

21 世纪高校规划教材

数据库 系统原理及应用

主编 晏争农 吴雪琴

江西高校出版社

图书在版编目(CIP)数据

数据库系统原理及应用/晏争农,吴雪琴主编. —南昌:江西高校出版社,2004.8

ISBN 7-81075-449-1

I. 数… II. ①晏… ②吴… III. 关系数据库-数据库管理系统,SQL Server 2000 IV. TP311.138

中国版本图书馆 CIP 数据核字(2004) 第 077716 号

江西高校出版社出版发行

(江西省南昌市洪都北大道 96 号)

邮编:330046 电话:(0791)8592235,8504319

江西太元科技有限公司照排部照排

江西教育印刷厂印刷

各地新华书店经销

*

2004 年 8 月第 1 版 2004 年 8 月第 1 次印刷

787mm×1092mm 1/16 20.25 印张 490 千字

印数:1~2000 册

定价:29.80 元

(江西高校版图书如有印刷、装订错误,请随时向承印厂调换)

前 言

随着计算机技术的飞速发展,计算机应用已渗透到千家万户,数据库技术的应用也从简单的事物管理扩展到各个应用领域。据不完全统计,在计算机应用中涉及到数据库应用的百分比已占 60% 以上,数据库不仅在传统的商业领域中、管理领域中发挥着主要作用,而且在非传统应用中也起着越来越大的作用,如在工程领域的工程数据库,在因特网的 Web 数据库,在多媒体技术的多媒体数据库等领域中都起着关键作用,数据库技术也将是所有信息技术和信息产业的基础。

数据库技术是目前高职院校计算机专业及其相关专业必修的一门学科,其主要的任务就是研究如何存储、使用和管理数据。学生通过学习可掌握数据库的基本原理、方法和应用技术,能有效地使用数据库管理系统和软件开发工具,掌握数据库结构的设计和数据库应用系统的开发。

随着数据库技术的逐渐成熟,数据库自身也发生了显著的变化,关系数据库技术业已成为当前数据库发展的主流,层次与网状数据库已经退出历史舞台,分布式数据库系统及面向对象数据库系统正在兴起。SQL Server 2000 是微软公司开发的数据库系统,是目前 Windows 平台下最为流行的一种关系数据库管理系统,它广泛地应用于客户机/服务器系统,多层构架系统或是 Web 网站系统,甚至工业控制系统。SQL Server 2000 是一个功能完备的数据库管理系统,它包括支持开发的引擎、标准的 SQL 语言、扩展的特性等功能。

本书比较系统、完整地讲述了数据库系统技术的基本原理和数据库系统 SQL Server 2000 的应用技术。全书共分 13 章,分别讲述了数据库系统概述、关系数据库、关系数据库规范化理论、数据库设计、数据库安全保护、SQL Server 2000 系统概述、SQL Server 2000 数据库操作、SQL Server 2000 数据库对象操作、用企业管理器创建基本表和管理数据、T-SQL 程序设计、SQL Server 2000 安全性管理、SQL Server 2000 数据库的备份和恢复、数据库应用实例等有关内容,每章后面均附有习题,以供课后练习。

本书由晏争农、朱用湖和段晓宁统稿,其中第一、二、五章由吴雪琴编写,第三、四、十章由占跃华编写,第六、七章由晏争农编写,第八、九章由朱用湖编写,第十一、十三章由周颖编写,第十二章由宋欣编写,同时赵宏之、谭晓芳和熊晓玲也参与了教材的编写工作。

由于编者水平所限,书中难免出现错误和不足之处,敬请广大读者批评指正。

编者

2004 年 6 月

目 录

第 1 章 数据库系统概述	(1)
1.1 数据库技术的产生、发展与研究领域	(1)
1.1.1 数据库技术的产生与发展	(1)
1.1.2 数据库技术的研究领域	(7)
1.2 数据库系统的组成和特点	(7)
1.2.1 数据库系统的组成	(7)
1.2.2 数据库系统各组成部分的特点	(7)
1.3 数据库系统的体系结构	(10)
1.3.1 三级模式结构	(10)
1.3.2 数据库系统的二级映像功能	(13)
1.4 数据库管理系统	(14)
1.4.1 DBMS 的主要功能	(14)
1.4.2 DBMS 的组成	(15)
1.4.3 用户访问数据的全过程	(15)
1.5 数据模型	(17)
1.5.1 数据模型概念	(17)
1.5.2 数据之间的联系	(17)
1.5.3 实体联系模型	(20)
1.5.4 结构数据模型	(22)
1.5.5 面向对象模型	(28)
第 2 章 关系数据库模型	(31)
2.1 关系模型的基本概念	(31)
2.1.1 关系数据库的优点和特点	(31)
2.1.2 关系数据库的内部结构	(31)
2.2 关系数据库的设计规划	(32)
2.2.1 表的关联与关系的性质	(32)
2.2.2 关系的键	(35)
2.2.3 数据完整性	(36)
2.3 关系代数	(38)
2.3.1 关系代数的分类及其运算符	(38)
2.3.2 传统的集合运算	(39)
2.3.3 专门的关系运算	(40)



