

Oracle 11g

SQL和PL/SQL编程指南

李伟 安永丽 胡雄 编著



清华大学出版社

014057260

TP311.1380R
361

Oracle 11g SQL 和 PL/SQL 编程指南

李 伟 安永丽 胡 雄 编 著



TP311.1380R
361

清华大学出版社
北 京



北航

C1742147

内 容 简 介

SQL(Structured Query Language)是关系数据库的基本操作语言。它主要包括数据查询、数据操纵、数据定义等功能,是应用程序与数据库进行交互操作的接口。PL/SQL(Procedural Language/SQL)是 Oracle 特有的编程语言,它可以像其他高级编程语言一样,编写出各种完成数据库操作功能的程序。由于 PL/SQL 由 Oracle 系统本身编译执行,所以程序运行效率更高。

本书为 Oracle 数据库应用开发人员提供 SQL 使用指南和 PL/SQL 编程技术。通过学习本书,读者不仅可以掌握 SQL 和 PL/SQL 的基础知识,而且还可以掌握 Oracle 11g SQL 和 PL/SQL 的许多高级特征。

本书可作为高等院校计算机相关专业的辅助教材,也可作为各类高级数据库编程人员的参考书。本书的编写既考虑了 SQL 和 PL/SQL 的初学者,同时也兼顾到了有经验的 PL/SQL 编程人员。本书还可以作为 Oracle 11g 培训班的教材。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

Oracle 11g SQL 和 PL/SQL 编程指南/李伟,安永丽,胡雄编著. --北京:清华大学出版社,2014
ISBN 978-7-302-36892-2

I. ①O… II. ①李… ②安… ③胡… III. ①关系数据库系统—指南 IV. ①TP311.138-62

中国版本图书馆 CIP 数据核字(2014)第 131463 号

责任编辑:汤涌涛 桑任松

封面设计:杨玉兰

责任校对:周剑云

责任印制:何 芊

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62791865

印 刷 者:北京鑫丰华彩印有限公司

装 订 者:三河市漂源装订厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:21.75 字 数:528 千字

版 次:2014 年 8 月第 1 版 印 次:2014 年 8 月第 1 次印刷

印 数:1~3000

定 价:39.50 元

产品编号:056298-01

前 言

Oracle 公司是世界排名前列的国际大型企业。Oracle 数据库是全球领先、性能优异的大型数据库管理系统。Oracle 数据库广泛地应用在金融、通信、航空等领域，以其并发数据量大、极高的可靠性、安全性和可扩展性赢得了广大高端用户的青睐。

早期 Oracle 数据库主要应用于 Unix 操作系统，影响了它的广泛使用。随着 Oracle 公司提供了基于 Windows 平台的 Oracle 版本以后，Oracle 数据库拥有了更为广泛的应用市场。近些年来，随着我国中小企业对数据库可靠性、安全性要求的提高，基于 Windows 平台的 Oracle 数据库服务器获得了广泛的青睐。随之而来，对 Oracle 数据库管理和开发的人员的数量需求不断增加，素质要求不断提高。本书的编写既考虑了 Oracle 数据库管理和开发的初学者，同时也兼顾到了有一定基础的管理和开发人员。凡是想学习 SQL 语句或利用 PL/SQL 提高 Oracle 数据库管理和开发能力的人士，都可以从本书获得借鉴。

本书共 12 章。其中第 1~3 章对 Oracle 11g 数据库进行了概述，第 4~9 章用大量范例详解了 SQL 语句，第 10~12 章精选了可供借鉴的 PL/SQL 编程实例。

第 1~3 章的内容包括数据库及 Oracle 11g 的产生与发展、数据库分类、SQL 和 PL/SQL 简介、SQL 和 PL/SQL 开发环境简介，这一部分为全书做了必要的铺垫，建立了本书中使用的数据库，详细地介绍了 SQL 语句和 PL/SQL 编程调试开发环境。

第 4~9 章的内容包括数据查询语句(SELECT)、数据操纵语句(DML)(INSERT/UPDATE/DELETE)、数据定义语句(DDL)(CREATE/ALTER/DROP/RENAME/TRUNCATE)、事务控制语句(COMMIT/ROLLBACK/SAVEPOINT)、数据控制语句(DCL)(GRANT/REVOKE)。对上述各种类型的 SQL 语句均结合例子详细讲解。

第 10~12 章包括两方面内容，一是 PL/SQL 程序设计基础知识，二是 PL/SQL 高级编程特性。前者包括 PL/SQL 程序的基本结构(顺序、分支、循环)，以及游标、异常处理等。后者包括复合数据类型，构成 PL/SQL 程序的基本模块(函数、过程、包)以及触发器等。对于 PL/SQL 程序设计的讲述，均结合范例数据库给出详尽的实例。

本书由河北联合大学的李伟、安永丽和胡雄三位老师编著，其中李伟负责编写 1、3、4、5、9、10、11、12 章，安永丽负责编写第 2、6、7、8 章，胡雄负责全书的统稿工作。

