Oracle 11g Developer
Curriculum

What you will learn

Learn the SQL essentials using SQL Developer on Linux. This course offers students an introduction to Oracle Database 10g database technology. In this class, students learn the concepts of relational databases and the powerful SQL programming language. This course provides the essential SQL skills that allow developers to write queries against single and multiple tables, manipulate data in tables, create database objects, and query meta data. Demonstrations and hands-on practices reinforce the fundamental concepts. In this course, students use Oracle SQL Developer on Linux as the main development tool.

Learn to:

- Use SQL Statements to retrieve data from tables
- Create and Manage Tables
- Employ SQL functions to generate and retrieve customized data
- Run data manipulation statements (DML) to update data in the
- Oracle Database 11g
-

Duration: 24 Hrs

- Application Developers
- Business Intelligence Developer
- Database Administrators
- Database Designers
- End Users
- Forms Developer
- PL/SQL Developer
- Portal Developer

SQL Fundamentals Course Topics

- Introduction
- List the Oracle Database 11g Main
- Features An Overview of: components, internet platform, app server and
- developer suite
- Describe Relational and Object Relational Database Designs
- Review the System Development Life Cycle
- Define the term Data Models
- Describe different means of Sorting Data
- Show how Multiple Tables can be related
- Describe how SQL Communicates to the Database

Writing SQL SELECT statements

- Define projection, selection, and join terminology
- Review the basic SQL SELECT statement syntax
- Select all columns using a wildcard notation from a table
- State simple rules and guidelines for writing SQL statements
- Write a query containing the arithmetic operators
- Use aliases to customize column headings
- Create a character expression with the concatenation operator
- Use the quote operator q to control text strings containing the quote character

Using the Oracle SQL Developer Environment

- Creating a Database Connection
- Browsing Database Objects
- Using the SQL Worksheet
- Executing SQL Statements
- Formatting the SQL Code
- Saving SQL Statements
- Running Script Files

Restricting Data

- Limit Rows Using a Selection
- Use the WHERE Clause
- List the main Comparison Conditions
- Use the LIKE Condition to Compare Literal
- Values List the Logical Conditions AND, OR, NOT
- Use Multiple Conditions in the WHERE clause
- Describe the Rules of Precedence

Sorting Data

- Sorting Rows with the ORDER BY Clause
- Use the && Substitution Variable
- Use the VERIFY Command

SQL Functions

Differentiate between Single Row and Multiple Row SQL Functions Categorize the Character Functions into Case Manipulation and Character Manipulation types Explain the Numeric Functions ROUND, TRUNC, and MOD List the Rules for Applying the Arithmetic Operators on Dates Use the Arithmetic Operators with Dates in the SELECT Clause Explain the DATE functions MONTHS_BETWEEN, ADD_MONTHS, NEXT_DAY, LAST_DAY, ROUND, and TRUNC Explain Implicit and Explicit conversion Nest Functions to Perform Multiple Tasks in One Statement

Conditional Expressions

- Use the CASE Expression
- Explain the DECODE Expression

Aggregating Data Using Group Functions

- Categorize the Types of Group Functions
- Use the AVG, SUM, MAX, MIN, and COUNT Functions in a Query
- Utilize the DISTINCT Keyword with the Group Functions
- Describe how Nulls are handled with the Group Functions
- Create Groups of Data with the GROUP BY Clause
- Group Data by more than one column
- Avoid Illegal Queries with the Group Functions
- Exclude Groups of Data with the HAVING Clause

Displaying Data from Multiple Tables

- Identify Types of Joins
- Retrieve Records with Natural Joins
- Use Table Aliases to write shorter code and explicitly identify
- columns from multiple tables
- Create a Join with the USING clause to identify specific
- columns between tables
- Use the ON clause to specify arbitrary conditions or specify
- columns to Join
- Create a Three-way join with the ON clause to retrieve

- information from 3 tables
- List the Types of Outer Joins LEFT, RIGHT, and FULL
- Generating a Cartesian Product

Using Subqueries

- Use a sub query to solve a problem
- Identify where sub queries can be placed in a SELECT statement
- Describe the types of sub queries (single row, multiple row)
- Show the single row sub query operators
- Use the group functions in a sub query
- Identify illegal statements with sub queries
- Show the multiple row sub query operators
- Explain how null values are handled in sub queries

