



Microsoft SQL Server 2005

中文版设计实务

施威铭研究室 著

- 数据库基本概念
- 关系式数据库与规范化
- SQL Server Management Studio集成工具
- 数据访问、查询、T-SQL程序设计
- 原生XML数据类型、自定义数据类型对象
- 善用索引和全文索引查找
- 视图、规则对象、默认值对象
- 存储过程、自定义函数、触发器、数据游标
- 事务、锁定和隔离级
- 前端应用程序访问数据库



附光盘



机械工业出版社
China Machine Press

TP311.138/501D

2008

Microsoft

SQL Server 2005

中文版设计实务



施威铭研究室 著



机械工业出版社
China Machine Press

本书详细介绍了Microsoft SQL Server 2005在设计方面的应用。全书共22章,包括SQL语言与数据类型、建立数据库、建立表与数据库关系图、数据的新增修改与删除、数据的查询、建立视图、规则对象默认值对象与用户定义数据类型对象、索引的使用、T-SQL程序设计、存储过程、自定义函数、触发器、使用数据游标、事务与锁定、SQL Server与XML以及开发用户端程序访问SQL Server等内容。本书所述内容详细,讲解循序渐进,可帮助读者快速掌握SQL Server 2005技术。

本书中文简体字版由中国台湾地区旗标出版股份有限公司授权机械工业出版社出版,未经本书原版出版者和本书出版者预先书面许可,不得以任何方式复制或抄袭本书的任何部分。

版权所有,侵权必究。

本书法律顾问 北京市展达律师事务所

本书版权登记号: 图字: 01-2007-4193

图书在版编目(CIP)数据

Microsoft SQL Server 2005中文版设计实务/施威铭研究室著. —北京:机械工业出版社, 2008.1

ISBN 978-7-111-22639-0

I. M… II. 施… III. 关系数据库—数据库管理系统, Microsoft SQL Server 2005
IV. TP311.138

中国版本图书馆CIP数据核字(2007)第165694号

机械工业出版社(北京市西城区百万庄大街22号 邮政编码 100037)

责任编辑:李南丰

三河市明辉印装有限公司印刷 · 新华书店北京发行所发行

2008年1月第1版第1次印刷

186mm×240mm · 31印张

定价:65.00元(附光盘)

凡购本书,如有倒页、脱页、缺页,由本社发行部调换

本社购书热线(010) 68326294

序

在当今这个信息爆炸的时代，如何找出一个有系统、有效率的方法，来整合并管理我们每天接收的信息，这是现代人面临的一个重要而急迫的课题。放眼望去，数据库似乎是目前最好的解决方案了。

每个人或多或少都拥有一些属于自己的数据库，而且不可避免地也成为别人数据库的一部分。数据库其实早已深入到我们的生活之中，小到您手机中的通讯录，大到医院里所有人的病历数据，我们已在无意中享受着数据库带给我们的便利。

数据库的应用是如此的广泛，您是否也希望能成为数据库的管理员呢？如果是，那就从SQL Server 2005开始着手吧！

SQL Server 2005无疑是数据库领域的重量级产品。不论是大型企业还是个人需求，SQL Server 2005都可以提供适合的解决方案。为了让初次接触SQL Server 2005的用户得到最有效的学习，我们规划了由浅入深的内容和完整的学习过程，从数据库的基础知识、数据库的建构，以及高级的数据库设计技巧、T-SQL程序编写等。详细的解说搭配扎实的操作练习，帮您奠定良好的基础。本书绝对是帮助您成为数据库高手的一块最坚实的阶石。

当然，要成为数据库高手不是一朝一夕的事。打好基础之后，还有许多SQL Server的维护、备份、管理等工作，等着我们继续挑战。有兴趣的读者不妨参考本书的姐妹篇《Microsoft SQL Server 2005中文版管理实务》一书。

施威铭研究室

2006年3月

光盘说明

使用范例数据库

本书总共使用到3个范例数据库——练习00、练习01、练习02，分别放置于同名文件夹中。可以从光盘菜单打开范例数据库所在的文件夹（或直接打开“您的光驱代号：\范例数据库\”）。只要按照各章的说明，将所需的数据库文件——练习xx.mdf 和练习xx_log.ldf复制到硬盘中，然后在范例文件上点击鼠标右键执行“属性”命令，确定已取消只读属性，之后再参照6.3节附加数据库的方法，将范例数据库附加到您的SQL Server上，便可以按照书中说明逐步操作学习了。

此外，考虑您可能需要单独阅读某一章节，为避免您因为没有依序阅读、操作，而造成在某一章中的执行结果和书中所列不同，我们另外存储了第8章到第22章的范例数据库，分别置于范例数据库\CHxx文件夹内的CHxx-Start目录中，各章文件夹中所存放的数据库，都是已完成前面各章操作后的结果。另外，为了方便查阅部分章节操作后的结果，我们也将执行完该章各项操作后的数据库另存于范例数据库\CHxx文件夹内的CHxx-Finish目录中，您同样可以自行查看。

