



进销存程序设计



清华大学出版社

Delphi 进销存程序设计

陈惟彬 编著

清华大学出版社

(京)新登字 158 号

内 容 简 介

Delphi 以其强大的功能,在众多基于 Windows 的开发工具中占有一席之地。本书以零售业进销存系统为专案,介绍了如何利用 Delphi 设计具有网络功能的商用系统。

主要包括:系统分析的方法及实践;客户机/服务器程序设计;零售业进销存系统实践,包含了进货、出货、库存管理、应收与应付账款的管理等。本书不讨论艰深的技术,只要掌握 Delphi 基本操作,通过本书的学习,都能循序渐进地设计出完善而实用的数据库管理系统。

本书内容详实,图文并茂,可帮助 Delphi 数据库开发人员深入学习商用软件的开发技巧,也可作为其他数据库设计人员的参考书籍。

本书繁体字版名为《Delphi 进销存程序设计》,由文魁资讯股份有限公司出版,版权属陈惟彬所有。本书中文简体字版由文魁资讯股份有限公司授权清华大学出版社独家出版。未经本书原版出版者和本书出版者书面许可,任何单位和个人均不得以任何形式或任何手段复制或传播本书的部分或全部内容。

北京市版权局著作权合同登记号:图字 01-2002-2550 号

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

Delphi 进销存程序设计/陈惟彬编著. —北京:清华大学出版社, 2002
ISBN 7-302-05956-X

I. D... II. 陈 III. DELPHI 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2002)第 076865 号

出 版 者:清华大学出版社(北京清华大学学研大厦,邮编 100084)

<http://www.tup.tsinghua.edu.cn>

责任编辑:刘学媛

印 刷 者:北京密云胶印厂

发 行 者:新华书店总店北京发行所

开 本:787×1092 1/16 印张:22.75 字数:538 千字

版 次:2002 年 10 月第 1 版 2002 年 10 月第 1 次印刷

书 号:ISBN 7-302-05956-X/TP·3545

印 数:0001~5000

定 价:35.00 元

前 言

Delphi 是 Borland 公司开发的基于 Pascal 语言的 Windows 平台应用程序开发工具，它将可视化编程与面向对象的卓越优点结合在 Pascal 语言开发平台上，不仅能开发一般的 Windows 应用程序，而且还具有强大的数据库应用程序开发功能，可以支持客户机/服务器模式的 SQL 数据库访问与 ODBC，是一个真正面向对象的开发工具。

Delphi 优化了对数据库的操作，包括了广泛的数据库支持，提供了数据库引擎 (Borland Database Engine, 简称 BDE)，使得 SQL 语言更为有效、快速。Delphi 最显著的特点是其强大的数据库访问能力，它主要通过使用数据库引擎来访问本地数据库和远程数据库，同时还提供了对 ODBC、ADO 等数据库驱动程序的支持，能够满足不同应用对数据库程序开发的需要。

Windows 环境与早期的 DOS 环境相比，前者的复杂度绝对超过后者。相当多在 DOS 时期即从事数据库研发行列的程序员，在转换到现在的 Windows 环境时，仍有一些不适应感，导致他们所开发出来的系统很像 DOS。本书作者从分析数据库系统的架构入手，认为没有必要把每套数据库应用系统都做得这么复杂艰深。一些规模不大的使用单位，或者预算有限的中小企业，可以利用比较简单的客户机/服务器架构，去规划并实践数据库管理系统，而不必采用较为复杂的 N 层体系架构。

本书以浅显的数据库应用程序的系统分析为起点，首先让读者对“系统分析”有一个基本概念。接着，以一些简单的案例，由浅入深地介绍用 Delphi 开发数据库应用程序的方法，使读者在实践中熟悉 Delphi 在数据库应用程序设计上的架构。最后，以零售业进销存管理系统为主题，用 Delphi 来把这个专案给做出来。

全书共分 16 章，内容分别为：

第 1 章：数据库概述

第 2 章：系统分析方法及实例

第 3 章：Delphi 数据库程序设计初步

第 4 章：深入客户数据管理程序

第 5 章：客户机/服务器架构程序设计

第 6 章：SQL 语言

第 7 章：客户机/服务器程序设计

第 8 章：一对多数据维护程序设计

第 9 章到第 16 章为零售业进销存管理实践，包括基本数据的建立、进货操作、出货操作、库存管理、付款冲账作业与收款冲账作业、系统管理、报表设计等内容。

书中对 Delphi 数据库程序中的各类细节均加以详述，例如访问数据库的常用方法——通过 BDE 数据库搜索引擎、通过 ODBC 等，书中对这两种方法的运行速度、应用范围、接口特性等加以分析。另外在介绍 SQL 指令过程中，通过多种范例说明了查询优化的重要性，引导读者尽量使用标准化语句。

