An excerpt from *Beginning Oracle Application Express*
Part I

Introducing Oracle Application Express

Chapter 1: Welcome to Oracle!
Welcome to Oracle!

Welcome to the world of Oracle technology. The book you are holding will give you all you need to start creating powerful, flexible, highly productive applications for yourself and your users. And the software included with this book will give you the path to even greater capabilities through a broad range of Oracle software. You are going to like it here.

This chapter will introduce you to the core technology you will be using to create applications, Oracle Application Express. The rest of the book will walk you through the process of creating an application with a hosted version of this product, finishing up with information on expanding your environment with packaged applications and migrating existing applications to your new platform.

But first, a few basic questions and their answers.

What Do You Need?

You need to be able to respond to the needs of users. And you need to respond rapidly, which means your development platform of choice has to be highly productive. You also need to retain that productivity over the life of these applications, as demand for new applications and functionality never seems to slow down.

These requirements are table stakes — your development platform must deliver in these areas. You cannot even consider any options that would impede the velocity of your work.

Why Don’t You Have It?

But these table stakes are not, by themselves, enough. These requirements have led, over the years, to an approach that frequently used tools and platforms that were designed for personal use, such as Microsoft Access or spreadsheets. Initially, these user- and developer-friendly tools seemed like a great idea — a good developer could create applications very rapidly.

The problems arose after that initial deployment. Because of inherent limitations with the scalability and functionality of these tools, organizations ended up with a mess — hundreds of
different applications, and dozens of sources of data and versions of the truth. This jumble of systems led to enormous maintenance requirements, which created a lot of extra overhead. Sometimes these overhead requirements were so high that standard practices were just abandoned, leaving critical data unprotected and insecure — or led to the development of even more systems, compounding the problem even more. Couple this with increasing regulatory demands, and you have a recipe for an unmanageable mess that could potentially put mission-critical data at risk.

This brief description only summarizes the acute pain that you and your client base feel. What this list of problems doesn’t cover is the lost opportunity that comes from such a fragmented set of systems. The value of data grows as that data grows, through continued use and, more significantly, through integration with additional data to create a broader view of the overall organization. This value is trapped inside the welter of discrete systems. You could spend a significant quantity of resources trying to bring this data together with the same tools, but the proliferation of moving parts makes this a patchwork solution at best — if it’s even possible.

So the need is clear — you don’t want to sacrifice the advantages of rapid development in a productive environment, but your situation is growing more untenable by the day, as every new system only increases the problems springing out of a vast, unintegrated pool of systems.

Why Oracle?

You have two problems, and both of them require solutions. But why would you look to Oracle for both of those solutions?

Of course, Oracle is widely known for their enterprise strength database. You would expect systems built on the robust foundation of this database to be highly scalable, and for each database system to be able to support many different applications. But isn’t this sophisticated technology both complex and expensive?

As you will see throughout the remainder of this book, the answer to both of these questions is a resounding “No!” The folks at Oracle have indeed included a host of advanced features in their database and supporting technology for decades. This leadership has given their technology a significant advantage in the market, and in the ever-increasing ease of use of their products. Typically, Oracle introduced features into their technology years ahead of other products, and has refined and simplified the interface to that technology, in cooperation with their user community, while others strove to catch up. At this point in time, Oracle has simplified the use of their standard features to the point where you get their advantages with virtually no effort on your part. The exercises in this book are an illustration of this approach — although the application you will create is robust and flexible, with the ability to transparently scale to thousands of users, you will not have to perform any special configuration or management tasks to gain these benefits.

Even better, there is still a vast domain of advanced functionality in Oracle technology that you can choose to add to your application stack if you wish. From analytical capabilities to clustering solutions and everywhere in between, Oracle technology has all the features you need, regardless of how demanding your requirements are.

Oracle technology provides a great on-ramp to the highest levels of technical capabilities. You can easily use Oracle technology to create your systems, with all the power and functionality of Oracle products right there when you need it. But, you say, surely all this capability must be expensive, right?
Once again, the answer is no. In this book, you will be creating an order entry system, and all you will need is a browser. You will use a hosted Oracle environment that is available to you without any charges. If you want to create the same application using your own installed software, everything you need is included on the DVD for this book, whose contents are described later in this chapter.

In fact, the primary tool you will be using, Oracle Application Express, is included with every Oracle database as a no-cost option. Once you become acquainted with the productive potential of Oracle Application Express, you can create almost any application using this tool and deploy that application onto any production Oracle database.

In other words, the best reason to use Oracle technology is that there are no reasons not to — the question should be “Why not Oracle?” And, with this book, you have the perfect introduction to using that technology to implement your own IT solutions — at least, that is what this author hopes!

What Is Oracle Application Express?

The focus of this book, like its title, points to Oracle Application Express. What exactly is Oracle Application Express, also known as Oracle APEX?

Oracle Application Express is a rapid application development (RAD) tool built on Oracle technology. Oracle APEX runs in an Oracle database instance and comes as a no-cost option with all Oracle databases.

APEX runs entirely in a browser and does not require any software to be loaded onto your client machines. You develop your applications from a browser-based environment, as shown in Figure 1-1, and deploy your applications to a browser-based environment.

![Figure 1-1](image-url)
Part I: Introducing Oracle Application Express

Oracle Application Express is a declarative development tool. As you will see throughout this book, APEX development consists of assigning values for properties which describe the various components of your applications. The Oracle APEX environment does the rest for you.

For most components, you will use wizards to guide you through the process of creating those components. After this initial creation, you have access to these properties to modify or shape the operation of your applications, combining ease-of-use and productive maintenance in the same development tool.

All APEX development is data-driven. Tight integration with the Oracle database which hosts APEX means that your applications will naturally flow from your data designs. For instance, APEX applications automatically understand and implement relationships between tables, generating fully-featured applications without additional programming overhead.

Oracle Application Express is an all-in-one development tool. You use the same techniques to create forms, reports, and charts, as well as integrating these components together with navigation methods.

Your Oracle APEX environment gives you all you need to create virtually any application through an easy declarative development process. However, APEX is also extensible, so you can use HTML to modify the presentation of your APEX-generated client interfaces and Oracle’s standard PL/SQL procedural language to supply additional logical operations on the back end.

When you start to use Oracle APEX, you will quickly discover that you are not alone. The Oracle Technology Network (OTN) has a vibrant community of APEX users and experts, who regularly contribute leading to continual improvement in best practices and technical solutions. In addition, you will find a large collection of application packages that have already been created with APEX, readily available for you to integrate in with your own applications.

Last, but certainly not least, Oracle Application Express gives you a way to protect the investment you have in all those personal applications spread throughout your organization. You can upload data from an Excel spreadsheet with a few simple mouse clicks, moving crucial information into an Oracle database. You can even generate a complete application to interact with that data with a few more clicks, completing the task in less than 10 minutes.

Oracle also provides a migration tool to help you move your Access applications to Oracle APEX, migrating from a personal database with limited scalability to the world’s most powerful database.

Ready to start yet? This book contains all you will need to jump into the world of APEX-created applications. But first, a little bit about the application you will be creating in the remainder of this book.

What Are You Going to Do?

So far, so good. You’ve got a handle on Oracle Application Express, the tool you will be using to accomplish the task laid out in this book.

That primary task is to build an order entry system from scratch. You will be creating the data structures to hold the order information, loading data into those structures, creating maintenance tools for administrators to use to access and modify this data, and creating a browser-based user interface for your application.
The application itself will have all the features you would expect in a robust application:

- Ability to enter data, while ensuring that the data contains appropriate values
- Ability to search for data, according to user-specified conditions
- Ability to report on that data, both at the detail level and through higher-level approaches, such as graphs and charts
- An easy-to-use, attractive interface that will enhance user productivity

You can see a couple of views of this application in the following figures. Figure 1-2 displays one of the key pages in your completed application, where users will see their orders and the items in those orders, along with the ability to add new items and modify or delete existing items.

Figure 1-2

Figure 1-3 shows you an example of using the charting capability of Oracle Application Express. Although you cannot tell from this one image, the chart shown has built-in drill-down functionality, so that a user can get more detailed information on any of the sections of the pie chart with a simple mouse click.

Although this application is fully functional and scalable, you hopefully are interested in this book for more than just the order entry application you will build. The order entry application you will create was chosen as an example because you can learn most of the techniques you will need for your own future work. The primary purpose of this book is to teach you how to use Oracle Application Express to realize your own development goals, rather than arriving at the end product of a complete order entry application. But by the time you reach the end of the examples in this book, you will be able to extend the application you have built in almost any direction your requirements dictate.
You will build this application over the remainder of this book. You can also get script files, and instructions on how to use them at the web site for this book, www.wiley.com/go/greenwald.

How Will You Do It?

The format of this book is to teach you about Oracle APEX while iteratively developing the sample order entry application. Each chapter explores a basic area of functionality and the development techniques you will need to achieve that functionality.

Oracle Application Express runs within the Oracle database environment. You have a few choices on how you want to use APEX.

- The DVD that accompanies this book, which is described in the next section, includes a copy of Oracle Database 11g and Oracle Application Express. If you wish, you can load that software following the installation instructions that accompany the software. These installation instructions also include information on how to access your Oracle APEX software.
- If you already have an Oracle database within your organization, you could use Oracle Application Express from that environment. If APEX is not already installed on your server, the APEX home page at http://otn.oracle.com/apex has a link to download the software for your database. The only requirement is that your Oracle database be version 9.2.0.3 or higher.

This book is based on Oracle Application Express version 3.1.2. All the examples in the book have been tested against that version, so you should be fine if you are running that version or a later version.
You can also use a hosted version of Oracle Application Express. This book is designed for this remote access to APEX, although the examples should work with either of the other two options.

The next section will describe exactly how to sign up for your hosted Oracle APEX account.

How to Sign Up for a Hosted APEX Account

Oracle Corporation provides free hosted APEX accounts for your trial use. Signing up is an easy process that will take you less than 2 minutes.

Open your browser and go to http://apex.oracle.com, which is shown in Figure 1-4.

Click on the Sign up for an account link underneath the big red Login button, which will bring up the first page of the Signup Wizard, shown in Figure 1-5. This page is also your first view of an APEX application.

Click on Next to move from the Welcome page to begin the process.

Your first step, shown in Figure 1-6, is to identify the person who will be the administrator of the hosted Oracle APEX account. Enter your name and email address, and click Next.
Oracle Application Express can support multiple workspaces to give you and your organization the ability to segregate work on different applications. The next page of the Signup Wizard prompts you for a workspace. Enter a name you can remember, and click Next. If the workspace name you enter is not unique, you will receive an error message and be prompted to enter the name again.
In the next step, you are prompted for the name of a schema. Oracle Application Express exists within an Oracle database. An Oracle schema is a collection of tables and objects — similar to the concept of a database in Access. Give your schema an appropriate name, the easiest being the same name as your workspace.

