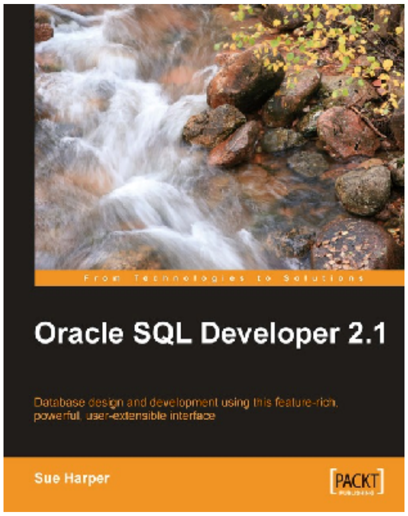
PACKT
PUBLISHING

Oracle SQL Developer 2.1

Sue Harper


From Technologies to Solutions
Oracle SQL Developer 2.1
Database design and development using this feature-rich, powerful, user-extensible interface
Sue Harper
PACKT

Chapter No. 1
"Getting Started with SQL Developer"

In this package, you will find:

A Biography of the author of the book

A preview chapter from the book, Chapter NO.1 "Getting Started with SQL Developer"

A synopsis of the book’s content

Information on where to buy this book

About the Author

Sue Harper is a product manager for Oracle SQL Developer, SQL Developer Data Modeler, and SQL Developer Migrations working in the Database Development Tools group at Oracle. She has a BSc in Mathematics and Physics from the University of the Witwatersrand, Johannesburg, and a post graduate teaching diploma. It was this teaching qualification that opened the doors to working with Oracle products in South Africa, where Sue started as an instructor in 1992, teaching SQL, PL/SQL, and SQL Forms and Reports. Before long, she'd added Oracle Designer to her repertoire and soon left for the UK to join the Oracle Designer curriculum development team. Sue was a course writer and then an Oracle Designer product manager, traveling extensively, meeting customers, presenting at conferences, and running train-the-trainer classes. Later, she became a product manager for Oracle JDeveloper working with the database and modeling tools and helping the Oracle Designer community learn about the new tools, before moving to the Database Development Tools group.

Based at home, Sue works with her team scattered around the world and with customers. She runs web-based training sessions and writes supporting material for the products. She frequently attends and presents at conferences and has a number of blogs, which she contributes to sporadically.

Sue lives in West London, where she enjoys walking her dog in the extensive local parks. As a walker, her favorite stomping ground is the Drakensberg in South Africa. With the 'berg' too many miles away, any local hills will do. As a keen photographer, she now enjoys the long distance trails in the UK and is often out in the countryside with her camera.

Oracle SQL Developer 2.1

Oracle SQL Developer is a relatively new product included in the range of Oracle products. It was first introduced to the world in September 2005, by its code name Project Raptor. Raptor is a name many still cling to, despite being replaced very quickly with the full product name, Oracle SQL Developer (and referred to in the rest of the book as SQL Developer). The first production version was released in early 2006 and had many Oracle customers both skeptical and excited.

SQL Developer is a Graphical User Interface (GUI) for browsing and working with databases. Primarily developed for the Oracle database, it also provides a GUI for a selection of popular non-Oracle databases. As technology has evolved, many developers are looking for modern tools to assist them in daily tasks. A clean, easy to navigate interface is essential and many tools are judged on looks, performance, and functionality.

Initially billed as a light-weight, clean, easy to install GUI for the Oracle database, SQL Developer is gaining momentum in the market and broadening its scope in its offering.

The skeptics mentioned are ever concerned that Oracle is not really interested in the product, backing up this concern with the fact that SQL Developer is free. Disappointing as it is that "free" is equated with "poor quality", the last couple of years have considerably stilled many of these dissenting voices and the market using SQL Developer continues to increase. Time will tell.

What This Book Covers

The book is designed to allow you to dip into any chapter that is your current area of focus. Having said that, if you have never used SQL Developer before, are new to Oracle and SQL, or have never used a GUI tool before, we recommend that you read the introduction, which gets you started slowly. You can then work through the first few chapters in order. Once you have covered the first three chapters, you should be on your way.

As the chapters progress, while they do not necessarily get more complex, they do rely to some extent on existing knowledge. The latter portion of the book is not considered to be more complex, but instead focuses on areas used by specific target audiences, such as Oracle Application Express or migrating from non-Oracle databases.

We have loosely divided the book into the following three sections.

I: Getting started

The first few chapters introduce SQL Developer. Use them to get to know the layout of the product and the essential features it provides. For those new to Oracle, or application developers who spend the day using other development tools, these first few chapters may be the only chapters needed.

In *Chapter 1: Getting Started with Oracle SQL Developer*, we provide general information about SQL Developer, where you can find it, how to install it, and how to get started. Using a simple example, you quickly walk through the product, touching a variety of areas to get a feel for the navigation, the layout, and some of the features available. We introduce the environment, how to navigate and manage the interface, the database, and schemas used in this book. The chapter includes tips for customizing the environment to suit your preferences.

In *Chapter 2: Browsing and Editing Database Objects and Data*, you will create, edit, and update database objects, such as tables, views, sequences, and instance data (the data in those tables), using data grids. You'll also learn how to access and use the utilities, wizards, and dialogs, which SQL Developer provides, to work with data and database objects. Lastly, you will use SQL commands and scripts, accessing the code formatter and other assistants that separate a GUI tool from a command–line environment.

In *Chapter 3: Working with the SQL Worksheet*, you see that the SQL Worksheet provides a scratch pad for all SQL, PL/SQL, and SQL*Plus statements and commands. In this chapter, you learn to work with the SQL Worksheet and the commands it supports, how to use code insight, templates, snippets, and formatting code. You learn how to manage the environment and to work with multiple worksheets. We introduce the Query Builder and how to write, test, and run SQL queries using the Query Builder.

In *Chapter 4: The Power of SQL Reports*, you look at the variety of SQL reports provided by SQL Developer, why they are useful in daily tasks, and how to run and use them. You'll learn how to create your own reports, such as master-detail or drill-down reports, and share your user-defined reports.

II: Delving a little deeper

These chapters are intended for all developers, perhaps more focused on those of you who spend your days working with database objects and data. While not strictly advanced features, there are areas here that can be more involved, and knowledge of how the tool works will complement your existing knowledge of developing and working with code.

In *Chapter 5: Working with PL/SQL*, we start by reviewing various PL/SQL structures. This chapter covers creating, editing, compiling, and debugging PL/SQL code. You'll see how the various search tools work, and use code templates and snippets. You'll learn how to refactor PL/SQL code, and search using the Oracle Database 11g PL/Scope feature.

Chapter 6: SQL and PL/SQL Tuning Tools. Although not designed to teach you how to optimize your code, this chapter shows you the different utilities that SQL Developer provides to help you see problem areas and work at fixing them. Utilities include using Explain Plan, reviewing trace files, and auto trace.

In *Chapter 7: Managing Files*, you'll see that SQL Developer provides an integrated file navigator that allows you to browse the file system and open files from within the tool. This chapter introduces the file navigator and the features it supports. Working in conjunction with the file navigator, SQL Developer integrates with open source version control systems. This chapter discusses the alternatives and shows you how to place your code under version control and work within a team sharing code, by comparing and merging code.

In *Chapter 8: Importing, Exporting, and Working with Data*, you'll learn how SQL Developer provides a number of utilities for moving or copying both the data structures and the instance data between schemas and databases. This chapter introduces you to the schema copy, diff, and export wizards and related export and import utilities.

III: Focus areas

The last few chapters are focused on specific audiences. We don't specifically cover complex features, but there are aspects of some of these features that require knowledge beyond the scope of this book.

In *Chapter 9: Database Connections and JDBC Drivers*, we'll look at the various types of database connections available, how to set them up, and when to use them. We'll look at various authentication methods, such as LDAP, OS authentication or strong authentication methods like Kerberos, and compare Basic, TNS, and JDBC connections.

This chapter also looks at setting up the required JDBC drivers and creating connections to non-Oracle databases. The chapter closes with a section on managing your connections.

Chapter 10: Introducing SQL Developer Data Modeler. Whether you just want a diagram of how your tables connect, or you want to build an entity relationship diagram, transform that to a relational model, and generate the DDL script to create the objects in your database, the SQL Developer Data Modeler provides the tools you need. This chapter introduces some of the modeling utilities available in SQL Developer Data Modeler. You'll also learn about the integration points with SQL Developer and the Data Modeler Viewer in SQL Developer.

In *Chapter 11: Extending SQL Developer*, you'll see that SQL Developer is an extensible tool. This means you, as an end user of the tool, can add in your own features. These may be small, providing just a single input or output field, and only require XML and SQL knowledge, while others are more extensive and require Java skills. This chapter reviews the various ways you can extend SQL Developer using SQL and XML.

In *Chapter 12: Working with Application Express*, you'll see how SQL Developer connects to and provides utilities for interacting with Application Express (Oracle APEX). In this chapter, features you'll learn to use include setting up the environment to work with Oracle APEX, importing, browsing, and deploying applications. You'll also see how to tune and refactor SQL and PL/SQL code.