第3章 关系数据规范化理论	(48)
3.1 规范化问题的提出	(48)
3.1.1 规范化理论的主要内容	(48)
3.1.2 不合理的关系模式存在的存储异常问题	(48)
3.2 函数依赖	(49)
3.2.1 函数依赖的定义及性质	(50)
3.2.2 函数依赖的相关概念	(50)
3.2.3 函数依赖的推理规则	(51)
3.2.4 函数依赖与码的联系	(51)
3.2.5 闭包及其运算	(52)
3.3 函数依赖的等价与覆盖	(53)
3.3.1 等价与覆盖	(53)
3.3.2 函数依赖集的最小集	(53)
3.4 关系模型的分解特性	(54)
3.4.1 无损连接性	(55)
3.4.2 函数依赖保持性	(55)
3.5 关系模式的规范化	(56)
3.5.1 第一范式(1NF)	(57)
3.5.2 第二范式(2NF)	(58)
3.5.3 第三范式(3NF)	(59)
3.5.4 BC范式(BCNF)	(60)
3.5.5 多值依赖与(4NF)	(61)
3.6 候选关键字的求解理论和算法	(62)
3.6.1 快速求解候选关键字的一个充分条件	(62)
3.6.2 左边为单属性的函数依赖集的候选关键字成员的图论判定法	(63)
3.6.3 多属性依赖集候选关键字求解法	(65)
第4章 数据库设计	(69)
4.1 数据库设计概念	(69)
4.1.1 数据库设计的任务、内容和特点	(69)
4.1.2 数据库设计方法简述	(69)
4.1.3 数据库设计步骤	(70)
4.2 规划	(71)
4.3 需求分析	(71)
4.3.1 需求描述与分析	(71)
4.3.2 需求分析的方法和步骤	(72)
4.4 概念结构设计	(73)
4.4.1 概念设计的必要性	(73)
4.4.2 概念模型	(73)

4.4.3	概念设计的方法与步骤	(74)
4.4.4	概念设计实例	(74)
4.5	逻辑结构设计	(78)
4.5.1	逻辑设计的任务和步骤	(79)
4.5.2	E-R模型向关系模型的转换	(79)
4.5.3	关系模式规范化	(80)
4.5.4	模式优化	(81)
4.6	物理结构设计	(82)
4.6.1	物理设计的环境	(82)
4.6.2	物理设计的步骤	(82)
4.6.3	物理设计的性能评价	(83)
4.7	数据库实施	(83)
4.7.1	建立实际数据库结构	(83)
4.7.2	装入数据	(84)
4.7.3	应用程序编码与调试	(84)
4.7.4	数据库试运行	(84)
4.8	数据库运行与维护	(85)
4.8.1	维护数据库的安全性和完整性	(85)
4.8.2	监测并改善数据库性能	(85)
4.8.3	重新组织和构造数据库	(85)
第5章	数据库安全保护	(87)
5.1	数据库的安全性控制	(87)
5.1.1	数据库安全性的含义	(87)
5.1.2	安全性控制的一般方法	(87)
5.1.3	安全控制模型	(88)
5.2	数据库完整性控制	(92)
5.2.1	完整性的含义	(92)
5.2.2	完整性规则	(92)
5.2.3	完整性约束条件的分类	(94)
5.3	事务的基本概念	(95)
5.3.1	事务的定义	(95)
5.3.2	事务的特征	(95)
5.3.3	SQL事务处理模型	(96)
5.4	数据库的并发控制与封锁	(97)
5.4.1	数据库并发性的含义	(97)
5.4.2	数据库的并发操作带来的问题	(97)
5.4.3	并发控制措施	(99)
5.4.4	封锁协议	(100)



