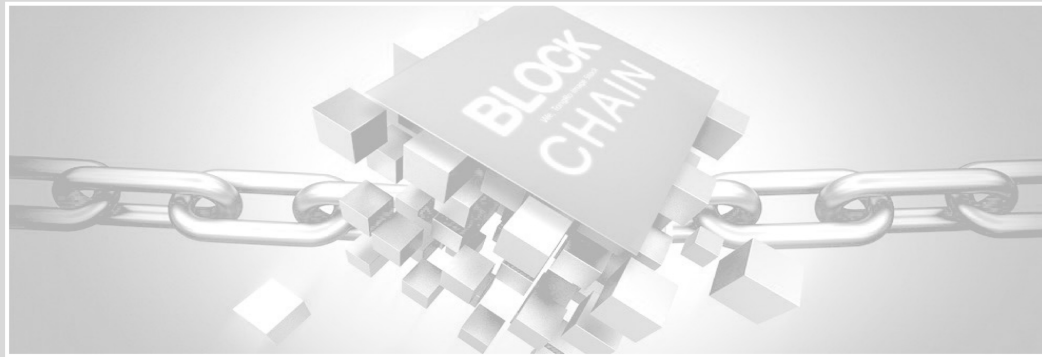
第 1 章 数据库基础知识	1
1.1 数据与数据处理.....	2
1.2 数据模型.....	8
1.3 关系数据库.....	14
1.4 数据库设计.....	19
1.5 本章小结.....	30
第 2 章 Access 2010 数据库系统概述	31
2.1 Access 的发展和特点.....	32
2.2 Access 2010 的启动和退出.....	35
2.3 Access 2010 的用户界面.....	37
2.4 Access 2010 的数据对象.....	41
2.5 数据库的建立.....	46
2.6 使用数据库.....	49
2.7 本章小结.....	53
第 3 章 表的建立和使用	54
3.1 创建表.....	55
3.2 数据类型.....	60
3.3 字段属性.....	62
3.4 修改数据表与数据表结构.....	64
3.5 建立表之间的关系.....	72
3.6 编辑数据表.....	79
3.7 设置数据表格式.....	88
3.8 本章小结.....	92
第 4 章 查询	93
4.1 查询概述.....	94
4.2 创建查询.....	96
4.3 编辑查询.....	100
4.4 参数查询.....	102
4.5 交叉表查询.....	104
4.6 操作查询.....	107



4.7	重复项、不匹配项查询	115
4.8	SQL 查询	119
4.9	本章小结	126
第 5 章	窗体	127
5.1	窗体的概述	128
5.2	创建窗体	132
5.3	使用“窗体向导”创建窗体	137
5.4	使用“设计视图”创建窗体	143
5.5	本章小结	171
第 6 章	报表	172
6.1	认识报表	173
6.2	创建报表	175
6.3	报表的分组及排序	181
6.4	报表预览及打印	185
6.5	本章小结	187
第 7 章	宏	188
7.1	宏概述	189
7.2	宏的基本操作	194
7.3	本章小结	208
第 8 章	库存管理系统	210
8.1	表设计	211
8.2	建立窗体	211
8.3	创建查询	219
8.4	创建报表	225
8.5	本章小结	225
参考文献	226

第1章

数据库基础知识



数据库(database)是按照数据结构来组织、存储和管理数据的建立在计算机存储设备上的仓库,它产生于距今60多年前,随着信息技术和市场的发展,特别是20世纪90年代以后,数据管理不再仅仅是存储和管理数据,而是转变成用户所需要的各种数据管理的方式。数据库有很多种类型,从最简单的存储有各种数据的表格到能够进行海量数据存储的大型数据库系统都在各个方面得到了广泛的应用。

在信息化社会,充分有效地管理和利用各类信息资源,是进行科学研究和决策管理的前提条件。数据库技术是管理信息系统、办公自动化系统、决策支持系统等各类信息系统的核心部分,是进行科学研究和决策管理的重要技术手段。



1.1 数据与数据处理

1.1.1 数据与信息

自 20 世纪 40 年代电子计算机问世以来,人们将十进制数变为计算机能够存储和处理的二进制数,随后将文字编码成位串形式,将图形、图像、声音等多媒体信息数字化。到目前为止,几乎所有信息都可以表示成计算机能识别的字符串和位串,为信息的迅速传播和处理提供了便利,现在的社会已成为信息化的社会。

数据(data)是对事实、概念或指令的一种表达形式,可由人工或自动化装置进行处理。数据经过解释并赋予一定的意义之后,便成为信息。具体来说,数据是指保存在存储介质上能够识别的物理符号。也可以说数据是用来记录信息的可识别的符号,它是信息的具体表现形式。数据不只是简单的数字,还包括文字、图形、图像、声音、物体的运动状态等。因此,数据都可以数字化后存入计算机。

软件中的数据有型(type)与值(value)之分,数据的型表示数据表示的类型,如字符型、日期型、整型等;数据的值表示符合给定型的值,如日期型“2018/10/28”。

同样的信息可以用不同的数据方式来描述,如可以用自然语言来描述这样一条有关学生的信息:王大锤,男,1996 年 12 月 30 日出生,2014 年入学,该学生的学号为 2014030401012,贵州遵义人。为了能在计算机中方便地存储和处理学生信息,可以用这样一个记录来描述上述的学生信息:(2014030401012,王大锤,1996-12-30,2014,贵州遵义)。

综上所述,数据是信息的符号表示或载体;信息是数据的内涵,是数据的语义解释。数据的含义称为数据的语义,数据与其语义是不可分的。例如,对于数据“2014”如果不和学生信息联系起来,还可以理解为 2014 个学生、2014 年出生等。因此,信息和数据这两个概念有时可以不加区别地使用,如信息处理也可以说成是数据处理。

信息(information)是指音讯、消息、通信系统传输和处理的对象,泛指人类社会传播的一切内容。人通过获得、识别自然界和社会的不同信息来区别不同事物,得以认识和改造世界。在一切通信和控制系统中,信息是一种普遍联系的形式。1948 年,数学家香农在题为“通信的数学理论”的论文中指出:“信息是用来消除随机不定性的东西。”

从狭义上来讲,信息就是符号排列的顺序。但作为一个概念,信息的定义呈现出多定义而又无定义的局面。一般来说,与信息这一概念密切相关的概念包括约束(constraint)、沟通(communicate)、控制、数据、形式、指令、知识、含义、精神刺激、模式、感知以及表达。信息是人们在适应外部世界并使这种适应反作用于外部世界过程中,同外部世界进行互相交换的内容和名称。

总而言之,信息是对现实世界中各种事物的存在方式、运动状态或事物间联系形式的反映的综合。例如,“贵阳大西门国贸商场服装在 11 月 11 日打 6.5 折”,这是一条有关商品打折的信息;“贵州财经大学商务学院招聘 10 名辅导员”,这是有关招聘的信息。信息是可以被感知和存储的,并且可以被加工、传递和再生。

数据与信息的关系:数据是具有某一目标定性、定量描述的原始资料,包括数字、文字、符号、图形、图像以及它们能够转换成的数据等形式。信息是向人们或机器提供关于现实世界新的事实的知识,是数据、消息中所包含的意义。

联系和区别:信息与数据是不可分离的。信息由与物理介质有关的数据表达,数据中所包含的意义就是信息。信息是对数据解释、运用与解算,数据是经过处理以后的数据,只有经过解释才有意义,才成为信息。就本质而言,数据是客观对象的表示,而信息则是数据内涵的意义,只有数据对实体行为产生影响时才成为信息。数据是记录下来的某种可以识别的符号,具有多种多样的形式,也可以加以转换,但其中包含的信息内容不会改变。即不随载体的物理设备形式的改变而改变。信息可以离开信息系统而独立存在,也可以离开信息系统的各个组成和阶段而独立存在;而数据的格式往往与计算机系统有关,并随载荷它的物理设备的形式而改变。

因此,数据是原始事实,而信息是数据处理的结果。不同知识、经验的人,对于同一数据的理解,可得到不同信息。

1.1.2 数据处理技术的发展

数据管理技术是对数据进行分类、组织、编码、输入、存储、检索、维护和输出的技术。数据管理的水平是和计算机硬件、软件的发展相适应的。随着计算机技术的发展和数据管理规模的扩大,数据管理技术的发展大致经历了3个阶段:人工管理阶段、文件系统阶段和数据库系统阶段。

1 人工管理阶段

20世纪50年代中期以前,人工管理阶段的计算机主要用于科学计算。这个阶段没有像磁盘这样的可直接存取的外存储设备,没有操作系统和数据管理方面的软件,数据完全由程序设计人员有针对性地设计程序进行管理,而且在编制应用程序时,要全面考虑数据的定义、存储结构、存取方法和输入方式等,这一阶段数据管理的特点如下:

- (1)数据不保存。在计算某一个项目时将数据输入,用完就退出,对数据不做长期保存。
- (2)应用程序管理数据。数据没有专门的软件进行管理,需要应用程序自己进行管理,应用程序中要规定数据的逻辑结构和设计物理结构(包括存储结构、存取方法、输入/输出方式等)。程序员必须花费大量精力于数据的物理布置上,因此程序员负担很重。
- (3)数据无独立性。数据是作为输入程序的组成部分,数据和程序同时提供给计算机运算使用。程序员对存储结构,存取方法及输入/输出的格式有绝对的控制权,要修改数据必须修改程序,因为数据无法独立存在。
- (4)数据不能共享。一组数据对应一个程序。不同应用的数据之间是相互独立、彼此无关的。数据不但高度冗余,而且不能共享。

