

钟守义 主 编

ORACLE

数据库系统

浙江大学出版社

ORACLE

ORACLE 数据库系统

钟守义主编

浙江大学出版社

(浙)新登字 10 号

ORACLE 数据库系统

钟守义主编

责任编辑 尤建忠 韩 东

浙江大学出版社出版

浙江大学出版社电脑室排版

富阳何云印刷厂印刷

浙江省新华书店发行

* * *

开本:787×1092 1/16 印张:26.25 字数:706 千

1991 年 12 月第 1 版 1991 年 12 月第 1 次印刷

印数:0001—4000

ISBN 7-308-00942-4/TP·066

定价:12.00 元

前 言

工厂企业、事业单位、金融机构、政府部门在存贮、处理和使用浩如烟海的数据和信息时,日益感到,如果缺乏组织得良好的数据库,它们将越来越难于应付各种各样要求越来越高的业务工作。这正是数据库理论与实践在近年来得到飞速发展的根本原因。

ORACLE 是当今世界各国最为广泛流行的数据库系统之一。它有如下几个特点:

1. 比较严格地实现了关系数据库的概念,保证了方便、安全、保密、一致地使用数据;
2. 独立于计算机型,甚至独立于操作系统,这使它可在数十家计算机厂家生产的大中小微型机的各种操作系统的环境下使用;
3. 提供了大量开发工具,便于用户快速开发各种应用系统。

由于这些特点,我国教育以及工业各界普遍接受了 ORACLE,并在许多部门作为优先推荐使用的数据库系统。

本书是作者在多年从事 ORACLE 数据库教学、使用和研究的基础上,历时两载,编写而成。

参加本书编写的有钟守义(第一章一、二、四节,第三章五至八节)、张建林(第二章)、郑岗(第一章三节,第三章一至四节,第七章)、卢向南(第四章,第五章)、葛亚力(第六章)。全书由钟守义担任主编,负责全稿的审核与总纂,并对全书内容进行了技术与文字的修改。

ORACLE 数据库系统庞大而复杂,作者水平有限,成书时间仓促,疏漏错误之处在所难免,恳请读者批评指正。

作 者

1991 年 10 月于杭州浙大求是村

目 录

第一章 数据库系统与 ORACLE 概况

§ 1.1 数据库系统的概念	1
§ 1.2 ORACLE 关系数据库系统	4
§ 1.3 ORACLE 数据库系统软件的安装	8
§ 1.3.1 VAX/VMS 下 ORACLE V5.1 版的安装	8
§ 1.3.2 PC-DOS 下 ORACLE V5.1A 版的安装	12
§ 1.4 ORACLE 的技术信息与语法符号	16

第二章 交互式命令语言 SQL * Plus

§ 2.1 概述	18
§ 2.1.1 SQL * Plus 概述	18
§ 2.1.2 SQL * Plus 的启动	18
§ 2.1.3 输入命令	19
§ 2.1.4 退出 SQL * Plus	21
§ 2.1.5 一个样本例子	21
§ 2.1.6 小结	24
§ 2.2 SQL * Plus 命令语言基础	25
§ 2.2.1 数据库客体	25
§ 2.2.2 数据的类型及其比较规则	25
§ 2.2.3 数据的格式	26
§ 2.2.4 运算符和操作符	28
§ 2.2.5 SQL * Plus 中的函数	29
§ 2.2.6 哑表、伪列和常量	32
§ 2.2.7 数据字典	33
§ 2.3 数据查询	34
§ 2.3.1 选择表中的列	35
§ 2.3.2 选择表中的行	37
§ 2.3.3 表的连接和多表同时查询	45
§ 2.3.4 子查询的使用	48
§ 2.3.5 树结构的使用	53
§ 2.3.6 对查询数据的处理	56
§ 2.3.7 小结	61
§ 2.4 数据模型定义 (DDL)	61
§ 2.4.1 分区和空间定义	61
§ 2.4.2 表的建立、修改和删除	64

