



Oracle 技术系列丛书

ORACLE®

AUTHORIZED ORACLE PRESS™—EXCLUSIVELY FROM OSBORNE

Oracle9i

PL/SQL 程序设计

Oracle9i PL/SQL Programming

(美) Scott Urman 著 陈维军 王蕾 等译



OFFICIAL • AUTHORIZED

Oracle Press

ONLY FROM OSBORNE



机械工业出版社
China Machine Press



OSBORNE

Oracle技术系列丛书

Oracle9i PL/SQL程序设计

(美) Scott Urman 著

陈维军 王蕾 等译

万方工作室 审校



机械工业出版社
China Machine Press

本书介绍Oracle数据库不同版本中PL/SQL语言的主要特性,以及如何在不同开发环境中开发、测试和调试PL/SQL程序。主要内容包括PL/SQL的使用方法、错误处理、集合、子程序和包以及高级特性等详细信息。本书附带光盘包括书中范例代码和开发环境。

Scott Urman: Oracle9i PL/SQL Programming(ISBN: 0-07-219147-3).

Copyright © 2002 by The McGraw-Hill Companies, Inc.

Original language published by The McGraw-Hill Companies, Inc. All rights reserved. No part of this publication may be reproduced or distributed in any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

Simplified Chinese translation edition jointly published by McGraw-Hill Education(Asia) Co. and China Machine Press.

本书中文简体字版由美国麦格劳-希尔教育出版公司授权机械工业出版社出版,未经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。

本书封面贴有McGraw-Hill公司防伪标签,无标签者不得销售。

版权所有,侵权必究。

本书版权登记号:图字:01-2002-0802

图书在版编目(CIP)数据

Oracle9i PL/SQL程序设计/(美)厄曼(Urman, S.)著,陈维军等译.-北京:机械工业出版社,2002.4

(Oracle技术系列丛书)

书名原文:Oracle9i PL/SQL Programming

ISBN 7-111-09950-8

I. O… II. ①厄… ②陈… III. 关系数据库-数据库管理系统, Oracle9i-程序设计 IV. TP311.138

中国版本图书馆CIP数据核字(2002)第013244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:瞿静华

北京忠信诚胶印厂印刷·新华书店北京发行所发行

2002年4月第1版第1次印刷

787mm×1092mm 1/16·27.75印张

印数:0 001-4 000册

定价:59.00元(附光盘)

凡购本书,如有倒页、脱页、缺页,由本社发行部调换

译者序

本书是由Oracle畅销书作者所编写的，他多年来始终注重PL/SQL的研究工作，编写了多部有关PL/SQL的图书。本书是以前所编写图书的升级版本，包含了最新的Oracle版本和信息。本书不仅适用于PL/SQL的初学者，而且适合于Oracle开发人员参考；它不仅非常实用，而且是非常宝贵的参考资源！

我们很荣幸能够有机会承担本书的翻译工作。在翻译过程中，我们经常为一句话、一个术语进行反复的讨论，并查阅了大量的资料，力图使本书的翻译能正确、贴切地反映原文的意思，同时注意使句子、段落符合中国人的语言习惯。我们真挚地希望你能够从本书中有所收获，这是作者的初衷，也是我们的愿望！

本书由陈维军、王蕾组织翻译，万方工作室的全体同仁都参加了本书的翻译、校正和输入等工作。具体参加本书翻译、录排、校对工作的人员为：陈维军、王蕾、韩冰、宋红利、田蕴哲、丁天、牛志奇、刘堃、刘砚、黄建立、丁胜利、蒋雯丽、葛丽、罗贤锋、罗天浩、王洞宾、赵文凯、王荣、王慧英、黄慧萍、李明、徐艳华、李小萍、吴莹、文瑶等。本书的出版是集体劳动的结晶，在此特别感谢万方工作室的全体工作人员。

由于时间仓促，且译者经验和水平有限，译文难免有不妥之处，恳请读者批评指正！

万方工作室

2002年1月

1576/04

前 言

Oracle是一种功能极其强大，并且非常灵活的关系数据库系统。就是因为这种强大的功能和灵活性，致使Oracle非常复杂。为了实现基于Oracle设计有用的应用程序，则有必要理解Oracle是如何操作存储在系统中的数据的。PL/SQL就是这样一种实现数据操作的重要工具，你不仅可以在Oracle内部使用它，而且可以在自己的应用程序中应用它。PL/SQL在许多环境下都是可用的，并且在不同环境下具有不同的优势。