人工管理阶段数据和程序的关系如图1-1所示。

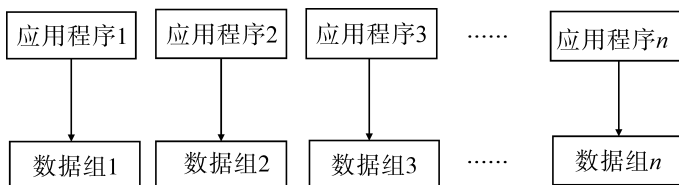


图 1-1 人工管理阶段程序和数据之间的关系

2 文件系统阶段

20 世纪 50 年代中期到 60 年代中期,计算机的应用范围从科学计算扩大到信息管理,硬件方面有了磁盘、磁鼓等能对数据进行直接存取的设备;软件方面出现了高级语言和操作系统。操作系统中有了专门的数据管理软件,即文件系统,这一阶段数据管理的特点如下:

(1)数据可以长期保存在外存储设备上。计算机需要对数据进行反复地查询、修改、增删等处理,数据长期存储在外存储设备上。

(2)由文件系统管理数据。数据放在相互独立的数据文件中,由文件系统统一管理。应用程序通过文件系统对存放在文件中的数据进行操作(如数据查询、修改、插入、删除)。程序员可以将更多的精力集中在算法上,而不必过多地考虑物理细节。

(3)数据共享性差,冗余度大。在文件系统中,文件是面向应用的,即一个文件和一个应用程序基本上是一一对应的,这就不便于数据的共享,同时增加了数据的冗余度,浪费了存储空间。

(4)数据独立性差。文件系统中的数据文件依赖于应用程序的存在而存在,程序和数据之间的依赖关系并未根本改变,数据与程序之间仍缺乏独立性。

文件系统阶段程序与数据之间的关系如图 1-2 所示。

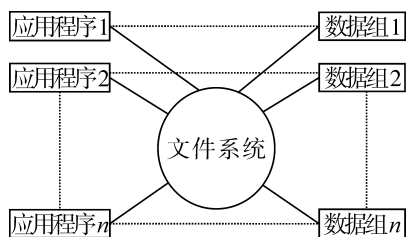


图 1-2 文件系统阶段程序与数据之间的关系

3 数据库系统阶段

20 世纪 60 年代后期至今,计算机的软硬件得到了进一步的发展,已出现了容量大、速度高的磁盘;需要管理的数据量急剧增加,联机实时处理的要求更多,数据管理技术得到了很大提高,应用程序和数据的联系通过数据库管理系统(Database Management System, DBMS)来实现。

数据库系统阶段对数据管理的特点如下:

(1)数据结构化。在数据库系统中数据按一定的模式组织与存储,即按一定的数据模型组织数据,最大限度地减小数据的冗余度。

(2)数据的共享性高,冗余度低,易扩充。数据库从整体的观点来看待和描述数据;面向整个系统,减小了数据的冗余,节约存储空间,缩短存取时间,避免数据之间的不相容和不一致;面向不同的应用,存取相应的数据库的子集。当需求改变或增加时,只要重新选择数据子集或者加上一部分数据,便可以满足更多更新的要求,保证了系统的易扩充性。

(3)数据独立性高。处理数据使用户所面对的是简单的逻辑结构,不涉及具体的物理存储结构,数据的存储和使用数据的程序彼此独立,数据存储结构的变化尽量不影响用户对程序的使用,用户对程序的修改也不要数据结构做大的改动。

(4)数据的安全性。设置数据的使用权限可以防止数据的非法使用,能防止数据的丢

失;当数据库被破坏时,系统可以把数据库恢复到可用的状态。

(5)数据的完整性。系统采用一些完整性检验以确定数据符合某些规则,保证数据库中的数据始终是正确的。

数据库能实现有组织地、动态地存储大量有关联的数据,能处理复杂的数据对象和对象之间的关系。

数据库系统阶段程序与数据之间的关系如图 1-3 所示。

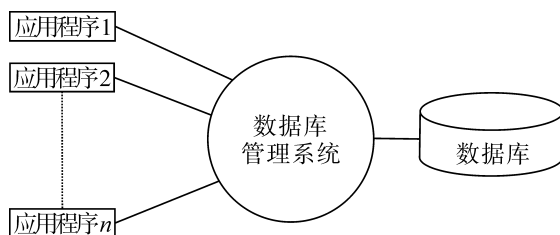


图 1-3 数据库系统阶段程序与数据之间的关系

1.1.3 数据库管理系统

数据库系统(Database System,DBS)是指在计算机系统中引入数据库后的系统构成,一般由数据库、数据库管理系统及其开发工具、数据库应用系统、数据库管理人员和用户、硬件平台和软件平台组成。通常将数据库系统简称为数据库。

数据库是依照某种数据模型组织起来并存放二级存储器中的数据集合。这种数据集合具有以下特点:尽可能不重复,以最优方式为某个特定组织的多种应用服务,其数据结构独立于使用它的应用程序,对数据的增、删、改、查由统一软件进行管理和控制。从发展的历史来看,数据库是数据管理的高级阶段,它是由文件管理系统发展起来的。

数据库是一个单位或是一个应用领域的通用数据处理系统,它存储的是属于企业和事业部门、团体和个人的有关数据的集合。数据库中的数据是从全局观点出发建立的,按一定的数据模型进行组织、描述和存储。其结构基于数据间的自然联系,从而可提供一切必要的存取路径,且数据不再针对某一应用,而是面向全组织,具有整体的结构化特征。

数据库中的数据是为众多用户所共享其信息而建立的,已经摆脱了具体程序的限制和制约。不同的用户可以按各自的用法使用数据库中的数据;多个用户可以同时共享数据库中的数据资源,即不同的用户可以同时存取数据库中的同一个数据。数据共享性不仅满足了各用户对信息内容的要求,同时也满足了各用户之间信息通信的要求。

数据库管理系统是指负责数据库存取、维护和管理系统软件,它是数据库系统的核心组成部分。它的基本功能有数据定义功能,数据存取功能,数据库的组织、存储和管理,数据库运行管理,数据库的建立和维护,通信功能和数据转换功能等。

1 数据定义功能

DBMS 提供数据定义语言(Data Definition Language,DDL),用户通过 DDL 可以方便地对数据库中的数据对象进行定义,包括数据结构、数据的完整性约束条件和访问控制条件等。

2 数据存取功能

DBMS 提供数据操纵语言(Data Manipulation Language,DML)实现对数据库的操作。

基本操作包括检索、插入、删除和修改。

3 数据库的组织、存储和管理

数据库中物理存在的数据包括两部分：一部分是元数据，即描述数据的数据，主要是数据库系统的三级模式，即外模式、模式、内模式，它们构成数据字典(Data Dictionary, DD)的主体，DD 由 DBMS 管理、使用；另一部分是原始数据，它们构成物理存在的数据库，DBMS 一般提供多种文件组织方法，供数据库设计人员选用。数据一旦按某种组织方法装入数据库，其后对它的检索和更新都由 DBMS 的专门程序完成。

4 数据库运行管理

数据库运行管理是数据库管理系统的核心部分。如前所述，数据库方法的最大优势在于允许多个用户并发地访问数据库，充分实现共享。相应地，DBMS 必须提供并发控制机制、访问控制机制和数据完整性约束机制，从而避免多个读写操作并发执行可能引起的冲突，数据失密或安全性、完整性被破坏等一系列问题。

5 数据库的建立和维护

DBMS 一般都要保存工作日志、运行记录等来用于恢复数据，一旦出现故障，使用这些历史和维护信息可将数据库恢复到一致状态。此外，当数据库性能下降，或系统软硬设备变化时也能重新组织或更新数据库。

6 通信功能和数据转换功能

DBMS 具有与操作系统的联机处理、分时处理、远程作业输入的相应接口，具有与网络中其他软件的通信功能。此外，DBMS 还具有与其他 DBMS 或文件系统的数据库转换功能。

数据库应用系统(Database Application System, DBAS)是由数据库系统、应用程序系统、用户组成的，具体包括数据库、数据库管理系统、数据库管理员、硬件平台、软件平台、应用软件、应用界面。数据库应用系统的 7 个部分以一定的逻辑层次结构方式组成一个有机的整体，它们的结构关系是应用系统、应用开发工具软件、数据库管理系统、操作系统、硬件。例如，以数据库为基础的财务管理系统、人事管理系统、图书管理系统等。无论是面向内部业务和管理的管理信息系统，还是面向外部，提供信息服务的开放式信息系统，从实现技术角度来看，都是以数据库为基础和核心的计算机应用系统。

从系统开发的角度来看，数据库应用系统具有结构特性和行为特性两个方面。

1 结构特性

结构特性与数据库状态有关，即与数据模型所反映的实体及其实体间的联系的静态特性有关。结构设计就是设计各级数据库模式。决定数据库系统的信息内容，由数据库设计来实现。

2 行为特性

行为特性与数据库状态转换有关，即改变实体及其特性的操作。它决定数据库系统的功能，是事物处理等应用程序的设计。

根据系统的结构和行为两方面的特性，系统设计开发分为两个部分：一部分是作为数据库应用系统核心和基石的数据库设计；另一部分是相应的数据库应用软件的设计开发。这两部分是紧密相关、相辅相成的，组成统一的数据库工程。

