O'REILLY®

6th Edition

Oracle PL/SQL Programming

COVERS VERSIONS THROUGH ORACLE DATABASE 12C

Steven Feuerstein
with Bill Pribyl

O'REILLY®

Oracle PL/SQL Programming

Considered the best Oracle PL/SQL programming guide by the Oracle community, this definitive guide is precisely what you need to make the most of Oracle's powerful procedural language. The sixth edition describes the features and capabilities of PL/SQL up through Oracle Database 12*c* Release 1.

Hundreds of thousands of PL/SQL developers have benefitted from this book over the last twenty years; this edition continues that tradition. With extensive code examples and a lively sense of humor, this book explains language fundamentals, explores advanced coding techniques, and offers best practices to help you solve real-world problems.

- Get PL/SQL programs up and running quickly, with clear instructions for executing, tracing, testing, debugging, and managing code
- Learn new 12.1 features, including the ACCESSIBLE_BY clause, WITH FUNCTION and UDF pragma, and BEQUEATH CURRENT_USER for views
- Take advantage of extensive code samples, from easy-to-follow examples to reusable packaged utilities
- Optimize PL/SQL performance with features like the function result cache and Oracle utilities such as PL/Scope and the PL/SQL hierarchical profiler
- Build modular, easy-to-maintain PL/SQL applications using packages, procedures, functions, and triggers

Steven Feuerstein is considered one of the world's leading experts on the Oracle PL/SQL language. Steven's been writing about—and training developers on—PL/SQL since 1990. He's the author of ten books on the subject, including *Oracle PL/SQL Best Practices*.

Bill Pribyl is an author, teacher, and software consultant. He's the author of *Learning Oracle PL/SQL* and coauthor of *Oracle PL/SQL Programming* and *Oracle PL/SQL Language Pocket Reference*.

DATABASES

US $69.99 CAN $73.99

ISBN: 978-1-449-32445-2

Twitter: @oreillymedia
facebook.com/oreilly

SIXTH EDITION

Oracle PL/SQL Programming

Steven Feuerstein
with Bill Pribyl

Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo

O'REILLY®

Oracle PL/SQL Programming, Sixth Edition
by Steven Feuerstein with Bill Pribyl

Printed in the United States of America.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (*http://my.safaribooksonline.com*). For more information, contact our corporate/institutional sales department: 800-998-9938 or *corporate@oreilly.com*.

Editor: Ann Spencer
Production Editor: Nicole Shelby
Copyeditor: Rachel Monaghan
Proofreader: Rachel Head
Indexer: Ellen Troutman
Cover Designer: Randy Comer
Interior Designer: David Futato
Illustrator: Rebecca Demarest

September 1995: First Edition

September 1997: Second Edition

September 2002: Third Edition

August 2005: Fourth Edition

September 2009: Fifth Edition

January 2014: Sixth Edition

Revision History for the Sixth Edition:

2014-01-22: First release

See *http://oreilly.com/catalog/errata.csp?isbn=9781449324452* for release details.

ISBN: 978-1-449-32445-2

[QG]

To my wife, Veva Silva, whose intelligence, strength, beauty, and art have greatly enriched my life.

—Steven Feuerstein

To my wife, Norma. Still melting my heart after a quarter century.

—Bill Pribyl

Table of Contents

Part VI. Advanced PL/SQL Topics

Preface

Millions of application developers and database administrators around the world use software provided by Oracle Corporation to build complex systems that manage vast quantities of data. At the heart of much of Oracle's software is PL/SQL—a programming language that provides procedural extensions to Oracle's version of SQL (Structured Query Language) and serves as the programming language within the Oracle Developer toolset (most notably Forms Developer and Reports Developer).

PL/SQL figures prominently as an enabling technology in almost every new product released by Oracle Corporation. Software professionals use PL/SQL to perform many kinds of programming functions, including:

- Implementing crucial business rules in the Oracle Server with PL/SQL-based stored procedures and database triggers
- Generating and managing XML documents entirely within the database
- Linking web pages to an Oracle database
- Implementing and automating database administration tasks—from establishing row-level security to managing rollback segments within PL/SQL programs

PL/SQL was modeled after Ada,[1] a programming language designed for the US Department of Defense. Ada is a high-level language that emphasizes data abstraction, information hiding, and other key elements of modern design strategies. As a result of this very smart design decision by Oracle, PL/SQL is a powerful language that incorporates many of the most advanced elements of procedural languages, including:

- A full range of datatypes from number to string, and including complex data structures such as records (which are similar to rows in a relational table), collections

1. The language was named "Ada" in honor of Ada Lovelace, a mathematician who is regarded by many to have been the world's first computer programmer.

