



Expert Oracle Database Architecture

Techniques and Solutions for High
Performance and Productivity

—

Fourth Edition

—

Darl Kuhn
Thomas Kyte

Apress®

Expert Oracle Database Architecture

**Techniques and Solutions for High
Performance and Productivity**

Fourth Edition

**Darl Kuhn
Thomas Kyte**

Apress®

Expert Oracle Database Architecture: Techniques and Solutions for High Performance and Productivity

Darl Kuhn
Morrison, CO, USA

Thomas Kyte
Denver, CO, USA

ISBN-13 (pbk): 978-1-4842-7498-9
<https://doi.org/10.1007/978-1-4842-7499-6>

ISBN-13 (electronic): 978-1-4842-7499-6

Copyright © 2022 by Darl Kuhn and Thomas Kyte

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Jonathan Gennick
Development Editor: Laura Berendson
Coordinating Editor: Jill Balzano

Cover designed by eStudioCalamar

Cover image by Adrian Cuj on Unsplash (www.unsplash.com)

Distributed to the book trade worldwide by Apress Media, LLC, 1 New York Plaza, New York, NY 10004, U.S.A. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a Delaware corporation.

For information on translations, please e-mail booktranslations@springernature.com; for reprint, paperback, or audio rights, please e-mail bookpermissions@springernature.com.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at <http://www.apress.com/bulk-sales>.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at www.apress.com/9781484274989. For more detailed information, please visit <http://www.apress.com/source-code>.

Printed on acid-free paper

Table of Contents

About the Authors.....xv

Acknowledgmentsxvii

Introductionxix

Chapter 1: Developing Successful Oracle Applications 1

 My Approach 3

 The Black Box Approach 5

 How (and How Not) to Develop Database Applications 17

 Understanding Oracle Architecture..... 18

 Understanding Concurrency Control..... 35

 Multiversioning 41

 Database Independence? 51

 How Do I Make It Run Faster? 75

 The DBA-Developer Relationship..... 78

 Summary..... 80

Chapter 2: Architecture Overview 83

 Oracle Database Types..... 84

 Single-Tenant (Non-container) Database 85

 Single-Tenant (Non-container) RAC Database..... 87

 Multitenant Container Database 89

 Multitenant RAC Database..... 94

 Sharded Database 95

 The SGA and Background Processes 97

TABLE OF CONTENTS

Connecting to Oracle.....	98
Dedicated Server	98
Shared Server.....	100
Mechanics of Connecting over TCP/IP	102
Summary.....	107
Chapter 3: Files.....	109
Parameter Files.....	110
What Are Parameters?.....	112
Legacy init.ora Parameter Files	117
Server Parameter Files (SPFILES)	120
Converting to SPFILES	120
Setting Values in SPFILES	123
Unsetting Values in SPFILES	127
Creating PFILES from SPFILES	127
Fixing Corrupted SPFILES	128
Pluggable Databases.....	129
Parameter File Wrap-Up	130
Trace Files.....	131
Requested Trace Files.....	133
Trace Files Generated in Response to Internal Errors	136
Trace File Wrap-Up	140
Alert Log File	141
Datafiles	142
A Brief Review of File System Mechanisms	144
The Storage Hierarchy in an Oracle Database.....	146
Temp Files.....	152
Control Files	155
Redo Log Files.....	155
Online Redo Log	157
Archived Redo Log.....	160

Password Files	162
Change Tracking File	165
Flashback Logs	167
Flashback Database	167
Fast Recovery Area (FRA)	169
Data Pump Files	170
Flat Files	173
Generating a CSV File	173
Generating HTML	174
Generating a JSON File	175
Summary	175
Chapter 4: Memory Structures	177
The Process Global Area and User Global Area	178
Manual PGA Memory Management	179
Automatic PGA Memory Management	181
PGA and UGA Wrap-Up	195
The System Global Area	196
Fixed SGA	204
Redo Buffer	204
Database Block Buffer Cache	206
Shared Pool	214
Large Pool	218
Java Pool	219
Streams Pool	220
In-Memory Pool	220
System Global Area (SGA) Memory Management	221
Summary	228