数据库管理员(Database Administrator, DBA),是从事管理和维护数据库管理系统的相关工作人员的统称,属于运维工程师的一个分支,主要负责业务数据库从设计、测试到部署交付的全生命周期管理。用户是指通过应用程序使用数据库的人员,最终用户无须自己编写应用程序。

数据库系统对硬件平台的要求是:计算机要有足够大的内存和外存,有较高的通道能力以保证数据库系统能正常运行。数据库系统的软件主要包括数据库管理系统、操作系统、具有与数据库接口的高级语言及其编译系统、以数据库管理系统为核心的应用开发工具、数据库应用系统。其中操作系统是支持 DBMS 运行的系统,没有合适的操作系统,DBMS 则无法正常运转。数据库管理系统包括 SQL Server、Oracle、Access 等,应用开发工具包括 C++、Java、Visual Basic、NET、Delphi等,通过这些工具,应用程序开发人员能够开发出合乎用户需求的应用系统。

1.1.4 数据库系统的三级模式结构

数据库系统的数据的高度独立性来自数据库系统的三级模式结构和二级映像功能。数据库的三级模式为模式、外模式和内模式,与之对应的是三级模式结构,即全局逻辑结构、局部逻辑结构和物理存储结构。数据库系统的三级模式结构,即全局逻辑结构、局部逻辑结构和物理存储结构。数据库系统的三级模式结构如图 1-4 所示。

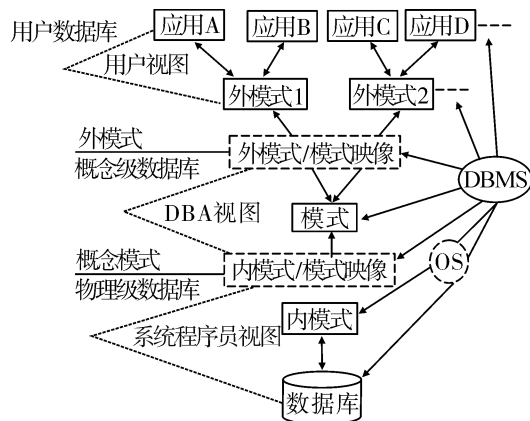


图 1-4 数据库系统的三级模式结构

模式又称为概念模式或逻辑模式,对应于概念级。它是数据库中全体数据的逻辑结构和特征的描述,是所有用户的公共数据视图。模式与具体的应用程序无关,也不涉及数据的物理存储内容。

模式是数据库数据在逻辑上的视图。数据库模式以某一种数据模式作为基础,综合地考虑了所有用户的需求,同时将这些需求统一起来形成了一个逻辑整体。它是由数据库管理系统提供的数据库模式描述语言来描述、定义,体现和反映了数据库系统的整体观,一个数据库只有一个模式。

外模式又称为子模式或用户模式,是用户和程序员看到并使用的局部数据逻辑结构和特征。不同的用户因其需求不同,看待数据的方式不同,因此不同用户的外模式的描述也不相同,一个数据库有若干个外模式。

内模式又称为存储模式或物理模式,是对数据物理结构和存储方式的描述,也是数据在存储介质上的保存方式。一个数据库只有一个内模式,是以数据模型为基础的。模式综合考虑所有用户的需求,并将其结合成有机逻辑整体。

在一个数据库系统中,只有唯一的数据库,因而作为定义、描述数据库存储结构的内模式和定义、描述数据库逻辑结构的模式,也是唯一的,但建立在数据库系统之上的应用则是非常广泛、多样的,所以对应的外模式不是也不可能是唯一的。

在数据库的三级模式中,逻辑模式是数据库的中心与关键,它独立于数据库的其他层次,所以在设计数据库模式结构时,应首先确定数据库的逻辑模式。

数据库系统的三级模式是对数据的三个抽象级别。它把数据库的具体组织留给了数据库管理系统管理,使用户能够从逻辑上处理数据,而不必关心数据在计算机中的具体表示方式和存储方式。为了实现三个抽象级别的联系和转换,数据库系统在三级模式中提供了两个层次的映像,即“模式/内模式映像”和“外模式/模式映像”。正是这两级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 模式/内模式映像

模式/内模式映像定义了数据的全局逻辑结构和存储结构的对应关系。当数据库的存储结构发生变化时,由数据库管理员对模式/内模式映像做相应的改变,可以使模式保持不变,从而应用程序不必改变。因此,保证了程序与数据的物理独立性。

2 外模式/模式映像

外模式/模式映像定义了某一个外模式与模式之间的对应关系。这些映像的定义通常包含在外模式当中。当模式改变时,由数据管理员对各个外模式/模式映像做相应的改变,就可以使外模式不变。应用程序是依据数据的外模式编写的,从而应用程序不必修改,保证了数据与程序的逻辑独立性。



1.2 数据模型

数据模型(data model)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。一般来讲,数据模型是严格定义的几个概念。这些概念精确地描述了系统的静态特征、动态特性和完整性约束条件。因此,数据模型通常由数据结构、数据操作及完整性约束条件三部分组成。

1.2.1 数据模型的组成要素

数据模型的组成要素有数据结构、数据操作及完整性约束条件。

数据结构是指数据的组织形式和数据之间的联系,是所研究对象类型的集合,是对系统静态特性的描述。在关系模型中就是采用的关系模型了,就是“二维表”的形式。如果用 D 表示数据,用 R 表示数据对象之间存在的关系集合,则将 $DS=(D,R)$ 称为数据结构。

例如,设有一个电话号码簿,它记录了 n 个人的名字和相应的电话号码。为了方便地查

找某人的电话号码,将人名和号码按字典顺序排列,并在名字的后面跟随着对应的电话号码。这样,若要查找某人的电话号码(假定他的名字的第一个字母是Z),那么只需查找以Z开头的那些名字就可以了。在这个案例中,数据的集合E就是人名和电话号码,它们之间的联系R就是按字典顺序的排列,其相应的数据结构就是 $DS=(E,R)$,即一个数组。

数据结构分为两类:一类是与数据类型、内容等有关的对象;另一类是与数据之间关系有关的对象。

数据结构是刻画一个数据模型性质最重要的方面,在数据库系统中按照其数据结构的类型来命名数据模型。常用的数据结构有3种:层次结构、网状结构和关系结构。这3种结构的数据模型分别命名为层次模型、网状模型、关系模型。

数据操作是指对数据库中各种对象(型)的实例(值)允许执行的操作的集合,包括操作及有关的操作规则。数据库的操作主要有检索和更新这两大类操作。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)及实现操作的语言。简单来说,就是对数据的一些操作,包括查询、删除、更新、插入等。

完整性约束是对所存数据的约束规则,有实体完整性,参照完整性等,就是取值唯一、不能为空等一系列操作。数据约束主要描述数据结构内数据间的语法、语义联系,它们之间的制约与依存关系,以及数据动态变化的规则,以保证数据的正确、有效与相容。例如,对于教学管理系统中教师的“教师编号”属性取值不能出现重复值,教师信息中的“系编号”属性的值必须取自院系信息中的某个具体的系编号。

1.2.2 数据模型的类型

数据结构是刻画一个数据模型性质最重要的方面,在数据库系统中按照其数据结构的类型来命名数据模型。常用的数据结构有3种:层次结构、网状结构和关系结构,这3种结构的数据模型分别命名为层次模型、网状模型、关系模型。数据模型按不同的应用层次分成3种类型:概念模型、逻辑模型、物理模型。

1 概念模型

1) 概念模型的基本概念

概念模型又称为信息模型,它是一种面向现实世界、面向用户的数据模型;它与具体的数据库管理系统无关,与具体的计算机平台无关。概念模型着重于对客观世界复杂事物的结构描述以及对它们之间的内在联系的刻画,E-R模型就是概念模型代表,它以实体—联系(Entity-Relationship,E-R)理论为基础,并对这一理论进行了扩充。它从用户的观点出发对信息进行建模,主要用于数据库的概念级设计。

(1)实体:客观存在并可相互区别的事物称为实体。一个实体是现实世界客观存在的一个事物,可以是一个具体的事物,如一个学生、一张桌子等,也可以是抽象的事物,如一个想法、一个计划或一个工程项目等。实体由它们自己的属性值表示其特征。

(2)属性:实体所具有的某一特性。一个实体可由若干个属性来描述。例如,学生实体可以用学号、姓名、性别、出生日期、民族等属性来描述。

(3)域:一个属性的取值范围。例如,学生的性别只能取“男”或“女”这两个值。

(4)码:唯一标识实体的属性集称为码。例如,学号是学生实体的码。

(5)实体型:用实体名及其属性名集合来描述和刻画同类实体,称为实体型。例如,学生(学号、姓名、性别、出生年月、政治面貌)是一个实体型。

(6)实体集:同一类型实体的集合称为实体集。例如,某校的全体学生就是一个实体集。

(7)联系:实体之间的关联称为联系。实体的联系分为实体内部的联系和实体之间的联系。实体内部的联系通常是组成实体的各属性之间的联系。实体之间的联系是指不同实体集之间的联系。两实体集之间的联系主要有以下3类。