In *Chapter 13: Working with SQL Developer Migrations*, you'll understand how SQL Developer helps you connect to and browse non-Oracle or third-party databases. This chapter looks at preparing the SQL Developer environment, and browsing the supported third-party databases. The focus of the chapter is on migrating from a third-party database to Oracle, from setting up the repository, through the capture and conversion phases, and ending with data migration. You'll also learn to translate pieces of code, such as translating T-SQL to PL/SQL.

Standards and assumptions

Unless otherwise stated, the environment used in the examples and referred to throughout the book is Oracle Database 11g Release 1, installed on Microsoft Windows XP. We use the shipped Oracle sample schemas, in addition to using the **SYSTEM** and **SYS** users.

Oracle offers a free database, Oracle Express Edition 10g (XE), which you can download and use for many of the examples. Some of the discussion is related to new Oracle Database 11g functionality that is not available on any Oracle Database 10g release, including XE. In the case where the feature is Oracle Database 11g specific, the text will indicate this. However, the text does not list anomalies for each database version.

This book was written using and is based on Oracle SQL Developer 2.1. Screenshots and menu items all refer to SQL Developer 2.1 and were accurate at the time of writing, completed just prior to the production release of the software. Minor differences are inevitable due to changes in the later stages of product development prior to production. You can also use the book, and much of the content, as a guide for working with earlier releases of SQL Developer. Where they are significant, we make note of the differences in features that have changed from earlier releases.

1
Getting Started with SQL Developer

This book is divided into chapters that focus on the different areas or functionality in SQL Developer. The progression through the chapters is from the more frequently used features to those less frequently used. This initial chapter is all about preparing your environment, installation, and getting started.

SQL Developer is easy to set up and use, so there is very little setup required to follow the examples in this book. The best way to learn is by practice, and for that you'll need a computer with access to an Oracle database and SQL Developer. This chapter, and indeed the rest of the book, assumes you have a computer with Microsoft Windows, Linux, or Mac OS X installed, and that you have access to an Oracle database. It focuses on the alternative installations available for SQL Developer, where to find the product, and how to install it. Once your environment is set up, you can follow a quick product walk-through to familiarize yourself with the landscape. You'll create a few connections, touch on the various areas available (such as the SQL Worksheet and Reports navigator), and learn about the control of the windows and general product layout.

Preparing your environment

Preparing your environment depends on a few factors, including the platform you are working on and whether you have an early edition of SQL Developer previously installed. First, you need to locate the software, download, and install it.

Finding and downloading the software

SQL Developer is available through a variety of sources as both a standalone download and as part of the Oracle Database and Oracle JDeveloper installations.

SQL Developer is a free product, and you can download it from the Oracle Technology Network, `http://www.oracle.com/technology/products/database/sql_developer`. Use this link to reach the download for the latest standalone production release. It also includes details of the release and is regularly updated with news of preview releases and new articles. While SQL Developer is free to download and use, you are required to read and agree to the license before you can proceed with the download. The product also falls under Oracle Support contracts, if you have a Support contract for the database, which means that you can log Oracle Support tickets.

Downloading and installing the Java Development Kit

SQL Developer requires the **Java** SE **Development Kit** (**JDK**); this includes the **Java Runtime Environment** (**JRE**) and other tools, which are used by SQL Developer utilities such as the PL/SL Debugger.

For Microsoft Windows, you can download and install SQL Developer with the JDK already installed. This means you'll download and unzip the product and will be ready to start, as there are no extra steps required. For the other operating systems, you'll need to download the JDK and direct SQL Developer to the path yourself. Indeed, as many other products require a JDK to be installed, you may already have one on your system. In this case, just direct the product to use an existing JDK installation. For Microsoft Windows, ensure you download SQL Developer without the JDK to make use of an existing JDK installation.

The SQL Developer download site offers a selection of download choices:

- Microsoft Windows (with or without the JDK)
- Linux (without the JDK)
- Mac OS X (without the JDK)

In each case, make your selection and download the required file.

The download sites for the JDK are as follows:

- For Microsoft Windows and Linux:

  `http://java.sun.com/javase/downloads/index.jsp`

- For Mac OS X:

  `http://developer.apple.com/java/download/`

> SQL Developer is shipped with the minimum JDK required. You can download and use the latest updates to the JDK. You should be aware that some updates to the JDK are not supported. This detail is posted on the SQL Developer Downloads page for each release. Starting from SQL Developer 2.1, JDK 1.6 is the minimum JDK supported.

Once you have installed the JDK, you can start SQL Developer.

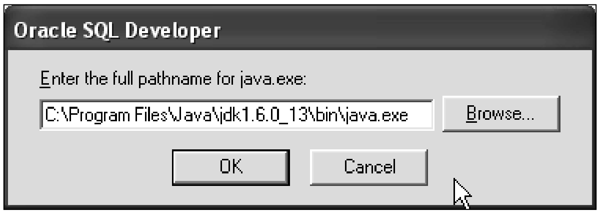
Installing and starting SQL Developer

SQL Developer does not use an installer. All you need to do is unzip the given file into an empty folder, locate, and run the executable.

> Do not unzip SQL Developer into an `$Oracle_Home` folder or an existing SQL Developer install.

Unzipping the file creates an `sqldeveloper` folder, which includes a selection of sub-folders and files, including the `sqldeveloper.exe` executable.

If your download does not include the JDK, then you'll be prompted for the full path of the `java.exe`. Browse to the location of the file and select it. The path should include the full path and executable (for example, `C:\Program Files\Java\jdk1.6.0_13\bin\java.exe`):



Oracle SQL Developer

Enter the full pathname for java.exe: C:\Program Files\Java\jdk1.6.0_13\bin\java.exe

Working with different platforms

Whether you are accessing SQL Developer as part of the Oracle Database 11g installation or as a stand-alone install, there is a selection of executables available to you. These are either platform specific or provide additional detail while running the product.

Microsoft Windows

The first executable you'll find is in the root folder `\sqldeveloper`. This is the executable more generally used. If you navigate down to `\sqldeveloper\bin`, there are two additional executables, `sqldeveloper.exe` and `sqldeveloperW.exe`. The latter is the same as the executable in the root folder. Use either of these for running SQL Developer.

The additional executable is often used for debugging purposes. Use `\sqldeveloper\bin\sqldeveloper.exe` to invoke SQL Developer and a separate console window which displays additional Java messages. You can use these messages when encountering errors in the product and if you want to log an issue with Oracle Support.

Oracle SQL Developer
Three steps to getting started on Microsoft Windows:

Download: Download the full file, with JDK, from the Oracle Technology Network web site.

Unzip: Unzip the file to an empty directory.

Double-click: Double-click on the `\sqldeveloper\sqldeveloper.exe` file.

Alternative platforms

Microsoft Windows is the predominant platform used by SQL Developer users. There is a steadily growing audience for Linux and Max OS X. As neither of these platform downloads include the JDK, you need to first access, download, and install the JDK. On starting either Linux or the Mac OS, you'll be prompted for the full path of the JDK as described.

Mac OS X

Download the file specific to Mac OS X and double-click to unzip the file. This creates an icon for SQL Developer on your desktop. Double-click to run the application.

Linux

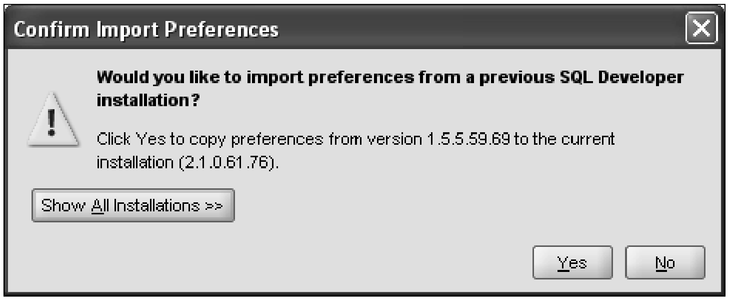
Use the Linux `rpm` command to install SQL Developer. For example, your command might look like this:

```
rpm -Uhv sqldeveloper-1.5.54.40-1.noarch.rpm
```

In the same way that unzip creates an `sqldeveloper` folder, with sub-folders and files, the `rpm` command creates an `sqldeveloper` folder, complete with files and sub-folders. Switch to this new folder and run the `sqldeveloper.sh` executable.

Migrating settings from a previous release

On the initial startup of any release of SQL Developer, you may be asked one or two questions. The first is the location of the Java executable of the JDK as discussed. If you have installed the full release with the JDK, this question is skipped. The second question is if you want to migrate any preferences from a previous release. Regardless of whether this is the first SQL Developer install on the machine or not, the first time you invoke SQL Developer, you are offered the choice of migrating your settings. You can migrate settings of any release from SQL Developer 1.5 and above. By default, the utility looks for the latest previous installation of the software.


Confirm Import Preferences

Would you like to import preferences from a previous SQL Developer installation?

Click Yes to copy preferences from version 1.5.5.59.69 to the current installation (2.1.0.61.76).