TABLE OF CONTENTS

Chapter 5: Oracle Processes..... 229

 Server Processes 230

 Dedicated Server Connections 231

 Shared Server Connections 233

 Connections vs. Sessions 235

 Dedicated Server vs. Shared Server vs. DRCP 244

 Dedicated/Shared Server Wrap-Up 250

 Background Processes 251

 Focused Background Processes 252

 Utility Background Processes..... 267

 Slave Processes..... 271

 I/O Slaves..... 271

 Pnnn: Parallel Query Execution Servers 272

 Summary..... 273

Chapter 6: Locking and Latching 275

 What Are Locks? 275

 Locking Issues 280

 Lost Updates..... 280

 Pessimistic Locking..... 282

 Optimistic Locking..... 284

 Optimistic or Pessimistic Locking? 294

 Blocking..... 295

 Deadlocks 299

 Lock Escalation 306

 Lock Types 307

 DML Locks 308

 DDL Locks..... 323

Latches	330
Mutexes	343
Manual Locking and User-Defined Locks	343
Summary.....	345
Chapter 7: Concurrency and Multiversioning	347
What Are Concurrency Controls?	347
Transaction Isolation Levels.....	349
READ UNCOMMITTED	351
READ COMMITTED	354
REPEATABLE READ.....	356
SERIALIZABLE	359
READ ONLY	363
Implications of Multiversion Read Consistency	364
A Common Data Warehousing Technique That Fails.....	364
An Explanation for Higher Than Expected I/O on Hot Tables.....	366
Write Consistency	370
Consistent Reads and Current Reads	370
Seeing a Restart	374
Why Is a Restart Important to Us?	378
Summary.....	380
Chapter 8: Transactions.....	383
Transaction Control Statements.....	384
Atomicity	386
Statement-Level Atomicity	386
Procedure-Level Atomicity	389
Transaction-Level Atomicity	394
DDL and Atomicity	394
Durability	395
WRITE Extensions to COMMIT.....	395
COMMITs in a Nondistributed PL/SQL Block	397

TABLE OF CONTENTS

Integrity Constraints and Transactions..... 400

 IMMEDIATE Constraints 400

 DEFERRABLE Constraints and Cascading Updates..... 401

Bad Transaction Habits 406

 Committing in a Loop 407

 Using Autocommit 416

Distributed Transactions 417

Autonomous Transactions 419

 How Autonomous Transactions Work 420

 When to Use Autonomous Transactions..... 423

Summary..... 427

Chapter 9: Redo and Undo 429

 What Is Redo? 429

 What Is Undo? 431

 How Redo and Undo Work Together 435

 Example INSERT-UPDATE-DELETE-COMMIT Scenario 436

 Commit and Rollback Processing 442

 What Does a COMMIT Do? 442

 What Does a ROLLBACK Do? 453

 Investigating Redo 455

 Measuring Redo 456

 Can I Turn Off Redo Log Generation? 458

 Why Can't I Allocate a New Log? 464

 Block Cleanout..... 466

 Log Contention 472

 Temporary Tables and Redo 474

Investigating Undo	482
What Generates the Most and Least Undo?	482
ORA-01555: Snapshot Too Old Error	485
Undo Segments Are in Fact Too Small	488
Delayed Block Cleanout.....	495
Summary.....	501
Chapter 10: Database Tables	503
Types of Tables.....	503
Terminology	507
Segment	507
Segment Space Management	510
High-Water Mark	511
PCTFREE	514
LOGGING and NOLOGGING	518
INITRANS	519
Heap-Organized Tables	519
Index-Organized Tables.....	524
Index-Organized Tables Wrap-Up	545
Index Clustered Tables	546
Index Clustered Tables Wrap-Up.....	556
Hash Clustered Tables.....	556
Hash Clustered Tables Wrap-Up	569
Sorted Hash Clustered Tables	570
Nested Tables.....	574
Nested Tables Syntax	575
Nested Table Storage.....	585
Nested Tables Wrap-Up	590

