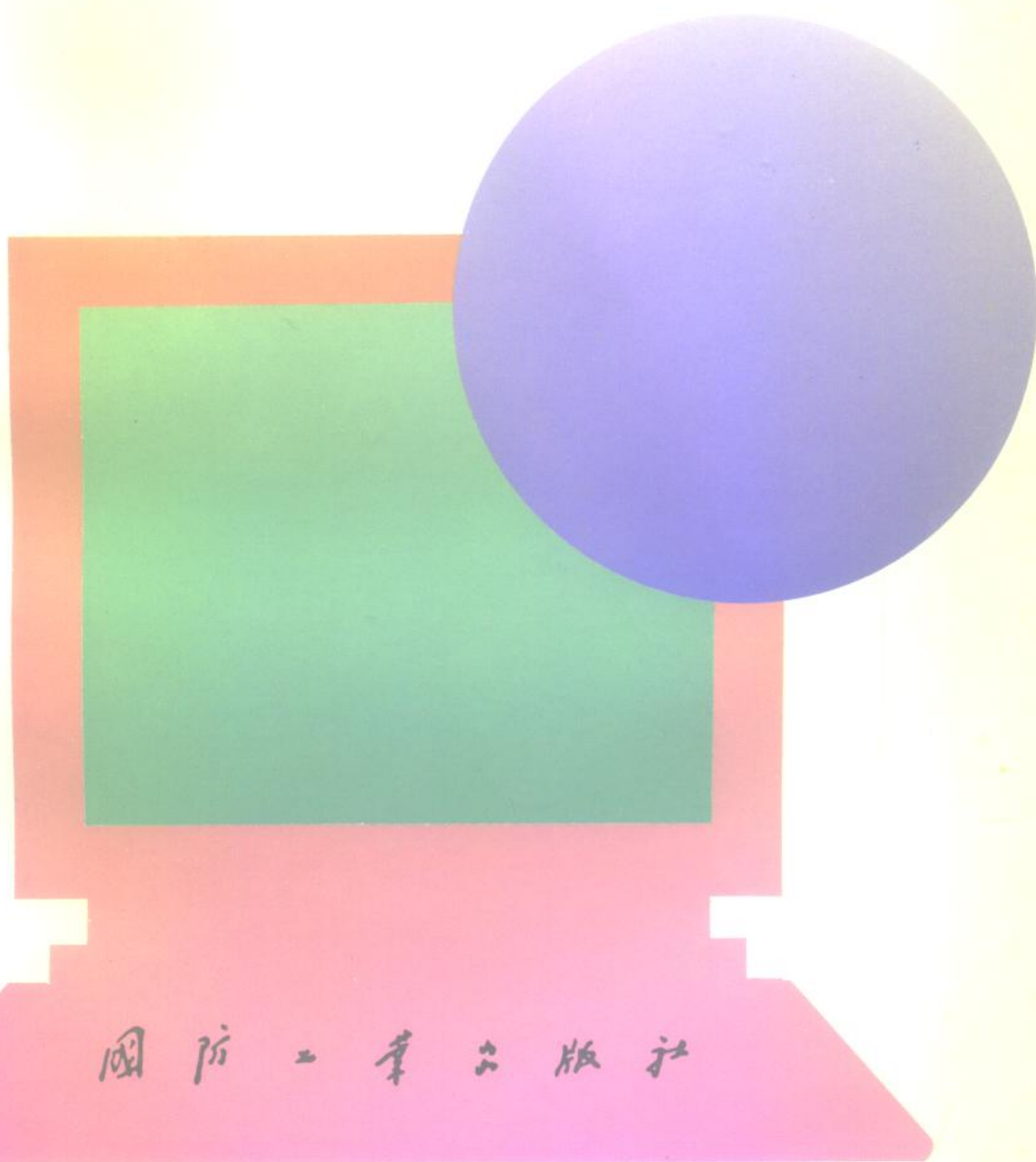


最流行软件丛书

谭浩强 主编

ORACLE 数据库技术

刘瑞挺 樊成 编著



国防工业出版社

ORACLE 数据库技术

刘瑞挺 樊 成 编著

国防工业出版社

(京)新登字 106 号

图书在版编目(CIP)数据

ORACLE 数据库技术/刘瑞挺,樊成编著. 北京:国防
工业出版社,1995.1
(最流行软件丛书)
ISBN 7-118-01313-7

I.O... I.①刘... ②樊... II. 关系型数据库-数据库
管理系统,ORACLE N.TP311.13

中国版本图书馆 CIP 数据核字(94)第 05710 号

国防工业出版社出版发行

(北京市海淀区紫竹院南路 23 号)

(邮政编码 100044)

