

精品课程配套教材

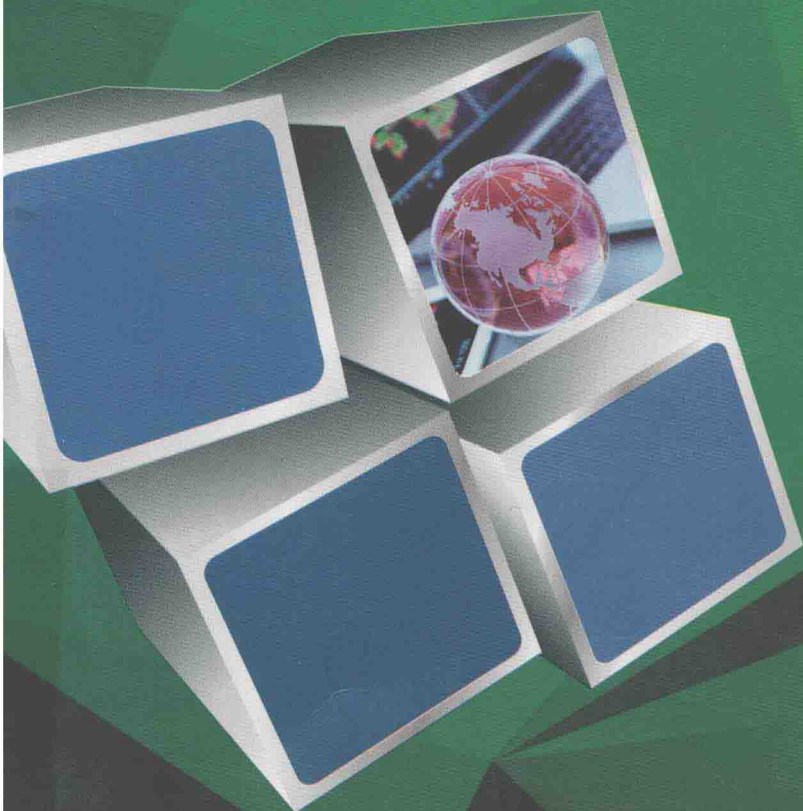
“双创”型人才培养优秀教材  
21世纪应用型人才培养“十三五”规划教材

# 数据库原理与应用

## (SQL Server 2012) —— 基于计算思维

SHUJUKU YUANLI YU YINGYONG (SQL SERVER 2012)  
—JIYU JISUAN SIWEI

主 编 刘征海 肖建田 张志强



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

程配套教材

型人才培养优秀教材

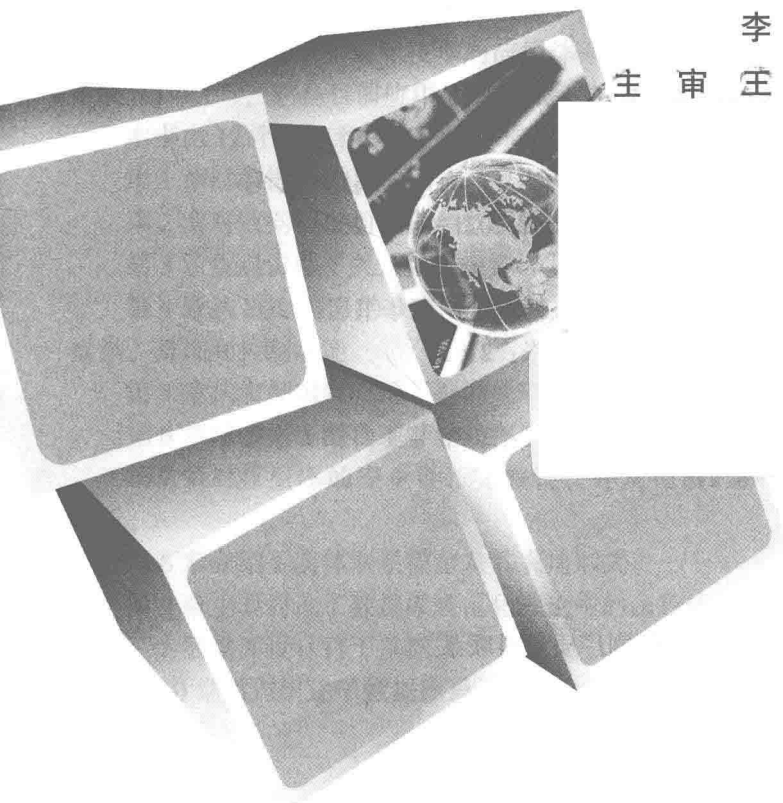
21世纪应用型人才培养“十三五”规划教材

# 数据库原理与应用

(SQL Server 2012) —— 基于计算思维

SHUJUKU YUANLI YU YINGYONG (SQL SERVER 2012)  
—JIYU JISUAN SIWEI

主 编 刘征海 肖建田 张志强  
副主编 宋国顺 孙 滨 万雪勇 蒋德荣  
许又泉 潘燕燕 柳 均 何燕飞  
李 力 吴勇刚  
主 审 王 宁



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

## 内 容 提 要

本书由 12 章组成, 从数据库基础理论、数据库设计、数据库发展、数据库实践几个方面全面阐述了数据库技术的应用体系。并选用 SQL Server 2012 平台作为数据库原理系统, 它是目前应用范围广泛且功能完善、操作界面友好的数据库管理系统。