对于客户机/服务器架构程序设计，本书采用大量篇幅，以真正实际的专案，从无到有、

由浅入深的来介绍如何利用 Delphi 的优势特色,来实践出可上线使用的商用数据管理系统。书中不讨论艰深的程序设计技巧,而是以任务带功能方式进行讲解,从完成任务入手,帮助开发人员深入学习商用软件的开发细节,并使读者对 Delphi 程序设计的本质有更深入的了解。

本书面向具有一定 Delphi 程序设计基础的读者,以及想深入学习商用数据管理系统开发的程序设计人员。书中提供了大量规范化的图例及程序代码,使读者能掌握规范化的设计模式及开发流程,进而设计出适合实际需求的数据库管理系统。

编者

2002 年 8 月

目 录

第 1 章 数据库概述..... 1	4.3 数据的搜索..... 66
1.1 数据库的规范化..... 1	4.4 豪华版客户数据管理系统..... 68
1.1.1 数据库概念..... 1	第 5 章 客户机/服务器架构程序设计
1.1.2 数据库的规范化..... 1	——与数据库服务器的连接..... 81
1.2 数据库系统的结构..... 5	5.1 使用 SQL Link 来连接 Microsoft
1.2.1 主机集中型数据库系统..... 6	SQL Server..... 81
1.2.2 文件型数据库系统..... 6	5.2 用 ODBC 连接 MySQL..... 82
1.2.3 客户机/服务器结构型	5.3 写一个简单的程序连接数据库..... 83
数据库系统..... 8	5.4 TDatabase..... 84
1.2.4 多层式结构型数据库系统..... 9	5.5 更多的数据..... 86
第 2 章 系统分析方法及实例..... 12	5.5.1 把范例数据准备妥当..... 86
2.1 产生需求..... 12	5.5.2 股票历史数据查询程序..... 88
2.2 搜集资料..... 13	5.6 SQL Monitor..... 90
2.3 深入讨论及实地了解..... 17	第 6 章 SQL 语言..... 91
2.4 制订完整的系统规格..... 18	6.1 SQL 语言基本介绍..... 91
2.4.1 制订系统规格的步骤..... 18	6.2 数据库处理语言..... 91
2.4.2 系统规格实例..... 21	6.2.1 SELECT——查询指令..... 92
2.5 系统开发..... 27	6.2.2 INSERT——新增
2.6 系统测试及验收..... 27	数据指令..... 99
2.7 零售业进销存系统的文件规格..... 28	6.2.3 UPDATE——更新
2.8 进销存管理系统的数据关联图..... 43	数据指令..... 100
第 3 章 Delphi 数据库程序设计初步..... 48	6.2.4 DELETE——删除
3.1 第一个数据库应用程序..... 48	数据指令..... 100
3.2 Delphi 数据库应用程序的架构..... 49	第 7 章 客户机/服务器程序设计..... 101
3.3 再探客户数据管理程序..... 51	7.1 出货单管理系统..... 101
3.3.1 BDE Administrator..... 52	7.2 TDataModule 数据模块..... 103
3.3.2 组件..... 54	7.3 产品数据维护..... 105
第 4 章 深入客户数据管理程序..... 57	7.4 更新版的产品数据维护程序..... 108
4.1 自定义专属的 DBNavigator..... 57	7.5 改良版的产品数据维护程序..... 109
4.2 深入 TDataSet..... 63	7.6 最终版的产品数据维护程序..... 113
4.2.1 TDataSet 的事件..... 63	7.7 数据的交易控制..... 114
4.2.2 TField 字段对象..... 64	7.8 数据的错误处理..... 117

