



SAGE Computing Services

Oracle SQL Developer Workshop

0419 904 458
www.sagecomputing.com.au

Oracle SQL Developer Workshop

Edition AUSOUG Conference 2006

© SAGE Computing Services 2005 - 2006

SAGE Computing Services believes the information in this documentation is accurate as of its publication date. The information is subject to change without notice.

SAGE Computing Services assumes no responsibility for any errors that may appear in this document.

JDeveloper, Oracle SQL Developer, 10G Internet Developer Suite, Application Express, Oracle, Oracle Internet Application Server, Oracle Discoverer, SQL*Plus, Oracle Forms Developer, Oracle Reports Developer, Oracle Graphics, Oracle Developer, Oracle Designer, Oracle Internet Developer Suite are registered trademarks of Oracle Corporation.

Oracle8i, Oracle9i, Oracle 10G and PL/SQL, are trademarks of Oracle Corporation.

Microsoft and Microsoft Excel are registered trademarks of Microsoft Corporation.

Windows is a trademark of Microsoft Corporation.

All trade names referenced are the services mark, trademark, or registered trademark of the respective manufacturer.

Microsoft Word was used in the production of this document.

SAGE Computing Services would like to thank Sue Harper, Oracle Corporation Database Tools group for the SQL Developer materials on which this presentation was based.



SAGE Computing Services

A Company Profile

Telephone: 0419 904 458
Facsimile: 08 9243 4335
www.sagecomputing.com.au

Services

Services provided by the company include:

- Customised Training Programs
- Post-training Mentoring
- Database Administration Services
- Complete Systems Development and Support
- Consultancy Advice
- Quality Assurance
- Tuning Reviews
- Business Analysis
- Telephone Advice and Assistance
- Database Health Checks

SAGE Computing Services Pty. Ltd. aims to provide the best Oracle technology experts to assist our clients in building, implementing and managing business solutions using Oracle software. Sage offers custom training workshops in Oracle products and a complete analysis, design, development and database administration consulting service.

SAGE is a small company which has succeeded by adopting a partnership approach with our clients. We have a long term relationship with many of our clients which is dependent on our providing services which achieve successful outcomes.

enquiries@sagecomputing.com.au

Course Descriptions

(not a full list of available courses, please see www.sagecomputing.com.au for a full course catalogue)



All courses can be customised to suit client requirements

enquiries@sagecomputing.com.au

Oracle JDeveloper 10.1.3

5 days

This course is designed to provide students with the skills and knowledge required to develop a web application using ADF Business Components and ADF Faces (JavaServer Faces) web pages within Oracle JDeveloper 10.1.3.

- | | | |
|--------------------------------------|---|---|
| • JDeveloper Introduction | • ADF Entity Objects | • ADF Input, Output and Command Components |
| • JDeveloper IDE | • ADF Associations | • ADF Forms, Tables, Trees |
| • Database Connections | • ADF Domains | • JSF Configuration, Navigation, Managed Beans and Event Handling |
| • Online Database Browser | • ADF Application Modules | • Expression Language |
| • Stored Procedure Editor | • ADF View Objects | • ADF Page Layout and Menus |
| • Workspaces and Projects | • ADF View Links | • ADF Faces Skins |
| • Offline Database Modelling Tools | • ADF Business Component Java Classes | • Deploying an ADF Application |
| • Java Language | • ADF Binding Layer and the ADF Model Layer Classes | • Web Security |
| • Application Development Framework | • ADF Faces | • ADF Selection, Validator, Converter and Visual Components |
| • ADF Business Components Diagrammer | • ADF Form | • Code Management and Source Control |
| • ADF Business Components | • ADF Read-Only Table | • Advanced JDeveloper |

Oracle 10G - New Features for Developers Workshop

1 day

Aimed at developers, this course is designed to provide the student with an understanding of the new features of Oracle 10G.

- | | | |
|-------------------------|-----------------------|----------------------------|
| • SQL*Plus Enhancements | • New SQL Features | • PL/SQL Packages |
| • Flashback | • Data Pump | • Performance Enhancements |
| • Enhanced SQL commands | • PL/SQL Enhancements | |

Application Express Workshop (previously HTMLDB)

3 days

The course is designed to provide the student with the skills and knowledge required to develop a complete application using Oracle's Application Express product. The student will develop web interfaces (including forms, reports and charts; addition of validation and customised formatting) to create a small application.

- | | | |
|-----------------------|-------------------------------|---|
| • Product Overview | • Regions and Items | • Utilities and Reporting |
| • SQL Workshop | • Page Processing | • Advanced Development Techniques |
| • Utilities | • Shared Components | • Administration and Deploying an Application |
| • Application Builder | • Themes and Templates | |
| • Creating Pages | • Other Page and Region Types | |

Oracle SQL and SQL*Plus Workshop - Oracle 10G Rel2

4 days

This course is designed to provide the student with a basis for developing systems using the Oracle database. The SQL language is covered from simple to complex constructs. Guidelines are provided on writing SQL for optimum performance and ease of maintenance.

- The Relational Model
- The SQL Language
- SQL*Plus and iSQL*Plus
- Oracle SQL Developer
- More about SELECT
- Substitution Variables
- Using SQL*Plus for Formatting Output
- Functions
- Joins
- Group and Analytical Functions
- Set Operators
- Subqueries
- Data Manipulation Language
- Database Objects
- Constraints
- Views and Sequences
- Indexes
- Clusters
- Security
- Locking and Read Consistency
- More Advanced SQL

PL/SQL Workshop

3 days

This course is for developers who will be designing or building applications using the Oracle server. It is relevant for developers who are using the Oracle Developer toolset, and for those using alternative front-end products accessing the Oracle database. The course covers basic PL/SQL syntax and the use of server level procedures, functions and triggers.

- PL/SQL Overview
- Basic PL/SQL Syntax
- SQL Statements in PL/SQL
- Procedural Statements - Assignment and Conditional Processing
- Procedural Statements - LOOPS
- Exceptions
- Nested Blocks and Cursors
- Tables, Arrays and Records
- Architecture Overview
- Procedures / Functions
- Execution and Error Handling
- Security and Dependency
- Packages
- More About Packages
- Supplied Packages
- Triggers
- More About Triggers
- Large Objects

Oracle Forms Developer Workshop

5 days

This course is designed for developers who will be designing or building applications using Oracle Form Builder. This is a practical course in which the student builds an application during a series of workshop sessions.

- Running a Form
- Forms Modules and Storage
- Working in the Builder
- Creating a Form
- Form and Data Block Properties
- Form Layout
- Items
- Introduction to Triggers
- Program Units
- Check Boxes, Radio Groups and List Items
- Other Item Types
- Visual Attributes
- Mouse Events
- Relations
- Alerts and Editors
- Lists of Values
- Record Groups
- Windows and Canvases
- Transaction Processing and Triggers
- Advanced Data Block Properties
- More Trigger Events
- Determining Form Properties
- Timers
- Integrating Multiple Forms Modules in an Application
- Forms Architecture and Java
- Integrating Forms with Reports
- PL/SQL Library Modules
- Managing Application Development
- Menu Modules

Oracle Reports Workshop - 10G

4 days

This course is designed for developers who will be designing or building applications using Oracle Reports. This is a practical course in which the student builds a series of reports ranging from simple to complex.

- Product Overview
- The User Interface
- The Designer Interface
- Storage
- Building a Paper Report
- The Data Model Editor
- Other Query Types
- Columns
- Multiple Queries and Links
- The Paper Layout - Basic Objects
- Standard Layouts
- General Paper Layout Properties
- Advanced Paper Layout Properties
- Web Reports
- Displaying Files, Images and Charts
- Matrices
- Parameters
- PL/SQL in Reports
- Report Templates
- Publishing Reports on the Web

Oracle 10G – Database Administration Workshop

5 days

This course is designed for Database Administrators. It covers the architecture of the Oracle 10g server, and the procedures required to effectively administrate the database. The course provides a series of practical workshops in which the students can practice the database administration techniques they have learnt.

- Oracle 10g Overview
- Oracle 10g Architecture
- Database Creation
- Startup and Shutdown and Oracle Database
- Oracle Enterprise Manager
- Database Structure
- Managing Tablespaces
- Managing Redo Log Groups and Members
- Database Storage
- Managing Undo
- Security
- Optimisation
- Database Tuning
- The Multi Threaded Server
- Backup and Recovery
- Data Pump

Application Tuning Workshop 10g

3 days

This course is designed for Designers, Developers, and Database Administrators, and examines all aspects of tuning SQL statements and applications.

- Defining a Tuning Methodology
- Diagnostic Tools
- Processing an SQL Statement
- Indexes
- Cost Based Optimisation
- Gathering Statistics
- Stored Outlines
- Storage Parameters
- Hash Clusters and Index Clusters
- Optimising PL/SQL
- Optimising Applications through Stored Procedures and Packages
- Data Design for Performance
- Tuning Tips
- Partitions
- Optimise using Parallelisation
- Tuning Tools

Oracle Discoverer Workshop

2 days

This course is designed for End Users and examines all aspects of using the latest versions of Oracle Discoverer. Both the web and client server interfaces are covered.

- Oracle Discoverer Overview
- Discoverer Workbooks
- Worksheets and Conditions
- Performing Analysis
- Customising Workbooks and Worksheets
- Printing and Exporting Query Results
- Scheduling and Administration

Oracle Portal Workshop

3 days

This course is designed to provide the student with the knowledge and skills required to build corporate portals. The course covers the use of Oracle Portal for content management and includes recommendations and guidelines on the classification and searching of content. The standardisation and customisation of the Portal interface and styles are described. The workshop includes the use of Portal to create simple application components such as forms, reports and graphs. Finally the security management of a corporate portal is considered.

- Product Overview
- Custom Types, Parameters and Events
- Other Components
- Page Groups and Pages
- Application Components - Forms
- Security
- Styles, Navigation Bars and Templates
- Application Components - Reports
- Item Regions and Classification
- Shared Component

Advanced Oracle Portal Workshop

1 day

The course is designed to provide the student with the knowledge and skills required to build custom portlets. The course describes the provider and portlet structure and the integration and management of custom portlets within the product. The attendee builds a simple provider and its custom portlets based on an example. The course focuses on PL/SQL portlets to demonstrate the techniques required, but also covers web portlets. Detailed coverage of the Portal API and its use, not only in custom portlets, but to enhance other Portal components is included.

- Programmable Portlets - Concepts
- PL/SQL Portlets
- API Services – Other Utilities
- PL/SQL Providers
- API Services – Session Storage
- Web Providers

Oracle End User Workshop

2 days

This course is designed for End Users who require a knowledge of SQL to query the Oracle database. It commences with a description of relational concepts and continues with coverage of the SQL statements required to access information from one or more Oracle Tables. Some basic formatting is also covered.

- The Relational Model
- More About SELECT
- Joins
- Structured Query Language
- Substitution Variables
- Group Functions
- SQL*Plus
- Using SQL*Plus for Formatting Output
- Oracle SQL Developer
- Functions

All our training is conducted at the client site and using the client's Oracle licences. Sage Computing Services provides all course materials which the attendees retain after the course as a reference.

All Sage trainers are consultants who are using the products in real world situations and can bring a wealth of experience to the classroom.

SECTION 1 - OVERVIEW	9
The Product	9
Interface	9
Help	10
Help Centre	10
Online Help Manual	10
Context Level Help	11
Database Connections	11
Set up	11
1 Hands On - Getting Started	12
1.1 Invoke SQL Developer	12
1.2 Create a Database Connection	12
More on Connections	14
Modifying Database Connections	14
Exporting and Importing Connections	15
Invoking SQL*Plus from SQL Developer	16
SECTION 2 – DATABASE OBJECTS	16
2 Hands On - Database Objects	17
2.1 Browsing of Database Objects	17
2.2 Creating Database Objects	19
2.3 Modifying Database Objects	24
2.4 Deleting Database Objects	26
SECTION 3 – SQL QUERIES	27
The SQL Language	27
Structure	27
Selecting All Rows	27
Expressions	28
Column Alias	28
Restricting the Rows	29
SELECT first_name, last_name, salary	30
FROM employees	30
WHERE salary*12 >100000	30
Ordering the Data	30
Group Functions	31
SUM	31
COUNT	31
Grouping the Output into Sets of Rows	31
Joins	32
Foreign keys	32
Select Statements with Join Conditions	32
Using a Table Alias	32
3 Hands On – SQL and the SQL Worksheet	33
3.1 The SQL Worksheet Toolbar	33
3.2 Using the SQL Worksheet to perform SQL statements	33
3.3 SQL History List	38
4 Hands On - PL/SQL Development	39
4.1 Creating and Editing PL/SQL	39
4.2 Execution and Debugging	41
5 Hands On – Reports	46
5.1 Data Dictionary Reports	46
5.2 User Defined Reports	47
5.3 Standard Actions	49
5.4 Bind Variable Reports	49

Section 1 - Overview

The Product

Functionality

Oracle SQL Developer is a free GUI tool aimed at simplifying and enhancing database development tasks. It facilitates the execution of SQL statements and scripts, the manipulation of Database Objects, editing and debugging of PL/SQL and the running of Database Reports.

SQL Developer has been developed in the JDeveloper IDE and works with any Oracle database 9.2.0.1 or later. It is installed simply by unzipping the downloaded file.

The functionality of the product includes:

- Creating, storing, importing and exporting database connections
- Creating and browsing users
- Browsing database objects using a tree based browser
- Creating database objects
- Querying and updating data using a grid interface
- Running SQL statements and scripts
- Entering, running and debugging PL/SQL
- In-built and custom Reporting

Interface

You will operate SQL Developer using the following features:

- Use the Navigator pane on the left hand side of the screen to locate and select objects. Use the right mouse menu to display context sensitive actions.
- Use the tabbed panes on the right side of the screen to display information about objects and to enter SQL and PL/SQL statements.

Click on a tab to display its information.

To close a tab, place your mouse cursor over the tab and click on the X that is displayed.

Double click on a tab to maximise. (Double clicking again returns to its previous state.)

Drag the boundaries between panes to change their sizes.

- Optionally display the message area at the bottom of the screen (using the menu option View, Log)
- Optionally display the Help Centre on the right of the screen (by selecting options from the Help menu).

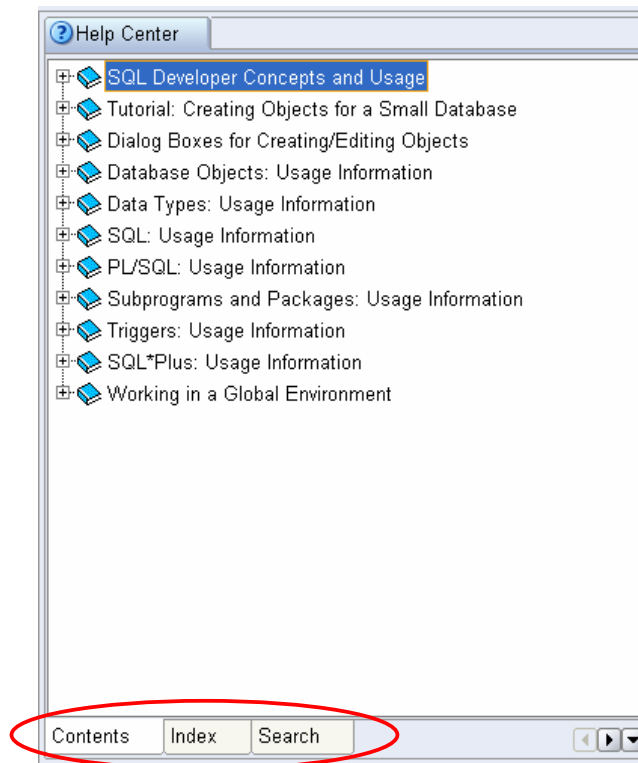
Note that this describes the default interface. The SQL Developer interface can be customised using a series of preferences (using the menu option Tools, Preferences). You can also “dock” the panes in different positions by dragging the tabs to a new location.