①一对一联系(1:1):如果对于实体集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具有一对一联系,记为1:1。例如,实体集班级与实体集班长之间存在一对一联系,意味着一个班长负责一个班,而且一个班也只有一个班长。

②一对多联系(1:n):如果对于实体集A中的每一个实体,实体集B中有 $n(n \geq 0)$ 个实体与之联系。反之,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具有一对多联系,记为1:n。例如,在本书涉及的教学管理系统中,实体集院系与实体集教师就是一对多联系。因为一个院系中有若干名教师,而每名教师只属于一个院系。

③多对多联系($m:n$):如果对于实体集A中的每一个实体,实体集B中有 $n(n \geq 0)$ 个实体与之联系。反之,对于实体集B中的每一个实体,实体集A中也有 $m(m \geq 0)$ 个实体与之联系,则称实体集A与实体集B具有多对多联系,记为 $m:n$ 。例如,在教学管理系统中,实体集课程与实体集学生之间的联系是多对多联系($m:n$)。因为一个课程同时有若干名学生选修,而一个学生可以同时选修多门课程。

以上3类联系如图1-5所示。

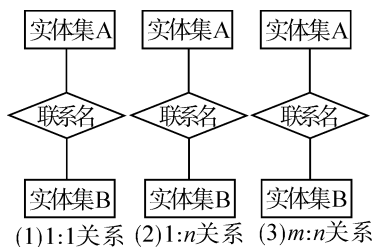


图1-5 两个实体集之间的3类联系

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。实体集之间这种一对一、一对多和多对多的联系不仅存在于两个实体集之间,也存在于两个以上的实体集之间。

实体集内部也有类似的联系。例如,对于教学管理系统中的教师实体集内部有领导与被领导的关系,即某一教师(如教研室主任)“领导”若干名教师,而一个教师仅被另一个教师直接领导。

2) 实体-联系模型

概念模型的表示方法很多,其中最为著名的是1976年P. P. S. Chen提出的实体-联系方法(Entity-Relationship Approach)。该方法用E-R图来描述现实世界的概念模型,称为实体-联系模型,简称E-R模型。E-R图提供了表示实体集、属性和联系的方法。用来描述实体集之间联系的图就是E-R图,如图1-5所示。

在E-R图中,用矩形框、椭圆形框、菱形框分别表示实体集、属性、联系。

(1) 矩形框:表示实体集,矩形框内写明实体名。

(2) 椭圆形框:表示某实体的属性或者实体间联系的属性,用无向边将其与相应的实体或联系连接起来。

(3) 菱形框:表示实体集之间的联系,菱形框内写上关系名,用无向边将菱形分别与有关实体集相连接,在无向边旁标上关系的类型。若实体集之间的关系也具有属性,则把属性和菱形也用无向边连接上。

教学管理系统中班级实体,如图1-6所示;实体集教师内部的联系,如图1-7所示;教学管理系统中院系、教师、专业、学生、课程、班级这6个实体之间的联系,如图1-8所示。

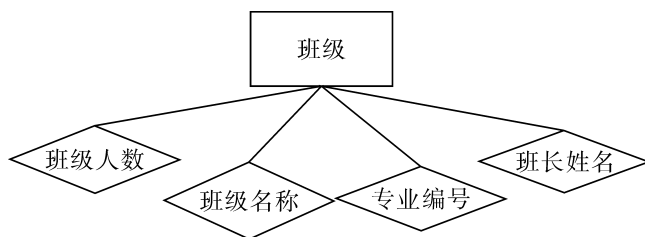


图 1-6 班级实体及属性

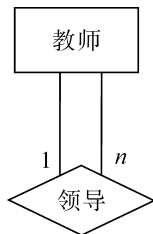


图 1-7 教师实体集内部的联系

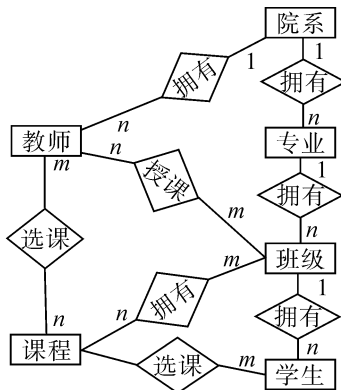


图 1-8 教学管理系统实体集及其联系图

2 层次模型

层次模型数据库管理系统是紧随网状数据库模型而出现的。现实世界中很多事物是按层次组织起来的。层次数据模型的提出,首先是为了模拟这种按层次组织起来的事物。层次数据库也是按记录来存取数据的。层次数据模型中最基本的数据关系是基本层次关系,它代表两个记录型之间一对多的关系,也叫作双亲子女关系(PCR)。数据库中有且仅有的一个记录型无双亲,称为根节点。其他记录型有且仅有的一个双亲。在层次模型中从一个节点到其双亲的映射是唯一的,所以对每一个记录型(除根节点外)只需要指出它的双亲,就可以表示出层次模型的整体结构。层次模型是树状的,最著名的层次数据库系统是1969年IBM公司的IMS(Information Management System),这是IBM公司研制的最早的大型数据库系统程序产品。

层次模型是数据库系统中最早出现的数据模型,采用树型结构来表示实体及实体间的联系的模型。这种模型体现出实体之间只有简单的层次关系,其特点是:(1)有且仅有一个根结点;(2)其他结点仅有一个根结点或父结点。结点之间的关系是父结点与子结点的关系,即一对多的关系。例如,一个学院的数据库结构中,一个学院有多个系,但每个系只对应一个学院;每个系下面有多个教研室,但每个教研室只对应一个系,如图1-9所示。

层次模型的优点:层次模型本身比较简单;实体间联系是固定的;层次模型提供了良好的完整性支持。层次模型的缺点:现实世界中很多联系是非层次性的,只能通过引入冗余数据或创建非自然的数据组织来解决;对插入和删除操作的限制比较多;查询子结点必须通过父结点;结构严密,层次命令趋于程序化。

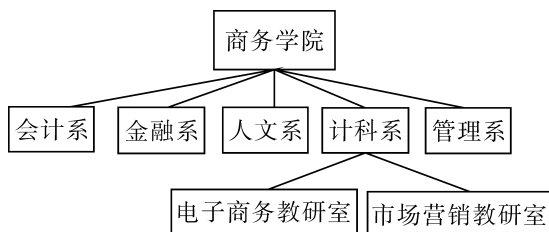


图 1-9 层次模型

3 网状模型

网状模型以网状结构表示实体与实体之间的联系。网中的每一个结点代表一个记录类型。网状模型可以表示多个从属关系的联系,也可以表示数据间的交叉关系。网状模型的特点:允许一个以上的结点无双亲;一个结点可以有多个的双亲。网状模型可以方便地表示各种类型的联系,但结构复杂,数据处理比较困难。如图1-10所示的是学生选课情况的网状模型。

网状模型的优点:能更为直接地描述现实世界,具有良好的性能,存取效率较高。网状模型的缺点:结构比较复杂,用户不容易使用。由于记录之间联系是通过存取路径实现的,应用程序在访问数据时必须选择适当的存取路径,因此用户必须了解系统结构的细节,这加

重了编写应用程序的负担。

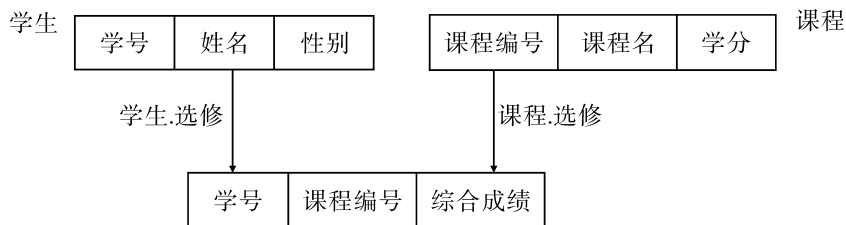


图 1-10 网状模型

4 关系模型

关系实际上就是关系模式在某一时刻的状态或内容。也就是说,关系模式是型,关系是它的值。关系模式是静态的、稳定的,而关系是动态的、随时间不断变化的,因为关系操作在不断地更新着数据库中的数据。关系模型是 1970 年由 E. F. Codd 提出的。它和层次、网状模型相比,有以下特点:(1)数据结构简单(二维表格);(2)扎实的理论基础:关系运算理论、关系模式设计理论。

关系模型以二维表结构表示实体以及实体之间的联系,是一个“二维表框架”组成的集合。关系模型是目前最流行的数据库模型。支持关系模型的数据库管理系统称为关系数据库管理系统,Access 就是一种关系数据库管理系统。

表 1-1 给出了教学管理系统中的班级实体关系。无论是实体还是实体之间的联系,在关系模型里面都用一张二维表来表示。

表 1-1 班级关系

班级名称	专业编号	班级人数	班长姓名
电商 1801 班	1	60	张三
信管 1801 班	2	61	李四
房开 1801 班	3	59	王二
市营 1801 班	4	58	小明
电商 1701 班	5	59	小红
信管 1701 班	6	60	小刘
市营 1701 班	7	61	小李

关系模型较层次模型和网状模型具有以下优点:关系模型以严格的数学概念作为基石,数据结构比较简单,具有很高的数据独立性。

所以,关系模型诞生以后发展迅速,深受用户的喜爱。当然,关系模型也有缺点。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。



