

万水计算机编程技术与应用系列

ORACLE Database Automation

ORACLE数据库 自动化编程

[美] Jonathan Ingram 著

前导工作室 译

CD-ROM



CORIOLIS
GROUP
BOOKS



中国水利水电出版社
www.waterpub.com.cn

万水计算机编程技术与应用系列

ORACLE 数据库自动化编程

【美】 Jonathan Ingram 著

前导工作室 译

李志、潇东、陈彦海 审校

中国水利水电出版社

内 容 提 要

本书主要介绍了 Oracle 8, 并将它和 Oracle 7 关系数据库进行比较。Oracle 8 引进了面向对象的技术, 它使用存储 PL/SQL 对象在数据库级实施事务规则, 是数据库自动化的一个重要部分; 同时它还增加了一些新的数据类型、嵌入式表的概念等, 这些新的内容使 Oracle 8 相对于 Oracle 7 关系数据库来说, 有了实质性的飞跃。

本书介绍了数据库自动化的概念, 强调了设计背后的概念, 提供了一些必需的调试代码的技巧, 提供了一些包的说明以及对 Oracle 8 未来的一些展望。全书结构严谨, 内容丰富, 是计算机爱好者和专业人员难得的学习和参考资料。

"Original English language edition published by The Coriolis Group, Inc., 14455N. Hayden Drive, Suite 200, Scottsdale, Arizona 85260 USA, telephone (602) 483-0192, fax (602) 483-0193. Copyright © 1997 by The Coriolis Group. All right reserved."

本书中文简体字版由中国水利水电出版社独家出版。未经出版者书面同意, 不得以任何方式复制或抄袭本书的任何部分。

图书在版编目(CIP)数据

ORACLE 数据库自动化编程/(美)英格拉姆(Ingram, J.)著;前导工作室译.—北京:中国水利水电出版社, 1998.8

(万水计算机编程技术与应用系列)

书名原文: High Performance Oracle Database Automation

ISBN 7-80124-812-0

I. O… II. ①英… ②前… III. 数据库系统—软件工具, Oracle-程序设计 IV.TP311.56

中国版本图书馆 CIP 数据核字(98)第 16059 号

书 名	ORACLE 数据库自动化编程
作 者	[美] Jonathan Ingram 著
译 者	前导工作室 译
审 校	李志、满东、陈彦海
出版、发行	中国水利水电出版社(北京市三里河路 6 号 100044) 网址: www.waterpub.com. Cn E-mail: sale @ waterpub. com. Cn 电话: (010) 63202266 (总机)、68331835 (发行部)
经 售	全国各地新华书店
排 版	北京万水电子信息有限公司
印 刷	北京市牛山世兴印刷厂
规 格	787×1092 毫米 16 开本 20.5 印张 472 千字
版 次	1998 年 9 月第一版 1998 年 9 月北京第一次印刷
印 数	0001—5000 册
定 价	49.00 元 (含光盘)

凡购买我社图书, 如有缺页、倒页、脱页者, 本社发行部负责调换

版权所有·侵权必究

译 者 序

Oracle 是当今世界数据库管理系统的领头羊，它占据了世界上数据库应用的大部分市场。

为了介绍当今有关 Oracle 最新的发展情况，我们翻译了这本书。此书译自 Donald Burleson 所著的 high performance ORACLE Database Automation。

本书介绍了 Oracle 数据库系统的特征，呈现给读者的是有关 Oracle 数据库自动化的重要设计与编程思想及其概念，而不是简单的语法。为了说明问题，本书还提供了一些详细的实例帮助读者理解和加深了解。

仔细阅读此书，对读者了解 Oracle 数据库系统，并应用它编程，调试，改善性能以完成特定的任务有很大帮助。同时对于了解 Oracle 数据库的现状及其实现思想来说，这也是一本难得的好书。