(which are Oracle's version of arrays), and XMLType (for managing XML documents in Oracle and through PL/SQL)

- An explicit and highly readable block structure that makes it easy to enhance and maintain PL/SQL applications
- Conditional, iterative, and sequential control statements, including a CASE statement and three different kinds of loops
- Exception handlers for use in event-based error handling
- Named, reusable code elements such as functions, procedures, triggers, object types (akin to object-oriented classes), and packages (collections of related programs and variables)

PL/SQL is integrated tightly into Oracle's SQL language: you can execute SQL statements directly from your procedural program without having to rely on any kind of intermediate application programming interface (API) such as Java Database Connectivity (JDBC) or Open Database Connectivity (ODBC). Conversely, you can also call your own PL/SQL functions from within a SQL statement.

Oracle developers who want to be successful in the 21st century must learn to use PL/SQL to full advantage. This is a two-step process. First, you must become familiar with and learn how to use the language's ever-expanding set of features; and second, after gaining competence in the individual features, you must learn how to put these constructs together to build complex applications.

For these reasons and more, Oracle developers need a solid, comprehensive resource for the base PL/SQL language. You need to know the basic building blocks of PL/SQL, but you also need to learn by example so that you can avoid some of the trial and error. As with any programming language, PL/SQL has a right way and many wrong ways (or at least "not as right" ways) to handle just about any task. It is my hope that this book will help you learn how to use the PL/SQL language in the most effective and efficient way possible.

Objectives of This Book

What, specifically, will this book help you do?

Take full advantage of PL/SQL
: Oracle's reference manuals may describe all the features of the PL/SQL language, but they don't tell you how to apply the technology. In fact, in some cases, you'll be lucky to even understand how to use a given feature after you've made your way through the railroad diagrams. Books and training courses tend to cover the same standard topics in the same limited way. In this book, I'll venture beyond the basics to the far reaches of the language, finding the nonstandard ways that a particular feature can be tweaked to achieve a desired result.

Use PL/SQL to solve your problems
: You don't spend your days and nights writing PL/SQL modules so that you can rise to a higher plane of existence. You use PL/SQL to solve problems for your company or your customers. In this book, I try hard to help you tackle real-world problems, the kinds of issues developers face on a daily basis (at least those problems that can be solved with mere software). To do this, I've packed the book with examples—not just small code fragments, but substantial application components that you can apply immediately to your own situations. There is a good deal of code in the book itself, and much more on the accompanying website. In a number of cases, I use the code examples to guide you through the analytical process needed to come up with a solution. In this way you'll see, in the most concrete terms, how to apply PL/SQL features and undocumented applications of those features to a particular situation.

Write efficient, maintainable code
: PL/SQL and the rest of the Oracle products offer the potential for incredible development productivity. If you aren't careful, however, this capability will simply let you dig yourself into a deeper, darker hole than you've ever found yourself in before. I would consider this book a failure if it only helped programmers write more code in less time; I want to help you develop the skills and techniques to build applications that readily adapt to change and that are easily understood and maintained. I want to teach you to use comprehensive strategies and code architectures that allow you to apply PL/SQL in powerful, general ways to the problems you face.

Structure of This Book

Both the authors and O'Reilly Media are committed to providing comprehensive, useful coverage of PL/SQL over the life of the language. This sixth edition of *Oracle PL/SQL Programming* describes the features and capabilities of PL/SQL up through Oracle Database 12*c* Release 1. I assume for this edition that Oracle Database 12*c* is the baseline PL/SQL version. However, where appropriate, I reference specific features introduced (or only available) in other, earlier versions. For a list of the main characteristics of the various releases, see the section "About PL/SQL Versions" on page 11 in Chapter 1.

PL/SQL has improved dramatically since the release of version 1.0 in the Oracle 6 database so many years ago. *Oracle PL/SQL Programming* has also undergone a series of major transformations to keep up with PL/SQL and provide ever-improving coverage of its features.