The schema name must not exceed 30 characters, cannot contain spaces or quotation marks, and must be unique for the hosted version. If you select a schema name that violates any of these restrictions, you will get an error and be prompted to select another name.

Leave the default initial space allocation of 10MBs. Although an Oracle database can scale to handle many terabytes of data easily, you will not need that much space for your initial explorations of APEX. Click Next.

The final step in your signup process is to let the folks at Oracle know why you want to obtain a hosted APEX account. Enter your reason, which is to work on the exercises in this book, and click Next.

The final step should be familiar to web users. You are asked to enter a verification code, which ensures that a human is making the request, rather than an automated program. Match the text and click Submit Request, which will bring up the page shown in Figure 1-7.

![Figure 1-7](image)

This page informs you that you will receive an email once your request has been reviewed. This review process typically only take a few minutes. The email message you will receive contains a summary of your request and an activation link.

Click on the activation link in your email. This action will create a workspace for you in the hosted APEX environment. Once the workspace has been created, the hosted environment will send credentials for that workspace to your email address.
Part I: Introducing Oracle Application Express

- Click on the link to go to the login page for your hosted account, shown in Figure 1-8. Enter your workspace name and your user name, which is the email address. Check your email for your credentials, and enter the password sent to you. Click Login, which will bring you to Oracle Application Express, your development destination, shown in Figure 1-9.

And that’s it! You are ready to start creating APEX applications — which you will begin in the next chapter.
What Comes with This Book?

Before starting your development work, you might be curious as to the contents of the DVD that is included with this book.

The DVD is a collection of both software and collateral material, which gives you all you need for hands-on exploration of the vast functionality available with Oracle technology. The software includes:

- **Oracle Database 11g**: Standard Edition One for Windows, a developer edition of the world’s leading enterprise database.
- **Oracle Application Express**: The subject of this book.
- **Oracle SQL Developer**: A powerful tool used to interact with your Oracle database, which is also covered, in part, in this book.
- **Oracle Application Server 10g Standard Edition One for Windows**: Oracle’s powerful application server.
- **Oracle Business Intelligence 10g Standard Edition One for Windows**: A collection of powerful analytic tools to unlock the value in your Oracle data.
- **Oracle Warehouse Builder 11g**: A tool to help you create data warehouses to use for robust analysis of your Oracle data.

The DVD also includes white papers and seminars, both from Oracle Corporation and respected analysts, on a wide range of relevant topics. Finally, the DVD has a large collection of data sheets, seminars, and tutorials to guide you through the process of learning about the included software.

Script Files for This Book

The enclosed DVD does not contain the various files you will occasionally need throughout the course of this book. You can get those at the web site for this book, www.wiley.com/go/greenwald, which also includes a script to create the entire application from scratch.

Summary

Oracle Application Express will give you the power to create robust and scalable applications in an easy, highly productive environment. This book will walk you through the process of creating a sample order entry application, which will teach you the techniques you need to create your own applications.

You will be creating your application in a hosted APEX environment, but this book includes a DVD that includes all the software you will need for your own Oracle APEX environment and much more.
Part II

Building an Application with Application Express

Chapter 2: Jumpstart Oracle Application Express

Chapter 3: Extending Your APEX Application

Chapter 4: Validation, Calculation, Navigation and Lists

Chapter 5: Reporting and Charting

Chapter 6: Components Working Together

Chapter 7: Customizing User Interfaces

Chapter 8: PL/SQL

Chapter 9: Security
Jumpstart Oracle Application Express

Oracle Application Express, more commonly known as Oracle APEX or APEX, gives you a highly productive environment to create applications from start to finish. You can use APEX to create the structures for your data and quickly build interactive HTML-based pages to give users access to this data.

This chapter will get you right into the mix. In the next few pages, you will learn about the basic data design for the sample application used in this book, create the tables specified in that design, and create your first few pages to interact with that data. By the end of the chapter, you will have the bulk of the functionality you saw in the previous chapter ready for actual use.

So let’s get going!

Data Design

In the old days, our field was called “data processing”—a term that has been superseded by more descriptive phrases but that still carries an important message. Although applications and reports may be the visible aspects of our work as developers and business analysts, the entire system rests on a foundation of data.

Given this foundation, you should begin your process of development by creating a design for your data. A good data design is the natural outgrowth of a thorough understanding of the business processes that your application system will be implementing. You will need to understand not only the discrete pieces of data that you will require to store and manipulate but also how to organize this data into logical units, and how to relate those units to each other.

You are probably aware that Oracle is one of the leading enterprise databases in the world — if not the leading one. Based on this heritage, Oracle Application Express is built to leverage good
Part II: Building an Application with Application Express

data design. In order to use APEX most effectively, you should fully understand the data requirements of your business scenario before beginning, which is the focus of this chapter.

Application Needs

The application we will be building is an order entry system. The multiple pages that make up this application will have to deal with a number of data areas, including:

- Orders, which will include data identifying the overall order
- Order detail lines, which will indicate the quantity and pricing of a particular product within an order
- Products, which will carry descriptive information about the products offered by your company
- Customers, those wonderful people who will be ordering products

You will need to collect and process data about these four entities in the application that you will be designing for this book. Later in the book, you will be adding some other groups of data to support the operation of your application, but these four groups of data are all you need to create the main portion of your functional application.

Data Structures

Oracle Application Express is built on top of an Oracle database. The Oracle database is a classic relational database. In a relational database, groups of data are implemented as tables. A table is the core component of a database. All data activity centers on tables; you read and write all individual pieces of data as operations on rows within one or more tables.

Depending on your background, you might be more used to referring to rows as records. A row within a table is the relational database equivalent of a record within a file or personal database. Since Oracle Application Express is built on top of Oracle, and since some readers may be coming from a relational background, this book will use the terminology of tables and rows.

Creating the data structure for this application will require the creation of four tables to implement the four functional data groupings listed above.

Figure 2-1 shows the four tables, as well as the relationships between them.

You can see the four tables, as expected, as well as lines connecting those tables. The lines indicate relationships between the tables. The tables connected by straight lines — Orders and Customers, Order Items and Products — have what is called a one-to-one relationship, with each row in a table being connected to a single row in the other table. One set of tables — Orders and Order Items — are connected by a line with what looks like a fork at one end. This line represents a one-to-many relationship, with the fork indicating the many end of the relationship. Each Order row can have one or more Order Items rows.

As you will see by the end of this chapter, the data groupings and relationships within this simple data design are more than a simple way of grouping data. Oracle Application Express uses the structure implemented in your data design to automatically create powerful functionality.
Your First Page

Data design should always come first in your development process, since no amount of clever code can correct for an invalid or inefficient data design. For the purposes of the sample application you will be creating in this book, the four tables in the current data design are enough to get started creating your application. By the end of this short section, you will have moved from the concept illustrated in the data design to a highly functional application. You understand your data design, so you can start right in on creating your application.

Creating a Table

You have a high level data design. The first step towards a working application is the creation of a table. If you are creating an application to work on data that already exists in some form, you probably want to also populate the table with that existing data as soon as you create it.

APEX gives you several ways to quickly create and populate the tables that will be used in your application. In this chapter, you will use the following methods:

- Automatic table creation and population directly from a spreadsheet
- Manual creation of table structure and population from a spreadsheet
- Automatic creation of related tables from data in an existing table
There are other methods you can use to bring data into the Oracle database that works with APEX, including standard Oracle utilities for loading and unloading data, as well as other potential data sources, such as text or XML files. You can explore these other options through the online help in APEX.

The fastest and easiest method for table creation is to import both structure and data from a spreadsheet. You are going to use this method for one of the core tables for the application, Orders.

**Try It Out  Importing a Table from a Spreadsheet**

APEX makes creating and populating a table from a spreadsheet easy — just a few clicks and you are done. This example will use the ORDERS.csv file from the sample files for this book to create the core table for the sample application.

You can get all the files needed for the examples in this book at the web site for the book – http://www.wiley.com/go/greenwald.

1. Log into your APEX-hosted account to bring up the page shown in Figure 2-2.

![Image of APEX page](image)

**Figure 2-2**

2. Click on the Utilities icon, and use the cascading menus to select Data Load/Unload, and then the Load choice, as shown in Figure 2-3.

3. Click on the Load Spreadsheet Data icon in the Load page, which will bring up the page shown in Figure 2-4.
Chapter 2: Jumpstart Oracle Application Express

Figure 2-3

Figure 2-4

Excerpted with permission from Beginning Oracle Application Express, ISBN 9780470388372.
4. On this page, accept the (Load To) New table choice, but change the Load From choice to Upload File. Click Next once you have made this change.

5. The next page, shown in Figure 2-5, prompts you for the name of the file to use as a data source. Click Browse and locate the ORDERS.csv file. The default values for other choices on this page can be left with their default values for this file. Click Next to bring up the next page.

6. The next page, shown in Figure 2-6, gives you a lot of information about the comma-separated file you indicated in the previous page. You can see that the column names, data types, and even first set of values slated to be loaded into the table are shown. All that is left for you to do is to add a name for the tables, which should be ORDERS, and click Next.

   You can also see a dropdown box asking for a schema. In Oracle terms, a schema is equivalent to a database — a set of tables that can be directly accessed, without further name qualification. You defined a schema when you created your online account in the last chapter.

   The next page, shown in Figure 2-7, has a few choices that control how APEX will implement logic for this particular table. In relational terms, every row in a table must have a primary key. The primary key is used to identify the particular row of the table. On this page, you identify either a new or existing column to hold the primary key value. APEX also asks for the name of a constraint to use to implement the primary key. A constraint is simply a database specification that indicates, in this case, that the particular column is a primary key.

   You will learn more about constraints in Chapter 4.
Chapter 2: Jumpstart Oracle Application Express

Figure 2-6

Figure 2-7
Finally, the page asks if you want to use a sequence to generate primary keys. APEX applications require that each row in a table will have a unique primary key value. The Oracle database includes an object that is ideal for generating unique values—a sequence, which is roughly equivalent to an Autonumber data type field in Access. A sequence object performs a simple task—each time you ask the sequence for a number, the sequence provides the next consecutive number.

The page shown in Figure 2-7 allows you to specify the creation of a new sequence to generate a primary key, use an existing sequence, or not generate a sequence, which requires you to provide your own unique primary key value.

7. For this data load, accept the default value of Create new column for the Primary Key From: choice, but change the name of the New Primary Key Column to ORDER_ID. You can leave the other choices with their default values and click Load Data.

8. After the data load has completed, you will be presented with a page that lists the most recent data loads into the repository, the Oracle database that underlies your APEX environment.

With these few steps, you have both created a table and loaded data into your Oracle database.

**How It Works**

You can now go and take a quick look at your data through the standard APEX data-browsing tools.

1. Start by clicking on the SQL Workshop tab in the set of tabs in the upper-right area of the page, which will bring up the page shown in Figure 2-8.
2. As with your initial step above, click on the down arrow in the Object Browser icon, and then select Browse and Tables from the cascading menus.