§ 2.4.3	使用视图	66
§ 2.4.4	使用索引加快查询速度	67
§ 2.4.5	使用簇改善性能	70
§ 2.4.6	使用同义词	73
§ 2.4.7	更改数据库客体的名字	74
§ 2.4.8	为表和列加注解	74
§ 2.4.9	小结	74
§ 2.5	数据操纵(DML)	77
§ 2.5.1	在表中插入行	77
§ 2.5.2	从表中删除行	79
§ 2.5.3	更新表中的数据	79
§ 2.5.4	小结	80
§ 2.6	数据控制(DCL)	81
§ 2.6.1	数据的共享与安全	81
§ 2.6.2	数据的一致性和完整性	85
§ 2.6.3	使用 ORACLE 审计	87
§ 2.6.4	小结	89
§ 2.7	Plus 命令的使用	90
§ 2.7.1	格式化报告	91
§ 2.7.2	编辑当前缓冲区	95
§ 2.7.3	在 SQL * Plus 中处理命令文件	98
§ 2.7.4	定义用户变量	101
§ 2.7.5	SQL * Plus 系统参数的设置和显示	101
§ 2.7.6	其它 Plus 命令的使用	105

第三章 屏幕格式生成器 SQL * Forms

§ 3.1	概述	109
§ 3.1.1	SQL * Forms 的特点	109
§ 3.1.2	格式的组成与形式	109
§ 3.1.3	SQL * Forms 的功能	110
§ 3.1.4	格式设计与运行时使用的功能键	110
§ 3.1.5	SQL * Forms 中的窗口技术	112
§ 3.1.6	设计和运行一个简单的屏幕格式	114
§ 3.1.7	SQL * Forms 的构成	116
§ 3.2	使用格式	117
§ 3.2.1	格式的运行	117
§ 3.2.2	数据查询	120
§ 3.2.3	数据插入	123
§ 3.2.4	删除记录	124
§ 3.2.5	数据修改	124

§ 3.2.6 运行格式功能键	125
§ 3.3 设计格式	127
§ 3.3.1 格式的规划	127
§ 3.3.2 格式的建立	129
§ 3.3.3 格式的修改	137
§ 3.3.4 字段的定义	146
§ 3.3.5 定义触发子	156
§ 3.4 SQL * Forms 运行参数与设计窗口	164
§ 3.4.1 SQL * Forms 的运行参数	164
§ 3.4.2 SQL * Forms 设计窗口	166
§ 3.4.3 屏幕格式的数据库形式	186
§ 3.5 触发子的基本概念	188
§ 3.5.1 事件分类	189
§ 3.5.2 触发子的级别	189
§ 3.5.3 定义触发子	190
§ 3.5.4 触发子的作用域	190
§ 3.5.5 当前概念和上下文	191
§ 3.5.6 功能键的上下文	191
§ 3.5.7 字段级触发子	192
§ 3.5.8 块级触发子	194
§ 3.5.9 格式级触发子	196
§ 3.5.10 光标导航及触发子引发机制	196
§ 3.5.11 键触发子	197
§ 3.5.12 用户命名触发子	198
§ 3.6 触发子结构与组织	199
§ 3.6.1 定义触发子步	199
§ 3.6.2 触发子步的结果	200
§ 3.6.3 触发子执行流程	200
§ 3.6.4 触发子成败原因	202
§ 3.6.5 触发子失败对格式运行的影响	202
§ 3.7 触发子命令和语法	204
§ 3.7.1 触发子中的 SQL 语句	204
§ 3.7.2 SQL * Forms 命令	204
§ 3.7.3 用户出口命令	211
§ 3.7.4 其它若干问题	217
§ 3.8 触发子设计实例	222
§ 3.8.1 格式布局设计和字段属性与验证规定	222
§ 3.8.2 触发子实例	224

第四章 菜单生成器 SQL * Menu

§ 4.1 概述	241
----------	-----

§ 4.1.1 菜单树及其优点	241
§ 4.1.2 SQL * Menu 的特性	242
§ 4.1.3 动态菜单方法	244
§ 4.1.4 运行 SQL * Menu 的基本要求	245
§ 4.2 菜单的使用	245
§ 4.2.1 使用菜单的一般步骤	245
§ 4.2.2 使用高级的 SQL * Menu 特性	246
§ 4.3 建立应用	249
§ 4.3.1 菜单设计	249
§ 4.3.2 文件管理	255
§ 4.3.3 高级菜单设计特性	256
§ 4.3.4 生成菜单文档	263
§ 4.4 SQL * Menu 系统管理	266
§ 4.4.1 对用户授权	266
§ 4.4.2 维护终端定义	268
§ 4.4.3 使用语言翻译设施	268
§ 4.4.4 库文件管理	269
§ 4.4.5 库文件及应用管理菜单	270