本书由张奇飞、张志伟组织翻译，参加翻译工作的还有倪晓强、阎琪、祝恩、张志敏、刘国庆、汤朝阳、史章军、刘汇、李志、张巧莉、李林、王建民、潇东、崔翔、陈彦海、刘辉、孙议、张志敏、刘国庆、汤朝阳等。全书由李志、潇东、陈彦海进行审校。

由于时间仓促，此书翻译存在错误是难免的，敬请读者批评指正。

译 者

1998 年 6 月

致 谢

很久以前，我就决定在我的一生当中写几本书，但我从未想过我所写的第一本书会是一本科技书。但是你瞧，这就是它。

科利欧里斯小组 (The Coriolise Group) 中极好的人们在我完成这本书时给我的帮助充分显示了他们极大的信任和耐心。这本书是对他们诚挚的回报。我希望我已经实现了最初的目标，即：至少与强调语法一样强调设计和测试技巧。我希望读者们能发现在这些目标上这本书是有用的。我想这些技巧在任何应用程序开发上将有所作用。

在完成这本书时，许多人在各方面给予了我帮助，我想在此对他们表示感谢：

- 我的妻子莎拉(Sara)富有耐心和爱心。在多方面给予了我支持。
- 我的朋友马克·利费 (Mark Rife) 给我出各种主意并且是一个优秀的技术编辑，他也给予了许多帮助。
- 我的编辑，汤利·祝卡立里 (Toni Zuccarini)，总是非常耐心，积极帮助，感谢汤利 (Toni)。
- 我的好朋友，以前的同事，荷纳奥笛 (Herald Witliam) 与窦革·杨 (Doug Young) 在这项工程中给予我特别的支持 (以他们特有的方式)。
- 我的同事们都对我予了支持，提出了有益的观点，所有这一切激励我完成此书。

另外有很多人给予了我很多支持，把他们全部列出来本身就要占一本书的空间。他们对我都特别友好，没有他们的友谊与支持，我是无法完成这项巨大的工程的。

作 者 简 介

Jonathon Ingram 拥有使用 SQL*Plus 和 PL/SQL 开发 Oracle 数据库应用的多年经验。他已经设计、编写、修改了难以数计的存储 PL/SQL 代码。Jonathan 拥有 Southern Mississippi 大学学位，现在是华盛顿某公司的数据库管理员。

引 言

1991 年，当我开始使用 Oracle 关系数据库时，我甚至还未想过我在做什么，也没意识到这种工具在我的将来会有多重要。而现在，在将近六年之后，我已在 SQL 和 PL/SQL 方面写了几十万行代码了。

Oracle 是一个非常重要的产品，它驱动世界上大多数最大的数据库，并在众多开放系统中运行，包括最受欢迎的 Unix。Oracle 比起其他关系数据库来，是一种更有伸缩性的工具，它允许商人们定义他们的操作并围绕他们建立有效的系统。

“数据库自动化”(Database automation)看起来像个含糊主题，但在现实中它是很简单易懂的。脚本化是数据库自动化的前半部分，它非常重要，它允许数据库管理者和应用开发者在简洁和高可靠性、高重复性的基础上执行复杂的任务。

使用存储 PL/SQL 代码在数据库级实施业务规则，是数据库自动化的第二个部分。数据库触发器(trigger)、过程、函数和包在实现一个复杂系统时非常有用，极大地减少了必须为系统前端书写的大量代码。

这本书的目的不在于教你有关 SQL 和 PL/SQL 编程的语法，而是强调隐藏在代码后面的概念。针对这一目的，我强调把设计、归档(documenting)以及测试代码作为应用开发者的最大责任。如果读者需要知道语法，请参看手册。技术是一个应用开发者拥有的、用来对付大系统的最重要的武器。

第一章是对数据库自动化主要概念的介绍。本章同时还讨论一些长期的、使数据库自动化更有力量的发展。第二章是 Oracle SQL 与 PL/SQL 的初级读本，这部分是为那些 Oracle 初学者或对数据库知之甚少的人而写的。