The Object Browser has a picklist at the top of the left-hand panel that allows you to select the type of database object you want to view, as well as giving you the option to search for instances of that object based on a search string. Below these two entry fields is a list of all the objects that match the type and search criteria.

3. Click on the ORDERS table to bring up the Object Browser window, as shown in Figure 2-9.

![Figure 2-9](image)

You can see you have a lot of options available for a table in the Object Browser. If you are familiar with SQL data definition language (DDL), you will recognize the lower row of options as a part of standard SQL syntax for the overall table. You will be using some of these options in this and subsequent chapters. Right now, you want to take a quick look at the data in the ORDERS table to make sure that the import from the spreadsheet succeeded as described.

4. Click on the Data option in the top line of choices in the right-hand panel to bring up the page shown in Figure 2-10.

This page displays the initial set of data for the table. If there were more rows, you would have the option to scroll through the rows. You can also see that there is a link labeled Download in the lower-left corner of the data display window. This link gives you the ability to download the data in the table into a .csv file — one of several ways to extract data back out of the Oracle database underlying the APEX environment.
Figure 2-10

To the left of each row of data, you can see an icon that will take you to an edit page for that row of data. You can click on it to see the editing interface, but with APEX, you can quickly create a page in an application to accomplish the same tasks with just a few clicks — as you will do in the next section.

Creating an Application

Now that you have data in your database, the next step is to create a way to manipulate that data. In this section, you will create an application with half a dozen clicks and a tiny bit of typing.

Try It Out Creating an Application

You will use the first of many Oracle APEX wizards to create an application.

1. Click on the Home tab of the APEX environment, and then click on the down arrow in the Application Builder icon to select Create Application. This action will bring up the page shown in Figure 2-11.
Chapter 2: Jumpstart Oracle Application Express

Excerpted with permission from Beginning Oracle Application Express, ISBN 9780470388372.

On this page, you select whether you will be creating an application from scratch, creating it from a spreadsheet, or installing one of the demonstration applications. Creating an application from a spreadsheet essentially combines the steps you took in the previous section with the steps in this section. In addition, you have the ability to define a few standard reports. In the interests of combining the rapid creation of an application with the process of learning how to use APEX, this chapter has taken the slightly longer route.

2. Select the Create Application option, and click Next to bring up the page shown in Figure 2-12.

The main requirement of this page in the Application Creation Wizard is to give a name to your new application. You also can see that the application is assigned a number. An APEX application is created with code stored in the underlying Oracle database. Each application is identified by a unique number.

For the purposes of this book, you should just accept the assigned number.

You can give the application any name you like, but the examples in this book will refer to this application by the name of Book Sample.

You can also see that you can create application design templates, which can be used to create default applications that use a structure you have designed.

3. Give the application a name and click Next to bring up the page shown in Figure 2-13.
Part II: Building an Application with Application Express

Figure 2-12

Figure 2-13

Excerpted with permission from Beginning Oracle Application Express, ISBN 9780470388372.
You can accomplish a lot on this page. You can specify any number of pages that will be part of your application with just a few clicks. Since you only have one table available right now, you will only create one set of pages but with a more fully populated database, you could create fairly sophisticated application systems with the point-and-click methods of this page.

4. Select the Report and Form option in the Add Page box.

5. Select the ORDERS table as the Table Name, and click Add Page. The result is shown in Figure 2-14, with two pages, the report and the edit form, added to the page list above.

![Image](https://example.com/image.png)

**Figure 2-14**

6. Click Create.

That’s it! You will be presented with the Summary page shown in Figure 2-15, listing the defaults applied to your application. These defaults include specifying one level of tabs, a standard method of authenticating users, and a standard template. You can change any of these, or you could have selected them during the application creation process by clicking Next on the Create Application page shown in Figure 2-14. You can also access the edit pages for these choices by using the Previous button on the page.

7. Click Previous, which will bring up the page shown in Figure 2-16.
Part II: Building an Application with Application Express

Figure 2-15

Figure 2-16
This page gives you the option of changing the template for the application, which provides the basic look and feel for the end user. The examples in this book will use Theme 13. Later in this book, you will learn more about templates and how to use them.

8. Select Theme 13 and click Next.

9. Click Create, which will create the application and take you to the development home page for the application, as shown in Figure 2-17.

That was pretty easy, and, as you will see in the next section, the application you created has a significant amount of built-in functionality. Oracle APEX provides you with both wizards (to speed the development process) and loads of automatic functionality (to reduce the amount of development you will have to do yourself).

**How It Works**

The ease of application creation is nice, but the default functionality provided in that application is even nicer.

To run your new application, you can click on the traffic light icon, either in the big button at the top of the page or as the little icon, just under the main tabs, to the right of the Page 1 label.

1. Click on the traffic light icon to run the application, which will bring up the page shown in Figure 2-18.
If you are coming from the world of single-user databases and applications, you might be a bit surprised to see a login page for a simple application like the one you just created. But APEX is designed from the ground up to support multiple users. The designers of APEX know that multi-user applications require security, so APEX applications leverage the built-in security of the Oracle database for all applications.

You could create an application for public use that did not require any login to access.

The page shown in Figure 2-18 also has a number of links at the bottom of the page in what is called the Developer’s Toolbar. As its name implies, these links are only visible to developers, not to end users running the application. These links make it easy for you to quickly move between the runtime environment of the application and the development view of the application, and to gather information about the runtime environment. As a developer, you will be using these links frequently throughout the cycle of development, and you will learn more about them over the coming sections and chapters.

2. Enter the name and the case-sensitive password you use as a developer, and click Login to bring up the page shown in Figure 2-19.

The page you see has everything you would expect and quite a bit more. The page includes the look and feel from the template you chose in the previous exercise, and a tab with the name of the table displayed by the page. The data in the \texttt{ORDERS} table is shown in a list, with an icon on the left side of the row that will take you to an edit page for that row.
The page also includes built-in search capabilities, with a text box for entry of a search string. This search box does a little bit more than your everyday average search function.

3. Enter la into the search box, and click Go to bring up the page shown in Figure 2-20.

Now, you can see that the search function limited the display to those records that had the character “la” somewhere in their columns. You could have limited the number of rows returned with the selection list to the right of the search term field. But notice that the search term now appears below the search header, along with an icon to indicate that the search term limits the rows, a check box to indicate whether the search term is used, and an icon to delete the search term. There is a lot of power in this little display, which you can see by adding another search condition.

4. Enter b into the search box, and click Go to bring up the page shown in Figure 2-21.

You can now see that both search conditions are listed and have been applied to limit the results. You can simply uncheck the check box next to either condition to remove that criterion from the search. You could even collapse the search conditions by clicking on the little minus sign to the left of the conditions. Although these types of multiple search conditions are not that impressive with a single table with a handful of rows, you can imagine how useful this could be in a production environment.
Figure 2-20

Figure 2-21
5. Click on the column heading labeled Order Timestamp to bring up the popup shown in Figure 2-22.

![Figure 2-22](image)

The popup you see brings a lot more functionality into every interactive report. The icons at the top of the popup allow you to sort based on this column, by either ascending or descending values. You can hide the column, create a control break on the column, or get information about the specification for the column. You can even select a search condition based on actual data values. Since APEX knows that the column is a date field, you are presented with the choices based on the age of the value.

6. Click on the value Last Year in the popup.

Notice that the condition you just specified with a single click now appears as one of the search conditions at the top of the page. All this functionality is automatically built into all APEX interactive reports. You will be learning much more about reports in Chapter 5.

7. Click on the Edit icon on the left-hand side of a row to bring up the page shown in Figure 2-23.
Figure 2-23

The edit page for the row presents you with all the editable columns for the row. The same page would appear if you clicked Create on the previous page but without values for any of the fields.

You might be thinking that there could be a more user-friendly way of editing some of these values, such as picklists or lookups. Don’t worry, you will be learning how to modify this default edit page in the next chapter.

You have the option of canceling or applying changes you make to the row in this page, or deleting the selected record.

8. Click Cancel to return to the previous page.

9. Return to the development environment by clicking on the Home link at the bottom of the page.

Expanding Your Application

The basic “work” of creating an application framework and the first few pages is done. Now you can add more data and pages to the application. For the purposes of learning about APEX development while building this sample application, your next task will be to add a table with data related to the ORDERS table.
**Importing Order Items**

The `ORDERS` table contains information that pertains to the overall order. The `ORDER_ITEMS` table contains details about individual products ordered. It makes sense that the `ORDER_ITEMS` table would somehow be connected with the `ORDERS` table. In this section, you will import data for order lines, establish the connection to the `ORDERS` table, and see how APEX automatically incorporates that relationship into the default pages you can use to access both the master and detail tables.

**Try It Out  Importing and Linking Another Table**

The order entry application that is the focus of this book requires more than one table. Now that you have imported the `ORDERS` table, performing the same operation for the `ORDER_ITEMS` table is simple. But once you create the table through the import process, you can then link the data in that table to related rows in the `ORDERS` table.

1. Go to the Utilities page, and then to the Load page under the Data Load/Unload icon.
2. Select the Load Spreadsheet Data icon.
3. Change the Load From: choice to Upload File, and click Next.
4. Select the `ORDER_ITEMS.csv` as the source, and click Next.
5. Give the new table the name of `ORDER_ITEMS`, accept all the defaults, and click Next.
6. Accept the default of New Primary Key column for the Primary Key From: option, but change the name of the primary key to `ORDER_ITEMS_ID`, then click Load Data.

Once again, the `ORDER_ITEMS` table will require a primary key, like all tables in an Oracle database. But you will have to add another constraint to this table to establish the relationship between `ORDER_ITEMS` and the `ORDERS` table.

7. Return to the SQL Workshop page.
8. Select the Browse and Tables choices under the Object Browser icon.
9. Select the just imported table `ORDER_ITEMS`, and select Constraints from the choices at the top of the right-hand panel.
10. Select the Create option, to bring up the page shown in Figure 2-24.

This page allows you to create any of the Oracle supported constraints. Oracle APEX has already created primary key constraints for the `ORDERS` and `ORDER_ITEMS` tables as part of the import process. You will now create a foreign key constraint.

A foreign key constraint links two tables in a parent-child relationship. The foreign key column is in the child, and the connection links to the primary key of the parent. As you will see in a few steps, the APEX environment is aware of foreign key constraints and implements default options to handle them.
11. Change the Constraint Type to Foreign Key, which will change the appearance of the page to look like the completed page shown in Figure 2-25.

The Constraint Type specifies the action taken on the child data if the parent row is deleted. You can prevent deletes of parents that have children, automatically delete the children of a parent if the parent is deleted, or simply set the value of the foreign key to NULL if the parent is deleted. For the purposes of this demonstration application, you should choose the most common implementation of the constraint, but in the real world, this decision would be based on the business process the data is supporting.

12. Choose Cascade Delete for the Constraint Type, which would cause the related rows in the ORDER_ITEMS table to be deleted when their associated ORDERS row is deleted.