SQL Developer is expected to be included in JDeveloper post release 10.1.3.

Help

Help Centre

The Help Centre Area within SQL Developer allows you to access all help information, as well as perform a full text search. To display the Help panel, select Help from the menu and then select the type of help you require, either: Table of Contents, Full Text Search or Index. You can switch between these by using the tabs at the bottom of the Help Centre.



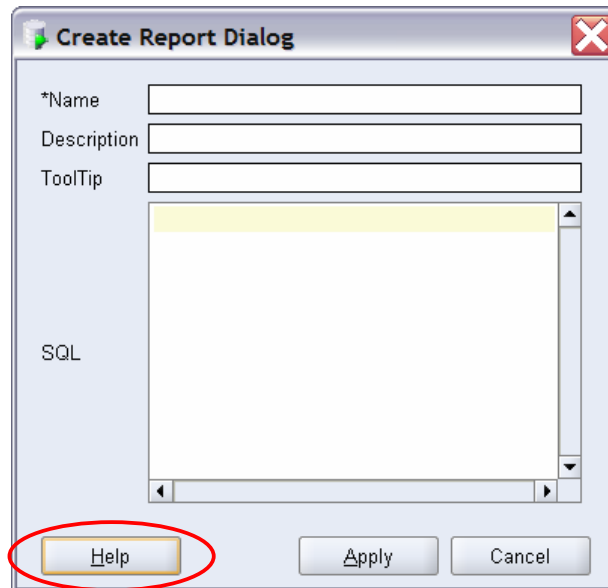
Online Help Manual

The Oracle SQL Developer Online Help document, available from the Oracle Technology Network web site, is a PDF form of the Help Centre within SQL Developer. You can view or download it from the SQL Developer Home Page on the Oracle Website. The following is a direct URL to the Online Help Document.

http://www.oracle.com/technology/products/database/sql_developer/pdf/online_help.pdf

Context Level Help

SQL Developer includes a context level help feature which can be accessed using the F1 key throughout the application. In addition dialogs such as the Create Object dialog contain context specific help that is accessed by selecting the displayed Help button.



Database Connections

Before you can use SQL Developer to access information from an Oracle database you must create a connection. A connection specifies which database you are connecting to and the username you will use to connect.

Set up

The hands on sessions which follow assume that you have followed the set up instructions provided, and have access to the product and the HR schema. If you have any problems with the hands on exercises please ask the presenter for assistance.



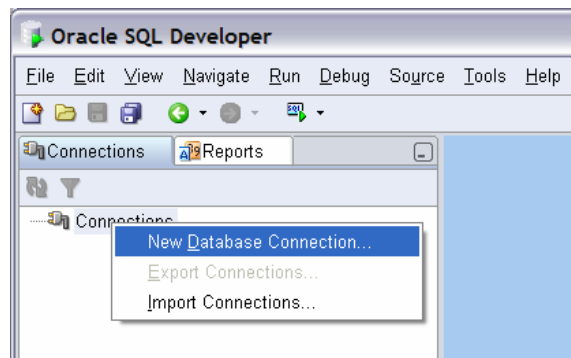
1 Hands On - Getting Started

1.1 Invoke SQL Developer

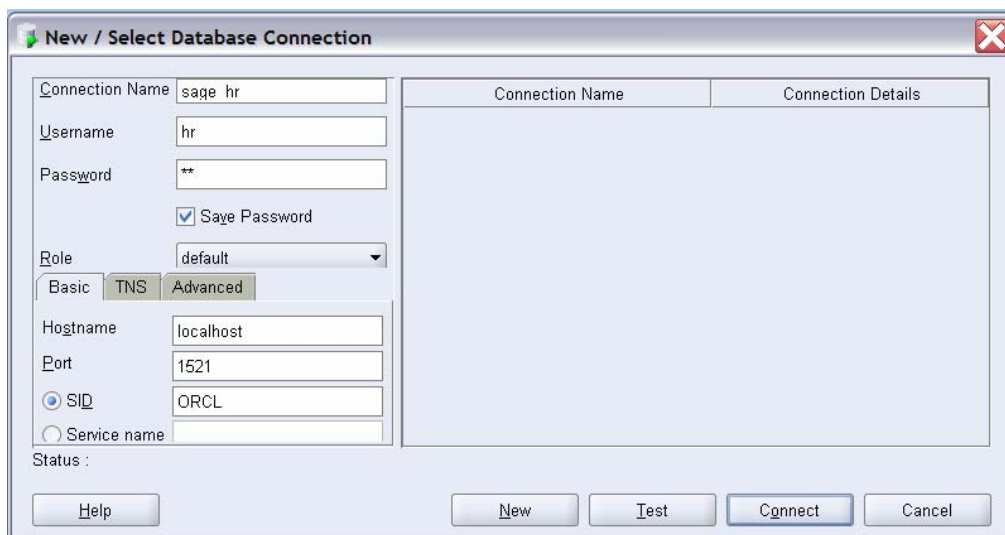
Display Windows Explorer and invoke SQL Developer by double clicking on SQLDeveloper.exe in the directory in which you unzipped the product.

1.2 Create a Database Connection

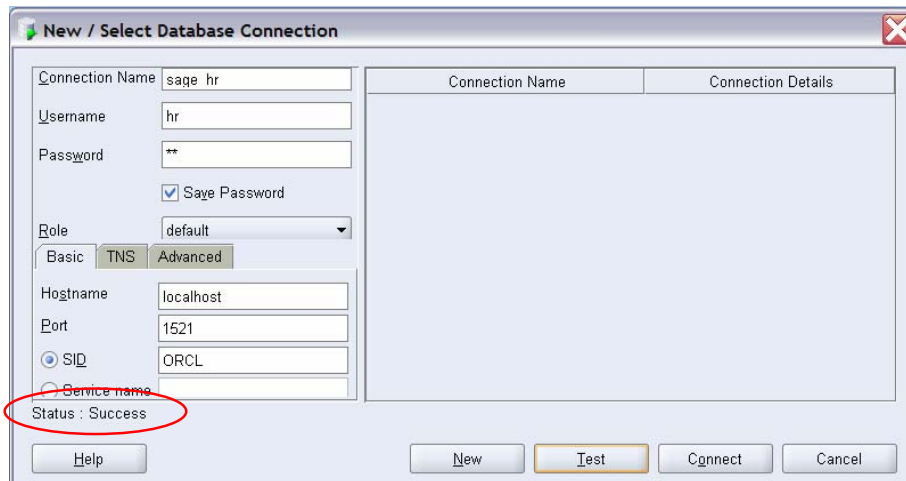
1. Right click on Connections within the Connections tab and Select New Database Connection from the menu.



2. In the New/Select Database Connection Window enter the following information. Enter *sage_hr* as the connection name, *hr* for the username and *hr* for the password. *hr* is the default password for the hr user but may be different for your instance of the hr schema. Enter your *machine name* for the Hostname and the Port and SID in the basic tab as in the example below.

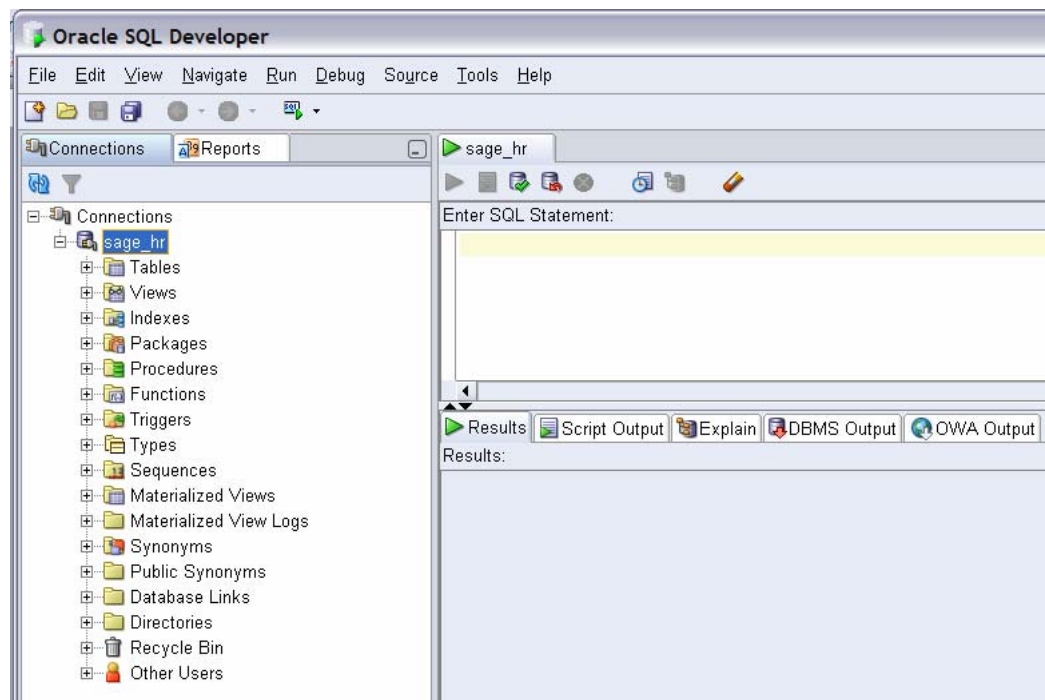


Click the Test button.



The Status above the Help button should show Success. Once you have a successful test click the Connect button to save the new connection.

3. You should now see the *sage_hr* connection in the Connections tab on the left of the SQL Developer application. Expand the *sage_hr* connection, by clicking on the plus sign to its left, to display the list of database object types within your hr schema.



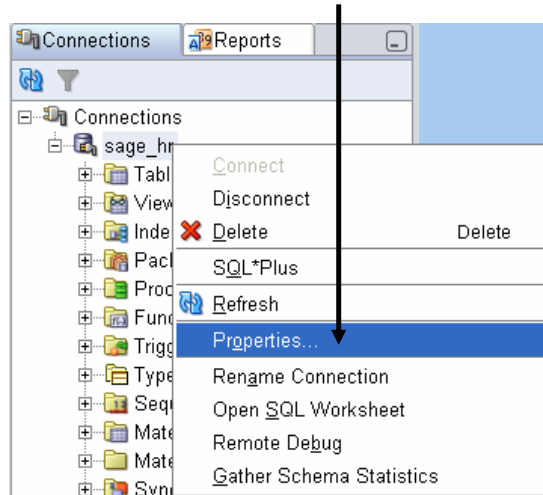
When a connection is opened a SQL Worksheet is automatically displayed for the connection. A SQL Worksheet allows you to execute SQL against the database connection you have just opened. You can also open a new SQL Worksheet by clicking Tools and then SQL Worksheet.

4. SQL Developer allows you to perform a series of operations on a database connection including modifying and renaming a connection and invoking a SQL*Plus session. These options can be accessed by right clicking on the connection and selecting the required operation.

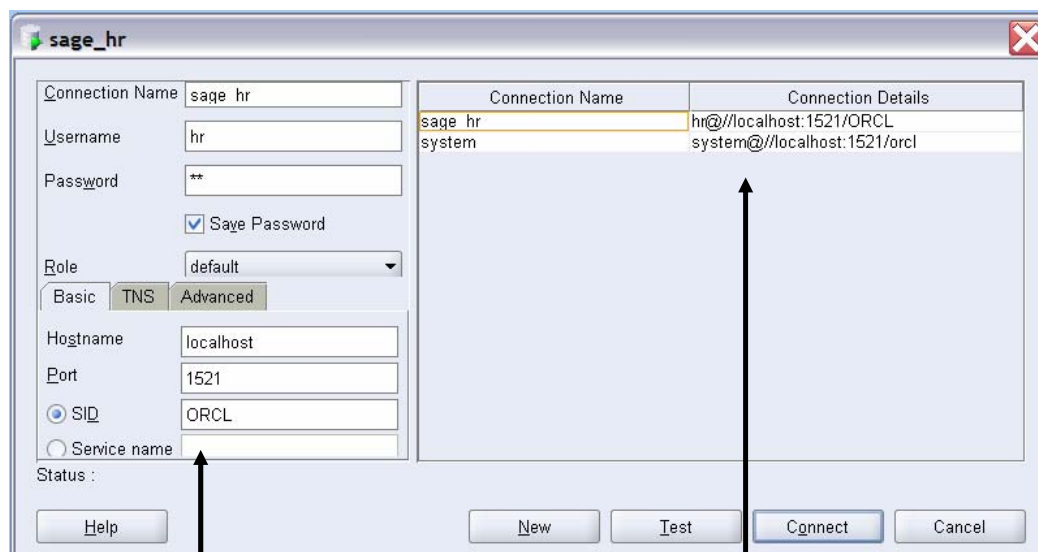
More on Connections

Modifying Database Connections

The properties of a Database Connection can easily be updated by right clicking on the appropriate connection and selecting Properties from the context menu.



The properties of all connections can be managed through this dialog box and details such as Username, Password, Hostname and SID can be modified.



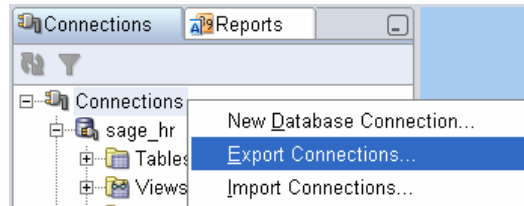
Connection Details

List of connections
defined in SQL Developer

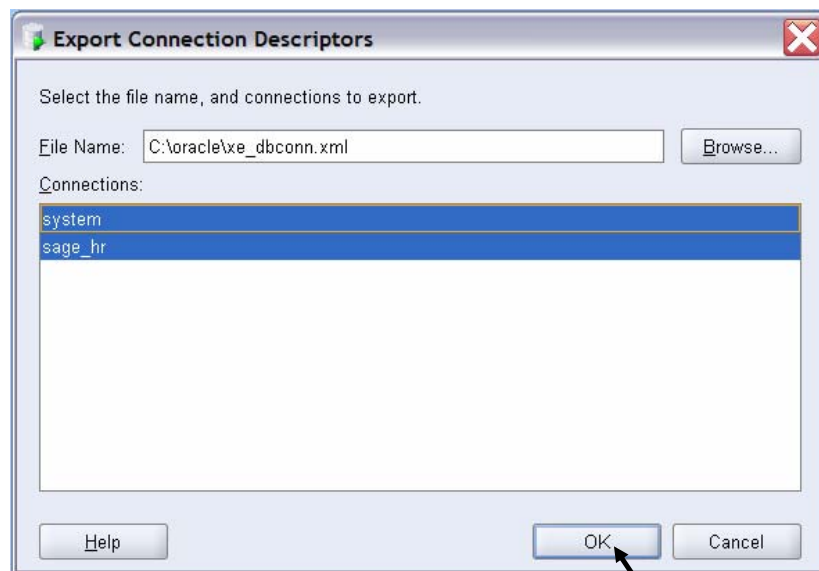
The new connection can be tested via the Test button, then saved by clicking the Connect button.

Exporting and Importing Connections

The Export and Import features within SQL Developer make it easy to store a central definition of Database connections which can then be imported when new SQL Developer applications are installed, and manage changes centrally. SQL Developer allows you to Export and Import Connections to and from an XML file by right clicking on the Connections node in the Connections Navigator.



