

ORACLE

Mc
Graw
Hill
Education

Oracle Database 12c
PL/SQL Programming

ORACLE[®] **12^c**
DATABASE

Oracle Database 12c

PL/SQL开发指南(第7版)

设计和部署强大的、数据库为中心的PL/SQL应用

[美] Michael McLaughlin 著
陶佰明 邓超 刘颖 译

Mc
Graw
Hill
Education

清华大学出版社

Oracle Database 12c PL/SQL 开发指南 (第 7 版)

[美] Michael McLaughlin 著

陶佰明 邓超 刘颖 译

清华大学出版社

北 京

Michael McLaughlin
Oracle Database 12c PL/SQL Programming
EISBN: 978-0-07-181243-6
Copyright © 2014 by McGraw-Hill Education.

All Rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including without limitation photocopying, recording, taping, or any database, information or retrieval system, without the prior written permission of the publisher.

This authorized Chinese translation edition is jointly published by McGraw-Hill Education and Tsinghua University Press Limited. This edition is authorized for sale in the People's Republic of China only, excluding Hong Kong, Macao SAR and Taiwan.

Copyright © 2015 by McGraw-Hill Education and Tsinghua University Press Limited.

版权所有。未经出版人事先书面许可,对本出版物的任何部分不得以任何方式或途径复制或传播,包括但不限于复印、录制、录音,或通过任何数据库、信息或可检索的系统。

本授权中文简体字翻译版由麦格劳-希尔(亚洲)教育出版公司和清华大学出版社有限公司合作出版。此版本经授权仅限在中华人民共和国境内(不包括中国香港、澳门特别行政区和中国台湾地区)销售发行。

版权©2015 由麦格劳-希尔(亚洲)教育出版公司与清华大学出版社有限公司所有。

北京市版权局著作权合同登记号 图字:01-2014-2935

本书封面贴有 McGraw-Hill Education 公司防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

Oracle Database 12c PL/SQL 开发指南:第7版 / (美) 麦克劳克林(McLaughlin, M.) 著;陶佰明,邓超,刘颖 译. —北京:清华大学出版社,2015

书名原文:Oracle Database 12c PL/SQL Programming

ISBN 978-7-302-40364-7

I. O… II. ①麦… ②陶… ③邓… ④刘… III. ①关系数据库系统—程序设计—指南 IV. ①TP311.138-62

中国版本图书馆 CIP 数据核字(2015)第 114479 号



责任编辑:王 军 于 平

装帧设计:孔祥丰

责任校对:邱晓玉

责任印制:沈 露

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者:清华大学印刷厂

装 订 者:三河市新茂装订有限公司

经 销:全国新华书店

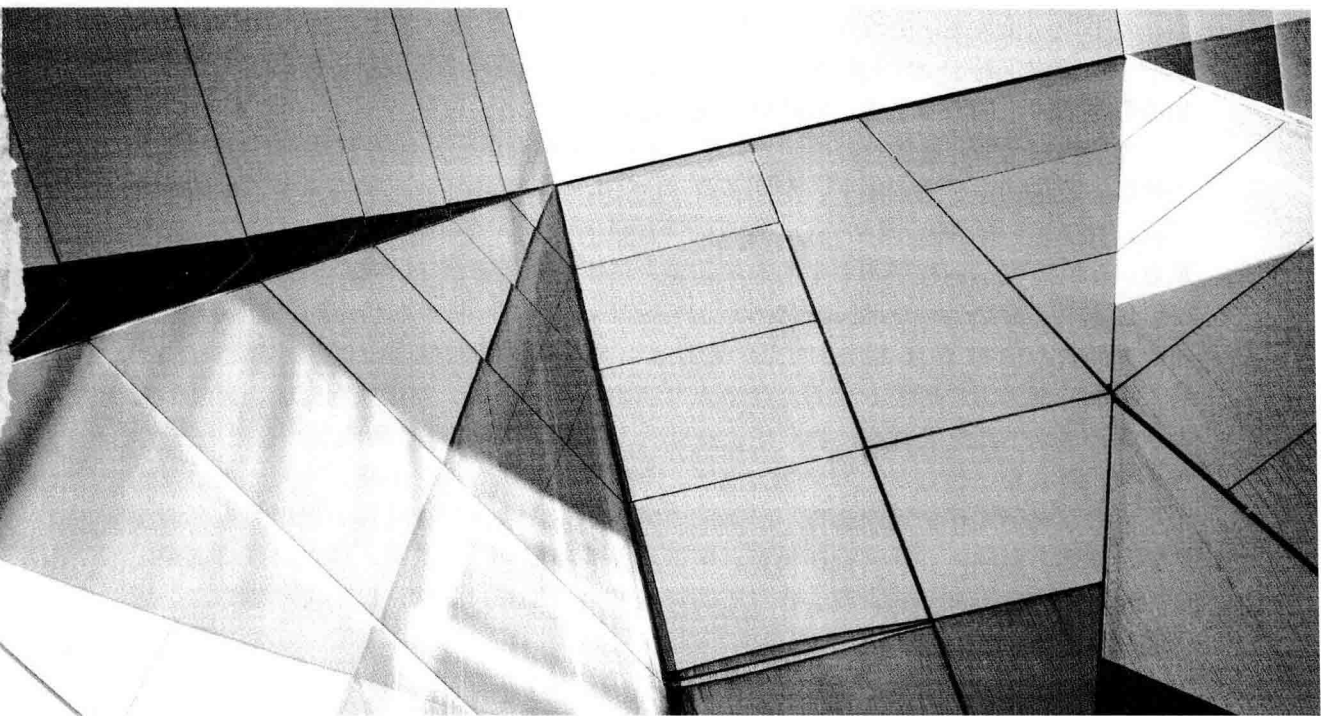
开 本:185mm×260mm 印 张:62.25 字 数:1633 千字

版 次:2015 年 8 月第 1 版 印 次:2015 年 8 月第 1 次印刷

印 数:1~3500

定 价:138.00 元

产品编号:059416-01



译者序

作为数据库行业的先驱和领军者，Oracle 早在 20 世纪 80 年代就认识到 SQL 语言在过程化编程中的不足。在原始的 SQL 规范上，Oracle 创建出自己的 SQL 语言：过程化语言/结构式查询语言(PL/SQL)。