本书适宜从事数据库技术开发和管理的相关师生、科研人员。

## 图书在版编目 (CIP) 数据

数据库原理与应用: SQL Server 2012: 基于计算  
思维/刘征海, 肖建田, 张志强主编. --上海: 上  
海交通大学出版社, 2018

ISBN 978-7-313-18575-4

I. ①数… II. ①刘…②肖…③张… III. ①关系数  
据库系统 IV. ①TP311.138

中国版本图书馆 CIP 数据核字 (2018) 第 015545 号

## 数据库原理与应用 (SQL Server 2012) ——基于计算思维

主 编: 刘征海 肖建田 张志强

出版发行: 上海交通大学出版社

邮政编码: 200030

出版人: 谈 毅

印 制: 三河市宇通印刷有限公司

开 本: 787mm×1092mm 1/16

字 数: 512 千字

版 次: 2018 年 1 月第 1 版

书 号: ISBN 978-7-313-18575-4/TP

定 价: 42.00 元

地 址: 上海市番禺路 951 号

电 话: 021-64071208

经 销: 全国新华书店

印 张: 21.25

印 次: 2018 年 1 月第 1 次印刷

版权所有 侵权必究

告 读 者: 如发现本书有印装质量问题请与印刷厂质量科联系

联系电话: 010-82477073

# 前 言

数据库技术是计算机科学中一个非常重要的部分，它已成为计算机信息管理系统及其应用系统的核心技术和重要基础，它是计算机、软件工程等相关专业学生必需学习和掌握的技术之一。

数据库技术是 20 世纪 60 年代末发展起来的一种数据管理技术。它的出现标志着以应用程序自己管理数据、数据无法共享等为特点的传统手工管理数据阶段的结束。经过几十年的发展，已经建立起一套较为完整的理论体系。

随着移动通信、物联网、云计算、大数据等新概念和新技术的发展，信息技术深刻地改变着人类的思维、生产、生活、学习方式，与之相关的计算思维成为人们认识和解决问题的能力之一。我们将计算思维的训练融入数据库原理的教学当中，对课程体系结构进行了重构，形成数据库知识体系和其对应的计算思维核心概念对照表。把渗透和融入了计算思维相关特征的数据库知识教授给学生，使学生能够更好地掌握数据库的基本概念、数据库开发及编程技术，能够熟练运用数据库技术去开发和管理数据库系统，努力提高学生的实际应用能力以及创新能力。

本书的特色：

内容安排求全、求新。本书从数据库基础理论、数据库设计、数据库发展、数据库实践几个方面全面阐述了数据库技术的应用体系。并选用 SQL Server 2012 平台作为数据库原理系统（DBMS），它是目前应用范围广泛且功能完善、操作界面友好的数据库管理系统。

本书由 12 个章节和 3 个附录组成：

第 1 章对数据库的基本概念、数据库系统的体系结构等内容进行了探讨。

第 2 章对数据模型的概念、数据模型的组成、常用数据模型进行了介绍。

第 3 章重点讲解了关系数据模型以及关系代数。

第 4 章从 SQL 语言的发展、SQL 支持的数据类型出发，全面讲解了 SQL 语言的定义、操纵、查询和控制功能。

第 5 章从实例出发介绍了查询处理的过程以及查询优化的常用方法。

第 6 章对数据库的物理存储的存储介质、文件组织方式以及索引等进行了详细的分析。

第 7 章对数据库的安全性、完整性、并发控制和恢复等数据保护技术进行了深入的分析。

第 8 章介绍了实体联系模型及其表达形式。

第 9 章主要讨论了数据库规范化理论中的函数依赖、关系模式的规范化以及关系模式的分解，为数据库设计打下基础。

第 10 章重点讲述了数据库设计的概念模型设计、逻辑模型设计以及物理结构设计的方法和步骤。

第 11 章全面介绍了 T-SQL 高级编程,包括编程基础、流程控制、游标的使用、存储过程和触发器的编写。

第 12 章通过对学生信息管理系统案例的分析、设计以及实现,将基础知识融入到实践活动当中。

附录 A 通过详细的图文描述了 SQL Server 2012 数据库的安装过程,使零起点的学生也能轻松完成 SQL Server 2012 的安装。

附录 B 重点给出了 SQL Server 2012 中常用的系统函数。

附录 C 给出了数据库原理的 8 个实验的实验指导书。

其中第 1 章、第 3 章、第 7 章、第 8 章由肖建田老师完成,第 2 章由宋国顺老师、孙滨老师、万雪勇共同完成,第 6 章由蒋德荣老师、许又泉老师、潘燕燕老师、何燕飞老师、柳均老师共同完成,其他部分由刘征海老师和张志强老师共同完成。