我所编写的第一本有关PL/SQL的图书是《Oracle PL/SQL Programming》，在1996年出版(英文版)。第一版主要涵盖了Oracle 7 Release 7.3中的PL/SQL 2.3版本——这是那时最新版本的Oracle数据库和PL/SQL。有关PL/SQL的第二本书是1997年出版(英文版的)《Oracle8 PL/SQL Programming》，该书在第一版本的基础上，进一步包含了Oracle8 Release 8.0的最新信息，当时Oracle8 Release 8.0是最新的Oracle版本。有关PL/SQL的第三本书是2000年出版(英文版的)《Oracle8i Advanced PL/SQL Programming》。第三本书注重强调高级特性，所涵盖版本最高为Oracle8i。

本书在以前版本的基础上继续扩展，包含了许多更新内容，集成了Oracle9i的特性。相对于以前的版本，本书增补了许多介绍性材料。同时，本书和《Oracle8i Advanced Programming》完全涵盖了PL/SQL语言。

本书包含的新内容

首先，这个版本的PL/SQL编程书与前几版本的主要的差别在于本书提供了Oracle9i的新特性。在本书中可以见到对这些新特性的讨论。比如，本书第3章在讨论基本PL/SQL语法的同时讨论了CASE语句（Oracle9i的一个新特性）。至于PL/SQL开发环境，本书包含了以前版本中提供的所有开发工具的新版本，以及一个新的IDE。

读者对象

我们设计本书的定位是用户指南和PL/SQL参考手册。本书适合于所有Oracle用户，尤其适合于那些以前从未使用过PL/SQL的用户。我们在第12章更详细地讨论了一些高级内容。

如何使用本书

本书包含12章和3个附录。其中章节被划分成三部分：概述和开发环境、PL/SQL基本特性，以及更多的PL/SQL特性。

第一部分：概述及开发环境

本书的第一部分介绍了PL/SQL和PL/SQL所运行的环境，还讨论了附赠光盘所提供的

PL/SQL开发环境。

第1章：PL/SQL简介

第1章介绍了PL/SQL，并且描述了该语言的一些主要特性，讨论了不同的PL/SQL版本，以及数据库版本和PL/SQL版本的对应关系。这一章结尾讨论了一个数据库模式，我们全书都使用该范例。

第2章：开发及运行环境

PL/SQL可以运行在许多不同类型的环境下，不仅可以运行在客户端，也可以运行在服务器端。在这一章中，我们讨论了PL/SQL引擎的不同位置，以及不同引擎之间的通信，还讨论了附赠光盘所包含的开发工具，包括屏幕抓图。

第二部分：PL/SQL基本特性

这部分讨论了PL/SQL的关系型特性，包括该语言的语法和数据类型，以及如何使用SQL、内置的SQL函数、游标、错误处理和集合。

第3章：PL/SQL基础

在讨论任何编程语言时，首先讨论的肯定是该语言的语法和数据类型。在这一章中，我们将了解PL/SQL提供的所有数据类型，以及它所支持的控制结构。

第4章：在PL/SQL中使用SQL

第4章讨论了PL/SQL的过程化结构。正是SQL的有效性，才使该语言功能强大和有用。在这一章中，还讨论了SQL查询和DML语句的用法，以及事务机制。

第5章：内置SQL函数

SQL语言有大量的内置函数，这些内置函数可以进一步增强该语言的作用。我们将在这一章学习这些函数，包括在不同数据库版本中的函数。

第6章：游标

游标被用于处理多数据行查询。借助于游标，你可以循环处理结果集，并且依次处理每个数据行。在这一章中，我们讨论了游标的语法，并且讨论了不同类型的检索循环，还讨论了如何使用游标变量。

第7章：错误处理

如果没有智能处理运行时错误的能力，则任何程序都不是完善的。PL/SQL支持异常的使用（类似于Java），其中异常提供了处理任何运行时问题的具体结构。在这一章中，我们将详细地讨论异常的工作机制和异常传播。