Using the Set Operators

- Use the UNION operator to return all rows from multiple tables
- and eliminate any duplicate rows
- Use the UNION ALL operator to return all rows from multiple
- tables (with duplicates)
- Describe the INTERSECT operator
- Use the INTERSECT operator
- Explain the MINUS operator
- Use the MINUS operator
- List the SET operator guidelines
- Order results when using the UNION operator

Inserting and Updating Data

- Write INSERT statements to add rows to a table
- Insert Special Values
- Copy Rows from Another Table
- Update Rows in a Table

Deleting Data

- Use DELETE statements to remove rows from a table
- Delete Rows Based on Another Table
- Describe the TRUNCATE Statement

Database Transactions

- Save and Discard Changes to a Table through Transaction
- Processing (COMMIT, ROLLBACK, and SAVEPOINT)
- Show how Read Consistency works

Using DDL Statements

- List the main database objects
- Identify the Naming Rules
- Display the basic Syntax for Creating a Table
- Show the DEFAULT option
- List the Data Types that are available for Columns

Managing Tables

- Explain the different types of constraints
- Show resulting exceptions when constraints are violated with
- DML statements
- Create a table with a sub query

- Describe the ALTER TABLE functionality
- Remove a table with the DROP statement

Creating Other Schema Objects

- Categorize simple and complex views and compare them
- Create a view
- Retrieve data from a view
- Explain a read-only view
- List the rules for performing DML on complex views
- Create a sequence
- List the basic rules for when to create and not create an index
- Create a synonym

Managing Objects with Data
Dictionary Views

- Describe the structure of each of the dictionary views
- List the purpose of each of the dictionary views
- Write queries that retrieve information from the dictionary views on the schema objects
- Use the COMMENT command to document objects

Appendices

- Oracle Join Syntax
- Using SQL*Plus
- Using SQL Developer
- Additional Practices

What you will learn

This course introduces students to PL/SQL and helps them understand the benefits of this powerful programming language. In the class, students learn to create PL/SQL blocks of application code that can be shared by multiple forms, reports, and data management applications. Students learn to create anonymous PL/SQL blocks and are introduced to stored procedures and functions. They learn about declaring variables, trapping exceptions and they also learn to declare and control cursors. In class students learn to develop, execute and manage PL\SQL stored program units like procedures, functions, packages and database triggers. Student also learns to manage object dependencies and recompilation of invalid objects. This course also describes the characteristics and ways of manipulation of large objects. Students are introduced to the utilization of some of the Oracle-supplied packages. Learn To: Create Executable Section and write Control Structures Create and manage Procedures, Functions, Packages and Triggers Work with Composite Data Types and cursors Utilizing Oracle-Supplied Packages in Application DevelopmentIncluding Exception HandlingManage Dependencies and Large ObjectsThis course counts towards the Hands-on course requirement for the Oracle Database 10g Administrator Certification. Only instructor-led inclass or instructor-led online formats of this course will meet the Certification Hands-on Requirement. Self Study CD-Rom and Knowledge Center courses are excellent study and reference tools but DO NOT meet the Hands-on Requirement for certification.

Duration : 40 Hrs
Oracle Database 11g: Program with PL/SQL
Course Topics
Introduction to PL/SQL

- What is PL/SQL
- PL/SQL Environment
- Benefits of PL/SQL
- Overview of the Types of PL/SQL blocks
- Create and Execute a Simple Anonymous Block
- Generate Output from a PL/SQL Block

- iSQL*Plus as PL/SQL Programming Environment

Audience

- Database Administrators
- Database Designers
- Forms Developer
- PL/SQL Developer
- Technical Consultant

Prerequisites

- Required Prerequisites
- Oracle Database 10g: Introduction to SQL

Declaring PL/SQL Identifiers

- Identify the Different Types of Identifiers in a PL/SQL subprogram
- Use the Declarative Section to Define Identifiers
- List the Uses for Variables
- Store Data in Variables
- Declare PL/SQL Variables

Writing Executable Statements

- Describe Basic Block Syntax Guidelines
- Use Literals in PL/SQL
- Customize Identifier Assignments with SQL Functions
- Use Nested Blocks as Statements
- Reference an Identifier Value in a Nested Block
- Qualify an Identifier with a Label
- Use Operators in PL/SQL
- Use Proper PL/SQL Block Syntax and Guidelines