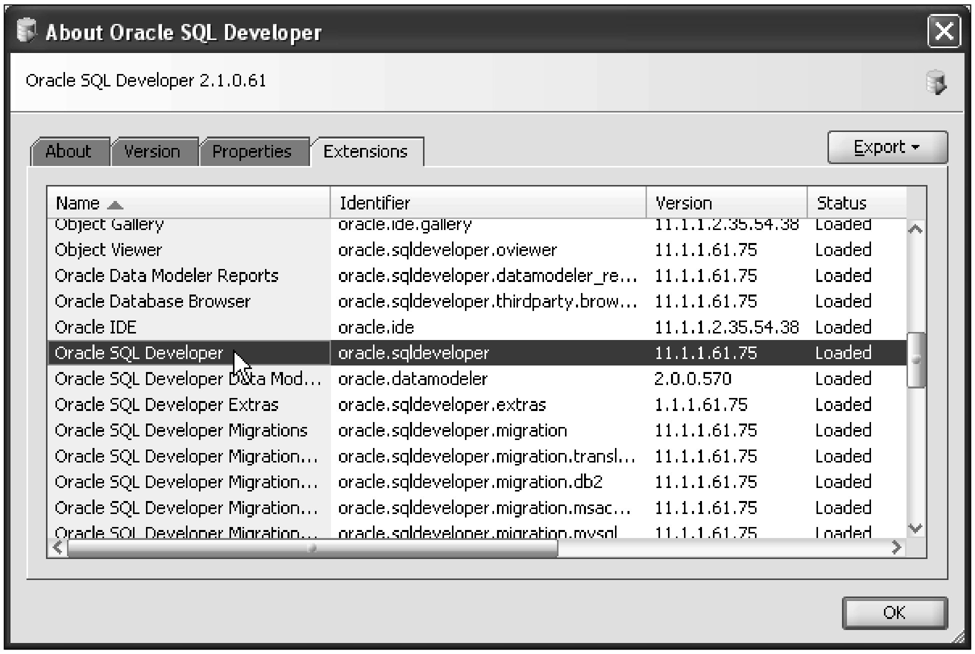
If you want to migrate from a different installation, select the **Show All Installations** button (seen above). This displays a list of all SQL Developer installations that have the system folder in the `Documents and Settings` system folder (for example, `C:\Documents and Settings\<your_user>\Application Data\SQL Developer\system1.5.1.54.40`) and includes releases from SQL Developer 1.5 and above. For releases prior to SQL Developer 1.5, the system folder was created within the SQL Developer install (for example, `D:\SQLDeveloper\Builds\1.2.1\1.2.1.3213\sqldeveloper\sqldeveloper\system`).

Maintaining your environment

Once you have SQL Developer installed, it is helpful to know about the environmental settings and some of the files that are created when you start the product. Knowing about the version you have installed is important if only to be able to identify this when asking questions on the forum, or when contacting Oracle Support.

Verifying the current release

To verify the SQL Developer release you have, select the **Help | About** menu once you start SQL Developer or JDeveloper. In the dialog invoked, select the **Extensions** tab and find the Oracle SQL Developer extension, as shown in the next screenshot. This will match the build number on the download site if you have the latest release. The screenshot shows a number of the extensions that make up SQL Developer. If your dialog does not show the **Version** or **Status** columns, you can select the column headers to resize the visible columns and bring the others into focus.



Name	Identifier	Version	Status
Object Gallery	oracle.ide.gallery	11.1.1.2.35.54.38	Loaded
Object Viewer	oracle.sqldeveloper.oviewer	11.1.1.61.75	Loaded
Oracle Data Modeler Reports	oracle.sqldeveloper.datamodeler_re…	11.1.1.61.75	Loaded
Oracle Database Browser	oracle.sqldeveloper.thirdparty.brow…	11.1.1.61.75	Loaded
Oracle IDE	oracle.ide	11.1.1.2.35.54.38	Loaded
Oracle SQL Developer	oracle.sqldeveloper	11.1.1.61.75	Loaded
Oracle SQL Developer Data Mod…	oracle.datamodeler	2.0.0.570	Loaded
Oracle SQL Developer Extras	oracle.sqldeveloper.extras	1.1.1.61.75	Loaded
Oracle SQL Developer Migrations	oracle.sqldeveloper.migration	11.1.1.61.75	Loaded
Oracle SQL Developer Migration…	oracle.sqldeveloper.migration.transl…	11.1.1.61.75	Loaded
Oracle SQL Developer Migration…	oracle.sqldeveloper.migration.db2	11.1.1.61.75	Loaded
Oracle SQL Developer Migration…	oracle.sqldeveloper.migration.msac…	11.1.1.61.75	Loaded
Oracle SQL Developer Migration…	oracle.sqldeveloper.migration.mysql	11.1.1.61.75	Loaded

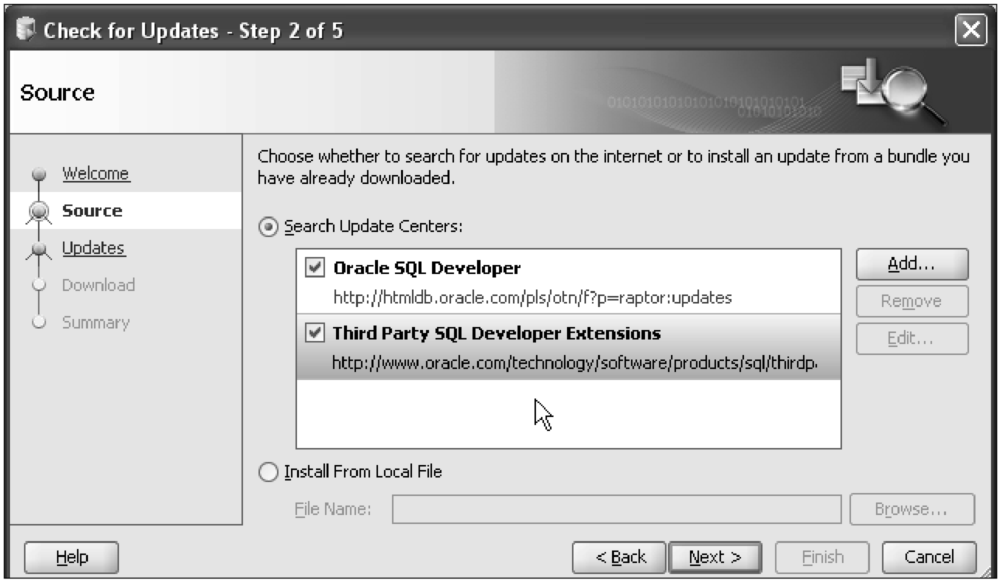
Using Check for Updates

SQL Developer offers a built-in patching and extensions utility, known as **Check for Updates**. Check for Updates is used to release:

- SQL Developer extensions
- General Oracle extensions
- Minor patches
- Third-party tools required by SQL Developer, such as the non-Oracle database drivers
- Third-party extensions

You can control whether Check for Updates warns you about new updates using the **Tools | Preferences** menu. Select **Extensions** and then select **Automatically Check for Updates**. For SQL Developer extensions, if you already have SQL Developer installed and you're not secured by a firewall, you'll be alerted about new updates. You need not use the utility to get the updates, but you'll be aware of the latest release from the alert. For all other extensions, you need to start **Check for Updates** to see what's available. To do this, select **Help | Check for Updates**. In either situation, just follow the dialog to find the updates you require.

You can initially elect to see just the third-party updates, or all updates available, by selecting all options, as shown in the following screenshot:



Choose whether to search for updates on the internet or to install an update from a bundle you have already downloaded.

● Search Update Centers:

☑ **Oracle SQL Developer**
http://htmldb.oracle.com/pls/otn/f?p=raptor:updates

☑ **Third Party SQL Developer Extensions**
http://www.oracle.com/technology/software/products/sql/thirdp

○ Install From Local File

File Name:

The database drivers for some of the non-Oracle databases are located in **Third Party SQL Developer Extensions**. The Third Party update center also includes a selection of customer developed SQL Developer extensions. The customer extensions are developed, supported, and updated by the customer involved, and are not tested, certified, or supported by Oracle.

As with all software downloads from the Internet, you are required to read and accept the license agreements. The **Check for Updates** utility directs you to the appropriate licenses, before downloading the software. If the updates are from Oracle, you will need to provide your Oracle Technology Network sign-on details.

Check for Updates is only used to apply patches to your install. Starting with SQL Developer 1.5.1, the team released patches which are applied to upgrade the product in place. These patches fixed one or two bugs and did not constitute a new download, or even a full install. For all other releases, you need to do a full install as described earlier.

Managing the system folder and other files

SQL Developer maintains a series of files as you work with the product. These files are created and stored in the `\Documents and Settings\<your_user>\Application Data\SQL Developer` folder. On Linux, these files are all stored in a folder `~/.sqldeveloper/`.

Deleting this folder is equivalent to returning a SQL Developer installation to its factory settings. The files at this highest level are:

- `CodeTemplate.xml`—created as you add your own user defined code templates
- `UserReports.xml`—created as you add your own user defined reports
- `SqlHistory.xml`—created as you execute SQL and PL/SQL commands in the SQL Worksheet
- `UserSnippets.xml`—created as you add your own snippets

These files are used by each of the SQL Developer installations you have. For example, you may elect to have the latest installation in addition to a number of earlier releases. Having several different releases of SQL Developer on one machine is acceptable, as the installations have no impact on each other, except that they do share these files.