PL/SQL 是一种程序化语言。PL/SQL 与 Java 等编程语言一样，是一种应用开发语言，支持块操作、条件判断、循环语句、嵌套等，可用于实现比较复杂的业务逻辑。PL/SQL 的引入使 SQL 成为一种高级程序设计语言。

PL/SQL 有自己的语法规则，是一门相对独立的语言。但是这并不影响它与原始 SQL 无缝地结合。PL/SQL 和 SQL 之间可以相互调用，使用起来灵活方便。实现了 SQL 的过程化编程，并支持事件驱动编程和面向对象编程。

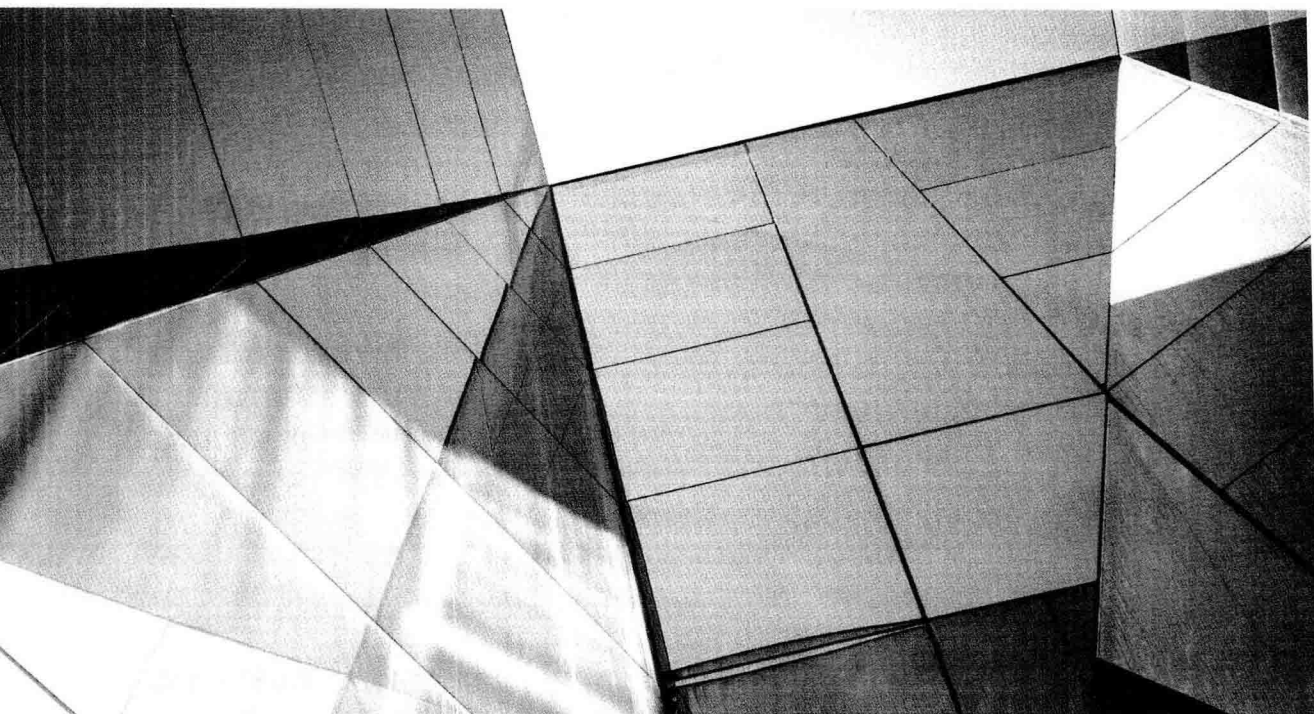
对于学习 PL/SQL 的读者来说，《Oracle Database 12c PL/SQL 开发指南(第 7 版)》是一本不可多得的参考指南，它涵盖了关于 PL/SQL 编程从入门到进阶的全部内容。本书共包含三部分：“Oracle PL/SQL”、“PL/SQL 编程”以及“附录和术语表”。第 I 部分介绍 PL/SQL 的基础知识，其中包括 Oracle 12c SQL 和 PL/SQL 的新特性；第 II 部分深入介绍如何使用 PL/SQL 编程；第 III 部分辅助介绍与 PL/SQL 相关的其他知识。

本书以理论为主，由浅入深，并辅以大量的程序示例，深入浅出，简单易懂。此外，本书中还提供了对于 PL/SQL 使用的忠告建议，这些都是作者汇集很多数据库大师级人物的智慧而得出的经验之谈，仔细研究这些建议会让读者受益匪浅。

本书内容全面，覆盖内容准确、完整。对于具有不同编程基础的 IT 从业者都有收藏的价值。建议初学者通读本书，全面了解 PL/SQL 知识。由于本书中各章节的依赖性并不强，对于有一定 SQL 或 PL/SQL 编程经验的读者来说，推荐采用跳跃式的阅读，把握重点，认真研究示例程序和书中的提示和忠告建议。对于了解 PL/SQL 程序的读者，不妨通过本书了解 Oracle Database 12c 以及 PL/SQL 的新特性，将本书当作 PL/SQL 的字典，随时翻阅查找与 PL/SQL 相关的知识，解决实际应用中遇到的问题。