The biggest change in the sixth edition is its comprehensive coverage of all new PL/SQL features in Oracle Database 12*c* Release 1. The major features are summarized in Chapter 1, along with references to the chapters where these features are discussed in detail.

I am very happy with the results and hope that you will be too. There is more information than ever before, but I think we managed to present it without losing the sense of humor and conversational tone that readers have told me for years make the book readable, understandable, and highly useful.

One comment regarding the "voice" behind the text. You may notice that in some parts of this book we use the word *we*, and in others *I*. One characteristic of this book (and one for which readers have expressed appreciation) is the personal voice that's inseparable from the text. Consequently, even with the addition of coauthors to the book (and, in the third, fourth, and fifth editions, significant contributions from several other people), we've decided to maintain the use of *I* when an author speaks in his own voice.

Rather than leave you guessing as to which lead author is represented by the *I* in a given chapter, we thought we'd offer this quick guide for the curious; you'll find additional discussion of our contributors in the Acknowledgments.

Chapter	Author	Chapter	Author
Preface	Steven	15	Steven
1	Steven	16	Steven
2	Bill and Steven	17	Steven
3	Steven and Bill	18	Steven
4	Steven, Chip, and Jonathan	19	Darryl and Steven
5	Steven and Bill	20	Steven
6	Steven	21	Steven and Adrian
7	Chip, Jonathan, and Steven	22	Bill and Steven
8	Chip, Jonathan, and Steven	23	Arup
9	Chip, Jonathan, and Steven	24	Bill, Steven, and Chip
10	Chip, Jonathan, and Steven	25	Ron
11	Steven	26	Bill and Steven
12	Steven and Bill	27	Bill and Steven
13	Chip and Jonathan	28	Bill and Steven
14	Steven		

About the Contents

The sixth edition of *Oracle PL/SQL Programming* is divided into six parts:

Part I

I start from the very beginning in Chapter 1: where did PL/SQL come from? What is it good for? I offer a very quick review of some of the main features of the PL/SQL language. Chapter 2 is designed to help you get PL/SQL programs up and running as quickly as possible: it contains clear, straightforward instructions for executing PL/SQL code in SQL*Plus and a few other common environments.

Chapter 3 reviews fundamentals of the PL/SQL language: what makes a PL/SQL statement, an introduction to the block structure, how to write comments in PL/SQL, and so on.

Part II

Chapter 4 through Chapter 6 explore conditional (IF and CASE) and sequential (GOTO and NULL) control statements, loops and the CONTINUE statement, and exception handling in the PL/SQL language. This section of the book will teach you to construct blocks of code that correlate to the complex requirements of your applications.

Part III

Just about every program you write will manipulate data, and much of that data will be local to (defined in) your PL/SQL procedure or function. Chapter 7 through Chapter 13 concentrate on the various types of program data you can define in PL/SQL, such as numbers, strings, dates, timestamps, records, and collections. You will learn about the new datatypes introduced in Oracle Database 11*g* (SIMPLE_INTEGER, SIMPLE_FLOAT, and SIMPLE_DOUBLE), as well as the many binary, date, and timestamp types introduced in other recent releases. These chapters also cover the various built-in functions provided by Oracle that allow you to manipulate and modify data.

Part IV

Chapter 14 through Chapter 16 address one of the central elements of PL/SQL code construction: the connection to the underlying database, which takes place through the SQL language. These chapters show you how to define transactions that update, insert, merge, and delete tables in the database; how to query information from the database for processing in a PL/SQL program; and how to execute SQL statements dynamically, using native dynamic SQL (NDS).

Part V

This is where it all comes together. You know about declaring and working with variables, and you're an expert in error handling and loop construction. Now, in Chapter 17 through Chapter 22, you'll learn about the building blocks of applications, which include procedures, functions, packages, and triggers, and how to move information into and out of PL/SQL programs. Chapter 20 discusses managing your PL/SQL code base, including testing and debugging programs and managing dependencies; it also provides an overview of the edition-based redefinition capability introduced in Oracle Database 11*g* Release 2. Chapter 21 focuses on how you can use a variety of tools and techniques to get the best performance out of your PL/SQL programs. Chapter 22 covers I/O techniques for PL/SQL, from DBMS_OUTPUT (writing output to the screen) and UTL_FILE (reading and writing files) to UTL_MAIL (sending mail) and UTL_HTTP (retrieving data from a web page).