1.3 关系数据库

针对关系数据库体系而言,其属于对关系数据模型予以充分支持的数据库体系。关系数据库借助于数学方式对数据库内的相关数据信息进行处理。首次主张将该种方式应用在数据处理的为1962年CODASYL所撰写的“信息代数”,紧接着,David Child于1968年通过7090机完成集合论数据结构,然而全面且严格地对关系模型予以阐述的为来自美国IBM公司的E. F. Codd。在1970年,E. F. Codd发表多篇相关论文,从而打下了关系数据库的相应理论根基。

1.3.1 关系数据库的逻辑语言

关系数据结构极其简单。其立足于关系数据模型,其中现实世界内的相应实体及不同实体彼此的关联都通过关系进行反映。基于逻辑以及用户的看法而言,其关系即为二维表。常用的关系术语如下:

(1)关系:一个关系就是一张二维表。

(2)属性:表内任意一列对应于一个属性(同样被叫作字段),比如学生表内的学号即为一个属性。

(3)元组:表内任一行对应于一个元组(还被叫作记录),其属于一组字段相应信息的集合。

(4)域:属性所对应的取值区间被叫作域。比如学生选课表所对应综合成绩的取值区间为0~100内的浮点数。

(5)关系模式:关系名与关系内的属性集合组成对应的关系模式,任一关系模式与一个关系的相应结构相对应。关系模式的格式为:关系名(属性名1,属性名2,属性名3,...,属性名n)。例如,班级表的关系模式为:班级(班级叫法,专业对应编号,班级所含人数,班长的姓名)。

(6)主关键字:主关键字其还被叫作主键,属于唯一标识表内所记录的相应字段或者字段的相应组合。比方学生表内的对应学号能够被视作主关键字,它可以对表内的所有记录进行唯一标识,也就是表内无法产生两个一致的学号。学生在选课表内的对应学号与课程编号两个字段的相应组合属于主关键字。

(7)候选码:倘若某字段的相应值可以对表内某一记录进行唯一标识,则此字段即被叫作候选码。在某一关系内,极易存在不同候选码。毋庸置疑,候选码同样能够为不同字段相应组合。

(8)外部关键字:外部关键字还被称为外键,用于和其他关系展开连接的相应字段,同时属于其他关系内的主关键字。例如,教学管理体系内,系编号于专业表内属于外键,然而于院系表内则属于主键,这源于专业表与院系表借助于系编号字段彼此关联。

1.3.2 关系的性质

关系和二维表及常规数据文件存在相似的地方,同样也存在不同之处。关系属于一类标准化的二维表,被视为关系的二维表应当符合以下属性:

(1)每一个属性是不可分解的。此属于关系数据库面向关系的最主要的限定,规定关系的所有分量应当属于一个无法划分的数据项,即应当避免表内还存在表。

(2)所有关系模式内属性的相应数据种类与属性的相应个数是不变的,同时所有属性应当命名,于相同关系模式内,属性名应当存在差异。

(3)所有关系只存在一类记录种类,也就是一类关系模式。

(4)针对关系而言,其元组对应的顺序(也就是行所对应的顺序)不存在影响,也就是行的相应次序能够随意展开交换。

(5)针对关系而言,其属性对应的顺序(也就是列所对应的顺序)不存在影响,能够随意进行交换,然而展开交换期间必须为整体展开交换,其中属性名与属性值应当视作整列同时展开交换。

(6)元组不能够重复,也就是于某一关系内随意两个元组应当存在差异。

1.3.3 关系模型的完整性约束

针对数据完整性而言,其通过完整性规则进行定义,其中关系模型所对应的完整性规则属于面向关系的某类制约因素。在关系模型内,能够存在3种完整性约束,分别为用户定义的完整性;实体完整性;参照完整性。围绕关系数据库实施修改、插入以及删除操作期间,应当符合以下完整性规则:

(1)用户定义的完整性规则:针对此规则而言,其为用户按照详细应用的相应语义规定,借助于DBMS给予的定义和检验此种完整性规则的相应制度,用户自身所定义的完整性规则。

(2)实体完整性规则(entity integrity rule):关系内的相应元组于组成主键的属性层面避免存在空值。

(3)参照完整性规则(reference integrity rule):外键的值避免参照没有的对应表的相应主键值,或者外键为空值。

1.3.4 关系代数

关系代数属于将关系视作运算对象的一组高级运算的相应集合。基于关系所对应的基本运算展开划分,可分成两种,即常规的集合运算与专门的关系运算。

1 常规的集合运算

针对常规的集合运算而言,其涉及并、交、差以及笛卡儿积。

1) 并

假定关系S与关系R存在的属性全为 n 个,同时对应的属性取于相同域,那么关系S与关系R的并为通过属于S或者属于R的元组所构成的集合,将其表示成 $R \cup S$ 。

【例 1-1】表 1-2 和表 1-3 内对关系 R 与 S 予以确立,它们包含一致的属性。

表 1-2 关系 R

K	G	F
k1	g1	f1
k2	g2	f2
k3	g3	f3

表 1-3 关系 S

K	G	F
k2	g2	f3
k1	g2	f1

则 $R \cup S$ 的结果如表 1-4 所示。

表 1-4 $R \cup S$

K	G	F
k1	g2	f1
k2	g1	f1
k2	g2	f2
k2	g2	f3

2) 交

假定关系 R 与关系 S 存在的属性数量全为 n 个,同时对应的属性取于相同域,那么关系 R 和关系 S 所对应的交为通过属于 R 与属于 S 的相应元组构成的集合,将其表示成 $R \cap S$ 。

【例 1-2】表 1-2 和表 1-3 内对关系 R 与 S 予以确立,它们包含一致的属性。则 $R \cap S$ 的结果如表 1-5 所示。

表 1-5 $R \cap S$

K	G	F
k1	g2	f1

3) 差

假定关系 R 与关系 S 包含的属性数量全为 n 个,同时对应的属性取于相同域,那么关系 R 和关系 S 所对应的差为通过属于 R 却不属于 S 的相应元组构成的集合,将其表示成 $R - S$ 。

【例 1-3】表 1-2 和表 1-3 内对关系 R 与 S 予以确立,它们包含一致的属性。则 $R - S$ 的结果如表 1-6 所示。

表 1-6 $R - S$

K	G	F
k2	g1	f1
k2	g2	f2

4) 笛卡儿积

假定关系 R 与 S 所包含的属性数量依次为 r 与 s, 那么关系 R 和关系 S 所对应的笛卡儿积为一个包含属性数量达到 $(r+s)$ 个的相应元组集合, 所有元组的前 r 个属性值来自于 R 的某一元组, 而后 s 个属性值则来自于 S 的某一元组, 记为 $R \times S$ 。若 R 有 m 个元组, S 有 n 个元组, 则 $R \times S$ 有 $m \times n$ 个元组。

【例 1-4】表 1-2 和表 1-3 内对关系 R 与 S 予以确立, 它们包含一致的属性。则 $R \times S$ 的结果如表 1-7 所示。

表 1-7 $R \times S$

R. K	R. G	R. F	S. K	S. G	S. F
k1	g2	f1	k2	g2	f3
k1	g2	f1	k1	g2	f1
k2	g1	f1	k2	g2	f3
k2	g1	f1	k1	g2	f1
k2	g2	f2	k2	g2	f3
k2	g2	f2	k1	g2	f1

说明: 由于 R 与 S 具有一致的属性名, 因此属性名之前加上对应的关系名, 如 R. A, S. A。

2 专门的关系运算

专门的关系运算包括: 选择、投影、连接、除等。本书不讨论“除”运算。

1) 选择

针对选择运算而言, 就是在其关系内找到符合条件的相应元组。值得注意的是, 条件属于逻辑表达式, 而运算之后的结果则属于条件为真的相应元组。例如, 自学生表内对“王”姓学生展开查找即为选择运算。通常而言, 选择运算将对记录的相应个数予以调整, 然而未对属性个数予以调整。

2) 投影

针对投影运算而言, 其为自关系模式内确立多个属性组成全新关系。例如, 自学生表对学生状况展开查询, 其中查询结果仅涉及姓名与学号属性, 此即为投影运算。一般而言, 投影运算未对元组相应个数予以调整, 仅调整属性的相应个数, 同样极易对属性的显示次序予以变更。

3) 连接

针对连接操作而言, 其为自笛卡儿积内挑选属性间符合相应要求的元组。于连接运算内, 依据字段值对应相等视作条件展开的连接操作为等值连接。对于自然连接来说, 其为对重复属性予以去除的等值连接。自然连接是最常用的连接运算。

【例 1-5】关系 R 和 S 如表 1-8 和表 1-9 所示。符合要求 $B < C$ 的相应连接结果如表 1-10 所示, 其中等值连接 $(R. A = S. A)$ 的相应结果如表 1-11 所示, 自然连接的相应结果如表 1-12 所示。