由于编者水平有限、本书存在的不当之处敬请读者不吝指正。

编 者

2018 年 1 月

# 目录 Contents

<b>第 1 章 数据库概述</b> .....	1
1.1 概述 .....	1
1.2 基本概念 .....	1
1.3 数据技术的发展 .....	3
1.4 数据库的结构 .....	10
1.5 数据库与计算思维 .....	14
<b>第 2 章 数据模型</b> .....	18
2.1 数据和数据模型 .....	18
2.2 概念层数据模型 .....	20
2.3 组织层数据模型 .....	24
<b>第 3 章 关系数据模型</b> .....	29
3.1 关系数据模型和关系数据库 .....	29
3.2 关系模型的基本术语与形式化定义 .....	32
3.3 关系代数 .....	36
<b>第 4 章 SQL 语言</b> .....	51
4.1 SQL 语言概述 .....	51
4.2 SQL 支持的数据类型 .....	53
4.3 数据定义功能 .....	55
4.4 数据查询功能 .....	60
4.5 视图 .....	102
4.6 数据更改功能 .....	107
4.7 数据控制功能 .....	112
<b>第 5 章 查询处理与优化</b> .....	116
5.1 概述 .....	116
5.2 SQL 的查询处理 .....	116

5.3 查询优化方法 .....	119
<b>第6章 数据库的存储 .....</b>	<b>127</b>
6.1 物理存储介质 .....	127
6.2 文件的组织 .....	130
6.3 索引 .....	134
<b>第7章 数据库安全与保护 .....</b>	<b>142</b>
7.1 安全与保护概述 .....	142
7.2 数据库的安全性 .....	143
7.3 数据库的完整性 .....	162
7.4 事务 .....	173
7.5 并发控制 .....	182
7.6 数据库恢复技术 .....	190
<b>第8章 实体-联系模型 .....</b>	<b>203</b>
8.1 E-R 模型的基本概念 .....	203
8.2 E-R 图符号 .....	211
<b>第9章 关系规范化理论 .....</b>	<b>213</b>
9.1 关系模式设计的问题 .....	213
9.2 函数依赖 .....	215
9.3 范式 .....	223
9.4 关系模式的分解准则 .....	230
<b>第10章 数据库设计 .....</b>	<b>237</b>
10.1 数据库设计概述 .....	237
10.2 需求分析 .....	238
10.3 概念模型设计 .....	240
10.4 逻辑模型设计 .....	246
10.5 物理结构设计 .....	251
10.6 数据运行与维护 .....	254
<b>第11章 T-SQL 高级编程 .....</b>	<b>256</b>
11.1 T-SQL 编程基础 .....	256
11.2 批处理与流程控制 .....	263
11.3 游标 .....	270
11.4 存储过程 .....	273
11.5 触发器 .....	279

<b>第 12 章 数据库系统开发实训</b> .....	283
12.1 基于计算思维的系统开发概述 .....	283
12.2 学生信息管理系统需求 .....	284
12.3 系统实现 .....	287
<b>附录 A SQL Server 2012 安装及操作</b> .....	295
A.1 SQL Server 2012 的版本与组件 .....	295
A.2 SQL Server 2012 的安装与配置 .....	297
A.3 数据库的创建与操作 .....	315
<b>附录 B SQL Server 2012 常用的系统函数及使用</b> .....	319
B.1 SQL Server 2012 的系统函数概述 .....	319
B.2 字符串函数 .....	320
B.3 日期函数 .....	321
B.4 其他函数 .....	322
<b>附录 C 实验指导</b> .....	324
C.1 利用管理工具创建数据库对象 .....	324
C.2 利用 SQL 语句创建数据库对象 .....	325
C.3 数据库单表查询 .....	325
C.4 数据库多表连接查询 .....	326
C.5 数据库嵌套和组合查询 .....	327
C.6 数据的更新 .....	328
C.7 实现存储过程 .....	329
C.8 实现触发器 .....	329
<b>参考文献</b> .....	331

# 第1章 数据库概述

数据库是管理数据的一种技术，现在数据库技术已经被广泛地应用到我们日常生活中的方方面面。本章首先介绍数据管理技术发展的过程，然后介绍使用数据库技术管理数据的特点和好处。

## 1.1 概述

随着信息管理水平的不断提高，应用范围的日益扩大，信息已成为企业的重要财富和资源，同时，作为管理信息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。人们在不知不觉中扩展着对数据库的使用，比如信用卡购物，飞机或火车订票系统，商场的进销存管理，图书馆对书籍及借阅的管理等，无一不使用了数据库技术。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)、地理信息系统等，数据库技术已经渗透到我们日常生活中的方方面面，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。

数据库是数据管理的最新技术，其主要研究内容是如何对数据进行科学的管理，以提供可共享、安全、可靠的数据。数据库技术一般包含数据管理和数据处理两部分。

数据库系统本质上是一个用计算机存储数据的系统，数据库本身可以看作一个电子文件柜，但它的功能不仅仅只是保存数据，而且还提供了对数据进行各种管理和处理的功能，比如安全管理、数据共享的管理、数据查询处理等。