第8章：集合

集合是PL/SQL对象组，并且包含index-by表、嵌套表和可变数组。这些数据类型非常类似于其他语言的数组。我们将讨论如何在数据库中存储嵌套表和数组，以及有关的集合方法。

第三部分：更多PL/SQL特性

本书的第三部分讨论了如何使用过程、包、函数和触发器，还将简单讨论PL/SQL的高级特性。

第9章：创建过程、函数和包

在第9章中，我们将讨论怎样创建过程和函数（通称为子程序），以及如何在数据库中存储

它们；还将讨论了创建子程序和包的语法，以及参数的传递机制。

第10章：使用过程、函数和包

第10章进一步扩展了第9章的内容，讨论了有关子程序和包的其他特性，包括如何从SQL语句中调用，以及相互间的依赖关系；还将讨论包怎样与共享池相互交互。

第11章：数据库触发器

触发器是特殊类型的PL/SQL语句块，无论何时触发事件发生时，相应触发器都会自动执行。触发事件可能是诸如INSERT语句的DML操作、DDL语句或者系统事件。触发器也可以代替特定DML语句被执行。我们将在这一章详细讨论这些触发器。

第12章：高级特性

第12章简单讨论了PL/SQL的一些高级特性。这包括对象类型、动态SQL、LOB和管道化函数；还将讨论一些更常见的内置包，比如DBMS_ALERT、DBMS_JOB、DBMS_LOB、DBMS_PIPE、DBMS_SQL和UTL_FILE。同时，还将讨论Oracle9i对通信包UTL_TCP、UTL_HTTP、UTL_SMTP和UTL_INADDR的增强特性。

附录

附录部分将提供有关PL/SQL的有用参考信息。

附录A：包指南

该附录讨论了Oracle中有效的内置包。这些包可以大大地扩展该语言的功能。

附录B：PL/SQL保留字

该附录给出PL/SQL的所有保留字。在命名变量和其他PL/SQL对象时，你应该避免使用它们。

附录C：数据字典

该附录总结许多数据字典视图，尤其是那些与PL/SQL程序员最相关的数据字典视图。

关于本书光盘

本书光盘包含两类信息：

- 本书所有范例的电子版本。你可以在code目录下找到这些代码文件。如想获得这些范例的信息，请参考code目录下的readme.txt文件。
- 来自5个不同供应商的6个完整PL/SQL开发环境的测试版本。所有这些工具都是完整的，并且被授权可以使用一定的天数。我们在第2章中对它们进行了详细的论述，这些开发工具都位于光盘的Development Tools目录下。下面是所提供的工具：

工 具	版 本	供应商
Rapid SQL	5.7.1	Embarcadero Technologies, www.embarcadero.com
DBPartner Debugger	4.0.1(Build 83)	Compuware Corporation, www.compuware.com
SQL Navigator 3	3.2d11	Quest Software, www.quest.com

(续)

工 具	版 本	供应商
TOAD	7.1.7.21	Quest Software, www.quest.com
SQL-Programmer	11.01.0	BMC Software , www.bmc.com
PL/SQL Developer	4.0.3.415	Allround Automations, www.allroundautomations.nl

当把本光盘插入到Windows系统的光驱中时，一个小应用程序将会自动启动，显示本光盘所包含的内容，以及提供有关如何安装每种工具的细节信息和授权需求。如想了解更多信息，请参考本光盘根目录下的readme.htm文件。

目 录

译者序

前言

第一部分 概述及开发环境

第1章 PL/SQL概述	1
1.1 为什么使用PL/SQL	1
1.1.1 PL/SQL和网络流量	2
1.1.2 标准	3
1.2 PL/SQL的功能特性	3
1.2.1 语句块结构	4
1.2.2 错误处理	4
1.2.3 变量和类型	5
1.2.4 条件语句	5
1.2.5 循环结构	6
1.2.6 游标	7
1.2.7 过程和函数	7
1.2.8 包	8
1.2.9 集合	9
1.3 本书使用的约定	9
1.3.1 PL/SQL和Oracle版本	9
1.3.2 Oracle文档	11
1.3.3 光盘内容	11
1.4 范例样表	12
1.5 小结	19
第2章 开发及运行环境	20
2.1 应用程序模型和PL/SQL	20
2.1.1 两层模型	20
2.1.2 三层模型	24
2.1.3 连接服务器	25
2.2 PL/SQL开发工具	25
2.2.1 SQL*Plus	26
2.2.2 Rapid SQL	30