The Export Connection Descriptors dialog allows you to specify which connections you wish to export, as well as the location and the name of the XML file.

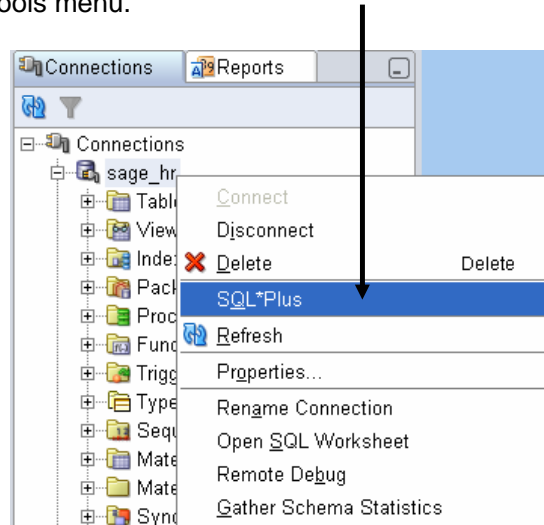


Click OK to complete the export and receive a confirmation message.

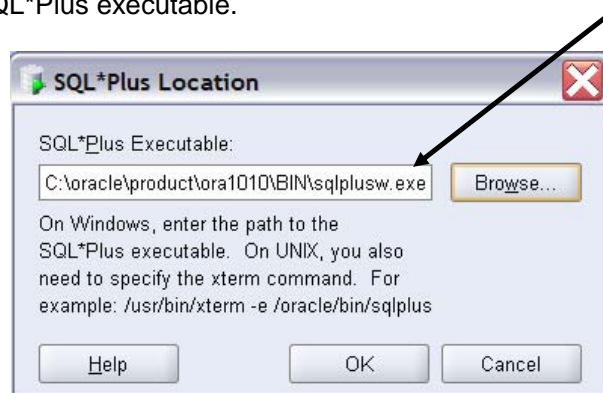
In the same way, first invoke the Import Connection Descriptors from the context menu, then specify the import file and select the connections from that file to import into SQL Developer.

Invoking SQL*Plus from SQL Developer

A SQL*Plus session for a specified connection within SQL Developer can be opened by right clicking the connection name or selecting SQL*Plus from the right mouse menu or from the Tools menu.



The first time this is completed a dialog will be displayed asking you to specify the location of the SQL*Plus executable.



Section 2 – Database Objects

The Connections Navigator in the left hand pane of the product displays all the common database objects under each connection. Expanding the object type node will display a list of the objects of that type in the schema. When you click on an object name, details of the object are displayed in the right hand pane via a series of tab pages. Using the right mouse menu while your cursor is on an object displays a list of actions you can perform on the object.

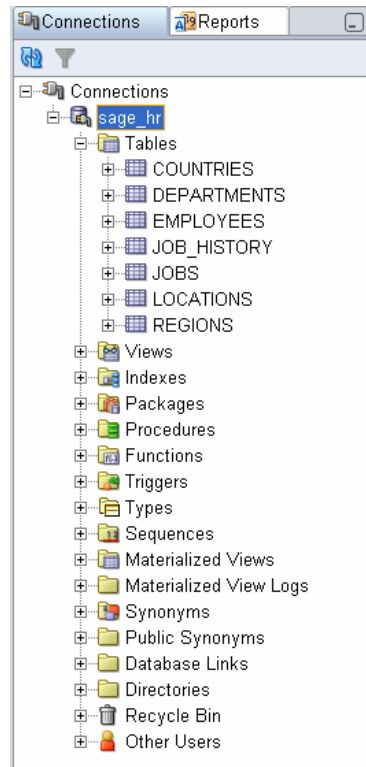


2 Hands On - Database Objects

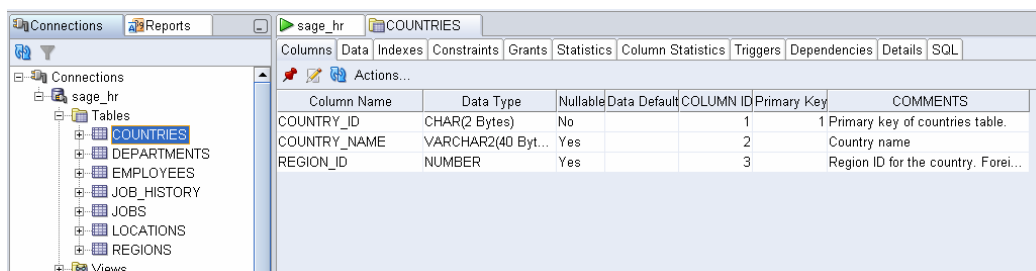
SQL Developer, through the connections tab, allows users to browse, create and edit database objects.

2.1 Browsing of Database Objects

1. When you expand the *sage_hr* database connection, you will see a list of all the possible objects types in your database connection. Expand the Tables node and you will see a list of all the tables available in your connection.

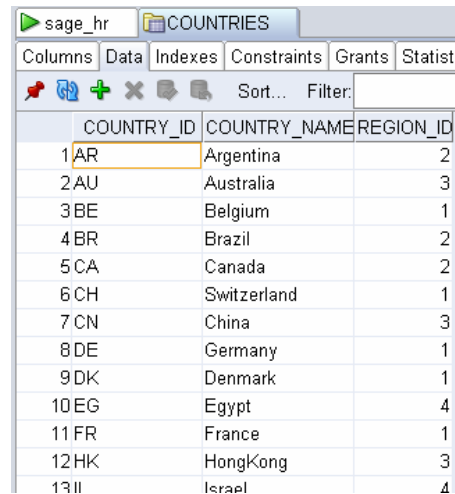


2. Select the COUNTRIES table from the list and you will see a new Tab appear next to your SQL Worksheet, showing you the table definition for the COUNTRIES table.



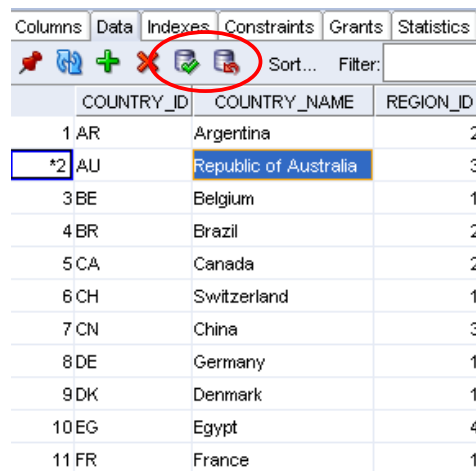
There are a series of Tabs showing Data, Indexes, Constraints and other information related to the selected table.

3. Select the Data tab to view all rows of data stored in the selected COUNTRIES table.



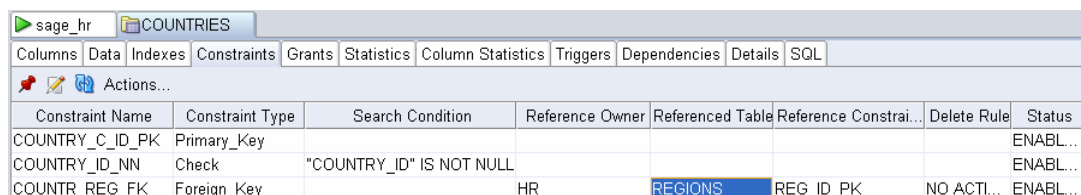
	COUNTRY_ID	COUNTRY_NAME	REGION_ID
1	AR	Argentina	2
2	AU	Australia	3
3	BE	Belgium	1
4	BR	Brazil	2
5	CA	Canada	2
6	CH	Switzerland	1
7	CN	China	3
8	DE	Germany	1
9	DK	Denmark	1
10	EG	Egypt	4
11	FR	France	1
12	HK	HongKong	3
13	IL	Israel	4

4. Click on Australia and change the value to Republic of Australia.



	COUNTRY_ID	COUNTRY_NAME	REGION_ID
1	AR	Argentina	2
*2	AU	Republic of Australia	3
3	BE	Belgium	1
4	BR	Brazil	2
5	CA	Canada	2
6	CH	Switzerland	1
7	CN	China	3
8	DE	Germany	1
9	DK	Denmark	1
10	EG	Egypt	4
11	FR	France	1

5. You can now click on the Commit Changes toolbar button to save your changes, or the Rollback Changes toolbar button to discard them. Discard (Rollback) your changes.
6. Select the Constraints tab to view all constraints for this table and their details.



Constraint Name	Constraint Type	Search Condition	Reference Owner	Referenced Table	Reference Constrai...	Delete Rule	Status
COUNTRY_C_ID_PK	Primary_Key						ENABL...
COUNTRY_ID_NN	Check	"COUNTRY_ID" IS NOT NULL					ENABL...
COUNTR_REG_FK	Foreign_Key		HR	REGIONS	REG_ID_PK	NO ACTI...	ENABL...

We can see from the constraints tab that three constraints exist including a Foreign Key constraint showing the Master Detail relations between the REGIONS and COUNTRIES tables.

- SQL Developer also shows us, under the SQL tab, the code required to create the table as it currently exists in the database.

```

REM HR JOBS

CREATE TABLE "HR"."JOBS"
(
  "JOB_ID" VARCHAR2(10 BYTE),
  "JOB_TITLE" VARCHAR2(35 BYTE) CONSTRAINT "JOB_TITLE_NN" NOT NULL ENABLE,
  "MIN_SALARY" NUMBER(6,0),
  "MAX_SALARY" NUMBER(6,0),
  CONSTRAINT "JOB_ID_PK" PRIMARY KEY ("JOB_ID") ENABLE
);

REM HR JOBS

COMMENT ON COLUMN "HR"."JOBS"."JOB_ID" IS 'Primary key of jobs table.';

COMMENT ON COLUMN "HR"."JOBS"."JOB_TITLE" IS 'A not null column that shows job title, e.g. AD_VP, FI_ACCOUNTANT';

COMMENT ON COLUMN "HR"."JOBS"."MIN_SALARY" IS 'Minimum salary for a job title.';

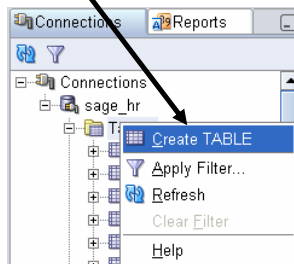
COMMENT ON COLUMN "HR"."JOBS"."MAX_SALARY" IS 'Maximum salary for a job title.';

COMMENT ON TABLE "HR"."JOBS" IS 'jobs table with job titles and salary ranges. Contains 19 rows.
References with employees and job_history table.';

```

2.2 Creating Database Objects

Database Objects can be created by right clicking on the type of object you wish to create and then selecting Create from the context menu. They can also be created in the traditional way by entering and executing the appropriate SQL statement in the SQL Worksheet.



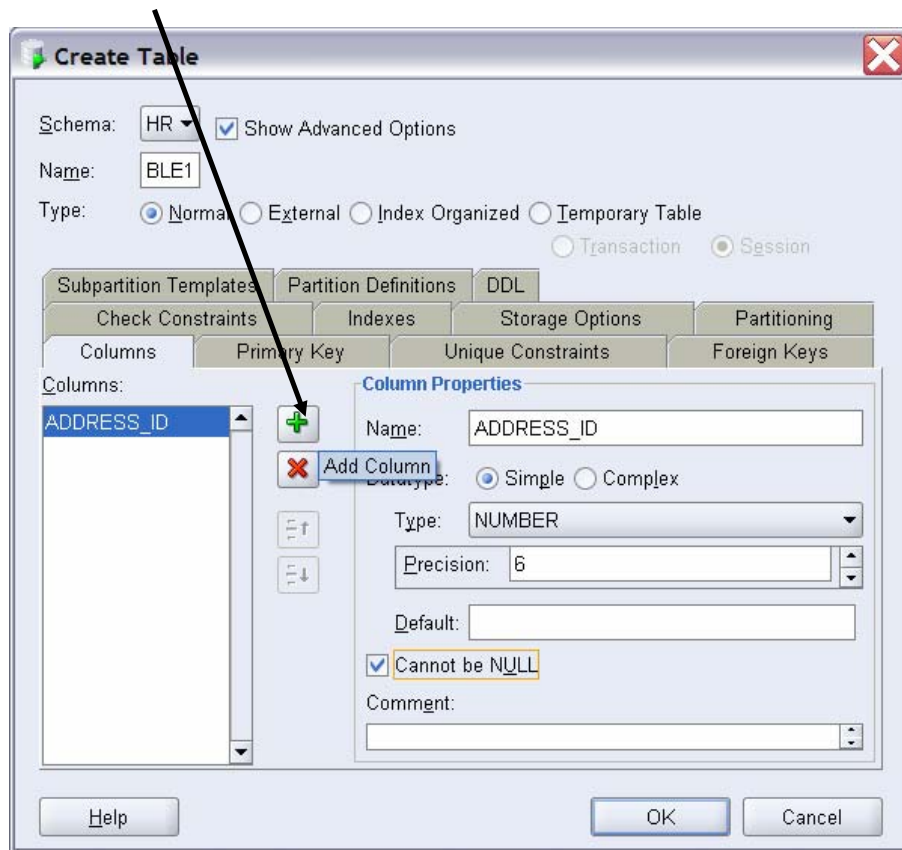
Creating a table using the Table Dialog

- Right click on the Tables node and then click Create TABLE. The Create Table dialog will appear. Enter the table name as ADDRESSES and then tick the Show Advanced Options checkbox. The dialog will change to allow you to define advanced details.

The 'Create Table' dialog box is shown with the following details:

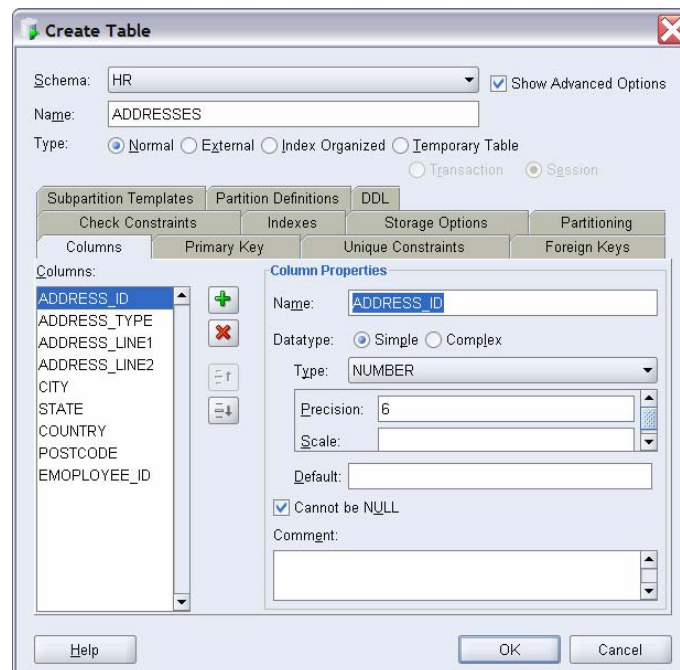
- Schema:** HR
- Name:** ADDRESSES
- Type:** Normal (selected), External, Index Organized, Temporary Table
- Transaction:** Transaction (selected), Session
- Advanced Options:** Show Advanced Options (checked)
- Columns:** A list containing COLUMN1.
- Column Properties for COLUMN1:**
 - Name:** COLUMN1
 - Datatype:** Simple (selected), Complex
 - Type:** VARCHAR2
 - Size:** 4000
 - Default:** (empty)
 - Cannot be NULL:** (unchecked)
 - Comment:** (empty)

2. Enter a name of ADDRESS_ID, in the Type pop-list select NUMBER, followed by a Precision of 6 and then tick the Cannot be NULL checkbox. When the column details are defined click the Add Column icon.