表 1-8 关系 R

K	G
k1	101
k2	201

表 1-9 关系 S

K	F	E
k1	150	120
k1	140	100
k3	110	130

表 1-10 关系 R 与关系 S 的连接运算

R. K	G	S. K	F	E
k1	101	k1	150	120

表 1-11 关系 R 与关系 S 的等值连接运算

R. K	G	S. K	F	E
k1	101	k1	150	120
k1	101	k2	140	100

表 1-12 关系 R 与关系 S 的自然连接运算

K	G	F	E
k1	101	150	120
k1	101	140	100

说明:自然连接时,在没指明连接条件的情况下,一般是按两个关系中相同属性进行的等值连接,然后去掉重复的属性。

【例 1-6】院系和专业为教学管理系统中的两个关系,如表 1-13 和表 1-14 所示。院系和专业的自然连接的结果如表 1-15 所示。

表 1-13 院系

系编号	系名称	系主任	系网址	联系电话
01	计科系	张三	http://jkkkkkx.cn	18215685568
02	管理系	李四		18265784585
03	会计系	王五	http://jkkkkkx.cn	18284656565
04	金融系	赵六		18296975489

表 1-14 专业

系编号	系名称	系编号
01	计科系	01
02	管理系	02
03	会计系	03
04	金融系	04

表 1-15 院系和专业的自然连接运算

系编号	系名称	系主任	系网址	联系电话	专业编号	专业名称
01	计科系	张三	http://jkkkkkx.cn	18215685568	01	计算机科学与技术
02	管理系	李四		18265784585	02	工商管理
03	会计系	王五	http://jkkkkkx.cn	18284656565	03	会计学
04	金融系	赵六		18296975489	04	金融学



1.4 数据库设计

数据库设计是对数据库及其运用体系予以构建的关键技术。

1.4.1 数据库设计的概述

数据库设计属于在相应平台约束下,按照信息需求和处理需求对性能极佳的数据模式予以设计。针对信息需求而言,其着重为用户数据对象的相应数据与其结构,它对数据库所涉及的静态需求予以体现;针对处理需求而言,其意为用户对象的相应行为与动作,其对数据库所涉及的动态需求予以体现。

在数据库设计方面存在以下方式:(1)将处理需求视作根本,同时顾及信息需求的针对过程的方式(process-oriented approach);(2)将信息需求视作根本,同时顾及处理需求的针对数据的方式(data-oriented approach)。现阶段,着重对第二类方式予以运用。

针对数据库设计,其通常对生命周期(life cycle)方式予以运用。此方式把数据库运用体系的开发细分成下列目标独立的不同阶段:(1)需求分析阶段;(2)设计概念结构;(3)设计逻辑结构;(4)数据库物理设计;(5)数据库实现;(6)数据库运作与维护。一般所论及的数据库设计意为前4个阶段。

1.4.2 需求分析

需求分析也就是分析用户的需求。需求分析属于软件开发期间最核心的步骤,围绕软件需求展开充分理解属于软件开发相关工作取得最终成功的先决因素,不管我们设计出何等优异的数据库,倘若无法迎合用户的相应需求,则会让用户产生失望,进而使开发者产生烦恼。针对需求分析而言,其核心为针对用户在数据管理方面的信息要求、处理要求以及安

全性和完整性要求展开调查、采集以及探究。

调查的重点是“数据”和“处理”。借助于调查的手段,掌握用户面向数据库的相应信息需求、处理需求及安全性和完整性需求。信息需求其意为用户需要自数据库内获取的信息的相应内容和属性,进而导出于数据库内需要对什么数据进行储存,此即为数据需求;处理需求即用户应当实现何种处理功能,面向处理的相应时长存在何种要求;安全性需求是对数据库应用体系内各个用户操作数据库的状况展开描述,其中完整性需求则是针对数据彼此的关联及数据所对应的取值区间要求进行定义。

【例 1-7】开发教学管理系统的用户需求分析。

面向某一本科院校所对应的教学工作展开调查,需要此教学管理系统可以针对不同系及不同专业的学生选课状况、老师授课状况及课程等主要状况展开记录。具体情况如下:

某一本科院校存在多个系,所有系都存在多个专业,所有专业存在多门课程,各个系或者各个专业极易存在一样的课程,比如信息科学和工程系所属的计算机科学与技术专业、电子科学和技术专业都对 C 语言程序设计相关课程予以设立,而外语系与文史系的不同专业全对大学计算机基础课程予以设置;所有专业都存在多个班级,所有班级都存在多名学生,所有学生能够修不少于一门课程,所有课程都被不少于一名学生修;所有系都存在多名教师,所有教师能够对不少于一门课程进行讲授,所有课程还能够被不少于一名教师进行讲授;每名教师可以给多个班级授课,每个班级有多名授课教师。

(1) 新任教师个人信息的录入。此信息应包含教师的个人资料,如教师编号等。

(2) 教师个人信息的修改。要求系统可以对教师个人信息的各项内容进行修改并保存。

(3) 教师记录的删除。如果教师离开该学校,要求系统能够删除数据库中该教师记录。

(4) 教师信息的查询。由于教师较多,要求系统可以进行查询。

(5) 教师能够面向所教课程学生的相应成绩展开管理,比方对成绩予以录入或者修改等。

(6) 学生个人信息的录入。此信息应包含学生的个人资料,如学号、姓名、性别、出生日期、政治面貌、家庭住址等。

(7) 学生个人信息的修改。要求系统可以对学生个人信息的各项内容进行修改并保存。

(8) 学生记录的删除。如果学生毕业或退学,要求系统能够删除数据库中的学生相应记录。

学生信息的查询主要有以下几点:

(1) 学生能够对自身某门课程的相应成绩进行查询,但杜绝进行修改。

(2) 课程相关信息的补加、删减、修改及查询等。

(3) 院系相关信息的补加、删减、修改及查询等。

(4) 专业相关信息的补加、删减、修改及查询等。

(5) 班级相关信息的补加、删减、修改及查询等。

教学管理者能够按照学生与教师相关信息及课程状况等实施录入、改正以及删减,面向系统内全部数据每个教学管理者可以实施任意操作。

在登录有关系统期间,应当实施密码与身份认证,倘若认证获得通过,则可以登进系统。

熟知用户相应需求,应当深入研究与表达用户的相应需求。表达所具有的含义为形成规格说明书等诸多相关文档。规格说明是将探析的相应结果准确且全面地进行表达。系统

分析相关工作者借助于调查探究之后打造起相应模型,立足于此,慢慢获得系统需求说明书,这同样属于需求分析的阶段性成就。

通常借助于下列方式对用户需求进行探究与表达:针对对象方式与结构化探析方式(Structured Analysis,SA)。对于结构化探析方式而言,其主要思想即为将系统从上至下进行深入分解,不断求精,主要原则为抽象和分解。特别为针对大型运用体系,控制体系的复杂性的主要方式即为分解。按照这种方法,无论系统多么大,总可以有计划地把它分解为足够小的子系统。

结构化探析法实际上是一个建模活动,用结构化分析建模获得一个分析模型,它的元素包含有描述系统的数据字典、各种图以及规格。在探究建模内,根本为数据字典,针对数据字典存在下列子模型,也就是行为模型、数据模型以及功能模型。通常而言,数据模型通过E-R图展开描述,功能模型则通过数据流图展开描述,而行为模型则通过状态转换图展开相应描述。

结构化探析法通过数据流图对数据与解决过程的相应关联进行表达,通过数据字典围绕系统内数据展开全面描述,属于不同类别数据属性的相应清单。

针对数据字典而言,其构建于需求分析时期,在数据库设计期间还将持续进行修改与优化。其中,数据项是数据的最小单位;数据结构属于多个数据项有意义的相应集合;数据流能够为数据结构,同样能够为数据项,针对某一解决过程的相应输入与输出进行反映;数据存储属于处理期间所存取的相应数据,比方计算机文件等;处理过程即处理的过程。

1.4.3 概念结构设计

概念结构设计属于全部数据库设计的核心。众所周知,概念结构设计所普遍运用的工具为E-R图,此项工作涉及以下方面:

- (1)确定实体。
- (2)确定实体的属性。
- (3)确定实体的码。
- (4)明确实体之间的联系以及联系种类。
- (5)画出表示概念模型的E-R图。
- (6)确定属性之间的依赖关系。

1 概念设计的策略

概念结构设计着重对下列对策予以运用:

- (1)自顶向下。首先对全局概念结构的相应框架进行定义,接着不断进行细化。
- (2)自底向上。先对所有局部运用的相应概念结构展开定义,接着遵照相应规则将它们进行集成,从而获得全局概念模型。
- (3)由里向外。首先对最关键的的概念结构进行定义,接着朝外进行扩充,直到总体概念结构。
- (4)混合策略。充分融合自底朝上与自顶朝下的方式,首先通过自顶朝下的方式对概念结构的相应框架进行设计,将其视作骨架集成通过自底朝上对策中设计的所有局部概念结构。该部分着重探讨自底朝上的设计对策,这同样属于现阶段最普遍的一类设计方式。

2 设计步骤

遵循自顶朝下的需求探究步骤,按照自底朝上的设计概念结构对策,借助于 E-R 方式的概念结构设计具有下列步骤。

第 1 步:按照需求分析的相应结果(包括数据流图等)围绕实际世界的有关数据实施抽象,针对不同局部视图展开设计,也就是 E-R 图。内容涉及挑选局部应用,也就是对不同局部 E-R 模型的相应范围进行具体化,针对不同局部 E-R 模型实体、联系以及实体与联系的对应属性展开定义。

第 2 步:设计全局 E-R 模型。内容涉及把全部局部 E-R 图集合为全局 E-R 图。

第 3 步:优化全局 E-R 模型。

3 设计局部 E-R 图

针对局部 E-R 图展开设计应当首先对局部应用予以挑选。一个局部应用实际上对应着应用系统中的一个子系统,也就是一个子需求,局部 E-R 图对应着一个应用系统中的一个局部应用。需要说明的是,按照信息理论的研究结果,一个局部应用中的实体数不能超过 9 个,不然要将局部应用对应的子系统进行细分。