Part VI
: A language as mature and rich as PL/SQL is full of features that you may not use on a day-to-day basis, but that may make the crucial difference between success and failure. Chapter 23 explores the security-related challenges we face as we build PL/SQL programs. Chapter 24 contains an exploration of the PL/SQL architecture, including PL/SQL's use of memory. Chapter 25 provides guidance for PL/SQL developers who need to address issues of globalization and localization. Chapter 26 offers a guide to the object-oriented features of Oracle (object types and object views).

Appendix A through Appendix C summarize the details of regular expression syntax and number and date formats.

The chapters on invoking Java and C code from PL/SQL applications, which were part of the hardcopy fourth edition, have been moved to the book's website.

If you are new to PL/SQL, reading this book from beginning to end should improve your PL/SQL skills and deepen your understanding of the language. If you're already a proficient PL/SQL programmer, you'll probably want to dip into the appropriate sections to extract particular techniques for immediate application. Whether you use this book as a teaching guide or as a reference, I hope it will help you use PL/SQL effectively.

What This Book Does Not Cover

As long as this book is, it doesn't contain everything. The Oracle environment is huge and complex, and in this book we've focused our attention on the core PL/SQL language itself. The following topics are therefore outside the scope of this book and are not covered, except in an occasional and peripheral fashion:

The SQL language
: I assume that you already have a working knowledge of the SQL language, and that you know how to write SELECTs, UPDATEs, INSERTs, MERGEs, and DELETEs.

Administration of Oracle databases
: While database administrators (DBAs) can use this book to learn how to write the PL/SQL needed to build and maintain databases, this book does not explore all the nuances of the Data Definition Language (DDL) of Oracle's SQL.

Application and database tuning
: I don't cover detailed tuning issues in this book, although Chapter 21 does discuss the many tools and techniques that will help you to optimize the performance of your PL/SQL programs.

Oracle tool-specific technologies independent of PL/SQL
: This book does not attempt to show you how to build applications in a tool like Oracle's Forms Developer, even though the implementation language is PL/SQL. I

have chosen to focus on core language capabilities, centered on what you can do with PL/SQL from within the database. However, almost everything covered in this book is applicable to PL/SQL inside Forms Developer and Reports Developer.

Conventions Used in This Book

The following conventions are used in this book:

Italic
: Used for file and directory names and for emphasis when introducing a new term. In the text, it is also used to indicate a user-replaceable element.

`Constant width`
: Used for code examples.

`Constant width bold`
: Indicates user input in examples showing an interaction. Also, in some code examples, highlights the statements being discussed.

`Constant width italic`
: In some code examples, indicates an element (e.g., a parameter) that you supply.

UPPERCASE
: In code examples, generally indicates PL/SQL keywords or certain identifiers used by Oracle Corporation as built-in function and package names.

lowercase
: In code examples, generally indicates user-defined items such as variables, parameters, etc.

Punctuation
: In code examples, enter exactly as shown.

Indentation
: In code examples, helps to show structure but is not required.

--
: In code examples, a double hyphen begins a single-line comment that extends to the end of a line.

/* and */
: In code examples, these characters delimit a multiline comment that can extend from one line to another.

.
: In code examples and related discussions, a dot qualifies a reference by separating an object name from a component name. For example, dot notation is used to select fields in a record and to specify declarations within a package.

[ ]
: In syntax descriptions, square brackets enclose optional items.

{ }
: In syntax descriptions, curly brackets enclose a set of items from which you must choose only one.

|
: In syntax descriptions, a vertical bar separates the items enclosed in curly brackets, as in {TRUE | FALSE}.

...
: In syntax descriptions, ellipses indicate repeating elements. An ellipsis also shows that statements or clauses irrelevant to the discussion were left out.

Indicates a tip, suggestion, or general note. For example, I'll tell you if a certain setting is version specific.

Indicates a warning or caution. For example, I'll tell you if a certain setting has some kind of negative impact on the system.

Which Platform or Version?

In general, all the discussions and examples in this book apply regardless of the machine and/or operating system you are using. In those cases in which a feature is in any way version-dependent—for example, if you can use it only in Oracle Database 11*g* (or in a specific release, such as Oracle Database 11*g* Release 2)—I note that in the text.