第21、22章分别介绍XML及应用程序的开发，因此其目录下还有数据库以外的范例文件，使用方法请直接参照该章的说明。

使用SQL语法范例文件

您可以从光盘菜单打开SQL语法范例文件所在的文件夹（或直接打开“您的光驱代号：\SQL语法范例文件\”），其中我们替第6章到第21章以及附录B、D各准备一个SQL文件（以CHxx.sql命名，xx为章的编号，例如第6章的SQL文件为CH06.sql），内含该章所有的SQL范例程序。

要使用某章的SQL文件，如第6章的CH06.sql，只要直接打开该文件即可。关于如何在SQL Server Management Studio中使用SQL文件的方法，可以参考3.5节的说明。

目 录

序
光盘说明

第一篇 基础概念篇

第1章 认识数据库系统	1
1.1 数据库系统简介	1
1.2 数据库的类型	2
1.3 关系数据库的内部结构	5
1.4 数据库系统的网络架构	5
1.5 数据库管理系统的基本功能	6
1.6 结构化查询语言SQL	7
1.7 数据库系统的用户	7
第2章 规划关系式数据库	9
2.1 简易的规划流程	9
2.2 如何设计一个完善的数据库	9
2.3 收集数据项并转换成列	11
2.4 认识关系、主键与外键	12
2.5 数据的完整性	15
2.6 表的关系种类	15
2.7 数据库的规范化分析	17
2.8 数据库规划实战	22

第二篇 准备篇

第3章 熟悉SQL Server的工作平台	27
3.1 SQL Server的管理架构类型	27
3.2 浏览SQL Server的各项工具程序	27
3.3 使用SQL Server Configuration Manager 管理服务器端相关服务	28
3.4 SQL Server Management Studio环境介绍	30
3.5 在SQL Server Management Studio执行 SQL语句	32

3.6 SQL Server联机丛书	40
3.7 登录账户与使用权限的设置	44
3.8 从用户端连接远程或多部SQL Server	55
3.9 检查连接的通信协议	60
第4章 认识SQL语言与数据类型	63
4.1 SQL语言的历史与语法标准	63
4.2 SQL语言与传统程序语言的差别	63
4.3 关键字、子句与语句	64
4.4 SQL语言的功能分类	64
4.5 数据类型	65
4.6 列的NULL值与DEFAULT值	71
4.7 标识符	72
第5章 查看SQL Server的数据库对象	75
5.1 SQL Server的内建数据库	75
5.2 解读数据库的相关信息	76
5.3 查看数据库的各类对象	80

第三篇 入门篇

第6章 建立数据库	91
6.1 使用SQL Server Management Studio管理 工具建立数据库	91
6.2 用CREATE DATABASE语句建立数据库	94
6.3 分离与附加数据库	100
6.4 使用SQL Server Management Studio管理 工具修改数据库设置	105
6.5 用ALTER DATABASE语句修改数据库	109
6.6 删除数据库	113
第7章 建立表与数据库关系图	115
7.1 使用SQL Server Management Studio管理 工具建立新表	115
7.2 使用SQL Server Management Studio管理 工具修改表	119

7.3	使用SQL Server Management Studio管理工具建立表间的关系	123
7.4	设置条件约束维护数据完整性	126
7.5	使用SQL Server Management Studio管理工具删除表	130
7.6	数据库关系图与关系图对象	131
7.7	用CREATE TABLE语句建立表	139
7.8	用ALTER TABLE语句修改表	143
7.9	用DROP TABLE语句删除表	147
7.10	临时表	147
第8章	数据的新增、修改与删除	149
8.1	使用SQL Server Management Studio管理工具编辑数据	149
8.2	新增记录——INSERT语句	153
8.3	简易查询——SELECT语句初体验	157
8.4	用查询结果建立新表——SELECT INTO语句	159
8.5	更新记录——UPDATE语句	160
8.6	删除记录——DELETE与TRUNCATE TABLE语句	162
第9章	查询数据	164
9.1	SELECT语句的基本结构	164
9.2	SELECT子句	164
9.3	FROM子句	169
9.4	WHERE子句	173
9.5	GROUP BY子句	173
9.6	HAVING子句	176
9.7	ORDER BY子句	177
9.8	COMPUTE子句	178

第四篇 实务篇

第10章	更多的查询技巧	179
10.1	用UNION合并多个查询结果	179
10.2	子查询	182
10.3	使用SQL Server Management Studio 管理工具设计SQL查询	186
10.4	T-SQL的常数	195

