

本书是创建 Oracle 数据库和应用的最完整的参考书。书中循序渐进地讲述了关系数据库的概念及理论，并结合实例引导读者从开始的设计到完成最后的集成，最终创建一个完整的 Oracle 数据库系统。书中大多数资料对初学者创建 Oracle 数据库是必读的。另外，本书还讨论了 Oracle8 最新的特性。

通过阅读本书，读者可以学到如何进行 PL/SQL 编程、如何完善数据库设计以及如何在 Intranet 或 Web 服务器上应用设计完善的数据库。总之，本书拥有了创建一个完整 Oracle 系统的全部资料。本书适合 Oracle 初学者，有经验的 Oracle 技术人员及大专院校师生阅读。

John Adolph Palinski: Oracle Database Construction Kit

Authorized translation from the English language edition published by Que Corporation.

Copyright© 1997 by Que Corporation.

All rights reserved, For sale in Mainland China only.

本书中文简体字版由机械工业出版社和美国西蒙与舒斯特国际出版公司合作出版，未经出版者书面许可，本书的任何部分不得以任何方式复制或抄袭。

本书封底贴有 Prentice Hall 防伪标签，无标签者不得销售。

版权所有，翻印必究。

本书版权登记号：图字：01-98-0638

图书在版编目 (CIP) 数据

Oracle8 数据库构造工具实用指南 / (美) 法兰斯基 (Palinski, J.A.) 著; 刘丽等译.
- 北京: 机械工业出版社, 1998.6

(Oracle 技术系列丛书)

书名原文: Oracle Database Construction Kit

ISBN 7-111-06347-3

I .O... II .①法... ②刘... III .关系数据库, Oracle-软件工具 IV .TP311.13

中国版本图书馆 CIP 数据核字 (98) 第 13415 号

出 版 人: 马九荣 (北京市百万庄大街 22 号 邮政编码 100037)

责任编辑: 傅豫波 廉一兵

北京第二外国语学院印刷厂印刷·新华书店北京发行所发行

1998 年 6 月第 1 版第 1 次印刷

787mm×1092mm 1/16·31.25 印张

印数: 0001—7000 册

定价: 77.00 元 (附光盘)

凡购本书，如有倒页、脱页、缺页，由本社发行部调换

目 录

贺 辞
序
译者序
引言

第一部分 数据库对象的设计

第 1 章 Oracle 工具介绍	1
1.1 Oracle 公司介绍	1
1.2 Oracle7——Oracle 公司的数据库产品	1
1.2.1 Oracle8——下一代数据库	2
1.2.2 SQL * PLUS——Oracle 的数据库语言	3
1.2.3 其他 Oracle 产品	3
1.2.4 Designer 2000	3
1.2.5 Personal Oracle7	3
1.3 本书中的习惯用法以及技术术语	3
1.4 小结	3
第 2 章 数据库设计	5
2.1 什么是关系型数据库	5
2.2 数据库组件	7
2.2.1 表空间	8
2.2.2 表和字段	8
2.2.3 索引	9
2.2.4 视图	10
2.2.5 同义词	11
2.2.6 授权与角色 (Roles)	11
2.3 关系数据库的设计	11
2.3.1 数据库术语	12
2.3.2 数据库设计的描述	13
2.3.3 数据库表	14
2.3.4 数据的规范化 (Normalizing)	15
2.4 小结	18
2.5 我们还将学到	18
2.6 习题	18
2.7 雇员数据库项目第一阶段	18
第 3 章 使用 Select 命令获取数据	21

3.1 登录到 SQL * PLUS	21
3.2 SQL * PLUS 编辑器	22
3.3 Select 命令格式详解	24
3.3.1 Select 子句	24
3.3.2 From 子句	25
3.3.3 Where 子句	25
3.3.4 Group by 子句	25
3.3.5 Having 子句	25
3.3.6 Order By 子句	25
3.3.7 简单的 Select 语句	25
3.4 记录排序	26
3.5 带条件的 Select 语句	27
3.5.1 等于符 (=)	28
3.5.2 不等符 (< > 或 !=)	29
3.5.3 大于 (>) 和小于 (<)	29
3.5.4 使用 IN 操作符	31
3.5.5 使用 Between 操作符	31
3.5.6 使用 LIKE 操作符	31
3.5.7 使用 NOT 操作符	32
3.5.8 理解多重条件	33
3.5.9 变量的使用	34
3.5.10 字段的算术运算和连接字符串	35
3.6 理解 NULL 值和表达式	37
3.7 日期运算	37
3.8 字符串函数	39
3.8.1 Length 函数	39
3.8.2 Lpad 函数	40
3.8.3 Ltrim 函数	40
3.8.4 Rpad 函数	41
3.8.5 Rtrim 函数	41
3.8.6 Substr 函数	42
3.9 Decode 函数	42
3.10 数值函数的应用	43
3.10.1 NVL 函数	44
3.10.2 Round 函数	44
3.10.3 Trunc 函数	44
3.11 日期函数	45