13. Select ORDER_ID for the Foreign Key Column(s).

14. Select ORDERS for the Reference Table Name and ORDER_ID from the Reference Table Column List.

15. Click Next and then Finish on the next page.
Chapter 2: Jumpstart Oracle Application Express

Excerpted with permission from Beginning Oracle Application Express, ISBN 9780470388372.

Figure 2-25

You have created a foreign key constraint. The constraint will work in this case because the value stored in the ORDER_ITEMS table matches the value in the primary key of the ORDERS table. The foreign key constraint is implemented within the Oracle database, so you can see it in the Object Browser. But the real payoff comes when you create a page for the ORDER_ITEMS table, and you can see how APEX handles the defined relationship.

Creating Another Page

It won’t take long to generate another page in your APEX application to handle interactions with both the ORDERS and the ORDER_ITEMS tables.

Try It Out  Creating Pages for Master-Detail Tables

1. Return to the Application Builder page, and double-click on the Book Sample application you just created, which will bring up the page you saw before, as shown in Figure 2-26.
At the top of this page, you can see four icons used to run your application and deal with administrative tasks connected with the application. You will learn more about the three icons for Supporting Objects, Shared Components, and Export/Import later in the book.

You can also see the pages that were automatically generated for you when you created the application. You could double-click on any of these pages to modify its appearance or functionality, which you will be doing in the next chapter. Right now, you want to create a new page for the application.

2. Click Create Page to bring up the page shown in Figure 2-27, which is where you will begin the process of creating a new set of pages to handle the ORDERS and ORDER_ITEMS tables together.

3. The page shown in Figure 2-27 gives you a number of choices for the type of pages you want to create. Select the Form choice and click Next.

4. The next page gives you a number of choices for the type of form you want to create. Select Master Detail Form and click Next, or simply click on the Master Detail icon.

You will have to go through a few steps to completely define the functionality implemented in the master-detail forms. Although each step is quite easy, the left-hand panel displays a list of the steps, and where you are in the process, to keep you informed of your progress.

5. The next page, shown completed in Figure 2-28, collects information about the master table. Select the ORDERS table from the selection list.
6. Select all the columns in the Available Columns list, and click on the right arrow button to move them to the Displayed Columns list. Click Next.

7. The next page asks you for the same information about the detail table. Select the ORDER_ITEMS table in the selection list and move all the columns to the Available Columns list. Click Next.

The next two pages will confirm the source of the primary key value for each of the tables. Both tables use a sequence to produce the primary key.

8. On the next page, shown in Figure 2-29, select Existing Sequence and the ORDERS_SEQ sequence from the selection list that appears. Click Next.

![Figure 2-29](image)

9. On the next page, make the same choice, except select the ORDER_ITEMS_SEQ sequence from the selection list. Click Next.

10. The next page, shown in completed form in Figure 2-30, allows you to specify how you want the user to navigate through the rows of the master table. For this example, you want to have a report on the rows of the master table, so leave the choice for Include master row navigation set to the default of Yes. Select ORDER_ID as the Master Row Navigation Order to allow navigation by ORDER_ID, and click Next.
11. The next page gives you the option of editing detail records in a multi-row table on the same page as the master record, or on a separate page. You should change the value to Edit detail on a separate page, since this is the most common choice for web-based applications. Click Next.

12. The following page, shown complete in Figure 2-31, gives you the option to change the default text associated with the pages APEX will create. The upper panel lets you set the title for the pages and regions within the pages. Set the Page Title and Region Title for the Master form to Orders, the Detail page title to Order Items, and the region titles to Edit Order Items and Order Items Detail; set the Page Title for Detail 2 to Order Items and the region title for that page to Edit Order Items.

Your APEX application can use breadcrumbs, a series of links in the top left hand corner of the page representing the hierarchical path to this page, to provide easy navigation through pages. You can see breadcrumbs at work in the APEX development environment, because the APEX development environment is written in APEX itself.

13. Set the Entry Name to Orders, and click Next.
Part II: Building an Application with Application Express

Figure 2-31

The last page of attributes for the master-detail pages asks how you want to use tabs to navigate to the page. Since these pages involve the ORDERS table, you will want to select that tab for the pages.

14. Select the third option to use an existing tab set and tab, and click Next.

15. On the next page, select the T_ORDERS tab, the only one available, and click Next.

16. The final page confirms the choices you have made and allows you to confirm the creation of the pages. Click Create to take this action.

When you have completed the process, you will see a success page with two icons. In the next section, you can see the functionality you have implemented with this small set of clicks.

How It Works

The pages you have just created do more than the simple report and edit pages for the ORDERS table.

Click on the Run Page icon. If you have logged out of your browser session since the last time you ran the application, you will be prompted to log in to the application again. The initial page, shown in Figure 2-32, with the report for the ORDERS table, is not that impressive. In fact, it’s just a simple report, not even as cool as the interactive report you created last time. But the cool stuff begins once you go to edit an order.

Does the interactive report on the ORDERS table seem nicer than the simple report that is part of the master-detail pages? Later in the book, you will learn how to substitute the interactive report for this report.
1. Click on the Edit icon for the first order, which will bring up the page shown in Figure 2-33.
You can see a lot more in the way of buttons and other elements on this page. At the top of the page are buttons that give users the ability to delete records, save changes, and scroll through the master records. You can also see that the date field of Order Timestamp has a little calendar icon next to it, which will bring up a date picker.

Below the fields for the ORDERS table is an area where you can display one or more records of order items associated with this order. There is a button that allows you to create a new row. You can display multiple detail rows or add rows until you fill up the space allotted for the detail table.

2. Click on the Create pushbutton to bring up the page shown in its completed state in Figure 2-34. Set the Product Id to 3, the Unit Price to 40, and the Quantity to 3. Click on Create to add the new record and return to the page for the main Order.

Figure 2-34

You can use these master-detail pages to edit values for either of the tables, or to delete rows from the ORDER_ITEMS table by going to the edit page for the existing record and deleting the row with the Delete button, which will be present.
Rounding Out Your Data Set

Before moving on and modifying your basic application in the next chapter, you still have to load one final table to complete the initial loading of your data set. Before jumping into this task, you should consider the implications of the foreign key you created in the previous section.

For that foreign key constraint, the value in the ORDER_ID column for the ORDER_ITEMS pointed to the value in the primary key of the ORDERS table, ORDER_ID. But you might remember that this column was created automatically when you loaded the table from the spreadsheet. Why did the values in the ORDER_ITEMS column match the appropriate values in the ORDERS table?

Well, this data is sample data, specifically designed for demonstrations like the ones in the book. The values for the ORDER_ID in the ORDERS table were nicely created to be sequential from 1 to 11, and the values in the corresponding column in the ORDER_ITEMS table matched that simple scheme. In the real world, however, you probably won’t be so lucky. In fact, a common task might involve breaking one spreadsheet up into more than one table, and loading the same column value into both tables involved in the foreign key restraint.

On one hand, this scenario is easy to deal with. When importing the table, you simply say that the value for the primary key is in an existing column. The issue arises with the use of a sequence to assign further values automatically. By default, the sequence defined as part of the loading process starts numbering from 1. This default choice means that you will eventually run into a primary key conflict, when the sequence tries to assign a value that already exists as a primary key, which must be unique.

There is an easy way to avoid this issue — create a sequence before loading the table, and ensure that the starting number for the sequence is greater than the maximum number currently existing in the primary key column of the table you are going to create.

You will take this minor detour as you load the table containing information about products.

Try It Out Create a Sequence and Use It for a New Table

You will start out your new task in the now familiar Object Browser within the SQL Workshop.

1. Return to the APEX environment main page by clicking on the Home link at the bottom of the runtime page.

2. Using the dropdown menus from the SQL Workshop icon, select Object Browser, Create, and Sequence to bring up the page shown in Figure 2-35 with the values you will enter.

3. Give the sequence the name of PRODUCTS_SEQ, a starting value of 12, a minimum value of 12 and a maximum value of 99999. Click Next and then Create.

You don’t actually have to assign a minimum and maximum value to the sequence. If no values are present, the defaults are 1 for minimum and a string of 9s for the maximum.
Now that you have a sequence created, you can use it when you are loading your new table.

1. Click on the Utilities tab and use the drop down menu to select the Load choice from the Data Load/Unload menu.

2. As you did before, select the Load Spreadsheet Data icon.

3. Select the Upload file choice, and click Next.

4. On the next page, select the `PRODUCTS.csv` file with the browser, and click Next.

5. On the next page, give the table the name of `PRODUCTS`. Note that the first field listed in the display is the `PRODUCT_ID` field, which contains the current value for the primary key for the product row.

6. Click Next.

7. On the next page, you want to change two default choices. First, change the Primary Key From: choice to Use an existing column. Leave the Primary Key selection list on the `PRODUCT_ID` column, which is the default.

8. For Primary Key Population — Choice, select Generated from an existing sequence, and then select the `PRODUCTS_SEQ` you just created from the Sequence selection list.

9. Click Load Data to complete this operation.
The next step is to create a foreign key constraint linking the **PRODUCTS** table to the **ORDER_ITEMS** table.

1. Go to the SQL Workshop by clicking on the appropriate tab.
2. Use the dropdown menus to browse a table.
3. Select the **ORDER_ITEMS** table, since this table will be referencing the **PRODUCTS** table to get more information about the products entered as part of the order line.
4. Click on the **Constraints** label over the table, and then click **Create** to start the process.
5. Change the name of the constraint to **ORDER_ITEMS_PRODUCT_CON**, since you already have a constraint with the default choice offered.

How can you tell you have a duplicate constraint name? You may have noticed that many of the pages in the development environment have shortcuts at the bottom of the page — in this case, Existing Columns and Existing Constraints. You can simply click on one of these labels to display a list of the related values.

6. Select Foreign Key from the top selection list, Cascade Delete from the Constraint Type, **PRODUCT_ID** as the Foreign Key Column(s), **PRODUCTS** as the Reference Table Name and **PRODUCT_ID** from the Reference Table Column List, to end up with the values shown in Figure 2-36.
7. Click **Next** and then **Finish** to create the new constraint.

---

**Figure 2-36**
You have done everything you need to do to create a relationship between the `ORDER_ITEMS` table and the `PRODUCTS` table, although you will not be linking them together until you learn a bit more in the next chapter.

The last step is to create a page for the `PRODUCTS` table, for which you will use a different wizard to create another set of pages.

1. Return to the main development page by clicking on the Application Builder tab.
2. Double-click on the Book Sample application to add more pages to the application.
3. Click Create Page.
4. On the following page, select the Form option and click Next.
5. Select Form on a Table with Report, and click Next.
6. Accept the default schema of BOOKSAMPLE (or whatever you named your schema), and click Next.
7. On the next page, select the `PRODUCTS` table from the selection list and click Next.
8. You can accept the defaults on the following page, although you might want to change the Page Name and Region Title to a more user-friendly version with initial caps instead of all caps. Click Next.