7.9 数据的更新	124	12.2 实践 DL110 出货单维护程序	263
第 8 章 一对多数据维护程序设计	132	12.3 数据筛选	273
8.1 TTable 版的出货单维护	132	第 13 章 零售业进销存管理系统	
8.2 完善 TTable 版的出货单维护程序	136	实践——库存管理	277
8.3 TQuery 版的出货单维护程序	147	13.1 期初开账操作	277
8.4 TQuery 版的出货单维护程序-改良版	155	13.2 IN110 存货异动单维护	279
第 9 章 零售业进销存管理系统实践	167	13.3 IN120 库存调拨单维护	285
9.1 应用程序的框架	167	第 14 章 零售业进销存管理系统	
9.2 主画面	168	实践——付款冲账操作	
9.3 TfmSimple	175	与收款冲账操作	289
9.4 TfmSingle	194	14.1 收付款冲账操作概论	289
9.5 TfmMasterDetail	200	14.2 AP110 付款冲账操作	290
9.6 TfmBaseDialog	215	14.3 AR110 收款冲账操作	296
9.7 TfmBaseSearch	217	第 15 章 零售业进销存管理系统	
9.8 如何快速开发应用程序	219	实践——系统管理	301
第 10 章 零售业进销存管理系统实践——基本数据的建立	224	15.1 SY110 用户数据维护	301
10.1 版权画面	224	15.2 SY120 用户权限设置与 SY130 程序使用权限设置	304
10.2 基本代码文件的数据维护程序	225	15.3 SY140 用户密码变更	306
10.3 BA110 业务员数据维护	227	15.4 SY150 公司文件维护	308
10.4 BA120 客户数据维护	230	15.5 SY990 月结操作	309
10.5 BA130 供货商数据维护	237	第 16 章 零售业进销存管理系统	
10.6 BA140 商品数据维护	241	实践——报表设计	328
第 11 章 零售业进销存管理系统实践——进货操作	245	16.1 报表底稿	328
11.1 进货操作的设计重点	245	16.2 条列式报表	332
11.2 实践 PR110 进货单维护程序	245	16.3 排行榜	333
11.3 数据筛选	259	16.4 商品低于安全存量表	334
第 12 章 零售业进销存管理系统实践——出货操作	263	16.5 年统计表	336
12.1 出货单设计时的注意事项	263	16.6 AR230 应收账款统计表	338
		16.7 AR250 应收账款账龄分析表	341
		16.8 AR260 逾期应收账款统计表	345
		16.9 N270 商品进销存汇总表	349

第 1 章 数据库概述

本章将介绍如何利用数据库规范化的方法，来设计出一个好的数据库。另外，也将认识各种不同的数据库运算方式，了解各种不同运算方式的数据库之间，它们的处理方式与差异性。

1.1 数据库的规范化

1.1.1 数据库概念

我们每天的生活都与数据息息相关。我们有很多的亲朋好友，如果没有一个通讯录，可能无法牢记所有好朋友的电话、地址或生日。如果需要得到客户每月的交易额，就必须去查出客户的交易记录。对于零售业而言，需要有完整的销售数据来统计分析各商品的销售金额，也需要有一个最新、最正确的库存数据来随时掌握最新的存货状况，如果目前仓库的库存低于安全库存，就必须赶紧向供货商订货。

每一位亲友的电话、地址、生日，每一家客户的销售记录，每一项商品的销售状况，所有仓库的商品存货信息，这些组合起来就是数据。

数据库，简单而言，就是要存储这些大量的数据，而且要能正确的存储。

数据会随着时间的增加，而积累，导致数据量日渐庞大。我们常常需要快速地在大量的数据中，找出某客户、在某段期间的销售记录与合计的销售金额，同时也希望数据能有一定的保密性。

而数据的最终作用，就是提供决策分析的信息。如果我们不能利用大量数据的分析，找出最有潜力的客户或商品，则数据只是简单的资料罗列而已。

由最基本的元素组合成一笔数据叫记录(Record)。由相同性质的记录组合起来叫表(Table)，相关的表组合起来就叫做数据库(Database)。处理数据的建立、查询、更新等工作的程序就叫做数据库管理系统(Database Management System, DBMS)。以销售表一(表 1.1)为例，纵向的每一列叫做字段，横向组合起来就成为一条记录，所有记录的集合就是表，与其他相关的表组合起来就是数据库。

1.1.2 数据库的规范化

为了让数据库的存取效率高并且便于维护，数据库的设计就占有很重要的地位。数据库规范化的目的就是要让我们设计出来的数据库能够很有效率的运行并能合乎逻辑的维护。让我们来以一家计算机公司的销售数据来了解数据库规范化的方法。

表 1.1 销售表一

销售日期	销货单号	客户简称	商品名称	数 量	单 价	金 额
1990/05/02	90050001	永业贸易	计算机主机	1	32,000	32,000
1990/05/02	90050001	永业贸易	打印机	1	10,200	10,200
1990/05/02	90050002	德昶食品	打印机	1	10,200	10,200
1990/05/08	90050003	永业	打印机耗材	10	870	8,700
1990/05/09	90050004	家革皮鞋	计算机主机	3	31,000	93,000
1990/05/15	90050005	德昶食品	打印机耗材	15	870	13,050

表 1.1 是某计算机公司的销售记录。我们发现，在这份销售记录中，“客户简称”这一数据有重复的，也就是说，一家客户可能会有一笔以上的销售记录。我们把客户简称独立出来成一个客户表，并给每家客户一个独一无二的编号。为每家客户编号的目的是为了输入方便并保持数据一致。譬如说，像永业与永业贸易，实际上应该是同一家公司。但这对计算机而言，却是不同的公司，只是因为数据输入人员对简称的取名方式不同而有不同的结果。如果我们统一输入客户编号，就不会有这样的事故发生。而且，在实际操作中，输入客户编号比输入客户简称要来得方便、迅速。下面是我们把销售记录表一(表 1.1)分割成客户表(表 1.2)与销售表二(表 1.3)。