北京怀柔新华印刷厂印刷

新华书店经售

开本 787×1092 1/16 印张 18 1/2 419 千字
1995 年 1 月第 1 版 1995 年 1 月北京第 1 次印刷

印数:1-4000 册 定价:19.50 元

(本书如有印装错误,我社负责调换)

丛书总序

电子计算机正以空前的速度发展,微型计算机更是其中的佼佼者,它几乎已深入到社会生活的一切领域。随着微型机的普及应用,众多的软件应运而生,其中有些软件因其功能丰富、实用性强、普及性好而流行于世。要使微型机发挥更大的作用,就必须掌握和熟悉这些软件的使用方法和技巧。为了适应广大初、中级计算机使用者的迫切需要,我们经过反复研究,特组织编写这套《最流行软件》丛书。我们企望尽此绵薄之力推动计算机在我国进一步普及应用。

本丛书采取“一种软件一本书”的模式,分别介绍国内广泛流行和经常使用的软件,力图突出其实用性强、普及面广、内容新颖、品种配套、概念清晰、通俗易懂等特点。

本丛书不同于计算机厂商销售的“使用手册”,也不同于一般教材。现在市面上有些译自国外资料的使用手册,虽然内容详实,但往往由于各种原因而难以阅读和理解,不适合于初、中级计算机使用者学习。考虑到多数读者的实际情况,我们采用循序渐进,深入浅出的编写方式,力求使那些从未接触过该软件的读者也可以做到“学了就能用,用了就见效”。限于篇幅不宜过大,每本书仅介绍该软件最基本、最常用功能的使用方法和技巧,不拟囊括其全部细节,也不列举较大规模的例题。一般也不详细介绍基本原理和名词概念,而以教会如何使用为目的。读者在掌握基本使用方法以后,可以通过实践更深入更巧妙地去使用有关软件。

考虑到国内微型机配置的现实情况,本丛书以 IBM PC 机及与其兼容的长城系列微型机上广泛使用的软件为主,兼顾其他。鉴于软件版本翻新很快,拟以当前广泛流行的版本为基础,并根据发展,不断更新。

本丛书的选题是根据我国软件应用发展状况和广大读者急需来确定的,特约高等院校和科研、设计单位有丰富实践经验的专家参加编撰,拟陆续分期分批奉献于世。“问渠哪得清如许,唯有源头活水来”。我们热切希望专家和读者能及时向我们提供有关信息,以使本丛书在选题、编撰、出版、发行等环节更具针对性和实时性。

本丛书无论在选题策划还是在编写细节上都可能会有不足甚至错误之处,恳切希望大家批评指正。谢谢!

丛书主编

前 言

美国 Oracle 公司于 1977 年成立。在瞄准刚刚起步的数据库市场时,公司做出了两个重要决策,并确信这两个决策能使 Oracle 公司获得成功。

第一个决策是对数据库模型的选择。当时层次模型和网状模型非常流行,但公司却选择了关系模型的 DBMS,这在当时的确有很大风险,然而事实证明这一决策是正确的。

第二个决策是对数据库语言的选择。当时有很多种语言可供选择,经过详细分析,公司选中了由 IBM 研究开发的 SQL 语言。今天,SQL 已被美国国家标准协会和国际标准化组织宣布为关系数据库管理系统的标准语言。这进一步证明了这些决策的远见卓识。

由于决策正确和产品对路,Oracle 成为世界上发展最快的软件公司之一。这个事实对目前我国星罗棋布的计算机公司来说,该是多么宝贵的经验啊!

无疑,基于 SQL 的关系数据库 ORACLE 是国际上最流行的软件之一。它可以运行在很多种大型主机、小型机、工作站以及微型机上,并且支持很多种操作系统。原则上,可以用 ORACLE 把不同的计算机、操作系统和网络连接起来。ORACLE 的优越性还体现在先进的体系结构上,这使它具有吞吐率大、生产率高、多用户事务处理、数据安全保密以及应付系统失败的能力。此外,它还提供了一系列应用开发工具,可以满足最常见的需求,从屏幕格式设计、报表生成、图形制作以及即席查询等。总之,可兼容性、可移植性、可连接性和多功能性构成了 ORACLE 的坚实基础。

目前,ORACLE 数据库在国内已有一定数量的用户。诚然,这一用户群比起 dBASE/FoxBASE 用户群要小很多。ORACLE 的用户大多是经费比较充裕的大中型企事业单位,它的昂贵价格使普通用户尚不敢问津。因此,在一般人心目中,ORACLE 仍是一座神秘的殿堂。