10.5	隐式类型转换	199
10.6	T-SQL的运算符	200
10.7	运算符的优先顺序	204
10.8	处理列中的NULL值	204
10.9	使用APPLY运算符在查询时同时调用 函数进行运算	206
10.10	排序函数: ROW_NUMBER()、 RANK()与DENSE_RANK()	207
10.11	使用PIVOT与UNPIVOT将直式表 转为横式	209
第11章	建立视图	212
11.1	视图的用途	212
11.2	使用SQL Server Management Studio 管理工具建立视图	215
11.3	用CREATE VIEW语句建立视图	217
11.4	用ALTER VIEW语句修改视图	221
11.5	运用UNION设计视图	221
11.6	编辑视图中的记录	222
11.7	删除视图	223
第12章	规则对象、默认值对象与用户 定义数据类型对象	225
12.1	规则对象 (Rule)	225
12.2	默认值对象 (Default)	229
12.3	用户定义数据类型 (UDT) 对象	232
第13章	善用索引提高查询效率	236
13.1	索引简介	236
13.2	聚集索引与非聚集索引	237
13.3	惟一索引与复合索引	237
13.4	由系统自动建立的索引	238
13.5	建立索引的注意事项	241
13.6	使用SQL Server Management Studio 管理工具建立索引	242
13.7	使用SQL Server Management Studio 管理工具管理索引	246
13.8	用SQL语法处理索引	248
13.9	查看查询的执行计划	252
13.10	设置计算列的索引	253
13.11	设置视图的索引	257

第14章	全文索引与查找	264
14.1	Microsoft SQL Server全文检索引擎服务	264
14.2	全文目录	264
14.3	建立全文索引	266
14.4	填充全文目录与索引	271
14.5	使用全文索引查找数据	274
14.6	图像类型列的全文检索查找	278

第五篇 高级篇

第15章	T-SQL程序设计	281
15.1	批处理执行	281
15.2	使用注释	282
15.3	局域变量与全局变量	283
15.4	条件判断与流程控制	284
15.5	特殊的程序控制	291
15.6	错误处理	293
15.7	使用CTE进行返回查询	298
15.8	SQL脚本	301
15.9	自动产生SQL脚本	304
15.10	使用不同数据库或不同服务器中的对象	307
第16章	存储过程	313
16.1	存储过程简介	313
16.2	存储过程的建立、使用与修改	316
16.3	设计存储过程的技巧	324
第17章	自定义函数	333
17.1	自定义函数的特点	333
17.2	自定义函数的建立、使用与修改	334
17.3	自定义函数的使用技巧	342
第18章	触发器	349
18.1	触发器的用途	349
18.2	触发器的种类与触发时机	350
18.3	触发器的建立与修改	350
18.4	设计触发器的技巧	354
18.5	建立AFTER触发器	359
18.6	建立INSTEAD OF触发器	365

第19章	使用数据游标	373
19.1	游标简介	373
19.2	游标的声明、打开、关闭与删除	375
19.3	使用FETCH读取游标中的记录	378
19.4	通过游标修改或删除数据	381
19.5	使用游标变量	382
19.6	使用存储过程的游标参数	383
19.7	游标的使用技巧	384
第20章	事务与锁定	388
20.1	事务简介	388
20.2	进行事务的3种模式	391
20.3	嵌套事务与@@TRANCOUNT	392
20.4	事务存储点的设置与恢复	395
20.5	分布式事务	396
20.6	事务的隔离等级	398
20.7	数据锁定	400
20.8	锁定的死锁问题	403

第六篇 网络篇

第21章	SQL Server与XML	405
21.1	XML简介	405
21.2	SELECT语句的FOR XML子句	409
21.3	原生XML的数据类型	411
21.4	通过HTTP传输XML数据	417
21.5	OPENXML语句	420
第22章	开发用户端程序访问SQL Server	425
22.1	从用户端访问SQL Server的途径	425
22.2	通过Access 2003访问SQL Server	431
22.3	编写VB程序访问SQL Server	439
22.4	编写ASP程序访问SQL Server	457
附录		461
附录A	安装SQL Server 2005	461
附录B	数据类型补充说明	474
附录C	聚簇索引与非聚簇索引的结构	480
附录D	SQL Server的常见错误消息	483
附录E	增加SQL Server性能的方法	486

第一篇 基础概念篇

第1章 认识数据库系统

想要学习设计数据库系统，需要一些数据库概念与理论来支持。假如您早已具备这些基础知识，可以略过本书的第1章和第2章，直接从第二篇开始学习。但如果您是数据库设计的新手，或以前所学的理论已经淡忘得差不多了，没关系，先通过本章来复习一下吧！

1.1 数据库系统简介

数据库系统 (Database System) 是计算机化的数据存储系统，用户可通过各种应用程序访问其中的数据。数据库系统可分为两个部分：**数据库** (Database) 和**数据库管理系统** (Database Management System, DBMS)，如图1-1所示。