表 1.2 客户表

客户编号	客户简称
00001	永业贸易
00002	家革皮鞋
00003	德昶食品

表 1.3 销售表二

销售日期	销货单号	客户编号	商品名称	数 量	单 价	金 额
1990/05/02	90050001	00001	计算机主机	1	32,000	32,000
1990/05/02	90050001	00001	打印机	1	10,200	10,200
1990/05/02	90050002	00003	打印机	1	10,200	10,200
1990/05/08	90050003	00001	打印机耗材	10	870	8,700
1990/05/09	90050004	00002	计算机主机	3	31,000	93,000
1990/05/15	90050005	00003	打印机耗材	15	870	13,050

在客户表中，客户编号是不允许重复的。像客户编号这种不许重复的字段，我们称它为主键字段。当我们在设计每个表时，一定要尽可能找出像客户编号这种不允许重复的字段出来。

而客户表与销售表二之间的关系，是通过客户编号这字段来连接。在销售表二中，客户编号这个字段就叫做关联字段，它参考了客户表中的客户编号这个字段。销售表二的客户编号，必须存在于客户表。如果客户编号 00001 已有销售数据，那么，我们不能从客户

表中把客户编号为 00001 的数据给删除, 否则在销售表二中找不到客户编号 00001 的客户数据。换言之, 客户可以不必拥有销售记录, 但是有销售记录的客户必须存在于客户表中。这就是关系型数据库名称的由来。

再来看看销售表二, 我们发现, 商品名称仍有重复, 每项商品都不会只有一笔的销售记录。按照分割客户表的方式, 我们把销售表拆成商品表(表 1.4)与销售表三(表 1.5)。商品表中的商品编号是主键字段, 销售表三与商品表通过商品编号这个字段来关联, 也就是说, 销售表三的商品编号是关联字段, 参考了商品表的商品编号这个字段。

表 1.4 商品表

商品编号	商品名称
P0001	计算机主机
P0002	打印机
P0003	打印机耗材

表 1.5 销售表三

销售日期	销货单号	客户编号	商品编号	数 量	单 价	金 额
1990/05/02	90050001	00001	P0001	1	32,000	32,000
1990/05/02	90050001	00001	P0002	1	10,200	10,200
1990/05/02	90050002	00003	P0002	1	10,200	10,200
1990/05/08	90050003	00001	P0003	10	870	8,700
1990/05/09	90050004	00002	P0001	3	31,000	93,000
1990/05/15	90050005	00003	P0003	15	870	13,050

再来看看销售表三, 我们发现销货单号有重复, 也就是说, 同一张销货单会有多笔的销货明细。于是我们把销货表三的销货日期、销货单号与客户编号分割成销货主表(表 1.6), 其余字段就是销货明细表(表 1.7)。销货单号是销货主表的主键字段。而销货主表与销货明细表是通过销货单号来关联, 所以销货明细表的销货单号是关联字段, 参考了销货主表的销货单号。

表 1.6 销货主表

销售日期	销货单号	客户编号
1990/05/02	90050001	00001
1990/05/02	90050002	00003
1990/05/08	90050003	00001
1990/05/09	90050004	00002
1990/05/15	90050005	00003

销货主表中的销售日期虽然有重复的情形, 但参考销售主表——日期(表 1.8)与销售主表——单号(表 1.9), 把销售日期分割出来对数据存储空间而言是不减反增的, 而且维护上也不见得便利, 更看不出什么好处, 所以我们不必这么分割。

表 1.7 销货明细表

销货单号	商品编号	数 量	单 价	金 额
90050001	P0001	1	32,000	32,000
90050001	P0002	1	10,200	10,200
90050002	P0002	1	10,200	10,200
90050003	P0003	10	870	8,700
90050004	P0001	3	31,000	93,000
90050005	P0003	15	870	13,050

表 1.8 销货主表——日期

销售日期
1990/05/02
1990/05/08
1990/05/09
1990/05/15

表 1.9 销货主表——单号

销售日期	销货单号	客户编号
1990/05/02	90050001	00001
1990/05/02	90050002	00003
1990/05/08	90050003	00001
1990/05/09	90050004	00002
1990/05/15	90050005	00003

销货明细表的金额是单价乘以数量，可以视情况将该字段省略。表 1.10 是把金额字段省略后销货明细表二。

表 1.10 销货明细表二

销货单号	商品编号	数 量	单 价
90050001	P0001	1	32,000
90050001	P0002	1	10,200
90050002	P0002	1	10,200
90050003	P0003	10	870
90050004	P0001	3	31,000
90050005	P0003	15	870

现在，看看客户表(表 1.2)、商品表(表 1.4)、销货主表(表 1.6)、销货明细表二(表 1.10)，找不到可以再分割的字段了。所以，对于销售数据(表 1.1)的规范化工作，到此告一段落。