TABLE OF CONTENTS

Temporary Tables	590
Global Temporary Tables.....	591
Global Temporary Table Statistics.....	597
Private Temporary Tables.....	603
Temporary Tables Wrap-Up.....	606
Object Tables.....	607
Object Tables Wrap-Up	616
Blockchain Tables	616
Blockchain Drop Table Clause	617
Blockchain Row Retention Clause.....	618
Blockchain Hash and Data Format Clause.....	618
Creating a Blockchain Table	618
DBMS_BLOCKCHAIN_TABLE Package	620
Blockchain Table Summary	622
Summary.....	622
Chapter 11: Indexes.....	625
An Overview of Oracle Indexes	626
B*Tree Indexes	629
Index Key Compression	632
Reverse Key Indexes	637
Descending Indexes	643
When Should You Use a B*Tree Index?	646
B*Trees Wrap-Up	663
Bitmap Indexes	664
When Should You Use a Bitmap Index?	665
Bitmap Join Indexes	672
Bitmap Indexes Wrap-Up.....	676

Function-Based Indexes	676
A Simple Function-Based Index Example	677
Indexing Only Some of the Rows	690
Implementing Selective Uniqueness	692
Caveat Regarding ORA-01743	693
Function-Based Indexes Wrap-Up	694
Application Domain Indexes	695
Invisible Indexes	696
Multiple Indexes on the Same Column Combinations	699
Indexing Extended Columns	700
Virtual Column Solution	701
Function-Based Index Solution	704
Frequently Asked Questions and Myths About Indexes	707
Do Indexes Work on Views?	707
Do Nulls and Indexes Work Together?	707
Should Foreign Keys Be Indexed?	711
Why Isn't My Index Getting Used?	713
Myth: Space Is Never Reused in an Index	721
Myth: Most Discriminating Elements Should Be First	723
Automatic Indexing	727
Managing Automatic Indexing	729
Automatic Indexing in Action	731
Auto Indexing Wrap-Up	735
Summary	736
Chapter 12: Datatypes	739
An Overview of Oracle Datatypes	739
Character and Binary String Types	743
NLS Overview	744
Character Strings	750

TABLE OF CONTENTS

Binary Strings: RAW Types 759

Extended Datatypes 762

Number Types 766

 NUMBER Type Syntax and Usage..... 769

 BINARY_FLOAT/BINARY_DOUBLE Type Syntax and Usage..... 774

 Non-native Number Types 775

 Performance Considerations 776

Long Types 778

 Restrictions on LONG and LONG RAW Types..... 779

 Coping with Legacy LONG Types 781

Dates, Timestamps, and Interval Types 787

 Formats 788

 DATE Type 790

 TIMESTAMP Type 798

 INTERVAL Type..... 809

LOB Types..... 812

 Internal LOBs 813

 BFILEs..... 834

ROWID/UROWID Types..... 836

JSON Type 838

Summary..... 840

Chapter 13: Partitioning 841

 Partitioning Overview..... 842

 Increased Availability..... 843

 Reduced Administrative Burden 846

 Enhanced Statement Performance..... 852

 Reduced Contention in an OLTP System..... 854

Table Partitioning Schemes	854
Range Partitioning	856
Hash Partitioning	860
List Partitioning	866
Interval Partitioning	869
Reference Partitioning	878
Interval Reference Partitioning	884
Virtual Column Partitioning	887
Composite Partitioning	889
Row Movement	892
Table Partitioning Schemes Wrap-Up	895
Partitioning Indexes	897
Local Indexes vs. Global Indexes	898
Local Indexes	899
Global Indexes	908
Partial Indexes	928
Partitioning and Performance, Revisited	931
Ease of Maintenance Features	939
Multiple Partition Maintenance Operations	940
Cascade Truncate	942
Cascade Exchange	944
Auditing and Segment Space Compression	946
Summary	948
Chapter 14: Parallel Execution	951
When to Use Parallel Execution	953
A Parallel Processing Analogy	954
Parallel Query	956
Parallel DML	964