本章将介绍数据库的基本概念，包括数据管理的发展过程、数据库系统的组成等。读者可从本章了解为什么要学习数据库技术，并为后续章节的学习做好准备。

## 1.2 基本概念

在系统地介绍数据库技术之前，首先介绍数据库中最常用的一些术语和基本概念。

### 1.2.1 数据

数据(Data)是数据库中存储的基本对象。早期的计算机系统主要应用于科学计算领域，处理的数据基本是数值型数据，因此数据在人们头脑中的直觉反应就是数字，但数字只是数据的一种最简单的形式，是对数据的传统和狭义的理解。目前计算机的应用范围已十分广泛，因此数据种类也更加丰富，比如，文本、图形、图像、音频、视频、商品销售情况等都是数据。

可以将数据定义为,数据是描述事物的符号记录。描述事物的符号可以是数字,也可以是文字、图形、图像、声音、语言等,数据有多种表现形式,它们都可以经过数字化后保存在计算机中。

数据的表现形式并不一定能完全表达其内容,有些还需要经过解释才能明确其表达的含义。比如20,当解释其代表人的年龄时就是20岁;当解释其代表商品价格时,就是20元。因此,数据和数据的解释是不可分开的。数据的解释是对数据演绎的说明,数据的含义称为数据的语义。因此数据和数据的语义是不可分开的。

在日常生活中,人们一般直接用自然语言来描述事物,例如描述一门课程的信息:数据库系统原理与技术A,3个学分,第5学期开设。但在计算机中经常按如下形式描述:

(数据库原理与技术A,3,5)

即把课程名、学分、开课学期信息组织在一起,形成一个记录,这个记录就是描述课程的数据。这样的数据是有结构的。记录是计算机表示和存储数据的一种格式或方法。

## 1.2.2 数据库

数据库(Database, DB),顾名思义,就是存放数据的仓库。只是这个仓库是存储在计算机存储设备上的,而且是按一定的格式存储的。

人们在收集并取出某个应用所需要的大量数据之后,就希望将这些数据保存起来,以供进一步从中得到有价值的信息,并进行相应的加工和处理。在科学技术飞速发展的今天,人们对数据的需求越来越多,数据量也越来越大。最早人们把数据存放在文件柜里,现在人们可以借助计算机和数据库技术来科学地保存和管理大量的复杂数据,以便能方便而充分地利用宝贵的数据资源。

严格地讲,数据库是长期存储在计算机中的有组织的、可共享的大量数据的集合。数据库中的数据按一定的数据模型组织、描述和存储,具有较小的数据冗余、较高的数据独立性和易扩展性,并可为多种用户共享。

概括起来,数据库数据具有永久存储、有组织和可共享3个基本特点。

## 1.2.3 数据库管理系统

在了解了数据和数据库的基本概念之后,下一个需要了解的就是如何科学有效地组织和存储数据,如何从大量的数据中快速地获得所需的数据以及如何对数据进行维护,这些都是数据库管理系统(Database Management System, DBMS)要完成的任务。数据库管理系统是一个专门用于实现对数据进行管理和维护的系统软件。

数据库管理系统位于用户应用程序与操作系统软件之间,如图1-1所示。数据库管理系统与操作系统一样都是计算机的基础软件,同时也是一个非常复杂的大型系统软件,其主要功能包括如下几个方面:

(1) 数据库的建立与维护功能。包括创建数据库及对数据库空间的维护,数据库的转储与恢复功能,数据库的重组功能,数据

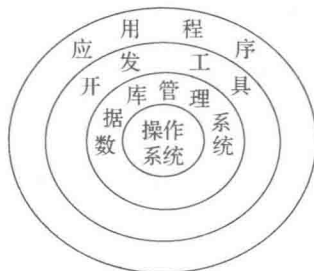


图 1-1 数据库管理系统在计算机系统的位置

库的性能监视与调整功能等。这些功能一般是通过数据库管理系统中提供的一些实用工具实现的。

(2) 数据定义功能。包括定义数据库中的对象,比如表、视图、存储过程等。这些功能的实现一般是通过数据库管理系统提供的数据库定义语言(Data Definition Language, DDL)实现的。

(3) 数据组织、存储和管理功能。为提高数据的存取效率,数据库管理系统需要对数据进行分类存储和管理。数据库中的数据包括数据字典、用户数据和存取路径数据等。数据库管理系统要确定这些数据的存储结构、存取方式以及存储位置,以及如何实现数据之间的关联。确定数据的组织和存储的主要目的是提高存储空间利用率和存取效率。一般的数据库管理系统都会根据数据的具体组织和存储方式提供多种数据存取方法,比如索引查找、Hash查找、顺序查找等。

(4) 数据操作功能。包括对数据库数据的查询、插入、删除和更改操作,这些操作一般是通过数据库管理系统提供的数据库操作语言(Data Manipulation Language, DML)实现的。

(5) 事务的管理和运行功能。数据库中的数据是可供多个用户同时使用的共享数据,为保证数据能够安全、可靠地运行,数据库管理系统提供了事务管理功能,这些功能保证数据能够并发使用并且不会产生相互干扰的情况,而且在数据库发生故障时能够对数据库进行正确的恢复。