5.4.5 死锁和活锁	(103)
5.5 数据库备份与恢复	(105)
5.5.1 数据库故障的种类	(105)
5.5.2 数据库备份	(106)
5.5.3 数据库恢复的原理及其实现技术	(106)
5.5.4 数据库恢复	(108)
第6章 SQL Server 2000 系统概述	(110)
6.1 SQL Server 2000 的功能和特点	(110)
6.1.1 SQL Server 2000 的功能	(110)
6.1.2 SQL Server 2000 的特性	(111)
6.2 SQL Server 2000 安装和配置	(111)
6.2.1 SQL Server 2000 安装前的准备	(111)
6.2.2 SQL Server 2000 的安装	(112)
6.2.3 SQL Server 2000 安装后的故障及其处理	(119)
6.3 SQL Server 2000 工具	(120)
6.3.1 服务管理器	(120)
6.3.2 企业管理器	(122)
6.3.3 查询分析器	(123)
6.3.4 网络实用工具	(125)
第7章 SQL Server 2000 数据库操作	(127)
7.1 创建数据库	(127)
7.1.1 使用企业管理器创建数据库	(127)
7.1.2 使用 CREATE DATABASE 语句创建数据库	(130)
7.2 设置数据库选项	(134)
7.2.1 使用企业管理器设置	(134)
7.2.2 使用查询分析器设置	(136)
7.3 修改数据库	(138)
7.3.1 更改数据库名称	(138)
7.3.2 修改数据库文件大小	(139)
7.3.3 修改数据库结构	(145)
7.4 删除数据库	(147)
7.4.1 使用企业管理器删除	(147)
7.4.2 使用 DROP DATABASE 语句删除	(147)
第8章 SQL Server 2000 数据库对象操作	(150)
8.1 基本概念	(150)
8.1.1 SQL 语言的特点	(150)

8.1.2 SQL 语言功能概括	(150)
8.2 SQL 的数据类型	(151)
8.2.1 字段数据类型	(151)
8.2.2 类型的说明	(153)
8.3 数据定义功能	(153)
8.3.1 基本表的定义与删除	(153)
8.3.2 修改基本表	(158)
8.3.3 创建和维护索引	(159)
8.4 SQL 数据查询功能	(161)
8.4.1 查询语句的基本结构	(161)
8.4.2 简单查询	(162)
8.4.3 条件查询	(163)
8.4.4 利用计算函数统计汇总数据查询	(167)
8.4.5 分组查询	(168)
8.4.6 对查询结果进行排序	(169)
8.4.7 数据表连接及连接的查询	(170)
8.4.8 子查询	(173)
8.4.9 合并查询	(176)
8.4.10 存储查询结果到表中	(176)
8.5 SQL 数据更新	(177)
8.5.1 插入数据记录	(177)
8.5.2 更新数据记录	(178)
8.5.3 删除数据记录	(179)
8.6 视图	(179)
8.6.1 视图的概念	(179)
8.6.2 定义和删除视图	(180)
8.6.3 查询视图	(181)
第 9 章 用企业管理器创建基本表和管理数据	(185)
9.1 用企业管理器定义表及完整性约束	(185)
9.1.1 定义表及约束	(185)
9.1.2 修改表结构	(191)
9.1.3 查看表	(192)
9.1.4 用 Enterprise Manager 器删除表	(194)
9.2 用 Enterprise Manager 管理数据	(194)
9.2.1 添加数据	(194)
9.2.2 删除数据	(195)
9.2.3 修改数据	(195)
9.3 索引	(195)



9.3.1	创建索引	(195)
9.3.2	查看与修改索引	(198)
9.3.3	删除索引	(200)
9.4	数据完整性	(201)
9.4.1	数据完整性概述	(201)
9.4.2	使用约束实施数据完整性	(202)
9.4.3	使用规则	(206)
9.4.4	使用默认值	(209)
9.5	存储过程	(212)
9.5.1	存储过程概述	(212)
9.5.2	创建存储过程	(212)
9.5.3	执行存储过程	(215)
9.5.4	查看存储过程	(217)
9.5.5	修改存储过程	(218)
9.5.6	删除存储过程	(218)
9.6	触发器概述	(219)
9.6.1	触发器的简介	(219)
9.6.2	创建触发器	(220)
9.6.3	查看触发器	(222)
9.6.4	修改触发器	(224)
9.6.5	删除触发器	(224)
9.7	企业管理器管理视图	(225)
9.7.1	创建视图	(225)
9.7.2	管理视图	(226)
9.8	数据库图表	(227)
9.8.1	创建数据库图表	(227)
9.8.2	添加外码限制	(227)
9.8.3	向数据图表中加入表格	(229)
第 10 章	T-SQL 程序设计	(236)
10.1	变量	(236)
10.1.1	全局变量	(236)
10.1.2	局部变量	(238)
10.2	SQL Server 函数	(241)
10.2.1	SQL Server 的数学函数	(241)
10.2.2	SQL Server 的字符串函数	(244)
10.2.3	SQL Server 的日期函数	(245)
10.2.4	SQL Server 的系统函数	(247)
10.2.5	其他常用函数	(248)