第五章 报表生成器 SQL * Report

§ 5.1 概述	272
§ 5.2 报表及报表控制文件的结构	272
§ 5.2.1 报表	272
§ 5.2.2 报表控制文件的结构	274
§ 5.3 报表生成过程	274
§ 5.4 报表格式化程序(RPF)及 RPF 命令	275
§ 5.4.1 报表格式化程序(RPF)	275
§ 5.4.2 RPF 命令	276
§ 5.4.3 使用报表格式化程序实例	283
§ 5.5 报表生成程序 RPT 及 SQL * Report 语句	285
§ 5.5.1 报表生成程序 RPT	285
§ 5.5.2 SQL * Report 语句	285
§ 5.6 实例	295
§ 5.7 嵌套报表	305
§ 5.8 析取报表	308

第六章 C 语言预编译程序接口 Pro * C

§ 6.1 概述	311
§ 6.1.1 SQL * Plus 和其它 ORACLE 开发工具的不足与问题	311
§ 6.1.2 Pro * C 简介	312

§ 6.1.3 Pro * C 程序的结构和实例	312
§ 6.2 应用程序首部	313
§ 6.2.1 SQL 变量说明节	314
§ 6.2.2 Pro * C 支持的数据类型	315
§ 6.2.3 数据类型的转换	318
§ 6.2.4 数据库通讯区	319
§ 6.3 应用程序本体	323
§ 6.3.1 连接 ORACLE 数据库	323
§ 6.3.2 非 DQL 语句的使用	324
§ 6.4 查询语句	328
§ 6.4.1 查询语句的组成	328
§ 6.4.2 使用游标查询的例子	331
§ 6.4.3 使用数组	333
§ 6.5 数据的一致性与保护	335
§ 6.6 错误检测和恢复	337
§ 6.7 预编译命令和 EXEC ORACLE 选项语句	339
§ 6.7.1 预编译命令语法及参数	339
§ 6.7.2 编译和连接	343
§ 6.7.3 条件预编译	344
§ 6.7.4 分别作预编译	344
§ 6.8 动态定义 SQL 语句	345
§ 6.8.1 动态定义语句的类型	345
§ 6.8.2 动态定义语句——方法 1	346
§ 6.8.3 动态定义语句——方法 2	347
§ 6.8.4 动态定义语句——方法 3	348

第七章 ORACLE 数据库管理与实用程序

§ 7.1 ORACLE RDBMS 体系结构	350
§ 7.2 数据库管理	352
§ 7.2.1 用户管理	353
§ 7.2.2 安全管理	356
§ 7.2.3 一致性与并发性	360
§ 7.2.4 磁盘空间管理	366
§ 7.2.5 数据备份与恢复	374
§ 7.2.6 数据库性能调谐	377
§ 7.2.7 数据库系统参数	380
§ 7.3 DBA 常用实用程序	384
§ 7.3.1 实用程序 IOR	384
§ 7.3.2 实用程序 SGI	385
§ 7.3.3 实用程序 CCF	387

§ 7.3.4	实用程序 AIJ	388
§ 7.3.5	实用程序 ODS	389
§ 7.3.6	实用程序 CRT	394
§ 7.4	普通用户实用程序	400
§ 7.4.1	实用程序 ODL	400
§ 7.4.2	实用程序 EXP	404
§ 7.4.3	实用程序 IMP	406

数据库系统与 ORACLE 概况

§ 1.1 数据库系统的概念

概略地说,数据库系统是一个由计算机和操作与管理人员构成的人机系统。这个系统的目的在于安全地存贮大量数据,提供各种统一处理数据的手段与工具,以便使各方面人员共享这些数据。数据库技术正在不断发展与完善,一个比较优良的数据库系统应包括计算机系统、数据模型与数据库、数据库管理系统、用户与应用程序、数据库管理员以及数据库开发工具等六个部分。下面分别对这些部分作一简要介绍。

1. 计算机系统

为使一个数据库系统现实可行地存贮和处理大量数据,需要配置一套合适的计算机系统。用在数据库系统中的最基本的计算机系统应包括主机、充分容量的磁盘存贮器以及操作系统。此外,还可包括计算机网络的软硬件配置与其它高级语言的编译程序、服务程序和通讯软件等。

2. 数据模型与数据库

顾名思义,数据库是一个存贮数据的仓库。用当代数据库技术观点来看,数据库有如下特点:

1) 大量相互之间有一定关联的数据,用一种统一的方式集中在计算机的存贮器中,形成一个具有一定结构的数据仓库。

2) 数据的管理工作已从各种业务部门分离出来,由专门人员或机构负责管理数据库中全部数据,这种人员或机构称为数据库管理员。

3) 数据库中数据由各种不同用户存入,它们将为各用户所共享。

简言之,数据库是用统一方式存贮数据的仓库,数据库中数据由各种用户共享并由统一的机构和方式进行管理。

数据库是从早年的文件系统发展而来。文件只包含数据本身,而数据库不仅包含数据,而且还要表示数据之间的各种联系。数据库通过什么方式表示数据及其联系呢?这就要采用所谓的数据模型来实现。

从表示数据之间联系的不同方式来研究,数据模型大致可分层次、网络 and 关系三大类。

层次模型 这种模型将数据划分成不同层次,下一层次上的数据从属于上一层数据,每个上层数据可与若干个下层数据相联系,但每个下层数据只能与一个上层数据相联系,用这种方式将数据组织成一个树状结构的整体。这种模型在计算机上实现简便,处理数据耗时少,但存贮的数据有大量冗余。限于对各种数据层次的安排,虽然有些数据检索的意义简单明白,但却难于做到,用户使用很不方便。

网络模型 网络模型是在层次模型基础上改进而成,虽然这种模型中,数据仍然划分成层次,上层中一个数据继续可与下层的若干个数据相联系,但下层中一个数据也可以与上面若干层中的若干个数据相联系,由此而形成网状结构。在网络模型中,数据之间存在的联系不一定是简单的从属关系,在两种数据之间可能存在一种以上的联系。这种模型在计算机上实现比层次模型困难

一些,数据冗余度比层次模型少,处理数据耗时较少。在使用这种模型描述各种业务系统中数据间联系时,经常需要设置、添加一些人为的联系,优化处理需要用户深入了解网络模型的机制与原理。

关系模型 这种模型用一些基本的二维表来存贮各种数据和表示这些数据间的联系。在这些基本二维表的基础上,根据对数据处理的要求,可以通过各种代数运算形成新的二维表,每张二维表就是一个关系,是具有某种特定联系的数据集合。这种数据模型概念简洁,使用方便,数据冗余度小,但在计算机上实现较为困难,在使用时耗时较多。

对于一个数据库,只能且只需要采用其中的一种模型存贮数据及表示数据间联系。早年数据库技术受限于计算机设备的效率,大都采用层次与网络模型,但随着计算机技术不断发展,特别是计算速度的加快和磁盘存贮技术的发展,层次模型与网络模型已日趋淘汰,关系模型已经成为数据库系统中的主流。

从数据及它们之间联系的面向来看,可将数据模型分为内部、概念、外部三种。

内部模型 数据及其间的联系在磁盘存贮器中具体表示的方式与描述,这种模型有时也称为物理模型。内部模型面向机器,它为计算机存取数据库中数据服务与使用。只有在设计或改进数据库管理系统时,才需对内部模型进行深入研究。对于一般数据库系统的用户与管理人员,既不能观察到这种模型,也无此种需要。

概念模型 这种模型是对一个数据库的逻辑描述,它从逻辑上规定数据库中数据的内容和结构方式的完整表示。它独立于各种特定的计算机系统,代表一个业务系统中各部门综合的数据观点,有时也称为逻辑模型或模式。

外部模型 有时称为子模式,它是数据库中某一部分数据及其联系的逻辑描述,常常代表特定用户对数据的观点。

对于一个数据库,必须包含这三个不同级别的数据模型。但对于处理业务的用户,只能接触外部模型。对于数据库设计人员和数据库管理员,处理的是概念模型。

3. 数据库管理系统

数据库管理系统是一种对数据库中各种数据通过管理与控制而进行存取的软件,它是数据库系统的核心。为了说明数据库管理系统的作用以及与数据库系统中其它部分之间的关系,以数据库中读取一个数据为例作一说明。