数据库规范化要遵循这样一个原则，就是：表与表之间，只可以存在着一对一的关系或者是一对多的关系。当我们发现在两个表间存在着多对多的关系时，就代表我们数据库的规范化工作还没完成。我们必须试着在造成多对多关系的两个表之间，再分割出来一个以上的表，来达到一对多或一对一的目的。

总之，数据库的规范化，主要有下列目的：

- 节省磁盘的存储空间。
- 便于数据维护。例如，当我们要把商品编号 P0001 的商品名称由计算机主机改成商用计算机主机时，如果没有商品表(表 1.4)，我们就必须在销货表二中逐笔修改。反之，我们只要在商品表中把商品名称改为我们要的“商用计算机主机”，再通过销货表三的关联字段商品编号，来参考到商品表中的商品编号，我们就能从商品表中找出这个新的商品名称了。

要做好数据库的规范化，要把握最重要的 3 项要领：

- 表与表之间必须是以一对一或一对多的关系存在。如果发现有多对多的关系，则需要再做进一步的分割。
- 每个表中都不能有重复的记录。
- 相同性质的字段放在同一个表中。

图 1.1 就是经我们规范化后的数据库的销售数据库关联图。

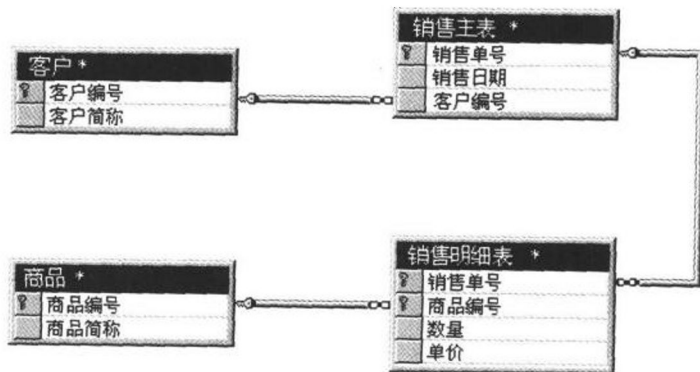


图 1.1 销售数据库关联图

客户表与销售主表是一对多的关系，商品表与销售明细表是一对多的关系，销售主表与销售明细表是一对多的关系。由这个关联图来看，我们不难发现，关系型数据库与实际操作是很接近的。这家计算机公司内部一定会有客户数据卡(客户表)，也一样会有销售的商品目录(商品表)，当然销售单更不可少。而销售单就是单头(销售主表)与单身(销售明细表)的组合。

1.2 数据库系统的结构

数据库系统的结构可按其数据运算及存取的方式，分为四大类型：

- 主机集中型数据库系统
- 文件型数据库系统

- 客户机/服务器(Client/Server)结构型数据库系统
- 多层式结构型数据库系统

1.2.1 主机集中型数据库系统

主机集中型数据库系统主要发展于早期个人计算机还不普及的时候。这种集中型数据库系统，是由一组大型计算机或微型计算机，再加上一群终端机组合而成，如图 1.2 所示。数据库的存储、计算、读取与应用程序的运行，全部集中在后端的主机上进行。前端的终端机只负责显示主机所传来的信息或把输入的信息返回到主机处理，有点类似家里的电视机接上游戏主机。

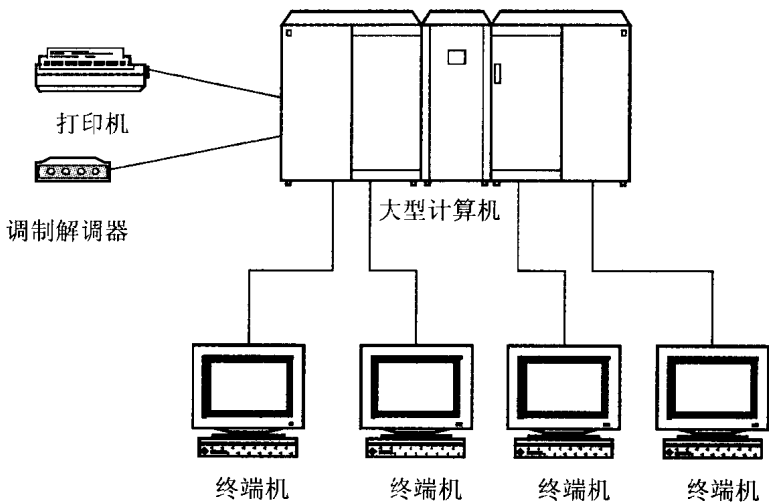


图 1.2 主机集中型数据库系统

主机集中型数据库系统的优点是，信息人员只要专心管理好这台主机，不太需要去前端的终端机做维护，所以管理上很方便。同时，所有的用户可以共享一些外围设备，如打印机、调制解调器等。在计算机还不普及的早期，打印机等计算机外设还是相当昂贵的东西。