TABLE OF CONTENTS

Parallel DDL.....	971
Parallel DDL	972
Parallel DDL and Extent Trimming	974
Procedural Parallelism	987
Parallel Pipelined Functions	988
Do-It-Yourself Parallelism.....	992
Old-School Do-It-Yourself Parallelism	997
Summary.....	1002
Chapter 15: Data Loading and Unloading	1005
External Tables	1005
Introduction to External Tables	1007
Creating an External Table Using Legacy SQLLDR Control File.....	1010
Dealing with Errors.....	1017
Using an External Table to Load Different Files	1021
Multiuser Issues	1022
Preprocessing.....	1023
External Table Summary.....	1036
Data Pump Unload	1036
SQLLDR	1039
Running SQLLDR in Express Mode	1044
Loading Data with SQLLDR FAQs.....	1048
SQLLDR Caveats	1085
SQLLDR Summary	1086
Flat File Unload	1086
Using SQL*Plus.....	1086
Using PL/SQL	1087
Summary.....	1099
Index.....	1101

About the Authors



Darl Kuhn is a DBA/developer working for Oracle. He also teaches Oracle classes at the University of Denver in Colorado, and is an active member of the Rocky Mountain Oracle Users Group. Darl enjoys sharing knowledge, which has led to several book projects over the years.



Thomas Kyte has been working for Oracle (the company) since version 7.0.9 (1993) and working with Oracle (software) since version 5.1.5c (the \$99 single-user version for DOS on 360KB floppy disks). Before working at Oracle, he worked for more than six years as a systems integrator, building large-scale, heterogeneous databases and applications, mostly for military and government customers. These days, he spends a great deal of his time working with

the Oracle database and, more specifically, helping people who are using the Oracle database. He works directly with customers, either in specifying and building their systems or, more frequently, helping them rebuild or tune them. In addition, he authors the “Ask Tom” column in *Oracle Magazine*, where he answers questions about the Oracle database and tools. On a typical day, he receives and answers dozens of questions at asktom.oracle.com. Every two months, he publishes a “best of” in the magazine (all of the questions asked are available on the Web, stored in an Oracle database). Additionally, he gives technical seminars covering much of the material in this book.

ABOUT THE AUTHORS

He spends a lot of his time helping people be successful with the Oracle database. In his spare time, he builds applications and develops software within Oracle Corporation. This book is a reflection of what he does every day, covering topics and questions that people struggle with every day. The issues are covered from the perspective of “When I use this, I do it this way.” It is the culmination of many years of experience using the product in myriad situations.

Acknowledgments

I would like to thank many people for helping me complete this book.

First, I would like to thank you, the reader of this book. There is a high probability that if you are reading this book, you have participated in my site <http://asktom.oracle.com> in some fashion, perhaps by asking a question or two. It is that act—the act of asking questions and of questioning the answers—that provides me with the material for the book and the knowledge behind the material. Without the questions, I would not be as knowledgeable about the Oracle database as I am. So, it is you who ultimately makes this book possible.

I would like to thank Tony Davis for his previous work making my work read well. If you enjoy the flow of the sections, the number of section breaks, and the clarity, then that is in some part due to him. I have worked with Tony writing technical material since the year 2000 and have watched his knowledge of Oracle grow over that time. He now has the ability to not only edit the material but in many cases tech edit it as well. Many of the examples in this book are there because of him (pointing out that the casual reader was not going to “get it” without them). This book would not be what it is without him.

Without a technical review team of the caliber I had during the writing of this book and the previous editions, I would be nervous about the content. The first edition had Jonathan Lewis, Roderick Manalac, Michael Möller, and Gabe Romanescu as technical reviewers. They spent many hours poring over the material and verifying it was technically accurate as well as useful in the real world. Subsequent editions had a team of similar caliber: Melanie Caffrey, Christopher Beck, and Jason Straub. I firmly believe a technical book should be judged not only by who wrote it but also by who reviewed it. Given these seven people, I feel confident in the material.