关于 ORACLE 的资料,主要是随机提供的鸿篇巨制的使用手册。翻译本或改编本多面向专业人员,篇幅大,价钱贵,个人承受较难。这就使一般读者与 ORACLE 之间的鸿沟不易填平。编著本书的目的就是想用比较小的篇幅,对 ORACLE 的精华进行阐述,使读者能用较短的时间了解它的原理、操作和使用。

虽然国内引进的 ORACLE 版本始于 V4. X,但它们的内容已经陈旧。因此,本书主要以 V5. X 为对象进行叙述。对于 ORACLE V6 的若干新增加的功能,也在适当的地方做了补充。至于它的最新版本 ORACLE V7,限于资料不多,只能泛泛谈及它的特点而已。

为了节省篇幅,本书没有在软件开发方面多施笔墨,只是把我们在 ORACLE 上开发的 EMIS 当作一个实例夹杂其中。笔者认为,只要把它的 SQL * Plus、SQL * Forms 及 Pro * C 搞清楚,那么用它来开发一个应用系统就是轻而易举的事。同样,为了节省篇幅,我们没有作为附录列出洋洋大观的系统出错信息、ORACLE 保留字以及语句格式等。

本书第一、二、六、七章由刘瑞挺编写,第三、四、五章由樊成编写,并由刘瑞挺负责统稿。中国计算机学会教育专业委员会副主任、全国高校计算机基础教育研究会理事长、本

目 录

第一章 ORACLE 的特点

1.1 SQL 数据语言	1
1.1.1 IBM 的开创性工作	1
1.1.2 System R 实验系统	2
1.1.3 SQL 语言的基本概念	2
1.2 ORACLE 版本演变	6
1.2.1 ORACLE 的早期版本	7
1.2.2 ORACLE 的中期版本	7
1.2.3 ORACLE 的最新版本	8
1.3 ORACLE 组成及其特点	9
1.3.1 ORACLE 的组成部分	10
1.3.2 ORACLE 的基本特点	11
1.3.3 ORACLE 的安装与运行	15

第二章 ORACLE 的核心——RDBMS

2.1 ORACLE 的存储结构	23
2.1.1 ORACLE 存储的信息分类	23
2.1.2 ORACLE 存储的物理结构	24
2.1.3 ORACLE 存储的逻辑结构	27
2.2 ORACLE 的数据类型	30
2.2.1 CHAR 数据类型	30
2.2.2 NUMBER 数据类型	31
2.2.3 DATE 数据类型	31
2.2.4 LONG 数据类型	32
2.2.5 RAW 和 LONG RAW 数据类型	33
2.2.6 ROWID 及其它	33
2.3 ORACLE 的内核:管理模块	35
2.3.1 ORACLE 的系统环境	35
2.3.2 RDBMS 的基本概念	37
2.3.3 INIT.ORA 的系统参数	39
2.3.4 SGA 系统全局区与进程管理	44
2.3.5 ORACLE 的封锁机制	48
2.3.6 ORACLE 的事务处理	53
2.4 DBA 实用程序	54
2.4.1 DBA 的职责	54

2.4.2	IOR 实用程序	55
2.4.3	SGI 实用程序	56
2.4.4	ODS 实用程序	57
2.4.5	AIJ 实用程序	59
2.4.6	CRT 实用程序	60
2.5	用户实用程序	62
2.5.1	ODL 实用程序	62
2.5.2	EXP/IMP 实用程序	65

第三章 SQL 语言

3.1	表的建立	69
3.1.1	表的实例	69
3.1.2	表的定义	72
3.1.3	表结构的修改	73
3.1.4	数据的插入	75
3.2	数据查询	78
3.2.1	SELECT 语句及简单查询	78
3.2.2	排序	83
3.2.3	分组查询	84
3.2.4	连接查询	86
3.2.5	子查询	89
3.2.6	树结构信息的查询	92
3.2.7	具有集合运算的查询	97
3.3	数据更新	99
3.3.1	数据的插入	99
3.3.2	数据的修改	101
3.3.3	数据的删除	102
3.4	视图的建立及其应用	103
3.4.1	视图的建立	104
3.4.2	视图的查询	105
3.4.3	视图的更新	106
3.5	索引	106
3.5.1	索引的建立	107
3.5.2	索引的引用	108
3.5.3	索引的删除	108
3.5.4	索引的有效性检查	108
3.6	簇聚概念及其应用	108
3.6.1	簇的概念	108
3.6.2	簇的建立	109
3.6.3	簇的使用	112
3.6.4	簇的删除	112
3.7	函数及报表输出	113
3.7.1	函数	113
3.7.2	报表格式输出	115