3.11.1 Add_Months 函数	45	5.6.1 Union (并集)	78
3.11.2 Months_Between 函数	46	5.6.2 Minus (差集)	79
3.11.3 Next_Day 函数	46	5.6.3 Intersect (交集)	80
3.12 日期描述符	47	5.7 用 Oracle8 组合数据	81
3.13 聚组函数	49	5.8 小结	81
3.13.1 Average 函数	49	5.9 我们还将学到	82
3.13.2 Count 函数	50	5.10 习题	82
3.13.3 多重聚组函数	50	第 6 章 数据库对象的定义	83
3.14 理解 Group By 函数	51	6.1 定义表项	83
3.15 理解 Having 子句	52	6.2 表的创建、更改和删除	85
3.16 理解子查询或子选择	52	6.3 表的修改	86
3.17 理解伪字段, dual 表, Describe 命令 和 Tab Table	54	6.4 定义表的约束条件	87
3.18 小结	55	6.4.1 Check 约束	88
3.19 我们还将学到	56	6.4.2 Default (默认) 选项	88
3.20 习题	56	6.4.3 Foreign Key (外部关键字) 约束	88
		6.4.4 Not Null (非空) 约束	90
		6.4.5 Unique (唯一性) 约束	90
		6.4.6 Primary Key (主关键字) 约束	91
		6.4.7 约束的禁止 (disable)	92
		6.5 索引的定义	92
		6.6 视图的创建与删除	93
		6.7 创建同义词 (synonyms)	95
		6.8 创建数据库链接	96
		6.9 创建用户帐号	97
		6.10 创建数据库权限	97
		6.11 对用户表授予存取权	100
		6.12 创建表空间	101
		6.13 定义一个序列	102
		6.14 定义数据库触发器	103
		6.15 Oracle8 数据库对象	104
		6.15.1 修改分区	105
		6.15.2 索引的分区	105
		6.15.3 创建类型	106
		6.15.4 使用 Varray	106
		6.16 小结	107
		6.17 我们还将学到	107
		6.18 习题	107
		6.19 雇员数据库项目的第二阶段	108
		第 7 章 使用 DML 命令对表进行修改	109
		7.1 向表中添加记录	109
		7.2 更新表中的字段	110
		7.3 删除表中的记录	113
第 2 部分 使用 SQL * PLUS 对数据库进行操作			
第 4 章 SQL * PLUS 中的报表设计	57		
4.1 创建报表标题	57		
4.2 创建一个报表的页脚 (footer)	59		
4.3 报表的列的设计	60		
4.4 数字的格式设计	62		
4.5 页暂停的定义	63		
4.6 数据运算的定义	66		
4.7 设定环境	67		
4.8 小结	69		
4.9 我们还将学到	69		
4.10 习题	69		
第 5 章 表及记录的连接	71		
5.1 在关系型数据库及对象数据库中进 行数据的组合	71		
5.2 字段的限定	73		
5.3 应用多个连接	74		
5.4 一对多关系中表连接的错误	75		
5.4.1 表连接所引起的计算偏差	75		
5.4.2 当数值仅存在于一个参加连接的 表时所发生的问题	76		
5.4.3 由数据格式不匹配而引起的性能 问题	76		
5.5 外部连接	77		
5.6 集合操作符	78		

7.4 保存修改结果	115
7.5 小结	116
7.6 我们还将学到	117
7.7 习题	117

第三部分 PL/SQL: Oracle 的编程语言

第 8 章 创建第一个 PL/SQL 程序	119
8.1 如何理解 PL/SQL	119
8.1.1 理解 PL/SQL 块	120
8.1.2 嵌套的块和块标号	121
8.1.3 定义 PL/SQL 变量	122
8.1.4 使用 %TYPE 和 %ROWTYPE	122
8.1.5 对变量进行赋值	123
8.2 使用 If-Then-Else 逻辑	124
8.2.1 使用嵌套的 if 语句	126
8.2.2 在 if 语句中使用 elsif	127
8.3 使用循环和退出	128
8.4 怎样理解 PL/SQL 游标 (cursors)	130
8.5 确定例外事件	132
8.6 记录并显示错误	133
8.6.1 重新定义标准错误事件的名称	135
8.6.2 用于用户自定义的例外事件的 Exception 部分	135
8.7 小结	137
8.8 我们还将学到	137
8.9 习题	137
第 9 章 增强 PL/SQL 块的功能	139
9.1 游标属性的使用	139
9.1.1 %found 属性的使用	139
9.1.2 %rowcount 属性的使用	141
9.1.3 %isopen 属性的使用	142
9.1.4 显式游标及隐式游标	142
9.2 For 循环的使用	145
9.2.1 创建及使用对数字的 For 循环	145
9.2.2 创建及使用对游标的 for 循环	147
9.3 For Update Of 的使用	150
9.4 Where Current Of 选项的使用	151
9.5 被命名过程的创建及使用	152
9.6 函数的创建及使用	154
9.7 包的创建及使用	156
9.7.1 包的结构	157
9.7.2 创建一个包的定义	158

9.7.3 包体部分的创建	159
9.7.4 从应用程序中调用包的对象	160
9.8 小结	162
9.9 我们还将学到	163
9.10 习题	163
第 10 章 SQL * LOADER 的使用	165
10.1 理解 SQL * LOADER	165
10.2 怎样启动 SQL * LOADER	166
10.3 各种加载文件的使用	169
10.3.1 日志文件的使用	169
10.3.2 坏记录文件的使用	172
10.3.3 废弃记录文件的使用	173
10.4 控制文件的定义	173
10.4.1 简单控制文件的定义	173
10.4.2 加载方式的定义	174
10.4.3 数据格式的定义	175
10.5 控制文件中逻辑条件的定义	176
10.6 使用同一加载过程加载多个表	177
10.7 其他控制文件的关键字和子句	178
10.8 Option 子句和 Recoverable/Unrecoverable 子句	179
10.8.1 Into Table 子句	179
10.8.2 Concatenation 子句和 Continueif 子句	181
10.9 从命令行中执行 SQL * LOADER	182
10.10 小结	183
10.11 我们还将学到	184
10.12 习题	184
10.13 雇员数据库项目的第三阶段	184

第四部分 Oracle 的用户界面 (Oracle Forms 4.5)

第 11 章 Oracle Forms 的使用	187
11.1 什么是 Form?	187
11.2 一个 Form 的例子	188
11.3 Form 的一些术语	189
11.4 Developer 2000 的 Oracle Forms 应用	189
11.5 在 Forms Runtime 中执行一个 Form	190
11.6 Form 操作模式	191
11.6.1 Form 的输入模式	191
11.6.2 Form 的查询模式	192
11.6.3 Form 的修改模式	193
11.6.4 在一个 Active Form 中使用功能	