10.2.6	用户自定义函数	(249)
10.3	程序流程控制	(252)
10.3.1	BEGIN...END 语句	(252)
10.3.2	IF...ELSE 语句	(253)
10.3.3	WHILE 语句	(254)
10.3.4	RETURN 语句	(255)
10.3.5	WAITFOR 语句	(256)
10.3.6	GOTO 语句	(256)
10.4	使用游标	(257)
10.4.1	声名游标	(257)
10.4.2	打开和使用游标	(260)
10.4.3	关闭和释放游标	(262)
10.5	使用注释	(263)
10.6	使用批处理	(263)
第 11 章	SQL Server 2000 数据库的安全性管理	(265)
11.1	SQL Server 2000 的安全模式	(265)
11.1.1	SQL Server 2000 的身份验证	(265)
11.1.2	选择身份验证模式	(265)
11.2	登录账号	(266)
11.2.1	创建登录账号	(266)
11.2.2	修改账号信息	(269)
11.2.3	查看登录账号信息	(269)
11.2.4	删除登录账号	(269)
11.3	数据库用户	(270)
11.3.1	创建数据库用户	(270)
11.3.2	删除数据库用户	(271)
11.4	数据库角色	(271)
11.4.1	创建数据库角色	(271)
11.4.2	为服务器固有角色添加成员	(272)
11.4.3	删除数据库角色	(273)
11.5	权限	(274)
11.5.1	权限的分类	(274)
11.5.2	管理对象权限	(275)
11.5.3	管理系统权限	(277)
11.6	使用 T-SQL 语句管理权限与角色	(278)
11.6.1	系统权限与角色的授予、收回与拒绝访问	(278)
11.6.2	使用 Transact-SQL 语句管理对象权限	(279)



第 12 章 数据库的备份和恢复	(282)
12.1 数据备份概念	(282)
12.1.1 备份策略	(282)
12.1.2 备份设备	(283)
12.2 数据库备份	(284)
12.2.1 使用备份向导	(284)
12.2.2 使用企业管理器	(284)
12.2.3 使用 BACKUP DATABASE 语句	(288)
12.2.4 备份事务日志	(289)
12.3 恢复数据库	(291)
12.3.1 使用企业管理器	(291)
12.3.2 使用 RESTORE DATABASE 语句恢复数据库	(293)
第 13 章 数据库应用实例	(296)
13.1 建立数据访问接口	(296)
13.1.1 建立 ODBC 数据源	(296)
13.1.2 用 ADO 数据控件进行数据绑定	(299)
13.1.3 DataGrid 数据绑定控件	(302)
13.2 三个实例	(302)
13.2.1 示例 1——学生基本信息录入	(303)
13.2.2 示例 2——课程基本情况录入	(305)
13.2.3 示例 3——学生成绩查询	(306)
主要参考文献	(310)



第1章 数据库系统概述

数据库(Database),简单地讲就是数据的仓库,即数据存放的地方。随着信息管理水平的不断提高,信息资源已成为企业的重要财富和资源,用于信息管理的数据库技术也得到了很大的发展,其应用领域也越来越广泛。数据库的应用形式日益多样,从小型事务处理到大型信息系统,从联机事务处理到联机分析处理,从一般企业管理到计算机辅助设计与制造(CAD/CAM),乃至地理信息系统等都应用了数据库技术。数据库技术已经渗透到我们日常生活的方方面面,比如用信用卡购物,飞机、火车订票系统,图书馆对书籍及借阅的管理等,无一不使用了数据库技术。数据库的建设规模、数据库中信息量的大小以及使用的程度已经成为衡量企业的信息化程度的重要标志。