(6) 其他功能。包括与其他软件的网络通信功能、不同数据库管理系统间的数据传输以及互访问功能等。

#### 1.2.4 数据库系统

数据库系统(Database System, DBS)是指在计算机中引入数据库后的系统,一般由数据库、数据库管理系统(及相关的实用工具)、应用程序、数据库管理员组成。为保证数据库中的数据能够正常、高效地运行,除了数据库管理系统软件之外,还需要一个(或一些)专门人员来对数据库进行维护,这个专门人员就称为数据库管理员(Database Administrator, DBA)。我们将在1.5节详细介绍数据库系统的组成。

一般在不引起混淆的情况下,常常把数据库系统简称为数据库。

### 1.3 数据技术的发展

数据库技术是应数据管理任务的需要而产生和发展的一门技术。数据管理是指对数据进行分类、组织、编码、存储、检索和维护,它是数据处理的核心,而数据处理则是指对各种数据的收集、存储、加工和传播等一系列活动的总和。

自计算机产生之后,人们就希望用它来帮助我们对数据进行存储和管理。最初对数据的管理是以文件方式进行的,也就是用户通过编写应用程序来实现对数据的存储和管理。后来,随着数据量越来越大,人们对数据的要求越来越多,希望达到的目的也越来越复杂,文件管理方式已经很难满足人们对数据的需求,由此产生了数据库技术,也就是用数据库来存储和管理数据。数据管理技术的发展因此也就经历了人工管理、文件管理和数据库管理3个阶段。

下面将介绍文件管理和数据库管理在管理数据上的主要差别。

### 1.3.1 人工管理阶段

这一阶段 (20 世纪 50 年代中期以前), 计算机主要用于科学计算。外部存储器只有磁带、卡片和纸带等, 还没有磁盘等直接存取存储设备。软件只有汇编语言, 尚无数据管理方面的软件。

数据处理方式基本是批处理。这个阶段有如下几个特点:

(1) 计算机系统不提供对用户数据的管理功能。用户编制程序时, 必须全面考虑好相关的数据, 包括数据的定义、存储结构以及存取方法等。程序和数据是一个不可分割的整体。数据脱离了程序就无任何存在的价值, 数据无独立性。

(2) 数据不能共享。不同的程序均有各自的数据, 这些数据对不同的程序通常是不相同的, 不可共享; 即使不同的程序使用了相同的一组数据, 这些数据也不能共享, 程序中仍然需要各自加入这组数据, 谁也不能省略。基于这种数据的不可共享性, 必然导致程序与程序之间存在大量的重复数据, 浪费了存储空间。

(3) 不单独保存数据。基于数据与程序是一个整体, 数据只为本程序所使用, 数据只有与相应的程序一起保存才有价值, 否则就毫无用处。所以, 所有程序的数据均不单独保存。

### 1.3.2 文件管理阶段

20 世纪 50 年代后期到 60 年代中期, 计算机的硬件方面已经有了磁盘等直接存取的存储设备, 软件方面, 操作系统中已经有了专门的数据管理软件, 一般称为文件管理系统。文件管理系统把数据组织成相互独立的数据文件, 利用“按文件名访问, 按记录进行存取”的管理技术, 可以对文件中的数据进行修改、插入和删除等操作。

在出现程序设计语言之后, 开发人员不但可以创建自己的文件并将数据保存在自己定义的文件中, 而且还可以编写应用程序来处理文件中的数据, 即编写应用程序来定义文件的结构, 实现对文件内容的插入、删除、修改和查询操作。当然, 真正实现磁盘文件的物理存取操作的还是操作系统中的文件管理系统, 应用程序只是告诉文件管理系统对哪个文件的哪些数据进行哪些操作。我们将由开发人员定义存储数据的文件及文件结构, 并借助文件管理系统的功能编写访问这些文件的应用程序, 以实现用户对数据的处理的方式称为文件管理。在本章后面的讨论中, 为描述简单我们将忽略操作系统中的文件管理系统, 假定应用程序是直接对磁盘文件进行操作。

用户通过编写应用程序来管理存储在自定义文件中的数据的操作模式如图 1-2 所示。

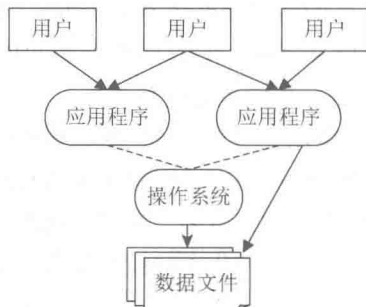


图 1-2 用文件存储数据的操作模式

假设某学校要用文件的方式保存学生及其选课的数据，并针对这些数据文件构建对学生及选课情况进行管理的系统。此系统主要实现两部分功能：学生基本信息管理和学生选课情况管理。假设教务部门管理学生选课情况，各系管理自己的学生基本信息。学生基本信息管理只涉及学生的基本信息数据，假设这些数据保存在 F1 文件中。学生选课情况管理涉及学生的部分基本信息、课程基本信息和学生选课信息，假设文件 F2 和 F3 分别保存课程基本信息和学生选课信息的数据。