At Oracle, I work with the best and brightest people I have ever known, and they all have contributed in one way or another. I would like to thank Ken Jacobs in particular for his support and enthusiasm over the years. Ken is unfortunately (for us) no longer with Oracle Corporation, but his impact will long be felt.

Lastly, but most important, I would like to acknowledge the unceasing support I’ve received from my family. You know you must be important to someone when you try to do something that takes a lot of “outside of work hours” and that someone lets you know

ACKNOWLEDGMENTS

about it. Without the continual support of my wife, Melanie (who also was a technical reviewer on the book), son Alan, and daughter Megan, I don't see how I could have finished this book.

—Thomas Kyte

I'd like to thank Tom for inviting me to work with him on this book; this is a great technical honor. I'd also like to acknowledge Jonathan Gennick; his guidance (over many years and books) laid the foundation for me being able to work on a book of this caliber. And I'd like to thank Heidi, Lisa, Evan, and Brandi; without their support, I could not have successfully participated.

—Darl Kuhn

Introduction

The inspiration for the material contained in this book comes from my experiences developing Oracle software and from working with fellow Oracle developers and DBAs to help them build reliable and robust applications based on the Oracle database. The book is basically a reflection of what I do every day and of the issues I see people encountering each and every day.

I covered what I felt was most relevant, namely, the Oracle database and its architecture. I could have written a similarly titled book explaining how to develop an application using a specific language and architecture—for example, one using JavaServer Pages that speaks to Enterprise JavaBeans, which in turn uses JDBC to communicate with Oracle. However, at the end of the day, you really do need to understand the topics covered in this book in order to build such an application successfully. This book deals with what I believe needs to be universally known to develop successfully with Oracle, whether you are a Visual Basic programmer using ODBC, a Java programmer using EJBs and JDBC, or a Perl programmer using DBI Perl. This book does not promote any specific application architecture; it does not compare three-tier to client/server. Rather, it covers what the database can do and what you must understand about the way it works. Since the database is at the heart of any application architecture, the book should have a broad audience.

As the title suggests, *Expert Oracle Database Architecture* concentrates on the database architecture and how the database itself works. I cover the Oracle database architecture in depth: the files, memory structures, and processes that comprise an Oracle database and instance. I then move on to discuss important database topics such as locking, concurrency controls, how transactions work, and redo and undo, and why it is important for you to know about these things. Lastly, I examine the physical structures in the database such as tables, indexes, and datatypes, covering techniques for making optimal use of them.

What This Book Is About

One of the problems with having plenty of development options is that it's sometimes hard to figure out which one might be the best choice for your particular needs. Everyone wants as much flexibility as possible (as many choices as they can possibly have), but they also want things to be very cut and dried—in other words, easy. Oracle presents developers with almost unlimited choice. No one ever says, “You can't do that in Oracle.” Rather, they say, “How many different ways would you like to do that in Oracle?” I hope that this book will help you make the correct choice.

This book is aimed at those people who appreciate the choice but would also like some guidelines and practical implementation details on Oracle features and functions. For example, Oracle has a really neat feature called parallel execution. The Oracle documentation tells you how to use this feature and what it does. Oracle documentation does not, however, tell you when you should use this feature and, perhaps even more important, when you should not use this feature. It doesn't always tell you the implementation details of this feature, and if you're not aware of them, this can come back to haunt you (I'm not referring to bugs, but the way the feature is supposed to work and what it was really designed to do).

In this book, I strove to not only describe how things work but also explain when and why you would consider using a particular feature or implementation. I feel it is important to understand not only the “how” behind things but also the “when” and “why” as well as the “when not” and “why not!”

Who Should Read This Book

The target audience for this book is anyone who develops applications with Oracle as the database back end. It is a book for professional Oracle developers who need to know how to get things done in the database. The practical nature of the book means that many sections should also be very interesting to the DBA. Most of the examples in the book use SQL*Plus to demonstrate the key features, so you won't find out how to develop a really cool GUI—but you will find out how the Oracle database works, what its key features can do, and when they should (and should not) be used.