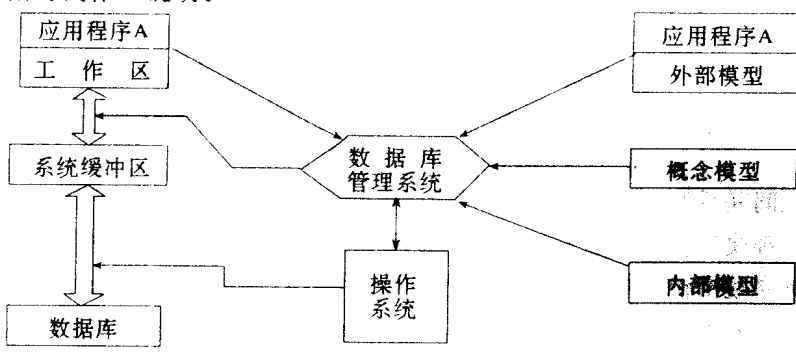


图 1.1-1 数据库管理系统作用示意图

当接到用户通过应用程序 A 发出的读取数据的命令后,数据库管理系统根据描述此数据的外部模型、概念模型与内部模型,指示操作系统向数据库发出命令,将数据读入系统缓冲区。然后,再由数据库管理系统发出将系统缓冲区中的数据送往应用程序 A 工作区的命令,详见图 1.1-1。

若要修改或插入数据,过程也与读取数据类似。在存取数据过程中,数据库管理系统还对用户

处理数据库数据的权限进行检查与控制,防止用户的越权行为,保护数据的安全性。

流行的各种数据库系统,如 INGRES、UNIFY、dBASE,以及本书介绍的 ORACLE 等,都是以所设计和实现的数据库管理系统的名字命名的。

4. 用户与应用程序

使用数据库进行各种业务处理人员与部门都是数据库系统的用户。虽然数据库管理系统是一套用来存取数据的软件,但由于其庞大与复杂,用户不可能直接使用它来处理业务,通常需对各种具体业务开发出应用程序。然后通过运行这些应用程序实现对数据的存取处理。

根据上述讨论,我们将数据库系统的用户分成两类。一类是负责设计、编制与维护应用程序的人员,简称设计人员。通常,设计人员采用经过扩充可使用数据库命令的高级语言编写各种应用程序。随着业务扩大与深化,设计人员可不断编制与维护各种应用程序。另一类是运行应用程序对业务进行具体处理的人员,本书中称为操作员。

还要注意两点,其一是许多文献中提到的系统分析人员。对于大型复杂组织中使用的数据库系统,常常需要有专人对数据和处理的要求进行深入调查与分析,提出数据库的方案与设计。不过考虑到对于这类组织中的数据库开发,常置于管理信息系统开发之下,数据库系统只是其中的组成部分(当然它是核心组成部分之一),系统分析与设计的任务常由管理信息系统开发者所负担,所以这里不再将其列入到用户中去了;其二是应用程序的开发,用高级语言编写应用程序仍然是一件费时又费力的工作,维护也比较困难,目前的趋势是由计算机厂家或软件公司提供各种开发工具,设计人员可以利用这些开发工具,快速开发出各种应用程序。

5. 数据库管理员

在一个数据库中,常常集中存放着业务系统中各部门的全部数据,成为许多用户共享的一种资源。对于这样一种数据资源的管理,比起传统的人工管理方式,则有了根本的变化。在一些比较完善的数据库系统中,已从各业务部门将数据管理职责分离出来,由一个或若干人构成的小组,专门行使数据管理职能,这个机构就是数据库管理员。

数据库管理员的基本功能与职责如下:

1) 控制与登录新用户。只有数据库管理员才能设置数据库新用户。

2) 对数据库中数据定期备份与故障后数据库的恢复。数据库中数据对于一个组织的正常运行是至关重要的,但计算机作为一种设备,出现各种故障的可能是存在的。严重的故障常常破坏了数据的一致性,一些严重故障排除后,为使数据库中数据恢复正常,就要依赖数据库管理员定期复制数据库中全部数据。此外,还要依赖操作系统管理员在日常管理中记录下的对数据库做的所有修改情况的日志。

3) 对各种用户有设置与调整使用数据的权限,保证数据的共享与安全。为了保证数据库中的数据起到恰如其分的作用,数据库管理员对各种用户有分别授予不同的使用数据的权限,这样既能使用户能够使用足够的数据处理业务,同时还能保证用户不能接触其它保密数据,或保证用户不能修改不在他权限范围内的数据。

4) 磁盘空间的管理。至于数据库系统的初始化、启动与停车等日常管理工作是由操作系统管理员承担的。

数据库管理员的职能要由相应的软件支持,支持数据库管理员职能的软件包括在数据库管理系统中。不同的数据库管理系统,其数据库管理员的职责也有所差别。在一些简单的数据库系统中,甚至没有软件支持数据库管理员职能,如流传很广的 dBASE 数据库管理系统就是这样。在不具备数据库管理员职责的数据库系统中,数据的严格保密、安全等功能也不存在,故难于在一些复杂的