2.2.3 DBPartner Debugger	34
2.2.4 SQL Navigator	37
2.2.5 TOAD	41
2.2.6 SQL-Programmer	44
2.2.7 PL/SQL Developer	48
2.2.8 开发工具小结	50
2.3 小结	52

第二部分 PL/SQL基本特性

第3章 PL/SQL基础	53
3.1 PL/SQL语句块	53
3.2 词法单元	59
3.2.1 标识符	60
3.2.2 分界符	62
3.2.3 文字	62
3.2.4 注释	64
3.3 变量声明	65
3.3.1 声明语法	65
3.3.2 变量初始化	66
3.4 PL/SQL类型	66
3.4.1 标量类型	67
3.4.2 复合类型	74
3.4.3 引用类型	74
3.4.4 LOB类型	74
3.4.5 对象类型	74
3.4.6 使用%TYPE	75
3.4.7 用户定义子类型	75
3.4.8 数据类型之间的转换	76
3.4.9 变量作用域和可见性	78
3.5 表达式和运算符	79
3.5.1 赋值	79
3.5.2 表达式	80

3.6 PL/SQL控制结构	82	4.4.1 对象和系统权限	119
3.6.1 IF-THEN-ELSE	82	4.4.2 GRANT和REVOKE	120
3.6.2 CASE	86	4.4.3 角色	121
3.6.3 循环	89	4.5 事务控制	123
3.6.4 GOTO和标号	93	4.5.1 COMMIT和ROLLBACK	123
3.6.5 编译指示	95	4.5.2 保存点	124
3.7 PL/SQL记录	96	4.5.3 事务和语句块	125
3.7.1 记录赋值	97	4.5.4 自治事务	126
3.7.2 使用%ROWTYPE	98	4.6 小结	130
3.8 PL/SQL风格指南	99	第5章 内置SQL函数	131
3.8.1 注释风格	99	5.1 概述	131
3.8.2 变量名风格	100	5.2 返回字符值的字符函数	131
3.8.3 大写风格	101	5.2.1 SUBSTR、SUBSTRB、SUBSTRC、 SUBSTR2和SUBSTR4	133
3.8.4 缩进风格	101	5.2.2 SOUNDEX	134
3.8.5 常见风格	101	5.3 返回数字值的字符函数	135
3.9 小结	102	5.3.1 INSTR、INSTRB、INSTRC、 INSTR2和INSTR4	136
第4章 PL/SQL中的SQL	103	5.3.2 LENGTH、LENGTHB、LENGTHC、 LENGTH2和LENGTH4	137
4.1 SQL语句	103	5.4 NLS函数	138
4.1.1 在PL/SQL中使用SQL	103	5.5 数字函数	139
4.1.2 使用动态SQL	104	5.6 日期和时间函数	142
4.2 在PL/SQL中使用DML	105	5.7 转换函数	145
4.2.1 SELECT	106	5.7.1 TO_CHAR(日期和日期时间)	147
4.2.2 INSERT	108	5.7.2 TO_CHAR(数字)	149
4.2.3 UPDATE	109	5.7.3 TO_DATE	151
4.2.4 DELETE	110	5.7.4 TO_NUMBER	151
4.2.5 WHERE子句	111	5.7.5 TO_TIMESTAMP和TO_ TIMESTAMP_TZ	151
4.2.6 批绑定	113	5.8 聚集和分析函数	152
4.2.7 RETURNING子句	114	5.9 其他函数	154
4.2.8 表引用	116	5.9.1 DUMP	156
4.2.9 数据库链接	116	5.9.2 USERENV	157
4.2.10 同义词	117	5.10 小结	158
4.3 伪列	117	第6章 游标	159
4.3.1 CURRVAL和NEXTVAL	117	6.1 什么是游标	159
4.3.2 LEVEL	118		
4.3.3 ROWID	118		
4.3.4 ROWNUM	119		
4.4 GRANT、REVOKE和权限	119		