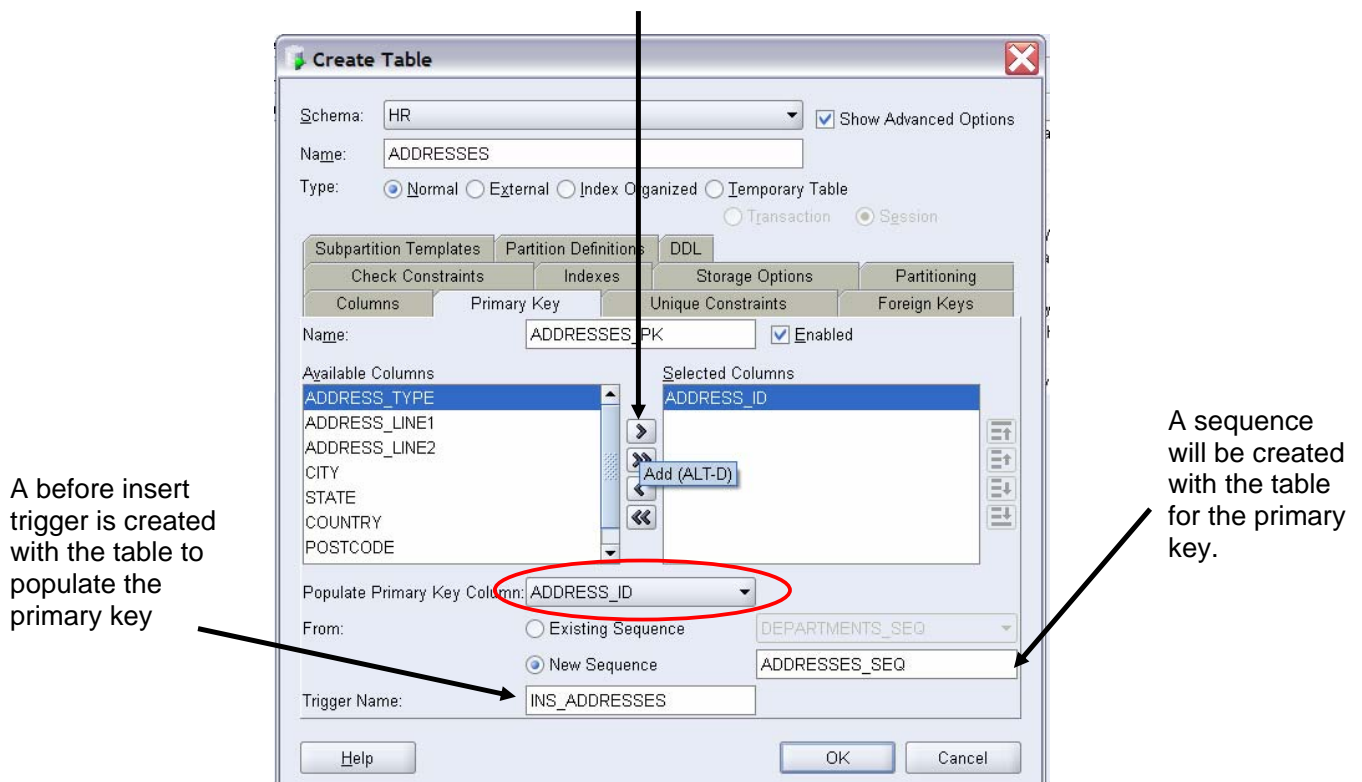


3. Add the following columns using the same process with the below details.

Column Name	Type	Precision	NULL
ADDRESS_ID	NUMBER	6	No
ADDRESS_TYPE	VARCHAR2	20	No
ADDRESS_LINE1	VARCHAR2	100	
ADDRESS_LINE2	VARCHAR2	100	
CITY	VARCHAR2	50	
STATE	VARCHAR2	50	
COUNTRY	VARCHAR2	50	
POSTCODE	VARCHAR2	20	
EMPLOYEE_ID	NUMBER	6	No



- Click on the Primary Key tab and then select ADDRESS_ID as the primary key column.



- Then select ADDRESS_ID from the Populate Primary Key Column list and New Sequence as the method. Click on the Foreign Keys tab.

6. Click Add in the Foreign Keys tab.

Create Table

Schema: **HR** ☒ Show Advanced Options

Name: **ADDRESSES**

Type: ☒ Normal ☐ External ☐ Index Organized ☐ Temporary Table

☐ Transaction ☒ Session

Subpartition Templates Partition Definitions DDL

Check Constraints Indexes Storage Options Partitioning

Columns Primary Key Unique Constraints Foreign Keys

Foreign Keys:

ADDRESSES_FK1 **Add** **Remove**

Name: **ADDRESSES_FK1** ☒ Enabled

Referenced Schema: **HR**

Referenced Table: **ADDRESSES**

Referenced Constraint: **ADDRESSES_PK**

Associations:

Local Column	Referenced Column on ADDRESSES
ADDRESS_ID	ADDRESS_ID

On Delete: **RESTRICT**

Help **OK** **Cancel**

7. Change the Reference Table to EMPLOYEES, the referenced constraint should change to EMP_EMP_ID_PK. Then change the local column to EMPLOYEE_ID.

Create Table

Schema: **HR** ☒ Show Advanced Options

Name: **ADDRESSES**

Type: ☒ Normal ☐ External ☐ Index Organized ☐ Temporary Table

☐ Transaction ☒ Session

Subpartition Templates Partition Definitions DDL

Check Constraints Indexes Storage Options Partitioning

Columns Primary Key Unique Constraints Foreign Keys

Foreign Keys:

ADDRESSES_FK1 **Add** **Remove**

Name: **ADDRESSES_FK1** ☒ Enabled

Referenced Schema: **HR**

Referenced Table: **EMPLOYEES**

Referenced Constraint: **EMP_EMP_ID_PK**

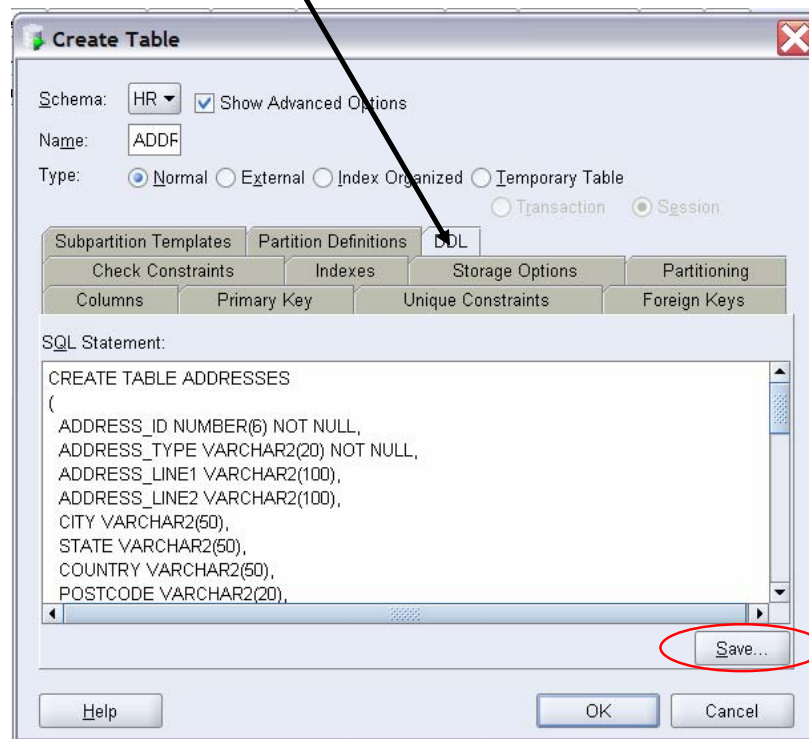
Associations:

Local Column	Referenced Column on EMPLOYEES
EMPLOYEE_ID	EMPLOYEE_ID

On Delete: **RESTRICT**

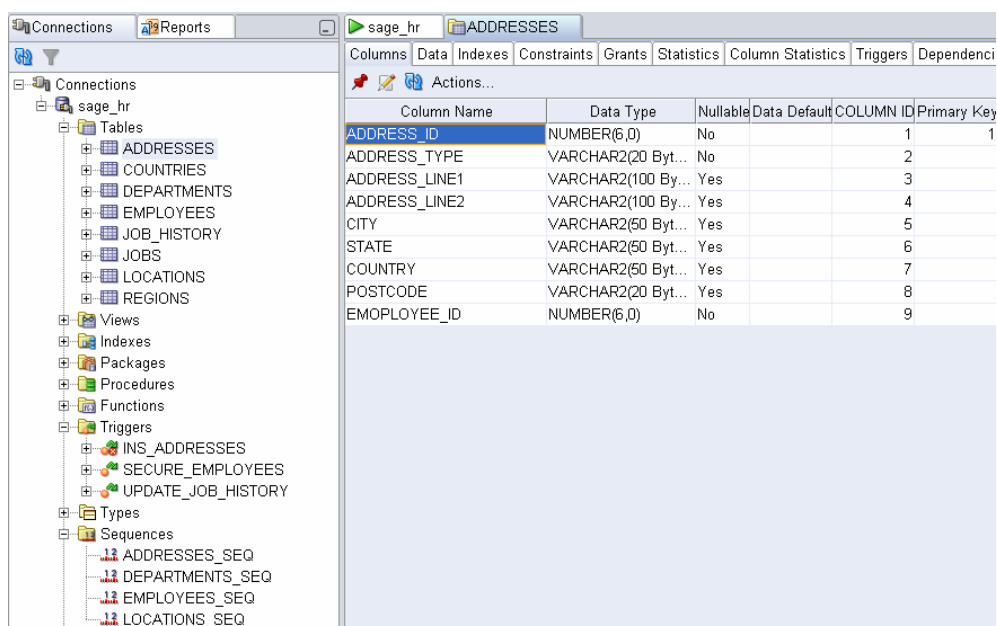
Help **OK** **Cancel**

8. Before we create the ADDRESS table we will first review the SQL that has been generated by the Create Table dialog. Click on the DDL tab to display the SQL that was generated.



You can also save the script to a file before executing.

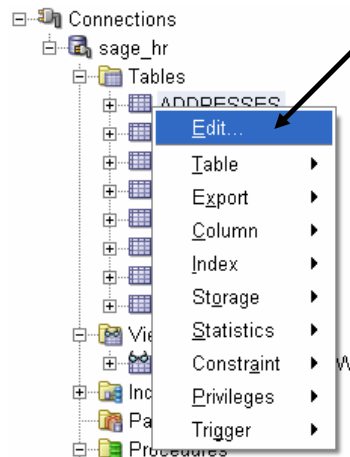
In the DDL tab you can see the SQL script that will create the table, primary key constraint, foreign key constraint, sequence and trigger to populate the primary key. Review this code before clicking OK to create the table. Review the Connections Navigator and check that the ADDRESSES tables, sequence and trigger have all been created correctly.



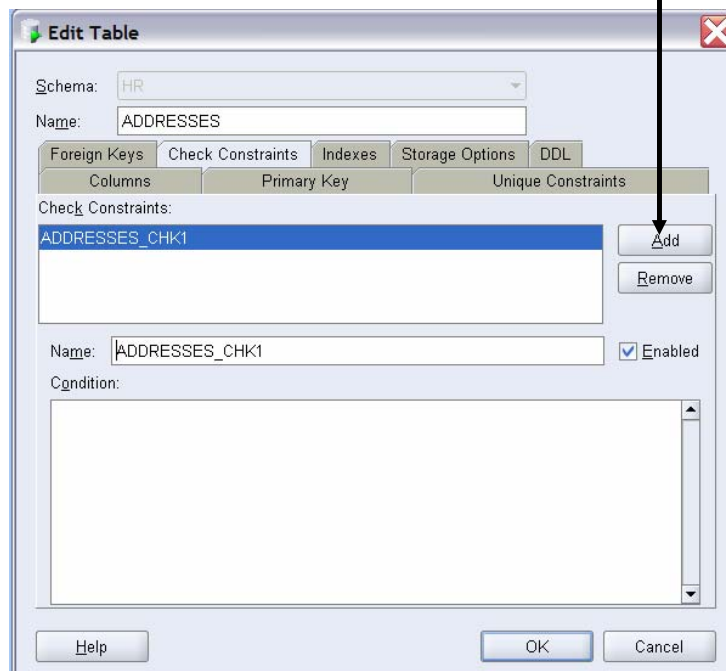
2.3 Modifying Database Objects

Database Objects can be modified by using SQL in the SQL Worksheet or by utilising the provided dialog boxes and tools within SQL Developer. To invoke the Edit dialog box for a particular object, simply right click on the object and select Edit from the context menu. Note that it is not possible to edit packages, functions or producers in this way. Next we will edit the ADDRESSES table using the Edit dialog box.

1. Right click on the ADDRESSES table in the Connections manager and select Edit from the context menu to invoke the Edit dialog box.



2. The Edit Table dialog will appear. You will see that it is in the same format and layout as the Create table dialog. We will now create a simple check constraint to restrict the ADDRESS TYPE column. Click the Check Constraints tab and click add.



3. Change the name to ADDRESS_TYPE_CHK and enter the following condition.

```
address_type IN ('HOME','POSTAL')
```

The screenshot shows the 'Edit Table' dialog box for the 'ADDRESSES' table in the 'HR' schema. The 'Check Constraints' tab is active. A list of constraints shows 'ADDRESSES_CHK1' selected. Below the list, the 'Name' field contains 'ADDRESS_TYPE_CHK' and the 'Condition' field contains 'address_type IN ('HOME','POSTAL')'. The 'Enabled' checkbox is checked. Buttons for 'Add', 'Remove', 'Help', 'OK', and 'Cancel' are visible.

4. Click on the DDL tab to review the statement that will be executed to modify the table.

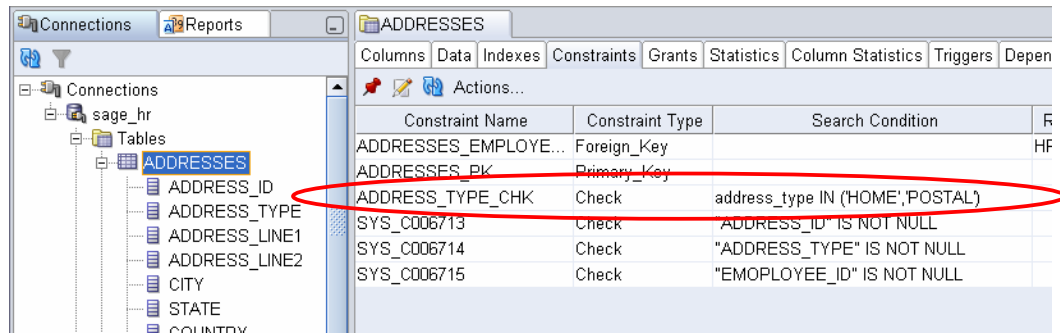
The screenshot shows the 'Edit Table' dialog box with the 'DDL' tab selected. The 'SQL Statement' field contains the following text:

```
ALTER TABLE ADDRESSES  
ADD CONSTRAINT ADDRESS_TYPE_CHK CHECK  
(  
address_type IN ('HOME','POSTAL')  
) ENABLE  
;
```


Buttons for 'Save...', 'Help', 'OK', and 'Cancel' are visible at the bottom.