组织机构中采用。

数据库管理员的设置有两种不同方式,一种是专职数据库管理员,他们以完全集中的方式管理数据。在这种方式下,数据的组织管理简单、安全性能好,缺点是数据管理缺乏灵活性。另一种方式是授权方式使某些用户具有一定的数据库管理员的功能,这种管理方式增加了使用数据的灵活性,但也为组织管理数据带来复杂性。处理不好的话,还有可能影响数据的安全。

6. 数据库开发工具

用高级语言开发数据库的应用程序是件复杂、繁琐、费力,而且周期长的工作,这类应用程序维护也不容易。越来越多的数据库系统致力于提供开发应用程序的工具,帮助数据库设计人员在短时间内开发出各种处理业务的应用程序。可以这样讲,在今后相当长时期中,一个数据库系统是否能大量推广使用,除了能否适应较多的计算机软硬件环境外,基本上就要取决于是否有许多为用户广泛接受的开发工具。

开发应用程序的工具有一个更为广泛,但概念尚未清晰的术语——第四代语言。众所周知,对于 FORTRAN、ALGOL、PASCAL 等第三代语言,因为大量采用了数学符号与表达方式,比起第一代机器语言与第二代汇编语言,比较接近于数学语言,较易读写。但从处理的内容与范围看,每一语句仍然处理比较窄小的事务,算法和结构中一切细节均要编写程序人员全面考虑与安排。第四代语言则企图突破这种传统编写程序的格局。一般来说,它们可能具有下列特色中的一种或若干种:

1) 宏语句编程

有一些开发工具仍然需要设计人员编写程序,但在这些程序中的每一句语句,其执行的功能将相当于第三代语言中数句、数十句甚至数百句语句,这些宏语句正式执行前,往往须经预编译转换成计算机能接受的语句。

2) 命令式语言

命令采用十分接近于普通英语的形式(当然还需要符合严格的语法),通过执行一串命令,对数据进行各种处理,处理事务的内容与范围都比较宽广。

3) 屏幕格式

代替第三代语言中一句接一句的序列式编写程序,屏幕格式采用在直观图形设计表格的方式,由相应的开发工具直接生成应用程序。使用这种方式不仅方便了应用程序的开发,常常在运行这些应用程序时,在屏幕上提供的各种图示与说明也大大方便操作员输入、处理与输出数据。

4) 菜单选择

利用菜单提示与选择可以将设计人员开发的各种应用程序联接而成一个完整、统一的系统,便于各种操作员用统一的方式调用各种应用程序。近年来第四代语言得到很大发展,新概念与新方法层出不穷,而且这种语言也不限于在数据库系统中使用。

使用开发工具开发出来的应用程序常常要占用大量计算机存贮空间,以及耗费较多的运行时间,但能节省大量开发应用程序的时间而得到补偿。随着计算机硬件性能的提高,大量使用开发工具来生成应用程序的方式也越来越变得现实了。

§ 1.2 ORACLE 关系数据库系统

ORACLE 是美国 ORACLE 公司研制的关系数据库系统软件,1979 年推出的 ORACLE 第二版是世界上首批作为商品的关系数据库系统之一,采用美国国家标准的 SQL 语言作为使用数据库的基本语言。1986 年该公司推出的 SQL * Star,由 ORACLE RDBMS V5. 1 版与 SQL * Net、SQL * Connect

构成,是一个具有分布式处理功能的关系数据库系统,可以使不同机型计算机、各种操作系统、多种网络和若干种数据库管理系统下的数据库统一在一个计算机信息处理与管理系统之中。ORACLE V5.1 版已被汉化,目前已能在 Micro VAX、VAX 小型机和 IBM 微型机上使用。

1. 使用 ORACLE 关系数据库系统的计算机环境

ORACLE 数据库管理系统本身是采用 C 语言写成的,这就使得系统具有超等的移植性。它可以在各种大型、小型、微型计算机的多种操作系统下工作,表 1.2-1 是迄今为止可以使用 ORACLE 的机型和操作系统,而且不难将它移植到其它机型和其它操作系统下。