Interacting with the Oracle

- Server Identify the SQL Statements You Can Use in PL/SQL
- Include SELECT Statements in PL/SQL
- Retrieve Data in PL/SQL with the SELECT Statement
- Avoid Errors by Using Naming Conventions When Using
- Retrieval and DML Statements
- Manipulate Data in the Server Using PL/SQL
- The SQL Cursor concept
- Use SQL Cursor Attributes to Obtain Feedback on DML
- Save and Discard Transactions

Writing Control Structures

- Control PL/SQL Flow of Execution
- Conditional processing Using IF Statements
- Conditional Processing CASE Statements
- Handle Nulls to Avoid Common Mistakes
- Build Boolean Conditions with Logical Operators
- Use Iterative Control with Looping Statements

Working with Composite Data Types

- Learn the Composite Data Types of PL/SQL Records and Tables
- Use PL/SQL Records to Hold Multiple Values of Different Types
- Inserting and Updating with PL/SQL Records
- Use INDEX BY Tables to Hold Multiple Values of the Same Data Type

Course Objectives

- Write PL/SQL code to interface with the database
- Design PL/SQL program units that execute efficiently
- Use PL/SQL programming constructs and conditional control
- statements
- Handle run-time errors
- Describe stored procedures and functions
- Write dynamic SQL for more coding flexibility
- Design PL/SQL code for predefined data types, local
- subprograms, additional programs and standardized constants and exceptions
- Use the compiler warnings infrastructure
- Manipulate large objects
- Create triggers to solve business challenges
- Manage dependencies between PL/SQL subprograms
- Schedule PL/SQL jobs to run independently
- Create stored procedures and functions
- Design PL/SQL packages to group and contain related constructs
- Create overloaded package subprograms for more flexibility
- Categorize the Oracle supplied PL/SQL packages

Using Explicit Cursors

- Cursor FOR Loops Using Sub-queries
- Increase the Flexibility of Cursors By Using Parameters
- Use the FOR UPDATE Clause to Lock Rows
- Use the WHERE CURRENT Clause to Reference the Current Row
- Use Explicit Cursors to Process Rows Explicit Cursor Attributes
- Cursors and Records

Handling Exceptions

- Handling Exceptions with PL/SQL
- Predefined Exceptions
- Trapping Non-predefined Oracle Server Errors
- Functions that Return Information on Encountered Exceptions
- Trapping User-Defined Exceptions
- Propagate Exceptions
- Use The RAISE_APPLICATION_ERROR Procedure To Report
- Errors To Applications

Creating Stored Procedures

- Describe the block structure for PL/SQL stored procedures
- Invoke a stored procedure/function from different tools
- Call a stored procedure with host variables from iSQL*Plus,
- Forms, Java, C, etc
- Invoke a stored procedure from an anonymous block
- or another stored procedure
- List the CREATE OR REPLACE PROCEDURE syntax
- Identify the development steps for creating a stored procedure
- Use the SHOW ERRORS command
- View source code in the USER_SOURCE dictionary view

Creating Stored Functions

- Describe stored functions

- List the CREATE OR REPLACE FUNCTION syntax
- Identify the steps to create a stored function
- Execute a stored function
- Identify the advantages of using stored functions in SQL statements
- Identify the restrictions of calling functions from SQL statements
- Remove a function

Creating Packages

- List the advantages of packages
- Describe packages
- Show the components of a package Diagram the visibility of constructs within a package
- Develop a package
- Create the package specification
- Declare public constructs
- Create the package body

Using More Package Concepts

- List the benefits of overloading
- Show overloading example
- Use forward declarations in packages Create a one-time only procedure (package code initialization)
- List the restrictions on package functions used in SQL
- Encapsulate code in a package demonstration
- Invoke a user-defined package function from a SQL statement
- Utilize the persistent state of package variables

Utilizing Oracle Supplied Packages in Application Development

- List the various uses for the Oracle supplied packages
- Reuse pre-packaged code to complete various tasks from developer to DBA purposes
- Use the DESCRIBE command to view the package specifications and overloading
- Explain how DBMS_OUTPUT works
- (in conjunction with SET SERVEROUPUT ON)
- Interact with operating system files with UTL_MAIL
- Describe file processing with UTL_FILE
- Review UTL_FILE routines and exceptions
- Use UTL_FILE to generate a report to a file