The next page allows you to specify how this report and form will use tabs. This application already has a default tab set, with a tab for Orders, which was used for both previous sets of pages. You will want to use the same tab set but create a different tab within the set.

9. Change the Tab Options: choice to Use an existing tab set and create a new tab within the existing tab set. Accept the default Tab Set, and enter Products as the name of the new tab. Click Next.
10. The next page shows you the columns available for the report. Select all the columns except the `PRODUCT_ID` column, which is only used internally by the application.
11. Click Next.
12. Accept the default icon edit choice, and click Next.
13. Change the Page Name and Region Title to Products, and click Next.
14. Accept the default primary key on the next page, and click Next.
15. On the next page, select Existing sequence, and PRODUCTS_SEQ from the Sequence selection list. Click Next.
16. Select all the columns to include in the form, and click Next.
17. Accept the defaults to display the Insert, Update, and Delete buttons, and click Next.
18. The final page summarizes the choices you have made. Click Finish to create the pages.
19. You can play around with the Products page, as shown in Figure 2-37, by clicking on the Run Application icon.

Figure 2-37

Summary

In this first hands-on chapter, you have come a long way. You have:

- Created the core tables for the application, ORDERS, ORDER_ITEMS and PRODUCTS and filled them with data.
- Created the application and added a few pages to allow your users to interact with this core data.
- Started to learn your way around the APEX development environment.
Keep in mind that this chapter has been designed to teach you how to use APEX, rather than focusing on creating applications in the most productive way. You could have created all the pages for this application when you created the application, a process that would have taken you less than five minutes to get where you are now in terms of application development.

In the next chapter, you will learn more about the capabilities of Oracle Application Express and add more functionality to your existing application.
Reporting and Charting

As developers, we sometimes get caught up in the fun of creating applications — applications that implement all kinds of logic, that perform cool tricks to help users enter and update data, and that are pleasing to the eye and the mouse.

But as we focus only on applications, we tend to concentrate on entering, updating, and deleting data. These processes are, in the real world, typically just precursors to unlocking the real power of collected data through reports and charts.

Oracle Application Express makes it as easy to generate powerful and flexible reports as it does to create data access pages. In this chapter, you will create some more sophisticated reports and explore the ways that users can interact with the reports.

More Data Design

The purpose of this chapter is to show you some of the wonderful things you can do with reports and charts, but the data model currently in place is less than challenging. You have one real master-detail relationship, with some other foreign keys used to implement data integrity.

Before you can really see what APEX reports and charts can do for you, you will have to add one more table to your schema. You will create the basic table for customer information using SQL Workshop, and then you will expand that table structure and use a script to load additional data.

Try It Out Create Another Lookup Table

You have already created a lookup table for CATEGORIES and taken the steps required to take advantage of the table in your application. The first part of adding customer information to your schema follows the same steps:

1. Go to the SQL Workshop area of the APEX development environment.
2. Use the cascading menus from the Object Browser to browse the tables.
3. Select the ORDERS table, and click on Create Lookup Table.

Excerpted with permission from *Beginning Oracle Application Express*, ISBN 9780470388372.
On the next page, you will see that, by default, you have been presented with a list of all character columns for the table.

4. Select CUSTOMER_NAME for the Column, and click Next.

The next page prompts you for a name for the new table and the new sequence that will be used to generate IDs for the table. You will want to change these to make the names more in keeping with the naming standards in your schema.

5. Change New Table Name to CUSTOMERS and New Sequence to CUSTOMER_SEQ. Click Next.

6. On the final page of the wizard, click Finish.

You have created a table containing a code and label for each customer, dropped the column that used to contain the CUSTOMER_NAME from the ORDERS table, added a column to hold the new value of CUSTOMER_NAME_ID, populated it with the right values, and created a foreign key to link the two tables together.

Accepting Defaults

You may not really like the fact that the new column in ORDERS is called CUSTOMER_NAME_ID rather than CUSTOMER_ID. You could have gone through each of these steps manually to gain full control over the naming of the new column. Of course, the best practice would be to design your data structures with these types of relationships in place from the beginning, but you may not have that option in all cases, especially when you are moving data from another source, a topic that will be covered later in this book.

You are now ready to represent the new functionality in your application.

Try It Out Adding a Lookup Table to Your Application

You have also done the steps in this section before, so you can go through them quickly this time through.

1. Return to your application and click on the master-detail page.

2. Click on the item for CUSTOMER_NAME.

You will want to make three changes to the attributes of this field.

3. In the Name section of the page, change the Display As value to Select List by clicking on the Select List link below the field.

4. Click on Source at the top of the page to isolate the attributes of the item.

You can see that this item is bound to the CUSTOMER_NAME column in the ORDERS table. If you tried to run this page now, you would get an SQL error, since this column no longer exists in that table. You have to change the source of the item, and you could also change the name of the item to reflect the new source. For the sake of simplicity, the examples in this book will leave the name of the item as it stands.
5. Change the Source value or expression to CUSTOMER_NAME_ID, the name of the new column in the ORDERS table.

6. Click the LOV button to bring up the section of the attributes concerning LOVs.

7. Click on the Create Dynamic List Of Values link at the bottom of the section.

8. In the wizard for creating dynamic LOVs, accept the default schema and click Next.

9. Select the CUSTOMERS table in the next page, and click Next.

10. Select CUSTOMER_NAME as the Display Column and CUSTOMER_NAME_ID as the Return Value, and click Next.

11. Accept the generated SQL, and click Finish. Click Apply Changes.

You have added the LOV to the edit page for the ORDERS table. As with the use of the lookup table in the previous chapters, you will also want to add the new column to the initial report by modifying the SQL source for that report.

12. Return to the main development page for the application, and click on the initial interactive report created for the ORDERS table. If you are doing the exercises for this book in order, this page should be Page 1.

13. Click on the Region Definition tab.

14. Change the SQL in the Source section from:
   
   ```sql
   select
   "ORDER_ID",
   "ORDER_TIMESTAMP",
   "CUSTOMER_NAME_ID",
   from   "ORDERS"
   ```
   
   to the following code:
   
   ```sql
   select
   "ORDER_ID",
   "CUSTOMERS"."CUSTOMER_NAME",
   "ORDER_TIMESTAMP"
   from   "ORDERS", "CUSTOMERS"
   where "ORDERS"."CUSTOMER_NAME_ID" = "CUSTOMERS"."CUSTOMER_NAME_ID"
   ```

   You could, as you did before, build this code with the Query Builder if you are not familiar with SQL syntax. You are only making three small changes to the code:

   - Deleting the reference to the CUSTOMER_NAME_ID and adding the reference to the CUSTOMER_NAME field qualified with the name of the CUSTOMERS table.
     
     Since you are adding the column to the second spot in the list, make sure to add a comma following it and remove the comma following ORDER_TIMESTAMP.

   - Adding the CUSTOMERS table to the list of tables following the keyword FROM.

   - Adding a WHERE clause to establish the link between the CUSTOMER_NAME_ID columns in both tables.

15. Click Apply Changes to save the changes.
You have integrated the new lookup relationship into the functionality of your application. These steps were necessary to create the new CUSTOMERS table and relate it to the existing ORDERS table. But you want to have access to more information about your customers than simply their name. You will expand the CUSTOMERS table and learn how to use SQL scripts to add data to the expanded table in the next section.

### Extending Existing Tables

There are three steps to go through to extend the data in a table. First, you have to expand the structure of the table. Next, you will have to get the customer data into the newly expanded table with an SQL script. The final step will be to quickly create some pages to maintain this data and add it to the list of maintenance pages you created in the last chapter.

#### Try It Out | Adding Columns to an Existing Table

You begin the process by adding more columns to the CUSTOMERS table.

1. Go to the SQL Workshop, and select Object Browser > Browse > Tables from the cascading menus.
2. Click on the CUSTOMERS table.

You can see that the CUSTOMERS table you created has just two columns. You have to add more columns to hold additional information about your customers.

3. Click on Add Column to bring up the page shown in Figure 5-1.

On this page, you have to add a name for your new column, and make sure that the Type is appropriate for the data you will be storing in the column. You can also indicate that the column may or may not contain data by leaving the Nullable value to NULL.

4. Set Add Column to CUST_STREET_ADDRESS, the Type to VARCHAR2 and the length to 200.

The Oracle database always stores VARCHAR2 columns as variable length columns, so the Length specification does not really affect storage requirements, only the maximum length for the data in the column.

5. Click Next to bring up the verification page, where you can click Finish to add the column to the table.
You will take the same approach to add more columns to the table. Table 5-1 lists the column names, types, and storage specifications for the additional columns for the CUSTOMERS table.

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Type</th>
<th>Storage Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>CUST_CITY</td>
<td>VARCHAR2</td>
<td>40</td>
</tr>
<tr>
<td>CUST_STATE</td>
<td>VARCHAR2</td>
<td>2</td>
</tr>
<tr>
<td>CUST_POSTAL_CODE</td>
<td>VARCHAR2</td>
<td>10</td>
</tr>
<tr>
<td>CUST_PHONE</td>
<td>VARCHAR2</td>
<td>15</td>
</tr>
<tr>
<td>CUST_CREDIT_LIMIT</td>
<td>Number</td>
<td>Precision: 7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Scale: 0</td>
</tr>
<tr>
<td>CUST_EMAIL</td>
<td>VARCHAR2</td>
<td>40</td>
</tr>
</tbody>
</table>
Normally, you could name the columns in a table anything you want. But you will be using an already written script to update the columns in this table, so the names of your column will have to be exactly the same as those listed in Table 5-1.

When you are done adding all the columns to the CUSTOMERS table, your table structure should look like Figure 5-2.

![Figure 5-2](image)

The next step is to add data to your new columns.

**Try It Out**  
**Using Scripts to Update Rows**

As you know by now, the data for your APEX applications, as well as the code for the application, is stored in an Oracle database. This architecture means that you can use all standard SQL functionality to interact with that data.

You have just added columns to the CUSTOMERS table, a table that already contains a primary key, CUSTOMER_NAME_ID, and the CUSTOMER_NAME data. It just so happens that data for these customers exists, and you would like to add it to the newly modified table.

You can accomplish this by running an SQL script. A script is nothing more than a text file that contains SQL commands. In this case, the command you will want to use is the UPDATE command.
An **UPDATE** statement is used to set the value of columns in one or more rows in the table. If you only want to update a subset of all the rows in the table, use a **WHERE** clause to limit the rows affected.

The script that you will use contains two flavors of **UPDATE** statements — a series of statements to update columns in specific rows, and one statement to set a value for all the rows.

SQL Workshop has a handy utility to help you store and run scripts.

1. Return to the SQL Workshop main page.
2. Click on the SQL Scripts icon to bring up the page shown in Figure 5-3.

![Figure 5-3](image)

You can see that you can either create a script from scratch or upload a script file.