There are many versions of PL/SQL, and you may find that you need to use multiple versions in your development work. Chapter 1 describes the various versions of PL/SQL and what you should know about them; see "About PL/SQL Versions" on page 11.

About the Code

All of the code referenced in this book is available from *http://oreil.ly/oracle-plsql-sixth*. You will also find the contents of some of the chapters from earlier editions that we removed or condensed in the different editions of the book. These may be especially helpful to readers who are running older versions of Oracle.

Information about all of Steven's books and accompanying resources can be found at *http://www.stevenfeuerstein.com*. You might also want to visit PL/SQL Obsession (Steven Feuerstein's PL/SQL portal) at ,, where you will find training materials, code downloads, and more.

To find a particlar example on the book's website, look for the filename cited in the text. For many examples, you will find filenames in the following form provided as a comment at the beginning of the example shown in the book, as illustrated here:

```
/* File on web: fullname.pkg */
```

If the code snippet in which you are interested does not have a "File on web" comment, then you should check the corresponding chapter code file.

A chapter code file contains all the code fragments and examples that do not merit their own file, but may prove useful to you for copy-and-paste operations. These files also contain the DDL statements to create tables and other objects on which the code may depend.

Each chapter code file is named *chNN_code.sql*, where *NN* is the number of the chapter.

Finally, the *hr_schema_install.sql* script will create the standard Oracle Human Resources demonstration tables, such as employees and departments. These tables are used in examples throughout the book.

Using Code Examples

Supplemental material (code examples, exercises, etc.) is available for download at *http://oreil.ly/oracle-plsql-sixth*.

This book is here to help you get your job done. In general, if example code is offered with this book, you may use it in your programs and documentation. You do not need to contact us for permission unless you're reproducing a significant portion of the code. For example, writing a program that uses several chunks of code from this book does not require permission. Selling or distributing a CD-ROM of examples from O'Reilly books does require permission. Answering a question by citing this book and quoting example code does not require permission. Incorporating a significant amount of example code from this book into your product's documentation does require permission.

We appreciate, but do not require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example: "*Oracle PL/SQL Programming, Sixth Edition* by Steven Feuerstein and Bill Pribyl (O'Reilly). Copyright 2014 Steven Feuerstein and Bill Pribyl, 978-1-4493-2445-2."

If you feel your use of code examples falls outside fair use or the permission given above, feel free to contact us at *permissions@oreilly.com*.

Safari® Books Online

Safari Books Online (*www.safaribooksonline.com*) is an on-demand digital library that delivers expert content in both book and video form from the world's leading authors in technology and business.

Technology professionals, software developers, web designers, and business and creative professionals use Safari Books Online as their primary resource for research, problem solving, learning, and certification training.

Safari Books Online offers a range of product mixes and pricing programs for organizations, government agencies, and individuals. Subscribers have access to thousands of books, training videos, and prepublication manuscripts in one fully searchable database from publishers like O'Reilly Media, Prentice Hall Professional, Addison-Wesley Professional, Microsoft Press, Sams, Que, Peachpit Press, Focal Press, Cisco Press, John Wiley & Sons, Syngress, Morgan Kaufmann, IBM Redbooks, Packt, Adobe Press, FT Press, Apress, Manning, New Riders, McGraw-Hill, Jones & Bartlett, Course Technology, and dozens more. For more information about Safari Books Online, please visit us online.

How to Contact Us

Please address comments and questions concerning this book to the publisher:

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
800-998-9938 (in the United States or Canada)
707-829-0515 (international or local)
707-829-0104 (fax)

We have a web page for this book, where we list errata, examples, and any additional information. You can access this page at *http://oreil.ly/oracle-plsql-sixth*.

To comment or ask technical questions about this book, send email to *bookquestions@oreilly.com*.

For more information about our books, courses, conferences, and news, see our website at *http://www.oreilly.com*.

Find us on Facebook: *http://facebook.com/oreilly*

Follow us on Twitter: *http://twitter.com/oreillymedia*

Watch us on YouTube: *http://www.youtube.com/oreillymedia*

Acknowledgments

Since *Oracle PL/SQL Programming* was first published in 1995, it has had a busy and productive history as the "go to" text on how to use the PL/SQL language. For that, I first of all express our appreciation to all our readers.