它的缺点是，当处理的数据愈来愈多时，或同时联机到主机的终端用户愈来愈多时，主机的性能可能就无法满足需求。这时，就必须更换、添加设备或升级主机。通常这种类型主机及其外围设备是由少数的大型制造商所研发生产的，量少价高。而且因为生产数量不多，各家厂商间的硬件、软件并不完全兼容。因此调整设备所需的费用相当大，并不是一般企业所能承受的。

1.2.2 文件型数据库系统

随着 Macintosh 计算机、IBM 个人计算机的诞生，其开放性的结构、合理的价格及愈来愈强的运行性能，已为一般企业所能负担。文件型数据库系统也就在此时趋势崛起。这种类型的数据库系统，是由一台性能较好(或说较高档)的计算机当主机(即文件服务器)，然后连上几台工作站，如图 1.3 所示。这台文件服务器只负责文件的存储工作，所有的数据运算、应用程序的处理都放在前端的工作站。从早期的 DBase III，到今日的 Access 和 SQL

Server，就是一些较为知名的文件型数据库系统。

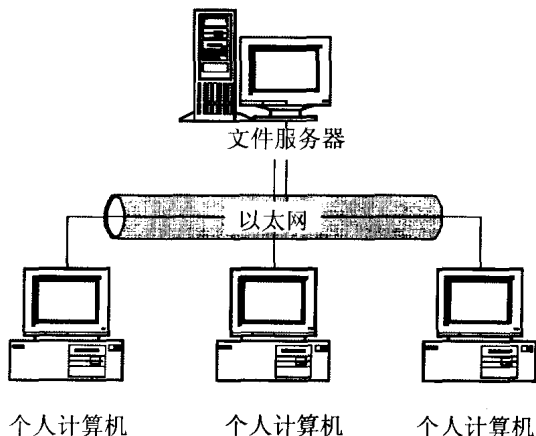


图 1.3 文件型数据库系统

文件型数据库系统最大的优点，就是价格便宜，技术普及，几乎任何的企业都负担得起。也因为它的普及与开放的结构，所以在更换设备时，就不像主机型数据库系统受限于制造商的规格。当计算机设备需要维修时，也不像大型主机一样，要找指定的计算机供货商才能维护。

由于文件型数据库系统并不提供运算的功能，故前端工作站有任何对数据读取的请求，都是通过局域网，由后端的文件服务器传至前端工作站处理；前端工作站处理完成后，再返回至后端文件服务器存储。这种结构在数据量很多时，会受网络带宽的限制，而影响运行性能；也因为它是按文件类型来进行操作，所以当有多用户要同时存取同一个数据文件时，就会有冲突或排队等候的情形发生。

例如，我们的客户表中，共有 10 000 笔客户数据。今天，我们想找出客户编号为 00001 的客户简称。文件型数据库系统对于这个需求的处理方法是：

- (1) 先把这 10 000 笔客户数据从文件服务器通过局域网络传至前端工作站。
- (2) 前端工作站再从这 10 000 笔客户数据中找出客户编号为 00001 的客户简称。
- (3) 查询操作结束后，再把这 10 000 笔数据返回文件服务器。

如果在查询客户简称的同时，网络上有另外一位用户也要做同样的查询工作，文件型数据库系统也是按同样的方式，将数据完整的传到前端工作站。但可能因网络带宽的不足而导致失败，而需等候前一位用户的结束。

此外，文件型数据库系统也有一个很严重的问题，就是它无法确保数据的完整性。譬如说，我现在要从仓库出货五套计算机系统给客户。我们可能要完成下列操作：

- (1) 打印一张出货单。
- (2) 开出销货发票。
- (3) 仓库的库存量要减少五套计算机。
- (4) 增加客户的应收账款。

实际操作中这 4 项操作一定得完成，才算是一个完整的出货操作。但对文件型数据库系统而言，如果在处理第 3 个步骤时，计算机系统突然死机，那就会造成有该客户的出货记录却没有应收账款的增加，或仓库的库存量也没跟着减少等数据错误的情形发生。

1.2.3 客户机/服务器结构型数据库系统

客户机/服务器结构型数据库系统的出现,解决了主机型数据库系统与文件型数据库系统的缺点。它一样是由一组性能良好且稳定的主机来作数据库服务器,然后连上一群工作站而成,如图 1.4 所示。服务器负责数据库的存取、计算,前端工作站(即客户机)负责数据的输入、验证及存取。