3. Click Upload to bring up the page shown in Figure 5-4.
4. Use Browse to bring up a file browser. Navigate to the sample code for this book and select the `update_customers.txt` file.
5. Set the **Script Name** to **Update Customers**, and click Upload. This action will store the script in the APEX environment, so you can easily retrieve it and run it.

When you return to the main SQL Scripts page, you can see your new script.
6. Click on the Update Customers script to bring up the page shown in Figure 5-5.
This page contains the script you just loaded. The script itself is:

```sql
UPDATE customers SET
    cust_street_address = '45020 Aviation Drive', cust_city = 'Sterling', cust_state = 'VA',
    cust_postal_code = '20166', cust_phone = '703-555-2143', cust_email = 'airport@dc.org'
WHERE customer_name = 'John Dulles';

UPDATE customers SET
    cust_street_address = '6000 North Terminal Parkway', cust_city = 'Atlanta', cust_state = 'GA',
    cust_postal_code = '30320', cust_phone = '404-555-3285', cust_email = 'airport@atl.org'
WHERE customer_name = 'William Hartsfield';

UPDATE customers SET
    cust_street_address = 'Harborside Drive', cust_city = 'Boston', cust_state = 'MA',
    cust_postal_code = '02128', cust_phone = '617-555-3295', cust_email = 'airport@bos.org'
WHERE customer_name = 'Edward Logan';

UPDATE customers SET
    cust_street_address = '10000 West OHare', cust_city = 'Chicago', cust_state = 'IL',
    cust_postal_code = '60666', cust_phone = '773-555-7693', cust_email = 'airport@ord.org'
WHERE customer_name = 'Edward Butch OHare';

UPDATE customers SET
    cust_street_address = '10701 Lambert International Blvd.', cust_city = 'St. Louis',
    cust_state = 'MO',
    cust_postal_code = '63145', cust_phone = '212-555-3923', cust_email = 'airport@stl.org'
WHERE customer_name = 'Fiorello LaGuardia';

UPDATE customers SET
    cust_street_address = 'Hangar Center - Third Floor', cust_city = 'Windsor Locks',
    cust_state = 'CT',
    cust_postal_code = '11371', cust_phone = '212-555-3923', cust_email = 'airport@lga.org'
WHERE customer_name = 'Albert Lambert';

UPDATE customers SET
    cust_street_address = 'Schoephoester Road', cust_city = 'Flushing', cust_state = 'NY',
    cust_postal_code = '06096', cust_phone = '860-555-1835', cust_email = 'airport@hfd.org'
WHERE customer_name = 'Eugene Bradley';

UPDATE customers SET cust_credit_limit = 1000;
```

The script is pretty direct. Each row in the customer table has a corresponding `UPDATE` statement that will set the values of most of the columns in the row. The final statement in the script updates the value of the `CUST_CREDIT_LIMIT` for all the rows in the table, which is fine since all customers will start out with the same credit limit.

7. Click Run, which will bring up a confirmation page. Click Run again to bring up the page shown in Figure 5-6.

This page displays information about all the scripts that have been run in this session. If you are working with scripts a lot, you will find this page very handy, since it contains links for each script, which can take you right back to the script page itself. You can also see a `Status` field, which initially displays Submitted.
8. Refresh the page in your browser.

Unless the server is exceptionally busy, you should see that the script status has changed to Completed. Once the script is complete, you can see how many of the statements in the script have been executed.

9. To get a little more detail on the results of the script, click View Results for the script, to bring up the page shown in Figure 5-7.

If any of the statements in the script resulted in errors, this page would let you see those results. This script, though, should have run perfectly without any errors.

10. Return to the SQL Workshop, and use the cascading menus to browse the tables.

11. Select the CUSTOMERS table, and click on the data link to bring up the page shown in Figure 5-8.

You should see your CUSTOMERS table populated with data.

If you are used to using SQL, you are probably very happy to see this scripting facility. If SQL is still unfamiliar to you, you may feel yourself shying away from using this capability. But those of you in this camp should still be aware that SQL is a very, very powerful language, and you should think about using this power to help you accomplish your data-centric tasks. And you can use the Query Browser to build queries with a point-and-click interface and then examine the generated SQL as a way of becoming more familiar with its syntax.
Chapter 5: Reporting and Charting

Figure 5-7

Figure 5-8

Excerpted with permission from *Beginning Oracle Application Express*, ISBN 9780470388372.
Part II: Building an Application with Application Express

The last thing to do before moving on to reports is to quickly create and integrate pages for maintaining your customer data.

Try It Out Adding Additional Maintenance Pages to Your Application

Now that you have your CUSTOMERS table, you will have to give your users a way to maintain the data in the table.

1. Return to the Application Builder, and click on your application.

2. Click Create Page, and then select the Form option and click Next.

3. Select the Form on a Table with Report option, and click Next.

4. Accept the default schema by clicking Next.

5. In the next page, select the CUSTOMERS table from the picklist, and click Next.

6. On the next page, you can change the Page Name and Region Title, if you want to make them more descriptive and click Next.

7. On the next page, leave the default of Do not use tabs, since you will be adding this page to your navigation list, and click Next.

8. On the next page, select all the columns except the CUSTOMER_NAME_ID and click Next. Accept the default edit icon on the next page, and click Next again.

9. You can change the Page Name and Region Title on the next page, if you want to give more descriptive names to the form, and then click Next.

10. Accept the primary key on the next page.

11. Change the source for the primary key to Existing sequence, and select the CUSTOMERS_SEQ in the picklist below. Click Next.

12. Select all the columns on the next page, and click Next. Accept all the default buttons on the next page, and click Next again. Click Finish to create the form and report pages.

You have created the appropriate pages to handle basic maintenance on the CUSTOMERS table. Now all you have to do is to add a link to the report to your navigation list to integrate these pages into your application.

13. Click on the Shared Components icon (the gear) in the upper right of the page.

14. Click on the Lists link in the Navigation section of the page to access the list you created.

15. Click on the list you created to bring up the table of list entries. Click Create List Entry.
16. Set the List Entry Label to Customers, and select the report page you just created for the Page in the Target section. Click Create.

Before seeing how the new list entry looks in your application, you want to make sure that the right tab is selected when your two new pages are in use.

17. Click on the Shared Components icon, and then the Tabs link in the Navigation section. 

18. Click on the Maintenance tab to select it, and then the Edit icon to edit the values. 

19. In the Tab Also Current for Pages text box in the Current For Pages section, add the page numbers for the two pages you just created. If you are not sure of the page numbers, you can use the picklist for the Tab Pages field above to find them. 

20. Click on Apply Changes to save the changes, and then click on Run in the upper-right corner, which will launch the application. 

21. Click on the Orders tab, and then click on Edit for one of the orders. The Edit form for the ORDERS table now looks like Figure 5-9.

![Figure 5-9](image-url)
22. You can see the new link for the maintenance page for **CUSTOMERS**. Click on the link to go there and see how the proper tab is selected.

You have now increased the scope of the data your application is built on. The rest of this chapter will be about creating and using reports and charts on that data.

---

**Report Creation**

Your data schema is set and populated. You have pages in place to look at small amounts of data, such as a list of customers or the order items in a particular order. But the value of data expands as you create reports on larger amounts of information. Reports give your users the ability to move from a view of a single entity, such as an order, to a larger universe, such as overall sales.

You unlock this value through the use of reports and charts, and APEX gives you some very nice options to easily deliver that increased value.

---

**Try It Out Creating a Standard Report**

You have already created reports as part of the page creation process. It’s time for you to design some reports that provide a wider view of data by accessing more than two tables.

The first report you will create is a standard APEX report.

1. Return to the APEX development environment, and click Create Page.
2. Select the Report choice, and click Next.

You can use a Report Creation Wizard to create a report quickly.

3. Select the Wizard Report, and click Next, which will bring up the page shown in Figure 5-10.
4. Change the **Page Title** to **Orders Report** and the **Region Title** to **Orders**. Click Next.
5. On the next page, accept the default, Do not use tabs, and click Next.

The next page, shown in its complete state in Figure 5-11, is the core page in the wizard. In this page, you can select one or more tables to use in your report.

6. Select the **ORDERS** table in the Table/View selection list. This action will cause the columns for that table to appear in the left-hand Available Columns list box.
7. Select **ORDER_TIMESTAMP**, and click the > to move it to the Displayed Columns list box.
Figure 5-10

Figure 5-11
This action has two effects. The select column(s) move to the right-hand box, but the column name is preceded by the table name. Also, when you select a column from the first table, a pair of radio buttons appears just below the Table/View picklist, relating to whether that picklist should now Show Only Related Tables to the ones that have been included in the report.

8. Leave Show Only Related Tables set to the default of Yes, and click on the dropdown button for the Table/View picklist.

This time, you see that the picklist only contains CUSTOMERS, the target of a foreign key relationship from ORDERS, and ORDER_ITEMS, which has a foreign key pointing to ORDERS.

9. Select the ORDER_ITEMS table, and then select the ORDER_ID, UNIT_PRICE, and QUANTITY columns and move them to the Displayed Columns list box.

Now that you have included ORDER_ITEMS in the report, the PRODUCTS table will appear as a related table.

10. Select the PRODUCTS table, and then the PRODUCT_NAME and LIST_PRICE to include in the report.

11. Select the CATEGORIES table and the CATEGORY column for inclusion in the report and to show the page as displayed in Figure 5-11. Click Next.

Previously, when you used the Query Builder, the generated SQL produced when you linked tables included a WHERE clause that described the logical basis of the link, which is referred to as a join condition. On this page, the Report Wizard has automatically created the most likely join conditions based on the defined foreign key relationships.

12. The join relationships shown in this page are appropriate, so click Next.

The next page, shown in Figure 5-12, gives you the option of changing some of the basic user interface options for the report, including the template, which provides a look and feel, how many rows per page, and whether you want to allow the user to sort the report based on a particular column.

13. You will see the report soon enough, so click Next to accept the defaults.

14. The final page contains a recap of your choices. Click Create Report Page to put the choices into your page.

Your report is complete — time to see it in action.

**How It Works**

Creating a report with the Report Wizard was pretty quick and easy. The wizard helped you, or any developer, to understand the relationships between tables, select columns, and only asked you for some simple user defaults.

1. Click the Run Page icon to see the report itself, as shown in Figure 5-13.
Chapter 5: Reporting and Charting

Figure 5-12

Figure 5-13
The report includes all the columns you specified. The columns are ordered in a slightly different order than you initially specified, but you can go to the report attributes for the page to change that order.

2. Click on the **Product Name** heading to change the report to the one shown in Figure 5-14.

Clicking on any column heading causes the rows in the report to be resorted, based on the selected column, in ascending order. The source and type of sorting is indicated by a little arrow in the column heading.

3. Click on the **Product Name** heading again to sort the rows in descending order.

Giving the end user the ability to sort the data in the report is a nice feature. The more you can empower the end user to shape their reports, the easier your development task will be, and the happier the end user will be. According to this principle, interactive reports is a major step forward for empowering end users, and reducing development effort, for APEX.