This book is for anyone who wants to get more out of Oracle with less work. It is for anyone who wants to see new ways to use existing features. It is for anyone who wants to see how these features can be applied in the real world (not just examples of how

to use the feature, but why the feature is relevant in the first place). Another category of people who would find this book of interest is technical managers in charge of the developers who work on Oracle projects. In some respects, it is just as important that they understand why knowing the database is crucial to success. This book can provide ammunition for managers who would like to get their personnel trained in the correct technologies or ensure that personnel already know what they need to know.

To get the most out of this book, the reader should have

- *Knowledge of SQL*: You don't have to be the best SQL coder ever, but a good working knowledge will help.
- *An understanding of PL/SQL*: This isn't a prerequisite, but it will help you to absorb the examples. This book will not, for example, teach you how to program a FOR loop or declare a record type; the Oracle documentation and numerous books cover this well. However, that's not to say that you won't learn a lot about PL/SQL by reading this book. You will. You'll become very intimate with many features of PL/SQL, you'll see new ways to do things, and you'll become aware of packages/features that perhaps you didn't know existed.
- *Exposure to some third-generation language (3GL), such as C or Java*: I believe that anyone who can read and write code in a 3GL language will be able to successfully read and understand the examples in this book.
- *Familiarity with the Oracle Database Concepts manual*.

A few words on that last point: due to the Oracle documentation set's vast size, many people find it to be somewhat intimidating. If you're just starting out or haven't read any of it as yet, I can tell you that the *Oracle Database Concepts* manual is exactly the right place to start. It's about 600+ pages long (I know that because I wrote some of the pages and edited every one) and touches on many of the major Oracle concepts that you need to know about. It may not give you each and every technical detail (that's what the other 10,000 to 20,000 pages of documentation are for), but it will educate you on all the important concepts. This manual touches the following topics (to name a few):

- The structures in the database and how data is organized and stored
- Distributed processing

INTRODUCTION

- Oracle’s memory architecture
- Oracle’s process architecture
- Schema objects you will be using (tables, indexes, clusters, and so on)
- Built-in datatypes and user-defined datatypes
- SQL stored procedures
- How transactions work
- The optimizer
- Data integrity
- Concurrency control

I will come back to these topics myself time and time again. These are the fundamentals. Without knowledge of them, you will create Oracle applications that are prone to failure. I encourage you to read through the manual and get an understanding of some of these topics.

How This Book Is Structured

This book has 15 chapters, and each is like a “minibook”—a virtually stand-alone component. Occasionally, I refer to examples or features in other chapters, but you could pretty much pick a chapter out of the book and read it on its own. For example, you don’t have to read Chapter 10 on database tables to understand or make use of Chapter 14 on parallelism.

The format and style of many of the chapters is virtually identical:

- An introduction to the feature or capability.
- Why you might want to use the feature or capability (or not). I outline when you would consider using this feature and when you would not want to use it.
- How to use this feature. The information here isn’t just a copy of the material in the SQL reference; rather, it’s presented in a step-by-step manner: here is what you need, here is what you have to do, and these are the switches you need to go through to get started. Topics covered in this section will include

- How to implement the feature
- Examples, examples, examples
- How to debug this feature
- Caveats of using this feature
- How to handle errors (proactively)
- A summary to bring it all together

There will be lots of examples and lots of code, all of which is available for download from the GitHub site. The following sections present a detailed breakdown of the content of each chapter.

Where Can I Find the Book's Source Code?

The best way to digest the material in this book is to thoroughly work through and understand the hands-on examples. As you work through the examples in this book, you may decide that you prefer to type in all the code by hand. Many readers choose to do this because it is a good way to get familiar with the coding techniques that are being used. Having said that, there are many complex examples in this book. Therefore, you may opt for downloading the source code and running examples without having to manually type them in.