所有局部应用通常与一组数据流图相对应,其中局部应用所运用的相关数据能够自对应的数据字典内进行提取。依托数据流图,对局部应用予以具体化之后,借助于抽象制度能够面向局部应用内的相应数据展开分类及组织(聚集),产生实体及实体的对应属性,以面向局部应用内的相关数据展开分类及组织(聚集),产生实体与实体所对应的属性,针对实体的码进行标记,具体化不同实体彼此的联系,设计相应的局部 E-R 图。实际而言,在需求分析时期获得的数据字典与数据流图内的相应数据项就反映出实体及实体属性等的相应划分。当然还需要对需求分析中得到的数据信息进行必要的调整。

在调整中应遵循的准则是,在实际世界内,倘若事物可以进行“属性”处理,则无须视为实体进行对待。如此有助于对 E-R 图进行简化,尤其能够降低数据库应用体系内的表对象,让系统获得进一步改进。

那么,怎么来确定某个事物是实体还是属性呢?一般来说,满足下面两个准则的事物都可以作为属性来看待。

准则一:倘若事物被视作“属性”,那么无法再包含需进行描述的相应属性。也就是说,作为属性的事物不能包含其他属性。也可以说,作为属性的事物完全可以用该事物的名字来描绘该属性名。

准则二:属性不能与其他实体具有联系,联系只能是实体之间的联系。

例如,教学管理系统中的某个教师的职称。职称是教师实体的一个属性,这是因为我们只需要知道该教师是什么职称就可以了;然而倘若差异的职称,平均课时费存在区别,则职称将具有课时费的相应属性,此时应当把职称独立视作一个实体。由于我们的教学管理系统不需要具备给教师算课时费的功能,所以,职称在这里只是教师的一个属性。即事物属于实体还是属性,这和事物所处应用体系的功能存在紧密关联。

例如,针对教学管理体系内的相应专业,一名学生与一个专业相对应,专业能够为学生实体的相应属性。但由于专业还要与院系实体发生联系,即一个院系对应着多个专业,则根据准则二,专业作为一个实体。

4 设计全局 E-R 图

对不同局部 E-R 图予以设计之后,需要把所有局部 E-R 图集合为全局 E-R 图,这就是集成。针对集成而言,其存在不同形式,一类为把不同局部 E-R 图一次实现集成;第二类为逐步进行集成,通过累积的形式对两个 E-R 图予以一次集成。

不管针对何种形式,在集成 E-R 图期间都要求两步。首先,对 E-R 图予以合并,进而形成初始的 E-R 图;其次,进行修改和重构,从而形成基本 E-R 图。

(1) 合并 E-R 图,生成初步的 E-R 图。

在把局部 E-R 图做成全局 E-R 图期间,要求对不同局部 E-R 图所形成的对立予以除去。恰当地对冲突予以除去属于第一步的核心。

各 E-R 图之间的冲突主要有 3 类:涉及属性、命名及结构 3 个方面的冲突。

① 针对属性冲突,其涉及属性取值单位冲突与属性域冲突。

属性域冲突其意为相同属性于差异局部 E-R 图内数据种类、取值区间以及取值集合存在差异。属性取值单位冲突,其意为相同属性于差异局部 E-R 图内包含差异单位。

② 针对命名冲突而言,其涉及以下状况:

首先为同名异义,即意义存在差异的对象于差异 E-R 图内包含一致的名字。其次为异名同义,即包含一致语义的两个对象于差异局部 E-R 图内名字存在差异。

③ 针对结构冲突而言,其涉及以下状况:

首先,相同对象于差异局部 E-R 图内包含差异抽象。例如,某对象于某个局部 E-R 图内属于实体,可是于别的局部 E-R 图内却为属性。

其次,相同实体于差异局部 E-R 图内所具有的属性数量与属性次序存在差异。这时能够借助于不同局部 E-R 图内属性的相应并集视作实体的相应属性,接着面向此实体的相应属性展开优化。

最后,不同实体彼此的联系于差异局部 E-R 图内包含差异的联系种类。例如,于某局部运用内两实体的相应联系属于多对多的相应联系,可是于另一局部运用内却属于一对多的相应联系。此时应根据实际的语义来进行调整。

(2) 修改和重构,产生基本的 E-R 图。

此步骤同样属于面向初步全局 E-R 图的改进。极佳的全局 E-R 图要求可以对用户的相应需求进行体现,还需要符合下列要求。

① 实体个数尽可能少。

② 实体具有的属性尽量少。

③ 实体间联系没有冗余。

对此,要求进行修改并重构初步 E-R 图,从而让冗余获得充分消除,着重运用探析方式。在探析方式之外,依然能够通过规范化的理论对冗余予以充分消除。在展开改进期间,能够把包含一致主码的两法外,同时通过规范化理论对冗余予以有效消除。改进期间,能够把包含一致主码的两个实体实施合并,同时能够思考把包含 1:1 联系的两个实体融成一个实体,对冗余属性以及冗余联系予以充分消除。然而需要关注的是,部分时候一定的冗余有助于效率的提升。

遵循以上所言方式,能够获得教学管理体系的全局 E-R 图,如图 1-8 所示。

需要强调的是,由于各实体包含的属性比较多,为了使所画的 E-R 图简明清晰,将各实

体所涉及的属性单独列出。下文依次对教学管理体系所牵涉的不同实体的相应属性(这当中不同实体的码通过下横线进行列出)予以罗列。

院系:(编号,称呼,主任,网址,电话)

专业:(编号,称呼,系编号)

学生:(学号,姓名,性别,民族,出生日期,政治面貌,班级称呼,家庭地址)

课程:(编号,名称,学分,周学时,理论课时,实践课时,全部学时,开课学期,课程种类)

教师:(编号,系编号,姓名,性别,民族,出生年月,参加工作日期,学历,学位,职称,是否为党员,基本工资,住宅电话,手机,院系编号,电子邮箱)

班级:(名称,专业编号,人数,班长姓名)

1.4.4 逻辑结构设计

针对逻辑结构设计而言,其目标即为将概念设计期间已设计的 E-R 图转变为和所选择 DBMS 产品所支持的相应数据模型彼此符合的逻辑结构。把 E-R 模型转变为关系模型即为把实体、实体的相应属性与不同实体彼此联系转变成关系模式,此类转换的相应规则涉及下列方面:

(1)实体型转变成关系模式。实体所对应的属性就是关系的相应属性,实体所对应的码即为关系所对应的码。

(2) $m:n$ 联系转变成关系模式,和此联系相连的不同实体所对应的码及联系自身的相应属性都转变为关系的相应属性,且关系所对应的码是两个实体码的相应组合。

(3) $1:n$ 联系能够转变成独立的关系模式,同时能够和 n 端相对应的关系模式进行合并。倘若转变为独立的关系模式,那么和此联系相连的不同实体所对应的码及联系自身的相应属性都转变成关系的相应属性,且关系所对应的码属于 n 端实体所对应的码。

(4) $1:1$ 联系能够转变成独立的关系模式,同时能够 and 所有端对应的相应关系模式进行合并。倘若转变成独立的关系模式,那么和此联系相连的不同实体所对应的码及联系自身的相应属性都转变成关系属性,所有实体所对应的码都属于此关系所对应的候选码。倘若和当中一端实体所对应的关系模式进行合并,那么要求于此关系的相应属性内添加其他关系模式所对应的码与联系自身的相应属性。

上面的关系实际上都是将一个实体转变成了一个关系。对于 $1:n$ 的实体之间的联系,没有进行转换,这是因为每个联系的 n 端的实体的属性都含有 1 端的码。

多对多关系的相应转换有以下几点:

(1)学生和课程联系的转换:

学生—课程(学号,课程相应编号,平常成绩,期末成绩,总体成绩)

(2)教师和班级联系的转换:

教师—班级(教师对应编号,班级相应名称)

(3)教师和课程联系的转换:

教师—课程(教师对应编号,课程相应编号)

(4)班级和课程联系的转换:

班级—课程(班级相应名称,课程对应编号)

在上面的转换过程中,后面的3个联系可以合为一个关系:教师授课(教师对应编号,课程对应编号,班级相应名称)

【例 1-9】把图 1-8 教学管理体系的 E-R 图转变为 Access 数据库管理体系予以支持的具体数据模型。

转换结果如表 1-16~表 1-23 所示。

表 1-16 院系

字段名	数据类型	字段大小	约束
系编号	文本型	2	主键
系名称	文本型	30	
系主任	文本型	10	
系网址	超链接		
系电话	文本型	10	

表 1-17 专业

字段名	数据类型	字段大小	约束
专业编号	文本型	2	主键
专业名称	文本型	20	
系编号	文本型	2	外键

表 1-18 学生

字段名	数据类型	字段大小	约束
学号	文本型	12	主键,类比身份证
姓名	文本型	20	非空
性别	文本型	2	
民族	文本型	20	
出生年月	日期时间型		
政治面貌	文本型	4	
班级名称	文本型	12	
家庭住址	文本型	50	