简单地讲,数据库技术就是研究如何科学地管理数据以便为人们提供可共享的、安全的、可靠的数据的技术。数据库技术一般包括数据管理和数据处理两部分内容。

数据库系统实质上是一个用计算机存储数据的系统,可以将数据库看作一个电子文件柜,也就是说,数据库是收集数据文件的仓库或容器。

1.1 数据库技术的产生、发展与研究领域

1.1.1 数据库技术的产生与发展

数据处理的中心问题是数据管理,数据管理是指对数据进行组织、分类、编码、存储、检索和维护。随着计算机硬件和软件的发展,数据管理经历了人工管理阶段、文件系统阶段和数据库系统阶段三个发展阶段。

1. 人工管理阶段

在20世纪50年代中期以前,计算机主要用于科学计算,硬件中的外存只有卡片、纸带、磁带,没有磁盘等直接存取设备;软件只有汇编语言,没有操作系统和管理数据的软件。数据处理的方式基本上是批处理方式。

人工管理数据的特点:

(1) 数据不保存

因为当时计算机主要用于科学计算,对于数据保存的需求尚不迫切。在计算某一课题时将数据输入主存,运算处理后将结果输出,计算机在处理数据过程中不保存程序和数据。

(2) 系统没有专用的软件对数据进行管理

每个应用程序都要包括数据的存储结构、存取方法和输入方式等,程序员编写应用程序时,还要安排数据的物理存储,因此,程序员的负担很重。

(3) 数据不共享

数据是面向程序的,一组数据只能对应一个程序。多个应用程序涉及某些相同的数据时,也必须各自定义,因此程序之间有大量的冗余数据。



(4)数据不具有独立性

数据的独立性是指逻辑独立性和物理独立性。程序依赖于数据,如果数据的类型、格式或输入输出方式等逻辑结构或物理结构发生变化,必须对应用程序做出相应的修改。

在人工管理阶段,程序与数据之间的关系可用图 1-1 表示。

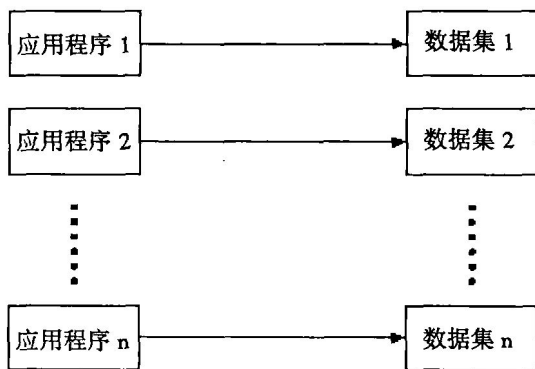


图 1-1 人工管理阶段

2. 文件系统阶段

早期的数据是采用文件系统进行管理的,即将数据保存在文件中。用户的应用程序直接操作文件中的数据。在文件系统中,数据按其内容、结构和用途分成若干个命名的文件。文件一般为某一用户或用户组所有,但也可以指定与其他用户共享。用户可以通过操作系统对文件进行打开、读、写、关闭等操作。

假设现在用文件系统来实现对学生进行管理的程序。在此系统中,要对学生的基本信息和选课情况进行管理。在管理学生基本信息时要用到学生的基本信息数据,假设此数据存在 F1 文件中。学生选课情况的管理包括学生的基本信息、课程的基本信息和学生的选课信息的管理假设用 F2 和 F3 两个文件分别存储课程基本信息和学生选课基本信息数据。学生选课情况管理中涉及的学生基本信息可以使用学生基本信息管理系统中的 F1 文件。假设实现学生基本信息管理功能的应用程序叫 A1,实现学生选课管理功能的应用程序叫 A2,则学生的基本信息和选课情况可用图 1-2 表示。

假设 F1、F2 和 F3 文件分别包含如下信息:

- F1 包含学号、姓名、性别、出生日期、所在系、专业、所在班、特长、家庭住址。
- F2 包含课程号、课程名、授课学期、学分、课程性质。
- F3 包含学号、姓名、专业、课程号、课程名、修课类型、修课时间、考试成绩。

我们将文件中所包含的每一个子项称为文件结构中的字段或列,将每一行数据称为一个记录。

“学生选课管理”系统的处理过程大致为:在学生选课管理系统中,若有学生选课,则先查 F1 文件,判断有无此学生。若有此学生,则再访问 F2 文件,判断其所选的课程是否存在。若课程也存在,就将学生选课信息写到 F3 文件中。