键	193	13.3.4 图形形状工具	246
11.7 在一个 Active Form 中查询一个记录	194	13.3.5 项 (Item) 工具	247
11.8 其他 Form 过程	196	13.3.6 对象着色 (Object Coloring) 工具	249
11.8.1 置 Form 为输入模式	196	13.4 画板属性	251
11.8.2 修改结果的保存	197	13.4.1 打开画板属性页 (Sheet)	252
11.8.3 错误的显示	197	13.4.2 画板属性页 (Sheet)	252
11.9 小结	197	13.4.3 复查与设定画板属性	254
11.10 我们还将学到	199	13.5 定义窗口 (Window) 属性	255
11.11 习题	199	13.6 设计 Employee Form	257
第 12 章 利用对象导航器创建第一个 Oracle Form	200	13.7 小结	260
12.1 进入 Forms Designer	200	13.8 我们还将学到	261
12.2 对象导航器的使用	201	13.9 习题	261
12.2.1 对象的扩展与折叠	202	第 14 章 创建及修改主 - 从 (Master - Detail) Form	262
12.2.2 在对象导航器中定义 Form 对象	202	14.1 将块添加到 Employee Update Form 中	262
12.2.3 对象导航器工具面板	203	14.1.1 创建 Tools 从块	262
12.3 Forms Designer 中的菜单选项	205	14.1.2 块所对应的对象及其设定	264
12.3.1 File 菜单项	205	14.1.3 格式化 Tools 块	265
12.3.2 Edit 菜单	209	14.1.4 创建 Glasses 块	265
12.3.3 Tools 菜单项	212	14.2 了解 Form 属性	267
12.3.4 Navigator 菜单	217	14.2.1 Coordinate Information 属性	269
12.4 创建第一个 Form	218	14.2.2 使用 Current Record Attribute (当前记录属性)	269
12.4.1 创建一个简单的单块 Form 的步骤	219	14.2.3 定义一个可视属性 (Attribute)	271
12.4.2 创建 Form	219	14.3 了解块属性	271
12.5 小结	224	14.3.1 使用 Order By 和 Where Clause 块属性	273
12.6 我们还将学到	224	14.3.2 改变 Navigation Style	273
12.7 习题	224	14.4 了解项属性 (Item Properties)	274
第 13 章 Form 的设计	226	14.4.1 Multi-Line 属性的使用	278
13.1 版面编辑器	226	14.4.2 关于一些实用的文本项属性的说明	278
13.2 版面编辑器的菜单条	227	14.5 了解项类型	280
13.2.1 View 下拉菜单	227	14.5.1 检查框 (check box)	281
13.2.2 Format (格式) 下拉菜单的使用	235	14.5.2 圆形按钮组 (Radio Group)	282
13.2.3 Arrange (安排) 下拉菜单的使用	240	14.5.3 列表 (List Item) 项	283
13.3 版面编辑器工具面板的使用	245	14.5.4 图像 (Image) 的使用	285
13.3.1 选择 (Select) 工具	245	14.6 创建 List of Values	287
13.3.2 旋转 (Rotate) 工具	245	14.6.1 LOV 的调用	291
13.3.3 整形 (Reshape) 工具	246	14.6.2 通过按钮显示 LOV	291

18.2.3	Format (格式) 菜单项	374
18.2.4	Arrange (安排) 菜单项	375
18.3	版面编辑器的工具	377
18.3.1	工具条	377
18.3.2	工具面板	377
18.4	版面对象属性	378
18.4.1	公共属性页	378
18.4.2	特殊属性页	381
18.5	创建美观的报表	382
18.5.1	更改不同对象的字体	383
18.5.2	更改列标题的文本	383
18.5.3	删除点状线以及线段变换	383
18.5.4	添加一个报表标题	383
18.5.5	添加页号和报表日期	384
18.5.6	更改域的位置	384
18.5.7	格式化域, 调整域尺寸	384
18.5.8	添加页首和结束页	384
18.6	从零开始创建一个报表版面	384
18.6.1	添加一个重复框架	385
18.6.2	添加域到版面	385
18.6.3	添加静态文本到版面	387
18.6.4	使框架更完美	387
18.7	添加公式列和汇总列	388
18.7.1	汇总列	388
18.7.2	公式列	389
18.8	小结	390
18.9	我们还将学到	390
18.10	习题	390

第六部分 Web 中的 Oracle 应用

第 19 章	将 Oracle 应用放到 Web 中 ...	391
19.1	Web 应用是怎样实现的	391
19.1.1	三级 Web 存取的描述	391
19.1.2	Web 应用的优点	392
19.2	所需软件及软件所安装的位置	392
19.2.1	前端所需软件	392
19.2.2	中间件或应用服务器的软件	393
19.2.3	后端或数据库服务器的软件	394
19.2.4	本章所用的软件配置	394
19.3	在 Web 上建立 Oracle Forms	394
19.3.1	插件与非插件实现	395
19.3.2	两种实现的通用设置	395

19.3.3	插件实现的要求	396
19.3.4	非插件的实现	401
19.3.5	从 Web 中运行一个 Form	402
19.4	在 Web 上创建 Oracle Reports	403
19.4.1	为 Web Reports 设置 Web Server	403
19.4.2	Reports 的运行	405
19.5	小结	406
19.6	我们还将学到	407

第七部分 完整的模板系统

第 20 章	开发一个雇员信息系统的 模板系统	409
20.1	第一阶段——数据库规范化和数据 库设计	409
20.2	第二阶段——创建数据库的对象	411
20.3	第三阶段——将数据加载到数据库 表中	415
20.4	第四阶段——设计及开发用户界面	421
20.4.1	Department Update Form (Deptupd. fmb)	422
20.4.2	Employee Update Form (Empupdl. fmb)	422
20.4.3	Employee Directory Form (Empdir. fmb)	426
20.4.4	Security Update Form (Empsec. fmb)	427
20.4.5	Employee 系统启动 Form (Eemptup. fmb) 和系统菜单 (Esysmenu. mmb)	428
20.5	第五阶段——开发并添加报表到该 系统中	429
20.6	小结	431