Once you have started SQL Developer, a folder with sub-folders and files is created in the `\Documents and Settings\<your_user>\Application Data\SQL Developer` folder. The top-level folder is labeled `systemx.x.x.x.x` (for example `system1.5.0.54.40`). The `systemx.x.x.x.x` folder contains all of the other preferences and the settings that pertain to the specific release in use. In this case, deleting the system folder is almost equivalent to resetting SQL Developer to its factory settings, except that any user defined reports, SQL history, and code templates are not lost.

Sharing preferences

Preferences are set for your local environment and are therefore not shared globally between teams. However, you can export the SQL Formatter preferences set. This allows you to share the settings between team members and ensure that you all code to the same settings. To export your SQL Formatter settings, select **Tools | Preferences** and expand the **Database** node in the tree. Select **SQL Formatter**, you can now export or import previous saved settings.

Alternative installations of SQL Developer

We have been discussing the installation and management of the independent release of SQL Developer available on the Oracle Technology Network.
SQL Developer is also available as part of the Oracle Database and Oracle JDeveloper installations.

Oracle JDeveloper

Most of SQL Developer is integrated into Oracle JDeveloper, which means you need to install JDeveloper to access and use the SQL Developer components. Having SQL Developer as part of JDeveloper means that, if you are building Java applications and working with the Fusion Middleware platform, you can access and work with the Oracle Database without an additional install of SQL Developer. JDeveloper does not consume all of the extensions for SQL Developer (for example, extensions like Migrations and Versioning are not included).

> Oracle JDeveloper 11g includes SQL Developer 1.5.6.

Oracle Database 11g

SQL Developer is also shipped with the Oracle Database. Initially, Oracle Database 11g Release 1. SQL Developer is installed by default when you install the database. Once the installation is complete, locate the sqldeveloper directory (for example, `\product\11.1.0\db_1\sqldeveloper\sqldeveloper.exe`) to start SQL Developer.

Be aware that Oracle database releases are less frequent than those of SQL Developer, which, by its nature and size, allows for more frequent updates. This means the version of SQL Developer shipped with the database may not be the most current release. Oracle Database 11g Release 2 is shipped with SQL Developer 1.5.5. All examples in this text are using SQL Developer 2.1. You may also update your database version less frequently than a client tool.

To upgrade the SQL Developer installation in Oracle Database 11g Release 1, you should do a full new install. As with other installs, create a new folder and unzip the latest download.

> Oracle Database 11g Release 1 ships with SQL Developer 1.1.3
> Oracle Database 11g Release 2 ships with SQL Developer 1.5.5

Troubleshooting

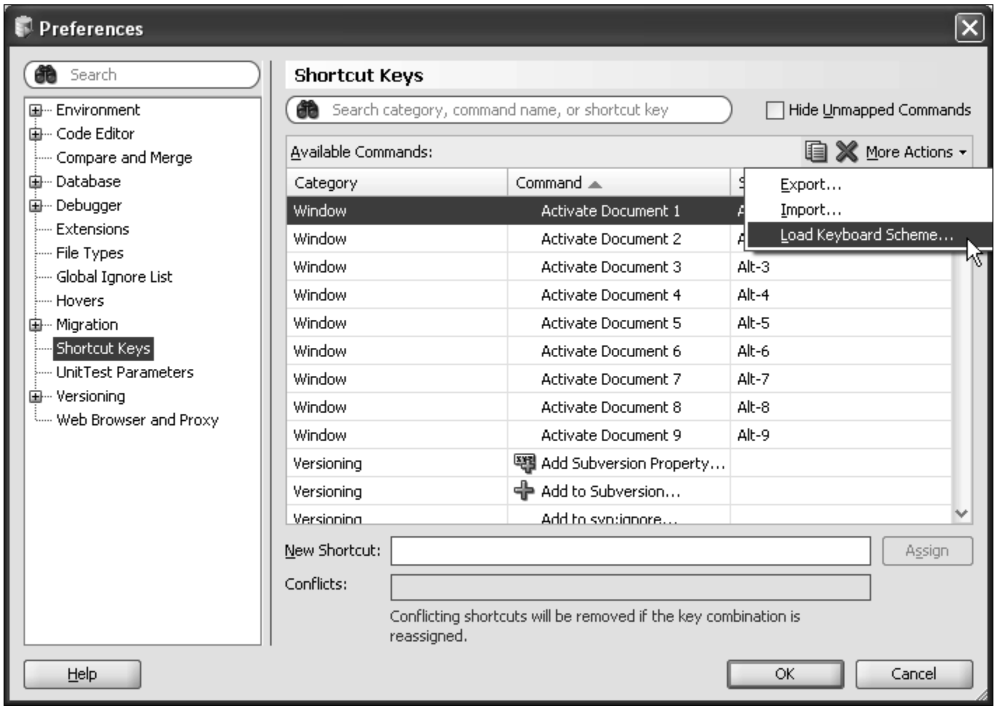
It seems ominous to provide a section on troubleshooting at the start of a book! If you accept that software can get in a tangle sometimes, either if you use the product as it's not designed, or perhaps include extensions that you'd prefer not to have and the product is no longer behaving as expected, then a few hints on how to escape that tangle can be useful.

Removing extensions

If you have created your own extensions, or have downloaded and installed other extensions that you no longer require, then invoke the preferences, using the menu **Tools | Preferences** and select **Extensions** from the tree. Here you see that SQL Developer includes a number of default extensions, such as the Oracle TimesTen extension. In addition, any extension that you have included is listed here. You can deselect extensions here and the product will no longer access them. This does not delete the files installed for the extension. You will need to manually delete any files downloaded for that to happen. However, it does mean that you can restart the product and see if the extension is the root of the problem.

Resetting shortcut keys

Some users find that their keyboard shortcuts no longer work as expected. In this circumstance, you can select the menu **Tools | Preferences,** and then select **Shortcut Keys** from the tree. Click on the **More Actions** drop-down list and select **Load Keyboard Scheme...**, as shown in the following screenshot. Select **Default** from the dialog to reset the keyboard accelerators to the shipped settings. This also replaces any settings you have added.



Category	Command	S
Window	Activate Document 1	A
Window	Activate Document 2	A
Window	Activate Document 3	Alt-3
Window	Activate Document 4	Alt-4
Window	Activate Document 5	Alt-5
Window	Activate Document 6	Alt-6
Window	Activate Document 7	Alt-7
Window	Activate Document 8	Alt-8
Window	Activate Document 9	Alt-9
Versioning	Add Subversion Property...	
Versioning	Add to Subversion...	
Versioning	Add to svn:ignore...	

Conflicting shortcuts will be removed if the key combination is reassigned.

> In releases prior to SQL Developer 2.1, the **Shortcut Keys** are called **Accelerators**. In these releases, to reset the keys, select **Load Preset**.

Reset the environment, do not reinstall the product

When things go wrong, users sometimes resort to deleting and reinstalling a product. This may even require downloading the files again. This is time consuming, and in the case of SQL Developer, not necessary. Assuming you have not edited any of the `.jar` files (it's been known to happen and not legally permitted), you can reset the product to the shipped factory settings by deleting the system folder. Before you delete the system folder, export your connections and shut down SQL Developer.

Export Connections: To export your connections, select **Connections**, right-click and select **Export Connections**. Save the file to a new location.

When troubleshooting, deleting the system folder is useful. However, by deleting this folder you are also deleting all of the changes made to the preferences, your connections, and any layout changes you have made. Therefore, it is recommended that you delete the folder as a last resort, and not as a standard approach to troubleshooting.

Reset to factory settings

For Microsoft Windows, delete the `\Documents and Settings\<your_user>\Application Data\SQL Developer` folder to reset SQL Developer to the shipped factory settings.

For Linux, remove the `~.sqldeveloper` folder and on the Mac, remove the `~/Library/Application Support/SQL Developer` folder.

In addition to deleting all of the preferences set and connections created, this action **also** deletes user-defined reports, your SQL history, and any code templates and snippets you have created. In general, delete the lower level system folder for a less drastic reset.

A quick overview

Let's start with a walk-through of the product. This book is all about SQL Developer, using the product, and getting to know it. You may well ask yourself why there is a need for a book if we can walk through the product in twenty minutes or less. By spending a little time dipping into a number of areas of the product, you can start laying down a map of how the pieces connect and provide a base that you can drill down into later.

Sample schemas

To follow the examples in the book, you need access to SYSTEM and the shipped sample schemas, **HR**, **OE**, **SH**, **PM**, and **IX** available in Oracle Database 9*i*, 10*g*, or 11*g*. Specifically, this book uses the sample schemas shipped with Oracle Database 11*g*.

There are two ways to install the sample schema. The first way is when you install the database. You can elect to have the sample schema installed at that point.

Second, if you have not installed these, then you can locate the sample schema in the `$ORACLE_HOME/demo/schema` folder and follow the instructions on installing them using the Oracle online documentation. Not all of these schemas are available for Oracle Express Edition. In this chapter, we use SYSTEM to verify the HR schema is unlocked, and then we use the HR sample schema, which is available in Oracle Express Edition.