All of the source code for this book can be downloaded from the GitHub website. You do not need a GitHub account to access the source code, but we recommend signing up to make the most of this service. To find the source code for this book (or any Apress book):

1. Go to the book's product page on Apress.com, located at www.apress.com/9781484274989.
2. There will be a button marked **Download Source Code**. Click this to be taken to the book's page on GitHub.
3. Once on GitHub, download the code as a zip using the green button, or, if you have a GitHub account, you can clone the source code directly to your machine using Git.
4. That's it!

Source code can be continuously updated after a book has published. That means that if there are any corrections, you will always get the latest version. If for any reason you want to get hold of the original source code, exactly as it is in your copy of the book, you can go to [https://github.com/Apress/\[repository-name-here\]/releases](https://github.com/Apress/[repository-name-here]/releases) and download release v1.0.

If you like to type in the code, you can use the source code files to check the results you should be getting—they should be your first stop if you think you might have typed an error. If you don't like typing, then downloading the source code from the GitHub site is a must! Either way, the code files will help you with updates and debugging.

Tip If you have any problems accessing the source code for an Apress book, email customerservice@springernature.com.

Setting Up Your Environment

In this section, I will cover how to set up an environment capable of executing the examples in this book. Specifically

- Accessing an Oracle database
- How to set up the EODA account used for many of the examples in this book
- How to set up the SCOTT/TIGER demonstration schema properly
- Installing Statspack
- Installing and running runstats, creating the BIG_TABLE, and other custom utilities used throughout the book

As described previously in the “Where Can I Find the Book’s Source Code?” section, all of the scripts used in this book are available for download from the GitHub site. There is a chNN folder that contains the scripts for each chapter (where NN is the number of the chapter). The ch00 folder contains the scripts listed here in the “Setting Up Your Environment” section.

Most of the examples in this book are designed to run 100 percent in the SQL*Plus environment. If you already have access to an Oracle database, then you can skip ahead to creating the EODA and SCOTT schemas in your database. You'll also need to set up Statspack and the custom scripts. These components are used extensively throughout the book.

Accessing an Oracle Database

This book is chock full of hands-on database examples. Therefore, it's critical that you have access to an Oracle database as you work through the examples in each chapter. If you want to use an Oracle database installed on your PC, there are a couple of free and simple ways of doing this. Both techniques listed next involve using Oracle VM VirtualBox:

- Installing Oracle VM VirtualBox and a pre-built database VM
- Installing Oracle VM VirtualBox, cloning a Git repository, and running Vagrant to build your environment

I'll briefly describe both of the prior techniques in the following sections.

Oracle VM VirtualBox and a Pre-built Database VM

One of the quickest free and easy ways to gain access to a fully functional Oracle database is to download and install Oracle VM VirtualBox and use it with a pre-built database VM. You can literally have a working database within a few minutes of downloading and installing the required software.

First, you must download and install VirtualBox. To do this, go to this link and download and install the software:

www.virtualbox.org/wiki/Downloads

After you have downloaded and installed VirtualBox, then download a pre-built database VM and follow the instructions for importing the appliance VM into VirtualBox. Use this link to download the pre-built VM:

www.oracle.com/downloads/developer-vm/community-downloads.html

The Database App Development VM includes a container database named CDB\$ROOT. Within this container database, there is a pluggable database named ORCL. When logging

INTRODUCTION

onto the Database App Development VM, use the oracle OS account (there should be a default terminal window open for you after starting the VM). In the default terminal window, you can access SQL*Plus as follows:

```
$ sqlplus / as sysdba
```

If you're prompted for a username and password, exit the SQL*Plus session, and set the following from the operating system prompt:

```
$ export TWO_TASK=
```

After you've set the prior environment variable, you should be able to access SQL*Plus via

```
$ sqlplus / as sysdba
```

Using Oracle VM VirtualBox with a pre-built VM is by far the easiest way to gain access to a fully functional Oracle database. If you are a bit more technically savvy, then I would suggest using a Vagrant box described in the next section to build an environment where you can access an Oracle database on your PC.