6.1.1 处理显式游标	160	8.1.2 嵌套表	221
6.1.2 处理隐式游标	167	8.1.3 可变数组	224
6.2 游标检索循环	168	8.1.4 多层集合	227
6.2.1 简单循环	168	8.1.5 集合类型之间的比较	228
6.2.2 WHILE循环	170	8.2 数据库中的集合	229
6.2.3 游标FOR循环	171	8.2.1 存储集合的隐含式	229
6.2.4 NO_DATA_FOUND与%NOTFOUND	173	8.2.2 操作整个集合	232
6.2.5 SELECT FOR UPDATE游标	173	8.2.3 操作单个集合元素	238
6.3 游标变量	177	8.3 集合方法	243
6.3.1 声明游标变量	177	8.3.1 EXISTS	244
6.3.2 为游标变量分配存储空间	178	8.3.2 COUNT	246
6.3.3 为查询打开游标变量	179	8.3.3 LIMIT	246
6.3.4 关闭游标变量	180	8.3.4 FIRST和LAST	247
6.3.5 游标变量范例一	180	8.3.5 NEXT和PRIOR	247
6.3.6 游标变量范例二	182	8.3.6 EXTEND	248
6.3.7 使用游标变量的限制	183	8.3.7 TRIM	251
6.4 小结	184	8.3.8 DELETE	252
第7章 错误处理	185	8.4 小结	254
7.1 什么是异常	185		
7.1.1 声明异常	187	第三部分 更多PL/SQL特性	
7.1.2 抛出异常	190		
7.1.3 处理异常	191	第9章 创建过程、函数和包	255
7.1.4 EXCEPTION_INIT 编译指示	196	9.1 过程与函数	255
7.1.5 使用RAISE_APPLICATION_ERROR	197	9.1.1 创建子程序	256
7.2 异常传播	200	9.1.2 子程序参数	260
7.2.1 可执行部分产生的异常	200	9.1.3 CALL语句	277
7.2.2 在声明部分出现的异常	203	9.1.4 过程与函数	279
7.2.3 在异常部分产生的异常	204	9.2 包	279
7.3 异常准则	205	9.2.1 包的说明	279
7.3.1 异常的作用域	206	9.2.2 包体	280
7.3.2 避免未处理异常	207	9.2.3 包和范围	282
7.3.3 指定错误位置	207	9.2.4 重载封装子程序	285
7.4 通用错误处理器	208	9.2.5 包的初始化	288
7.5 小结	216	9.3 小结	289
第8章 集合	217	第10章 使用过程、函数和包	290
8.1 声明和使用集合类型	217	10.1 子程序位置	290
8.1.1 index-by表	217	10.1.1 存储子程序和数据字典	290
		10.1.2 局部子程序	292

10.1.3 存储子程序与局部子程序	297	11.4 小结	364
10.2 存储子程序和包的考虑事项	298	第12章 高级特性	365
10.2.1 子程序依赖性	298	12.1 语言特性	365
10.2.2 包运行时状态	307	12.1.1 外部例程	365
10.2.3 特权和存储子程序	312	12.1.2 本地动态SQL	367
10.3 在SQL语句中使用存储函数	321	12.1.3 成批绑定	372
10.3.1 纯度等级	321	12.1.4 对象类型	380
10.3.2 缺省参数	327	12.1.5 大对象	384
10.3.3 在Oracle8i中从SQL调用存储函数	327	12.1.6 管道化表函数	386
10.4 在共享池中的锁定	330	12.2 高级包	387
10.4.1 KEEP	330	12.2.1 DBMS_SQL	387
10.4.2 UNKEEP	331	12.2.2 DBMS_PIPE	388
10.4.3 SIZES	331	12.2.3 DBMS_ALERT	390
10.4.4 ABORTED_REQUEST_ THRESHOLD	331	12.2.4 UTL_FILE	391
10.5 小结	331	12.2.5 UTL_TCP	392
第11章 数据库触发器	332	12.2.6 UTL_SMTP	392
11.1 触发器的类型	332	12.2.7 UTL_HTTP	393
11.2 创建触发器	335	12.2.8 UTL_INADDR	394
11.2.1 创建DML触发器	335	12.2.9 DBMS_JOB	394
11.2.2 创建Instead-Of触发器	343	12.2.10 DBMS_LOB	395
11.2.3 创建系统触发器	349	12.3 小结	399
11.2.4 其他触发器问题	354		
11.2.5 触发器和数据字典	358		
11.3 变异表	359		
11.3.1 变异表的例子	361		
11.3.2 变异表错误的解决	362		