Creating your first connection

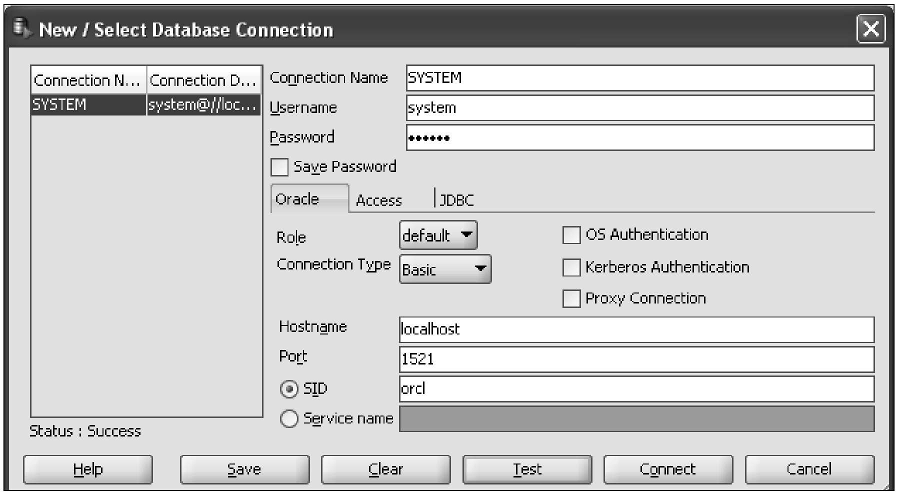
To complete this quick walk-through, you need to know the username and password of the SYSTEM user. You also need to know the location of the database, whether this is the machine name or the IP address, and the database SID.

To begin, start SQL Developer. The very first time you start SQL Developer, you'll be asked if you want to migrate from a previous version. Select **No** and allow the tool to start up.

The first thing you need to do after you have started SQL Developer for the first time is to create your initial connections.

Create a connection for SYSTEM:

1. Select **Connections**, right-click and select **New Connection**. This invokes the **New Database Connection** dialog. You can edit and control all of the connection details using this dialog.
2. Complete the details, as seen in the following screenshot, relevant to your environment.
3. Click on **Test** to ensure you have the connection details correct and click on **Connect**.



New / Select Database Connection

Connection N...	Connection D...
SYSTEM	system@//loc...

Connection Name: SYSTEM
Username: system
Password: ••••••
□ Save Password

Role: default
Connection Type: Basic
□ OS Authentication
□ Kerberos Authentication
□ Proxy Connection

Hostname: localhost
Port: 1521
● SID: orcl
○ Service name:

Status : Success

You are now connected as SYSTEM. Use this connection to verify your other users, by continuing as follows:

1. Select the new connection you have created, expand the node, and scroll down to **Other Users**.
2. Expand **Other Users** and find the user **HR**. Right-click and select **Edit User**. Verify that the account for HR is unlocked and the Password has not expired, that is, the properties **Account is Locked** and **Password Expired** are deselected. If either of these is selected, deselect them. You can change the password for HR at this point too. It's good practice to modify the passwords of the shipped sample schemas once you have unlocked them.

Now you are really ready to begin!

1. Once again, select **Connections**, right-click and select **New Connection**.
2. Give the connection a name (for example, **HR_11g**).
3. Provide the **Username** (**HR**) and a **Password**. If you are working on Oracle Database 11g, be aware that passwords are now case sensitive.
4. Select the **Save Password** checkbox. This makes life easy while you are working with SQL Developer. Passwords are stored in an encrypted file. However, you should always be aware of saving passwords and possible security implications this may have.
5. Use the **Basic** connection. This requires no more detail than the location of the database and the **SID**, details you have.
6. Click on **Test** to test the connection.
7. Click on **Connect**.

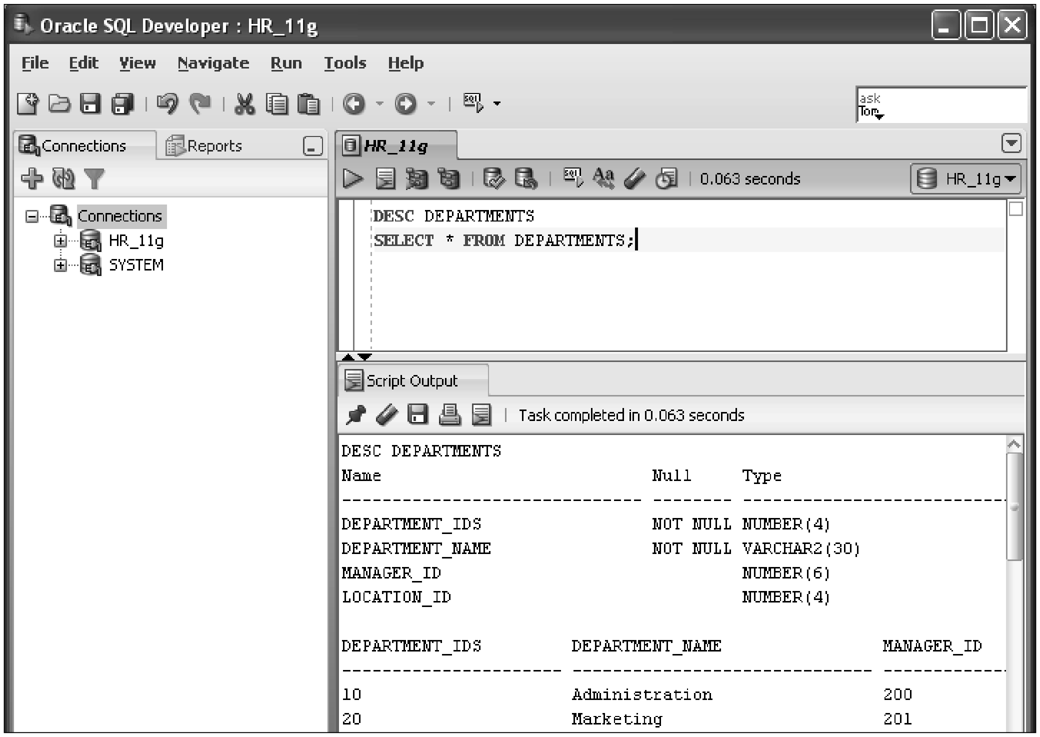
Using basic commands in the SQL Worksheet

As soon as you connect to a user, SQL Developer opens an SQL Worksheet. You may have started working with Oracle using the SQL*Plus command line, or even the GUI window. Either way, you'd start with a selection of SQL*Plus and SQL commands.

Enter the following into the SQL Worksheet:

```
DESC DEPARTMENTS
SELECT * FROM DEPARTMENTS;
```

Press the *F5* key (or use the **Run Script** button).



Oracle SQL Developer : HR_11g
File Edit View Navigate Run Tools Help
Connections
Reports
HR_11g
0.063 seconds
Connections
HR_11g
SYSTEM
Script Output
Task completed in 0.063 seconds

```
DESC DEPARTMENTS
SELECT * FROM DEPARTMENTS;
```

```
DESC DEPARTMENTS
Name                           Null     Type
------------------------------ -------- ---------------------------
DEPARTMENT_IDS                 NOT NULL NUMBER(4)
DEPARTMENT_NAME                NOT NULL VARCHAR2(30)
MANAGER_ID                              NUMBER(6)
LOCATION_ID                             NUMBER(4)

DEPARTMENT_IDS         DEPARTMENT_NAME                MANAGER_ID
---------------------- ------------------------------ -----------
10                     Administration                 200
20                     Marketing                      201
```

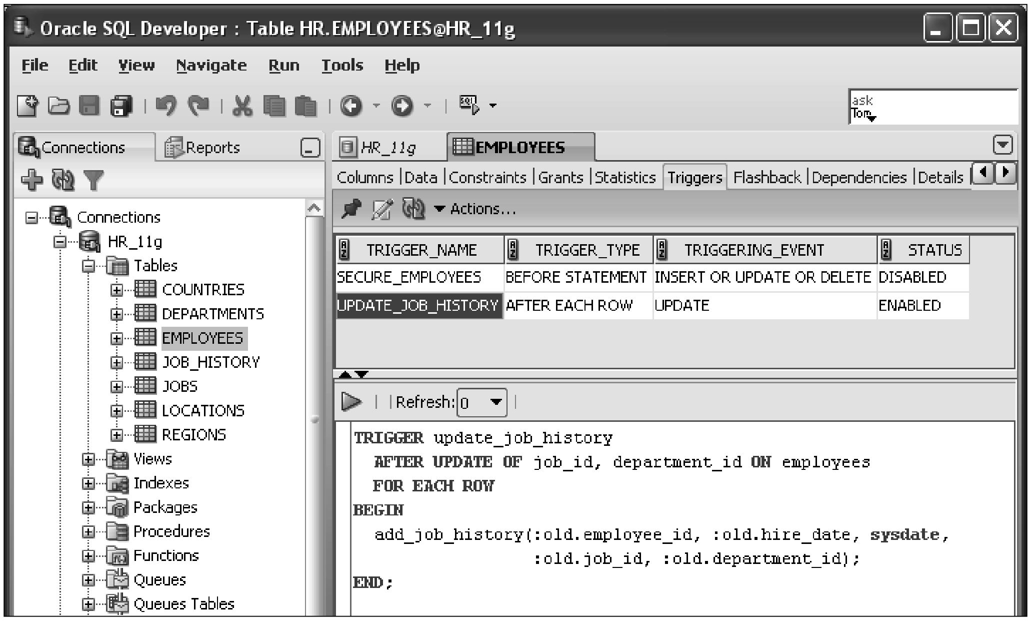
The output of both commands appears in the **Script Output** tab, which appears below the SQL Worksheet (as seen in the previous screenshot). Both commands are handled by a few simple clicks of the mouse in SQL Developer.