设 A1 为实现“学生基本信息管理”功能的应用程序，A2 为实现“学生选课管理”功能的应用程序。图 1-3 为用文件存储并管理数据的两个系统的实现示例（图中省略了操作系统部分）。

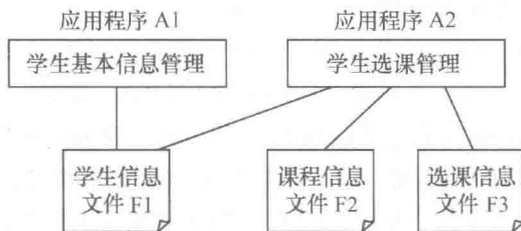


图 1-3 用文件存储并管理数据的实现示例

假设文件 F1、F2 和 F3 分别包含如下信息：

F1 文件：学号、姓名、性别、出生日期、联系电话、所在系、专业、班号。

F2 文件：课程号、课程名、授课学期、学分、课程性质。

F3 文件：学号、姓名、所在系、专业、课程号、课程名、选课类型、选课时间、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”的处理过程大致为：在学生选课管理中，若有学生选课，则先查 F1 文件，判断有无此学生；若有则再访问 F2 文件，判断其所选的课程是否存在；若一切符合规则，就将学生选课信息写到 F3 文件中。

这看似很好，对数据的管理比起人工管理阶段已经进了一大步，但仔细分析一下，就会发现用文件方式管理数据有如下缺点：

(1) 编写应用程序不方便。应用程序编写者必须清楚了解所用文件的逻辑及物理结构，如文件中包含多少个字段，每个字段的数据类型，采用何种逻辑结构和物理存储结构。操作系统只提供了打开、关闭、读、写等几个底层的文件操作命令，而对文件的查询、修改等操作都必须在应用程序中编程实现。这样就容易造成各应用程序在功能上的重复，比如图 1-3 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，而共享这两个功能相同的操作却很难。

(2) 数据冗余不可避免。由于 A2 应用程序需要学生选课信息文件（F3 文件）中包含学生的一些基本信息，比如学号、姓名、所在系、专业等，而这些信息同样包含在学生信息文件（F1 文件）中，因此 F3 文件和 F1 文件中存在重复数据，从而造成数据的重复，称为数据冗余。

数据冗余所带来的问题不仅仅是存储空间的浪费（其实，随着计算机硬件技术的飞速发展，存储容量不断扩大，空间问题已经不是我们关注的主要问题），更为严重的问题是造成了数据的不一致（inconsistency）。例如，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 文件中应做同样的修改。由此就造成了同一名学生在 F1 文件和 F3 文件中的“专业”不一样，也就是数据不一致。当发生数据不一致时，人们不能判定哪个数据是正确的，尤其是当系统中存在多处数据冗余时，更是如此。这样数据就失去了其可信性。

文件本身并不具备维护数据一致性的功能，这些功能完全要由用户（应用程序开发者）负责维护。这在简单的系统中还可以勉强应对，但在复杂的系统中，若让应用程序开发者来保证数据的一致性，几乎是不可能的。

(3) 应用程序依赖性。就文件管理而言，应用程序对数据的操作依赖于存储数据的文件的结构。定义文件和记录的结构通常是应用程序代码的一部分，如 C 程序的 struct。文件结构的每一次修改，比如添加字段、删除字段，甚至修改字段的长度（如电话号码从 7 位扩到 8 位），都将导致应用程序的修改，因为在打开文件进行数据读取时，必须将文件记录中不同字段的值对应到应用程序的变量中。随着应用环境和需求的变化，修改文件的结构不可避免，这些都需要在应用程序中做相应的修改，而（频繁）修改应用程序是很麻烦的。人们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；甚至修改了文件的存储位置或者文件名，也需要对应用程序进行修改，这显然给程序的维护带来很多麻烦。

所有这些都是由于应用程序对文件的结构以及文件的物理特性过分依赖造成的，换句话说，用文件管理数据时，其数据独立性（data independence）很差。

(4) 不支持对文件的并发访问。在现代计算机系统中，为了有效利用计算机资源，一般都允许同时运行多个应用程序（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。回忆一下，某个用户打开了一个 Word 文件，当第二个用户在第一个用户未关闭此文件前打开此文件时，会得到什么信息呢？他只能以只读方式打开此文件，而不能在第一个用户打开的同时对此文件进行修改。再回忆一下，如果用某种程序设计语言编写一个对某文件中内容进行修改的程序，其过程是先以写的方式打开文件，然后修改其内容，最后再关闭文件。在关闭文件之前，不管是在其他的程序中，还是在同一个程序中都不允许再次打开此文件，这就是文件管理方式不支持并发访问的含义。