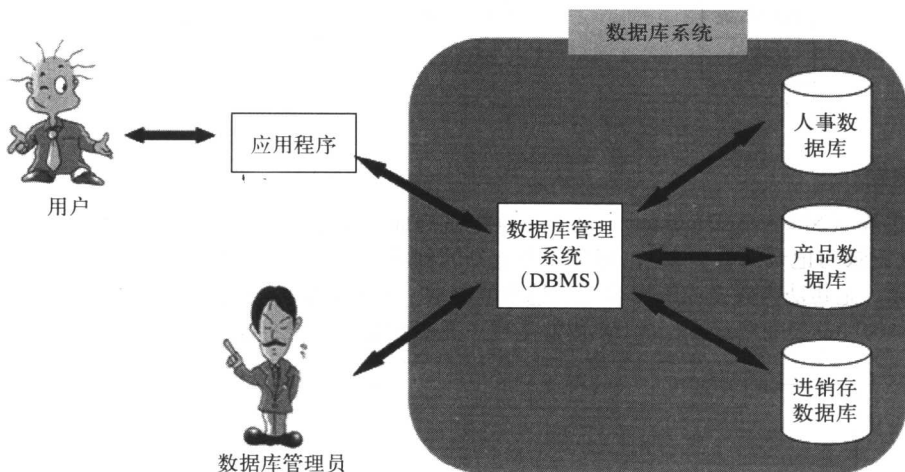


图 1-1

数据库是存储数据的地方。一个**数据库系统**中可以有多个**数据库**，每个**数据库**都是一组整理好的数据集合。一般来说，我们会将**数据库**想象成一个存放数据的容器，但**数据库**的真实形态其实是一个个的电子文件。

数据库管理系统则是指管理数据库的软件，它负责用户与数据库之间的沟通，如访问数据库中的数据，以及管理数据库的各项事务等。Microsoft的Access，还有许多用在大型数据库系统上的Microsoft SQL Server、Oracle、SyBase、Informix、MySQL、PostgreSQL等，都是**数据库管理系统**。稍后，我们会再对**数据库管理系统**所具备的基本功能做更为详细的介绍。

1.2 数据库的类型

就数据库中数据的存储架构来看,数据库可分为多种类型,较常见的有层次式、网状式、关系式以及面向对象式4种。下面,我们就针对这4种数据库类型做简单的介绍。

► 层次数据库

层次数据库 (Hierarchical Database) 采用树状结构,将数据分门别类存储在不同的层次之下。这种类型的优点是数据结构很类似金字塔,不同层次间的数据关系直接且简单;缺点是因数据纵向发展,横向关系难以建立,所以数据可能会重复出现,造成管理维护上的不便,如图1-2所示。IBM的IMS就是属于此类的数据库管理系统。

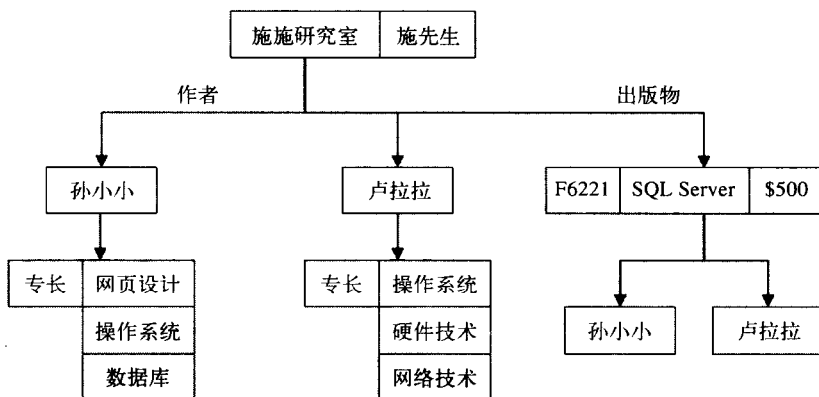


图 1-2

► 网状数据库

网状数据库 (Network Database) 是将每条记录当成一个节点,节点与节点之间可以建立关系(也就是建立记录与记录间的关系),形成一个复杂的网状架构。优点是避免了数据的重复,缺点是关系比较复杂,尤其是当数据库的内容越来越多的时候,关系的维护会变得非常麻烦,如图1-3所示。Computer Associates公司曾经推出的IDMS就是属于此类的数据库管理系统。

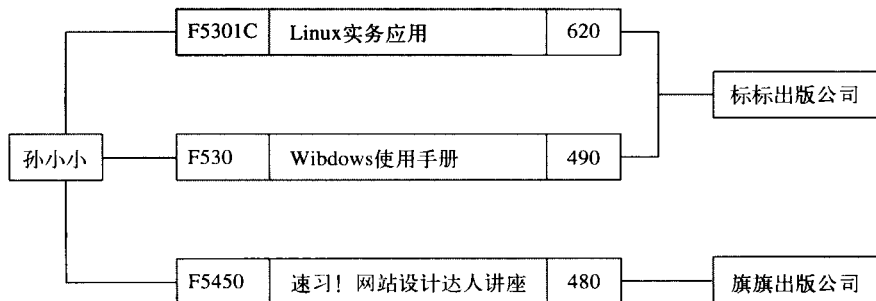


图 1-3

图1-3表示从作者姓名可以查到他写过的书,以及这些书是由哪个出版社出版的。当记录的数量较多的时候,关系就容易变得牵扯不清。

► 关系数据库

关系数据库 (Relational Database) 是以二维的矩阵来存储数据 (可以说是将数据存储表格的行、列之中), 而存储在行、列里的数据必然会有“关系”, 所以这种存储数据的方式才会叫做关系数据库, 而存储数据的表格则称为“表”。举例来说, 通讯录表的每一行可以划分为“姓名”、“地址”、“电话”等, 如图1-4所示。