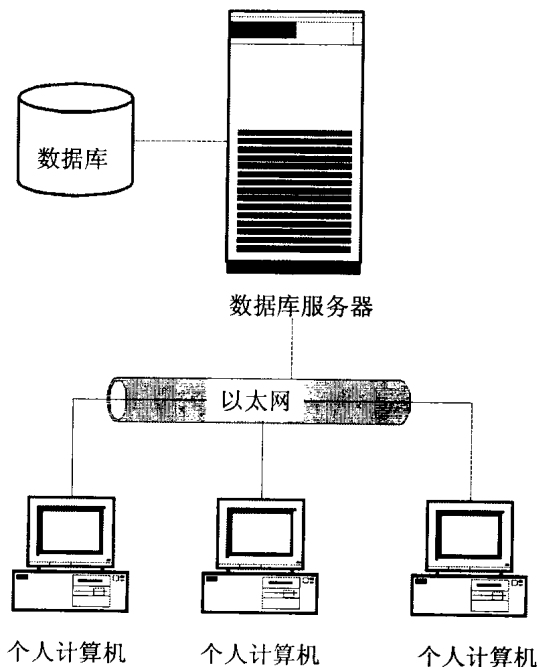


图 1.4 客户机/服务器结构型数据库系统

由于客户机/服务器数据库系统的大量数据计算与数据存取,都是集中在数据库服务器进行,不用通过局域网络传送至前端处理,所以它的效率比文件型数据库高。另外,由于服务器的扩充性很好,当有性能更优异的服务器问世时,我们只要把原来的旧服务器换下,即可享受新的数据库服务器所带来的效率提升。至于前端的工作站,因只负责显示、报表输出及数据输入的工作,而这些工作并不要求微处理器的运算速度,所以对于机器的更换,并不像服务器要求那么高。

我们还以从 10 000 笔客户数据中找出客户编号为 00001 的客户简称为例。客户机/服务器数据库系统对于这种需求的处理方式是:

- (1) 客户机向后端数据库服务器发出请求。
- (2) 数据库服务器在收到客户机的请求后,自客户表找出编号为 00001 的客户简称。
- (3) 数据库服务器再把查询到的结果传至客户机。

由此可看出来,整个查询操作,网络只负担一个查询指令与传送结果,所以与文件型数据库系统比较起来,能够节省带宽并且更有效率。如果此时网络上有另外一位用户,也要做同样的查询,首先,他也一样向后端数据库服务器发出请求。由于前一位用户并没占用多少带宽,所以数据库服务器应该能马上就收到这个请求。当数据库服务器发现后来的

查询要求与之前的查询要求是一样的，它也不用到硬盘去找了，直接到缓存区就把答案找出来，很迅速的把结果返回到前端。

再以先前 1.2.2 节的出货操作为例。当数据库在处理这个出货操作时，如果这四项操作有其中任何一项没有做完，那么数据库就会把这个出货操作取消到处理前的样子，好像什么事也没发生。也就是说，客户机/服务器数据库系统可以确保每项交易都能正确无误的完成。

客户机/服务器数据库系统的缺点在于：客户机数量一旦增加到一定程度时，数据库服务器无法同时处理这么多的联机请求。另外，由于把工作分散处理，管理人员除了做数据库服务器的维护外，客户机的管理也不能少，所以工作量势必增加。另外，当操作环境不是在同一个地方时，客户机/服务器数据库系统也不一定是个很好的解决方法。

1.2.4 多层式结构型数据库系统

多层式结构型数据库系统是最近这几年才兴起。它由一组主机当数据库服务器，连接几台由主机级的计算机所搭建的应用程序服务器，然后前端工作站再连接应用程序服务器来存取数据库服务器上的数据，如图 1.5 所示。这个结构主要是为了简化前端的应用程序，后端数据库则主要负责数据的存取，剩下一些计算的工作就交由应用程序服务器来进行。通常的设计方式是，前端应用程序类似于主机集中型数据库的终端机，只负责信息的显示与数据输入的工作，不直接存取后端的数据库，而是交由应用程序服务器来处理。应用程序服务器除了存取数据库外，还负责绝大部分商业逻辑的计算工作，所以它分担了部分数据库服务器的工作，减轻了数据库服务器的负荷。而且，应用程序服务器可以按需求扩增，不局限在一台。