经仔细分析,就会发现文件管理系统管理数据有如下的特点和存在的问题:

(1)文件系统管理数据的特点

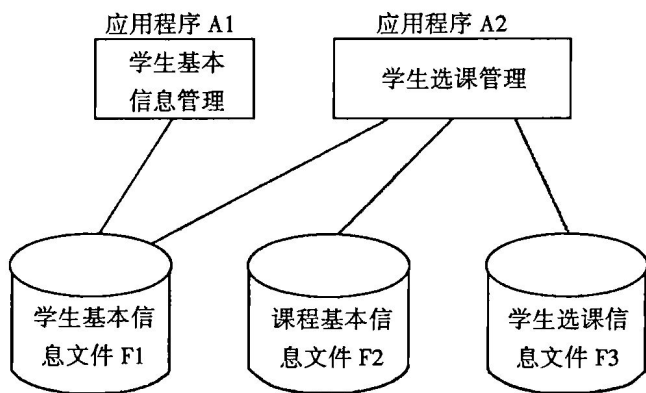


图 1-2 文件系统阶段

①数据以文件形式长期保存。数据以文件的组织方式,保存在计算机的存储设备上,应用程序可对文件进行查询、修改和增删等处理。

②文件系统可对数据的存取进行管理。程序员只与存储在存储设备上的文件的文件名打交道,不必关心数据的物理存储,大大减轻了程序员的负担。

③文件形式多样化。文件有顺序文件、倒排文件和索引文件等,因而对文件的记录可顺序访问,也可随机访问,更便于存储和查找数据。

④程序与数据间有一定独立性。数据由专门的软件即文件系统进行管理,程序和数据间由软件提供的存取方法进行转换,数据存储发生变化不一定影响程序的运行。

(2) 文件系统管理数据中存在的问题

①编写应用程序不方便

应用程序编写者必须对所用文件的逻辑及物理结构有清楚的了解,例如文件中包含多少个字段,每个字段的数据类型,采用何种存储结构,比如链表或数组等。操作系统只提供了打开、关闭、读、写等几个低级的文件操作命令,而文件的查询、修改等处理都必须在应用程序中通过编程实现。这样也容易造成各应用程序在功能上的重复,例如图 1-2 中的“学生基本信息管理”和“学生选课管理”都要对 F1 文件进行操作,但这两个功能相同的操作却很难共享。

②数据冗余不可避免

假设 A2 需要用到 F3 文件中包含的学生的所有或大部分信息,例如,除了学号之外,还需要姓名、性别、专业、所在系等信息,而 F1 中也包含了这些信息,因此 F3 和 F1 文件中有重复的信息。但这些重复的信息只是不同文件的部分内容,因此很难在两个文件中公用这些公共信息,从而造成数据的重复,也叫数据的冗余。

数据冗余不仅会造成存储空间的浪费,更为严重的是造成了数据的不一致。例如,假设某个学生所学的专业发生了变化,我们一般只会想到在 F1 文件中进行修改,而往往忘记了在 F3 中要进行同样的修改。这样就会造成同一名学生在 F1 文件和 F3 文件中的“专业”不一样,也就是数据不一致性,人们不能判定哪个数据是正确的,尤其当数据冗余很多的时候,情况更是如此。这样数据就失去了其可信性。

文件系统本身不具备维护数据一致性的功能,这些功能完全由用户或应用程序开发者



来负责维护。这在简单的系统中还可以勉强应付,但在复杂的系统中,若让开发者来保证数据的一致性,几乎是不可能的。

③应用程序有依赖性

就文件系统而言,应用程序要依赖于文件的结构。文件和记录的结构通常是应用程序代码的一部分,如 C 语言的 Struct。文件结构每进行一次修改,比如添加字段、删除字段甚至是修改字段的长度(如电话号码从 7 位扩到 8 位),都要对应用程序进行相应的修改,因为我们在打开文件读取数据时,必须要将文件记录中的不同字段的值对应到程序变量中。随着应用环境和需求的变化,修改文件的结构是不可避免的事情,这样就需要在应用程序中进行相应的修改,而频繁修改应用程序是很麻烦的。这是因为首先要熟悉原有程序,修改后还需要对程序进行测试、安装等。