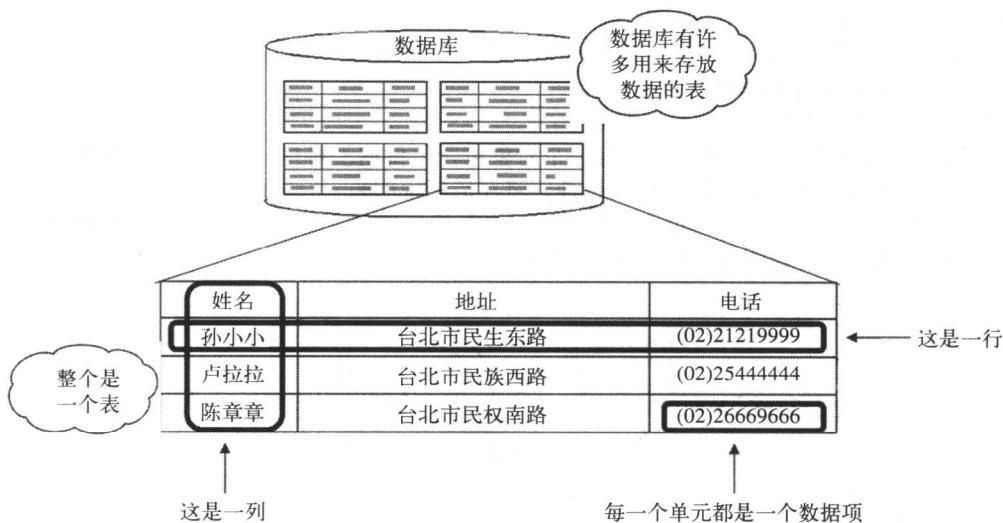


图 1-4

假如我们要从图1-4所示的表寻找“卢拉拉”的地址, 则是由横向的“卢拉拉”与纵向的“地址”, 求相交关系而得来, 如图1-5所示。

姓名	地址	电话
孙小小	台北市民生东路	(02)21219999
卢拉拉	台北市民族西路	(02)25444444
陈章章	台北市民权南路	(02)26669666

图 1-5

除了存储在表中的行与列之间会有关系, 关系式数据库里面的表之间通常也会有关系。这种方式的优点是可以从一个表中的列, 通过表的关系找到另一个表中的数据, 如图1-6所示。

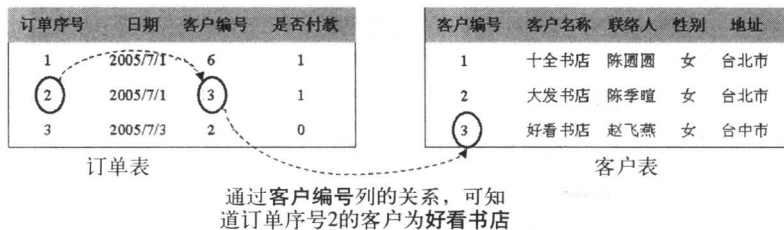


图 1-6

目前市场上以关系数据库最为常见,像Microsoft SQL Server、SyBase、Informix、MySQL、PostgreSQL、Access等,都属于关系数据库管理系统(Relational DBMS, RDBMS)。

► 面向对象数据库

面向对象数据库(Object-Oriented Database)以面向对象的方式来设计数据库,其中包含了对象的属性、方法、类别、继承等特性。属于这类的数据库管理系统有Computer Associates公司的Jasmine、Eastman Kodak公司的Alltalk、Servio公司的GemStone、O2 Technology的O2等。

此外也有关系数据库为主,再在其上架设面向对象概念的数据库,如PostgreSQL。

图1-7所示是一个**面向对象数据库**的结构示意图。

图1-7所示的示意图中有几个重点,说明如下:

- 每一行就是一个**对象**:以订单为例,每一个对象包含了**日期、客户、订购项目、金额**等属性(OID是产生对象时的ID,不是对象的属性,将在后面说明)。这些属性可以是文字数据、数值数据,甚至是另一个对象,而且一个属性不必是惟一的值,如图1-7所示的订单数据库中OID008的对象,其**订购项目**属性就包含OID43及OID46两个对象。

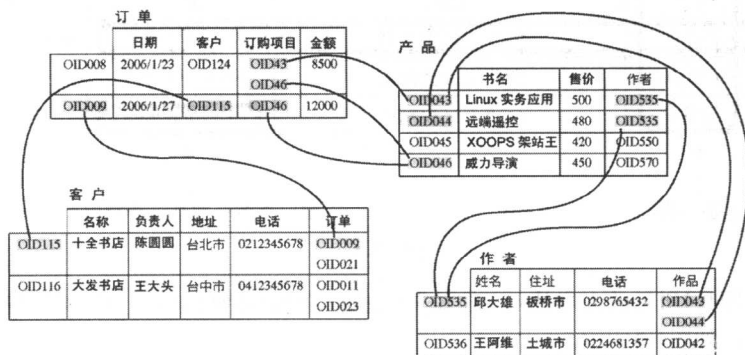


图 1-7

- 每个对象拥有**惟一**的OID (Object Identity):