第八部分 附 录

附录 A	内置子程序	433
附录 B	触发器	443
附录 C	系统变量	449
附录 D	练习数据库安装指导	451
附录 E	习题答案	455

第一部分 数据库对象的设计

第 1 章 Oracle 工具介绍

本章向用户介绍 Oracle 公司以及它所开发的主要产品：Oracle7 数据库和 Developer 2000 工具集。它们是开发一个 Oracle 关系数据库的主要工具。

1.1 Oracle 公司介绍

Oracle 公司创建于 1980 年初，那时，它是只具有为数不多的几个软件产品的关系数据库公司。当时，关系数据库正处在婴儿时期。创建 Oracle 公司的先驱们预见到了计算机产业将转向关系数据库。在这个时期，IBM 公司通过它们的一个叫做 DB2 的非常优秀的关系数据库统治着业界。但遗憾的是该产品只能在 IBM 的硬件上运行。对 Oracle 公司来说，作为竞争者，就必须采取不同的策略。Oracle 提出一个开放系统的概念，即原则上 Oracle 数据库可以在任何计算机上使用。这对那些非 IBM 的硬件制造商以及软件开发商来说，是一个非常吸引人的选择。对那些不喜欢被束缚在 IBM 主机上的计算机客户来说，也同样是非常吸引人的。

今天，Oracle 是计算机界最大的公司之一。围绕着 Oracle7 数据库，它有着丰富的产品支持。最主要的是 Oracle7 数据库、Developer 2000 和 Designer 2000。Oracle 还对系统规划、系统开发以及系统实现提供咨询。

Oracle 公司的所在地是加利福尼亚州的雷德伍德。公司的业务遍及 90 多个国家。其产品在无数计算机上运行。在信息高速公路上，Oracle 也担当着一个主要的角色。

1.2 Oracle 7 ——Oracle 公司的数据库产品

Oracle7 是 Oracle 的关系数据库产品。该产品在各种计算机平台上运行，它是许多软件供应商的数据库选择。Oracle 支持各种配置。可用于集中式的结构。在这里，所有用户要连接到装有数据库的同一台主机上。这类结构的例子之一就是主机（Mainframe）应用。主机中装有数据库和软件。哑终端或用做仿真终端的 PC 机对主机进行存取。与该结构相对应，Oracle7 也支持客户/服务器（client/server）应用。这时，数据库装在服务器上，而应用程序装在不同的平台上。应用软件和数据库的存取是通过网络来实现的。Oracle7 还支持分布式处理的结构。这意味着数据库可分布到多个平台上去。例如，一部分数据库可位于洛杉矶，而另外一部分可以处在缅因州的波特兰。后两种结构在合理使用计算机以及分布式数据处理方面具有非常强的性能。

Oracle7 本身也有许多重要的特性。包括广泛的安全机制——敏感数据将通过特权的使用将其保护起来。也就是说，数据库管理员可以授予其他用户特权，使他们能够去查看，更

新或删除记录。管理员可以利用大量的特权组合，对数据库的存取加以控制。

数据库还有大量的备份以及恢复程序用来减少数据丢失的可能性。对数据库的每一个操作都要在档案文件中进行记载。所以，如果必要的话，Oracle 可以对数据库进行恢复。Oracle7 还提供了灵活的空间管理工具，使得数据库管理员可以控制数据库的数据的增长。最后，Oracle7 数据库还支持开放式的连接。这意味着用户可以通过该产品来存取其他厂家的数据库。Oracle7 可以存取象 DB2, Sybase, Microsoft Access 这样的数据库。此外象 Visual Basic 和其他大量的软件产品都可以使用 Oracle7。多个软件包可以存取同一个数据库是一个非常重要的特性。它使得客户不再担心被绑在某一个产品上。

Oracle7 有一些重要的组件，其中之一是 PL/SQL，或者称为 Oracle 的程序语言。PL/SQL 实现的功能与其他语言，如 C、FORTRAN、COBOL、BASIC 等语言类似。PL/SQL 是一个非常容易学习的且功能很强的语言。我经常用它将数据从一个系统转到另一个系统。

Oracle7 提供了开发存贮过程的工具。所谓存储过程指存在于数据库本身的 PL/SQL 程序或脚本。它实现一些特殊的功能，如对客户打印出一些字母，对其提供某个服务表示感谢等等，这些程序由发生在数据库中的操作来触发或初始化。

另一个组件是数据库触发器。它是存在数据库中的 PL/SQL 代码，当插入，删除或者修改数据库记录的事件发生时，这些 PL/SQL 代码就被执行。Oracle7 还允许用户将程序组合成包，这些程序也存放在数据库中。

数据库还具有分布功能，数据可以在多个地方存放。而数据的存放地点对用户是透明的。此外，数据库还有一个并行服务器功能，有些计算机具有 cluster 结构，这意味着每个计算机具有各自的内存，但外部存储设备是共同的。数据库的这项功能使 Oracle7 可以在该结构下运行。

1.2.1 Oracle 8 ——下一代数据库

Oracle8 是 Oracle 的下一代数据库。它最终将替代 Oracle7。这是一个关系型对象数据库 (object-relational)。这意味着数据库是一个具有面向对象特点的关系数据库。这类数据库允许开发者不用重写程序就可以将现存的应用进行移植。当条件成熟时，应用可以被转换来使用新的对象技术。这使得 Oracle8 向上兼容现存的数据库。这是 Oracle 提出的安全妥善的方法。从关系数据库这一主要支柱过快地移开会使 Oracle 失去市场份额。我相信人们都愿意平稳过渡，而不喜欢混乱。面向对象技术提供了未来的一些非常优秀的特性。我赞成 Oracle 公司逐渐向这个方向过渡的立场。

本书是一本主要讨论开发的书。它遵循已经使用了多年的传统的关系数据库设计。所使用的概念都源于 Oracle7 数据库，但在 Oracle8 中，这些概念同样适用。在本书中的关键部分，将论及 Oracle8 的一些新特性。这些特性中最重要的是新的数据类型。它使得数据库可以存放如图像，声音及空间数据等新的对象。Oracle8 还有支持数据库 warehouse 的能力。这些特性不是本书的重点。本书的目的是讨论如何开发用于一个业务部门的，普通的，处理日常事务的系统。这类系统是一个开发者要耗费大量精力和时间才能开发成功的。

