

Understanding Oracle APEX 20 Application Development

Think Like an Application Express Developer

—

Third Edition

—

Edward Sciore

Apress®

Understanding Oracle APEX 20 Application Development

**Think Like an Application
Express Developer**

Third Edition

Edward Sciore

Apress®

Understanding Oracle APEX 20 Application Development: Think Like an Application Express Developer

Edward Sciore
Newton Center, MA, USA

ISBN-13 (pbk): 978-1-4842-6164-4
<https://doi.org/10.1007/978-1-4842-6165-1>

ISBN-13 (electronic): 978-1-4842-6165-1

Copyright © 2020 by Edward Sciore

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 image designed by Freepik (www.freepik.com)

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. 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/9781484261644. For more detailed information, please visit <http://www.apress.com/source-code>.

Printed on acid-free paper

*To my parents, for their many years of unwavering
love and support.*

Table of Contents

About the Author	xiii
About the Technical Reviewer	xv
Acknowledgments	xvii
Introduction	xix
Chapter 1: The SQL Workshop	1
Downloading Tables	3
Object Browser	5
SQL Command Tool	9
Summary.....	12
Chapter 2: Applications and Pages.....	13
Creating an Application.....	13
Editing Application Properties.....	17
Deleting an Application	19
Running an Application Page	19
Running a Page from Within APEX.....	19
Running a Page from Outside APEX.....	22
Creating a New Page	22
Page Designer.....	25
Property Editor	28
Accessing Built-in Help.....	32
Summary.....	32

TABLE OF CONTENTS

- Chapter 3: Regions 33**
- Creating and Deleting Regions..... 34
- Type-Independent Region Properties..... 35
 - Identification and Source Sections..... 36
 - Layout Section..... 39
 - Appearance Section 45
- Drag and Drop..... 46
- Static Content Regions..... 49
 - Formatted vs. Unformatted Text 50
 - Substitution Strings..... 53
 - Referring to APEX Pages..... 53
 - Referring to Local Images 54
- Classic Report Regions 57
 - Column-Specific Properties..... 60
 - Report-Specific Properties 69
 - Using the Source Query for Formatting..... 70
- Chart Regions..... 71
 - Creating a Chart Region..... 73
 - Configuring a Chart..... 75
 - Configuring a Chart Series 75
- Multi-Series Charts 78
 - Customizing a Chart 79
- Interactive Report Regions..... 84
 - Using the Search Bar 86
 - Configuring the Search Bar 93
 - Link Column..... 96
- Summary..... 98

Chapter 4: Navigation	101
Lists	101
Viewing List Details	102
Editing List Entries.....	103
Creating New List Entries	105
Creating a New List	108
Modifying the Navigation Interface	109
Configuring the Navigation Menu	110
Configuring the Navigation Bar	112
List Regions	112
Hierarchical Lists	116
Mega Menus	122
Page Hierarchy	123
Breadcrumbs.....	125
Breadcrumb Regions	129
Global Page.....	130
Summary.....	130
Chapter 5: Items and Buttons	133
Items	133
Item Sampler Page.....	134
Creating and Deleting Items.....	137
Positioning an Item	141
Text-Based Items	142
Text Fields.....	143
Number Fields	143
Password Items	143
Text Fields with Autocomplete.....	143
Color Pickers	146
Date Pickers	146
Properties of All Text-Based Items.....	148

TABLE OF CONTENTS

- List-Based Items 150
 - Dynamic List–Based Items 152
 - Static List–Based Items..... 154
- Multi-Value List Items 155
- Yes/No Items 158
- Display-Based Items 160
- Initial Item Values..... 160
- Buttons..... 162
- Summary..... 166
- Chapter 6: Session State 167**
 - Session IDs 167
 - Submit vs. Redirect..... 168
 - Using an Item to Submit a Page..... 169
 - Redirect Can Set Session State 171
 - Referring to Session State Variables..... 172
 - Customized Reports 173
 - Filter by Job and Department Page 173
 - Filter by Possible Department Page 175
 - Filter by Salary Range Page 178
 - Master-Detail Reports 179
 - Chart Drill Down..... 182
 - Customized Titles and Labels..... 185
 - Cascading Lists..... 187
 - Conditional Rendering..... 190
 - Region Display Selectors 193
 - Summary..... 195

Chapter 7: Processes	197
Stages of a Submit Action.....	197
Creating a Process.....	198
PL/SQL to Access the Database.....	200
Delete Employee Region.....	202
Insert Employee Region.....	203
Update Employee Region.....	204
PL/SQL to Compute Values.....	206
Local Variables.....	210
Comments.....	210
Assignment Statements.....	210
Conditional Statements.....	210
Handling Multi-Row SQL Queries.....	211
Concurrent Database Updates.....	214
Lost Update Problem.....	214
Avoiding Lost Updates.....	216
Using a Hash Function.....	220
Success and Error Messages.....	222
Conditional Processes.....	224
Submitting via Items.....	224
Multiple-Use Processes.....	226
Page Rendering Processes.....	229
Executing a Process Without a Submit.....	231
Moving Between Regions.....	235
Finding Previous and Next Rows.....	235
Single Row Update.....	239
Implementing the Report Region.....	241
Implementing the Insert Region.....	241
Implementing the Update/Delete Region.....	242
Summary.....	245