Dynamic SQL and Metadata

- Describe using native dynamic SQL
- List the execution flow of SQL

- Show the syntax for the EXECUTE IMMEDIATE statement for native dynamic SQL
- Create a procedure to generate native dynamic SQL using
- EXECUTE IMMEDIATE to delete rows from a table
- Describe the DBMS_SQL package
- Provide an example of DBMS_SQL
- List the advantages of using Native Dynamic SQL Over the DBMS_SQL package

Design Considerations for PL/SQL Code

- Standardize constants with a constant package
- Standardize exceptions with an exception handling package
- Introduce local sub-programs
- Use local sub-programs
- Track run time errors with an exception package

- Describe the NOCOPY compiler hint
- Use the NOCOPY compiler hint
- Explain the effects of NOCOPY

Managing Dependencies

- Define dependent and referenced objects
- Diagram dependencies with code, views, procedures, and tables
- Manage local dependencies between a procedure, view, and a table
- Analyze a scenario of local dependencies
- Display direct dependencies using the USER_DEPENDENCIES view
- Run the UTL_DTREE.SQL script to create objects that enable you
- to view direct and indirect dependencies
- Predict the effects of changes on dependent objects

Manipulating Large Objects

- Describe a LOB object
- Diagram the anatomy of a LOB
- Manage and list the features on internal LOBs
- Describe, manage, and secure BFILEs
- Create and use the DIRECTORY object to access and use BFILEs
- Prepare BFILEs for usage
- Use the BFILENAME function to load BFILEs
- Describe the DBMS_LOB package

Creating Triggers

- Describe the different types of triggers and how they execute
- List the benefits and guidelines of using database triggers
- Show how triggers are executed with a basic database trigger example
- Show syntax and create DML triggers, and list the DML trigger components
- Explain the firing sequence of triggers
- Create a DML statement and row level triggers
- Use the OLD and NEW qualifiers to reference column values
- Use conditional predicates with triggers

Applications for Triggers

- Create triggers for DDL events of CREATE, ALTER, and DROP
- Create triggers for system events of SERVERERROR, STARTUP,
- SHUTDOWN, LOGON and LOGOFF
- Define a mutating table
- Describe business application scenarios for implementing with triggers
- Describe the privileges required to manage triggers

Understanding and Influencing the PL/SQL Compiler

- List the features of native compilation
- Describe the features of the PL/SQL compiler in Oracle Database 11g
- Identify the 3 parameters used to influence compilation
- (PLSQL_CODE_TYPE, PLSQL_DEBUG, PLSQL_OPTIMIZE_LEVEL)
- Show how to set the parameters
- Describe the dictionary view used to see how code is compiled
- (USER_PLSQL_OBJECTS)
- Change the parameter settings, recompile code, and view the results
- Describe the compiler warning infrastructure in Oracle Database 10g
- List the steps used in setting compiler warning levels

What you will learn

Leverage your investment by taking advantage of web technologies to easily and quickly construct sophisticated database forms and business logic with minimal effort. This course focuses on teaching students to use Oracle Forms Developer 11g (10.1.2.0.2) to rapidly build scalable, high-performance applications for the Internet. In this course students build, test, debug, and deploy interactive Internet applications. Working in a graphical user interface (GUI) environment, they develop an order entry application from the ground up. This application incorporates several advanced features that provide a rich user experience while implementing business rules. This course counts towards the Hands-on course requirement for the Oracle Forms Developer Certified Professional Certification. Only instructor-led inclass or instructor-led online formats of this course will meet the Certification Hands-on Requirement. Self Study CD-Rom and Knowledge Center courses are excellent study and reference tools but DO NOT meet the Hands-on Requirement for certification.