作者在本书的写作过程中，可能有不当甚至错误之处，敬请读者批评指正。

目 录

第 1 章 Oracle 简介	1	2.7.1 oracle 用户以图形终端方式 登录	40
1.1 数据库的产生与发展	1	2.7.2 安装操作过程	41
1.1.1 数据管理技术的产生与发展	1	2.7.3 登录 OEM 界面验证安装	46
1.1.2 数据库技术的发展	3	2.8 Oracle 11g 数据库实例的创建	47
1.1.3 关系数据库产品简介	6	2.8.1 在 Windows 环境下创建 数据库	47
1.2 Oracle 的产生与发展	11	2.8.2 在 Linux 环境下用 DBCA 创建 数据库	47
1.3 Oracle 11g 简介	14	2.9 Oracle 11g 实例启动与关闭	55
1.3.1 Oracle 版本号的含义	14	2.9.1 Windows 操作系统下 Oracle 的 启动与关闭	55
1.3.2 Oracle 11g 的新特性	15	2.9.2 Linux 环境下 Oracle 实例的 启动	55
第 2 章 数据库的建立	18	2.9.3 Linux 环境下 Oracle 实例的 关闭	57
2.1 数据库的逻辑设计	18	2.9.4 Linux 环境下 Oracle 监听进程 的启动和关闭	58
2.1.1 关系数据库设计基础	18	第 3 章 SQL 与 PL/SQL 概述	60
2.1.2 关系数据库规范化	20	3.1 SQL 与 PL/SQL 简介	60
2.2 数据库的物理设计	23	3.1.1 SQL 简介	60
2.3 数据库的实施	25	3.1.2 PL/SQL 简介	63
2.4 Oracle 11g 在 Windows 下的安装 要求	26	3.2 SQL*Plus 简介	65
2.5 在 Windows 环境下安装 Oracle 11g	27	3.3 范例数据库表的建立	67
2.6 在 Linux 环境下安装前的准备	31	3.3.1 Oracle 11g 基本数据类型	67
2.6.1 检查硬件要求	32	3.3.2 定义表的结构	69
2.6.2 检查操作系统要求	33	3.3.3 查看表结构	72
2.6.3 安装 Red Hat AS4 操作系统时 的注意事项	36	3.3.4 删除表	73
2.6.4 检查安装 Oracle 11g 所必需 的包	37	3.3.5 添加数据	74
2.6.5 配置内核参数	38	3.3.6 查看数据	77
2.6.6 创建安装数据库所需要的 用户(组)	38	3.3.7 删除数据	79
2.6.7 添加 oracle 用户的限制参数	38	第 4 章 单表查询	81
2.6.8 创建安装所需要的目录	39	4.1 简单查询	81
2.6.9 设置 Oracle 的环境变量	39		
2.6.10 解压缩安装包	40		
2.7 在 Linux 环境下安装 Oracle 11g	40		



4.1.1 查询指定列.....	81	7.1.1 插入数据(INSERT).....	141
4.1.2 改变输出.....	85	7.1.2 更新数据(UPDATE).....	143
4.1.3 空值(NULL)处理.....	87	7.1.3 删除数据(DELETE、 TRUNCATE TABLE).....	144
4.2 条件查询.....	90	7.1.4 数据库完整性.....	145
4.2.1 单一条件查询.....	91	7.1.5 含有子查询的 DML 语句.....	147
4.2.2 复合条件查询.....	95	7.2 数据事务处理.....	149
4.3 记录排序.....	99	7.2.1 显式处理事务.....	150
4.3.1 按单一列排序.....	99	7.2.2 隐式处理事务.....	152
4.3.2 按多列排序.....	102	7.2.3 特殊事务.....	153
4.4 分组查询.....	103	第 8 章 SQL 函数	154
4.4.1 列(Aggregate)函数 及其应用.....	103	8.1 数值函数.....	154
4.4.2 GROUP BY 子句.....	105	8.1.1 数值函数概述.....	154
4.4.3 HAVING 子句.....	107	8.1.2 数值函数示例.....	154
第 5 章 子查询与集合操作	109	8.2 字符函数.....	160
5.1 子查询.....	109	8.2.1 字符函数概述.....	160
5.1.1 单行子查询.....	109	8.2.2 字符函数示例.....	161
5.1.2 多行子查询.....	111	8.3 日期时间函数.....	166
5.1.3 多列子查询.....	113	8.3.1 日期时间函数概述.....	166
5.1.4 相关子查询.....	113	8.3.2 日期时间函数示例.....	166
5.1.5 嵌套子查询.....	115	8.4 转换函数.....	170
5.2 集合操作.....	116	8.4.1 转换函数概述.....	170
5.2.1 使用集合操作符.....	117	8.4.2 转换函数示例.....	171
5.2.2 集合操作的进一步讨论.....	123	第 9 章 数据控制语言与数据定义 语言	176
第 6 章 连接查询	128	9.1 数据控制语言(DCL).....	176
6.1 内连接查询.....	128	9.1.1 数据库的权限.....	176
6.1.1 简单内连接.....	129	9.1.2 权限控制.....	179
6.1.2 复杂内连接.....	130	9.2 表.....	184
6.2 外连接查询.....	132	9.2.1 建立表.....	184
6.2.1 左外连接.....	133	9.2.2 获得表的相关信息.....	189
6.2.2 右外连接.....	134	9.2.3 修改表定义.....	191
6.2.3 全外连接.....	135	9.2.4 修改表名.....	198
6.3 其他特殊连接.....	137	9.2.5 删除表.....	198
6.3.1 交叉连接.....	137	9.3 索引.....	198
6.3.2 自连接.....	139	9.3.1 建立索引.....	198
第 7 章 数据操纵语言与事务处理	141	9.3.2 获得索引信息.....	199
7.1 数据操纵语言(DML).....	141	9.3.3 修改索引名字.....	200