3.8 数据库保护	128
3.8.1 数据库的安全性	129
3.8.2 安全管理方法	129
3.8.3 数据库的并发控制	132

第四章 应用系统开发工具——SQL * Forms

4.1 SQL * Forms 的基本概念	138
4.1.1 SQL * Forms 的功能	138
4.1.2 Form 的基本组成	139
4.1.3 Form 的存储形式	139
4.1.4 SQL * Forms 的软件结构	140
4.1.5 设计与运行 Form 的常用概念	141
4.2 Form 的开发过程	142
4.2.1 规划 Form	142
4.2.2 设计 Form	145
4.3 Form 的运行过程	149
4.3.1 启动 Form	149
4.3.2 运行 Form	150
4.3.3 Form 操作实例	152
4.4 Form 级的建立及定义	156
4.4.1 登录 SQL * Forms	156
4.4.2 建立和定义 Form	156
4.4.3 建立 Form 的实例	160
4.5 BLOCK 级的建立及定义	160
4.5.1 选择块窗口	160
4.5.2 建立块的标准方法	162
4.5.3 建立和定义块的一般方法	164
4.5.4 屏幕设计	167
4.5.5 块级设计举例	169
4.6 域级的建立及定义	170
4.6.1 域的类型	170
4.6.2 域的建立、修改及删除	171
4.6.3 域的定义	172
4.6.4 域级设计举例	179
4.7 触发子的设计	181
4.7.1 触发子的基本概念	181
4.7.2 触发子的定义	189
4.7.3 编写触发子的命令	193

第五章 PRO * C 应用程序设计

5.1 Pro * C 接口工具概述	203
5.1.1 Pro * C 预编译概念	203

5.1.2 Pro * C 应用程序的结构	205
5.1.3 Pro * C 应用程序的运行步骤	205
5.2 Pro * C 应用程序首部	206
5.2.1 说明节(DECLARE)	206
5.2.2 定义 SQL 通信区(SQLCA)	209
5.2.3 连接 ORACLE 系统	211
5.3 Pro * C 应用程序体	212
5.3.1 WHENEVER 语句	212
5.3.2 事务提交及撤销	213
5.3.3 数据定义及更新操作	213
5.4 Pro * C 的数据查询	218
5.4.1 查询的组成及数据转换	218
5.4.2 单行查询	219
5.4.3 多行查询	219
5.4.4 游标方法	219
5.5 动态定义语句	222
5.5.1 EXECUTE IMMEDIATE 语句	223
5.5.2 PREPARE 和 EXECUTE 语句	223
5.5.3 PREPARE 和 FETCH 语句	225
5.5.4 动态定义的描述符方法	227
5.6 预编译命令	230
5.6.1 预编译命令的一般形式	230
5.6.2 预编译命令的可选项	230

第六章 ORACLE 的网络功能

6.1 ORACLE 分布式体系结构	233
6.1.1 从 SQL * Star 谈起	233
6.1.2 ORACLE 的分布查询	234
6.1.3 分布式数据库的概念	236
6.2 SQL * Net 网络软件	238
6.2.1 客户-服务器结构	238
6.2.2 SQL * Net 支持的网络协议	243
6.2.3 SQL * Net 的驱动程序	248
6.2.4 SQL * Net 的相关文件	251
6.3 SQL * Connect 网关软件	253
6.3.1 SQL * Connect 的功能	253
6.3.2 SQL * Connect 的种类	253

第七章 ORACLE 的应用工具

7.1 应用工具概述	255
7.1.1 ORACLE 实用程序:SQL * Loader 和 SQL * DBA	255
7.1.2 ORACLE 应用工具:CASE * Dictionary	257

7.1.3 ORACLE 编程接口	260
7.1.4 ORACLE 终端用户工具: Easy * SQL...	261
7.2 SQL * Report	262
7.2.1 RPT 报表生成程序	262
7.2.2 RPF 报表正文格式化程序	266
7.2.3 SQL * Report 工作实例	268
7.3 SQL * Graph	271
7.3.1 直方图的绘制	272
7.3.2 折线图的绘制	274
7.3.3 圆饼图的绘制	275
7.4 SQL * Calc	277
7.4.1 SQL * Calc 的功能特点	277
7.4.2 SQL * Calc 的表格结构	278
7.4.3 SQL * Calc 与数据库的接口	280
参考文献	282

第一章 ORACLE 的特点

ORACLE 是基于 SQL 的著名关系型数据库系统,是世界上最流行的 DBMS 软件之一。它能在多种计算机硬件平台上运行,并能通过连网形成分布式数据库环境。它的强大功能受到广大数据库研究及开发人员的青睐。