对 Oracle8 的讨论将出现在本书的第一部分以及第二部分。这是本书讨论数据库设计的一部分。本书后面的部分将包括 Developer 2000 的内容。

1.2.2 SQL * PLUS——Oracle 的数据库语言

SQL * PLUS 使用 SQL 语言。SQL 语言即结构查询语言 (structure query language)，它是被所有数据库厂家所采取的一个工业标准。SQL 定义数据库的对象，如表、索引、视图和同义词。它也允许用户通过各种命令来对数据进行操纵。从 Oracle 中所学到的命令也可以适用于 DB2 或 Access 数据库。SQL 是遵循 ANSI 标准的。Oracle7 数据库只使用 SQL 语言。

1.2.3 其他 Oracle 产品

Developer 2000 是一套工具，用于开发、实现和维护有关 Oracle7 数据库的软件。它由 Oracle Forms 4.5 组成。它是一个可以使用户开发 Windows 界面、对数据进行操纵的功能很强的产品。这是 Oracle 的前端产品，并且它还提供了快速的应用开发。另一个重要的产品是 Oracle Reports 2.5。它是 Oracle 主要的报表设计产品。他可以设计很复杂的报表。这套工具还包括 Oracle Graphics，一个使用户可以开发显示模块的产品。一个模块可以是一个从数据库中计算出来的图表，也可以是与数据库无关的图形显示。Oracle Browser 是该套工具的另一个工具。它使用户可以开发简单的查询。对一个偶尔使用 Oracle，并希望通过使用一个工具进行一个简单查询，而不需要输出报表的用户来说，这是一个理想的工具。

在该套工具中，还包括 Oracle Book。这是一个对 Oracle 产品的所有共享文本进行在线文档浏览的工具。该套工具中还有其他产品：SQL * NET，Oracle Procedure Builder。

1.2.4 Designer 2000

Designer 2000 是一个 Oracle 工具产品。它由 Case Dictionary，Oracle Forms Generator，Oracle Reports Generator 和 Case Designer 组成。

1.2.5 Personal Oracle7

Personal Oracle7 是个人计算机的 Oracle 数据库。它可以在 DOS 和 Windows 下运行。用于 Oracle7 数据库的大多数概念都适用于该产品。

1.3 本书中的习惯用法以及技术术语

本书使用几种惯用法。书中可随时看到“提示”及“注意信息”。并且在它前面都标明其目的。提示的大部分源于作者以往的工作经验。

本书包括许多程序清单和例子。这些程序清单和例子的代码都可以在与本书配套的 CD 盘中找到。程序清单以字母 L 打头，后面是所在章的号码和清单在该章的号码。通过这种命名法可在 CD 中找到相应的文件。

各章用于实例的 Form 名可以在该章找到。每一章都有一个完整的 Form，包括了在该章中所进行的修改。这些 Form 在下一章讨论的技术中要使用到。

大多数章节后面都有练习，这些练习的答案可以在附录 E 和配套 CD 中找到。

1.4 小结

本书的目的是提供一个使用 Developer 2000 工具来开发一个 Oracle 数据库的指南和参考

手册。本书按照开发一个应用的步骤分为几部分，在每一个部分的最后，用户可以开发雇员数据库项目的几个部件。

本书的实例需要一个 Oracle 数据库的支持，附录 D 包含了这个练习数据库的安装指导。通过应用 Personel Oracle7 可以将数据库安装在用户的 PC 上，也可以安装在学校或公司的服务器上。不管数据库被安装在什么地方，都可以使用同样的指令和程序。

Oracle 具有一个很好的 Web 站点。地址为：www.Oracle.com，从该地址可以链接到其他网页。在写本书的时候，Oracle 推出了非常好的产品试用计划。用户可以免费下载一些产品进行试用。还可以以一个很低的价格得到产品的 CD。

除了 Oracle Web 站点之外，还有许多本地的 Oracle 用户组织，用于向用户介绍产品信息和解答用户问题。许多这样的组织在 Web 上都有自己的主页，可以通过它来与该组织联系。

我参加的一个组织是 Oracle Developer Tools User Group。这是与 Oracle Tools Development Group 关系非常密切的一个组织。他们给会员提供软件 Beta 版本的拷贝。他们还组织一些高级的研讨会。该组织的 Web 地址是：www.odtug.com/index/shtm。

像所有的软件以及硬件厂家一样，Oracle 不断地开发和发布着新的产品。为一个最新的产品写一本书是真正的挑战。一般来说，新发布的版本总与老版本看上去有些相似，相似的同时又有一些新的东西。在这本书中，我将试着提供 Oracle8 和 Developer 2000 2.0 版本的一些信息。

第2章 数据库设计

- 什么是关系型数据库 这一部分讨论了关系型数据库与层次型数据库的不同之处，同时也阐述了为什么产业界在众多数据库中选择了 Oracle7 这样的数据库的原因。

- 数据库组件 这一部分覆盖了各种各样的数据库组件，它解释了表空间、表、域或表项、视图、同义词、索引、授权和角色的概念。

- 关系型数据库的设计 这一部分揭示了数据库设计的一些基本原理，包括数据库方面的技术术语，数据库设计的描述和数据的规范化 (Normalizing)。

数据库是数据和数据的应用程序的集合，没有应用程序，就不能组织和存储数据，而没有数据，就没有使用应用程序的必要。通过用户能够在关系型数据库管理系统 (RDBMS) 下定义程序和触发器，Oracle7 真正实现了将应用程序和数据结合起来。对数据库的操作就是执行这些程序或触发器。

注意 第一次看到这种对数据库的定义时，我感到非常惊奇。在这以前我总是认为数据库应当包含数据和容纳数据的对象，我从来没有想过应用程序会成为数据库的一部分。在我想象中，应用程序仅仅是数据的各种组合，是与数据库本身无关的，但我喜欢这种定义，它表明了数据和应用程序的相互依存关系。