9.3.4 删除索引.....	200	11.1.1 定义记录	262
9.4 视图.....	201	11.1.2 在 SELECT 语句中使用 记录	264
9.4.1 建立视图.....	201	11.1.3 在 DML 中使用记录.....	267
9.4.2 使用视图.....	202	11.2 记录表类型.....	271
9.4.3 获得视图定义信息.....	206	11.2.1 定义记录表	271
9.4.4 修改视图.....	207	11.2.2 使用记录表	272
9.4.5 删除视图.....	207	11.3 联合数组类型.....	275
第 10 章 PL/SQL 编程基础	208	11.3.1 定义联合数组	275
10.1 PL/SQL 语言基础	208	11.3.2 使用联合数组	275
10.1.1 PL/SQL 块简介	208	11.4 嵌套表类型.....	277
10.1.2 PL/SQL 基本语法要素	211	11.4.1 定义嵌套表	278
10.1.3 变量及其数据类型.....	214	11.4.2 使用嵌套表	278
10.2 在 PL/SQL 中执行 SQL 语句.....	218	11.5 变长数组类型.....	282
10.2.1 执行 SELECT 语句.....	218	11.5.1 定义变长数组	282
10.2.2 执行 DML 语句.....	220	11.5.2 使用变长数组类型	282
10.2.3 执行事物处理语句.....	226	11.6 集合操作.....	286
10.3 PL/SQL 程序控制结构	227	11.6.1 集合属性与方法	286
10.3.1 顺序结构.....	227	11.6.2 使用集合操作符	294
10.3.2 分支结构.....	228	第 12 章 应用程序结构	301
10.3.3 循环结构.....	233	12.1 子程序.....	301
10.3.4 GOTO 语句与 NULL 语句... ..	236	12.1.1 过程	301
10.4 异常处理.....	238	12.1.2 函数	312
10.4.1 异常的基本概念.....	238	12.2 包.....	321
10.4.2 系统异常处理.....	242	12.2.1 定义包	321
10.4.3 自定义异常处理.....	245	12.2.2 包的管理	323
10.4.4 使用异常函数.....	247	12.2.3 调用包	326
10.5 游标.....	248	12.2.4 包中子程序的重载	327
10.5.1 游标应用基础.....	248	12.3 触发器.....	329
10.5.2 游标应用.....	250	12.3.1 概述	329
10.5.3 游标 FOR 循环.....	253	12.3.2 DML 触发器.....	333
10.5.4 游标的复杂应用.....	255	12.3.3 INSTEAD OF 触发器	336
第 11 章 复合数据类型	262	12.3.4 系统事件触发器	338
11.1 记录类型.....	262		

第 1 章 Oracle 简介

Oracle 公司目前是世界第二大独立软件公司和世界领先的信息管理软件供应商。Oracle 数据库是著名的关系数据库产品。其市场占有率名列前茅。在介绍 Oracle 公司及其数据库产品之前，首先介绍数据库的产生与发展。

1.1 数据库的产生与发展

在计算机产生以前，人们利用纸和笔来记录数据，利用常规的计算工具来进行数据计算，并主要是通过人工来管理这些数据。

研制计算机的最初目的是利用它进行数值计算，但随着计算技术的进步和发展，计算机的应用已远远地超出了数值计算的范围。

在计算机硬件、软件不断发展的基础上，受应用需求的推动，人们开始借助于计算机进行大规模的数据管理，使数据管理技术得到迅速发展。

在利用计算机进行数据管理的发展过程中，经历了人工管理、文件系统、数据库系统三个阶段。

1.1.1 数据管理技术的产生与发展

1946 年，第一台电子计算机问世。计算机问世后的前十年主要用于解决数值计算问题。到 20 世纪 50 年代后期，计算机开始应用于事务管理，用来解决非数值计算问题，如人事管理、工资管理、库存管理、辅助教学等。20 世纪 70 年代以后，计算机一方面朝着高速(数亿次/秒)、大容量和智能化的巨型计算机方向发展，另一方面又朝着品种繁多、功能不断增强的微型计算机系统方向发展。

随着科学技术的发展，计算机技术作为信息管理的先进技术，其优越性越来越明显。计算机能存储大量数据，并能长期保存，这是任何其他工具所无法比拟的，它处理数据的速度快，能够及时地为人们提供大量他们所关心的信息。计算机数据管理技术的发展大致可分为人工管理、文件系统、数据库系统 3 个阶段。

1. 人工管理阶段

20 世纪 50 年代，计算机主要用于数值计算。当时的计算机硬件功能较弱，输入输出设备简单。计算机的外存只有纸带、卡片、磁带，没有直接存取设备，尚不能支持大量数据的联机存取。在软件方面，还没有操作系统，也没有文件管理功能的软件，只能处理简单输入输出操作。数据无结构，并且数据缺乏独立性，依赖于特定的应用程序。数据处理的性质只是使用计算机代替手工劳动，如做计算工资等数值运算。数据和程序完全结合在一起，成为一个整体。它的传输和使用由程序控制完成。数据不保存，使用时随程序一起全部调入内存，使用完以后就全部撤出计算机。数据面向应用，一组数据对应于一个程序。数据之间相互独立、无关，程序之间也相互独立。数据不能共享，存在着大量数据冗余。

2. 文件系统阶段

20 世纪 50 年代后期到 20 世纪 60 年代中期, 计算机不仅用于数值计算, 而且开始大量用于数据管理。在硬件方面, 外存装置有了很大的发展, 磁鼓、磁盘、大容量的磁盘组等直接存取的存储设备, 成为主要的辅助存储装置。输入输出功能大大增强。软件方面出现了操作系统, 其中包含文件管理系统, 具有了文件管理和一定的数据管理功能。

1954 年出现了第一台商业数据处理的电子计算机 UNIVACI, 它标志着计算机开始应用于以非数值计算为主的事务处理。人们得益于计算机惊人的处理速度和大容量的存储能力, 从而摆脱了从大量传统纸张文件中寻找数据的困难, 这种基于计算机的数据处理系统也就从此迅速发展起来。该数据处理系统是把计算机中的数据组织成相互独立的、被命名的数据文件。数据文件可以按文件的名字来进行访问。对文件中的记录进行存取的数据管理技术, 可以实现对文件的修改、插入和删除, 这就是文件系统。文件系统实现了记录内的结构化, 即给出了记录内各种数据间的关系。但从文件的整体来看, 却是无结构的。其数据面向特定的应用程序, 因此数据共享性和独立性差, 并且数据冗余度大, 管理和维护的代价也很大。