---

**Figure 5-14**

---
Try It Out  

Creating an Interactive Report

Creating an interactive report is slightly different from creating a report with a wizard, but the process is not really any harder. In this section, you will create an interactive report that includes all the tables used for the previous report, along with the CUSTOMERS table for some additional analysis possibilities.

1. Return to the application development page, click Create Page, and then click on the Report choice.

2. On the next page, select Interactive Report, and click Next.

3. You should change the Page Name to Customer Orders Report and the Region Name to Customer Orders. Select the Breadcrumb choice for Breadcrumb, and leave the default. Click Next.

4. On the next page, choose to use an existing tab set and create a new tab. Give the new tab the name of Reports, and click Next.

The next page gives you the option of entering an SQL statement directly or using the Query Builder to create your SQL.

5. Click Query Builder to bring up the page shown in its completed form in Figure 5-15.

![Query Builder Image](image-url)

Figure 5-15
You used the Query Builder in an earlier exercise, so you can quickly design the data source for this interactive report.

6. In the left-hand panel of the Query Builder page for interactive reports, click on the CUSTOMERS, ORDERS, ORDER_ITEMS, and PRODUCTS tables, in that order.

7. Create connections between the related tables by clicking on the fields in the defined foreign key relationships for each table. In other words, click on:
   a. The CUSTOMER_NAME_ID in the CUSTOMERS table and the CUSTOMER_NAME_ID in the ORDERS table to create a link.
   b. The ORDER_ID in the ORDERS table and the ORDER_ID in the ORDER_ITEMS table.
   c. The PRODUCT_ID in the ORDER_ITEMS table and the PRODUCT_ID in the PRODUCTS table.

8. Once you have these links created, click on the Table Actions icon in the upper-left corner of the PRODUCTS table heading. Use the Add Parent selection list to add CATEGORIES to the Query Builder window, which will automatically also include the link for the table.

   You can also use this shortcut to list and include child tables or to select all columns in a table with a single click.

9. Select the following columns for inclusion in the report by clicking on the check boxes to the left of each column:
   a. For the CUSTOMERS table, the CUSTOMER_NAME column.
   b. For the ORDERS table, the ORDER_ID and the ORDER_TIMESTAMP columns.
   c. For the ORDER_ITEMS table, the ORDER_ITEMS_ID, UNIT_PRICE, and QUANTITY columns.
   d. For the PRODUCTS table, the PRODUCT_NAME and LIST_PRICE columns.
   e. For the CATEGORIES table, the CATEGORY column.

When you check each column, the column appears in the column list at the bottom of the page. You could include selection conditions for any of these columns, but for this particular report, you can leave the selection conditions to the user.

The bottom panel of the Query Builder also lets you take a quick look at the results of the query, or move the order of the columns. You will see the results of the query soon enough, and you will leave the column order to the users.

10. Click on Return to copy the SQL from the query you just built to the text box in the Interactive Report Wizard, as shown in Figure 5-16, and then click Next.

   There is one powerful option enabled by default just below the SQL text box. You can leave the default of Yes for Link To Single Row View, which will automatically include a link to display the entire row for entries in the report.

11. Click Finish on the final page of the wizard to create the page.

Interactive reports put loads of flexibility in the direct control of your users. You will have to understand what they will be able to do for themselves in order to understand how little you will have to do to provide all of this for them. The next section is all about the capabilities of APEX interactive reports.
Chapter 5: Reporting and Charting

The interactive report you have just created with a few pages of specifying attributes can change in many ways. This section will go through many of the ways your users can shape their interactive reports.

1. Click the Run Page icon to run the interactive report you just created. The report will initially look like Figure 5-17.

You have already seen, and played with, an interactive report in an earlier chapter. You can now poke around and explore much more of the functionality inherent in these reports. In the following steps, you will look at most of the choices available to you on the Action menu for the interactive report. Your first task is to revisit the choices available through the column headings and to change the formatting of your currency fields.

2. Make sure that the Edit Links are showing for the report page, and click on the column heading List Price to bring up the menu shown in Figure 5-18.
### Figure 5-17

The screenshot shows a section from a database or application interface titled "Customer Orders Report". The interface displays orders with columns for Customer Name, Order ID, Order Time stamp, Order Item Id, Unit Price, Quantity, Product Name, List Price, and Category. The data includes various entries with details such as John Dukes ordering a 3.2 GHz Desktop PC for $1200, and so on.

### Figure 5-18

Another screenshot is from the same application, showing a similar "Customer Orders Report" interface. This view includes more data entries and filters or sorting options at the top, indicating a dynamic and interactive interface for viewing orders.

---

Excerpted with permission from *Beginning Oracle Application Express*, ISBN 9780470388372.
3. Click on the yellow arrow on the far right of the icons to edit the properties of the report.

4. Select the LIST_PRICE edit icon in the attribute page and change the Number/Date Format to represent currency formatting.

5. Click Apply Changes, and take the same actions for the UNIT_PRICE column. Once you have finished, click Run to return to the runtime environment.

You probably don’t need to see the Order_Items_Id column in this report.

6. Click on the gear in the top bar of the report to bring up the Actions menu.

7. Click Select Columns in the Actions menu to bring up the page shown in Figure 5-19, and move the Order_Items_Id column into the left Do Not Display list box by selecting the column and clicking on the corresponding arrow between the list boxes. Click Apply.

8. Select the Actions menu, and click Filter to bring up the page shown in Figure 5-20. In Chapter 2, you added filtering conditions to your interactive report through the Search box at the top of the report. Your users can still add conditions with this method, but they can also add conditions in this dialog, with a little more flexibility.

9. Select the Product_Name column, the contains operator, and the value 3 as the expression, and click Apply to limit the rows displayed.
10. Return to Filter for interactive reports, and select the Order_Timestamp column. Click on the dropdown list for Operator, and you can see that between is listed for this date field.

The between operator makes sense for date fields and is more flexible than the choices offered in the column heading, which you used in Chapter 2. You can also see that both of the Expression fields for this operator include a date picker.

11. Click Cancel to return to the report without imposing a date condition.

Because of the test data, it may be difficult to assign a date range for your report. But once you get to the end of this section and see all the things you can do with your interactive reports, you will be able to see how the ability to specify a date range will be very useful.

12. Click Sort in the Actions menu to bring up the section shown in Figure 5-21.

13. Select the Order_Id for the first sort column, and then the Order_Item_Id for the second sort column. Click Apply to add the sort conditions to your report.

You saw that the Order_Item_Id is in a separate part of the column list, under the heading of Other, because the column is no longer shown on the page. You can sort on any column that is a part of the results for the report, whether it is visible or not. You can also see that the sort conditions for the report are not displayed at the top of the report, so you will have to go to the menu to see or change the sort order.

It’s time to start exploring some of the more advanced features in your interactive report.
Chapter 5: Reporting and Charting

14. Click Control Break in the Actions menu to bring up the page shown in Figure 5-22.
Specifying a control break has a few effects. The data in the report will be sorted on any control breaks specified. The rows for each value in any of your control breaks will be grouped together, with a header and footer. Finally, the column value and its column heading will be moved to the heading of the control break group.

15. Select the Customer_Name field for the first control break and click Apply to display the report as shown in Figure 5-23.

16. Add another control break for the Order_Id field, and click Apply to change the appearance of your interactive report, as shown in Figure 5-24.

![Figure 5-23](image.png)

Notice that the interactive report does not add separate headings for multiple control breaks if they occur with the same row. Although both the Customer_Name and Order_Id changed with the last two customers, the interactive report only has a single heading for both of the changes.

As you saw when you added selection conditions earlier, each time you add conditions, or control breaks, to your report, an entry for the attribute is shown at the top of the report. Users can easily turn each of these attributes on or off with the check box in front of the listing, or remove the attribute entirely.

17. Uncheck the Customer_Name control break to show that column moving back into the report lines. Reselect the Customer_Name control break.

On to the next item to explore in the Actions menu.

18. Click Compute in the Actions menu to bring up the page shown in Figure 5-25.
Chapter 5: Reporting and Charting

Figure 5-24

Figure 5-25
End users can increase the value of an interactive report by adding computations that derive additional information from the existing data. The Compute section gives you helper tools to build computations. The columns on the left are represented by letters that are added to the computation, along with operators and numbers selected from the keypad.

The examples at the bottom of the section point out that you can create “computations” that are not necessarily numbers. For instance, you could use the concatenation operator (||) to join two strings into a single column.

You can see that you also have access to many functions that are implemented as part of the Oracle database. You can review the syntax and operation of these functions in the Oracle documentation.

19. Give the computation the Column Heading of Total Price and a Format Mask for a currency field.

20. Click on the Unit_Price field in the Columns list, the multiplication operator (*) in the keypad, and then on the Quantity column.

The formula added to the Computation text box will look something like $E \times F$, although the actual letters may be different if the columns in your interactive report are in a different order.

21. Click Apply to bring up the report as shown in Figure 5-26.
Now things are getting interesting. Your users can see very relevant information derived from the core data in your database. To make the examples in the rest of this chapter more interesting, you can add another computation.

For the purposes of the examples in this book, the Unit_Price represents the actual price the customer paid for some units, while the List_Price is the price listed in the price book. The sample data you loaded contains instances where the actual price is both greater than and less than the list price — a scenario that could occur when a product is in high demand and low supply to create a premium on the price, or overstocked items where the actual price is lower than the list price, respectively.

22. Click Compute in the Actions menu. Give the new computation a Column Heading of Savings and a currency format. Create the computation with the Columns list and the keypad to represent the following description:

\[(\text{List\_Price} \times \text{Quantity}) - (\text{Unit\_Price} \times \text{Quantity})\].

In your actual computation, the columns will be replaced by letters representing those columns.

23. Click Apply to bring up the new page, as shown in Figure 5-27.

The next logical step would be to find a way to combine the price and savings for the entire order, which is easily accomplished with another Actions menu choice.
24. Click Aggregate in the Actions menu to bring up the page shown in Figure 5-28.

**Figure 5-28**

Defining an aggregate value is quite easy. You just select the aggregate function and the column on which to apply the function.

25. Select the Sum function and the Total Price column, and click Apply.

26. Add another aggregate value to the report by selecting Aggregate in the Actions menu, the Sum function, and the Savings column. Click Apply to display that page shown in Figure 5-29.

You can see that, as with sort specifications, you cannot change the aggregate columns shown in your interactive report through a list choice at the top of the report. You can take an aggregate out of your report from the Actions menu by selecting the name of the Aggregate and then clicking on the now visible Delete button.

One of the great things about using aggregates and control breaks in the same report is how they work together so well.

27. Click on the Order_Id control break check box to change the display of the report to the page shown in Figure 5-30.
Aggregate values are only applied at the end of control breaks, so by simply changing the control breaks included in the report, your user can display different calculations. This flexibility makes your interactive reports even more popular with the user crowd.

But wait, there’s more — the most visual aspect of interactive reports.