这一章讨论了在 Oracle7 或任何其他 RDBMS 内的各种数据库对象 (Object)。为了在 Oracle Forms 4.5 或其他产品中成功地设计数据库的各种应用程序，理解这种对象究竟是什么是非常重要的。数据的建立影响着应用程序工作的方式，也影响着应用程序最终对用户的支持。一个设计糟糕的数据库不会产生好的应用程序。

注意 Oracle Forms 4.5 是一个用来开发和执行 GUI (图形用户界面) 的产品。Form 可用来管理和查看数据。

如果读者具有设计关系型数据库较好的基础，可以跳过这一章。如果读者对数据库设计及其术语比较陌生，这一章会帮助读者理解实现一个应用程序的基础是什么，什么是关系型数据库，什么是数据库组件和怎样设计一张表。

雇员信息系统项目的第一阶段出现在本章的末尾。这一阶段介绍了项目需求，建立了数据库的表的模型。这一模型包括了将在项目的第二阶段建立的数据库对象的所有细节，我们将在本书第二部分的末尾看到项目的第二阶段。

2.1 什么是关系型数据库

关系型数据库是通过共同的值 (Common Value) 彼此相互关联的一系列表或记录体的集合。想象一下在一个文件柜中有两套索引卡片文件，第一个文件包含了每一个雇员的记录。记录中有每一个雇员不同于其他雇员的信息，包括工资单号码、住址、电话号码、部门号码和社会保险号码。第二个文件包含了每一个雇员采购工具的记录，包含了雇员的工资单号码和与工具采购相关的信息。两个文件包含了不同的信息，但却通过一个共同的值——工资单号码将两表关联起来见，图 2-1。

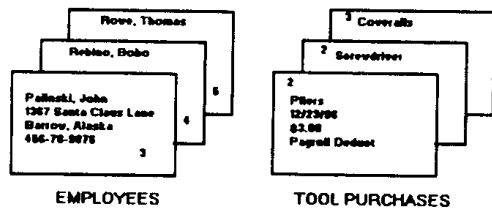


图 2-1 雇员卡片和工具采购卡片

图 2-1 显示了两彼此互相关联的卡片。工具采购记录是按照雇员号排序的，如果你想找到雇员 Palinski 的采购工具记录，你必须先到雇员文件中，找到 Palinski 的工资单号码，然后再在采购工具文件中查找包含此工资单号码的工具记录。如果在两个文件中没有共同的值——工资单号码，你就不可能将两个文件联系在一起使用。而这时要想达到利用雇员文件的某些信息来查寻其相应的工具信息的目的，工具采购文件就必须完全包含雇员文件的信息。例如，如果想通过雇员的姓从工具采购文件中提取记录，在工具采购文件的每个记录中，就必须有雇员文件中相应的信息。这样，在数据库中会造成数据冗余。相比之下，用两个文件一起来存储和提取数据会更有效。

如果数据是存放在关系型数据库管理系统（RDBMS）中，如 Oracle7，其数据库结构都是类似的。索引卡片的文件被称之为表。文件中的每一张卡片被称之为记录或行，而记录中的每一个数据值称为字段或数据项。图 2-1 所示的两张表通过存在于各表中的工资单号（payroll_number）字段相关联。

在关系型数据库之前，数据库开发者使用层次型数据库结构，他们是一个由段（Segment）组成的文件。每一个段相当于关系型数据库中的一个表，但在其各个层次的表中，并不包含共同值。父段通过使用一个指针来得知子段的位置。在图 2-2 中，父段包含了一个雇员的信息，子段包含了工具采购信息。由于在子段上并不包含与父段相同的字段，他们必须使用位置指针来确定记录的位置。父记录为每一个子记录存放一个指针，用来指向子记录的磁盘位置。

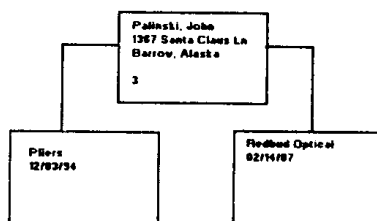


图 2-2 一个层次型数据库的结构

注意 在子段产生一个记录以前，相应的该记录在父段中必须存在。例如，一个雇员记录必须首先存在，然后才能建立该雇员的工具采购记录。

这种数据库结构在一般情况下可以工作得很好，但当需要在段中添加一个字段时，问题就产生了。这种数据库结构和软件是与位置相关的。在一个顺序文件中，程序员准确地知道每一个记录开始和结束的位置。如图 2-3 所示，第一个记录开始于位置 1，结束于位置 68；下一个记录开始于位置 69，结束于位置 136。每一个记录总是在前一个记录的 67 个位置或字节之后开始。这种定位被固定写在程序代码中。

当在文件当中添加一个字段时，添加的字段改变了第二个记录的开始位置，并且影响了

	ORIGINAL		MODIFIED	
	RECORD1	RECORD2	RECORD1	RECORD2
LAST_NAME	1-20	69-00	1-20	76-95
ADDRESS	21-39	89-107	21-39	96-114
CITY	40-50	100-110	40-50	115-125
STATE	51-52	119-120	51-52	126-127
HOME_PHONE	53-60	121-128	53-60	128-135
WORK_PHONE	61-68	129-136	61-68	136-143
NEW COLUMN (FAX #)				69-76

图 2-3 由于新的字段的增加引起的记录位置的变化

后续的所有记录。这时，应用程序就不再会显示正确的数据，因为它得不到正确的记录开始位置和记录结束位置。要解决这个问题，必须修改应用软件。这种修改是非常费力的，而且有风险，错误会由此导入应用程序。此外，你必须重新组织数据库文件中的数据。为了解决这个问题，开发者们在不同的地方附加上了段。这引起了数据库中的冗余数据的存在。正是由于这个原因，关系型数据库在行业界风行。关系型的表使得开发者可以任意添加一项而不用考虑对现存的应用程序的影响。因为关系型数据库不是面向位置的，没有一个现存软件需要重写。此外，也不需要重新组织数据库。这使得关系型数据库成为数据库的必然选择。