对于以数据为中心的系统来说，必须要支持多个用户对数据的并发访问，否则就不会有我们现在这么多的火车或飞机的订票点，也不会有这么多的银行营业网点。

(5) 数据间联系弱。当用文件管理数据时，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。比如对上述的 F1 文件和 F3 文件，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存在于 F2 文件中（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实际应用当中所要求的很自然的联系，但文件本身不具备自动实现这些联系的功能，我们必须通过编写应用程序，即手动建立这些联系。这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，用文件管理数据时很难反映现实世界事物间客观存在的联系。

(6) 难以满足不同用户对数据的需求。不同的用户(数据使用者)关注的的数据往往不同。例如,对于学生基本信息,负责分配学生宿舍的部门可能只关心学生的学号、姓名、性别和班号,而教务部门可能关心的是学号、姓名、所在系和专业。

若多个不同用户希望看到的是学生的不同基本信息,那么就需要为每个用户建立一个文件,这势必会造成很多的数据冗余。我们希望的是,用户关心哪些信息就为他生成哪些信息,对用户不关心的数据将其屏蔽,使用户感觉不到其他信息的存在。

可能还会有一些用户,其所需要的信息来自多个不同的文件,例如,假设各班班主任关心的是:班号、学号、姓名、课程名、学分、考试成绩等。这些信息涉及了三个文件:从F1文件中得到“班号”,从F2文件中得到“学分”,从F3文件中得到“考试成绩”;而“学号”“姓名”可以从F1文件或F3文件中得到,“课程名”可以从F2文件或F3文件中得到。在生成结果数据时,必须对从三个文件中读取的数据进行比较,然后组合成一行有意义的数据。比如,将从F1文件中读取的学号与从F3文件中读取的学号进行比较,学号相同时,才可以将F1文件中的“班号”与F3文件中的当前记录所对应的学号和姓名组合起来,之后,还需要将组合结果与F2文件中的内容进行比较,找出课程号相同的课程的学分,再与已有的结果组合起来。然后从组合后的数据中提取出用户需要的信息。如果数据量很大,涉及的文件比较多时,我们可以想象这个过程有多复杂。因此,这种复杂信息的查询,在按文件管理数据的方式中是很难处理的。

(7) 无安全控制功能。在文件管理方式中,很难控制某个人对文件能够进行的操作,比如只允许某个人查询和修改数据,但不能删除数据,或者对文件中的某个或者某些字段不能修改等。而在实际应用中,数据的安全性是非常重要的且不可忽视的。比如,在学生选课管理中,我们不允许学生修改其考试成绩,但允许他们查询自己的考试成绩。在银行系统中,更是不允许一般用户修改其存款数额。

人们对数据需求的增加,迫切需要对数据进行有效、科学、正确、方便的管理。针对文件管理方式的这些缺陷,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。

### 1.3.3 数据库管理阶段

20世纪60年代后期以来,计算机管理数据的规模越来越大,应用范围越来越广泛,数据量急剧增加,多种应用同时共享数据集成的要求也越来越强烈。

随着大容量磁盘的出现,硬件价格的不断下降,软件价格的不断上升,编制和维护系统软件和应用程序的成本相应地不断增加。在数据处理方式上,对联机实时处理的需求越来越多,同时开始提出和考虑分布式处理技术。在这种背景下,以文件方式管理数据已经不能满足应用的需求,于是出现了新的管理数据的技术——数据库技术,同时出现了统一管理数据的专门软件——数据库管理系统。

从1.3.2节的介绍我们可以看到,在数据库管理系统出现之前,人们对数据的操作是通过直接针对数据文件编写应用程序实现的,这种模式会产生很多问题。在有了数据库管理系统之后,人们对数据的操作全部是通过数据库管理系统实现的,而且应用程序的编写也不再直接针对存放数据的文件。有了数据库技术和数据库管理系统之后,人们对数据的操作模式发生了根本的变化,如图1-4所示。



图 1-4 数据库管理阶段示意图

比较图 1-2 和图 1-4，可以看到主要区别有两个：第一个是在操作系统和用户应用程序之间增加了一个系统软件——数据库管理系统，使得用户对数据的操作都是通过数据库管理系统实现的；第二个是有了数据库管理系统之后，用户不再需要有数据文件的概念，即不再需要知道数据文件的逻辑和物理结构及物理存储位置，而只需要知道存放数据的场所——数据库即可。

从本质上讲，即使在有了数据库技术之后，数据最终还是以文件的形式存储在磁盘上的（这点我们将在本书附录 A 中的创建数据库部分可以体会到），只是这时对物理数据文件的存取和管理是由数据库管理系统统一实现的，而不再是每个用户通过编写应用程序实现。数据库和数据文件既有区别又有联系，它们的关系类似于单位名称和地址之间的关系。单位地址代表了单位的实际存在位置，单位名称是单位的逻辑代表。而且一个数据库可以包含多个数据文件，就像一个单位可以有多个不同地址一样（就像我们现在的很多大学，都是一个学校有多个校址），每个数据文件存储数据库的部分数据。不管一个数据库包含多少个数据文件，对用户来说他只针对数据库进行操作，而无须对数据文件进行操作。这种模式极大地简化了用户对数据的访问。