这一阶段的基本特征是, 数据不再是程序的组成部分。数据有结构、有组织地构成文件形式, 由操作系统自动存放在磁带或磁盘上, 并为各个文件起一个名字加以标识。文件管理系统是应用程序和数据之间的一个接口, 应用程序必须通过文件管理系统才能建立和存储文件; 而应用程序也只有在文件管理系统的支持下才能使用数据文件中的数据。

通过文件管理系统对数据文件实行统一管理, 是数据管理技术的一个重大进步, 但是数据文件还是面向应用的, 基本上对应于一个或几个特定的应用程序。文件与应用程序之间存在着密切的相互依赖关系。文件一旦离开了它所依附的程序, 便会失去存在的价值。数据文件之间彼此独立存在, 即文件只能反映现实世界中客观存在的事物及其特征, 却不能反映出各种事物之间的客观存在的本质联系。因而各数据文件中同一数据的重复出现就不可避免, 浪费存储空间, 并且文件之间还会存在不相容性。此外, 由于每次输入输出存取的只是文件记录, 因此文件系统的操作不能精细到记录中的字段, 也不能使不同数据文件之间的记录产生联系, 而必须在数据处理应用程序中对此做出安排。

此阶段文件存取的方式既可以是顺序方式, 也可以是随机方式。数据的逻辑结构不同于物理结构, 它们之间有了变换, 但关系相当简单。

3. 数据库系统阶段

文件系统数据管理方式存在着一系列缺点, 如各数据文件之间存在着重复数据, 应用程序仍依赖于数据, 增加了程序的维护工作; 由于更新重复数据而产生数据的不一致性, 不但需要使用专用程序来检查数据, 增加了工作量, 而且各数据文件集中成一个数据整体时, 还存在着如何保持各数据文件的匹配及保密问题, 以及在信息的及时性方面有一定的限制等。

20 世纪 60 年代后期, 数据库技术得到了迅速发展和广泛的应用。数据库系统的出现, 一方面是由于社会对于数据管理技术发展需要, 另一方面也是因为计算机硬件与软件的迅速发展, 为数据库技术提供了充分的条件。在这一阶段, 已完全使用大容量和快速存取的磁盘作为存储装置。这样的存储装置有很强的输入输出能力。在软件方面出现了面向数据

管理的数据库管理系统。

数据库管理系统克服了文件系统管理数据时的不足，解决了实际应用中多个用户、多个应用程序共享数据的要求，从而使数据能为尽可能多的应用程序服务。数据库的特点是数据不再只针对某一特定应用，而是面向全组织，具有整体的结构性，共享性高，因此冗余度小，具有一定的程序与数据间的独立性，并且实现了对数据进行统一的控制。数据库技术的应用使数据存储量猛增，用户增加，而且数据库技术的出现使数据处理系统的研制从围绕加工数据的程序为中心转向围绕共享的数据来进行。这样，既便于数据的集中管理，又有利于应用程序的研制和维护，从而提高了数据的利用率和相容性，并且有可能从企业或组织的全局来利用数据，从而提高了决策的可靠性。

从文件系统到数据库系统，标志着数据管理技术质的飞跃。目前不仅在大、中型计算机上实现并应用了数据管理的数据库技术，而且在微型计算机上也配置了功能较强的数据管理软件，如常见的 Visual FoxPro、Access 等数据库管理系统等，像 Oracle 这样的大型数据库也开发出基于微机的版本，促使数据库技术得到广泛的应用和普及。

综上所述，数据管理发展阶段及特点可以归纳为如表 1.1 所示。

表 1.1 数据管理发展阶段及特点

	人工管理	文件管理	数据库管理
数据的管理者	用户(程序员)	文件系统	数据库系统
数据的针对者	特定应用程序	面向某一应用	面向整体应用
数据的共享性	无共享	共享差，冗余大	共享好，冗余小
数据的独立性	无独立性	独立性差	独立性好
数据的结构化	无结构	记录有结构，整体无结构	整体结构化

1.1.2 数据库技术的发展

数据库技术最初产生于 20 世纪 60 年代中期。40 多年来，数据库技术已成为计算机科学技术中发展最快的重要分支之一，是现代计算机信息系统和计算机应用系统的基础和核心。它从第一代的网状、层次数据库技术和第二代的关系数据库技术，发展到第三代的面向新一代应用的数据库技术。

数据库技术与网络通信技术、人工智能技术、面向对象程序设计技术、并行计算技术等互相渗透，有机地结合，成为当代数据库技术发展的重要特征。

根据数据模型的变化，数据库技术的发展可以划分为 3 个阶段：第一代的网状层次数据库系统，第二代的关系数据库系统，第三代的以面向对象模型为主要特征的数据库系统。

1. 第一代数据库系统

第一代数据库系统是 20 世纪 60、70 年代研制的层次和网状数据库系统。1964 年，美国通用电气公司开发成功了世界上的第一个数据库系统 IDS(Integrated Data Store)。IDS 奠定了网状数据库的基础，并且得到了广泛的应用，成为数据库系统发展史上的一座丰碑。