2.2 数据库组件

各种各样的组件构成了关系型数据库管理系统，Oracle7 就是这样。让我们首先从容纳数据的对象开始。数据库对象与存放手工记录的设备有某种相似性。在 Oracle7 上，所有的数据库对象必须有一个所有者。所有权是通过用户帐户及用户标识符 (ID) 来决定的。在任何数据库对象被建立之前，数据库管理员要建立一个用户帐号或用户标识符。因为数据库对象是与用户标识符紧密相连的。

注意 我在书桌的三个抽屉中存放了卡片。它很象一个数据库。注意，这个书桌是我的。因为我是拥有者，我可以把分类卡片、橡皮筋或任何我所选择的东西放进去，可以将我的书桌上的所有抽屉想象成数据库，抽屉将我的书桌上的卡片分开存放着。Oracle7 也有类似于我的书桌上的抽屉这样的对象。它们称之为表空间。在表空间内可以创建数据文件。数据文件非常类似于我抽屉里存放分类卡片的地方。

在最高水平的应用上，一个数据库由几个表空间 (tablespace) 组成。表空间为记录的存储保留了磁盘空间。每一个表空间包含了一个或多个数据文件，每个数据文件有固定的磁盘空间容量，表空间的大小等于所有数据文件大小的总和。虽然一个数据文件可能被数据和数据库对象占满，但可以通过往表空间中添加另外一个数据文件来扩充表空间的容量。

表空间中的主要对象之一是表，一个表相当于一系列的分类卡片的集合。每一张卡片相当于表中的一行或一条记录。也可以将表看作是一张数据表。一个数据表包括一行或多行，每一行又都由一列或多列数据组成。图 2-4 描述了表的行与列。

	A	B	C	D
1	Kennedy	John	1/20/61	
2	Johnson	Lyndon	11/22/63	
3	Eisenhower	Dwight	1/20/53	
4	Truman	Harry	4/13/45	
5	Roosevelt	Franklin	1/20/32	
6				
7				

图 2-4 一张表的行与列

这张表包括 5 条记录或 5 行，每一条记录包括 3 列。表的列也称之为字段或数据项。该表的列包含了我们已经熟悉的信息。列 A 包括了雇员的姓。行用来关联这些列的表项。一行中的所有数据属于表对象的一个实例。在第 4 行，每一列都包含着属于雇员 Harry Truman 的信息。在这个记录中，没有一个值是属于数据表中其他雇员的信息。

注意 放在我书桌抽屉里的卡片和存放在数据库中的记录的不同之处是数据库中的每一个字段都有一个唯一的名字。每一个字段都需要一个名字，是因为需要以某种方法告诉系统管理员要提取哪一条记录。而数据库管理员也需要以某种方法来确定所要提取的记录。

图 2-4 所说明的记录，是叫做 Employee 的表的一系列记录的一部分。第一列叫做 last_name，第二列叫做 first_name，第三列叫做 employment_date。这张表属于表空间 EMP。假如要想获得 Truman 的雇佣日期 (Employment date)，首先可以让 Oracle7 打开一个 Oracle 会话 (Session)，在 EMP 表空间中找到 Employee 表，获得第四项记录，并且显示 employment_date 列。

2.2.1 表空间

Oracle7 使用不同种类的表空间，在上述例子中用到的表空间叫做数据表空间。它包含了数据和用来管理数据的对象。另一种表空间称为系统表空间，Oracle7 利用系统表空间区域存放 Oracle7 实现其系统功能所需要的信息。

第三种表空间是临时表空间，Oracle 用它来存放临时信息。第四种表空间是工具表空间，Oracle7 在这块空间中存放与各种工具相关的数据库对象。这种工具由像 Oracle Forms 4.5 这样的产品组成。第五种表空间是回退 (Rollback) 表空间，当对数据库做了一些修改操作，在发出一个叫做“提交” (Commit) 的命令之前，数据库并没有真正修改。所有的修改操作被存放在临时表空间和回退表空间中。因此，Oracle 可以在数据库未真正改变之前，将数据库恢复成原来的状态。这可以保护系统不受错误修改的影响。此外，如果你正在使用一个程序去修改数据，而这个程序又被非正常地终止了，那么，所有由程序所造成的变化和影响被回退了。

提示 我在以下的两种情况出现时关心表空间。其一是当表空间快充满而我已不可能存储任何剩下的数据时，在这种情况下，我会在表空间中增加新的数据文件来解决这个问题。其二，也是最通常的情况，是发生在当我在程序中对一个大文件进行数据转换的时候，在这种情况下，临时表空间和回退表空间会变得越来越拥挤。这是因为程序在完成运行之前，数据不断地更新，又不能及时进行提交而造成的。

当这些表空间变满时，程序会非正常终止，所有已转换的数据将被丢失。为了避免这种情况，你可以将数据分成几小块。这种分割使你有可能及时提交更新的数据，释放表空间中所占用的空间。

2.2.2 表和字段

表和字段是最经常用到的数据库对象。表是一系列记录的集合，记录包括各种各样称之为字段的值。当创建一个表时，用户必须在表定义中定义字段的名称和字段的属性。

有三种类型的字段。第一种字段类型叫做字符型。字符型字段可以存放字符。即字符表中的字母，数字字符或特殊字符。第二种字段类型是数字型。数字型字段存放一个数字。如：整型数或小数。第三种字段类型是日期型。日期型字段存放日期，日期是一种特殊类型

的数据，它作为数字存放在数据库中，显示时却作为一串字符被显示出来。Oracle 的缺省日期显示的一个例子是“08-APR-51”。日期作为数字存放在表中的一个原因，是需要用日期型字段做一些数学运算。表 2-1 包括了一张在表定义中要用到的数据类型列表。