As mentioned earlier, the sample data for this book includes ORDER_ITEMS where the price paid by the customer is both greater than and less than the list price of the item. You can use the Highlight feature to bring each of these situations into focus.

28. Click Highlight in the Actions menu to bring up the page shown in completed form in Figure 5-31.

![Figure 5-31](image)

29. Give the highlight a Name of Savings and a Highlight Type of Cell.

You can either highlight a cell or a row, based on the conditions you will enter next.

30. Select the Savings column, the greater than operation (>) and enter 0 as the Expression.

31. Select the green choice to the right of Background Color.
You could also use the color palette to the right of the Background Color text field. Once you have selected a color, the code for the color shows up in the text field and the color itself is displayed in the second box to the right of the field.

32. Click Apply to see your newly colorful report, as shown in Figure 5-32.

![Figure 5-32](image)

It's only fair to highlight the bad news for the user as well as the good.

33. Select Highlight from the Actions menu, and create a new highlight named Overcharge, which will display a cell in red if the Savings column is less than 0. Click Apply to produce the even more colorful report shown in Figure 5-33.

The highlights, like the selection conditions and the control breaks, are shown at the top of the page, and can be turned on and off simply by changing the check box.

You are now, hopefully, pretty excited with the flexibility of interactive reports. Interactive reports have so much flexibility that a single report could be used for many different presentations. The next section will cover some of the different ways that users can freeze the output of an interactive report.
Some readers may be starting to feel a bit uneasy about now. Interactive reports seem to offer almost too much power to end users. Can this awesome power be limited somehow?

The answer is yes, in several ways. First of all, remember that all data access through Application Express is built on the Oracle database, and that you, or your database administrator, can impose security on that data. If a user does not have access to the data directly, he or she will not have access through APEX.

At the same time, remember that reports only read data. The one thing you cannot allow is for users to mess up the integrity of the data, either unintentionally or on purpose. You spent a fair amount of time in earlier chapters preventing this type of potential problem. But since a report only presents information for reading, no harm can come from any amount of flexibility in interactive reports.

But you can also limit the areas of flexibility offered by any single interactive report. The properties for an interactive report include the ability to dictate what choices are available in the Actions menu in the Search Bar section of the attributes, as shown in Figure 5-34. If you don’t want your users to have access to the search box, any particular Actions menu choice, or the entire menu, you can just disable the display of those features. The options available for the column drop-down lists can also be controlled for each column.

You could also create a middle ground by creating and saving an interactive report with a number of selection, control break, and highlight features, but without access to the Actions menu, so that the users could select any of the choices you gave them but not add their own.
Freezing Interactive Report Output

You are probably starting to realize that creating an interactive report, on your part, is just the first step towards the use of that report. All you have to do is to specify the way that different tables are joined and the complete set of columns a user may want to include in the report. Your users can take the report the rest of the way to produce the output they really want.

But what if they want to save one, or more, of the possible versions of an interactive report? This section will cover how users can save multiple versions of reports, the data that those reports capture, and even the data at different points in time!

Try It Out  Saving a Report

You can probably already imagine the following scenario — a user painstakingly modifies an interactive report to just the way he likes it. He goes to lunch, and on the way back to his desk, he thinks of an even better option for the report. Can he get there without losing the beautiful configuration he has now?

1. If you are not in the interactive report, open it, and select Save Report on the Actions menu to bring up the page shown in Figure 5-35.
You can see that you have two basic choices as a developer. You could save the current configuration of the interactive report as the default view for all users. This option, only available to developers, allows you to create a version of the report to serve the needs of most users and act as a starting point for further modifications. The other choice, to save the configuration as a named report, is primarily for end users to save multiple personalized versions of the information in the report.

2. Give the report a name, such as Good Report, and a description, and click Apply to return to the page shown in Figure 5-36.

You can see that you now have a tabbed interface to the interactive report, with tabs for the report you just saved and what is called Working Report, a title that refers to the unsaved version of the report. If you move your mouse over the tab of this or any saved report, the description you entered will appear in a tip box.

You could make further changes to the working report and save the new versions as another saved report. Once you save a report, the configuration options become the defaults for that report.

3. Go to the saved report and change the report by unchecking the control break.

4. On the Actions menu, select Reset.

You will be prompted as to whether you really want to reset the report options to the default configuration.

5. Click Apply to bring the saved report back to its original state.
Chapter 5: Reporting and Charting

Excerpted with permission from *Beginning Oracle Application Express*, ISBN 9780470388372.

Figure 5-36

The saved reports for an interactive report are associated with the individual user, so when that user returns to the report, the saved reports will still be there. Later in this book you will learn about creating users. You can have APEX applications that do not differentiate between users — they do not have any method to log in and authenticate users. In these applications, the ability to save versions of interactive reports is automatically disabled, as you would expect.

Saving report versions is a great feature. But what about if you wanted to share the data used in a report with applications and users outside the scope of your APEX application?

---

Try It Out  Saving Data

Whenever you have a report, you can give your users the option to save the data shown in the report. There are a variety of reasons why your users would want to do this. They might want to include it in a Word document or email message. They might want to import the data into another database or application, or they might just want to create a read-only version of the report in a portable format.

Your users can do any of these things with another feature of interactive reports.

1. On the Actions menu, select Download to bring up the section shown in Figure 5-37.
Figure 5-37

You can see that you have one option to save the data, but weren’t you promised more than one? There’s a simple explanation.

2. Click Cancel, and then click Edit Page and the Interactive Report entry in the Regions section to take you to the attributes page for the report. Click Download to bring up the section shown in Figure 5-38.

There they are — the additional download formats.

3. Select all of the formats and click Apply Changes, then run the page and select the Download option from the Actions menu. The new set of download options looks like Figure 5-39.

4. Select the PDF format and, when prompted, open the report in the Adobe PDF Reader.

Well, it’s not the most beautiful report in the world, but the results are now in PDF format, which can be used across many platforms. You could modify the appearance of the PDF report by using the Print Attributes page for the report, where you can add information and graphics to the header and footer of the report or change the display fonts for column headings and columns. Of course, you could also modify the column headings by changing the Column Heading attribute for the column.

You should be aware that you can create very robust report templates for Word documents with Oracle BI Publisher, and that you can use these templates with the report data from your interactive reports.

You have saved reports and data. Can you also save time, in the form of running a report on data from the past?
Chapter 5: Reporting and Charting

Figure 5-38

Figure 5-39
Try It Out  Save the Past

You may have noticed a choice in the Actions menu named Flashback. This choice allows you to query data as it existed in the recent past.

The Oracle database has included, since the late 1980s, the ability to maintain multiple versions of a data row. If someone changes data in a row, Oracle maintains the previous image of the row and automatically provides this version when a query against this row has started before the row has been changed. This feature was designed to eliminate the need to protect data being read with locks to preserve data integrity.

If you feel a bit lost with this brief explanation, don’t worry — you don’t have to understand it to take advantage of flashback. But if you want to learn more about it, check out the Concepts guide in the Oracle documentation, or Oracle Essentials by this author.

Many years later, the development team at Oracle realized they could use these same multiple versions to support a view of data from the past, and flashback was born. The easiest way to appreciate flashback is to see it in action.

1. Return to your interactive report. Add a new Control Break on Category and disable the Control Break for Customer_Name to reduce the complexity of the report and make the individual rows easier to see.

2. Take note of the value for the Order_Id and the Quantity for the first row for the Audio group, which should refer to an MP3 player.

3. Click on the Orders tab and select the order with the Order ID you noted previously.

4. Find the Order with the Order_Id previously noted for that first row. Edit the order line for the MP3 player, changing the Quantity, and click Apply Changes.

5. Click on the Reports tab. You should see a changed value in the first row, since the report, by default, operates on the most recent version of the data.

6. Click Flashback in the Actions menu, which will bring up a section as shown in Figure 5-40.

7. Enter 10 into the text box and click Apply. The new report should show the old value.

8. Just to prove there is no trickery here, deselect the flashback specification. The report should refresh with the most current values.

Magic, right?

The documentation for the flashback capability of APEX states that you can go back up to 3 hours in time. In fact, the amount of time you can flashback is dependent on the space reserved for multiple versions of the data, and this space is used up in different amounts of time, depending on the allotment and the amount of updates to the data. If you try to flashback past the time when the Oracle database can give you a consistent view of your data, you will simply receive an error from the Oracle database.

The best part about the flashback capability is that you get it for free — it’s a feature of the Oracle database that is simply exposed in your APEX interactive reports.

In the next section, you will learn about using charts from an APEX application.
Charts

Up until now, you have been looking at reports, which present detailed information along with some aggregate summary data. Sometimes your users will want a higher-level view of their data, which can provide some telling information across a greater number of rows. For this type of examination, charts are an excellent choice.

You have no doubt noticed that one of the choices in the Page Wizard is a chart and that there is a Chart choice in the Actions menu of an interactive report. When you create a chart on a page, you have a very broad range of customization options, which you will explore in the next chapter. But for the purposes of this quick fly-by of the main APEX functionality, you can just as easily create charts based on your interactive report.

1. Select Chart from the Actions menu, which will bring up the section shown in Figure 5-41.
2. Select the pie chart option, the third from the left.

Once you select the pie chart, the fields for the axis labels disappear. All you have to do is to designate a column to supply the value used for the chart, a function to aggregate the values, and a sort order, if relevant.

3. Select the Product_Name column as the Label, the Total Price column as the Value, and Sum as the Function.

4. Click Apply to bring up the chart, as shown in Figure 5-42.

Your chart doesn’t look too bad, especially considering how easy it was to make. If you hold the mouse over either section of the chart, the label and value will appear in a tip window.

The only real problem with the chart is that the information displayed is not really that interesting. The reason for this lack of interest has to do with the fact that the selection conditions were applied to the report to limit the number of detail rows shown. Since the purpose of a chart is to aggregate data, the chart would be a lot more interesting if it showed the total amount of sales for all the products.

With an interactive report, you can make this change with a single mouse click.

5. Click on the filter condition check box to remove the condition. The new version of the chart will look like Figure 5-43.
Chapter 5: Reporting and Charting

Figure 5-42

Figure 5-43
This new version of the chart presents aggregates based on more data, resulting in a more informative chart for your users.

This brief look at the charting capabilities of your interactive report has just scratched the surface of the charting possibilities in APEX, but you are probably already imagining some of the ways you can use charts in your own applications.

Summary

In this chapter, you delved into the world of APEX reports and charts. The chapter focused on the capabilities of the reports, including:

- Standard reports
- Using Query Builder to create reports based on multiple tables
- Interactive reports and modification of reports through the Actions menu
- Flashback reports
- Creation of standalone charts

As part of the preparation for the chapter, you also worked a bit more with SQL Workshop to both add columns to an existing table and to use the SQL script utility in the Workshop.

In the next chapter, you will learn how to combine multiple views of data to give your users a richer application experience by providing them with the option to drill down from the higher-level view of data provided by a chart, as well as the display of data within a calendar.