Select and expand the **HR_11g** connection in the **Connections** navigator. Expand the **Tables** node and select **DEPARTMENTS**.

The **DEPARTMENTS** tab now opens, displaying a list of the column names and details. These are the same details as given by the `DESC` (describe) SQL*Plus command that you entered in the SQL Worksheet. It also provides additional detail, such as the Primary Key and column comments.

Select the **Data** tab and notice that you now see the output from your second command. These two tabs are included with a number of other tabs, each with additional details about the **DEPARTMENTS** table. You would need to write a number of SQL queries in order to get the additional detail from the data dictionary if you were working in SQL*Plus.

Select the **EMPLOYEES** table. Notice that the new table, **EMPLOYEES**, immediately replaces the previous **DEPARTMENTS** table with its details. Select the **Triggers** tab, and select one of the triggers. The trigger and related trigger detail is displayed in a master-detail window:



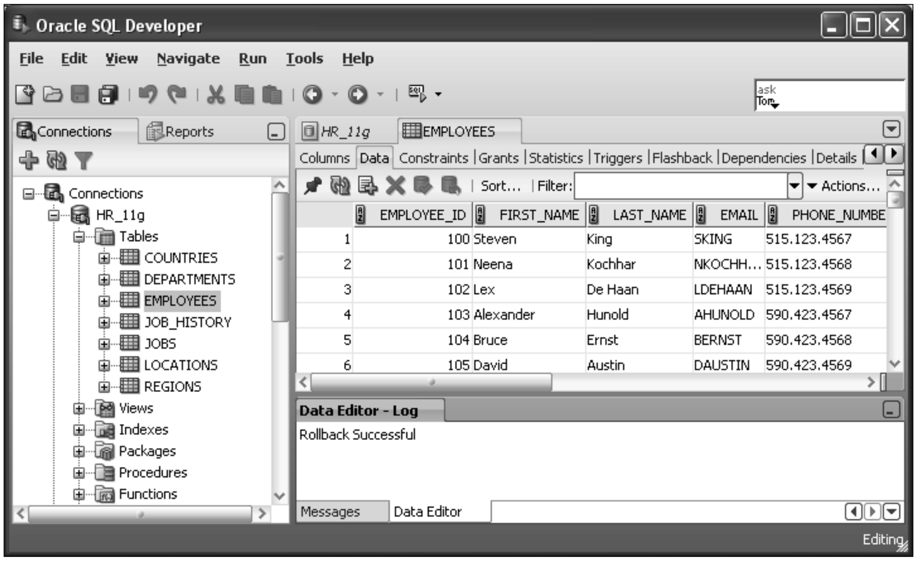
TRIGGER_NAME	TRIGGER_TYPE	TRIGGERING_EVENT	STATUS
SECURE_EMPLOYEES	BEFORE STATEMENT	INSERT OR UPDATE OR DELETE	DISABLED
UPDATE_JOB_HISTORY	AFTER EACH ROW	UPDATE	ENABLED

```
TRIGGER update_job_history
  AFTER UPDATE OF job_id, department_id ON employees
  FOR EACH ROW
BEGIN
  add_job_history(:old.employee_id, :old.hire_date, sysdate,
                  :old.job_id, :old.department_id);
END;
```

Browsing and updating data

Return to the **EMPLOYEES** data by again selecting the **Data** tab. The data grid that is displayed provides a variety of options. To get started with the data grid, double-click on an item or field, such as the name of one of the employees, and change it. Tab out of the field and notice that the change is applied to the data grid and an asterisk (*) flags the record. **Commit** and **Rollback** buttons are available to send the change to the database, or to undo your action. Roll back the changes.

Once again you get feedback, this time in the **Data Editor** log, as shown in the following screenshot:



	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBE
1	100	Steven	King	SKING	515.123.4567
2	101	Neena	Kochhar	NKOCHH...	515.123.4568
3	102	Lex	De Haan	LDEHAAN	515.123.4569
4	103	Alexander	Hunold	AHUNOLD	590.423.4567
5	104	Bruce	Ernst	BERNST	590.423.4568
6	105	David	Austin	DAUSTIN	590.423.4569

Rollback Successful

Running reports

Select the **Reports** navigator and expand the **Data Dictionary Reports** node. Expand the **Table** node and review the available reports. Expand **Constraints** and select the **Unique Constraints** report. As you select the report, a dialog displays requesting the Connection name. Select the connection you created, **HR_11g**, and click on **OK**. An **Enter Bind Values** dialog now appears, requesting the table name as an input parameter. Click on **Apply** to accept the default, which in this case, means all tables:

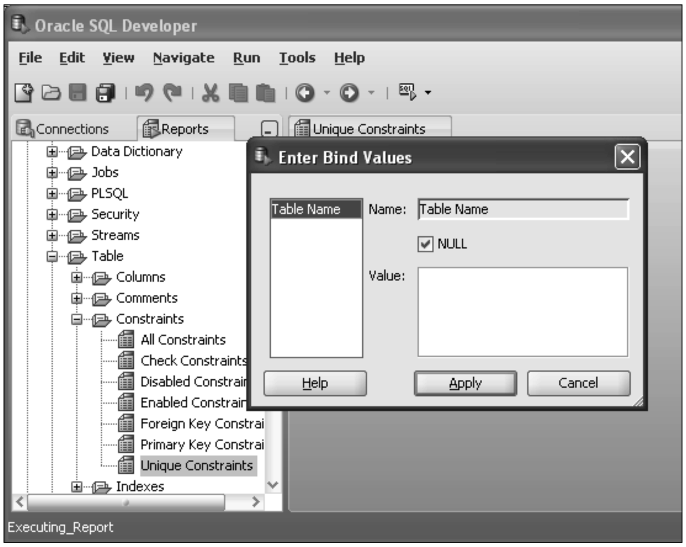


Table Name

Name: Table Name

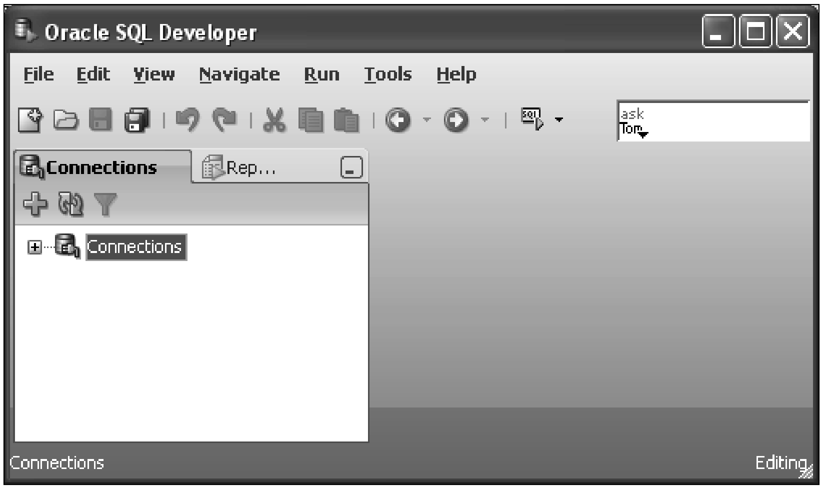
☑ NULL

Value:

Run the same report for any user by selecting the **Connections** drop-down list on the right-hand side.

Navigating around SQL Developer

SQL Developer has a selection of windows, navigators, and tabs. On start-up, you are presented with the main navigator toolbars and menus:

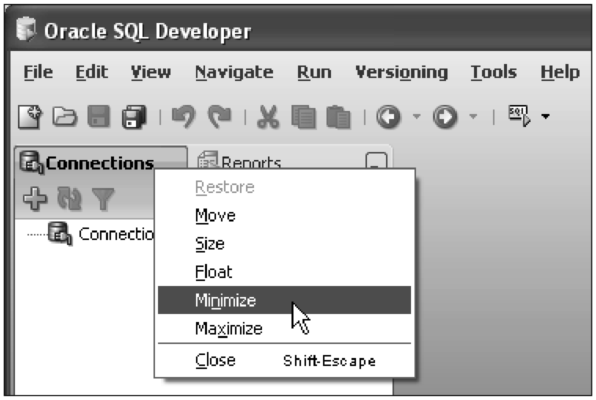


The two main navigators: **Connections** and **Reports**, are presented in a tabbed window. These and other navigators, such as the **Versioning Navigator**, are available through the main **View** menu. You can also open windows such as **Snippets**, **Recent Objects**, and **Find DB Objects** using the **View** menu.

Any navigators that you open during a session, and that are still open when you close the product, are automatically opened when you restart the product.