表 2-1 数据类型

数据类型	说明
Char (len)	定义一个固定的、总长度是 len 的字符型字段，一个 char 字段的最大长度是 255 个字符
varchar2 (len)	定义一个变长的、总长度是 len 的字符型字段，一个 varchar2 字段的最大长度可达 2000 个字符
number (len, prec)	定义一个数字型字段，长度为 len，精度为 prec。精度是指小数点后数字的个数。（用做货币时一般取精度为 2）
date	定义一个日期型字段。这个字段可以存放从公元前 4712 年 1 月 1 日到公元后 4712 年 12 月 31 日范围内的数据

字段的第二个属性是长度。每一个字段（日期字段除外）都包含了长度的说明。数字型字段长度说明包括两项：数字的最大长度和小数点后的位数或精度。

Char 类型是固定长度的字段。这意味着该字段完全使用了所定义的最大字节数。即使该字段没有被字符完全充满。空白字符（类似于按空格键所产生的空格）将自动占据没有字符的空位置。

Varchar2 数据类型是变长字段。这意味着该字段只使用与字段中实际字符数相等的字节数。在此类数据中，字段中不包含字符的位置将被 null 值充满，null 意味着空白或未知。数量相同的字符所代表的字段，例如：工资号码，模具号码，ZIP 号码等等，是用 Char 字段来存放的。Varchar2 字段用来存放地址或文本信息。

2.2.3 索引

数据库中表的索引相当于一本书中的索引。本书最后的索引包含了用来遍历全书的关键字。每一个词条都对应着一个页码，它指出所讨论的关键字所在的位置。索引帮助读者快速确定所需要的信息的位置。数据库中表的索引起着同样的作用。它是一个“小型表”，它包含了表数据的一个子集。它用来快速查找记录。

由于避免了全表空间的搜索，数据库索引加快了记录查询的速度。当数据库管理员要在一个不含检索的表查询记录时，他必须读取表中的每一条记录。这种对表从头到尾的搜索称为表空间搜索。由于从硬盘上读取记录是一个很慢的过程，开发者试图尽可能地避免这种搜索。索引是由一个或几个表字段的值和相应记录的位置所组成的小型表。由于数据库管理员只读取这种小型表，而不是表的全部，从而减少了进行整个表空间搜索的必要。

图 2-5 表明了读取磁盘的工作量上的差别。如果一个表 Customer Information 是未经过索引的，要想通过帐户号（Account Number）来查找一个记录，Oracle7 必须进行表空间的搜索。数据库管理员必须读 80M 字节才能获得所要求的记录。对帐户号字段的索引建立起来之后，数据库管理员就可以用索引来确定这个记录。这个索引仅有 2M 字节。数据库管理员所必须读取的数据量减少 40 倍。

索引还有其他有助于记录检索的特性。第一个特性是索引表的记录是经过排序的，这意味着当数据库管理员在找到所要找的记录后，就不必继续查找下面的索引表了。因为我们可以知道另外的具有同样值的记录不会存放在索引中的其他地方。这种排序的特性也允许数据

CUSTOMER INFORMATION TABLE
200,000 records * 400 bytes each = 80,000,000 bytes
INDEX ON THE ACCOUNT NUMBER FIELD ONLY
200,000 records * 10 bytes each = 2,000,000 bytes

图 2-5 比较 Customer Information 表和对帐户号字段的索引表的大小

库管理员去实现一些特殊的检索方法。从而避免了搜索整个索引。管理员可以首先查看在索引表中间位置的记录的值，然后与所要查询的记录进行比较，所需要的记录值有可能比该中间值大或小，这样管理员可以因此而排除索引表中的一半记录。下一个索引表记录的读取可以是在前半部分的中部（即 1/4 的索引表部分）或是在后半部分的中部（即 3/4 的索引表部分）。读取该记录后，数据库管理员要判断该记录的值是大于还是小于所要查找的值。通过不断重复这个过程，数据管理员可以在仅读取一小部分索引表记录之后就可以找到所需要的记录。

真正好的数据库要求每一张表中应当有一个这样的字段，它使得每一个记录的该字段值都是唯一的。索引提供给数据库管理系统维护这种唯一性的机制。Oracle7 使得开发者能够定义唯一性索引（Unique Index）。这一类索引不允许在一列中出现两个相同的值。在记录加入表之前，该记录的索引项的值要在索引表中进行比较，如果该值与现存的某个记录值相同，数据库管理系统就停止该事务，只有当两值不相同，事务才进行下去。

提示 在数据库中，索引是提高数据库性能最重要的对象之一，不管什么时候在查询、Form 或报表中出现性能上的问题，第一个要查看的应是现在使用的索引表。

2.2.4 视图

一个数据库视图类似于一个窗口，它使得用户可以看到或存取数据库的局部数据。实际上一个视图是一个预先定义的查询。执行时产生一个虚拟的、临时的记录集合或表。由于视图仅仅是定义了数据的集合形式，在未执行以前，视图对数据库不产生任何作用。

视图具有许多优点，视图使开发者可以将表的局部信息隐藏起来。只要不在视图的定义中定义这些要隐藏的字段，我们就可以作到这一点。视图也可以减少数据的复杂性。在关系型数据库中，数据很少存储在一张表中，一般总是放在几张相关联的表中。开发者将几个表的数据以某种复杂的逻辑关系集合起来，定义在一个视图中。这样你就可以在检索中象使用表一样地使用视图了。你不必在你的检索中使用这种逻辑关系，用户永远也不会知道他正在使用的是表还是视图（见图 2-6）。

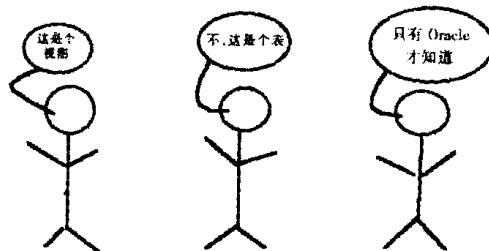


图 2-6 这是视图？

视图使开发者可以为字段定义一个用户字段名。这在数据库中开创了用户友好的例子。