Learn To:

Customize forms with user input items such as check boxes, list items, radio groups, and Pluggable Java Components Integrate Java into Forms applications by using JavaBeans Control navigation, data access, validation, and transactions by creating event-related triggers Enable Forms applications running on the Web to access files and applications on the client computer Display Forms elements and data in multiple canvases and windows Deploy Forms applications to the Web

Duration : 40 Hrs

Oracle Forms Developer 11g: Build Internet Applications Audience

- Application Developers
- Developer
- Forms Developer
- PL/SQL Developer
- Support Engineer
- Technical Consultant

Course Objectives

- Create form modules, including components for database interaction
- and GUI controls
- Display form modules in multiple windows and use a variety of layout styles
- Test form modules in a Web browser
- Debug form modules in a 3-tier environment
- Implement triggers
- Reuse objects and code
- Link one form module to another

Course Topics

- Introducing Oracle Forms Developer and Forms Services
- Grid Computing
- Oracle 11g Products
- Oracle Application Server 10g Architecture
- Benefits and Components of Oracle Developer Suite 11g
- Running a Forms Developer Application
- Working in the Forms Developer Environment

Creating Forms Modules

- Creating a Basic Forms Module
- Creating a Master-Detail Forms Module
- Modifying the Data Block
- Modifying the Layout

Adding Functionality to Items

- Coding Item Interaction Triggers
- Defining Functionality for Check Boxes
- Changing List Items at Run Time
- Displaying LOVs from Buttons
- Populating Image Items
- Populating and Displaying Hierarchical Trees
- Interacting with JavaBeans

Working with Data Blocks and Frames

- Using the Property Palette
- Managing Object Properties
- Creating and Using Visual Attributes
- Controlling the Behavior and Appearance of Data Blocks
- Controlling Frame Properties
- Creating Control Blocks
- Deleting Data Blocks

Working with Input Items

- Creating Text Items
- Controlling the Behavior and Appearance of Text Items
- Creating LOVs
- Defining Editors
- Creating Check Boxes
- Creating List Items
- Creating Radio Groups

Working with Non Input Items

- Creating a Display Item
- Creating an Image Item
- Creating a Push Button
- Creating a Calculated Item
- Creating a Hierarchical Tree Item
- Creating a Bean Area Item

Working with Windows and Canvases

- Overview of Windows and Canvases
- Displaying a Forms Module in Multiple Windows
- Creating a New Window
- Displaying a Forms Module on Multiple Layouts
- Creating a New Content Canvas
- Creating a New Stacked Canvas
- Creating a New Toolbar
- Canvas Creating a New Tab Canvas

Producing Triggers

- Grouping Triggers into Categories
- Defining Trigger Components: Type, Code, and Scope
- Specifying Execution Hierarchy
- Using the PL/SQL Editor
- Writing Trigger Code
- Using Variables and Built-ins
- Using the When-Button-Pressed and When-Window-Closed Triggers

Debugging Triggers

- The Debugging Process
- The Debug Console
- Setting Breakpoints
- Debugging Tips
- Running a Form in Debug Mode
- Stepping through Code

Run-Time Messages and Alerts

- Built-ins and Handling Errors
- Controlling System Messages
- The FORM_TRIGGER_FAILURE Exception
- Using Triggers to Intercept System Messages
- Creating and Controlling Alerts
- Handling Server Errors

Query Triggers

- SELECT Statements Issued During Query Processing
- WHERE and ORDER BY Clauses and the ONETIME_WHERE Property
- Writing Query Triggers
- Query Array Processing
- Coding Triggers for Enter-Query Mode
- Overriding Default Query Processing
- Obtaining Query Information at Run Time

Validation

- Validation Process
- Controlling Validation Using Properties
- Controlling Validation Using Triggers
- Performing Client-Side Validation with PJCs
- Tracking Validation Status
- Using Built-ins to Control When Validation Occurs

Navigation

- Navigation Overview
- Understanding Internal Navigation
- Using Object Properties to Control Navigation
- Writing Navigation Triggers: When-New--Instance, Pre- and Post- Triggers
- The Navigation Trap
- Using Navigation Built-ins in Triggers

Transaction Processing

- The Commit Sequence of Events
- Characteristics and Common Uses of Commit Triggers
- Testing the Results of Trigger DML DML
- Statements Issued During Commit Processing
- Overriding Default Transaction Processing
- Running Against Data Sources Other Than Oracle
- Getting and Setting the Commit Status
- Implementing Array DML

Writing Flexible Code

- What Is Flexible Code?
- Using System Variables for Flexible Coding
- Using Built-in Subprograms for Flexible Coding