本章将首先介绍 SQL 数据语言,然后叙述 ORACLE 的版本演变,最后归纳一下它的基本特点,以便使您对 ORACLE 数据库有个全面而清晰的了解。

1.1 SQL 数据语言

SQL 数据语言是 ORACLE 数据库的基础。它起源于 IBM 公司对数据库的早期研究,Oracle 公司采用了这种语言并对它做了扩充。SQL 已成为数据库系统的工业标准语言。

1.1.1 IBM 的开创性工作

1970 年 6 月,IBM 公司 San Jose 研究实验室的 Edgar F. Codd 发表了一篇奠基性的论文,题目是“大型共享数据库的关系模型”。他引进了关系代数;提出了关系模型;推导了关系演算;概括了关系规范,从而在计算机科学中开创了研究关系数据库理论与方法的新领域。

Codd 生于英国。在牛津大学数学系毕业后,转美国密执安大学攻读并获得计算机与通信科学的博士学位。他从 1949 年开始就参加了 IBM 公司的工作,搞过许多早期系统软件的研制。

1981 年 11 月在 ACM(美国计算机协会)洛杉矶年会上,由于他“对数据库管理系统的理论与实践作出奠基性和持续性的贡献”,荣获了计算机科学的最高荣誉——图灵奖。他发表的例行“图灵演讲”题目是:“关系数据库:提高生产率的实践基础”^[2]。

我们知道在数据库的发展史上,描述客观世界的实体及其相互联系曾采用过三种数据模型:即用树型结构描述的层次模型、用网络结构描述的网状模型、以及用表格结构描述的关系模型。与这些数据模型相对应的数据库就分别称为层次型数据库、网状型数据库以及关系型数据库。

在关系模型中,数据都安排到表格内。每个表都由水平的行(row)和垂直的列(column)组成,因此又称为二维表格(table)。每个表格就称为一个关系(relation),每个表的信息就用来描述客观世界中的一个事实。

不难理解二维表具有简单、明白、精确、灵活的优点。实际上,在日常生活和科学描述中,人们经常利用表格形式。例如:上面对三种数据模型的文字叙述就比较罗嗦。如果用

表列出就清楚得多,如表 1-1 所示。

表 1-1 数据模型

模 型	结 构	数据库实例
层次(hierarchical)	树结构(tree)	IBM-IMS
网状(network)	网结构(set)	Cullinane/ICL-IDMS
关系(relational)	表结构(table)	DB2, INGRES, ORACLE

又如,我们要比较一下微机上运行的关系数据库管理系统,列出一张二维表就会一目了然,如表 1-2 所示。

表 1-2 微机关系型数据库概况

软件名称	常用版本	操作系统	适用机型
dBASE II	1.0A, 1.10	DOS	XT, AT, 386 及其兼容机
dBASE II +	1.00	DOS	XT, AT, 386 及其兼容机, 网络
FoxBASE+	2.00, 2.10	DOS, XENIX	同上
dBASE IV	1.00	DOS	同上
INFORMIX	3.30	DOS, UNIX, XENIX	XT, AT, AT&T3B, Altos986
UNIFY	3.20	UNIX	AT, Altos, Dual83, Univers
INGRES		UNIX	Dual 68000, PDP-11
ORACLE	4.1.4, 5.1A/B	DOS, UNIX, XENIX, VMS	XT, AT, PS/2, 386, AT&T3B, Micro VAX- II

我们还可以列出诸如学生登记表、教师授课表、学生选课成绩表等,在 1.1.3 中介绍 SQL 时再作详细讨论。

1.1.2 System R 实验系统

System R 是 IBM San Jose 研究实验室开发的关系数据库实验系统。

这一项目共分 3 个阶段完成。第一阶段从 1974~1975 年,目的是建立原型,开发 SQL 用户接口并实现一个 SQL 子集。第二个阶段从 1976~1977 年,主要是设计与建造功能完善的 System R 多用户版本。第三阶段从 1978~1979 年,通过实际使用建立了许多试点单位,并把研究原型开发成产品。例如,SQL/Data System 就是与 IBM 主机的 DOS/VSE 操作系统一起提供的关系数据库管理系统。

这里,我们感兴趣的问题是第一阶段对用户接口 SQL 的研制,提供了早期版本即 1974 年的 SQUARE^[3]。它比较数学化,抽象难懂。1975 年发展为结构化英语的查询语言 SEQUEL (Structured English Query Language)^[4],它把公式口语化,明确易懂。1976 年又改进为 SEQUEL-2^[5],也就是目前的 SQL 语言。1986 年美国国家标准局(ANSI)确认 SQL 为数据库语言的工业标准。