从第三章到第七章强调设计背后的概念，包括编写代码以及测试源程序，存储过程，存储函数，包，数据库触发器。设计、编写和测试比语法更重要。

第八章提供一些必须的调试代码的技巧，着重强调在运行时出错调试的最好方法以及写代码时最易出错的调试和维护方法。第九章讨论 Oracle 提供的包以及举例说明描述现实世界如何利用这个工具。最后，第十章讨论你可以使用的提示和技术，它们使 SQL 和 PL/SQL 代码得到最好的性能。另外还包括对 Oracle8 的未来的一些展望，以及一个 PL/SQL 编程标准，它可用来符合读者业务的需要。

该书首先以覆盖必需为宗旨，然后提取最有用的。但像这样大小的工程，读者总会发现它还有更多的事需做。如果您对本书有何评论，请指出不妥之处，或只是在附加上一句鼓励的话，欢迎来信，E-MAIL: Jingram@teleport.com。

来自 Donald Burleson 的问候

今天的 Oracle 专业人员正处在一个转折点。随着 Oracle 技术进入 21 世纪，我们看到数据库系统的复杂度变得几乎不可预测。今天的 Oracle 专业人员在数据库性能与性能调整，数据库管理，数据仓库，在 web 上使用 Oracle，使用 OLAP 和空间数据以及许多别的领域都必须成为专家。Oracle 这些强大的新特征给每个使用 Oracle 技术来解决复杂数据问题的人提出了巨大的挑战。

Oracle 是当今世界数据库管理系统的带头者，在其特征和可供选择方面提供了令人吃惊的财富，其程度远超出一个 Oracle 专业人员所能简单地消化。随着 Oracle 引入对象管理、数据仓库和能用于 web 的应用，Oracle 市场占有率将不断增加。遍布全球的 Oracle 专业人员正努力理解这些新概念，以充分利用这些新特征。

Oracle 专业人员成为高手的日子已为期不远了，他们必须逐渐熟悉 Oracle 技术的所有方面并懂得这些技术是如何相互操作。Oracle 专业人员必须有足够充分的深度来有效地适应技术，而不是简单地拓宽知识面。为得到这些知识，我们必须依靠专家，指导我们通过这一复杂的工具和技术迷宫，而不是希望通过常用的技术手册。

我们的需求很清晰，把从 Oracle 专业人员那里得来的忠告加以精简，这就是科利欧里斯小组的高性能 Oracle 丛书的目标。随着你们面临与这令人兴奋的新技术保持同步的挑战，我们同样面临挑战——提供优秀的书本来帮助你们掌握确保您们成功的众多的 Oracle 新特征。

Don Burleson
Rochester, New York

目 录

译者序

致谢

作者简介

引言

第一章 现实世界的数据库自动化	1
1.1 脚本化	1
1.1.1 备份与恢复	1
1.1.2 创建和维护用户	4
1.1.3 生成动态代码	4
1.1.4 测试	7
1.1.5 其他用途	8
1.2 存储 PL/SQL 对象	9
1.2.1 过程	9
1.2.2 函数	11
1.2.3 包	12
1.2.4 数据库触发器 (Database Trigger)	14
1.3 数据库自动化的未来	16
1.3.1 业务规则引擎 (Engine)	17
1.4 小结	17
第二章 ORACLE 数据库的体系结构、SQL 与 PL/SQL	18
2.1 Oracle 数据库的体系结构	18
2.1.1 约束	18
2.1.2 创建索引	22
2.1.3 特权与角色	23
2.1.4 模式	24
2.1.5 顺序机制	25
2.1.6 快照	25
2.1.7 同义字	25
2.1.8 表与列	26
2.1.9 视图	26
2.1.10 系统全局区域	26
2.2 什么是 SQL	28