1969 年，IBM 研制的基于层次模型的数据库管理系统 IMS(Information Management

System)问世。这是最早的一个典型数据库系统,具有代表性。1969年,美国数据库系统语言协会 CODASYL(Conference On DATA SYstem Language)的数据库任务组 DBTG(Database Task Group)提出了网络数据模型的数据库规范,并于1971年4月公布了它的研究成果 DBTG 报告。该报告是一个重要文献,它以文件形式确定了数据库设计的 DBTG 方法,即网络方法。在该报告中,真正地把数据库和文件区别开来。同年5月 CODASYL 成立了“数据库语言任务组”(DBLTG)接替 DBTG 的工作,进一步开发 DBTG 规范。在 DBTG 方法和思想的指引下,网络数据库系统的实现技术不断成熟,出现了许多商品化的网络数据库管理系统。

IBM 公司的层次模型的数据库管理系统 IMS(Information Management System)和美国数据库系统语言协会 CODASYL 的数据库任务组 DBTG 提出了层次数据库的数据模型与网络数据模型的数据库规范,确定并建立了层次数据库和网络数据库系统的许多概念、方法和技术。它们是层次数据库和网状数据库的典型代表。

层次数据库的数据模型是有根的定向有序树,网状模型对应的是有向图。这两种数据库奠定了现代数据库发展的基础。这两种数据库具有如下共同点:支持外模式、模式、内模式等三级模式,保证数据库系统具有数据与程序的物理独立性和一定的逻辑独立性;用存取路径来表示数据之间的联系;具有独立的数据定义语言,包括模式数据定义语言、子模式数据定义语言;导航式的数据操纵语言。

2. 第二代数据库系统

第二代数据库系统是关系数据库系统。1970年6月 IBM 公司 San Jose 实验室软件研究所的高级研究员 E.F.Codd 发表了题为“大型共享数据库数据的关系模型”的论文,文中提出了关系数据模型,定义了某些关系代数运算,开创了关系数据库方法和关系数据库理论,为关系数据库技术奠定了理论基础。

20世纪70年代是关系数据库理论研究和原型系统开发的时代。IBM 公司 San Jose 实验室在 IBM370 系列计算机上研制的 System R 是成功的关系数据库系统的代表作。另外,加利福尼亚大学伯克利分校还研制出关系数据库系统 Ingres。经过大量的高层次研究和开发,关系数据库系统的研究取得了一系列的成果。

(1) 奠定了关系模型的理论基础,确立了完整的关系模型理论、数据依赖理论和关系数据库的设计理论。

(2) 关系模型的概念单一化,实体的描述和实体之间的联系均用关系来表示;而网状、层次数据模型用节点描述实体,用存取路径来表示实体之间的联系。

(3) 提出了关系数据库语言,关系数据库语言是非过程化的。如关系代数、关系演算、SQL 语言、QBE 语言等。这些描述性语言一改以往程序设计语言和网状、层次数据库语言的面向过程的风格,以其易学易懂的优点得到了最终用户的欢迎,为20世纪80年代数据库语言标准化打下了基础。

(4) 研制了大量的关系数据库系统原型,攻克了系统实现中查询优化、并发控制、故障恢复等一系列关键技术。这不仅大大丰富了数据库管理系统的实现技术和数据库理论,更重要的是促进了关系数据库系统产品的蓬勃发展和广泛应用。

20世纪70年代后期,关系数据库系统从实验室走向了社会。因此,计算机领域中很

多人把 20 世纪 70 年代称为关系数据库时代, 20 世纪 80 年代几乎所有新开发的数据库系统均是关系型的。这些数据库系统的运行, 使数据库技术日益广泛地应用到企业管理、情报检索、辅助决策等各个方面, 成为信息系统和计算机应用系统的重要基础。

3. 第三代数据库系统

第三代数据库产生于 20 世纪 80 年代。鉴于数据库技术在商业领域的巨大成就, 各个行业对数据库技术提出了更多的需求, 第二代数据库系统已经不能完全满足需求, 于是产生了第三代数据库系统。

第三代数据库支持多种数据模型, 如关系模型和面向对象的模型; 并与许多新技术相结合, 如分布式处理技术、并行计算技术、人工智能技术、多媒体技术、模糊技术; 广泛应用于多个领域, 如商业管理、GIS、计划统计等, 由此也衍生出多种新的数据库技术, 如计算机辅助设计与制造系统、计算机集成制造系统、计算机辅助软件工程、地理信息系统、办公自动化和面向对象程序设计环境等。

分布式数据库允许用户开发的应用程序把多个物理上分开的、通过网络互联的数据库当作一个完整的数据库看待。并行数据库通过 `cluster`(集群) 技术把一个大的事务分散到 `cluster` 中的多个节点去执行, 提高了数据库的吞吐和容错性。

多媒体数据库提供了一系列用来存储图像、音频和视频对象的类型, 能更好地对多媒体数据进行存储、管理、查询。模糊数据库是存储、组织、管理和操纵模糊数据的数据库, 可以用于模糊知识处理。

以上这些领域需要的数据管理功能有相当一部分是传统数据库所不能满足的, 它们主要有以下特征。

- (1) 复杂对象的存储和处理。复杂对象不仅内部结构复杂, 相互之间的联系也很复杂。
- (2) 复杂数据类型的支持。复杂数据类型包括抽象数据类型、无结构的超长数据、时间、图形、图像、声音和版本数据等。
- (3) 数据、对象、知识的统一管理。
- (4) 长事务和嵌套事务的处理。
- (5) 程序设计语言和数据库语言无间隙的集成。
- (6) 巨型数据库(数据量可超过 10TB)的管理。

4. 未来数据库技术的发展

20 世纪 90 年代以后, 数据库应用环境发生了巨大的变化, Internet/Web 向数据库领域提出了前所未有的挑战, 一大批新一代数据库应用应运而生, 如支持高层决策的数据仓库、OLAP 分析、数据挖掘、数字图书馆、电子出版物、电子商务、Web 医院、远程教育、基于 AdHoc 无线网的移动数据库、Web 上的数据管理与信息检索、数据流管理等。