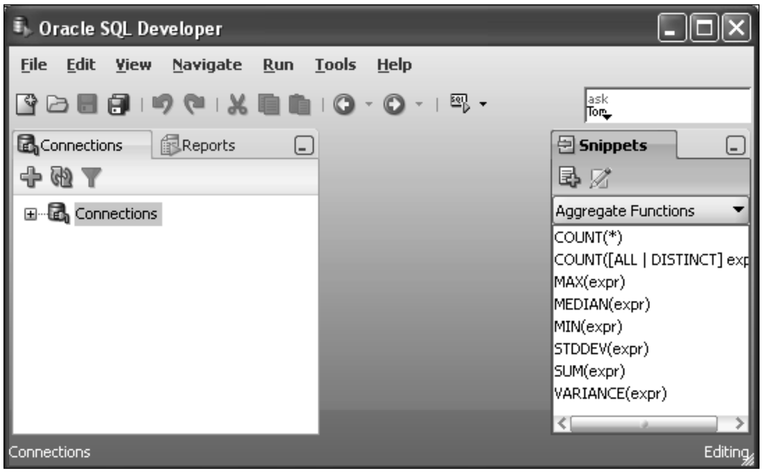
Managing SQL Developer windows

With the exception of the SQL Worksheet and its associated tabs, all of the main tabbed dialogs can be minimized or maximized and accessed while docked or undocked. These menu controls are available through context menus in the tabs:



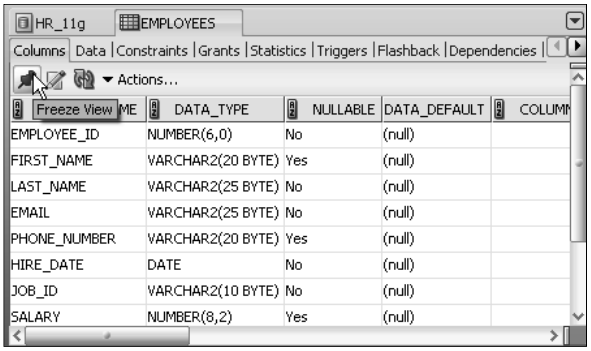
You can rearrange tabbed windows by selecting and dragging the tab into place. Once any window is minimized, roll your mouse over the minimized tab to display a floating window that stays active while your mouse lies over it and rolls back into place when you move off. This is very useful when working with temporary windows such as **Snippets** and **Find DB Object**. The following screenshot shows the floating window for the **Snippets** dialog. If you roll the mouse over the area, you can work in the window (for example, navigating about until you have located the snippet of code you are after, and then drag the code onto the worksheet). The window will minimize out of the way once you have moved off it.

You can undock the floating window, move it off to one side, and keep it undocked while you work with the SQL Worksheet. In a dual-monitor setup, you can drag the floating window onto one monitor, while working with the SQL Worksheet on the other monitor.



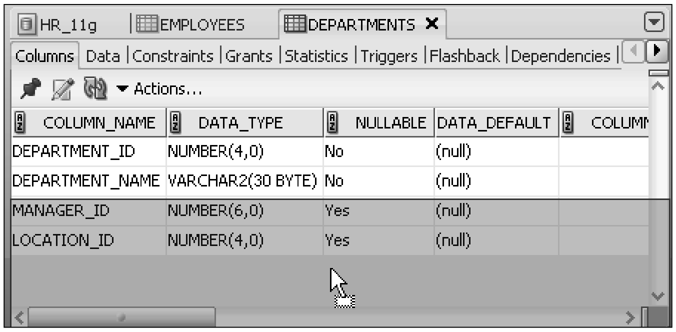
Tiling windows

Once you start working with connections, you have more windows and tabs to deal with, especially if you have more than one connection created. Select the **HR_11g** connection created in the previous section, expand the connection and **Tables** node, and select **EMPLOYEES**. In the table definition window, select the pin button, as shown below, to freeze the view.

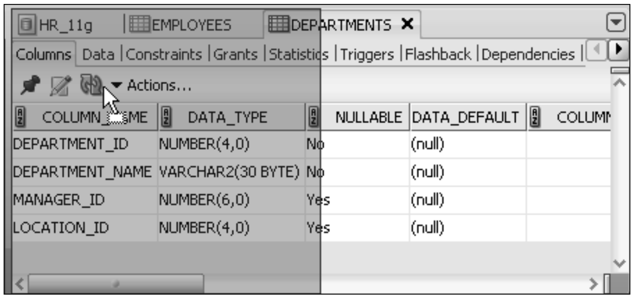


ME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN
EMPLOYEE_ID	NUMBER(6,0)	No	(null)	
FIRST_NAME	VARCHAR2(20 BYTE)	Yes	(null)	
LAST_NAME	VARCHAR2(25 BYTE)	No	(null)	
EMAIL	VARCHAR2(25 BYTE)	No	(null)	
PHONE_NUMBER	VARCHAR2(20 BYTE)	Yes	(null)	
HIRE_DATE	DATE	No	(null)	
JOB_ID	VARCHAR2(10 BYTE)	No	(null)	
SALARY	NUMBER(8,2)	Yes	(null)	

Now, select the **DEPARTMENTS** table. A second table definition window opens to display the details from the new table. Select the **DEPARTMENTS** tab and drag it down to the lower portion of the screen. Notice the shape of the dragged object change as you drag it slightly to the left, to the center, and the lower portion of the window. Each of the shapes represents a different layout position. Release the mouse to secure the new position. The screenshots, which follow, display two of the available positions:



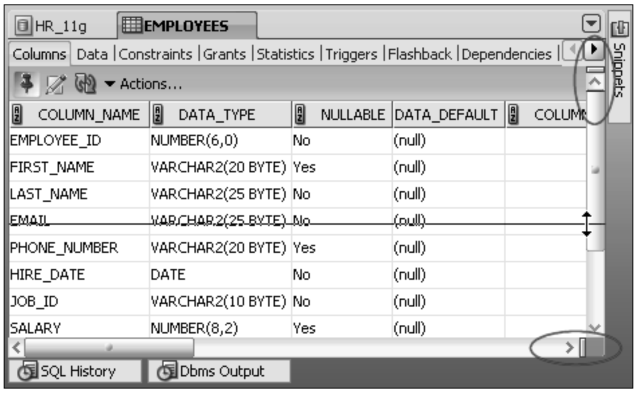
COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN
DEPARTMENT_ID	NUMBER(4,0)	No	(null)	
DEPARTMENT_NAME	VARCHAR2(30 BYTE)	No	(null)	
MANAGER_ID	NUMBER(6,0)	Yes	(null)	
LOCATION_ID	NUMBER(4,0)	Yes	(null)	



COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN
DEPARTMENT_ID	NUMBER(4,0)	No	(null)	
DEPARTMENT_NAME	VARCHAR2(30 BYTE)	No	(null)	
MANAGER_ID	NUMBER(6,0)	Yes	(null)	
LOCATION_ID	NUMBER(4,0)	Yes	(null)	

Splitting documents

When you tile windows, you can compare the details of two tables. However, as each table has a selection of tabs, it's useful to be able to review details in the tabs without having to switch back and forth between tabs. As is true for other layout features, you can split the document using a menu, or by drag-and-drop. Each of the object definitions tabbed displays has a drag bar on the top and bottom right that you can select and drag to split the window horizontally, or vertically:



COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN
EMPLOYEE_ID	NUMBER(6,0)	No	(null)	
FIRST_NAME	VARCHAR2(20 BYTE)	Yes	(null)	
LAST_NAME	VARCHAR2(25 BYTE)	No	(null)	
EMAIL	VARCHAR2(25 BYTE)	No	(null)	
PHONE_NUMBER	VARCHAR2(20 BYTE)	Yes	(null)	
HIRE_DATE	DATE	No	(null)	
JOB_ID	VARCHAR2(10 BYTE)	No	(null)	
SALARY	NUMBER(8,2)	Yes	(null)	

Maximizing detail

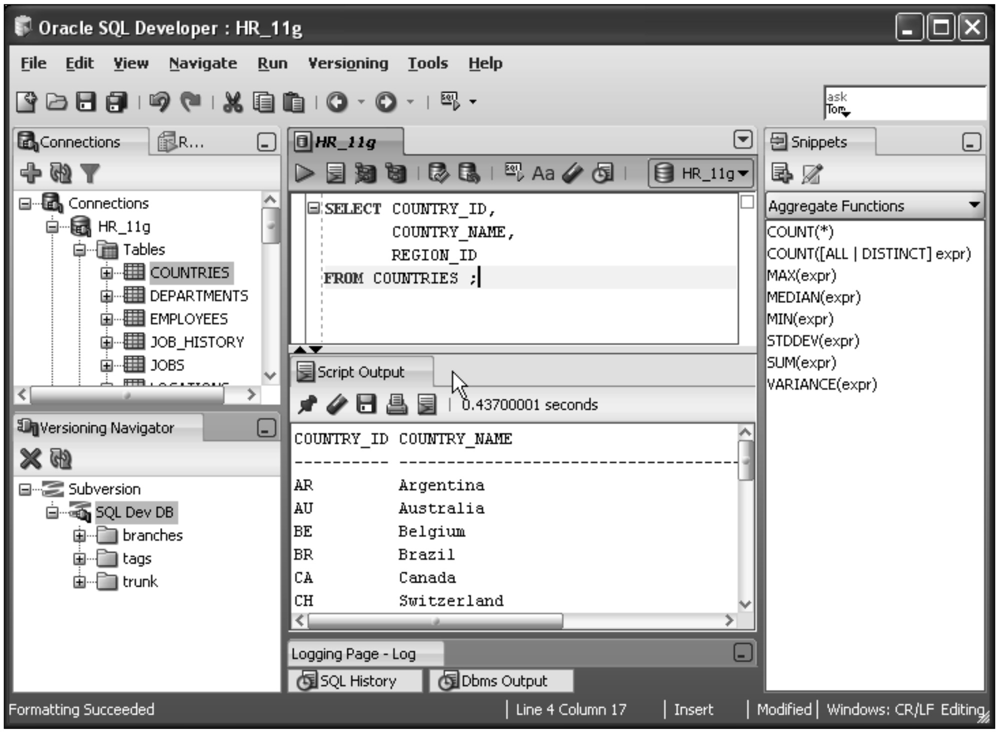
Almost all of the tabs in SQL Developer will maximize when double-clicked. There are a few that do not follow this rule, such as the tabs related to the SQL Worksheet. In general, this works for top-level tabs, which is any tab you can undock and move about, and not for secondary tabs. To maximize a tab, double-click on the tab. A second double-click will reverse the process.