Maintaining *Oracle PL/SQL Programming* as an accurate, readable, and up-to-date reference to PL/SQL has been, from the start, a big (all right, I admit it—sometimes overwhelming) job; it certainly would not have been possible without the help of many Oracle specialists, friends, and family, and of course the incredible staff at O'Reilly Media.

You will find below rather detailed thank yous for those who helped pull together the sixth edition of *Oracle PL/SQL Programming*. Following that, you will find an acknowledgment of the many people who were instrumental in the earlier editions.

First and foremost, I thank those who contributed chapters and/or substantial content for the book; listed alphabetically, they are Adrian Billington, Chip Dawes, Jonathan Gennick, Ron Hardman, Darryl Hurley, and Arup Nanda. As of this edition, Chip Dawes has taken over responsibility for updating a half-dozen chapters. Jonathan Gennick wrote or substantially updated six chapters in past editions. Darryl Hurley has updated the fine chapter on database triggers for several editions and contributed insights on Oracle's internationalization features. Arup Nanda wrote the excellent chapter on security. Ron Hardman stepped up to the plate and wrote the chapter on globalization and localization. Adrian Billington provided excellent material in Chapter 21 on pipelined table functions.

I have invited each of our contributors to say a few words about themselves.

Adrian Billington is a consultant in database design, development, and performance tuning who has been working with Oracle databases since 1999. He is the man behind oracle-developer.net, a website full of SQL and PL/SQL features, utilities, and techniques for Oracle developers. Adrian is also an Oracle ACE and a member of the OakTable Network. He would like to thank James Padfield (Padders), Tom Kyte, and Steven Feuerstein for inspiring him to become a better developer during his impressionable early years as an Oracle professional. He lives in the UK with his wife Anji and three children, Georgia, Oliver, and Isabella.

Chip Dawes has been working with Oracle database technologies for over 20 years as a DBA, developer, teacher, and mentor. He is currently a manager at PwC, where he helps clients find value in their data. Chip lives in Chicagoland with his wife and children.

Jonathan Gennick is an experienced technology professional who is well known for his Oracle database expertise. His past experience encompasses both software development and database administration. As a developer, he has always enjoyed troubleshooting

and debugging. He loves working with SQL and PL/SQL, and is well known for his books and articles on those topics. In his off hours, Jonathan enjoys a rather low-tech approach to life. He serves actively in his local church, where you'll often find him engaged in scripture with a class of high school and sometimes college-age students, or even speaking from the pulpit. He is also an avid mountain biker, riding even in the dead of winter on very cool, studded bicycle tires imported from Finland. In his Oracle work, he is currently working his way through an exploration of Oracle SQL's built-in statistic functions.

Ron Hardman is founder of SettleOurEstate.com, an estate management solution built on Oracle Apex and the Oracle Cloud Database. He also consults around the world on Oracle Text and Oracle globalization technologies, and has been working with Oracle both as an employee and as a customer for more than 17 years. Ron enjoys writing about more than technology, releasing in 2010 his first historical fiction book, titled *Shadow Fox: Sons of Liberty*, which he cowrote with his daughter.

Darryl Hurley has been working with Oracle technology for more than 20 years, focusing on PL/SQL and DBA work. He lives in Richmond, British Columbia, with his lovely wife, Vanessa, and beautiful daughter, Bianca.

Arup Nanda has been an Oracle DBA since 1993, touching all aspects of the job—modeling, performance troubleshooting, PL/SQL coding, backups, disaster recovery, and more. He works as the principal database architect at a major corporation, has written about 500 articles, coauthored five books, and presented about 300 sessions at various conferences. He offers training sessions, engages in special projects like audits and DR, and writes about Oracle technology on his blog, arup.blogspot.com. He was *Oracle Magazine*'s 2003 DBA of the Year and 2012 Architect of the Year. He is an OCP, an OTN ACE Director, and a member of the OakTable Network. He lives in Connecticut with his wife, Anu, and son, Anish.

With such a big book, we needed lots of reviewers, especially because we asked them to test each code snippet and program in the book to keep to an absolute minimum the number of errors that made it into the printed version. I am deeply grateful to the following men and women of the Oracle PL/SQL world, who took time away from the rest of their lives to help make *Oracle PL/SQL Programming* the best book that it could be.