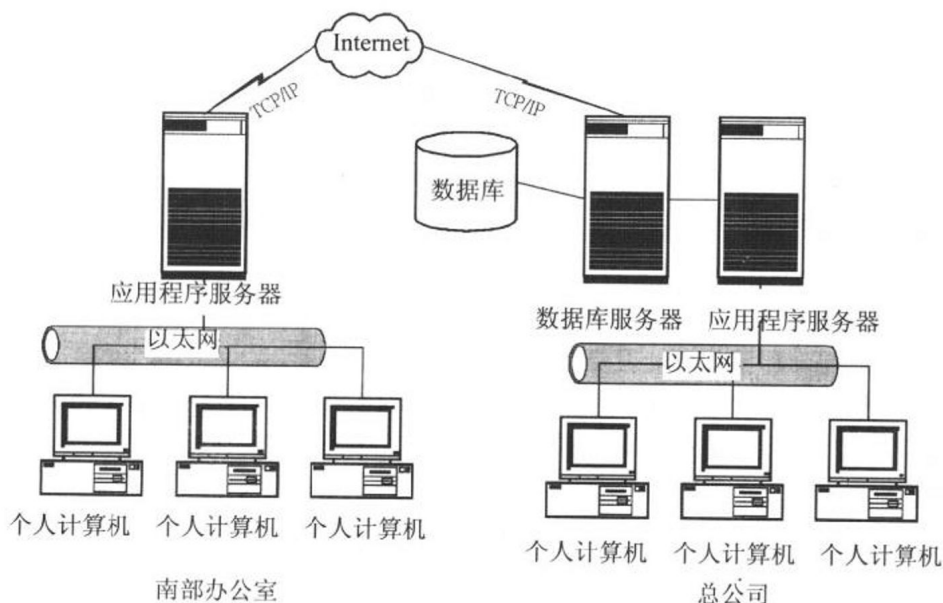


图 1.5 多层式数据库结构系统

事实上, 多层式结构型数据库系统, 是客户机/服务器结构型数据库系统的扩展。如图 1.6 所示, 它主要解决了客户机/服务器结构型数据库系统在多人同时联机所产生的问题, 并且可以适应不同地域的运算环境。但是目前这种技术尚未达到一个纯熟的标准, 各家厂商竞相推陈出新, 一套新的技术方案还没来得及实际操作, 更新的技术又出台了, 致使研发人员一直追随新技术而无法投入到实际应用。

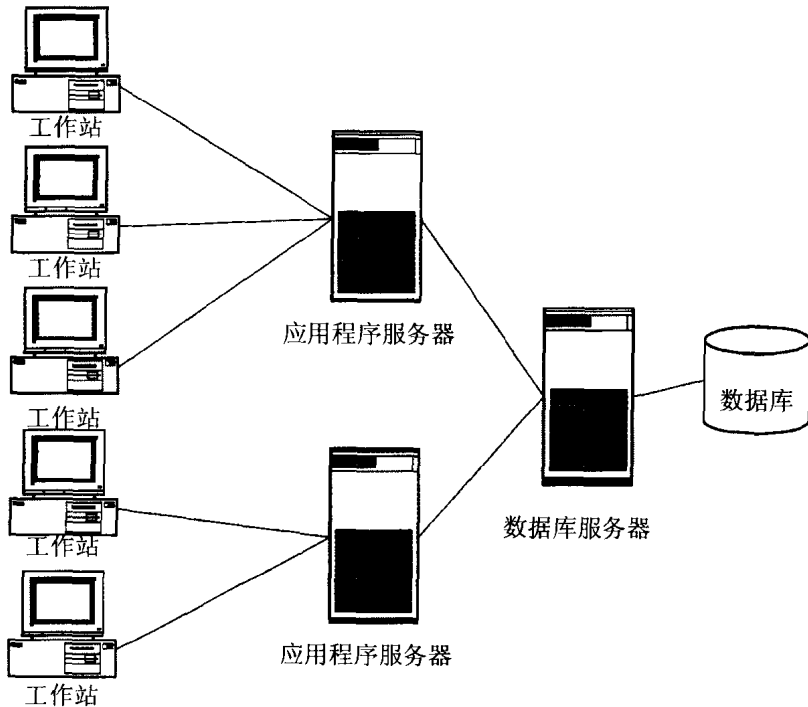


图 1.6 多层式结构数据库系统

如果数据处理的环境是跨区域的, 而且应用系统采用多层式结构型数据库系统的处理方式, 例如, 北部的总公司与南部的分公司, 数据库服务器放置在北部, 南部分公司通过应用程序服务器及慢速的广域网, 连接北部总公司数据库服务器时, 这时要考虑南北两方线路中断的情况。对于这个问题, 我们应就实际的操作层面, 来了解南部分公司能忍受断线的最大程度。如果南部分公司本身的数据量也不少, 而总公司对于南部分公司的数据, 也不要求实时取得, 那我们可能就要考虑如图 1.7 所示的模式。北部总公司与南部分公司都设有数据库服务器, 两边的数据库的集成是通过数据库本身的复制(Replication)功能来达成。如果是像银行取款机这种商业模式, 则还是要考虑使用多层式结构数据库系统。数据专线可能要租用更快速的, 而且还要有备份线路, 以防备主干线中断。

在认识了数据库系统的四大类型结构后, 我们应能针对实际的商业模式的不同, 所在环境的不同, 选择最适宜的数据库系统类型。了解数据库的规范化后, 接下来我们要进一步了解系统分析的方法。因为惟有完备的系统分析, 才能设计出一套有效率、维护方便的数据库来。