附 录

附录A 包指南	401
附录B PL/SQL保留字	414
附录C 数据字典	417

第一部分 概述及开发环境

第1章 PL/SQL概述

PL/SQL是用于从各种环境中访问Oracle数据库的一种比较复杂的编程语言。它与数据库服务器集成在一起，因此，PL/SQL代码可以得到快速有效的处理。同时，也可在一些客户端的Oracle工具中使用。在本章中，我们将讨论人们开发PL/SQL的动机，PL/SQL的功能特性，以及认知PL/SQL和数据库版本的重要性。此外，还将介绍一些术语，我们将会在本书中详细研究它们。最后将说明有关本书所使用的一些约定以及本书将用作范例的一些数据库表。

1.1 为什么使用PL/SQL

Oracle是一种关系型数据库，用于访问关系型数据库的语言是结构化查询语言（Structured Query Language, SQL）。SQL是一种灵活有效的语言，它的一些功能特性专门用来处理和检验关系型数据。例如，下面的SQL语句将从数据库中删除所有主修营养学的学生记录：

```
DELETE FROM students
WHERE major = 'Nutrition';
```

（本书所用到的数据库表，其中包括students表，将在本章末尾给出）

SQL是第四代语言（the fourth-generation language, 4GL）。这意味着它描述的是应该做什么，而不是怎么做。例如，在前面的DELETE的语句中，我们并不知道服务器是如何确定哪些学生主修营养学。也许，它是按一定的次序循环处理所有学生以决定应该删除的是哪些表项。但是，这些处理细节都被系统隐藏，我们将无法知晓。

第三代语言(the third-generation language, 3GL)，例如C或者COBOL，在本质上就更过程化。用第三代语言编写的程序在解决问题时实现的是一步接一步的算法。例如，我们用以下的程序来完成DELETE操作：

```
LOOP over each student record
  IF this record has major = 'Nutrition' THEN
    DELETE this record;
  END IF;
END LOOP;
```

面向对象语言，例如，C++或者Java，也是第三代语言。虽然它们融合了面向对象设计的原则，但是它们的算法仍然是一步接一步的。

每种类型的语言都有其优点和缺点。一般而言，像SQL这样的第四代语言相当简单（与第三代语言相比），并提供了较少的命令。第四代语言也使用户不必关心那些由运行系统实现的底层数据结构和算法。但是在某些情况下，3GL中的过程性结构对于某些程序的表达是非常有用的。

这就是开发PL/SQL的原因所在——它将SQL语言（4GL）的强大性和灵活性与3GL的过程性结构融为一体。

PL/SQL是Procedural Language/SQL的缩写。正如其名字所示，PL/SQL通过增加了用在其他过程性语言中的结构来对SQL进行了扩展，例如：

- 变量和类型（包括预定义和用户定义）。
- 控制结构，如IF-THEN-ELSE语句和循环。
- 过程和函数。
- 对象类型和方法（Oracle8以及更高版本）。

过程性结构与Oracle SQL无缝地集成在一起，这便产生了一种结构化的强有力的语言。例如，假设我们想修改一个学生记录的主修课。如果这个学生不存在，我们就想创建一个新记录。那么，我们可以通过以下的PL/SQL代码实现这个目的：

```
-- Available online as 3gl_4gl.sql
DECLARE
  /* Declare variables which will be used in SQL statements */
  v_NewMajor VARCHAR2(10) := 'History';
  v_FirstName VARCHAR2(10) := 'Scott';
  v_LastName VARCHAR2(10) := 'Urman';
BEGIN
  /* Update the students table. */
  UPDATE students
    SET major = v_NewMajor
    WHERE first_name = v_FirstName
    AND last_name = v_LastName;
  /* Check to see if the record was found. If not, then we need
  to insert this record. */
  IF SQL%NOTFOUND THEN
    INSERT INTO students (ID, first_name, last_name, major)
      VALUES (student_sequence.NEXTVAL, v_FirstName, v_LastName,
              v_NewMajor);
  END IF;
END;
```