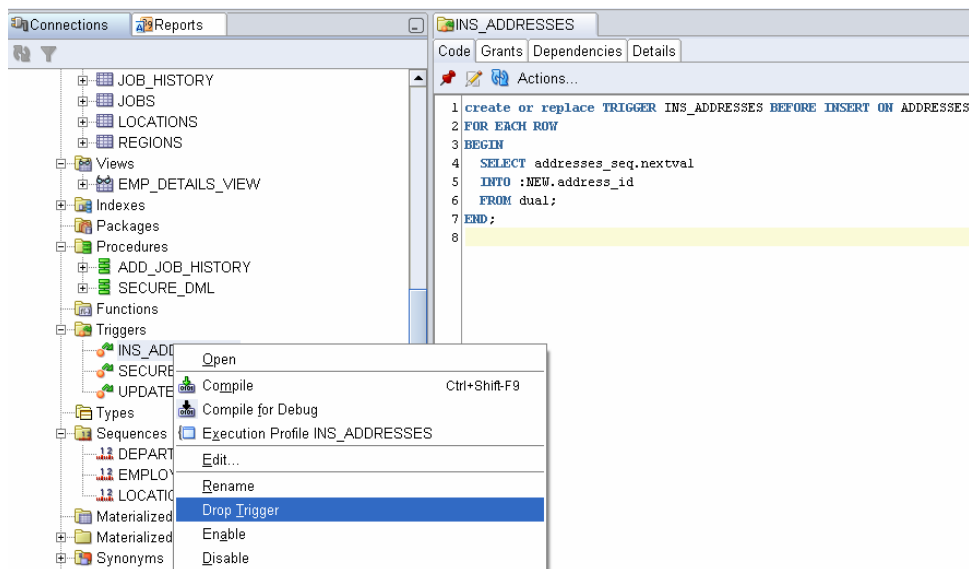
5. Click OK.

Now go to the ADDRESSES table browser and check the Constraints tab to ensure the constraint has been created correctly.

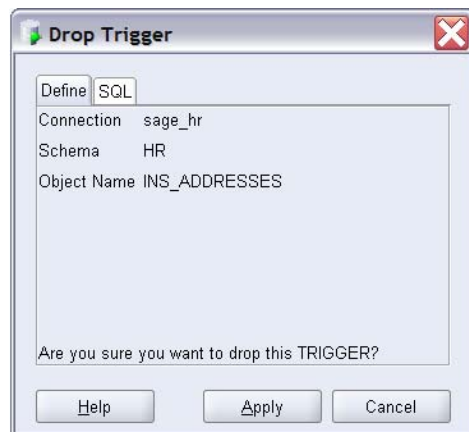


2.4 Deleting Database Objects

To delete or drop a database object, simply right click on the object in the Connections Navigator and select Drop from the context menu.



A dialog box will appear to allow you to review the SQL to be performed and confirm that you wish to continue. Click Apply to execute the drop statement. Note the drop feature for a table is located under the Table menu of the context menu.



Section 3 – SQL Queries

The SQL Language

The SQL language allows you to view and change information in a database. In this workshop we will only cover queries.

Structure

- SQL command words (Select, Insert, etc.) cannot be written over more than one line; however the entire statement may span one or more lines.
- For readability and maintainability it is suggested that individual clauses are placed on separate lines.
- SQL commands are not case sensitive. Any character data which you reference in a command is case sensitive. (If its in quotes its case sensitive).
- If you are entering multiple SQL statements in the SQL Developer worksheet you should terminate each statement with a semicolon.

The SELECT Statement

Selecting All Rows

The following statement will select the contents of three columns for all of the rows within the table:

```
SELECT    column1, column2, column3
FROM      tablename;
```

Example:

```
SELECT    first_name, last_name, salary
FROM      employees;
```

To select all of the columns in a table, an * can be specified.

Example:

```
SELECT    *
FROM      jobs;
```

Expressions

In addition to viewing column data, columns can be combined with other columns and literals into an expression. The following standard arithmetic operators can be used to achieve this:

+	add
-	minus
*	multiply
/	divide

Example:

In this example the job_title is being selected. However, as part of this select statement, we are also working out the difference between maximum and minimum salary.

```
SELECT    job_title, max_salary – min_salary
FROM      jobs;
```

Column Alias

A column alias is a method of giving an item in the select list a different name in the returned data. The alias will be used as the column header instead of the column name. This is particularly useful when selecting expressions.

The alias follows the column or expression. Additionally, the word as can be included between the expression and alias.

Examples:

Example 1:

```
SELECT    job_title, max_salary – min_salary AS salrange
FROM      jobs;
```

Example 2:

```
SELECT    first_name, last_name, salary*12 annualsal
FROM      employees;
```

Restricting the Rows

The rows returned by the select statement can be restricted by adding a WHERE clause.

```
SELECT    column1, column2, column3
FROM      tablename
WHERE     boolean expression;
```

If the boolean expression evaluates to TRUE the row will be included in the results set.

If the boolean expression evaluates to FALSE or UNKNOWN the row will not be included in the results set.

The boolean expression can involve any of the columns in the table as well as literals.

A column can be compared with another column or expression.

Equality comparisons or ranges can be used (<, >, <=, >=).

Multiple clauses using AND or OR can be used. If AND and OR are combined then the AND conditions will be evaluated first.

Note that a column alias cannot be referenced in the WHERE clause.

Examples:**Example 1:**

```
SELECT    street_address, postal_code, city
FROM      locations
WHERE     state_province = 'New South Wales';
```

Example 2:

```
SELECT    street_address, postal_code, city
FROM      locations
WHERE     country_id = 'UK'
AND       state_province = 'Oxford';
```

Example 3:

```
SELECT    street_address, postal_code, city
FROM      locations
WHERE     country_id = 'UK'
OR        country_id = 'AU';
```

Example 3:

```
SELECT    first_name, last_name, salary
FROM      employees
WHERE     salary*12 >100000
```

Ordering the Data

The order in which rows are returned can be defined by adding an ORDER BY clause. By default the rows will be in ascending order of the specified column. Add the keyword DESC to return rows in descending order of the columns or expression specified in the ORDER BY clause.

Columns in the ORDER BY clause do not have to be included in the SELECT list.

Examples:**Example 1:**

```
SELECT    country_id, street_address, postal_code, city
FROM      locations
ORDER BY  country_id;
```

Example 2:

```
SELECT    first_name, last_name, salary
FROM      employees
ORDER BY  salary, hire_date DESC
```

Group Functions

Group functions work on sets of rows and return a single value for the set.

SUM

To select the sum of values in a set of columns use SUM(numcolumn)

Example

Select the total cost for all bookings:

```
SELECT    SUM(salary)
FROM      employees;
```

COUNT

To select the number of non empty (NOT NULL) columns in a set of columns use COUNT(column)

Example

Select the number of employees that have a commission

```
SELECT    COUNT(commission_pct)
FROM      employees;
```

Grouping the Output into Sets of Rows

Group functions work on sets of rows and return a single value for each set. Rather than grouping all the rows into a single set, the results can be divided into a number of sets using the GROUP BY clause

Example

Select the total salary for all employees in each department

```
SELECT    department_id, SUM(salary)
FROM      employees
GROUP BY  department_id;
```

Joins

So far we have selected information from one table only in each statement. In the following section we will select from multiple tables using join conditions.

Foreign keys

Tables are usually joined where a primary key / foreign key relationship exists between them.

Example:

```
COUNTRIES(country_id, country_name, region_id)
```

```
LOCATIONS(location_id, street_address, postal_code, city, state_province,  
country_id)
```

In order to display the country name for a location, rows would be required where the country_id in the LOCATIONS table matches the country_id in the COUNTRIES table.

It is important to understand the structure of the tables from which the selections are being made. The names of the columns do not have to be the same in both tables.

Select Statements with Join Conditions

Example:

```
SELECT      c.country_id, c.country_name, l.street_address, l.city  
FROM        countries c, locations l  
WHERE       c.country_id = l.country_id;
```

N.B. Omitting the join condition will result in a cartesian product resulting in each location row being joined to each country row.

Using a Table Alias

If there are column names that are common to both of the tables, it is necessary to prefix the column with the table name or a table alias. A table alias is a method of referring to a table using a short name.

In the example above the table alias for COUNTRIES is c, the table alias for LOCATIONS is l.



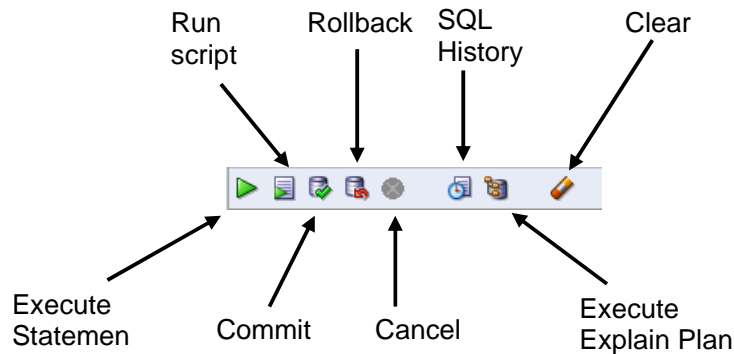
3 Hands On – SQL and the SQL Worksheet

The SQL Worksheet allows you to execute SQL, SQL*Plus and PL/SQL statements and scripts. Upon successful execution, the SQL Worksheet displays the data results, as well as an Explain plan, DBMS output, OWA and script output if necessary. Your SQL Worksheet will display the name of the database connection it is managing in the tab at the top of the worksheet.

You can use the same SQL Worksheet for multiple database connections. If you have multiple database connections open you can use the drop down box in the top right corner of your SQL Worksheet to change between connections using the same Worksheet.


3.1 The SQL Worksheet Toolbar

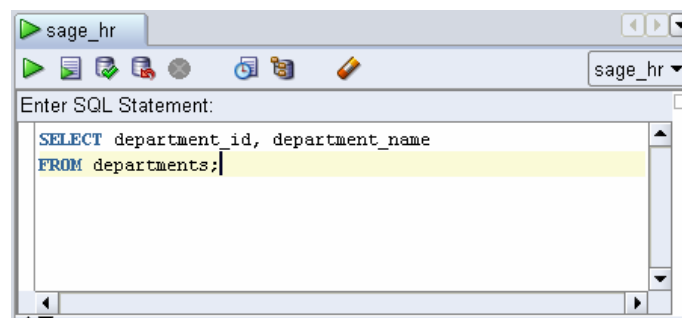
The SQL toolbar is used to perform common functions for statements and scripts in the SQL Worksheet.



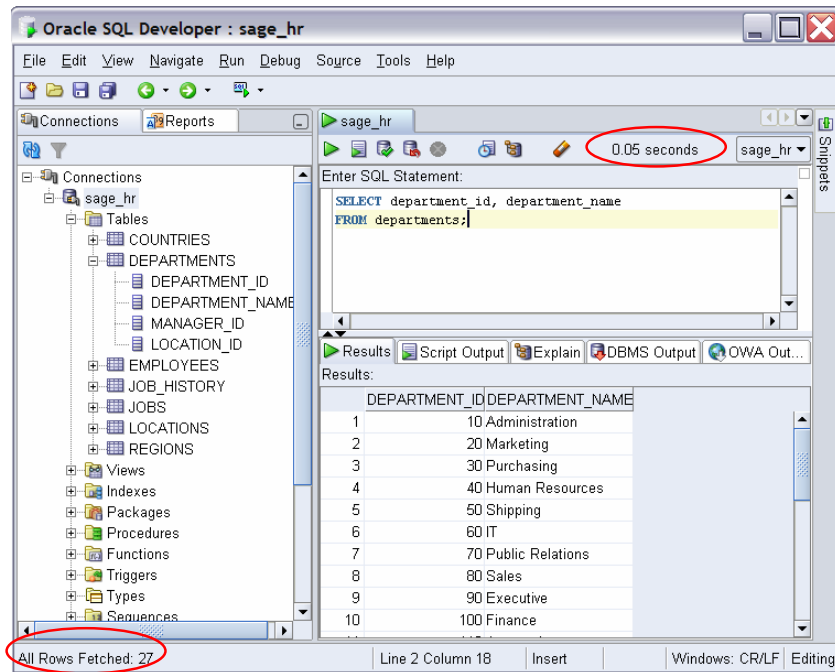
3.2 Using the SQL Worksheet to perform SQL statements

If you do not have a SQL Worksheet open, right click on your *sage_hr* connection and select Open SQL Worksheet or select the SQL Worksheet from the Tools menu. Confirm that your Worksheet is using the *sage_hr* connection by checking it is selected in the drop down list located in the right corner of the window.

1. Enter a Select statement in the SQL area of the Worksheet to retrieve the DEPARTMENT_ID and DEPARTMENT_NAME of all the rows in the DEPARTMENTS table as in the following representation. Then with your cursor on the statement, click the Execute Statement button  or press F9. There are other useful functions on the worksheet toolbar such as options to Run a Script, Commit or Rollback any changes, Cancel, access the SQL History, display the access path Oracle has used using the Explain Plan facility, and Clear the entry area.



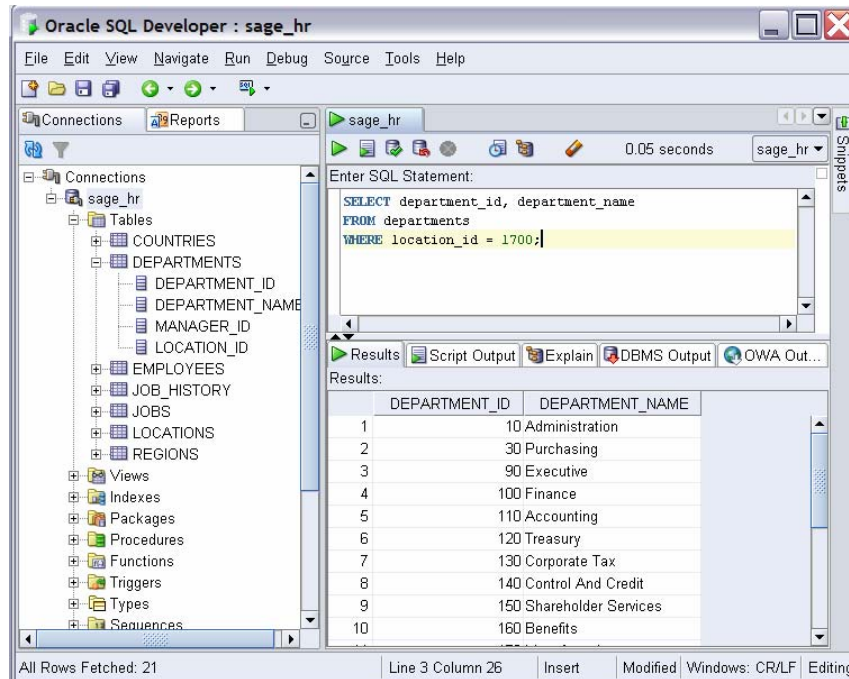
2. With a correct statement, the results should display under the Results tab below. The worksheet also displays the time the statement took to execute in the toolbar. In this case, 0.05 seconds.