本书的翻译工作历经 6 个月，全部章节由陶佰明、邓超、刘颖共同完成，另外参与本次翻译活动的还有孔祥亮、陈跃华、杜思明、熊晓磊、曹汉鸣、陶晓云、王通、方峻、李小凤、曹晓松、蒋晓冬、邱培强、洪妍、李亮辉、高娟妮、曹小震、陈笑。由于译者水平有限，翻译工作中可能会有不准确的内容，如果读者在阅读过程中发现失误和遗漏之处，望多多包涵，并欢迎批评指正。敬请广大读者提供反馈意见，读者可以将意见发到 bmtao0807@gmail.com，我会仔细阅读读者发来的每一封邮件，以求进一步提高今后译著的质量。

译 者



作者简介

Michael McLaughlin 是爱达荷州杨百翰大学商业与通信学院计算机信息技术系的教授。他也是 McLaughlin Software, LLC 的创始人，同时活跃于犹他 Oracle 用户组。他还是其他 8 本 Oracle Press 图书的作者，例如 Oracle Database 11g & MySQL 5.6 开发手册、Oracle Database 11g PL/SQL 程序设计和 Oracle Database 11g PL/SQL 编程实战(清华大学出版社引进并出版)。

当 PL/SQL 还是 Oracle 6 的附加产品时，Michael 就开始编写 PL/SQL 程序。他也编写 C、C++、Java、Perl、PHP 和 Python 程序。

Michael 曾在 Oracle 公司工作过 8 年多，担任顾问、开发和支持的工作。在 Oracle 时，他领导了用于 Oracle Applications 11i(11.5.8 和 11.5.9)的直接路径 CRM 升级的版本工程，还领导了 PL/SQL 的向前兼容性测试，对象是 Oracle Applications 11i 和 Oracle Database 9i。他是 ATOMS 事务架构(U.S. Patents #7,206,805 和 #7,290,056)的发明人。这些专利属于 Oracle 公司。

在 Oracle 公司任职前，Michael 从 Oracle 6 开始，担任过 Oracle 开发人员、系统和业务分析师及 DBA。他的博客是 <http://blog.mclaughlinsoftware.com>。

Michael居住在爱达荷州东部，距离卡里布-塔基国家森林、大提顿国家公园和黄石国家公园两个小时车程以内。他喜欢和他的妻子及孩子们进行户外运动。

合著者简介

John Harper 是首席数据库工程师。他非常喜欢与数据仓库、商业智能和数据库工程师打交道。

John 的导师包括 Michael McLaughlin、Robert Freeman、Danette McGilvary，还有很多其他人，他们花费了大量的时间成为这个行业中的专家。他们的能力让 John 感到敬佩并倍受鼓舞，同时为能和他们联系在一起而感到幸运。

现在，John 有机会密切接触数据库安全方面的顶尖思想。他希望可以创作出一系列关于 Oracle 产品的出版物，例如 *Oracle Audit Vault and Database Firewall* 和 *Oracle Data Redaction*。

John 喜欢日本武术。在他的青少年时期，他学习了柔术、空手道、柔道和合气道。他喜欢合气道并希望有一天能够教授它。如果有时间的话，他还想要学习弓道。

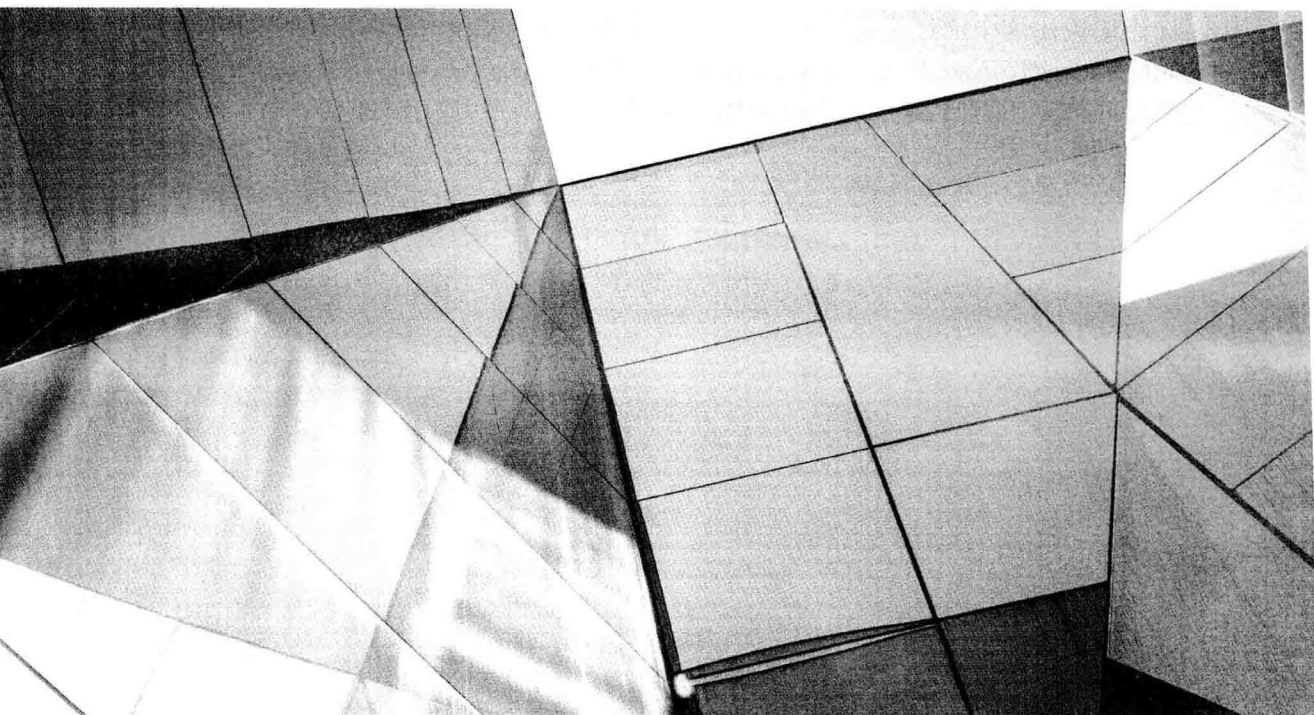
John 和他的妻子在犹他州的犹他郡北部生活了 23 年多。他们有一个收养的女儿，他们非常珍视和溺爱她。他在过去的 14 年里一直和数据库打交道，专攻 Oracle 管理员、数据库架构、数据库编程、数据库安全和信息质量。