For this sixth edition, I first thank Valentin Nikotin, one of the best technical reviewers I've ever had for this book. He not only checked the Oracle Database 12*c* content for accuracy, but also helped me remove ambiguities and correct mistakes in several other key chapters that had not changed for this new edition. My other technical reviewers also had a big impact on the quality of this book. Many thanks, Patrick Barel and Arup Nanda!

Next, I offer my deep appreciation to Bryn Llewellyn, Oracle's PL/SQL Product Manager, and other members of the PL/SQL development team, most notably Charles Wetherell. Bryn provided crucial information and feedback on Oracle Database 12*c*'s new features and answered endless questions about various PL/SQL features with bottomless patience. There is no doubt that my understanding of PL/SQL and the accuracy with which I present it owe a great debt to Bryn.

From a non-Oracle perspective, grateful thoughts go to Joel Finkel, my favorite jack-of-all-trades, who makes up for the narrow specialization that simultaneously benefits and constrains *my* capabilities when it comes to computers and software.

Of course, that's just the technical content. Once I feel that we've got our treatment of PL/SQL "right," it's time for the remarkable crew at O'Reilly Media—led by my editor, Ann Spencer—to transform our many chapters and code examples into a book worthy of the O'Reilly imprint. Many thanks to Julie Steele, editor of the fifth edition; Nicole Shelby, production editor for this edition; Rob Romano, who created the excellent figures; and the rest of the crew. This was the first time that Ann edited my book. For all previous editions (that is, from 1994 to 2007), I had the great honor and pleasure of working with Debby Russell. Thanks, Debby, for your many years of commitment to making the entire O'Reilly Media Oracle series a big success!

And here are the many people we thanked (and continue to be grateful to) for their contributions to the first five editions of this book: Sohaib Abassi, Steve Adams, Don Bales, Cailein Barclay, Patrick Barel, John Beresniewicz, Tom Berthoff, Sunil Bhargava, Jennifer Blair, Dick Bolz, Bryan Boulton, Per Brondum, Boris Burshteyn, Eric Camplin, Joe Celko, Gary Cernosek, Barry Chase, Geoff Chester, Ivan Chong, Dan Clamage, Gray Clossman, Avery Cohen, Robert A. G. Cook, John Cordell, Steve Cosner, Tony Crawford, Daniel Cronk, Ervan Darnell, Lex de Haan, Thomas Dunbar, Bill Dwight, Steve Ehrlich, Larry Elkins, Bruce Epstein, Joel Finkel, R. James Forsythe, Mike Gangler, Beverly Gibson, Steve Gillis, Eric Givler, Rick Greenwald, Radhakrishna Hari, Gerard Hartgers, Donald Herkimer, Steve Hilker, Bill Hinman, Gabriel Hoffman, Chandrasekharan Iyer, Ken Jacobs, Hakan Jakobsson, Giovanni Jaramillo, Dwayne King, Marcel Kratochvil, Thomas Kurian, Tom Kyte, Ben Lindsey, Peter Linsley, Vadim Loevski, Leo Lok, Debra Luik, James Mallory, Raj Mattamal, Andrew McIlwrick, Nimish Mehta, Ari Mozes, Steve Muench, Jeff Muller, Kannan Muthukkaruppan, Dan Norris, Alex Nuijten, James Padfield, Rakesh Patel, Karen Peiser, Fred Polizo, Dave Posner, Patrick Pribyl, Nancy Priest, Shirish Puranik, Chris Racicot, Sri Rajan, Mark Richter, Chris Rimmer, Alex Romankevich, Bert Scalzo, Pete Schaffer, Drew Smith, Scott Sowers, JT Thomas, David Thompson, Edward Van Hatten, Peter Vasterd, Andre Vergison, Mark Vilrokx, Zona Walcott, Bill Watkins, Charles Wetherell, Edward Wiles, Daniel Wong, Solomon Yakobson, Ming Hui Yang, and Tony Ziemba.

My wife, Veva Silva, has supported me every step of the way through my career in the world of software, and I thank her deeply. My boys, Christopher Tavares Silva and Eli

Silva Feuerstein, have tolerated very well the diversion of my attention from them to PL/SQL (and when they were teenagers, positively delighted in this diversion). And finally I thank Chris and his lovely, smart, and creative wife, Lauren, for providing me with my first grandchild, Loey Lucille Silva.