表 1.2-1 可用 ORACLE 数据库的计算机环境表

计算机型	计算机厂家	型 号	操作系统
大 型	IBM	370,43xx,30xx	VM/CMS,MVS,UTS
大 型	AMDAHL	470,580	VM/CMS,MVS,UTS
大 型	SPERRY	1100	UNIX
小 型	DEC	VAX-11,3xxx,8xxx	VMS,UNIX,ULTRIX
小 型	DEC	PDP-11	RSX-11M+
小 型	DG	MV 系列	AOS/VS,DG/UX
小 型	HP	9000	HP/UX
小 型	AT&T	3B 系列	UNIX
小 型	APOLLO	Domain	AEGIS
小 型	HARRIS	700,800,1000	VOS
小 型	Honeywell	DPS-6,7,8	GCOS
小 型	PRIME	50 系列	PRIMOS
小 型	STRATUS	32	VOS
小 型	SPERRY	5000,7000	UNIX
小 型	WANG	VS 系列	VS
小 型	SUN	(所有型号)	UNIX
微 型	IBM	PC/XT,PC/AT,PC/RT	DOS,XENIX,AIX
微 型	AT&T	6300,PC7300,3B2	DOS,UNIX
微 型	CT	Mini,Megaframe	UNIX
微 型	DEC	Rainbow,Micro VAX	DOS,VMS
微 型	MOTOROLA	6300,6600,2000	UNIX
微 型	NCR	Tower	UNIX
微 型	TI	Professional	DOS
微 型	WANG	PC	DOS

2. 数据模型与数据库

ORACLE 采用关系型的数据模型。ORACLE 数据库系统中,数据按二维表的结构进行组织,每张二维表称为一个关系(Relation),本书将采用通俗的表代替。有关表的术语介绍如下:

表(Table) 表由若干列和若干行组成,每张表有一表名。在数据库中,表名必须是唯一的,即不允许有两张或两张以上的表有同一名称。取名规则见 § 1.4 节。

列(Column) 表中每列包含同类的数据,有时列也称属性。在一张表中各列有一定排列顺序,但这种列的顺序并不重要。在一张表内的各列名不能重复,但不同表中可有相同名字的列。列的命名规则见 § 1.4。

行(Row) 行用来对某个客体进行描述,它由若干分属各列的数据组成。行没有名称,行与行之间没有固定的排列顺序。

字段(Field) 表中行与列交叉处的数据称为字段,字段常借用相应列的名字。

值域(Domain) 值域是数据的基本类型,表中每列都以某个值域为基础,从中取得数值。

主键(Key) 若用一列中字段值或若干列中字段值集合对行进行标识,则这个列或列的集合

称为主键。

一个 ORACLE 数据库中由全体用户建立的所有表综合而成为概念模型,每个应用程序中用到的表及以后要讨论到的视图(View)则构成与此应用程序相应的外部模型。通过 ORACLE 数据库管理系统软件,将概念模型转化为对各种数据及其数据间联系在磁盘存贮器中的位置与具体联系,形成数据的内部模型。本书将对数据的外部模型与概念模型进行详尽讨论,但对数据在磁盘中存放情况,即内部模型,不作深入研究。

由上述概念模型、外部模型和内部模型与根据这些模型提供的结构存放在磁盘存贮器上和全部数据构成了一个数据库。在 ORACLE 数据库系统中,只能包含一个联机数据库。

3. 数据库管理系统

ORACLE RDBMS 是 ORACLE 数据库系统的核心,它由数据库管理核心模块和其它帮助用户和数据库管理员维护、监视和使用数据库的一些实用程序组成,现作简单介绍。

数据库管理核心模块 核心模块主要包括 ORACLE 核心代码、系统全局区 SGA 和 BIW、BWR、ARH、CLN 等四个后台进程,用于控制管理数据库中数据的存取。

IOR 用于对 ORACLE 系统的初始化、启动和停车。

SGI 用于估算 ORACLE 所使用的系统全局区 SGA。

ODS 用于监视用户进程和 ORACLE 系统进程。

AIJ 当介质发生故障时,若破坏了数据库的一致性,那么等故障排除后,AIJ 实用程序可利用数据库备份以及 AIJ 日志进行向前恢复,使数据库恢复到一致的当前状态。

EXPORT 将 ORACLE 数据库中全部或部分数据移到某个备份文件中。

IMPORT 将用 EXPORT 移出的数据重新装入数据库。

ODL 将操作系统下的数据文件装入到 ORACLE 数据库中。

4. 用户与应用程序

本书绝大部分篇幅中都在讨论用户与应用程序,故此处从略。

5. 数据库管理员

ORACLE 数据库系统是一个比较典型的关系数据库系统,在此系统中必须设有数据库管理员。ORACLE 数据库管理员的基本职能如下:

- * 负责用户登记以及对用户授权
- * 磁盘空间的分配与管理
- * 对数据库定期进行备份
- * 系统出现故障时,负责恢复数据库
- * 监视数据库系统运行

还有一些数据库的管理工作,如 ORACLE 数据库系统的初始化、启动和停车,以及数据库系统运行日志,则由系统管理员负责。

ORACLE 数据库系统可以设立专职数据库管理员,也可以将数据库管理员的职权授予某些用户。此外,ORACLE 还将一部分数据管理权力下放给其它用户:

1) ORACLE 数据库系统中用户可将自己建立的表授权给别的用户使用。这种权力使数据管理具有灵活性,增加了各用户之间的联系,但也使数据管理变得复杂。处理不好,会影响数据的安全。

2) 用户可对自己建立表的数据进行备份。

3) 用户在使用数据库的过程中,可根据需要对数据库表进行各种级别的封锁。当用户使用数据库表中数据时,ORACLE 数据库系统会自动对有关的表或行进行封锁,防止数据或修改情况丢

失。但这还不能防止数据库系统死锁等其它情况,为此 ORACLE 数据库系统允许用户在使用数据库时,根据需要对有关的表或行设置各种不同的封锁。

6. ORACLE 开发工具

ORACLE 数据库系统提供大量开发工具供用户使用。

1) SQL * Plus

这是一种交互式的命令语言,由 SQL 命令和 Plus 命令组成。SQL 的原意是结构化英语查询语言,但实际上标准 SQL 语言已远远超出查询的范围。SQL 命令可用于建立表,存贮、修改和查询数据,以及控制其它用户使用数据库中数据的权限等。Plus 命令是对标准 SQL 语言的补充与增强,主要用于形成格式化的报表,对数据进行计算,设置数据库运行环境,命令编辑等。SQL * Plus 是 ORACLE 数据库系统的基本开发工具,它免去了用户与 ORACLE 数据库管理系统直接打交道之劳。同时,SQL * Plus 中各种命令,尤其是 SQL 命令,也是其它开发工具中经常使用的命令和语句。

2) SQL * Forms

由此种工具开发出的应用程序,利用屏幕上直观的格式(Form)形式,直接处理数据库中数据,包括将数据输入数据库,修改数据库中数据或删除数据库中数据等。这是一种快速开发应用程序的工具。SQL * Forms 中引入的触发子(Trigger)概念与机制,极大地增强了对各种数据校验及处理的功能,它是采用原型(Prototype)法开发应用系统的最好工具之一。

3) SQL * Report

这是一个生成报表的工具,它实质上是一个宏语言的预编译程序。它分成报表生成和报表格式化两部分。用户通过编写一个宏语言的程序,将用 SQL 语言从数据库中检索来的数据,与其它文字、标题等内容综合安排在一起,生成一张报表。然后使用定界、页号、多格式窗口、嵌套等功能,将具有格式化的报表在终端或打印机上输出。

4) SQL * Menu

它是帮助用户设计应用程序的驱动菜单。它可将一个或若干个用户开发的多种应用程序统一组织在一个应用系统中,通过操作员的选择,由菜单驱动相应的应用程序的运行。SQL * Menu 的功能包括用户菜单生成、菜单信息维护和应用系统的维护。

5) SQL * Calc

这是一个称为电子表格的软件。它开辟一个由行与列构成的相当巨大的表格,在屏幕上可显示这个表格中某些部分的窗口,组成表格的基本单元(Cell),可存放各种文字、字符、数字,甚至还可以安排各种 SQL 语句,这样就可以直接从 ORACLE 数据库中迅速取出数据。基本单元上的数据与 SQL 语句,可以复制、移动到别的单元中去,整个表格可以保存和编辑。这种工具使用户可以在继续保留手工制作表格的各种熟悉习惯的同时,又能利用数据库系统从重复抄写、数字计算、填写报表的繁重、枯燥的劳动中解放出来。

6) SQL * Graph

这是具有交互式图形编辑功能的电子图形软件。主要用于绘制各种统计图形,它对从数据库中检索出来的数据,生成彩色的直方图、折线图或圆饼图,并在终端的屏幕或绘图仪上输出。

7) SQL * Design Dictionary

这是一套应用系统开发与维护的设计词典。为应用系统的设计、开发、维护提供辅助和自动生成的文档,它对系统模型的规划、分析、设计、实施、运行和维护等各个阶段以菜单形式提供支持,是辅助系统分析、设计与开发人员从事各项工作的一种工具。

8) SQL * Net