TABLE OF CONTENTS

- Chapter 8: Data Validation 247**
 - Constraint Preservation..... 247
 - Input Validation 252
 - Constraint Validations 259
 - Process Validation..... 260
 - Summary..... 261

- Chapter 9: Branches 263**
 - Separating Input from Output 263
 - Creating a Branch 266
 - Conditional Branching..... 268
 - Wizard-Like Interfaces 269
 - Implementing the Progress Bar..... 272
 - Basic Info Page..... 274
 - Manager Info Page 277
 - Hiredate Info Page..... 278
 - Confirm Employee Info Page 278
 - Summary..... 280

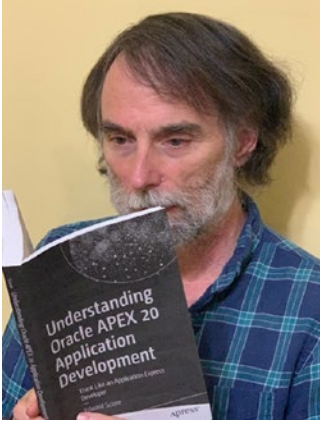
- Chapter 10: Forms 281**
 - Form Regions..... 281
 - Form Update 288
 - Updating Records 289
 - Clearing Session State 293
 - Inserting Records 295
 - Computations..... 298
 - Sending Email 300
 - Report with Form Pages 302
 - The Report Page 309
 - The Form Page 310
 - Customizing the Report and Form..... 312

Form Pages	313
Implementing the Tree Page	315
Implementing the Form Page	316
Master-Detail Forms	318
Summary.....	326
Chapter 11: Interactive Grids.....	327
Read-Only Interactive Grids	327
Editing Interactive Grids.....	331
Ensuring Consistency.....	337
Grid Validations.....	338
Restricting Interactive Grid Input.....	340
Grid Processes	343
Summary.....	348
Chapter 12: Dynamic SQL	349
Dynamic Reports.....	349
Dynamic Processes.....	355
Combining Dynamic Reports and Processes	361
Summary.....	365
Chapter 13: Security.....	367
Authentication.....	367
Managing Authentication Schemes	368
Writing an Authentication Function	372
Public Pages	377
Authorization.....	378
Creating an Authorization Scheme	378
Component Authorization	383
Page Authorization.....	384
Not Public User Scheme	386

TABLE OF CONTENTS

Avoiding Malicious Use 387
 SQL Injection 387
 Cross-Site Scripting..... 395
 URL Modification 403
Summary..... 411
Index..... 413

About the Author



Edward Sciore is a recently retired associate professor in the computer science department at Boston College. He taught college students for more than 35 years. His research specialty is database systems, and he thoroughly enjoys teaching the wonders of database technology to captive students.

About the Technical Reviewer



Armando Plascencia has been a database engineer and software architect in multiple software languages, building systems and solutions for enterprises since the early days of information technology. Specializing in all things Oracle, Java, Linux, Open Source, and APEX, Armando is a natural principal architect. He thrives in dynamic, complex environments, where he leads teams of diversely skilled individuals, one business challenge and deadline at a time. Being an avid learner has made Mr. Plascencia a master technologist. Attending conferences and reading dozens of books each year keep his skills up to date, but teaching others is what keeps him honed and sharp to new technological iterations and innovations. A strong believer in the power of positive thinking and ongoing service to others (Armando and his mother recently made enchiladas for a team filming a documentary about death, grief, and surfing), Armando cares deeply about the larger context of our world, as well as the people who are special in his life. Armando's creed is that writing code, drinking coffee, and developing strong relationships are the foundations of everything. Outside of work, Armando enjoys running, cycling, designing, and working on landscaping projects and the hunt for the perfect cup of Java.

"It was an absolute pleasure to be associated with Apress and I want to acknowledge the amazing work of this author."

—Armando Plascencia

Acknowledgments

First and foremost, I would like to thank the APEX user community. Numerous people routinely and generously share their APEX knowledge by writing blogs, creating demonstration APEX sites, and answering all kinds of questions on the APEX web forums. I learned much from them. This book is my attempt to give something back.

I also want to thank my Apress editors, Jonathan Gennick and Jill Balzano. Jonathan convinced me to write the book, and then to revise it. He provided guidance and encouragement every step of the way. Jill was always supportive and smoothed out the inevitable bumps in the road.