技术编辑简介

Joseph McLaughlin 是犹他州盐湖城的 Deseret Book 出版社的 iPhone 和 Ruby Web 开发人员。他拥有大量的使用 Oracle、MySQL 和 PostgreSQL 的后端数据库开发经验。他最喜欢的开发语言是 Objective-C 和 Ruby。

Joseph 刚从爱达荷州杨百翰大学毕业，并获得了计算机信息技术专业学位。作为大学生和独立的顾问，Joseph 设计、开发并部署了 4 个移动应用程序，可以应用于 iPhone 或 iPod Touch。

除了编程，Joseph 喜欢篮球和观看波士顿红袜队赢得比赛，尤其是当他们获得世界职业棒球大赛的胜利时。



致 谢

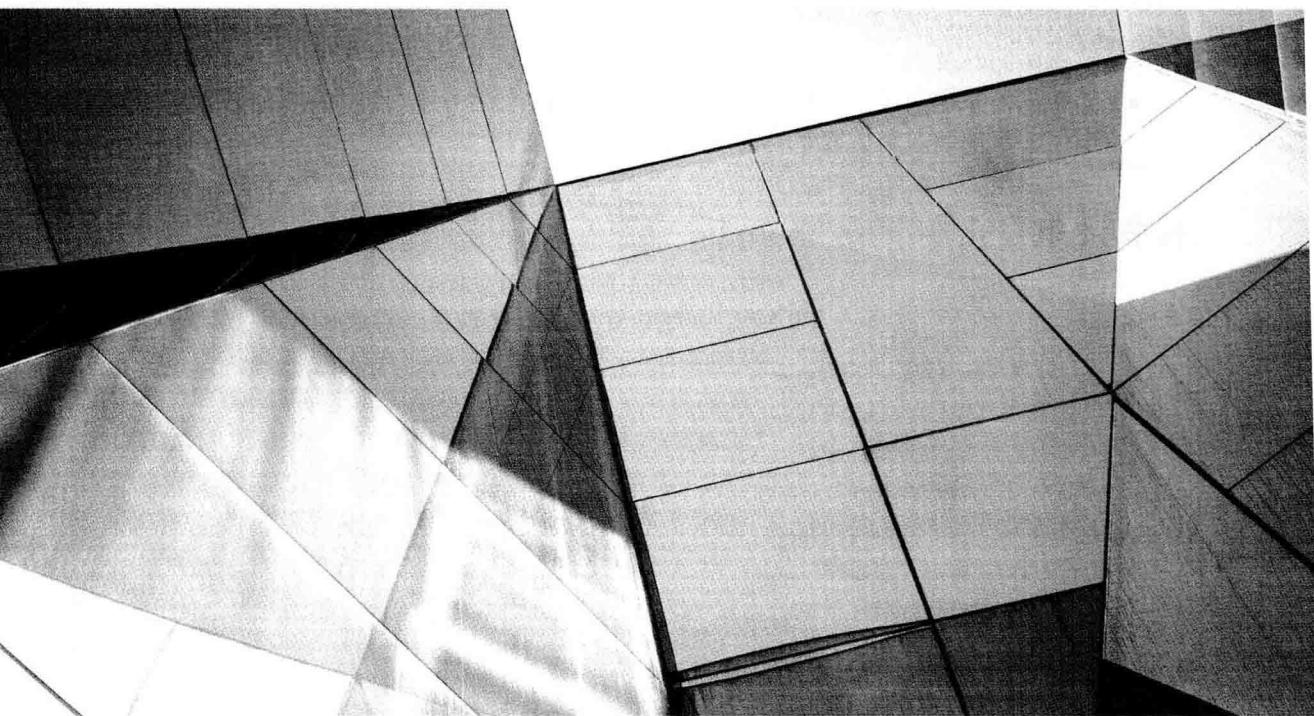
非常感谢 Paul Carlstroem、Amanda Russell、Harry Popli，以及帮助完成本书的制作团队。在制作部门有很多幕后英雄，因为他们隐藏在幕后。制作部门排版、校对并贡献他们的所有，确保本书名副其实，尽管我不知道他们的名字，但是他们应当因为他们一丝不苟的工作而获得感谢。