这个例子包括了两个不同的SQL语句（UPDATE和INSERT），这是4GL的结构，同时还有3GL的结构（变量声明和IF条件语句）。

注意 为了能执行前面的例子，你需要先创建被引用的数据库对象（students表和student_sequence序列）。这可通过执行tables.sql脚本语句实现，该程序是在线代码的一部分。

PL/SQL是独一无二的，因为它融合了灵活的SQL和强大的并具有可配置性的3GL。同时，可以使用该语言集成的过程性结构和数据库访问接口。这些特性使该语言更加健壮和功能强大，非常适合于设计复杂的应用程序。

1.1.1 PL/SQL和网络流量

许多数据库应用程序的构建都使用客户机/服务器模型或者三层模型。在客户机/服务器模型

中，程序本身在客户机上，它将请求发往数据库服务器，等待所需的信息。这些请求使用SQL语言实现。通常，这样做会产生许多次网络传输交互，每个SQL语句都有单独的传输交互。这个过程可用图1-1左边的框图表示。同时，将它与右边的框图进行比较。在右图中，几条SQL语句可被绑定在一个PL/SQL语句块中，作为一个单独的单元发往服务器。这样做网络流量会减少，使应用程序执行得更快。

即使当客户机和服务器都运行在一台机器上时，这样做也可以提高整个系统的性能。在这种情况下，不存在任何网络，但是将SQL打包会获得更简单的对数据库进行更少次数调用的程序。

PL/SQL打包的优点同样适用于三层模型。在这种情况下，客户机（通常运行在HTML浏览器中）和应用程序服务器进行交互，而应用程序服务器随后将同数据库进行交互。后一种交互充分体现了PL/SQL的优点。我们将在第2章中更多地讨论这种类型环境。

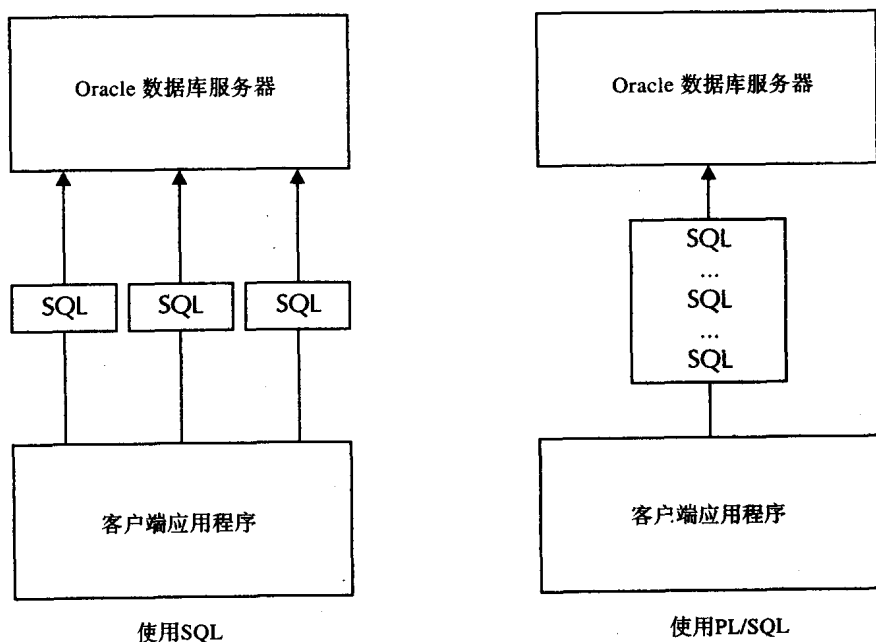


图1-1 客户机/服务器环境中的PL/SQL

1.1.2 标准

在SQL语言方面，Oracle支持美国国家标准组织（American National Standards Institute, ANSI）标准，在文档ANSI/ISO/IEC 9075-1:1999 “Database Language SQL”中对SQL语言进行了定义。这个标准，即通常所知的SQL99(或SQL2)，仅仅定义了SQL语言，它没有定义PL/SQL所提供的对该语言的3GL扩展。Oracle9i 包含了该标准CORE部分所要求的大多数特性。详情请参见Oracle SQL Reference。