Most importantly, I want to acknowledge my wife Amy. She has accompanied me through all three editions, listening to my ideas, helping me resolve technical issues, working through the APEX examples, and pointing out passages in the book that needed clarification. She is a relentless proofreader. I could have written the book without her, but it would not have been anywhere near as good. Thanks.

Introduction

Application Express (otherwise known as APEX) is a web application tightly coupled to an Oracle database. It has several uses: you can use its SQL Workshop tool to query and modify the database, you can use its App Builder tool to create your own web applications that interact with the database, and you can run the web applications created by you and others.

The app builder is especially interesting because it provides a simple, nontraditional way to build web pages. You do not specify code for the page directly; instead, you choose from a set of built-in templates. There is a template for the overall page and templates for each kind of component that you want to put on the page (such as reports, buttons, etc.). Each template has a set of properties, whose values determine where each component is located on the page, what it looks like, and how it behaves. You create a page simply by choosing templates for the components you want and assigning values to their properties.

The APEX app builder saves the property values for each component in its own database. When a browser requests one of your application's pages, the APEX server extracts the property values relevant to that page from its database, constructs the HTML code corresponding to those values, and returns that code to the browser. This process is called *rendering* the page, and APEX is called an *HTML generator*.

The advantage of using an HTML generator such as APEX is that you can build web pages without any knowledge of HTML (or CSS, JavaScript, or PHP). Moreover, because APEX is tightly coupled to an Oracle database, it automatically handles the intricacies of database interaction. APEX makes it possible to easily create good-looking, highly functional, and database-aware pages with only a rudimentary knowledge of SQL.

Why This Book?

Designing a page with APEX seems straightforward—all you have to do is choose the components you want and then assign the appropriate values to their properties. Choosing components is straightforward, but assigning property values is not. A page

and its components have many properties, and you have to know the purpose of those properties to know what values to assign. These properties range from the essential (such as the *source query* of a report) to the obscure (such as the *static ID* of a report). Some properties (such as the *HTML expression* of a report column) are hooks that allow you to insert customized HTML or JavaScript code into the generated web page.

The purpose of this book is to gently lead you through this cornucopia of properties. To that end, the book develops a demo web application that illustrates various APEX techniques for building typical web page functionality. The pages of this application start out simply and gradually increase their level of sophistication. With each page, I introduce a few new properties, discuss their purpose, and illustrate their usefulness. By the end of the book, you will have been so immersed in the world of APEX properties that you should feel confident enough to tackle any website project of your own. And if your project requires even more sophistication than appears here, you should be comfortable enough to use properties that are not covered, perhaps by looking at the documentation, examining the numerous prepackaged applications provided by Oracle, checking a web forum, or even figuring it out on your own.

Another way to build web pages in APEX is to rely on wizards. APEX provides wizards to generate common components, such as report pages and data entry forms. Each wizard asks you a series of questions (such as “What is the name of the page?” “What table do you want to display?” “Should the page have an entry in the navigation menu?”) and then uses your responses to generate appropriate components having appropriate properties. The advantage, of course, is that you don’t need to know anything about properties. The disadvantage is that wizards tend to produce “one size fits all” pages, in terms of both their appearance and their functionality.

Wizards can take you only so far. If you want any kind of control over the look, feel, and behavior of your page, you need to get involved with its properties. This book provides the guidance you need.

Demo Application

As this book explains each part of the APEX app builder, it guides you through the development of a small application, named Employee Demo. I encourage you to build your own version of the application as you follow along. You can run my version of the application by going to the URL `apex.oracle.com/pls/apex/f?p=91392:1`. You can also download the source code for the application from the Apress website and import it into your own workspace.

Unlike demo applications in many books, this application does not “do” anything particularly interesting. Instead, each page is constructed to illustrate one or more techniques. Some of the pages have similar functionality, to illustrate the trade-offs between different implementation techniques.

The `Employee Demo` application uses the `DEPT` and `EMP` database tables available to every APEX workspace. The `DEPT` table lists the departments of a company, and the `EMP` table lists the employees in those departments. Their columns are as follows:

```
DEPT(DeptNo, DName, Loc)
```

```
EMP (EmpNo, EName, Job, Mgr, HireDate, Sal, Comm, DeptNo)
```

The key of `DEPT` is `DeptNo`, and the key of `EMP` is `EmpNo`. Each table has a built-in sequence for generating unique values for these keys, as well as an associated insertion trigger. If you insert a record into one of the tables and omit a value for the key, the trigger will automatically generate a key value from the appropriate sequence.

The `Employee Demo` application assumes that the `EMP` table has been modified to have an additional column `OffSite` of type `char(1)`. An employee will have the value ‘N’ for this column if the employee works in the department office and ‘Y’ if the employee works offsite. For your reference, here is the SQL code you will need to add this new column to your `EMP` table.

```
alter table EMP
add OffSite char(1);
```

After altering the table, you will also need to assign an `Offsite` value for each existing employee. In my `Employee Demo` application, the employees `SCOTT`, `ALLEN`, `WARD`, and `TURNER` work offsite; the others work onsite. Chapter 1 describes how to import the tables if they are not already in your workspace and discusses the APEX tools needed to make these modifications to them.