特别感谢文字编辑 Bill McManus，他为保持本书的一致性、条理性和组织性做出了卓越的成就。为精神上和项目上的支持特别感谢 Paul Carlstroem 和 Amanda Russell，因为对于我的成功，他们是非常重要的，特别是随项目走过了一年多的时间。感谢 Sheila Cepero，她管理 Oracle 出版者程序，她帮助度过 Oracle Database 12c beta 测试周期。同样感谢 Lynn Snyder 管理 Oracle Database 12c 程序。

感谢 John Harper，他贡献了附录 A 的基础部分并编写了附录 D。John 还为本书的剩余部分扮演了第二技术编辑的角色，他对细节的把握为本书的质量做出了重要的贡献。

感谢 Pablo Ribaldi 对附录 A 中数据管理的贡献。作为信息管理经理，他领导 LDS 教会的信息通信服务团队，从 DebTech International LLC 赢得了 Data Governance Best Practice Award。

感谢关注这个项目的学生和实验室助教，如Craig Christensen、Jeremy Heiner、Matthew Mason、Alan Pynes和Jordan Smith。还要感谢Kent Jackson阅读本书并提供了改进意见，以及感谢我的部门主管Steve Rigby，感谢他对本项目的支持。



前言

本书介绍如何使用 PL/SQL 编程语言。书中包含很多例子和技术，可以帮助创建稳健的以数据库为中心的应用程序。附录 A 介绍 Applications DBA 或者开发人员应该了解的基本信息，例如启动和停止 Oracle 数据库及侦听器、使用 SQL*Plus 作为命令行接口、使用 SQL Developer 作为免费的跨平台的 GUI 界面，以及 SQL 调优的技术。附录 B、C 和 D 介绍如何编写 SQL、使用 SQL 内置函数，以及使用 PL/SQL 内置包。其余附录介绍如何使用正则表达式函数、包装 PL/SQL 代码、对 PL/SQL 使用分层的配置文件，并了解保留字和关键字。

对于作者来说，前言通常是最后或最初撰写。与前面版本的策略不同，这次我选择在开始时写前言，这帮助我始终如一地按照计划完成本书。正如我在致谢中说过的，项目成员也帮助我理清了本书的内容，而且他们的才能是出版这一本好书的关键。

前言包含以下方面：

- “本书大纲”小节用一到两句话总结了每一章，值得快速浏览一下，以便对本书的组织结构有一个大概的了解。

- “词法”小节介绍本书中的变量命名规则的基本原理,并为调试代码提供推荐的、节省时间的技术。
- “数据模型和源代码下载”小节介绍示例的基本信息,并告知哪里可以找到创建和生成示例音像店数据库的代码。

本书大纲

本书分为三部分:“Oracle PL/SQL”、“PL/SQL 编程”以及“附录和术语表”。在本书的前两个部分中,每一章的每一个重要的小节都以“章节回顾”结尾,列出该小节中的关键点。而且,前两部分中的每一章都包含“精熟测验”,含有 10 道判断题和 5 道多选题,帮助你确认对本章覆盖的知识理解程度。答案在附录 I 中。

第 III 部分“附录和术语表”包含有关 Oracle Database 12c、SQL、SQL 内置函数、PL/SQL 内置包、正则表达式、包装 PL/SQL 代码、PL/SQL 分层的配置文件,以及保留字和关键字的入门知识。如提到过的,附录 I 中提供“精熟测验”的答案。术语表在最后一个附录之后。

第 I 部分: Oracle PL/SQL

- 第 1 章“Oracle PL/SQL 程序开发概览”介绍 PL/SQL 的历史和背景,并介绍 Oracle 开发架构。历史和背景小节介绍 SQL 如何成为主要接口和 PL/SQL 如何使用内置命令式编程语言扩展 SQL 的行为,启用对象关系型功能,允许 DBA 和开发人员利用 Oracle Database 12c 的能力。Oracle 开发架构小节介绍 SQL 接口如何作为一个交互式的调用命令行接口(CLI),以及两层或者 n 层模型如何与 Oracle Database 12c 一起使用。
- 第 2 章“新特性”介绍 Oracle Database 12c SQL 和 PL/SQL 新特性。该章假定你了解 Oracle Database 11g 特性。新的 SQL 特性包含非可见列和标识列、VARCHAR2 数据类型扩展过的长度、强化过的外联接操作。新的 PL/SQL 特性包含调用者权限结果缓存函数、PL/SQL 调用者白名单、新的错误堆栈管理特性、在 SQL WITH 子句中嵌入函数,在嵌入式的 SQL 语句中使用局部 PL/SQL 数据类型。
- 第 3 章“PL/SQL 基础知识”介绍并提供 PL/SQL 编程语言基础知识的示例。该章介绍 PL/SQL 块结构、块中变量的行为、基本标量和复合数据类型、控制结构、异常、批量操作、函数、过程、包、事务作用域和数据库触发器。在第 3 章中可以找到 PL/SQL 语法的所有基本元素的示例。如果你需要知道如何编写 PL/SQL 程序的基础知识,这也是一个很好的起点。
- 第 4 章“语言基本知识”介绍词汇单元(分隔符、标识符、字面值或注释)以及变量和数据类型。在第 4 章中,可以学习 PL/SQL 程序单元的基本构建块。还可以学习哪些数据类型可用及如何声明这些数据类型的变量。后面的章节会假定你知道哪些数据类型可用及如何在匿名和命名 PL/SQL 块中声明它们,这使得该章成