新一代应用提出的挑战极大地激发了数据库技术的研究和开发, 使数据库技术的研究和开发出现了一个新的高潮, 出现了一大批具有 Internet 时代特征的数据库技术和相应的数据库管理系统, 如 Web 信息检索技术与系统、Web 数据集成与共享技术与系统、数据流技术与系统、电子商务和电子政务技术与系统、数字图书馆技术与系统、安全数据库技术与系统等。

1.1.3 关系数据库产品简介

在数据库发展的历史上,有层次数据库、网状数据库和关系数据库三种主要类型的数据库。但到目前为止,在世界范围内得到主流应用的还是关系数据库系统,比较知名的如 Microsoft 公司的 Access、Visual FoxPro、SQL Server, Oracle 公司的 Oracle 数据库, Sybase 公司的 Sybase 数据库产品, Informix 公司的 Informix 数据库产品, IBM 公司的 DB2 以及完全免费的 MySQL 数据库等。这些数据库产品可以分为桌面关系型数据库系统和网络关系型数据库系统两类。

Access、Visual FoxPro 等小型数据库产品被称为桌面关系型数据库系统,其主要特点是广泛应用在单机环境,不提供或仅提供有限的网络应用功能。计算机操作系统为 Windows XP、Windows 7/8 等。基于桌面关系型数据库开发的数据库应用系统主要目的是为了满足日常小型办公需要。桌面关系型数据库安全措施较弱,开发工具与数据库集成在一起,既是数据库管理工具,又是数据库应用开发的前端工具。这种内嵌式开发工具提供的数据库管理功能较弱。如在 Visual FoxPro 里就集成了应用开发工具,在 Access 2013 里集成了脚本语言等。桌面关系型数据库侧重于可操作性、易开发性和管理简单性等。

SQL Server、Oracle、Informix、DB2、MySQL 等为网络关系型数据库系统,与传统意义上的桌面关系型数据库系统相比,具有以下主要特点。

它们一般需要网络操作系统支持,如 Windows NT Server、Windows 2008 Server、Linux Server、Unix 等。数据库系统管理工具、前端开发工具和后台数据库是可以分离的,通常所说的网络数据库管理系统指的是管理工具和后台数据库。同时,它们具有强大的网络功能和分布式功能,可以根据软、硬件和网络环境的不同,组织各种技术先进,支持超大规模数据库技术、并行查询和多线程的服务器;它们还提供完备的数据安全性方案,完善的数据库备份和恢复手段。

网络关系型数据库在大中型计算机上开发、使用的历史较长,功能日臻完善。这类数据库移植到微机上以后,仍然保留了它们在大中型系统中的特点,即具有数据库管理工具功能强大、用户操作灵活方便、完整的安全性/一致性/可靠性保障措施、运行效率高/速度快、系统功能完善等特点。

下面分别介绍这些典型的数据库产品。

1. Access 数据库

Microsoft Office Access(原名 Microsoft Access)是 Microsoft 公司推出的基于 Windows 的桌面关系数据库管理系统(RDBMS),它结合了 Microsoft Jet Database Engine 和图形用户界面两项特点,是 Microsoft Office 的成员之一。Microsoft Access 1.0 版本在 1992 年 11 月发布。目前,最新版本的 Access 数据库为 Microsoft Office Access 2013。

Access 数据库具有界面友好、易学易用、开发简单、接口灵活等特点,是典型的新一代桌面数据库管理系统。其主要特点如下。

(1) 具有完善的各种数据库对象管理、数据组织、用户管理、安全检查等功能。

(2) 强大的数据处理功能。在一个工作组级别的网络环境中,使用 Access 数据库开发的多用户数据库管理系统具有传统的 XBASE(DBASE、FoxBASE 的统称)数据库系统所无

法实现的客户/服务器(Client/Server)结构和相应的数据库安全机制, Access 数据库具备了许多先进的大型数据库管理系统所具备的特征, 如事务处理、错误恢复能力等。

(3) 可以方便地生成各种数据对象, 利用存储的数据建立窗体和报表, 可视性好。

(4) 作为 Office 套件的一部分, 可以与 Office 集成, 实现无缝连接。

(5) 能够利用 Web 检索和发布数据, 实现与 Internet 的连接。

Access 主要适用于中小型应用系统, 或作为客户机/服务器系统中的客户端数据库。

2. Visual FoxPro 数据库

自 1989 年 Microsoft 公司推出 FoxPro 1.0 以来, 由于它具有简单易学、功能强大、界面友好等特点, 一直受到我国广大数据库用户的关注, 也是中小型数据库应用系统开发的首选编程语言。

1995 年推出的 Visual FoxPro 3.0 更是 FoxPro 系列产品的一次历史性突破, 因为它首次在 XBASE 语言中引入了面向对象编程技术, 采用了可视化的概念, 首次明确提出支持客户机/服务器体系结构, 并且彻底更新了“数据库”的概念, Visual FoxPro 3.0 无疑是 FoxPro 系列产品发展的一个重要里程碑。

目前, 最新版本的 Visual FoxPro 数据库为 Visual FoxPro 10.0。

Visual FoxPro 具有良好的兼容性, 这意味着使用以往版本开发的应用程序在 Visual FoxPro 新版本中仍然可以正确地运行。但仅仅运行以往版本开发的应用程序并不能真正地体现 Visual FoxPro 新版本的优越性和强大功能。所以在使用 Visual FoxPro 新版本时, 了解它所包含的新功能是很有必要的。Visual FoxPro 具有如下特点。