Required Background

This book is for people who are comfortable using a database system and want to learn how to write nontrivial web applications in APEX. Many of the techniques used to write APEX pages involve various skills in the following database and web design languages and technologies.

SQL

The most important skill you need is the ability to write SQL queries. All data access in APEX is performed via SQL statements, and the value of many properties involves SQL in some way. The more fluent you are in SQL, the more sophisticated your reports and forms can be. This book assumes that you are comfortable with SQL. For the most part, the Employee Demo application uses relatively simple SQL statements, but occasionally I include something more complex (such as an outer join or nested query) to illustrate what is possible.

HTML

This book also assumes that you have a basic familiarity with HTML—in particular, how tags such as `<p>`, ``, `<a>`, and `` can be used to format text and display images. I will ignore advanced features such as JavaScript and CSS.

PL/SQL

APEX uses PL/SQL to specify behavior. PL/SQL is Oracle's programming language; its main feature is an embedded SQL syntax that makes it easy to write procedures that interact with the database. You should have a rudimentary understanding of programming, although prior knowledge of PL/SQL is not necessary. This book introduces PL/SQL in Chapter 7 and uses only basic features of the language.

APEX

Finally, this book does not require you to have prior experience with APEX. Although it is possible to follow the book without actually using APEX, doing so seems rather pointless. So you should get an APEX account. The easiest and best way to get an account is by going to the apex.oracle.com site. Because I created my Employee Demo application from there, you should see the same screens that appear in this book.

Distinguishing Screens from Pages

APEX is a web application that is used to create other web applications. Thus, APEX has a home page, and its various tools have their own sets of pages. Throughout this book,

I describe how to use APEX to build a page of an application. This can lead to some strange sentences, such as “Clicking the Run button from the application’s APEX home page runs its home page.” To avoid such confusion, I denote all APEX pages as “screens.” The previous sentence becomes “Clicking the Run button from the application’s home screen runs its home page,” which is less awkward and hopefully less confusing.

New to This Edition

Although APEX is a polished and highly functional application, it is (and always has been) a work in progress. The APEX developers have been relentless in their quest to improve the system and have not been shy about making large changes to the functionality of APEX and its user interface.

The previous edition of this book was written in 2015, just as APEX 5 was introduced. Since then, the user interface has undergone a series of significant changes. In fact, when I recently read through my APEX 5 book, I was dismayed to discover that its screenshots and instructions were often inaccurate and occasionally meaningless. Many APEX properties are now in different locations, sometimes with different names. New component types (in particular *forms* and *interactive grids*) have replaced old ones and have different functionality. Clearly it was time to revise the book.

The most significant additions to this book are the two new chapters on Forms and Interactive Grids. In APEX 5, a “form” was a static content region containing an intricately configured set of items and built-in processes. Now APEX has a dedicated Form region, which automatically generates the preconfigured items and processes for you and provides an easily understood set of properties for managing them.

Interactive grid regions have replaced the “tabular form” regions of APEX 5. They provide the same basic functionality as tabular forms—namely, the ability to update a report’s values in place—but interactive grids are easier to use, more powerful, and have a cleaner semantics. In fact, the Interactive Grids chapter (Chapter 11) of this book is much shorter than the older Tabular Forms chapter because interactive grids require much less explanation.

CHAPTER 1

The SQL Workshop

Congratulations! You are on the brink of learning how to build APEX web applications. But before you can begin, you need an APEX account. There are several ways to obtain an account: you can register for a free account through the `apex.oracle.com` website, you might be given an account to an APEX server at your job, or you can even install an APEX server yourself and create your own account to it.

No matter the circumstance, your APEX server will have an associated URL. Invoking the URL from your browser takes you to a login screen. Figure 1-1 shows the login screen used by the `apex.oracle.com` server.