1.1.3 SQL 语言的基本概念

SQL 是一种类似于英语的数据语言。它的最大特点是功能强而简单易学。但是,这种

简单易学却来源于晦涩难懂的关系代数表达式。

早期版本 SQUARE 的基本概念都是用抽象的数学形式来描述的。例如,它的核心概念——映射(mapping)就表示成如下的一般形式:

$$A_1 \cdots, A_n R B_1 \cdots, B_m (\theta_1 b_1, \cdots, \theta_m b_m)$$

对此你可能感到很陌生,这正是我们举这个例子的用意。它还可以改写成:

$$\pi_{A_1, \cdots, A_n}(\sigma_{B_1 \theta_1 b_1 \wedge \cdots \wedge B_m \theta_m b_m}(R))$$

仍然相当艰涩。其实,稍加解释还是能大致了解它的含义。这里的 R 表示一个关系名, $A_i (i=1, 2, \cdots, n)$ 和 $B_j (j=1, 2, \cdots, m)$ 都是 R 的属性表。 θ_i 表示算术比较运算符,例如: $=, \neq, <, \leq, >, \geq$, 如果是等号,则可以省略。 b_i 表示某个常数,通常它规定了选择条件。此外,在关系代数中, $\pi_{A_1, \cdots, A_n}(R)$ 表示从关系 R 中投影(project)出 A_i ; $\sigma_F(R)$ 表示从关系 R 中按条件 F 进行选择(select); $\wedge$ 则表示按与(and)逻辑进行连接(join)。

这样一来,前面两式的含义就逐渐明白了。原来是按照 $B_j \theta_j b_j$ 条件,从关系 R 中选择出 A_i 来。映射实质上就是对二维表先按水平方向求子集,即选择运算;再按垂直方向求子集,即投影运算。关系代数正是以选择、投影、连接等运算为基础的一个数学分支。因此,我们说 SQL 是基于映射的语言,这是了解 SQL 应该明白的一个基本概念。

在随后发展的 SEQUEL 语言中,前述抽象形式的映射演化成下面的英语子句:

```
SELECT    A1, A2, ..., An
FROM      R
WHERE     B1θ1b1 AND...AND Bmθmbm
```

这就比抽象的数学式好懂多了。SELECT 子句指出被选择属性表的列名;FROM 子句指出关系表的名称;WHERE 子句指出应该满足的条件。

在 SEQUEL 出现的同一年,即 1975 年 IBM 沃森研究中心的 M. M. Zloof 等为图形显示终端用户设计了一种举例查询语言 QBE(Query By Example)^[6],它是面向图形的语言,又称为域演算语言。

在 QBE 中,前述的查询语句直接写成如下形式(二维表 STUDENTS 如表 1-3 所示):

```
SELECT    SNAME
FROM      STUDENTS
WHERE     AGE<20 AND SEX=M
```

它的意思已经很通俗,就是从学生名单中找出年龄小于 20 岁的男生名字。这也正是今天在 SQL 中普遍使用的查询语言。

值得指出的是,SQL 并不仅仅是一个查询语言。作为完整的数据语言,它由 4 个基本部分组成,即 SQL 使用的命令可分为以下 4 类:

一、数据定义语言 DDL(Data Definition Language)

我们知道,关系数据库是由若干二维表格组成的。因此,用户在建立自己的数据库时,首先就要创建(create)自己的表格,把用户将来要用的数据库框架搭起来。

DDL 就是用来创建数据库中的各种对象,包括表、视图、索引、同义词、簇等的数据定义语言。例如,创建一个学生登记表,表名为 STUDENTS。包括的栏目有学号 S#, 姓名 SNAME, 性别 SEX, 年龄 AGE。创建方法如下:

```
SQL>CREATE TABLE STUDENTS
```

```
(S# number(6),
  SNAME char(8),
  SEX char(1),
  AGE number(2));
```

其中,SQL>是提示符,表示可以输入 SQL 命令。CREATE TABLE 就是创建表格的 SQL 命令。其后必须跟一个由用户决定的表名。下面括号内,分别定义各栏目的列名。每个列名的后面是它的值类型,值类型后面括号内的数字表示该列名的最大长度。例如:学号和年龄都是数字类型,学号用 6 位数,年龄用两位数。又如:姓名和性别都是字符类型,姓名用 8 个字符,而性别只用一个字符就可区别。好在这些长度均可根据实际需要由设计人员决定。此外,对标点符号也要注意用对。例如:每个列名定义完后用逗号分隔,整个表都定义完后用分号作为结束。对于大小写则一视同仁。

关于创建表、视图、别名、同义词,簇聚等的细节,在第三章 SQL * Plus 中将会进一步讨论。

二、数据操纵语言 DML(Data Manipulation Language)