以上弊端都是由于应用程序对文件结构过分依赖造成的。换句话说,文件系统的数据库独立性不好。

④不支持对文件的并发访问

在现代计算机系统中,为了有效地利用计算机资源,系统一般允许多个应用程序并发运行,尤其是在现在的多任务操作系统环境中。文件最初是作为程序的附属数据出现的,它一般不支持多个应用程序同时对同一个文件进行访问。我们可以回忆一下,假设某个用户打开了一个 Excel 文件,如果第二个用户在第一个用户没有关闭此文件之前就想打开此文件,他会得到什么信息?他只能以只读方式打开此文件,而不能在第一个用户打开文件的同时对此文件进行修改;我们再回忆一下,如果我们用 C 语言编写一个修改某文件内容的程序,其过程是先以写的方式打开文件,然后写入新内容,最后再关闭文件。在文件关闭之前,无论在其他的程序中还是在同一个程序中都是不能再打开此文件的,这就是文件系统不支持并发访问的含义。

对于以数据为中心的应用系统来说,必须要支持多个用户对数据的并发访问。

⑤数据间的联系弱

在文件系统中,文件与文件之间是彼此独立、毫不相干的,文件之间的联系必须通过程序来实现。比如在上述的 F1 和 F3 文件中,F3 文件中的学号、姓名等学生的基本信息必须是 F1 文件中已经存在的(即选课的学生必须是已经存在的学生)。同样,F3 文件中的课程号等与课程有关的基本信息也必须是 F2 文件中已经存在的(即学生选的课程也必须是已经存在的课程)。这些数据之间的联系是客观需求当中所要求的很自然的联系,但文件系统本身不具备自动实现这些联系的功能,所以必须通过应用程序来保证这些联系,也就是说必须编写代码来手工地保证这些联系。这样不但增加了编写代码的工作量和复杂度,而且当联系很复杂时,也难以保证其正确性。因此,文件系统不能反映现实世界事物间的联系。

⑥难以按不同用户的需要表示数据

如果用户需要的信息来自多个不同的数据文件,我们就需要对多个文件的信息内容进行提取、比较、组合和表示。例如,假设有用户希望得到如下信息:

所在班、学号、姓名、课程名、学分、考试成绩。

这些信息涉及到了三个文件。从 F1 文件中可以得到“所在系”信息,从 F2 文件中得到“学分”信息,从 F3 文件中得到“考试成绩”信息。而“学号”、“姓名”信息可以从 F1 或 F3 文件中得到,“课程名”可以从 F2 或 F3 文件中得到。我们在生成一行数据时,必须比较从三个

文件中读取的数据,然后组合成一行有意义的数。比如,将从 F1 文件中读取的学号与从 F3 中读取的学号进行比较,学号相同时,才可以将 F1 中的“所在系”、F3 中的“考试成绩”以及当前所对应的学号和姓名组合成一行数据的内容。同样,在处理完 F1 和 F3 文件的组合后,我们可以将组合的结果再与 F2 文件的内容进行比较,找出课程号相同的课程的学分,再与已有的结果组合起来。如果数据量很大、涉及的文件比较多时,可以想像这个过程有多复杂。因此,这种大容量复杂信息的查询,在文件管理系统中是很难处理的。

⑦无安全控制功能

在文件管理系统中,很难控制某个人对文件的操作,例如控制某人只能读和修改文件,不能删除文件,或者不能读或修改文件中的某个或者某些字段等等。而在实际应用中,数据的安全性是非常重要的且不可缺少的。例如,在学生选课管理系统中,我们不允许学生修改考试成绩。在银行系统中,更是不允许一般用户修改其存款数额。

随着人们对数据需求的增加以及计算机科学技术的不断发展,对数据进行有效、科学、正确、方便的管理就成为人们的迫切需求。针对文件系统的这些缺陷,人们逐步开发出了以统一管理和共享数据为主要特征的数据库管理系统。

3. 数据库系统阶段

从 20 世纪 60 年代后期开始,计算机应用于管理的规模更加庞大,数据量急剧增加,硬件方面出现了大容量磁盘,使计算机联机存取大量数据成为可能。硬件价格下降,而软件价格上升,使开发和维护系统软件的成本增加。文件系统的管理方法已无法适应开发应用系统的需要。于是为解决多用户、多个应用程序共享数据的需求,出现了统一管理数据的专门软件系统,即数据库管理系统。

数据库系统管理数据的特点有:

(1) 数据共享性高、冗余低

这是数据库系统阶段的最大改进,数据不再面向某个应用程序而是面向整个系统,当前所有用户可同时存取数据库中的数据。这样便减少了不必要的数据冗余,节约了存储空间,同时也避免了数据之间的不相容性与不一致性。