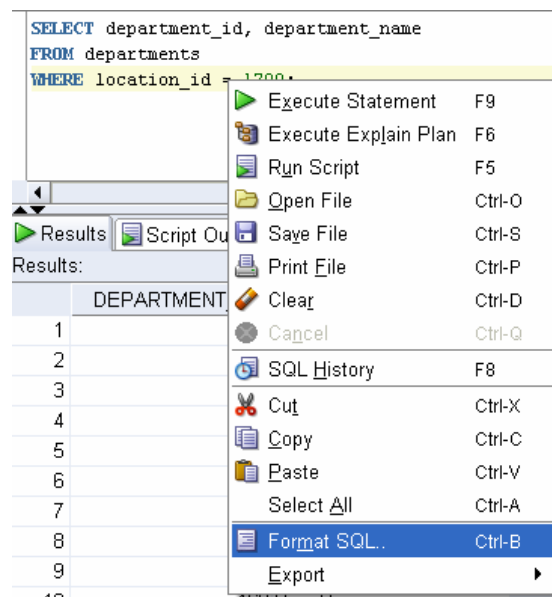
The number of rows returned is also displayed in the status bar. If the status bar is not displayed, you can make it visible by clicking the View menu, then selecting Status Bar.

Note: SQL Developer also allows you to drag and drop columns from the Connections Navigator into the SQL work area instead of typing them in. You can test this drag and drop feature if you wish.

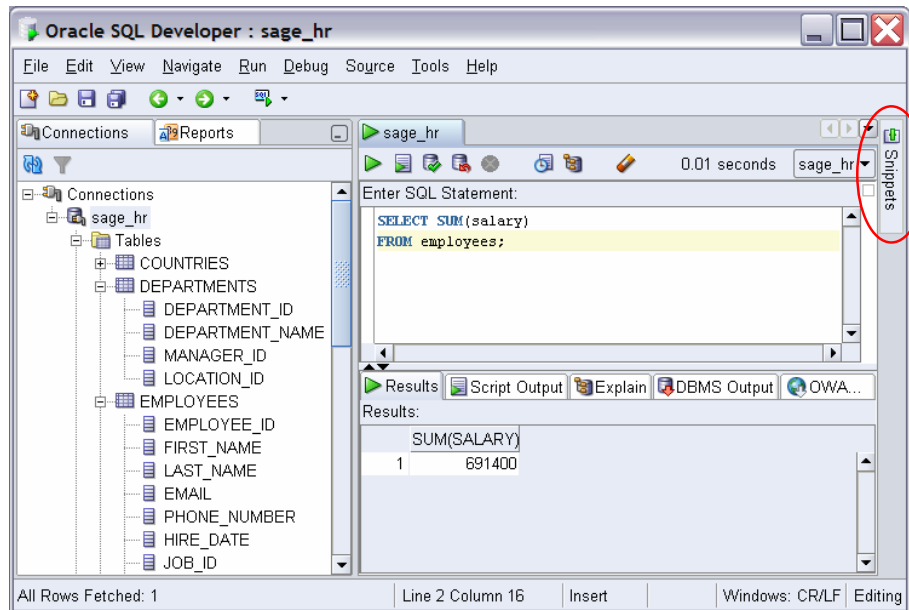
- We now want to restrict the rows being returned from the query and only display departments from a specific location. This is achieved by adding a WHERE clause to the select statement. Add a WHERE clause to the end of this statement to only display DEPARTMENTS with a LOCATION_ID of 1700 and then execute the query.



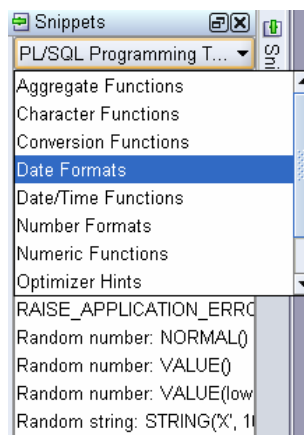
Use the Format SQL feature from the SQL Worksheet's right mouse button context menu to make complex statements easier to read.



- Next we will use an aggregate function to calculate the total salary for all employees. SQL Developer provides a Snippets panel which can be accessed from the View menu or at the right hand side of the application window. Snippets are code fragments you can use by dragging and drop them into your SQL Worksheet. They range from SQL functions and PL/SQL programming techniques to optimizer hints and examples. Enter a SELECT statement into your worksheet for the *sage_hr* connection to calculate the total salary for all employees in the EMPLOYEES table. You can use the Snippets panel to search for an appropriate function.



Take some time to explore and test the other snippets available in SQL Developer.



5. Lastly we want to display the combined salaries of all employees for each department. This query requires an aggregate function as used above, a join between two tables and a GROUP BY clause. We need to use a foreign key join between the employees and departments tables by including a WHERE clause. We then also need to include a GROUP BY clause to display the SUM for each department.

The screenshot shows the Oracle SQL Developer interface with the 'sage_hr' schema selected. The 'DEPARTMENTS' table is highlighted in the left pane. The 'Enter SQL Statement' window contains the following query:


```
SELECT d.department_id, d.department_name, SUM(e.salary)
FROM departments d, employees e
WHERE e.department_id = d.department_id
GROUP BY d.department_id, d.department_name;
```

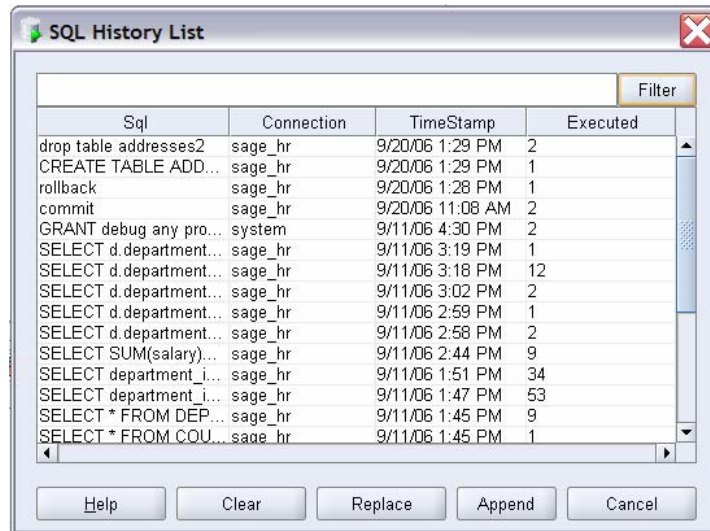
The 'Results' window displays the output of the query, showing the sum of salaries for each department. The results are as follows:

DEPARTMENT_ID	DEPARTMENT_NAME	SUM(E.SALARY)
1	10 Administration	4400
2	20 Marketing	19000
3	30 Purchasing	24900
4	40 Human Resources	6500
5	50 Shipping	156400
6	60 IT	28800
7	70 Public Relations	10000
8	80 Sales	304500
9	90 Executive	58000
10	100 Finance	51600
11	110 Accounting	20300

The status bar at the bottom indicates 'All Rows Fetched: 11', 'Line 4 Column 47', 'Insert' mode, and 'Windows: CR/LF Editing'.

3.3 SQL History List

SQL Developer maintains a list of SQL statements executed in the SQL Worksheet. To display the SQL History press F8 or click the SQL History icon  in the worksheet toolbar. Please note that the SQL History does not store scripts which have been executed, nor does it store statements which contain a password.



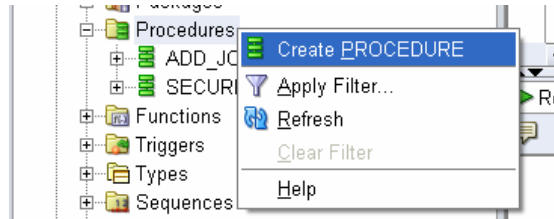
These stored statements can be retrieved at anytime and reused. They can be used to replace the text, or be appended to the current text in the SQL Worksheet. The filter at the top of the dialog box provides a case sensitive search feature for the SQL History. The history can be cleared using the Clear button.



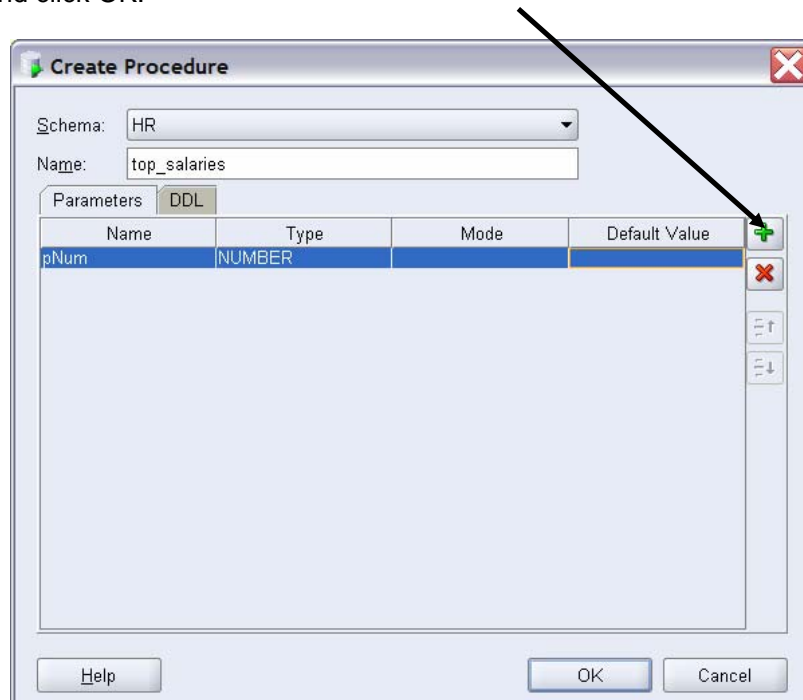
4 Hands On - PL/SQL Development

4.1 Creating and Editing PL/SQL

1. In the connections navigator, expand the Procedures node and right click to display the context menu. Select Create PROCEDURE.



2. The Create Procedure dialog will display. With the HR schema selected, enter top_salaries as the procedure name and add a parameter pNum NUMBER and click OK.



3. The procedure skeleton is created and displayed in a tab window. Replace the BEGIN; NULL; END; code with the following and then click the save icon to compile the procedure.

```
CURSOR c_top_sal IS
  SELECT e.employee_id,
         e.last_name,
         e.salary
  FROM employees e
  ORDER BY salary DESC;
r_top_sal c_top_sal%ROWTYPE;
i NUMBER :=1;
```

```
BEGIN
```

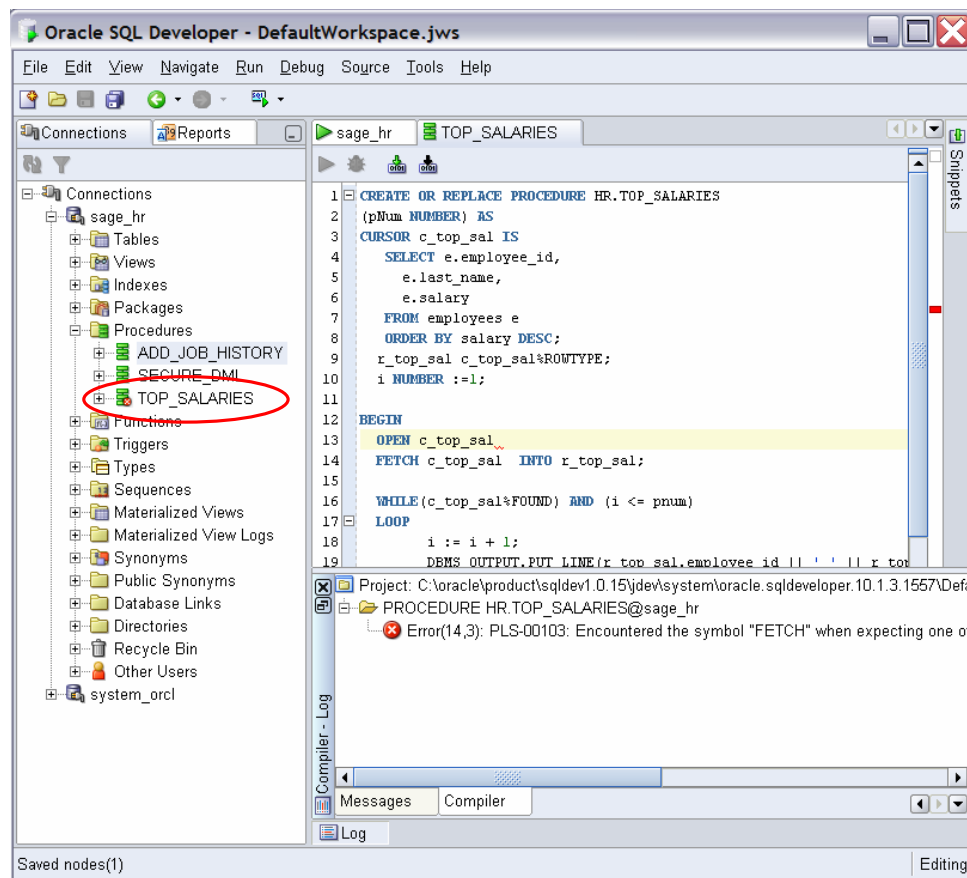
```

OPEN c_top_sal
FETCH c_top_sal INTO r_top_sal;

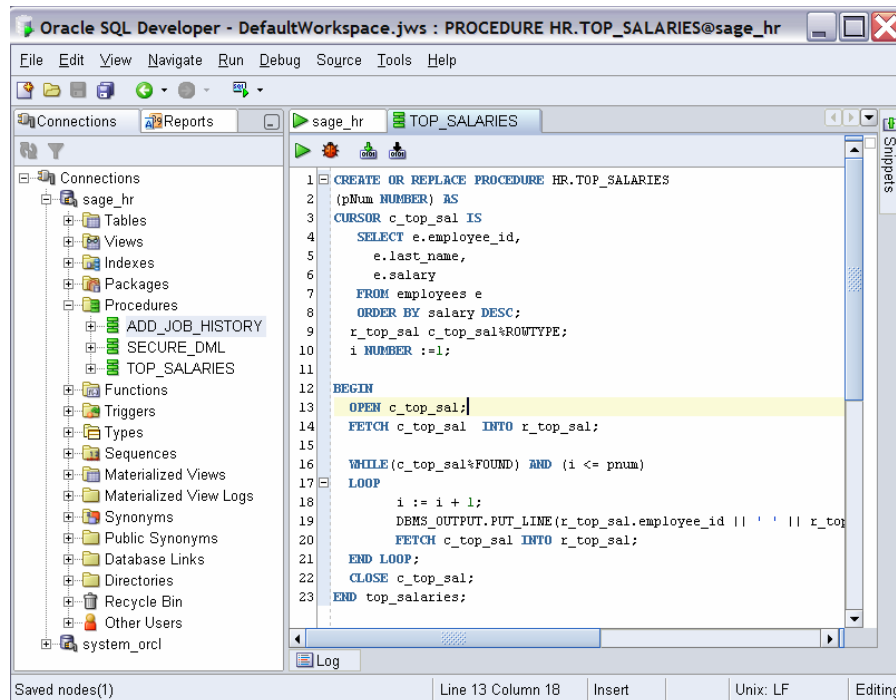
WHILE(c_top_sal%FOUND) AND (i <= pnum)
LOOP
    i := i + 1;
    DBMS_OUTPUT.PUT_LINE(r_top_sal.employee_id
    || ' ' || r_top_sal.last_name || ' '
    || r_top_sal.salary);
    FETCH c_top_sal INTO r_top_sal;
END LOOP;
CLOSE c_top_sal;
END top_salaries;

```

- The Compiler log in the bottom panel shows an error, you will also notice that under the Procedures node in the Connections Navigator, a red cross is displayed against the procedure. Double click on the error in the log to navigate to the error line.