(1) 面向对象编程技术(Object Oriented Programming)

Visual FoxPro 的最大特点是使用面向对象编程技术(OOP), 用户可以首先构造类, 通过继承可以产生子类。每一个类都包含一系列属性、事件和方法。由类创建的对象几乎可以完成用户所有要实现的任务。通过封装, 可以把对象内部的复杂性隐藏起来。使用 OOP 方法, 通过继承使得代码的重用性大大提高, 最大限度地减少了代码出错的可能性。Visual FoxPro 提倡使用面向对象编程技术的同时, 也支持以往版本所使用的结构化程序设计。

(2) 可视化(Visual)编程技术

在使用 Windows 时, “所见即所得”是我们体会非常深刻的。在使用 Visual FoxPro 设计一个用户界面时, 无须运行程序, 就可看到用户的最终界面, 这也是 FoxPro 前面冠以“Visual”的缘故。

(3) 重新定义数据库的概念

在 FoxPro 最初的版本中, 数据库就是一个二维表, 表中的每一行数据表示一条记录, 每一列表示一个字段。数据库仅仅是数据的集合。而在 Visual FoxPro 新版本里, 数据库的概念被彻底更新了。数据库是由若干个表及表与表之间的关系、视图、连接、存储过程等组成的一个集合。以往版本的数据库在这里被称为自由表。自由表是独立于数据库之外而存在的表, 而属于某一数据库的表被称为数据库表。数据库表有着与自由表不同的许多属性。比如, 数据库表可以定义长表名、可以包含有效性规则等。

数据库概念的重新定义, 使得采用面向对象编程技术奠定了重要的理论基础, 也使得采用面向对象编程技术成为可能。因为在面向对象编程技术中, 对象必须有自己的属性和

方法，而 Visual FoxPro 所定义的数据库既有自己的属性，如记录，又有自己的方法，如有效性规则，所以 Visual FoxPro 所定义的数据库也成为可处理的对象。

(4) 有效性规则(Validation Rule)

Visual FoxPro 数据库表的有效性规则有两种类型：字段级规则和记录级规则。当为某一个数据库表的某一字段定义了字段级规则时，如果输入或修改该字段值，对应的字段级规则将被激活。它可用来检验字段值输入的正确性。记录级规则是与记录相关的有效性规则，当插入或修改记录时被激活，它用来检验记录数据输入的正确性。

有效性规则只适用于数据库表，在自由表里不存在有效性规则。有效性规则是数据库的一部分。

(5) 触发器(Trigger)

触发器是在一个插入、更新或删除操作之后运行的记录级事件代码。不同的事件对应不同的动作，比如，插入操作可以激活插入触发器，删除操作可以激活删除触发器等。

触发器在有效性规则之后运行，它们常用于检验已建立永久性关系的数据库表之间的数据完整性。

触发器只适用于数据库表，在自由表中不存在触发器。触发器是数据库的一部分。

(6) 存储过程(Stored Procedure)

存储过程是指存储在数据库里的过程。该过程可以包含任何允许在用户自定义函数中出现的命令和函数。比如，触发器代码就存放在存储过程里。在远程数据源上，存储过程是指在任何 SQL 数据库中被命名的一组 SQL 语句的集合。

存储过程属于数据库的一部分。当打开数据库时，存储过程所包含的过程和函数就立即产生作用。

(7) 事务(Transaction)

数据库从一个有效性状态到另一个有效性状态的操作集称为事务。在对数据库进行操作时，通常是以一条命令或一个函数作为一个操作单位，这种操作所产生的结果往往是不可恢复的。而事务是以一组命令或函数作为一个操作单位，如果在事务处理过程中发生意外，即可取消事务，让数据库恢复初始状态。事务处理只适用于数据库操作。

(8) 本地和远程视图(Locate and Remote View)

视图是一个定制的、虚拟的、可更新的结果集。本地视图是与本地数据源相关联的视图。远程视图是与远程数据源相关联的视图。另外还有参数化视图，它是指根据输入的参数而查询得到的结果集。在实现客户机/服务器应用程序时，视图是一个非常有效的方法。

(9) 向导(Wizard)

学习和使用 Visual FoxPro 的快捷方法是使用系统所提供的向导。例如，当不知道如何设计一个表时，使用表向导可以一步一步地根据 Visual FoxPro 对话框里所提示的内容进行设计。在向导对话框里，包括了一些需要回答的简单内容，通过若干个步骤，即可完成指定任务的操作。

(10) 项目管理器

使用 Visual FoxPro 时会创建很多文件，这些文件有着许多不同的格式，因此就需要专门的管理工具来提高工作效率。使用 Visual FoxPro 的主要工作界面——项目管理器，将 Visual FoxPro 的文件用图示与分类的方式，依据文件性质的不同，把文件放置在不同的标

签上，并针对不同类型的文件提供不同的操作选项。

项目管理器采用可视化和自由导航，各项以类似大纲的视图形式组织，通过展开或折叠，可以清楚地查看项目在不同层次上的详细内容。项目管理器提供简易可见的方式组织处理表、表单、数据库、报表、查询、源程序、菜单程序等文件，用于管理表和数据库或创建应用程序。使用设计工具和向导能够很容易地快速产生结果。

最好把应用程序中的文件都组织到项目管理器中，这样便于查找。程序开发人员可以用项目管理器把应用程序的多个文件组织成一个文件，生成一个*.app 文件或者*.exe 文件，其中*.app 文件可以用 DO 命令来执行。可以用 Visual FoxPro 专业版编译成*.exe 文件。应用程序中的所有文件如*.prg、报表格式文件和标签格式文件，都能组合在一个文件中。