为很重要的章节。在深入研究 PL/SQL 语言核心特性之前，要先阅读或者研究第 4 章。

- 第 5 章“控制结构”介绍条件语句、迭代语句、游标结构和批语句。该章全面地介绍 IF 语句和循环。Oracle 实现 IF 语句或 CASE 语句来管理条件逻辑，用简单、FOR 和 WHILE 循环来管理迭代语句。关于循环的讨论介绍了守卫和标记值以及对动态标记值的保护。这一章介绍如何在循环中管理游标和如何管理批处理 DML 语句。
- 第 6 章“集合”介绍如何使用 SQL 变长数组和表集合以及 PL/SQL 关联数组(之前称为 PL/SQL 表或索引表)。该章通过讨论变长数组和表集合，介绍了如何使用属性数据类型(ADT)和用户自定义类型(UDT)。同时介绍了使用和处理 ADT 和 UDT 变量之间的区别。这一章还介绍了如何处理使用标量数据类型或者复合数据类型的 PL/SQL 专有关联数组，它可能是记录类型或者对象类型。该章最后介绍 Oracle 集合 API 的函数和过程，并提供使用这些函数和过程的示例。
- 第 7 章“错误管理”介绍如何在 PL/SQL 中使用异常。该章介绍异常类型和作用域、异常管理内置函数、用户自定义异常和异常堆栈函数。同时介绍如何找到并解决编写 PL/SQL 程序时发生的典型的错误。该章还介绍如何编写异常处理程序，用于管理意外出现的运行时异常。还可以学习到如何管理异常栈。

第 II 部分：PL/SQL 编程

- 第 8 章“函数和过程”介绍 PL/SQL 函数和过程的架构、事务作用域、函数选项和实现以及过程的实现。架构部分介绍按值传递和按引用传递的函数和过程的工作方式，包括如何为 Oracle Database 12c 中的独立函数和过程创建白名单。同时介绍 SQL 的位置、命名、混合和排除调用表示法。它还介绍用于定义函数的各种方式，例如确定性的、启用并行的、管道的和结果缓存的函数。它介绍如何实现返回用户自定义类型的集合的对象表函数。该章还介绍如何编写递归和自治函数。
- 第 9 章“包”介绍如何操作和使用包。该章涉及了包结构、包规范和包主体。还将定义者权限和调用者权限模型进行比较，并介绍数据库目录如何管理包规范和包主体的状态和有效性。该章介绍了如何编写向前引用的占位程序以及如何重载函数和过程。还介绍了如何为包规范创建白名单。
- 第 10 章“大对象”介绍如何处理 BLOB、CLOB 和 NCLOB 内部管理数据类型和 BFILE 外部管理数据类型。该章介绍如何创建和处理内部管理的字符和二进制大型对象，以及如何处理外部管理的二进制文件。
- 第 11 章“对象类型”介绍如何处理对象类型。该章介绍如何声明、实现、过滤对象类型，还有如何实现 getter、setter 和对象比较函数。在介绍了这些基本信息之后，该章介绍在实现对象类型集合之前如何实现继承和多态。

- 第 12 章“触发器”介绍数据库触发器，然后介绍如何理解并实现各种类型的触发器，包括 DDL、DML、复合、INSTEAD OF 以及系统和数据库事件触发器，还介绍触发器约束。
- 第 13 章“动态 SQL”介绍动态 SQL 语句的基本信息。该章涉及本地动态 SQL(NDS) 和 dbms_sql 包。对动态 SQL 语句的 4 个方法都予以介绍，例如静态 DDL 和 DML 语句、动态 DML 语句、带静态 SELECT 列表的动态 SELECT 语句和带动态 SELECT 列表的动态 SELECT 语句。

第 III 部分：附录和术语表

- 附录 A “Oracle 数据库入门”介绍 Oracle Database 12c 架构、如何启动和停止 Oracle Database 12c 服务器和 Oracle 侦听器、多版本并发控制、定义者权限和调用者权限、SQL 交互和批处理、数据库管理、SQL 调优和数据库跟踪。
- 附录 B “SQL 入门”介绍在 Oracle Database 12c 中使用 SQL。该附录包含 SQL 数据类型、DDL 语句、DML 语句、TCL 语句、SELECT 语句和 SET 语句。该附录还介绍如何及嵌套查询和如何使用持久对象类型。
- 附录 C “SQL 内置函数”提供代码完整的示例，介绍如何使用 Oracle Database 12c 的关键 SQL 内置函数。该附录包含字符函数、数据类型转换函数、日期时间函数、集合管理函数、集合运算符、数字函数、错误处理函数和杂项函数。
- 附录 D “PL/SQL 内置程序包和类型”介绍如何在 Oracle Database 12c 中使用 SQL。该附录介绍新的 PL/SQL 内置包并提供一些关键包的示例。
- 附录 E “正则表达式入门”介绍在 SQL 和 PL/SQL 中如何使用正则表达式。
- 附录 F “包装 PL/SQL 代码入门”介绍如何使用 dbms_ddl 包的 create_wrapped 或 wrap 过程。
- 附录 G “PL/SQL 分层的配置文件入门”介绍如何使用 PL/SQL Hierarchical Profiler。该附录展示如何配置这个模式、收集分析数据、理解 Profiler 输出，以及使用 plshprof 命令行工具。
- 附录 H “PL/SQL 保留字和关键字”指出哪些是 Oracle Database 12c 中的保留字和关键字。
- 附录 I “精熟测试答案”提供各章中“精熟测试”小节的答案。
- 术语表提供本书中出现的概念的定义。

词法

编写程序有很多方式，通常编程语言之间是有区别的。SQL 和 PL/SQL 代码分享这一共性：它们是不同的语言，需要不同的方法。3 个小节分别介绍：SQL 词法、PL/SQL 存储程序以及语法中的其他约定。