用户的各种表格创建后,就可以对它们输入数据了。DML 就是用来插入(insert)、修改(update)、删除(delete)数据的 SQL 命令。

例如,要对上述学生情况表输入数据,必须准备好与该表栏目要求符合的学生记录。插入方法如下:

```
SQL>INSERT INTO STUDENTS
      VALUES(&S#,&SNAME,
             &SEX,&AGE);
```

其中,INSERT INTO 就是插入数据的 SQL 命令。其后必须跟一个表名。下面的 VALUES 称为值表,括号内是与列表对应的数据。每个参数前面都用 & 开头。运行时,SQL 将提示你为每个参数输入一个值。如果不用 & 开头,则每次只能输入一行数据。使用 & 后,对于多个学生记录的输入就方便了。

三、数据控制语言 DCL(Data Control Language)

为了对数据库进行监视和管理,必须对用户访问或增删数据库的特权加以限制。DCL 就是用来授予或收回访问数据库的某种特权,控制数据增删事务的发生时间及效果,从而能对数据库进行监控的 SQL 命令。

通常,只有数据库管理员 DBA(Data Base Administrator)才拥有全部特权,即最高的系统特权。例如,只有 DBA 才能定义一个新用户。授权方法如下:

```
SQL>GRANT CONNECT TO ZHANG
      IDENTIFIED BY KOULING;
```

结果,建立了一个姓张的新用户,对他的识别是通过口令字实现的。

授出的特权,在必要时也可以由 DBA 收回。撤消方法如下:

```
SQL>REVOKE CONNECT FROM ZHANG;
```

这样,该用户就无权再访问数据库了。

四、查询语言 QL(Query Language)

QL 就是对数据库中的数据按特定的组合、条件表达式或顺序进行检索的命令。前面

介绍过的 SELECT、FROM、WHERE 3 个子句组成的查询块就是最典型、最常用的查询语言。

今天,人们已经把 SQL 广义地理解为系统查询语言(System Query Language)。我们所以用 SQL 作为展开本书的序幕,就是因为它是构成 ORACLE 数据库的基础。

为了巩固 SQL 的基本概念,下面给出 3 个简单的表:表 1-3 是学生登记表;表 1-4 是教师授课表;表 1-5 是学生选课成绩表。表中已经给出栏目的名称。

前面我们已用实例介绍过表 1-3 的创建方法,以及表格创建后用插入命令把记录输入表内的方法。请你再用同样的方法,把表 1-4 和表 1-5 创建起来,并把数据输入进去。

表 1-3 学生登记表(S)

学号 SN	姓名 SNAME	性别 SEX	年龄 AGE
S1	ZHANG	M	20
S2	WANG	F	19
S3	LI	M	18
S4	ZHAO	F	21

表 1-5 学生选课成绩表(SC)

学号 SN	课程号 CN	成绩 GRADE
S1	C1	85
S1	C2	76
S1	C4	90
S2	C1	56
S3	C3	67
S3	C2	74
S3	C4	69
S4	C1	84
S4	C3	72

表 1-4 教师授课表(C)

课程号 CN	课程名 CNAME	教师 TEACHER	教室 ROOM
C1	AR 体系结构	LIU	206
C2	DB 数据库	FAN	307
C3	AI 人工智能	FAN	307
C4	SE 软件工程	LIU	206

下面我们从多种角度做些简单的查询练习。

〔例 1〕 找出年龄小于 20 岁的男生。

```
SQL>SELECT SN,SNAME
      FROM S
      WHERE AGE<20 AND SEX=M
```

〔例 2〕 按年龄升高顺序列出女生的姓名。

```
SQL>SELECT SN,SNAME,AGE
      FROM S
      WHERE SEX=F
      ORDER BY AGE ASC
```

其中,只有满足 WHERE 条件的才能被选择出来,并由 ORDER BY 子句进行排序。ASC 表示从小到大的升序(ascending),DESC 则是与它相反的从大到小的降序(descending)。

〔例 3〕 找出樊老师所开的全部课程。

```
SQL>SELECT CN CNAME
      FROM C
      WHERE TEACHER=FAN
```

〔例 4〕 找出至少听刘老师讲授一门课程的学生姓名。

```
SQL>SELECT SNAME
```

```

FROM S
WHERE SN IN
    SELECT SN
    FROM SC
    WHERE CN IN
        SELECT CN
        FROM C
        WHERE TEACHER=LIU