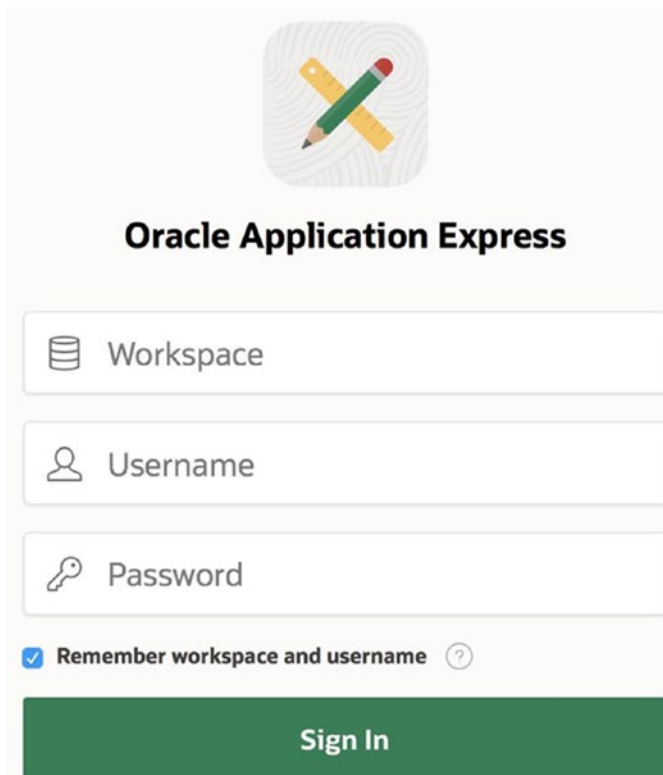


Figure 1-1. APEX login screen

Entering your credentials then takes you to the APEX home screen, the top of which is shown in Figure 1-2.

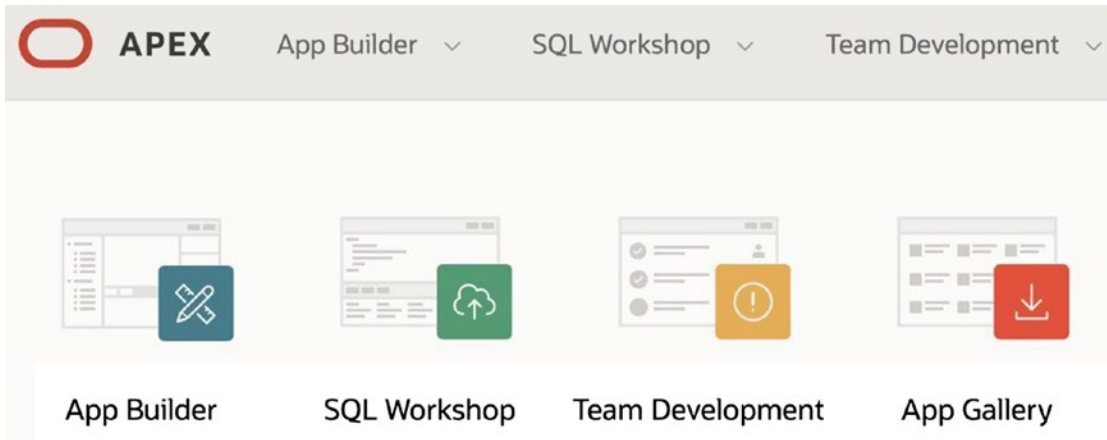


Figure 1-2. APEX home screen

The APEX development environment contains several tools. Of primary importance is the App Builder tool, which will be covered in depth starting in Chapter 2. We begin here by looking at two tools in the APEX SQL Workshop: the object browser and the SQL command tool. These tools allow you to directly access the database—the object browser lets you manipulate a graphical user interface, and the SQL command tool lets you execute SQL statements and PL/SQL code blocks.

Although the SQL Workshop tools are not essential for application development, using them can make your life much easier. Here are five ways that they can help:

- *The object browser reminds you of the database objects and their structures.* For example, a typical application involves several tables, each of which can have numerous columns. It is often impractical to memorize the details of each one. When building a page that references a table, you can use these tools to help refresh your memory.
- *To modify the structure of the database.* For example, these tools are the easiest way to execute the `alter table` command given in the introduction.
- *To modify the contents of the database.* For example, you might want to insert or modify records to test the behavior of a page with new or altered data, or to reset the database after testing the page.

- *To examine the contents of the database tables.* After running a page, you can verify that the database updated correctly.
- *To debug an SQL statement or PL/SQL block.* By executing code in the SQL commands tool first, you can verify that it produces the expected result before you actually assign it as the value of some property on a page.

To get to the SQL Workshop, click the SQL Workshop button on the APEX home screen. Figure 1-3 shows the resulting screen. From this screen, you can click the Object Browser or SQL Commands button to get to the desired tool.