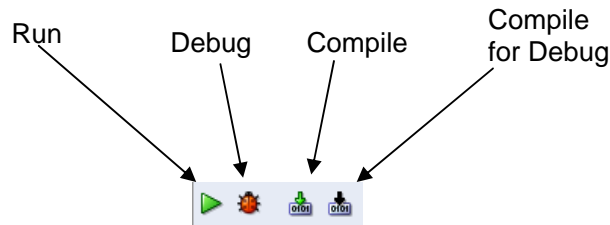


- A semi-colon is missing from the OPEN cursor line. Fix the error and then click the Compile icon. The procedure compiled successfully and is ready to be executed.

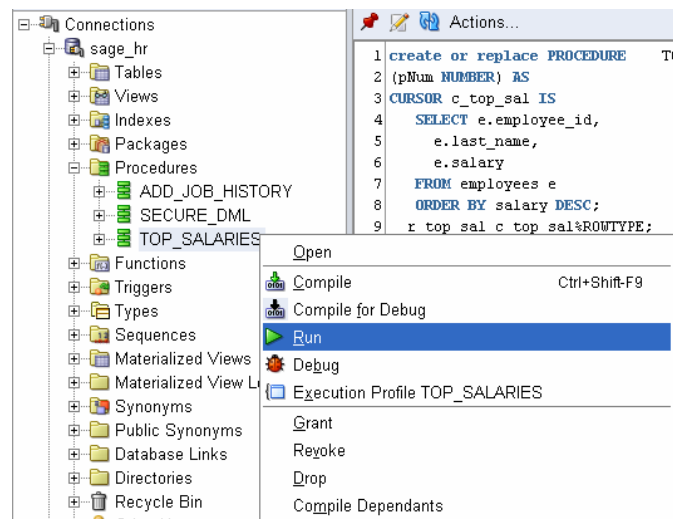


4.2 Execution and Debugging

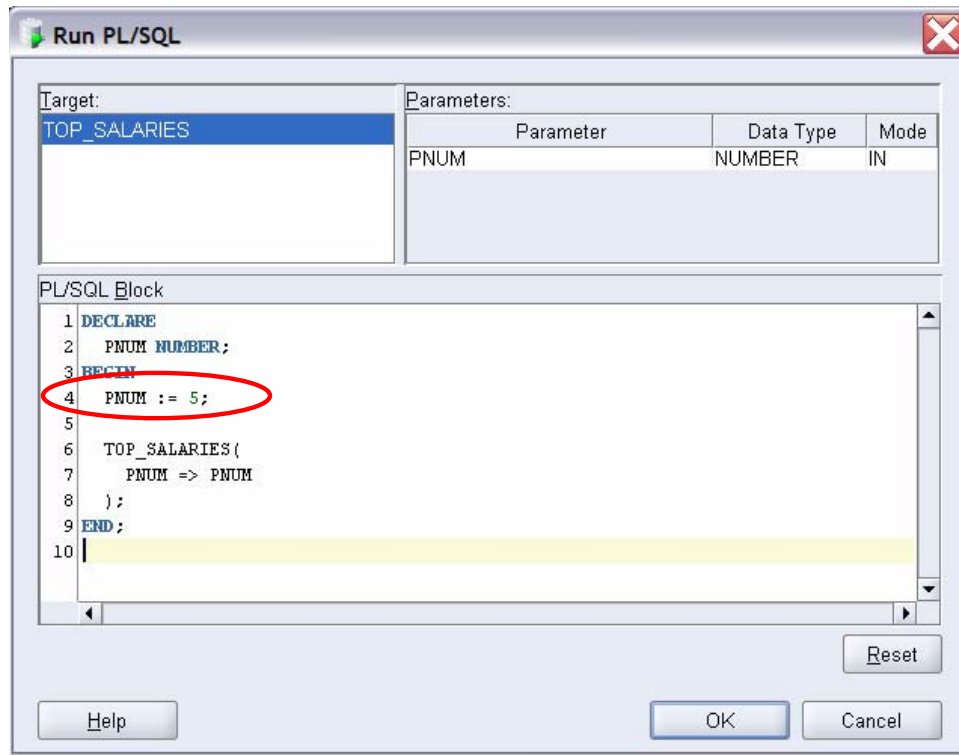
You can execute a procedure by right clicking on the procedure in the Connections Navigator and selecting Run from the context menu. Procedures and Functions can also be run from the toolbar in the Source tab or from the SQL Developer menu.



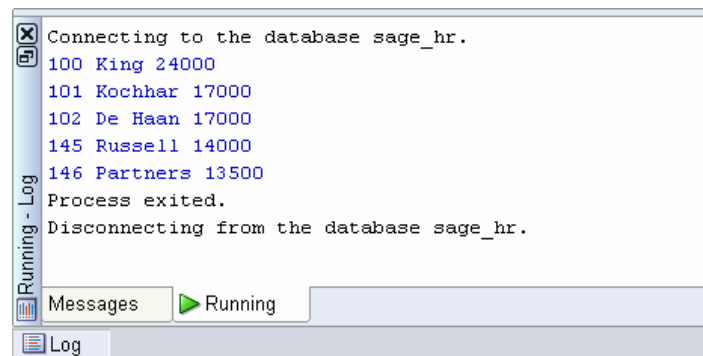
1. From the context menu, run your TOP_SALARIES procedure.



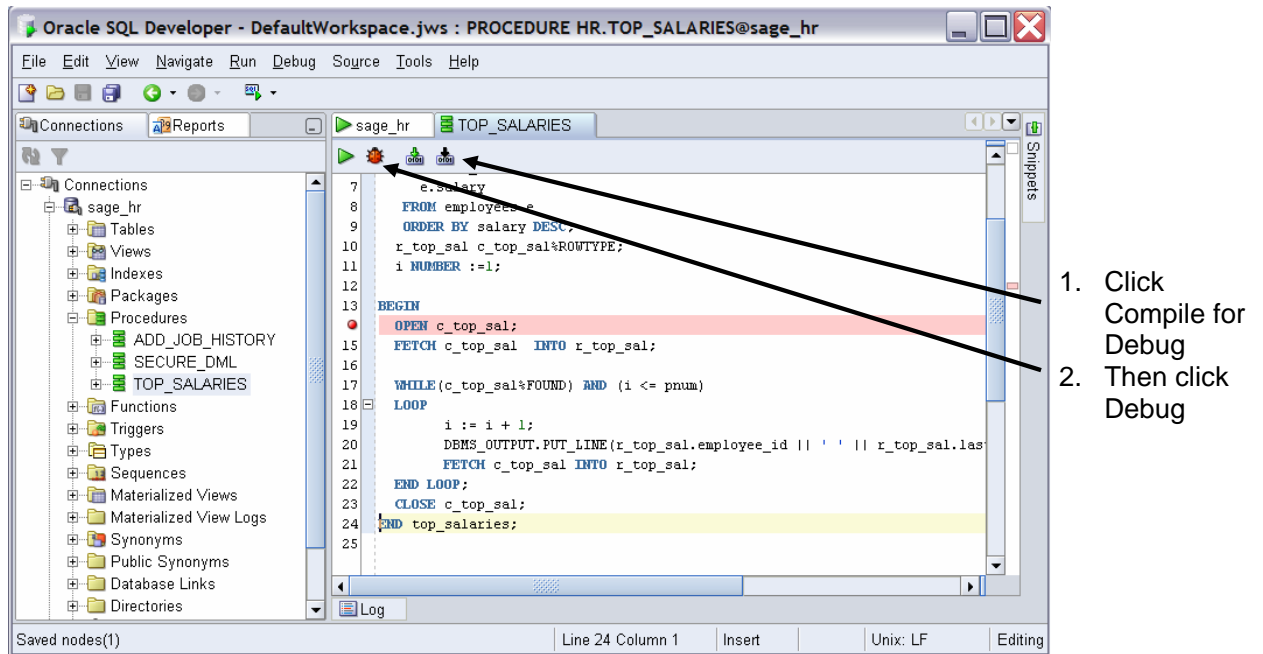
- The run PL/SQL dialog will be displayed. Change the value of the pNum parameter in the execution block to 5. Click OK to close the dialog and execute the procedure.



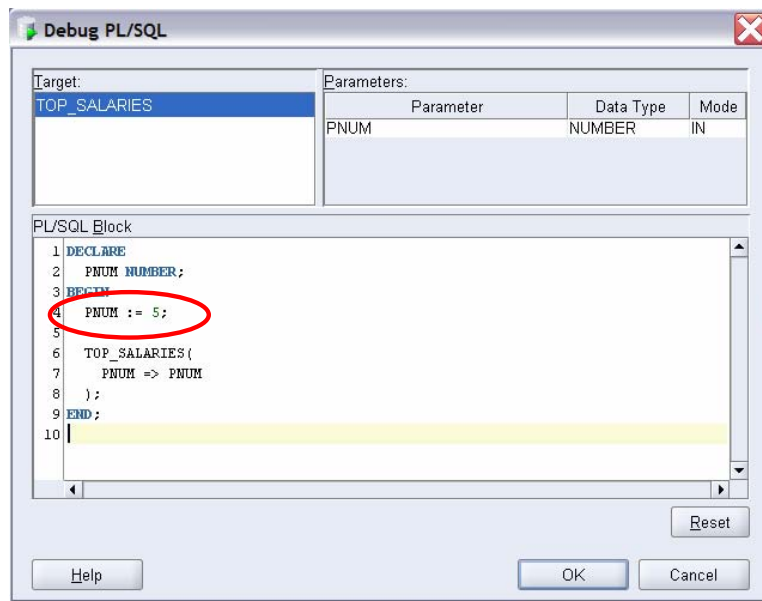
- The Log panel at the bottom of the SQL Developer Window shows the results of the procedure including the 5 top salaries.



- With your TOP_SALARIES procedure open in the Source tab, click in the border next to your OPEN cursor statement to place a breakpoint.

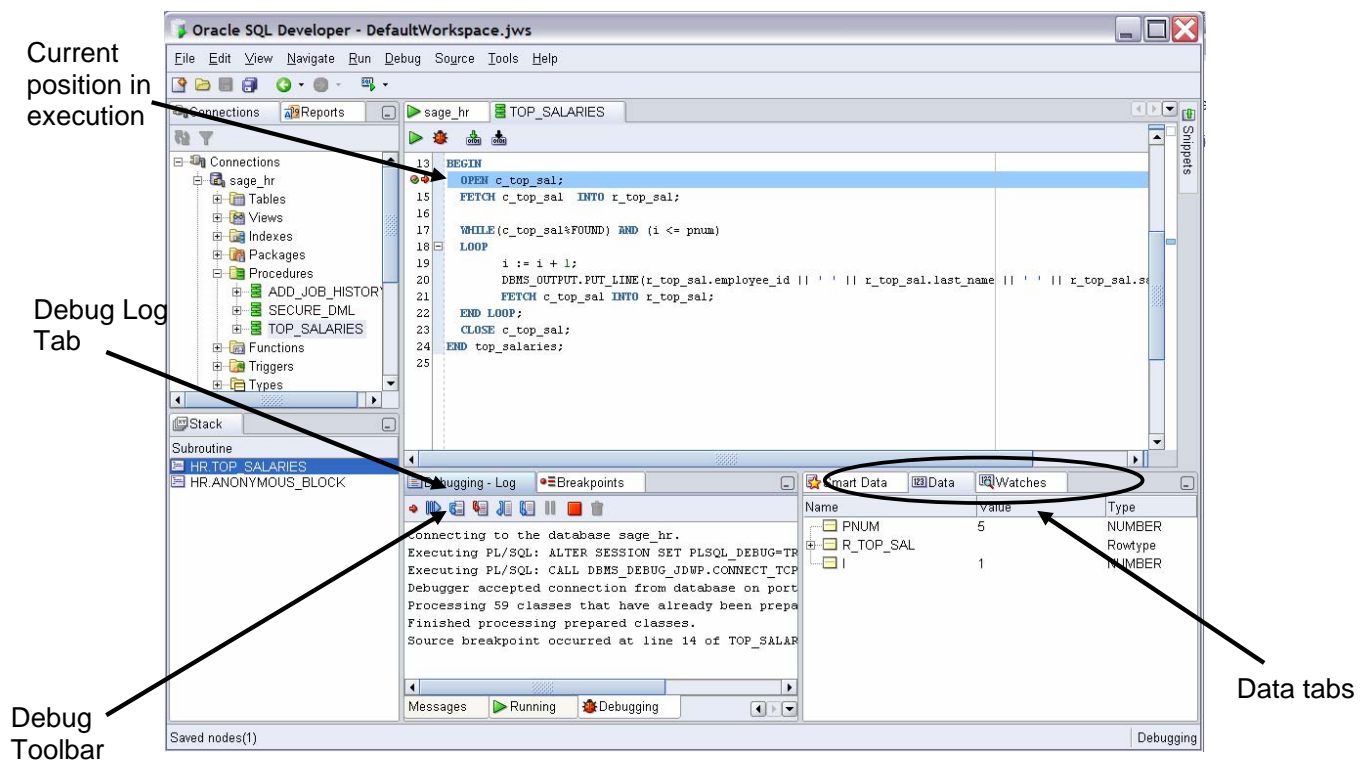


5. Click Compile for Debug from the TOP_SALARIES source tab toolbar and then click the ladybug icon to debug the procedure. In the debug PL/SQL dialog, enter 5 for the value of pNum and then click ok.

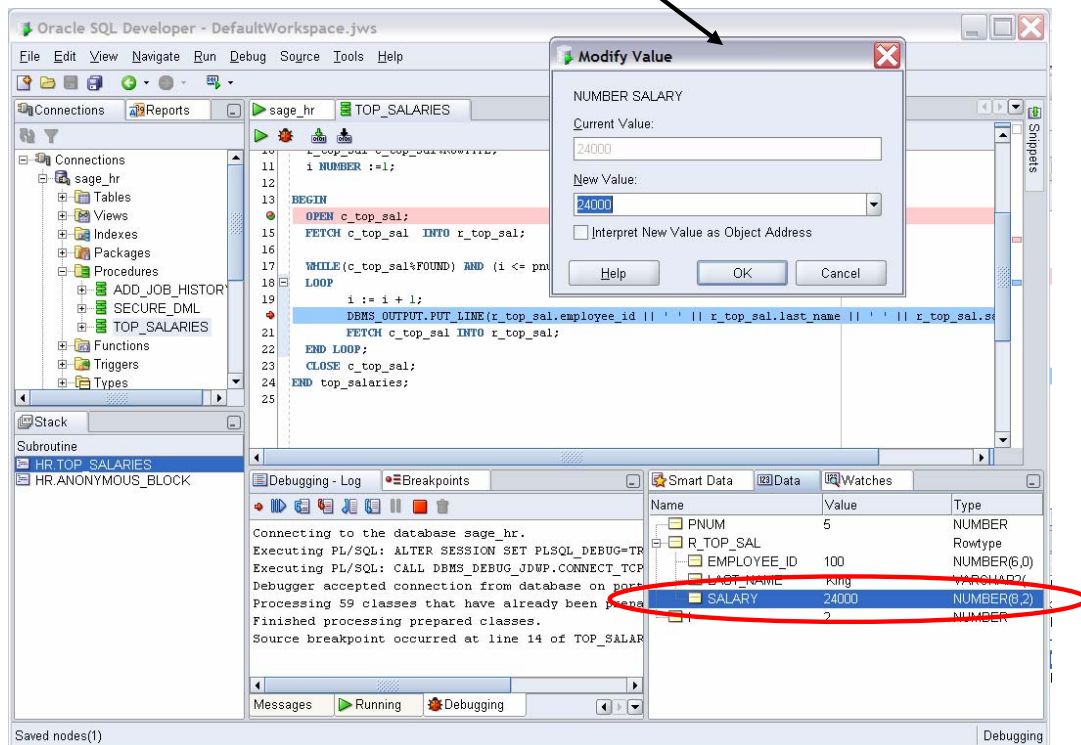



6. The debugger-log will display, as well as the Smart Data, Data and Watches tabs. You can now use the debugger toolbar to step through the procedure and determine how the data and parameters change.