2.2.1	数据定义语言	28
2.2.2	数据操作语言	29
2.2.3	提交与回退 (rollback)	29
2.2.4	数据类型	29
2.2.5	连接	31
2.2.6	表别名	32
2.2.7	锁	32
2.2.8	空值 (NULL VALUE)	33
2.2.9	操作符	33
2.2.10	子查询	34
2.3	PL/SQL 总览	34
2.3.1	块	35
2.3.2	游标 (Cursor)	36
2.3.3	数据类型	37
2.3.4	异常与异常处理	41
2.3.5	IF-THEN-ELSE 逻辑	44
2.3.6	NULL 语句	44
2.3.7	循环结构	44
2.3.8	存储对象	46
2.3.9	PL/SQL 愿望表	51
2.3.10	数据库怎样解析 SQL 与 PL/SQL	52
2.4	小结	45
第三章	SQL 与 PL/SQL 脚本化	55
3.1	脚本面面观	55
3.1.1	备份与恢复	56
3.1.2	动态生成代码	56
3.1.3	安全与用户管理	57
3.1.4	报表	59
3.1.5	测试	60
3.1.6	其他任务	64
3.1.7	对脚本进行归档	65
3.2	SQL*Plus 中的脚本	67
3.2.1	SQL*Plus 命令	67
3.2.2	连接到 SQL*Plus	76
3.2.3	退出 SQL*Plus	76
3.2.4	脚本参数	77
3.2.5	成卷输出到文件	77
3.2.6	替代变量	77

3.3	脚本中的 PL/SQL	78
3.3.1	匿名 PL/SQL 块	78
3.3.2	使用 DBMS_Output 和 UTL_File	78
3.4	循序渐进: 在 SQL*Plus 中建立报表	79
3.4.1	需求	79
3.4.2	伪代码	79
3.4.3	代码	79
3.5	小结	80
第四章	过程	82
4.1	过程的优点	82
4.1.1	嵌入式 SQL	82
4.1.2	可维护性	82
4.1.3	模块化	82
4.1.4	提高性能	83
4.1.5	可重用性	84
4.2	创建与删除 (drop) 过程	84
4.2.1	重编译过程	85
4.2.2	局部过程	85
4.3	依赖性	88
4.4	参数	89
4.4.1	数据类型	89
4.4.2	参数类型	90
4.4.3	省缺值	90
4.5	调用过程	91
4.5.1	匿名 PL/SQL 块	91
4.5.2	存储 PL/SQL 对象	92
4.5.3	表示法	94
4.6	过程的结构	95
4.6.1	过程声明	95
4.6.2	变量声明	97
4.6.3	可执行声明	98
4.6.4	过程体	100
4.6.5	异常处理器	101
4.7	归档过程	106
4.7.1	标头	107
4.7.2	伪代码	108
4.7.3	注释	109
4.7.4	标识符	110

4.8	循序渐进：设计一个过程.....	110
4.8.1	过程要求	110
4.8.2	设计	111
4.8.3	伪代码	111
4.8.4	编码	112
4.8.5	测试过程	116
4.9	小结	118
第五章	函数	119
5.1	函数的优点	119
5.1.1	嵌入式 SQL.....	119
5.1.2	可维护性	119
5.1.3	模块化	120
5.1.4	提高性能	120
5.1.5	可靠性	121
5.2	创建与删除函数	121
5.2.1	局部函数	122
5.3	依赖性	124
5.4	参数	125
5.4.1	数据类型	125
5.5	返回值	126
5.5.1	数据类型	126
5.6	调用函数	128
5.6.1	DML 语句	128
5.6.2	匿名 PL/SQL 块.....	129
5.6.3	存储 LP/SQL 对象.....	129
5.7	函数的结构	130
5.7.1	函数声明	130
5.7.2	变量声明	132
5.7.3	可执行声明	134
5.7.4	函数体	136
5.7.5	异常处理	138
5.8	归档函数	139
5.8.1	标头	140
5.8.2	伪代码	141
5.8.3	注释	142
5.8.4	标识符	142
5.9	循序渐进：创建一个函数.....	143
5.9.1	函数要求	143