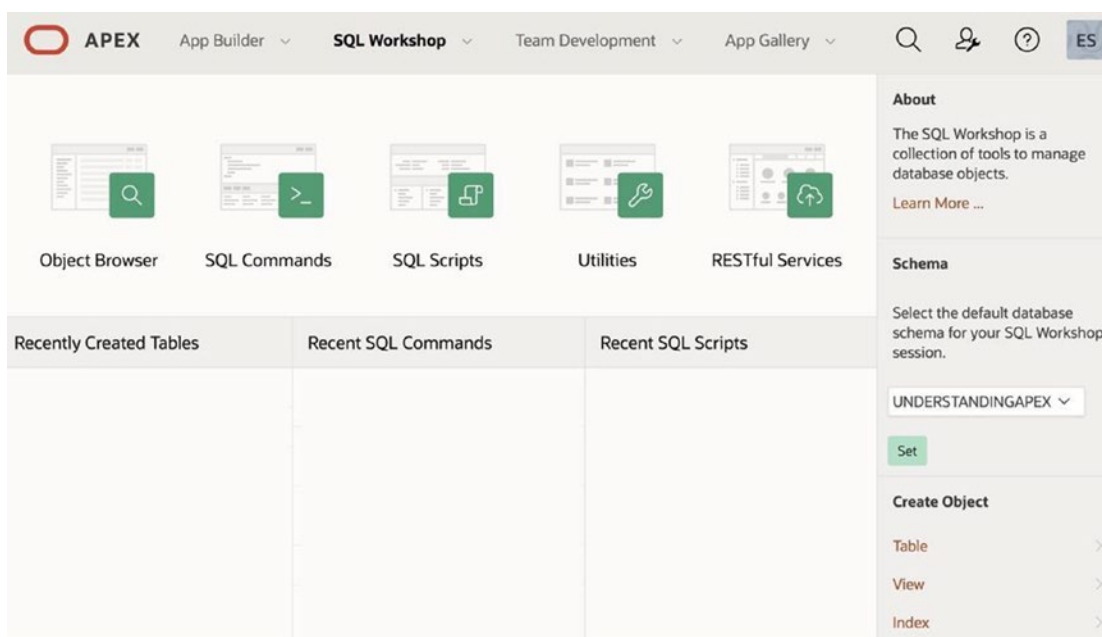


Figure 1-3. SQL Workshop home screen

Downloading Tables

This book makes frequent use of the sample tables EMP and DEPT. If your workspace does not contain these tables, here is how to load them. Click the arrow of the SQL Workshop tab, select Utilities, and then Sample Datasets; see Figure 1-4. Figure 1-5 shows the resulting screen, which lists the available tables. Click the Install button for the EMP/DEPT dataset, and follow directions.

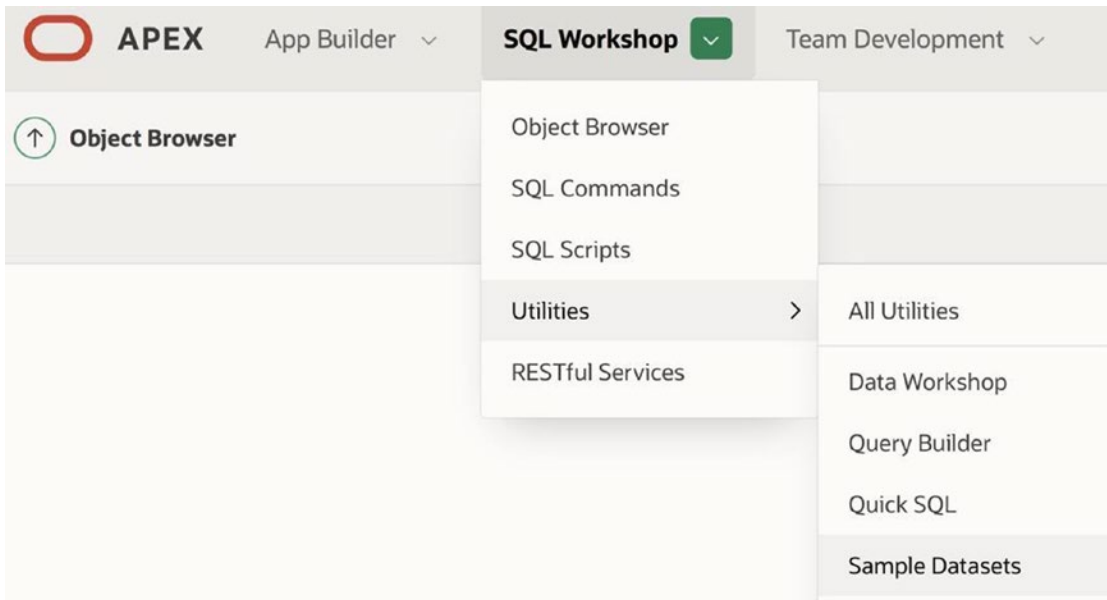


Figure 1-4. Getting to the sample datasets

Utilities \ Manage Sample Dataset

Sample Datasets

Action	Name	Languages	Description
Install	Countries	English	Listing of countries, population, and capital.
Install	Customer Orders	English	A collection of customers, stores, products, and orders. This dataset includes JSON data for the product description, and longitude / latitude for the stores.
Install	EMP / DEPT	English, Chinese, Czech, French, German, Japanese, Korean, Polish, Russian, Spanish	The generic EMP and DEPT tables.

Figure 1-5. The Sample Datasets screen

To verify that the tables have been installed, return to the SQL Workshop home screen (as shown in Figure 1-3). Entries for EMP and DEPT should now appear in the Recently Created Tables region.