```

这里采用了嵌套查询技术,如图1-1所示。其操作顺序是由内层(C表)开始,逐层向外(SC表、S表)执行。在SQL中,嵌套深度由系统限制。

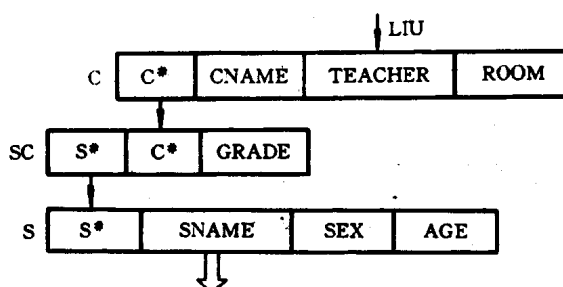


图1-1 嵌套查询

〔例5〕 从学生登记表中,删去课程不及格的学生。

```

SQL>DELETE S
    WHERE SN IN
        SELECT SN
        FROM SC
        WHERE GRADE<60

```

〔例6〕 把课程 C1的授课教师改为谭老师。

```

SQL>UPDATE C
    SET TEACHER=TAN
    WHERE CN=C1

```

其中,SET子句指出被更新的字段,WHERE子句指出被更新的记录行。

1.2 ORACLE 版本演变

1977年,Oracle 公司在美国旧金山东南的 Belmont 成立。经过十多年的发展,它已成为世界上最大的 RDB(关系数据库)厂家。据 IDC 统计,Oracle 一家便独占1990年 UNIX·RDB 国际市场的48.2%,这是第二名 Informix(20.8%)的两倍多。在1990年的世界计算机厂家百强排名中,Oracle 公司进入第51位,年营业收入已达10.6亿美元。那么 Oracle 公司成功的原因是什么呢?

1.2.1 ORACLE 的早期版本

Oracle 公司成立初期,即有 ORACLE 第1版面世,它曾是为大型计算机配套的数据库。

1979年推出用于 VAX 小型机的 ORACLE 第2版,这被认为是第一次实现了 SQL(比 IBM 的 SQL/DS 还早些)的第一个商品化的关系模型 DBMS。1980年在 ORACLE 第3版中,又确立了可移植的 RDBMS 模式。

今天,这些早期版本已很少使用。但它们正好体现了 Oracle 公司在刚刚起步时就做出的两个主要决策。

第一个决策就是选择了关系模型 DBMS,而在当时,层次数据库和网状数据库却十分流行。尽管在当时看来这一决策有很大风险,但现在的事实已经证明这一决策是正确的。

第二个决策则是对数据语言的选择。在对很多种语言进行了分析后,最后选中了由 IBM 开发的 SQL 语言。现在,SQL 不仅已是美国国家标准,而且 ISO 也宣布它为关系数据库管理系统的国际标准语言。这更进一步证明 Oracle 公司选择 SQL 的远见卓识。早在其它公司还没有在 SQL 方面展开竞争时,Oracle 就捷足先登了,率先实现了 SQL。因此,Oracle 公司在 SQL 方面的经验确实是相当丰富的。

这两个正确的技术决策正是 Oracle 公司获得成功的重要因素。

1.2.2 ORACLE 的中期版本

1984年 Oracle 公司又推出在微型机上运行的 SQL-RDBMS,这就顺应了微机普及的形势,使 ORACLE 第4版能从小型机市场又切入微型机市场。最早传入我国的 ORACLE 产品也正是 V4版本。例如:大家熟悉的 V4.1.4版。

1986年是 Oracle 公司大丰收的一年。它相继推出 SQL * Star、SQL * Net、SQL * Calc、SQL * Graph、SQL * Connect Gateway Toolkit 等产品,形成了丰富多彩的 ORACLE 第5版。

顺便指出,V5产品开始采用前缀与三位号码来完全标识的命名方法。

采用的前缀共分3个级别:Easy 前缀为初级用户级,表示全屏幕产品。SQL 前缀为数据库管理人员级,表示交互命令驱动产品。Pro 前缀为编程开发人员级,表示应用编程接口产品。

采用的标识号码也分3个层次:版本号、发行号、级别号。例如:V5.1.12,其中5为版本号,只有增加了重要功能才发布一个新的版本。1为发行号,又称维修发行号,每当找出毛病并纠正了错误或为现有程序增加了新特性时,维修发行号才增1。后面的12为级别号,它指明目标代码的特定级别,又称代码修订级别号,它主要由公司自己使用。

还应说明的是,各个产品的版本标识是独立排列而彼此无关的。例如,你使用的 ORACLE 可能由以下产品组成:RDBMS V5.1.12,SQL * Plus V1.0.9,SQL * Forms V2.0.3,以及 Pro * FORTRAN V1.1.2等。

在 V4、V5发展的基础上,ORACLE 形成3大系列:

一、单用户 ORACLE

1. Professional ORACLE

它是用于 PC/AT 和 MS-DOS 及其兼容机上的单用户版本,又称专用 ORACLE。