5.9.2	设计函数	144
5.9.3	伪代码	144
5.9.4	编码	145
5.9.5	测试	150
5.10	小结	154
第六章	包	155
6.1	包说明	155
6.1.1	全局变量	155
6.1.2	过程	158
6.1.3	函数	159
6.1.4	重载过程与函数	161
6.2	包体	162
6.2.1	过程	163
6.2.2	函数	164
6.2.3	初始化包	166
6.3	循序渐进: 建立包	166
6.3.1	要求	166
6.3.2	伪代码	168
6.3.3	编码	169
6.3.4	测试	173
6.4	小结	174
第七章	数据库触发器	175
7.1	数据库触发器的特征	175
7.1.1	嵌入式 DML 语句	175
7.1.2	事件驱动进程	176
7.1.3	可维护性	179
7.1.4	提高性能	179
7.1.5	引用列值	179
7.1.6	可重用性	179
7.1.7	:old 与 :new 定义	179
7.1.8	布尔函数	179
7.2	创建和删除触发器	181
7.2.1	重编译数据库触发器	182
7.2.2	禁止和允许触发器	182
7.3	互斥表错误	183
7.3.1	引用与触发器相关的表	183
7.3.2	外来关键字的引用	185

7.3.3	重叠删除	186
7.3.4	围绕规则工作	186
7.4	数据字典和触发器源代码	189
7.5	触发器的典型应用	189
7.5.1	实施复杂业务规则	189
7.5.2	更新相关数据	190
7.5.3	为处理而标记行	190
7.5.4	通知事件	190
7.6	触发器结构	190
7.6.1	触发器声明	190
7.6.2	触发事件	191
7.6.3	相关表	192
7.6.4	触发器级别	193
7.6.5	WHEN 子句	193
7.6.6	触发器体	194
7.7	归档触发器	195
7.7.1	触发器标头	195
7.8	循序渐进：创建一个触发器	196
7.8.1	触发器的要求	196
7.8.2	决定触发器的级别	197
7.8.3	伪代码	197
7.8.4	编码	198
7.8.5	测试触发器	199
7.9	小结	200
第八章	调试	201
8.1	编译错误	201
8.1.1	列出编译错误	201
8.1.2	修改编译错误	203
8.1.3	使用数据字典表	204
8.1.4	行号	205
8.1.5	当行号不正确时	206
8.2	运行错误	207
8.2.1	使用 DBMS_Output 包	207
8.2.2	定位运行错误	211
8.2.3	在执行过程中干净地处理异常	213
8.2.4	有用的函数	217
8.3	小结	218

第九章 特殊包	219
9.1 DBMS_Alert.....	219
9.1.1 Register () 过程	220
9.1.2 Remove () 过程	220
9.1.3 RemoveAll () 过程	220
9.1.4 Set_Defaults () 过程	220
9.1.5 Signal () 过程	220
9.1.6 WaitAny () 过程	221
9.1.7 WaitOne () 过程	222
9.1.8 使用信号	222
9.2 DBMS_DDL.....	223
9.2.1 Alter_Compile () 过程	224
9.2.2 Analyze_Object () 过程	224
9.3 DBMS_Describe	225
9.4 DBMS_Job	226
9.4.1 Broken () 过程	227
9.4.2 Change () 过程	227
9.4.3 Interval () 过程	227
9.4.4 ISubmit () 过程	228
9.4.5 Next_Date () 过程	228
9.4.6 Remove () 过程	228
9.4.7 Run () 过程	228
9.4.8 Submit () 过程	229
9.4.9 User_Export () 过程	229
9.4.10 What () 过程	229
9.4.11 安排一个工作	229
9.5 DBMS_Output.....	230
9.6 DBMS_Pipe.....	230
9.6.1 Create_Pipe () 函数	231
9.6.2 Next_Item_Type () 函数	232
9.6.3 Pack_Message () 过程	232
9.6.4 Purge () 过程	233
9.6.5 Receive_Message () 函数	233
9.6.6 Remove_Pipe () 函数	233
9.6.7 Send_Message () 函数	234
9.6.8 Unique_Session_Name () 函数	234
9.6.9 Unpack_Message () 过程	234
9.6.10 应用管道	235
9.7 DBMS_SQL.....	236
9.7.1 Bind_Variable () 过程	237