在有了数据库技术之后，用户只需要知道存放所需数据的数据库名，就可以对数据库对应的数据文件中的数据进行操作。将对数据库的操作转换为对物理数据文件的操作是由数据库管理系统自动实现的，用户不需要知道，也不需要干预。

对于 1.3.2 中列举的学生基本信息管理和学生选课管理两个子系统，如果使用数据库技术来实现，其实现方式如图 1-5 所示。

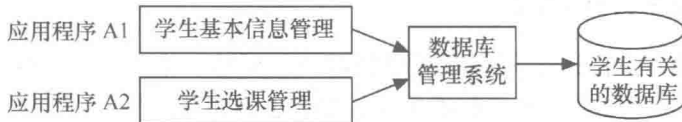


图 1-5 用数据库存储数据的实现示例

与用文件管理数据相比，用数据库技术管理数据具有以下特点：

(1) 相互关联的数据集合。在用数据库技术管理数据时，所有相关的数据都被存储在一个数据库中，它们作为一个整体定义，因此可以很方便地表达数据之间的关联。比如学生基本信息中的“学号”与学生选课管理中的“学号”，这两个学号之间是有关联的，即学生选课中的“学号”的取值范围在学生基本信息的“学号”取值范围内。在关系数据库中，数据之间的关联是通过参照完整性实现的。

(2) 较少的数据冗余。由于数据是被统一管理的，因此可以从全局着眼，对数据进行最合理的组织。例如，将 1.3.1 节中文件 F1、F2 和 F3 的重复数据挑选出来，进行合理的管

理,这样就可以形成如下所示的几部分信息:

学生基本信息:学号、姓名、性别、出生日期、联系电话、所在系、专业、班号。

课程基本信息:课程号、课程名、授课学期、学分、课程性质。

学生选课信息:学号、课程号、选课类型、选课时间、考试成绩。

在关系数据库中,可以将每一类信息存储在一个表中(关系数据库的概念将在后边介绍),重复的信息只存储一份,当在学生选课中需要学生的姓名等其他信息时,根据学生选课中的学号,可以很容易地在学生基本信息中找到此学号对应的姓名等信息。因此,消除数据的重复存储不影响对信息的提取,同时还可以避免由于数据重复存储而造成的数据不一致问题。比如,当某个学生所学的专业发生变化时,只需在“学生基本信息”一个地方进行修改即可。

同1.3.2节中的问题一样,当所需的信息来自不同的地方,比如班号、学号、姓名、课程名、学分、考试成绩等信息,这些信息需要从3个地方(关系数据库为3张表)得到,这种情况下,也需要对信息进行适当的组合,即学生选课中的学号只能与学生基本信息中学号相同的信息组合在一起,同样,学生选课中的课程号也必须与课程基本信息中课程号相同的信息组合在一起。过去在文件管理方式中,这个工作是由开发者编程实现的,而现在有了数据库管理系统,这些烦琐的工作完全交给了数据库管理系统来完成。

因此,在用数据库技术管理数据的系统中,避免数据冗余不会增加开发者的负担。在关系数据库中,避免数据冗余是通过关系规范化理论实现的。

(3) 程序与数据相互独立。在数据库中,组成数据的数据项以及数据的存储格式等信息都与数据存储在一起,它们通过DBMS而不是应用程序来操作和管理,应用程序不再需要处理文件和记录的格式。

程序与数据相互独立有两方面的含义:一方面是当数据的存储方式发生变化时(这里包括逻辑存储方式和物理存储方式),比如从链表结构改为散列结构,或者是顺序存储和非顺序存储之间的转换,应用程序不必做任何修改;另一方面是当数据所包含的数据项发生变化时,比如增加或减少了一些数据项,如果应用程序与这些修改的数据项无关,则不用修改应用程序。这些变化都将由DBMS负责维护。大多数情况下,应用程序并不知道也不需要知道数据存储方式或数据项已经发生了变化。

在关系数据库中,数据库管理系统通过将数据划分为三个层次来自动保证程序与数据相互独立。我们将介绍数据的三个层次,也称为三级模式结构。

(4) 保证数据的安全和可靠。数据库技术能够保证数据库中的数据是安全的和可靠的。它的安全控制机制可以有效地防止数据库中的数据被非法使用和非法修改;其完整的备份和恢复机制可以保证当数据遭到破坏时(由软件或硬件故障引起的)能够很快地将数据库恢复到正确的状态,并使数据不丢失或只有很少的丢失,从而保证系统能够连续、可靠地运行。保证数据的安全是通过数据库管理系统的安全控制机制实现的,保证数据的可靠是通过数据库管理系统的备份和恢复机制实现的。

(5) 最大限度地保证数据的正确性。数据的正确性也称为数据的完整性,它是指存储到数据库中的数据必须符合现实世界的实际情况,比如人的性别只能是“男”和“女”,人的年龄应该在0~150之间(假设没有年龄超过150岁的人)。如果在性别中输入了其他值,或者将一个负数输入年龄中,在现实世界中显然是不对的。数据的正确性是通过在数据库中