SQL 词法

对于 SQL 语句,我的建议是将关键字排列在左边。那意味着将 SELECT 列表逗号和 WHERE 子句逻辑 AND [NOT]或者 OR [NOT]放在左边,因为它允许你视读(sight read)代码错误。这个建议很容易理解,但是对于如何编写联接语法,我的建议更复杂一些,因为你可能会编写使用 ANSI SQL-89 或 ANSI SQL-92 的联接。ANSI SQL-89 允许你使用逗号分隔列表来组织表,而 ANSI SQL-92 允许你使用关键字指定联接类型。

下面是我对于联接语法的建议:

- 总是使用表的别名,因为它们保证当 SELECT 列表能够返回两个或者多个同名列的时候,不会发生歧义列的错误。当联接那些共享相同列名的表时,会发生这种情况。当编写单一表查询的时候,这也是个很好的操作,因为你随后可能会通过联接添加另一个表。附录 B 包含了 SELECT 语句和支持这个建议的语法。
- 当使用 ANSI SQL-89 和逗号分隔表的时候,将每个表放到它自己的行上,并将分隔列放到左边,例如 SELECT 列表列。这帮助你视读你的程序。这个不能应用于附录 B 中提到的多表 UPDATE 和 DELETE 语句,你可以参考那些章中的示例。
- 当使用 ANSI SQL-92 的时候,使用 ON 子句或者 USING 子句将联接条件放到 FROM 子句中。对于含有 ON 或者 USING 子句的 FROM 子句,有两个通用的方法看起来对多数开发人员是最合适的。在小的(两个或者最多三个)表联接中,在联接同一行之后加上 ON 或 USING 子句。在大的联接中(三个或者更多),在联接语句的下一行添加 ON 或 USING 子句。当联接涉及多个列的时候,左对齐逻辑 AND [NOT]或者 OR [NOT]语法允许你视读你的代码。这和本小节开始的时候我对 WHERE 子句提出的建议是相同的,通常它确实很有用。
- ANSI SQL-92允许你使用充分描述的关键字或者只使用被需要的关键字。尽管我们中的多数人想要输入最低限度的命令,但最终我们的代码会转给支持人员,而且它的明确性可以帮助避免琐碎的错误报告。因此,考虑使用INNER JOIN代替JOIN、用LEFT OUTER JOIN或RIGHT OUTER JOIN代替LEFT JOIN和RIGHT JOIN、用FULL OUT JOIN代替FULL JOIN。我在本书中缩短了语法,因为页宽规定命令行仅限70个字符(或者可以要求缩小字体,但是会影响阅读)。

接下来我要分享未遵循语法建议的那些体验。这些建议是 1985 年我在 IBM 圣特雷莎实验室(现在的 IBM 硅谷实验室)的导师教我如何编写 SQL(实际上是 SQL/DS [Structured Query Language/Data System])的时候告诉我的。他告诉我将逗号放在左边可以节省我搜索漏掉的逗号的时间。我忽略了这个建议,并将它们放到了右边,最终在意识到他是对的之前浪费了好几个月的时间。他在那一周总是对我重复一个准则:“好的编程遵循简单的原则。”

现在在学校,我每个学期都强调这个建议。有些学生接受并使用它,而有一些则不然。那些不接受它的学生在课程中自始至终与语法较劲,因为他们总是尝试找到 SQL 语句中漏掉的逗号或者组件。SQL 不是容易学习的东西,因为它要求创建数据的空间图,

这项技能不是所有的开发人员都能立刻掌握的。有时它需要相当多的时间来理清关系型数据库中数据的关系。练习会让这变得简单起来,前提是你努力维护语句的明确性、方法的一致性以及坚持使用可移植 SQL 语法。

关于工具

本书关注在命令行中编写 SQL,因为那是在 C++、C#、Java 或者 PHP 程序中的工作方式,但是 CASE(Computer-Aided Software Engineering)工具很好。它帮助你发现语法和可能性,假如你不是将它们作为拐杖来使用的话。

最好的开发人员不是那些口若悬河并使用所有煽情的口号来推销自己的商业用户。最好的开发人员是那些知道哪种技术真的能提供最好的解决方案来解决业务问题的人们。

那些应用好的开发技术的人当他们限定自己仅使用 CASE 工具的时候,他们并不是专属社团的成员。之所以这么说,是因为 CASE 工具通常只通过拖放界面来解决一般问题。那些主张 NoSQL 解决方案的人们基本上都是那些永远不知道如何使用数据库或者数据库如何满足日常事务业务需求的人。

简单来说,使用工具进行学习,但是不要成为它的奴隶。要一直询问为什么可以成功和如何能做得更好。如果你这样做了,那么就会发现CASE工具对完成工作是一个福音,而不是一个潜在的绊脚石(正如在过去一些年中发现的那些案例一样)。

PL/SQL 存储程序

PL/SQL是一个非常成熟的编程语言。它允许你编写存放在数据库中的程序,用于作为一个完整的事务管理SQL语句的集合。

变量命名约定在一些组织中是有争议的,因为很多开发人员相信变量应当是语义上有意义的。对于命名约定的争论是约定(例如前缀)会降低代码的可读性。这个争论简单来说是想法的冲突。双方都有优点,总有要选择一个比另一个更有逻辑的情况。我的观点是,关键是在提供有意义的变量名和为公司或企业增加稳定性这两者之间找到平衡点。

在本书中,我尽量保持一致并使用前缀。在有些地方,我选择变量名中语义明确(例如第 2 章中的 Oracle 会话或者绑定变量:whom)。我相信使用前缀会增加代码的可读性,我建议使用表 0-1 中的前缀。