9.7.2	Close_Cursor () 过程	238
9.7.3	Column_Value () 过程	238
9.7.4	Define_Column () 过程	240
9.7.5	Execute () 函数	240
9.7.6	Execute_And_Fetch () 函数	241
9.7.7	Fetch_Rows () 函数	241
9.7.8	Open_Cursor () 函数	241
9.7.9	Parse () 过程	241
9.7.10	Variable_Value () 过程	242
9.7.11	DBMS_SQL 包的应用	243
9.8	DBMS_Utility	244
9.9	UTL_File	244
9.9.1	Fclose () 过程	245
9.9.2	FClose_All () 过程	245
9.9.3	FFlush () 过程	245
9.9.4	FOpen () 函数	245
9.9.5	Get_Line () 函数	246
9.9.6	Is_Open () 函数	246
9.9.7	New_Line () 过程	246
9.9.8	Put () 过程	246
9.9.9	PutF () 过程	247
9.9.10	Put_Line () 过程	247
9.9.11	UTL_FILE 包的应用	247
9.10	小结	248
第十章	性能调整	249
10.1	SQL 的性能调整	249
10.1.1	EXPLAIN PLAN 语句的应用	249
10.1.2	应用 TKPROF	251
10.1.3	其他调整性能的提示	255
10.2	调整 PL/SQL 性能	257
10.2.1	使用游标	257
10.2.2	异常处理	258
10.3	小结	261
附录 A	使用内嵌的 SQL 和 PL/SQL 函数	262
附录 B	DML 命令语法和释例	280
附录 C	与其他 ORACLE 专业人员联系	283

附录 D SQL 和 PL/SQL 编程标准	286
附录 E ORACLE8 中有些什么新内容	305

第一章 现实世界的数据库自动化

随着 Oracle7 进入关系数据库世界，Oracle 公司改变了基于 Oracle 的系统开发的方向。开发者曾经依靠 Oracle *Forms 来包含过程逻辑实施了复杂的业务规则，而现在他们能利用存储 PL/SQL 对象来实施这些规则。

对于 Oracle 公司和建立在 Oracle 数据库上的系统来说，这是一个开发范例的改变。存储在 Oracle7 数据库中的代码实施系统逻辑，减少在多个应用中的冗余代码。这导致了系统开发的一个新时代，即数据库触发器和其他存储 PL/SQL 对象介入系统开发的设计阶段。

尽管此范例的转变很重要，存储 PL/SQL 对象的使用只是数据库自动化的一个方面，它的另一方面是：预期甚至预先打包软件，包括使用脚本完成各种任务，如：

- ◆ 备份与恢复
- ◆ 创建/维护用户
- ◆ 动态代码生成
- ◆ 测试
- ◆ 其他各种目的

本章提供了大量详细的脚本实例，这些脚本有各自不同的作用；还提供了大量关于存储 PL/SQL 对象的例子。熟悉这些例子中的实际任务的操作是很有用的，但并非绝对必需。

1.1 脚本化

最古老的系统自动化形式包括使用脚本和其他机制来成批地处理工作。每种主要的操作系统提供一些开发脚本的能力，从简单的 DOS 批处理程序，到在 Unix 系统中无限制地应用 awk, sed 与 perl。在 Oracle 系统中，开发者能利用 SQL*Plus 与 PL/SQL 以及任何由此操作系统支持的脚本。

在 Oracle 数据库背景下，脚本化通常属于几个预定义类型中的一种。关于使用 SQL 与 PL/SQL 开发脚本的更多细节可查看第三章。

1.1.1 备份与恢复

在 Oracle 数据库系统中，所有用脚本描述来完成的的任务中，系统自动备份（其目的是为了数据库的恢复）可能是最重要的。绝大多数安装的 Oracle 系统（尤其是那些用热备份的）能够自动执行系统备份过程。

程序清单 1-1 是一个普通的 Oracle 数据库冷备份程序。它排除了 OS 级（操作系统级）命令执行在文件的备份中的工作。