(11) 表单(Form)

每一个应用程序至少要包含一个用户界面。在以往版本里，屏幕(Screen)是设计用户界面的核心部分。但屏幕往往无法响应用户的许多动作。在 Visual FoxPro 新版本中，以表单(Form)代替了屏幕(Screen)。表单作为一个可处理的对象，有许多个属性、事件和方法。同时，表单又可以包含多个控件，每一个控件通过各自的属性、事件和方法来实现用户指定的动作。表单不应被认为仅仅是功能扩大了屏幕，它是一个全新的概念，是在界面设计上实现面向对象编程技术的最充分体现。

3. SQL Server 数据库

SQL Server 是美国 Microsoft 公司推出的一种关系型数据库系统。SQL Server 是一个可扩展的、高性能的、为分布式客户机/服务器计算所设计的数据库管理系统，实现了与 Windows NT / Windows Server 的有机结合，提供了基于事务的企业级信息管理系统方案。

其主要特点如下。

- (1) 高性能，可充分利用 Windows NT / Windows Server 的优势。
- (2) 系统管理先进，支持 Windows 图形化管理工具，并且支持本地和远程的系统管理和配置。
- (3) 可靠的事务处理功能，采用各种方法保证数据的完整性。
- (4) 支持对称多处理器结构、存储过程、ODBC，并具有自主的 SQL 语言。SQL Server 以其内置的数据复制功能、强大的管理工具、与 Internet 的紧密集成和开放的系统结构，为广大的用户、开发人员和系统集成商提供了一个出众的数据库平台。

4. Oracle 数据库

Oracle 数据库是美国 Oracle 公司研制的一种关系型数据库管理系统(RDBMS)，是一个协调服务器和用于支持任务决定型应用程序的开放型 RDBMS。它可以支持多种不同的硬件和操作系统平台，从台式机到大型机和超级计算机，为各种硬件结构提供高度的可伸缩性，支持对称多处理器、群集多处理器、大规模处理器等，并提供广泛的国际语言支持。

Oracle 数据库是一个多用户系统，能自动从批处理或在线环境的系统故障中恢复运行。系统提供了一个完整的软件开发套件，包括交互式应用程序生成器、报表打印软件、字处理软件以及集中式数据字典，用户可以利用这些工具生成自己的应用程序。Oracle 以二维表的形式表示数据，并提供了以 SQL(结构化查询语言)完成数据查询、操作、定义和控制



的基本数据库管理功能。Oracle 数据库具有很好的可移植性,通过它的通信功能,微型计算机上的程序可以与小型乃至大型计算机上的 Oracle 相互传递数据。

Oracle 属于大型数据库系统,主要适用于大、中型应用系统,或作为客户机/服务器系统中服务器端的数据库系统。

5. Sybase 数据库

Sybase 数据库是美国 Sybase 公司研制的一种关系型数据库系统,是一种典型的 Unix 或 Windows NT(Windows Server)平台上客户机/服务器环境下的大型数据库系统。

Sybase 提供了一套应用程序编程接口和库,可以与非 Sybase 数据源及服务器集成,允许在多个数据库之间复制数据,适于创建多层应用。系统具有完备的触发器、存储过程、规则以及完整性定义,支持优化查询,具有较好的数据安全性。Sybase 通常与 Sybase SQL Anywhere 用于客户机/服务器环境,前者作为服务器数据库,后者为客户机数据库。采用该公司研制的 PowerBuilder 为开发工具,在我国大中型系统中具有广泛的应用。

6. Informix 数据库

Informix 数据库是美国 Informix Software 公司研制的关系型数据库管理系统。Informix 有 Informix-SE 和 Informix-Online 两种版本。Informix-SE 适用于 Unix 和 Windows NT (Windows Server)平台,是为中小规模的应用而设计的,Informix-Online 在 Unix 操作系统下运行,可以提供多线程服务器,支持对称多处理器,适用于大型应用。

Informix 可以提供面向屏幕的数据输入询问及面向设计的询问语言报告生成器。数据定义包括定义关系、撤消关系、定义索引和重新定义索引等。Informix 不仅可以建立数据库,还可以方便地重构数据库,系统的保护措施十分健全,不仅能使数据得到保护而不被权限外的用户存取,而且能重新建立丢失了的文件及恢复被破坏了的数据。其文件的大小不受磁盘空间的限制,域的大小和记录的长度均可达 2KB。采用加下标顺序访问法,与 COBOL 软件兼容,并支持 C 语言程序。Informix 可移植性强、兼容性好,Informix-SE 在很多微型计算机和小型机上得到应用,尤其适用于中小型企业的人事、仓储及财务管理。

7. DB2 数据库

DB2(DATABASE 2)数据库是 IBM 公司研制的一种关系型数据库系统。IBM 在 1983 年发布了 DB2 for MVS,2007 年发布最新版本 DB2 9.5。DB2 主要应用于大型应用系统,具有较好的可伸缩性,可支持从大型机到单用户环境,应用于 OS/2、Windows 等平台下。

DB2 提供了高层次的数据利用性、完整性、安全性、可恢复性,以及小规模到大规模应用程序的执行能力,具有与平台无关的基本功能和 SQL 命令。DB2 采用了数据分级技术,能够使大型机数据很方便地下载到局域网(LAN)数据库服务器,使得客户机/服务器用户和基于 LAN 的应用程序可以访问大型机数据,并使数据库本地化及远程连接透明化。DB2 以拥有一个非常完备的查询优化器而著称,其外部连接改善了查询性能,并支持多任务并行查询。

DB2 具有很好的网络支持能力,每个子系统可以连接十几万个分布式用户,可同时激活上千个活动线程,对大型分布式应用系统尤为适用。