Object Browser

The object browser lets you interact with your tables quickly and easily. From it, you can examine the description of each table—that is, the types and properties of its columns and its constraints, indexes, and triggers—as well as its contents. You can also use the object browser to make simple changes to the description or contents of a table.

The home screen for the object browser displays a list of table names along its left side. Clicking a table name displays information about that table. For example, the screen for the EMP table appears in Figure 1-6.

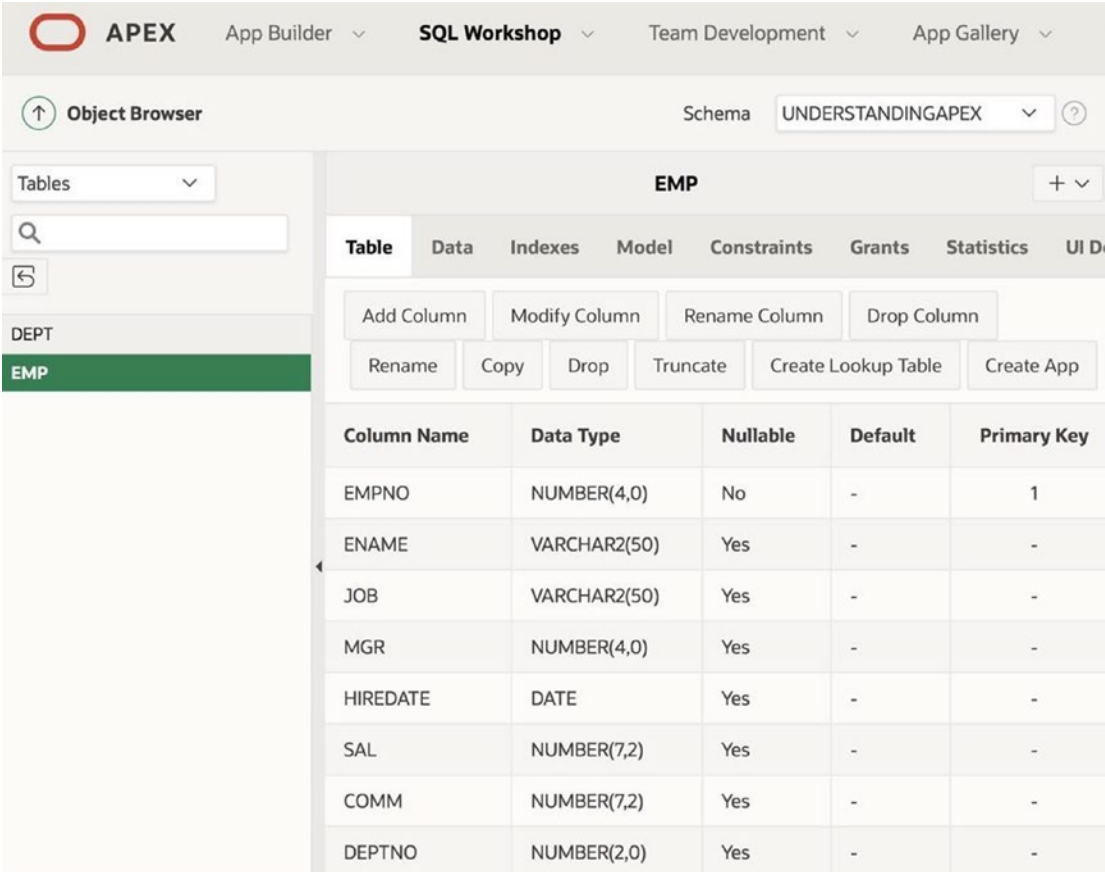


Figure 1-6. Viewing the EMP table from the object browser

The main portion of the screen displays information about each column of the table. Above this information is a series of buttons that let you modify it. As an example, recall that the introduction discussed the need for an `Offsite` column; let's add that column to the table now. Clicking the `Add Column` button displays a form for you to fill in the details of the new column. Figure 1-7 shows how I filled in this form.

The screenshot shows a dialog box titled "EMP" with a dropdown arrow. Below the title is a section labeled "Add Column". The form contains the following fields:

- Schema: UNDERSTANDINGAPEX (with a help icon)
- Table: EMP (with a help icon)
- * Add Column: OFFSITE (with a help icon)
- Preserve Case
- Type: CHAR (dropdown menu, with a help icon)
- Length: 1 (with a help icon)
- Precision: (empty field, with a help icon)
- Scale: (empty field, with a help icon)
- Nullable: NULL (do not require a value) (dropdown menu, with a help icon)

At the bottom of the dialog are two buttons: "Cancel" and "Next >".

Figure 1-7. Adding a new column to EMP

Clicking the `Next` button takes you to a confirmation screen; from there, click `Finish` to complete the action. The `EMP` screen should now display the new column.