同样以订单为例,每个对象的第一栏就是对象的OID。OID并不是数据库设计者赋予的,而是在该对象成立时,自动产生的;要特别注意,OID并不是对象的属性,实际上我们是看不到OID的。当对象内包含其他对象时,就能通过这个惟一的OID快速找到对应的对象。

若要比关系数据库和面向对象数据库,关系数据库必须由数据库设计者来设计、建立和管理关系。但面向对象数据库中,对象和对象之间的关系是因其**属性**而必然发生的。

我们先看图1-8所示的关系数据库的表。

订单序号	日期	客户编号	是否付款
1	2005/7/1	6	1
2	2005/7/1	3	1
3	2005/7/3	2	0

订单表

客户编号	客户名称	联络人	性别	地址
1	十全书店	陈圆圆	女	台北市
2	大发书店	陈季喧	女	台北市
3	好看书店	赵飞燕	女	台中市

客户表

通过客户编号列的关系,可知道
订单序号2的客户为好看书店

图 1-8

由图1-8可知，两个表是通过**客户编号**来达成关系的，而这个关系性在关系数据库中，必须由设计者自行建立才会真正产生关系。接着，看图1-9所示的面向对象式数据库。

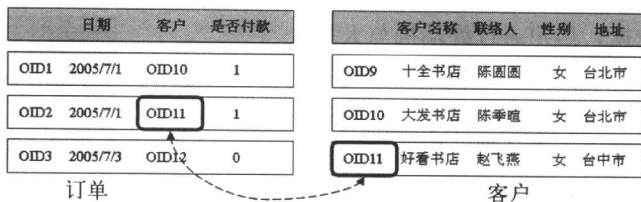


图 1-9

图1-9中，两个对象是通过OID联系起来的。简单地说，在关系数据库中，表间的**关系**必须靠设计者自行建立产生，而面向对象数据库中，各对象之间的关系则是在对象建立时自行联系起来的。

本书的主角Microsoft SQL Server属于关系数据库管理系统，所以稍后我们会再对关系数据库的内部结构做介绍。若读者对另外3种数据库类型有兴趣，请自行参考相关书籍。

1.3 关系数据库的内部结构

关系数据库是由表（Table）组成的，其最大的特色是将数据分类存储在表中。

如图1-10所示的**客户表**专门用来存放客户方面的数据。其中第一行的项目，如**客户编号**、**客户名称**、**联络人**等，是客户数据中所具备的各项**属性**（attribute）。数据库的用语为**列**（Field或Column）；从第二行起则存放各列实际的值。例如**十全书店**便是客户编号1的**客户名称**。将同一行各列的实际值集合起来，就称为一条**记录**（Record或Row）：

客户表			这是一条记录	
客户编号	客户名称	联络人	地 址	电 话
1	十全书店	陈圆圆	台北市仁爱路二段56号	02-23219845
2	大发书店	陈季瞳	台北市敦化南路一段1号	02-23334444
3	好香书店	赵飞燕	台北市忠孝东路四段4号	02-25984333
4	英雄书店	孟庭亨	台北市南京东路三段3号	02-27225652
5	娱人书店	刘金城	台北市北平东路24号	02-25786666
6	新新书店	黎国明	台北市中山北路六段88号	02-25557444

图 1-10

表的内部结构其实就是**列和记录**。在设计关系数据库时，最重要的工作是妥善规划数据的配置，以避免产生数据重复存储、数据不一致或表间的关系不完整等问题。这一部分我们将在下一章讨论。

1.4 数据库系统的网络架构

“网络架构”涉及的是数据库系统要如何部署的问题。通常，我们会根据组织的规模、数据量的多少、使用的人数、软/硬件设备等条件来考虑，常见的有下列4种网络架构：

► 单机架构

单机架构是由同一部计算机负责所有数据库系统的工作，包括保存数据、处理数据、管理及使用数据库系统等，适合用户少、数据量也不多的数据库系统使用，如小公司或个人用户所建立的数据库系统。通常用Access、FoxPro所设计的数据库系统多采用这种架构。

► 大型主机/终端机架构

大型主机/终端机架构是由一部大型主机负责存储及处理庞大的数据，用户通过终端机与大型主机连接，以访问数据库的内容，如图1-11所示。这种架构的缺点在于，当多人同时使用时，由于所有的工作都要由大型主机来处理，因此主机会非常忙碌，易造成响应缓慢的问题。目前除了一些大型机构外，已比较少使用这一类的架构了，而且此类的大型主机价格都相当昂贵，一般中小企业可能负担不起。