1.2 PL/SQL的功能特性

最好通过范例来说明PL/SQL的许多不同特性和功能。下面的章节描述了该语言的一些主要

特性。我们将在本书中详细讨论它们。

1.2.1 语句块结构

PL/SQL的基本单元是语句块。所有的PL/SQL程序都是由语句块构成的，它们之间可以相互嵌套。通常，每一语句块负责完成程序中某单元的工作，这样每一语句块就可以分担不同的任务了。语句块拥有以下的结构。

```
DECLARE
    /* Declarative section - PL/SQL variables, types, cursors,
       and local subprograms go here. */
BEGIN
    /* Executable section - procedural and SQL statements go here.
       This is the main section of the block and the only one
       that is required. */
EXCEPTION
    /* Exception handling section - error-handling statements go
       here. */
END;
```

只有可执行部分是必需的，声明部分和异常处理部分是可选的。可执行部分必须至少包含一条可执行语句。语句块的不同部分将分派PL/SQL程序的不同处理功能。

PL/SQL的设计依然遵循以Ada为代表的第三代语言的特征。Ada中的许多结构也可以在PL/SQL中找到，其中就包括语句块结构。另外，在PL/SQL中也可以发现其他的Ada特征，包括异常处理，用于声明过程和函数的语法，以及包。

1.2.2 错误处理

语句块的异常处理部分用于响应程序所遇到的运行时错误。通过将错误处理代码与程序主体分离可以使程序本身的结构变得清晰。例如，下面的PL/SQL语句块将进行这样的异常处理，它将错误发生的当前时间和遇到错误的用户都记录下来。

```
-- Available online as Error.sql
DECLARE
    v_ErrorCode NUMBER;          -- Code for the error
    v_ErrorMsg VARCHAR2(200);    -- Message text for the error
    v_CurrentUser VARCHAR2(8);   -- Current database user
    v_Information VARCHAR2(100); -- Information about the error
BEGIN
    /* Code that processes some data here */
EXCEPTION
    WHEN OTHERS THEN
        -- Assign values to the log variables, using built-in
        -- functions.
        v_ErrorCode := SQLCODE;
        v_ErrorMsg := SQLERRM;
        v_CurrentUser := USER;
```



```

v_Information := 'Error encountered on ' ||
TO_CHAR(SYSDATE) || ' by database user ' || v_CurrentUser;
-- Insert the log message into log_table.
INSERT INTO log_table (code, message, info)
VALUES (v_ErrorCode, v_ErrorMsg, v_Information);
END;
```

注意 上面的例子与本书其他例子都可以在本书配套光盘中找到。详情请参见本章末尾的“范例位置”。

1.2.3 变量和类型

在PL/SQL和数据库之间是通过变量传送消息的。一个变量是一个存储单元，程序可读取它，或给它赋值。在前面的例子中，v_CurrentUser、v_ErrorCode和v_Information都是变量。变量是在语句块的声明部分声明的。

每一个变量都有与之相关的明确类型。类型定义了该变量能够存储何种类型的信息。PL/SQL变量可以是数据库列拥有的类型：

```

DECLARE
v_StudentName VARCHAR2(20);
v_CurrentDate DATE;
v_NumberCredits NUMBER(3);
```

或者是其他类型：

```

DECLARE
v_LoopCounter BINARY_INTEGER;
v_CurrentlyRegistered BOOLEAN;
```

PL/SQL也支持用户定义类型：表和记录。用户定义类型允许用户定制自己程序中要处理的数据结构。

```

DECLARE
TYPE t_StudentRecord IS RECORD (
    FirstName VARCHAR2(10),
    LastName VARCHAR2(10),
    CurrentCredits NUMBER(3)
);
v_Student t_StudentRecord;
```

1.2.4 条件语句

PL/SQL程序可根据测试结果有条件地执行代码的不同部分。这样的语句称为条件语句。其主要结构就是IF语句。例如，下面的语句块将查询数据库以得到学生总数，并相应地向temp_table表中插入不同的信息：

```

-- Available online as Conditional.sql
DECLARE
v_TotalStudents NUMBER;
```