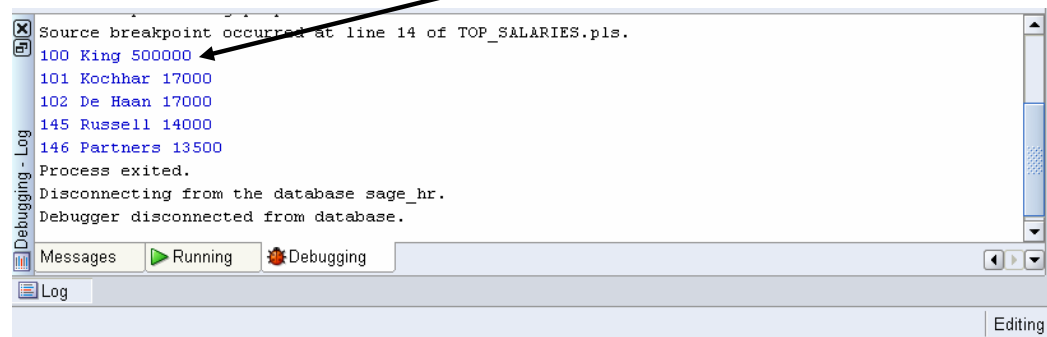
SQL Developer in Debug mode.



- Step into the procedure and stop at the DBMS_OUTPUT.PUT_LINE statement. Check the cursor record data in the Data tab. You can change the values by double clicking to display the Modify Value dialog.



8. Change the salary in the dialog and click OK. Click the Resume button  from the Debugging Log toolbar to execute the rest of the procedure. In the Debugging Log panel you will see your results including the value you changed at runtime in the debugger.



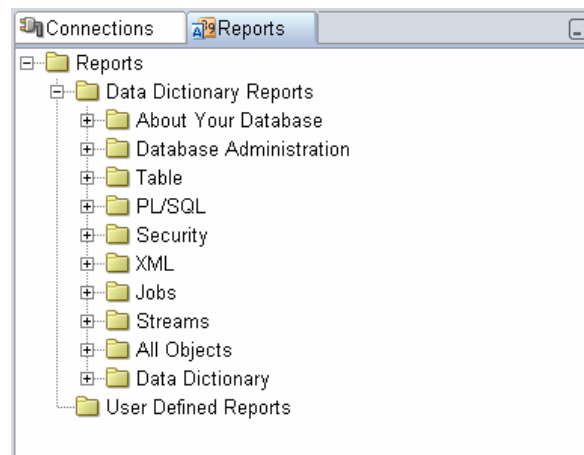


5 Hands On – Reports

SQL Developer provides an array of reports that you can execute at any time. The pre-defined Data Dictionary Reports allow you to analyse database information ranging from Security to Data storage. SQL Developer also allows you to create and save your own reports.

5.1 Data Dictionary Reports

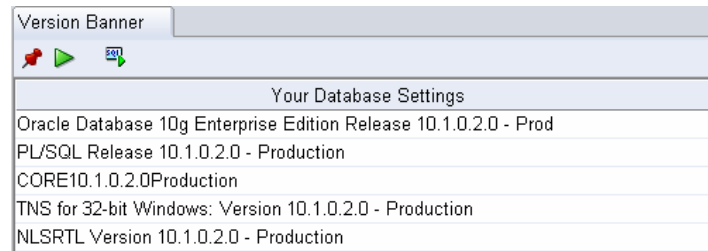
1. Within SQL Developer, click on the Reports tab next to the Connections tab to display the Reports Navigator. Expand the Reports and Data Dictionary Reports nodes to display the available categories of pre-defined reports.



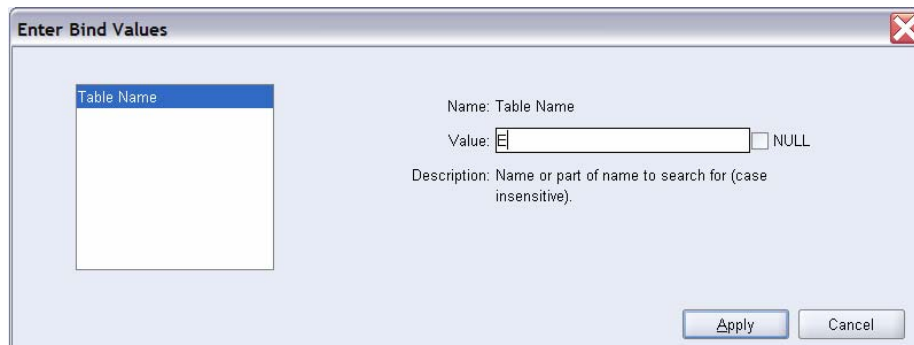
2. Expand the About your Database node and select the Version Banner Report. You will be asked to select the connection you wish the report to run against. Select our *sage_hr* connection and then click OK.



3. A simple report is displayed containing information about the database with which your connection is associated.




- Next expand the Table node underneath, and select the User Tables report. This Report will display information about the tables owned by the user. An Enter Bind Values dialog will display.



- Click NULL to remove the tick and enter E in the value field. Click Apply to run this report and limit the results to table names containing an uppercase E.

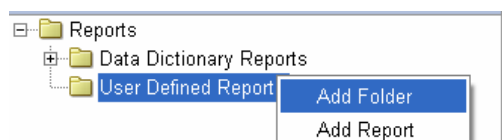
Table Name	TABLESPACE	NAME	LOGGING	Number of Rows	Blocks	EMPTY BLOCKS	AVERAGE ROW LENGTH	CACHE	DATE LAST ANALYZED	Last Analyzed	Partitioned	IOT	TYPE	COMPRESSION
ADDRESSES	SYSAUX		YES					N		minutes ago	NO			DISABLED
BINIONGZ1YES0SvzhJW...	SYSAUX		YES					N		minutes ago	NO			DISABLED
COUNTRIES	SYSAUX		YES	25				N	06-SEP-06	12.3 days ago	NO	IOT		DISABLED
DEPARTMENTS	SYSAUX		YES	27	5	0		N	20-JUL-05	427.2 days a...	NO			DISABLED
EMPLOYEES	SYSAUX		YES	107	5	0		N	21-JUL-05	426.1 days a...	NO			DISABLED
REGIONS	SYSAUX		YES	4	5	0		N	20-JUL-05	427.2 days a...	NO			DISABLED

- Review the information in the report and then click the Run Report in SQL Worksheet  icon to display the SQL statement in the SQL Worksheet. In the Statement area you can see the query details which will be executed to produce the report.

5.2 User Defined Reports

We will now create a user defined report that can be saved, allowing it to be executed or modified in the future. You can also create Folders to group your reports together in appropriate categories.

- Right click on User Defined Reports and select Add Folder from the context menu.

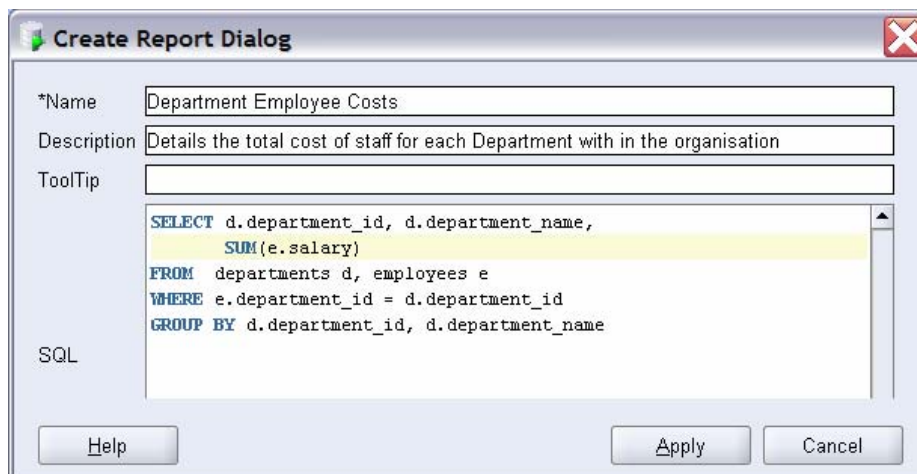


2. Enter a Folder Name of HR Reports and a meaningful description then click Apply.

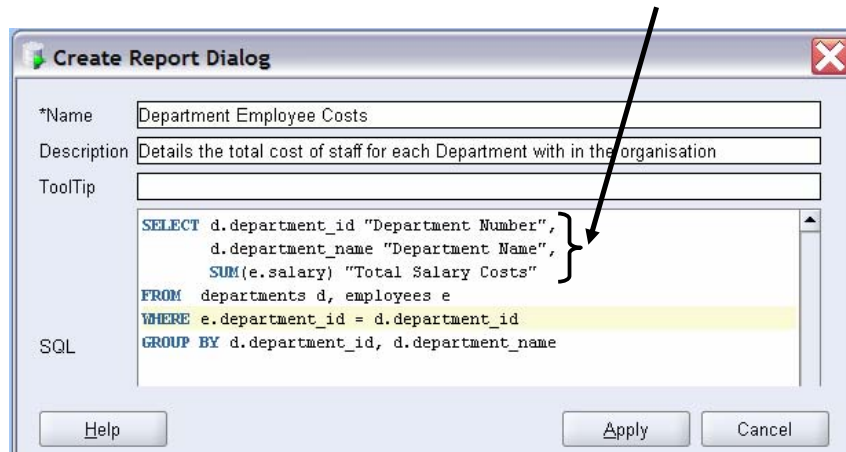


The new folder should be displayed in the Reports navigator under User Defined Reports.

3. Right click on the HR Reports folder and select Add Report from the Context menu. In the Create Report Dialog enter a Report name and description as in the following screen. Then enter the SQL statement you performed earlier in the "Using the SQL Worksheet to perform SQL statements" hands-on session, to display the total salaries for each department. You can enter this manually or retrieve it from the SQL History within your SQL Worksheet.



4. Add an Alias as in the following screen to each of the columns to improve the appearance and readability of the report when it displayed. Click Apply.



- The report will be displayed under your folder. Click the report and then select *sage_hr* as the connection for the report to be executed. Click OK.

Your Report should be displayed in a new tab.

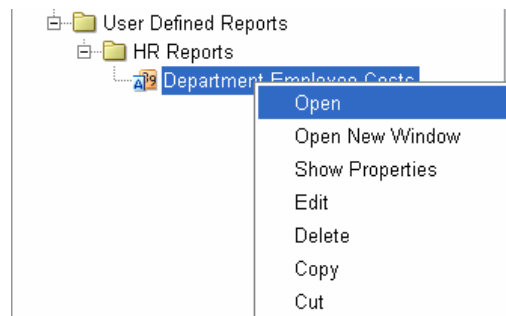
Report Title

Department Employee Costs		
Department Number	Department Name	Total Salary Costs
10	Administration	4400
20	Marketing	19000
30	Purchasing	24900
40	Human Resources	6500
50	Shipping	156400
60	IT	28800
70	Public Relations	10000
80	Sales	304500
90	Executive	58000
100	Finance	51600
110	Accounting	20300

Column Name Alias's

5.3 Standard Actions

The context menu allows you to access the standard actions for both reports and folders.



Your user defined reports can be found in the following directory.

Windows *C:\Documents and Settings\<user-name>\.sqldeveloper\UserReports.xml*

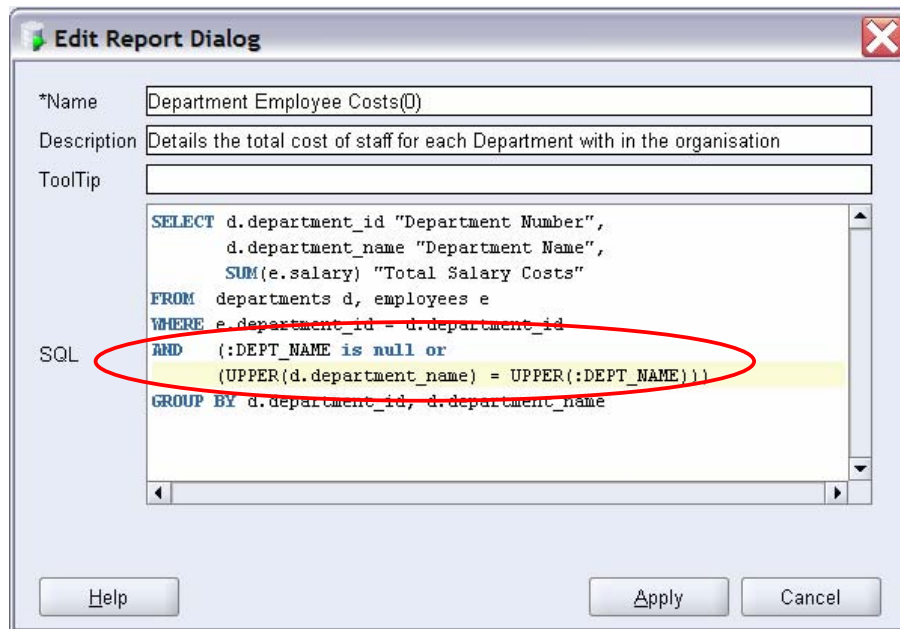
Linux or Mac OS X *~/.sqldeveloper/UserReports.xml*

5.4 Bind Variable Reports

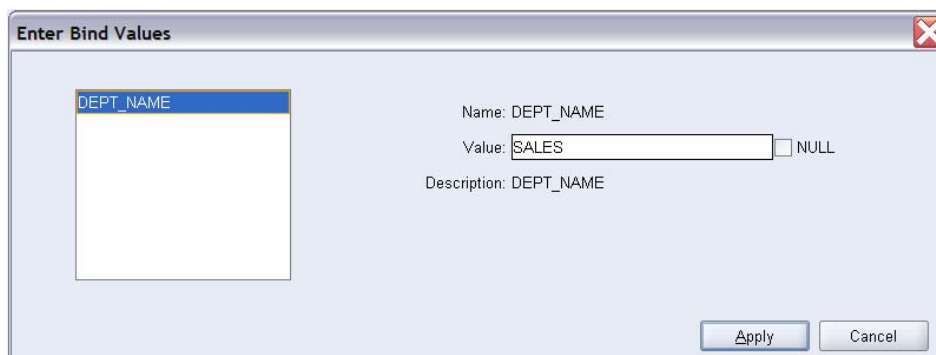
You can include bind variables in your user defined reports to further restrict the results.

- Using the context menu you can Copy and then Paste the Departments Employee Costs reports within the HR reports folder to create a second report or simply modify the original. Next add the following to the WHERE clause to restrict the query by Department Name when desired.

```
AND (:DEPT_NAME is null or
      (UPPER(d.department_name) = UPPER(:DEPT_NAME)))
```



- Next run your report with the extra WHERE clause. The Enter Bind Values dialog will appear. You can run the report without entering a variable by clicking Apply or de-select the NULL checkbox and enter the name of a department such as SALES to restrict the report.



- Click Apply.

Department Employee Costs		
Department Employee Costs(0)		
Department Number	Department Name	Total Salary Costs
80	Sales	304500

The displayed report will contain only the Sales Department's details.