某些高级变量数据类型(即复合变量)同时需要前缀和后缀。后缀表明复合变量的类型。这些要求对 Oracle 数据库都是唯一的。表 0-2 展示我推荐的 Oracle 复合变量类型的后缀(带有前导下划线)。表 0-2 展示后缀的长名和短名版本。

为复合数据类型使用后缀是普遍接受的做法,因为它们是 UDT。然而,在 PL/SQL 编程语言中它不是规则或者要求。

PL/SQL是一种带有声明、执行和异常块的强类型语言。块程序使用关键字开始和结束程序单元,与C++、C#、Java和PHP中使用花括号相反。和在GeSHi(Generic Syntax Highlighter)库中找到的一样,PL/SQL块关键字都是大写字母,我已经在本书中采用那个约定。

表 0-1 PL/SQL 变量前缀

前 缀	示 例	描 述
cv	cv_input_var	表示游标参数变量。这些是 PL/SQL 存储程序中游标的按值传递输入参数
lv	lv_target_var	表示 PL/SQL 存储程序中定义的局部变量
pv	pv_exchange_var	表示PL/SQL存储函数和过程的参数。它们不是专用的输入参数，因为PL/SQL支持存储函数和过程中的输入和输出参数
sv	sv_global_var	表示会话变量。它们在客户端连接到数据库存续期间作为全局变量。Oracle 允许在变量名前使用冒号(sv_global_var)来在匿名块之间共享变量值。这也称为绑定变量

表 0-2 PL/SQL 变量后缀

后 缀		描 述
长	短	
_ATABLE _AARRAY	_ATAB _AA	_ATABLE、_AARRAY、_ATAB 和 _AA 用于描述 PL/SQL 中的关联数组。我的首选是 _ATABLE 或 _ATAB 后缀，因为其他后缀不直观而且在代码中需要说明
_CURSOR	_CUR _C	_CURSOR、_CUR 和 _C 用于描述基于 PL/SQL 中局部声明块或者包规范内定义的游标结构的变量。我的首选是 _CURSOR 或者 _C 后缀
_EXCEPTION	_EXCEPT _EX _E	_EXCEPTION、_EXCEPT、_EX 和 _E 用于描述 PL/SQL 中用户自定义的异常。我的首选是 _EXCEPTION 或者 _E 后缀
_OBJECT	_OBJ _O	_OBJECT、_OBJ 和 _O 用于描述 SQL 和 PL/SQL 中的用户自定义类型 (UDT)。对象类型可以像 PL/SQL RECORD 数据类型一样，是记录数据结构。它们不同是因为它们是模式级别 SQL UDT 而不是专用 PL/SQL UDT。对象类型也可以是可实例化对象，例如 C++、C# 和 Java 类。我的首选是 _OBJECT 或者 _O 后缀
_NTABLE _TABLE	_NTAB _TAB	_NTABLE、_TABLE、_NTAB 或者 _TAB 用于描述嵌套表，它们是 SQL 和 PL/SQL 中的集合类型。它们像列表一样，因为集合中元素数量是没有上限的。我的首选是 _TABLE 或者 _TAB 后缀，因为嵌套表是集合，像其他编程语言中的列表一样
_RECORD	_REC _R	_RECORD、_REC 和 _R 专门用于描述 PL/SQL 的 UDT。它们是记录数据结构的 PL/SQL 实现。它们可以是 PL/SQL 集合的元素但不是 SQL 集合的元素。我的首选是 _RECORD 或者 _R 后缀，因为它们是完整描述或者简写，但是很多开发人员选择 _REC

(续表)

后 缀		描 述
长	短	
<code>_TYPE</code>	<code>_T</code>	<code>_TYPE</code> 和 <code>_T</code> 用于描述 UDT, 这和第 4 章中介绍过的普通标量数据类型的子类型一样。两个对我来说都很合适, 但是 <code>_TYPE</code> 在代码库中更常使用
<code>_VARRAY</code>	<code>_VARR</code> <code>_VA</code>	<code>_VARRAY</code> 、 <code>_VARR</code> 和 <code>_VA</code> 用于描述 <code>VARRAY</code> (对于这个 Oracle 数据类型, 我的记忆是虚数组)。 <code>VARRAY</code> 集合是更像编程语言中的标准数组的集合, 因为它有最大值而且必须有顺序索引值。可以用于定义 SQL 和 PL/SQL 集合。我的首选是 <code>VARRAY</code> 或者 <code>_VA</code> 后缀, 因为 <code>_VARR</code> 太类似普通变量简写

其他约定

有时代码块需要明确性。在 Oracle 的 PL/SQL 和 SQL 示例中一直提供行号, 因为它们是 SQL*Plus 环境的显示特性。

本书的文本约定包含醒目、斜体和分隔语法。它们在表 0-3 中介绍。

希望这些约定可以使本书更容易阅读。你也会在本书中的阴影区域中找到补充信息。

表 0-3 文本约定

约 定	意 义
粗体字	关注示例程序中的特定代码行
斜体字	关注新单词或者概念
大写形式	表示 SQL 和 PL/SQL 中使用的关键字, 以及 SQL 内置函数名
小写形式	表示用户自定义表、视图、列、函数、过程、包和类型的名称
<code>[]</code>	表示可选择的语法并在原型中显示
<code>{}</code>	组合选项列表, 使用单个竖线(<code> </code>)分隔
<code> </code>	表示选项列表间的逻辑 OR 运算符
<code>...</code>	表示内容重复或者为了节省空间被移除的部分

数据模型和源代码下载

这个数据模型是一个小的音像店。创建并生成 Oracle 数据模型的源代码可以在本书的支持网站中找到:

www.OraclePressBook.com