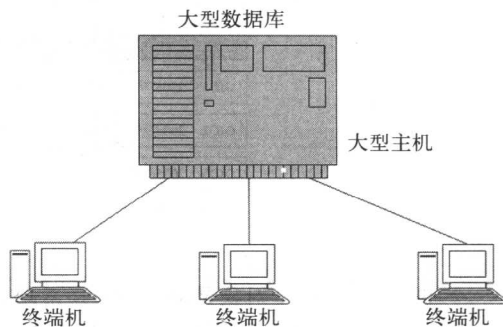


图 1-11

► 主从式架构

由于个人计算机的价格低廉，运算速度也不错，利用网络互相连接之后，作为用户端（Client）的各台计算机只要连接到作为数据库服务器端（Server）的计算机，就可以访问数据库，而且部分工作可由用户端计算机来处理，分散数据库服务器的负荷。这就是**主从式架构**的部署方式，同时也是目前一般公司中最普遍采用的方式，如图1-12所示。若采用**主从式架构**，通常还会另外编写用户端程序，以提供用户易学易用的操作界面。

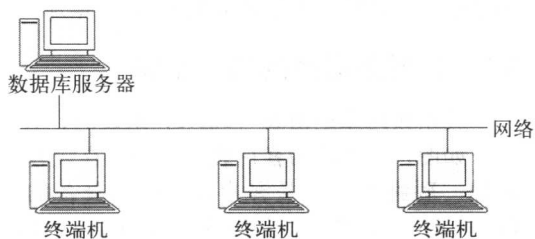


图 1-12

► 分布式架构

分布式架构是由数台数据库服务器所组成，用户在访问数据时，数据可以来自于不同的服务器，这样在访问的效率上会比较好。**分布式架构**的数据访问方式和**主从式架构**类似，只不过是多了几台数据库服务器而已，如图1-13所示。

看过上述的网络架构介绍后，您可能会问，MS SQL Server适合用于哪一种网络架构？原则上说，MS SQL Server适用于**主从式架构**的环境，但实际除了**大型主机/终端机架构**之外，其他3种架构都可以。我们可以依组织的规模及需要来选择，例如在5到10人的小公司中，用**单机架构**就绰绰有余了；而在规模庞大的大型企业中，则可建立多台SQL Server（数据库服务器），以构建出大容量、高效率的分布式工作平台。

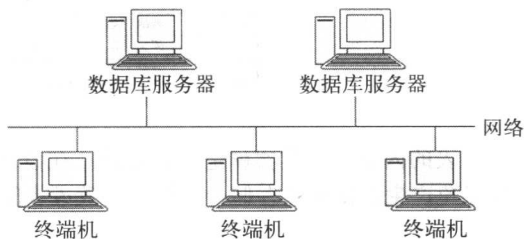


图 1-13

1.5 数据库管理系统的基本功能

在第1.1节中，我们已经提到过，**数据库管理系统**其实就是管理数据库的软件系统，它们要负责整个数据库的建立、数据访问、权限设置、数据备份、操作的监督与记录等工作。下面，我们再进行

一步详细介绍数据库管理系统 (Database Management System, DBMS) 所应具备的基本功能。

- **数据定义**：DBMS必须能够充分定义并管理各种类型的数据项目，例如关系式数据库管理系统必须具备建立数据库、表、定义各列的数据类型，以及表之间的关系等能力才行。
- **数据处理**：DBMS必须提供用户对数据库的访问能力，包括新增、修改、查询和删除等基本功能。有时DBMS提供的功能虽然完善，但是并不是很适合一般的用户操作，这就需要程序员另外再编写用户端的应用程序，以供一般用户操作。
- **数据安全**：DBMS应该具备设置用户账户、密码及权限的功能，让每一个用户只能访问授权范围内的数据，以防止机密数据外泄或数据库遭受任何有意或无意的破坏。
- **数据备份**：DBMS必须提供方便的数据备份功能，这样在数据库不幸意外毁损时，可以还原到备份数据时的状态，以减少损失。

此外，维护数据库的效率也非常重要，尤其是在数据量很大或用户很多的时候，数据库若因效率不佳而导致访问速度变慢，也会严重影响到操作人员的工作效率。

也许有人会认为，上述的说明可能有以偏盖全之嫌。因为数据库管理系统可分为多种类型，而且各厂商的数据库管理系统的详细功能也不尽相同。但是每种数据库管理系统所应具备的基本功能其实是差不多的，而且只要各位读者掌握这些基本功能，学习数据库管理系统的过程就很容易了。

1.6 结构化查询语言SQL

SQL (Structured Query Language，一般习惯念成“sequel”，但正确的念法应该是“S-Q-L”，即结构化查询语言。它是目前关系式数据库管理系统所使用的查询语言，也就是说，大部分的关系式数据库管理系统都支持SQL，所以用户可以利用SQL语言直接对关系式数据库进行访问与管理。