Double-click on the tab to maximize a top-level tab.
Double-click again to revert to the previous layout.

Resetting the window layout

If you move your windows about a great deal, you may find that you want to get things back to the default settings.

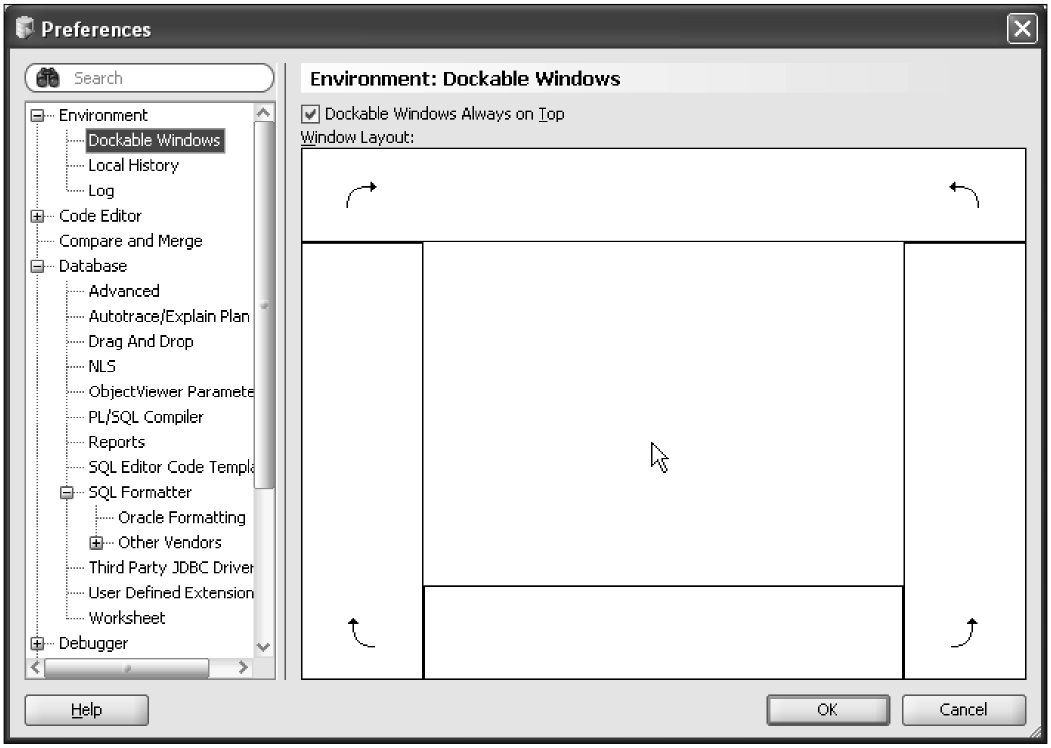
The example in the following screenshot displays the standard docked **Connections** and **Reports** windows to the left. We have also opened the **Versioning Navigator**, which by default docks below the connections. We have also docked the **Snippets** window to the right. These windows fill the columns to the left and right, leaving a central window for the editors and log files.



```
SELECT COUNTRY_ID,
       COUNTRY_NAME,
       REGION_ID
FROM COUNTRIES ;
```

```
COUNTRY_ID COUNTRY_NAME
---------- ----------------------------------------
AR         Argentina
AU         Australia
BE         Belgium
BR         Brazil
CA         Canada
CH         Switzerland
```

The layout is controlled by the window layout set in the preferences. Select **Tools | Preferences**, under the **Environment** node in the tree select **Dockable Windows.** The default layout, and the one that matches the example in the previous screenshot, is shown in the following screenshot:



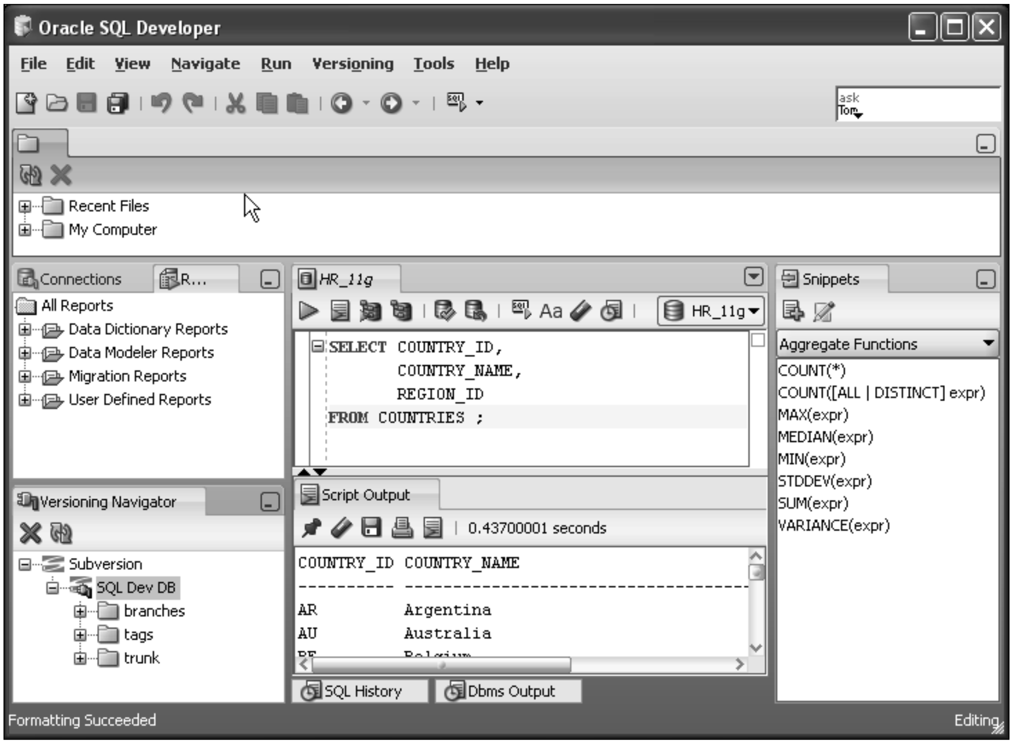
Environment: Dockable Windows

☑ Dockable Windows Always on Top

Window Layout:

Each of the little curved arrows on the diagram is clickable, and as such controls the positioning of the windows. Clicking on the arrow extends or contracts the area taken up by the docked window.

In our example, and in the default SQL Developer environment, there is no full-docked window across the top of the screen. However, if you drag a window into the docked position below the main tool bar, it would stretch across the screen, as shown in the following screenshot:



```
SELECT COUNTRY_ID,
       COUNTRY_NAME,
       REGION_ID
FROM COUNTRIES ;
```

0.43700001 seconds

```
COUNTRY_ID COUNTRY_NAME
---------- ----------------------------------------
AR         Argentina
AU         Australia
BE         Belgium
```

If you find your windows are in a muddle, first verify that the **Dockable Windows** layout is what you want, and then drag the various dockable windows back into place. Some suggestions on the SQL Developer forum are to remove the system folder (it works, but that's an extreme solution).

Finding more help

SQL Developer has a site on the Oracle Technology Network, `http://www.oracle.com/technology/products/database/sql_developer/index.html`. This provides links to current and past magazine articles, white papers, and team blogs. It also has links to brief product demonstrations and longer hands-on exercises.

There is an active user forum on OTN, `http://forums.oracle.com/forums/forum.jspa?forumID=260`, which is monitored by the development team and end users.

The SQL Developer Exchange, `http://sqldeveloper.oracle.com`, is a site where anyone using SQL Developer can log feature requests and vote on other requests already posted. In addition to posting feature requests, the site hosts reports and code snippets.

Summary

You've started and should now have SQL Developer installed. You should have a few connections created and an initial idea of how to navigate around the product. You are now set to learn a lot more about SQL Developer. From here you can dip into different chapters, focusing on the areas you're most interested in.

In the next chapter, we'll show you how to browse different types of objects, and use SQL Developer to look at them in greater detail. We'll review the different editors and dialogs available and how you can manage what you see using preferences. We'll also show you the different ways you can create objects and how to manipulate the data.

Where to buy this book

You can buy Oracle SQL Developer 2.1 from the Packt Publishing website:
`http://www.packtpub.com/oracle-sql-developer-2-1/book`

Free shipping to the US, UK, Europe and selected Asian countries. For more information, please read our shipping policy.

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.

PACKT PUBLISHING

www.PacktPub.com