Oracle VM VirtualBox, Git, and Vagrant

This approach requires that you download and install Oracle VM VirtualBox, Git, and Vagrant. You also need to download the Oracle installation media. After you've installed those, then use Git to clone a Vagrant repository and then use a Vagrant box to build a virtual machine on your laptop. This approach might seem a little daunting at first, so I would suggest you look up Tim Hall's YouTube video titled "Vagrant: Oracle Database Build." That YouTube video walks you through the entire process.

Described next are the high-level steps for building an Oracle environment. First, navigate to Oracle's database download site and download the Oracle installation software:

www.oracle.com/database/technologies/oracle-database-software-downloads.html

Now navigate to this link and download and install Oracle VM VirtualBox on your laptop:

www.virtualbox.org/wiki/Downloads

Next, navigate to the Git download page and download and install Git on your laptop:

<https://git-scm.com/downloads>

Next, navigate to the Vagrant download page and download and install Vagrant on your laptop:

www.vagrantup.com/docs/installation

Now start up the git bash shell. On Windows, you do this by typing in “git bash” in the Startup box. Using the git bash shell window, create a directory:

```
mkdir c:\vagrantboxes
```

Change directories to that directory:

```
cd c:\vagrantboxes
```

Now use git to clone the Vagrant box repository:

```
git clone https://github.com/oracle/vagrant-boxes
```

Next, change directories to the version of the Oracle database that you want to install:

```
cd c:\vagrant-boxes\OracleDatabase\<Database Version>
```

Copy the Oracle installation media to this directory also:

```
cp <Oracle Install Media> c:\vagrant-boxes\OracleDatabase\<Database Version>
```

Now type in vagrant up:

```
vagrant up
```

After the VM is started, you should be able to access the VM via a secure shell session:

```
vagrant ssh
```

Once you’ve logged on to the VM, you should be able to access the root and oracle OS account. For example:

```
$ sudo su - oracle
```

INTRODUCTION

Using a Vagrant box is an extremely powerful way to create your own VMs that contain Oracle databases. You can even easily build a RAC database environment from scratch using these techniques.

Database Setup

All of the examples in this book are run against a container database named CDB. I'll connect to the root CDB container to perform tasks such as modifying initialization parameters, stopping/starting the database, and so forth. The CDB container database contains two pluggable databases PDB1 and PDB2. When demonstrating application-type examples, I'll connect to the PDB1 pluggable database as either EODA or SCOTT. For reference, here's the Database Creation Assistant (DBCA) code that I used to create the CDB multitenant database:

```
dbca -silent -createDatabase \  
-templateName General_Purpose.dbc \  
-createAsContainerDatabase true \  
-pdbName PDB -numberOfPDBs 2 \  
-sid CDB -gdbName CDB \  
-characterset AL32UTF8 \  
-sysPassword foo \  
-systemPassword foo \  
-pdbAdminUserName pdbadmin \  
-pdbAdminPassword foo \  
-initParams sga_target=1024M,pga_aggregate_target=512M
```

After the database is created, I can connect to the CDB root container as SYS:

```
$ sqlplus / as sysdba  
  
SQL> show con_name  
CON_NAME  
-----  
CDB$ROOT
```

As SYS, if I want to switch containers to the pluggable database, I can do so as follows:

```
SQL> alter session set container=PDB1;
```

To connect to the pluggable database as SYSTEM, EODA, and/or SCOTT, I'll use the pluggable database default service to connect, for example:

```
$ sqlplus eoda/foo@PDB1
```

```
SQL> show con_name
```

```
CON_NAME
-----
PDB1
```

For reference, here are the `tnsnames.ora` file contents on my laptop:

```
CDB =
(DESCRIPTION =
  (ADDRESS = (PROTOCOL = TCP)(HOST = localhost)(PORT = 1521))
  (CONNECT_DATA =
    (SERVER = DEDICATED)
    (SERVICE_NAME = CDB)
  )
)

PDB1 =
(DESCRIPTION =
  (ADDRESS = (PROTOCOL = TCP)(HOST = localhost)(PORT = 1521))
  (CONNECT_DATA =
    (SERVER = DEDICATED)
    (SERVICE_NAME = PDB1)
  )
)
```