表 1-19 班级

字段名	数据类型	长度	约束
班级名称	文本型	12	主键,取名时用年级缩写+专业缩写+班级序号,如 12 中文 1 班
专业编号	文本型	2	外键
班级人数	数字	整型	
班长姓名	文本型	20	

表 1-20 教师

字段名	数据类型	长度	约束
教师编号	文本型	4	主键,全为数字
姓名	文本型	20	非空
性别	文本型	2	
民族	文本型	20	
出生年月	日期/时间型		
参加工作日期	日期/时间型		

续表 1-20 教师

字段名	数据类型	长度	约束
学历	文本型	10	
学位	文本型	10	
职称	文本型	10	
党员否	是/否型		
基本工资	货币型		
系编号	文本型	2	外键
住宅电话	文本型	13	
手机	文本型	11	
电子邮箱	超链接		

表 1-21 课程

字段名	数据类型	长度	约束
课程编号	文本型	8	主键
课程名	文本型	20	
学分	数字型	整型	
理论学时	数字型	整型	
实践学时	数字型	整型	
总学时	数字型	整型	
周学时	数字型	整型	
开课学期	文本型	2	
课程类型	文本型	10	包括公共必修、公共选修、专业必修、专业选修

表 1-22 学生选课

字段名	数据类型	长度	约束
学号	文本型	12	组合主键,外键
课程编号	文本型	8	组合主键,外键
平时成绩	数字型	单精度	
末考成绩	数字型	单精度	
综合成绩	数字型	单精度	

表 1-23 教师授课

字段名	数据类型	长度	约束
课程编号	文本型	8	组合主键,外键
教师编号	文本型	4	组合主键,外键
班级名称	文本型	12	组合主键,外键

1.4.5 数据库的物理设计

数据库的物理结构是数据库在物理装置中的存储结构和存取形式。数据库所对应的物理设计是给予所给定的相应逻辑数据模型挑选最满足应用环境的相应物理结构。

在对数据库进行物理设计时,首先要确定数据库的物理结构;其次对所设计的物理结构设计进行评价。

1 物理设计的内容与方法

数据库物理设计没有一个通用的准则,这是因为不同的 DBMS 提供的硬件环境、存储结构、存取方法及提供给数据库设计者的系统参数及其变化范围有所不同。极佳的数据库物理设计可以让在数据库内进行运作的不同事务具有较短的响应时长、较高的事务吞吐水平及运用水平,以及极大的存储空间。因此,在设计数据库时首先要对数据进行更新的事务及经常用到的查询进行详细分析,获得物理设计所需要的各种参数;其次要充分了解使用的 DBMS 的内部特征,特别是体系给予的存取形式与存储结构。

2 关系模式存取方法选择

数据库系统属于共享体系,面向相同关系应当构建多个存取路径才能符合多用户的诸多需求。

物理设计具有的一个任务即为针对挑选何种存取方式予以明确,即构建起何种存取路径。对于某个具体的 DBMS 来说,通常都给予不同的存取形式,具体运用什么方式通过 DBMS 按照数据的存储形式进行决定,使用者通常无法进行干预。在关系数据库内,打造的存取路径着重为明确怎样构建索引。所谓索引,其属于自数据库内取得数据的一种有效形式。高达 95% 的数据库性能问题能够借助于索引技术获得有效化解。

索引方法是根据应用需求在关系的一个属性或多个属性上建立索引,在有些属性上建立复合索引,有些索引要设置为唯一索引,有些属性要设置为聚簇索引。聚簇索引为遵循索引的相应属性列在物理上进行有序排列记录之后获得的相应索引,而非聚簇索引则为遵循

索引的相应属性列在逻辑上展开有序排列记录之后获得的相应索引,即聚簇索引所对应的次序属于数据在物理存储方面的次序,而非聚簇索引所对应的索引次序和数据物理排列次序不存在关联。

3 确定数据库的存储结构

针对数据库所涉及的存储结构进行明确时,涉及对数据所涉及的存储结构与存放场所进行明确,对索引以及聚簇等的存储规划与存储结构进行明确,对系统配置等进行明确。

针对数据库所涉及的存储结构进行明确,应当对多个要素(包括维护代价、存取时间以及存储空间运用水平)进行全面考虑。不同要素往往彼此冲突,因而应当予以折中,挑选出最佳的方案。

4 物理设计的评价

针对数据库展开物理设计期间,应当围绕时间与空间效率、维护代价以及不同用户需要展开权衡,因此极易形成诸多方案。从事数据库设计方面的工作者应当面向此类方案展开充分评估,进而挑选出最佳的方案用作数据库所对应的物理结构。

针对物理数据库展开评判的方式只依据所挑选的 DBMS。着重围绕不同方案的相应存取时间、存储空间以及维护代价展开定量估算,进而挑选最佳的方案。

1.4.6 数据库的实施

针对数据库的实施,其着重涉及以下步骤。

1 定义数据库结构

数据库的逻辑结构和物理结构确定后,就可以用确定的关系数据库管理系统提供的数据库定义语言 DDL 来对数据库的结构进行描述了。

2 编制与调试应用程序

数据库应用程序的设计与数据库设计并行运行,调试数据库应用程序时如果实际可用的数据还没有入库,可以先使用模拟数据来进行。

3 数据的载入

构建起数据库结构之后,随即能够把数据载进数据库内。安排数据进库属于数据库推行的基本工作。在常见的数据库运用体系内,数据库往往较大,同时数据源自不同机构,数据在组织形式、结构以及格式方面和全新设计的相应数据库体系要求存在差异。另外,系统对数据的完整性也有要求,这需要对数据进行以下操作:

(1)筛选数据。自分散在不同部门的有关数据文件以及初始文件内挑选应当入库的相应数据。

(2)输入数据。如果数据的格式与系统要求的格式不一样,就要进行数据格式的转换。如果数据量小,可以先转换后输入;如果数据量大,可以设计数据录入子系统来完成数据的自动转化工作。

(3)校验数据。旨在确保数据库内的相应数据能够正确,应当高度关注数据所涉及的校验工作,在运用设计数据录入子系统展开数据转变期间,应当实施多次的校验。特别是针对关键数据,尤其需要展开多次校验,明确没有错误之后方可载进数据库内。

另外,倘若最新建立的数据库源自已存在的数据库或者文件,则应当关注以往的数据库结构和全新数据库结构有无保持一致,然后再把以前的数据导进全新数据库内。

4 数据库试运行

实现针对应用程序的调试工作,同时载进一些数据之后,即能够展开数据库的相应试运作。

此时期应当对数据库应用程序展开实际执行,执行面向数据库的不同操作,检验应用程序的相应功能是否符合设计的要求。倘若不符合,则需要针对应用程序展开相应修改以及优化,直至满足设计需要。

1.4.7 数据库的运用和维护

针对数据库展开试运行,表明数据库设计和应用程序的相应开发工作已基本结束,进入运行及维护阶段,这一阶段是个长久的过程。

数据库所涉及的维护工作着重通过 DBA 实现,其基本工作涉及以下几个方面。

1 数据库的安全性和完整性控制

在数据库的运行过程中,对安全性的要求会随着应用环境的变化而变化。例如,某些数据的取值范围发生变化、某些用户的权限要收回、增加某些用户的权限、原来机密的数据现在可以公开化等。上述变化要求 DBA 按照具体情况来调整已有的安全性控制。此外,数据库所对应的完整性约束同样极易发生改变,这些全都要求 DBA 展开优化,从而符合用户相应需要。

2 数据库的备份和恢复

数据库体系运行期间,DBA 的一个核心工作为面向差异的应用,要求确立差异的备份安排,在一定时间内围绕数据库展开备份,倘若发生故障,则可以迅速把数据库恢复至最佳状态,充分降低数据的相应损失。

3 数据库性能的监视、分析和调整

数据库运行期间,DBA 的另外一个核心工作为针对系统运行展开监督,围绕监测数据实施探究,发现优化系统性能的最佳方式。现阶段,部分 DBMS 产品给予系统性能的相应检测方式,DBA 能够借助于此类方式对数据库展开有效监视。

4 数据库的重组与重构

在数据库运行一段时间后,伴随数据的持续添加、完善以及去除,将对数据库所涉及的存取效率产生影响,这时 DBA 需要针对数据库展开再次组织,或者针对数据库展开局部组织(仅面向频繁增删的表再次展开组织)。DBMS 一般提供相关的实用程序。在重组组织期间,依据已有设计要求再次规划数据的存放位置,回收垃圾以及降低指针链等,从而让系统性能获得优化。

针对数据库展开重组,未对已有设计的相应逻辑与物理结构造成改变,其中数据库的重新构建属于局部优化数据库的模式及内模式。

伴随数据库运用环境的改变如,对全新实体或者全新应用予以增加,对部分应用予以去除,实体间的关系发生变化等,有必要重新调整数据库的模式和内模式,简称数据库的重构。当然数据库的重构只能是部分重构,如果应用环境变化太大,重构也无济于事,标志数据库应用系统的生命周期结束,此时应当对全新数据库应用体系展开设计。



1.5 本章小结

本章着重阐述数据库管理技术的相应发展时期、数据库体系的相应概念、数据库体系的三级模式结构、数据模型的相应概念及其数据模型的类别划分、关系代数运算,并对数据库设计的主要知识进行讲解。希望读者借助此章的学习熟知数据库体系的主要知识及数据库设计的主要理论,进而从理论层面保障后续章节的有效学习。