Returning to Figure 1-6, observe the tab bar above the row of modification buttons. The `Table` tab is currently selected, which displays column information for the table. The other tabs show you other kinds of information and provide appropriate ways to view

and modify that information. For example, clicking the Indexes tab displays the current indexes for the table. Figure 1-8 shows the three indexes for EMP. Clicking the name of an index displays additional detail about that index.

EMP								
Table	Data	Indexes	Model	Constraints	Grants	Statistics	UI Defaults	
Create		Drop						
Index Name	Uniqueness	Columns	Status	Index Type	Temporary	Partitioned		
EMP_1	NONUNIQUE	MGR	VALID	NORMAL	N	NO		
EMP_2	NONUNIQUE	DEPTNO	VALID	NORMAL	N	NO		
EMP_PK	UNIQUE	EMPNO	VALID	NORMAL	N	NO		

Figure 1-8. Browsing the indexes of EMP

Finally, consider the Data tab, which displays the contents of the table. The top of this table appears in Figure 1-9. Note that there is a button to insert a new row and an edit link at the beginning of each row.

EMP									
Table	Data	Indexes	Model	Constraints	Grants	Statistics	UI Defaults	Triggers	Depend
Query		Count Rows		Insert Row		Load Data			
EDIT	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	OFFSITE
	7839	KING	PRESIDENT	-	11/17/1981	5000	-	10	-
	7698	BLAKE	MANAGER	7839	05/01/1981	2850	-	30	-
	7782	CLARK	MANAGER	7839	06/09/1981	2450	-	10	-
	7566	JONES	MANAGER	7839	04/02/1981	2975	-	20	-

Figure 1-9. Viewing the contents of EMP

Clicking a row's edit link displays a form for modifying it. Figure 1-10 shows this form for employee 7698. Clicking the Apply Changes button performs any modifications that may have been made to the column values; clicking the Delete button deletes the record.

The screenshot shows the 'Edit Row' form for the 'EMP' table. At the top, there are tabs for 'Table', 'Data', 'Indexes', 'Model', 'Constraints', 'Grants', 'Statistics', and 'UI Defau'. Below the tabs are three buttons: 'Cancel', 'Delete', and 'Apply Changes'. The main area is titled 'Edit Row' and contains the following fields:

Column	Value
Empno	7698
Ename	BLAKE
Job	MANAGER
Mgr	7839
Hiredate	05/01/1981
Sal	2850
Comm	
Deptno	30
Offsite	

Figure 1-10. Editing the contents of employee 7698

If you wish, you can edit this record, setting the value for `Offsite` to **N**. You can then proceed to edit the other records, setting their column value to **Y** or **N** as desired. Given the tediousness of this approach, however, it is easier to use the SQL command tool, which is discussed in the next section.

SQL Command Tool

Most of the actions that you can perform in the object browser correspond to one or more SQL statements. In effect, the object browser is merely a convenient way to formulate and execute simpler SQL statements. If you want to perform more complex activities, use the SQL command tool.

The SQL command tool divides the screen into two sections. You type an SQL statement or PL/SQL block into the top section, and the result appears at the bottom. Figure 1-11 shows the screen after executing the SQL statement `select * from EMP`.

The screenshot shows the SQL command tool interface. At the top, there is a 'Rows' selector set to 10, along with 'Clear Command' and 'Find Tables' buttons. Below this is a toolbar with navigation icons and a settings gear. The SQL command area contains the query: `1 select * from EMP`. Below the command area are tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is active, displaying a table with 14 rows of employee data. At the bottom of the results area, a message states: 'More than 10 rows available. Increase rows selector to view more rows.'

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	OFFSITE
7839	KING	PRESIDENT	-	11/17/1981	5000	-	10	-
7698	BLAKE	MANAGER	7839	05/01/1981	2850	-	30	-
7782	CLARK	MANAGER	7839	06/09/1981	2450	-	10	-
7566	JONES	MANAGER	7839	04/02/1981	2975	-	20	-
7788	SCOTT	ANALYST	7566	12/09/1982	3000	-	20	-
7902	FORD	ANALYST	7566	12/03/1981	3000	-	20	-
7369	SMITH	CLERK	7902	12/17/1980	800	-	20	-
7499	ALLEN	SALESMAN	7698	02/20/1981	1600	300	30	-
7521	WARD	SALESMAN	7698	02/22/1981	1250	500	30	-
7654	MARTIN	SALESMAN	7698	09/28/1981	1250	1400	30	-

Figure 1-11. Using the SQL command tool

APEX displays only 10 of the 14 employee records. The reason is due to the select list labeled Rows at the top of the figure, whose value specifies the maximum number of rows to display. By default, its value is set to 10; if you want more rows displayed, you must first select a larger number. This feature is intentional. By forcing you to explicitly specify the output size of your query, APEX protects you from yourself. Suppose, for example, that you execute a multi-table query in which you forgot to include the join conditions. The resulting output could easily have billions of records that, if not truncated, would cause your APEX session to be unusable.