SQL语言的基本语法是由一些简单的英文句子构成，相当简单易学。下面，我们就来看个例子。假设要在订单数据库中建立一个客户表，可以执行以下的SQL语句：

```
USE 订单数据库          ←使用此数据库
CREATE TABLE 客户表    ←建立名称为客户表的表
(
    客户编号 int,
    联络人  char(10),
    送货地点 varchar(50)
)
```

指定表中的列名称及数据类别

在SQL Server中执行上面的SQL语句，就可以建立如下形式的表：

客户编号	联络人	送货地点

有关SQL语法及使用技巧，请各位读者先不要着急，以后我们会陆续详细介绍。

1.7 数据库系统的用户

最后，让我们来看看，从数据库系统的设计、建立、操作到管理阶段，需要哪些用户的参与，各位读者可试着从这些用户类型中找出自己的位置。

- **数据库设计者**：数据库设计者 (Database Designer) 负责整个数据库系统的设计，根据用户的需求设计适当的格式来存放数据；同时对于整个数据库的用户访问权限也需要做规划。设计完成后，就可交由数据库管理员进行管理维护的工作了。在一般中小型企业中，数据库的

设计者与管理员可能是同一个人；若是大型企业，则可能设计者是一组人，而管理员又是另外一组人。

- **数据库管理员：**数据库建好之后，便可以交给**数据库管理员**（Database Administrator, DBA）来进行管理及维护。DBA最主要的任务，就是维护数据库的有效运行，并监督、记录数据库的操作状况，必要时还得修改数据库的数据结构或各项设置，以符合实际需求或提高工作效率。

由于数据库中的数据对企业非常重要，而数据库系统难免会碰到人为疏失、硬件或操作系统的问题而损坏，所以DBA必须设置数据库备份的方法和时机，并且在数据库受损时尽快让数据库恢复原状。

除此之外，DBA也要负责数据库的账户管理，决定哪些人有权登录数据库，哪些人有权执行哪些动作。例如，最基本的用户可能只有查询功能，需要输入数据的用户则具有写入数据的功能，数据备份人员必须具有备份数据的权限等。

TIP 本书的内容将着重在数据库的设计部分，至于和数据库管理相关的各项工作，可参考本书姊妹篇：机械工业出版社出版的《Microsoft SQL Server 2005中文版管理实务》。

- **应用程序设计者：**应用程序设计者（Application Designer）负责编写访问数据库的用户端应用程序，让用户用方便的操作界面来使用数据库。可用来开发应用程序的工具很多，早期的程序员可能用C或PASCAL等语言，现今的程序员多采用Visual Basic、JAVA、Delphi、C++、.NET或PowerBuilder等开发工具。

TIP 有关如何开发用户端应用程序的详细情形，在第22章有比较深入的探讨。

- **最终用户：**最终用户（End user）就是真正经常访问数据库的用户，他们只需要学会用户端的应用程序，不需要担心数据库的维护或管理方面的任何问题。如果遇到问题，他们只要请DBA处理即可。

第2章 规划关系式数据库

规划数据库是相当重要的一件事，好的规划可以节省数据库的存储空间、减少数据输入错误的机会、加快数据库的运作效率等。本章介绍如何做好关系式数据库的规划工作。

2.1 简易的规划流程

数据库的规划，说起来是一门大学问，市面上常可看到一本本厚达上千页的英文书，内容只是介绍数据库的原理及规划而已。当然，本书没有那么大的篇幅为各位读者讲述数据库理论，因此经过一番的去芜存菁，以下我们从比较轻松、简单、实用并且不偏离数据库理论的角度，向各位读者介绍规划数据库的工作。

首先，我们将关系式数据库的规划工作概略分为两个阶段：

- **第一阶段：**收集完整且必要的数据项，并转换成表的列形式。
- **第二阶段：**将收集的列做适当分类后，归入不同的表中，并建立表间的关系。

从上面两个阶段的语句，各位读者应该不难看出，关系式数据库的规划工作，主要就是找出数据库所需的表，以及各表之间的关系。

2.2 如何设计一个完善的数据库

数据库设计包含两大部分：一是操作界面设计；二是结构设计。

- **操作界面设计：**就SQL Server而言，操作界面就是表单的设计，或以程序语言（例如Visual Basic）所编写的操作界面。让用户不必接触数据库的结构，就能操作数据库，如新建、删除数据等工作。

- **结构设计：**结构设计是指设计出适当且最优化的表。一个结构良好的数据库可提高其整体的访问效率及存储效率。

► 数据库的设计流程

数据库发展初期，数据库规划的完善与否，通常根据设计者的经验、方法及知识水准不同而有所差别，且最后的成果未必能符合用户的需求。

近年来，随着计算机普及，加上用户对数据库的需求越来越高，数据库的应用范围越来越广、越来越复杂。为了避免设计者闭门造车，直到规划后期才发现错误，规划的过程应分为几个阶段分别执行，并随时与用户沟通，这样才能设计出既完善又符合需求的数据库。

数据库的规划过程大致可分为4个阶段，如图2-1所示。



图 2-1

以下我们就分别说明各个阶段的工作。