(2) 数据结构化

在数据库系统中,按照某种数据模型,将应用的各种数据组织到一个结构化的数据库中,系统不仅要考虑某个应用的数据结构,还要考虑整个应用的数据结构,这样整个应用的数据不是一盘散沙,可表示出数据之间的有机关联。

例如,要建立学生成绩管理系统,系统包含学生(学号、姓名、性别、系别、年龄)、课程(课程号、课程名)、成绩(学号、课程号、成绩)等数据,分别对应三个文件。若采用文件处理方式,因为文件系统只表示记录内部的联系,而不涉及不同文件记录之间的联系,要想查找某个学生的学号、姓名、所选课程名称和成绩,必须编写一段比较复杂的程序来实现。而采用数据库方式,由于数据库系统不仅描述数据本身,还描述数据之间的联系,上述查询可以非常容易地联机查到。

(3) 数据独立性高

数据的逻辑独立性是指当数据间的总体逻辑结构改变时,数据的局部逻辑结构不变。由于应用程序是依据数据的局部逻辑结构编写的,所以应用程序不必修改,从而保证了数据与程序间的逻辑独立性。例如,在原有的记录类型之间增加新的联系,或在某些记录类型中



增加新的数据项,均可确保数据的逻辑独立性。

数据的物理独立性是指当数据的存储结构改变时,数据的逻辑结构不变,从而应用程序也不必改变。例如,在改变存储设备和增加新的存储设备,或改变数据的存储组织方式时,均可确保数据的物理独立性。

数据的独立性是由系统提供的两个映像的功能保证的,关于二级映像,将在后一节讨论。

(4)有统一的数据控制功能

在数据库系统中,数据由数据库管理系统 DBMS(DataBase Management System)进行统一管理和控制。数据库可为多个用户和应用程序所共享,不同的应用需求,可以从整个数据库中选取所需要的数据子集。另外,对数据库中数据的存取往往是并发的,即多个用户可以同时存取数据库中的数据,甚至可以同时存取数据库中的同一个数据。为确保数据库数据的正确、有效和数据库系统的有效运行,数据库管理系统提供下述四方面的数据控制功能:

①数据的安全性(Security)控制:防止不合法使用数据造成数据的泄露和破坏,保证数据的安全。例如,系统提供口令检查用户身份或用其他手段来验证用户身份,以防止非法用户使用系统。也可以对数据的存取权限进行限制,只有通过检查后才能执行相应的操作。

②数据的完整性(Integdy)控制:系统通过设置一些完整性规则,确保数据的正确性、有效性和相容性。

正确性是指数据的合法性,如年龄属于数值型数据,只能含 0,1,⋯,9,不能含字母或特殊符号。

有效性是指数据是否在其定义的有效范围,如月份只能用 1~12 的正整数表示。

相容性是指表示同一事实的两个数据应相同,否则就不相容,如一个人不能有两个性别。

③并发(Concurrency)控制:多个用户同时存取或修改数据库时,防止相互干扰而提供给用户不正确的数据,并使数据库受到破坏。

④数据恢复(Recovery):由于计算机系统的硬件故障、软件故障、操作员的误操作及其他故意的破坏等原因,造成数据库中的数据不正确或数据丢失时,系统有能力将数据库从错误状态恢复到最近某一时刻的正确状态。

在数据库系统阶段,程序与数据之间的关系可用图 1-3 表示。

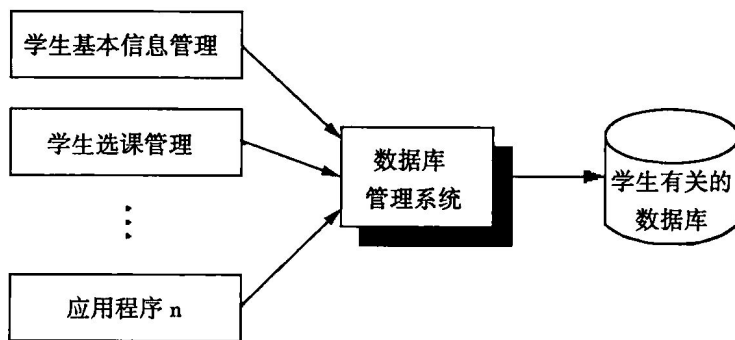


图 1-3 程序与数据之间的关系