- Referencing Objects by Internal ID
- Referencing Items Indirectly

Sharing Objects and Code

- Working with Object Libraries
- Working with SmartClasses
- Reusing PL/SQL
- Working with PL/SQL Libraries

Using WebUtil to Interact with the Client

- Benefits of WebUtil
- Integrating WebUtil into a Form
- Interacting with the Client`

Introducing Multiple Form Applications

- Multiple Form Applications Overview
- Starting Another Forms Module
- Defining Multiple Form Functionality
- Sharing Data Among Modules

What you will learn

In this course, participants learn how to design and build a variety of standard and custom Web and paper reports using Oracle Reports Developer (9.0.4.0.1). Working in the declarative environment of Reports Builder, participants learn how to retrieve, display, and format data from any data source in numerous reporting styles and publish the output to any destination. Participants learn how to add dynamic content to a static Web page and publish reports to the Web. In addition, participants learn how to customize more complex reports, embed graphical charts, and use OracleAS Reports Services to deploy the reports.

Duration : 40 Hrs
Oracle Reports Developer 11g: Build Reports
Course Topics

- Introduction to Oracle Reports Developer
- Business Intelligence
- Enterprise Reporting
- Oracle Reports Developer
- Oracle Database 11g
- Oracle Developer Suite 11g
- Oracle Application Server 11g
- OracleAS Reports Services
- OracleAS Reports Services Architecture for the Web

Designing and Running Reports

- Understanding User Requirements
- Designing Reports
- Tabular
- Master-Detail
- Master with Two Details
- Matrix
- Retrieving and Sharing Data
- Running a Report

Audience

- Business Intelligence Developer
- Reports Developer
- Technical Consultant

Course Objectives

Use report parameters and customize a runtime parameter form Enhance reports using graphs and PL/SQL trigger code Maximize report performance using OracleAS Reports Services and tune reports Publish a report on the Web and build reports to run in various languages Manage report templates, create mailing labels and letters Identify the main components in a report document and how they are related Identify the differences between Web and paper reports Identify standard report design styles and run existing reports to various output destinations Create and modify basic tabular reports and high quality Web reports Create other report styles such as break reports and matrix reports Create reports by accessing data from various data sources Add dynamic data to an existing HTML page

Exploring Oracle Reports Developer

- Reports Developer Executables
- Invoking Reports Builder
- Reports Builder Modules Report Data and Layout
- Reports Builder Components
- Object Navigator
- Report-Level Objects
- Data Model Objects

Creating a Paper Report

- Report Module Components
- Building a Paper Report
- Viewing the Paper Report Output
- Saving the Report Definition
- Reentering the Wizard
- Creating Break Reports
- Break Report Labels
- Creating Mailing Labels and Letters

Enhancing a Basic Paper Report

- What Is the Paper Design?
- The Paper Design Window
- Modifying a Report
- Aligning Columns
- Setting a Format Mask
- Manipulating Objects
- Modifying Visual Attributes
- Applying Conditional Formatting

Managing Report Templates

- Using Report Templates
- Modifying a Template
- Customizing the Template Margin
- Customizing the Template Body
- Adding Web Links to a Template for Report HTML Output
- Predefining Your Own Templates
- Adding a Template Preview Image

Enhancing Reports Using the Paper Layout

- Viewing the Paper Layout
- Designing Multipanel Reports
- Printing Multipanel Reports

- Different Objects in the Paper Layout
- The Paper Layout Layers
- Report Processing
- Paper Layout Tools
- Report Bursting

Controlling the Paper Layout:
Common Properties

- Modifying Paper Layout Object Properties
- Common Layout Properties
- Sizing Objects
- Anchors
- Layout Object Relationships
- Pagination Icons in the Paper
- Layout Controlling Print Frequency
- Using Format Triggers

Controlling the Paper Layout: Specific Properties

- Properties of a Repeating Frame
- Specifying Print Direction
- Controlling the Number of Records per Page
- Controlling Spacing Between Records
- Minimum Widow Records
- System Variables
- Valid Source Columns
- Displaying File Contents

Web Reporting

- Comparing Static and Dynamic Reporting
- Adding Dynamic Content
- Creating a Report Block
- Invoking the Report Block Wizard
- Examining the Web Source Code
- rw:foreach Tag
- rw:field Tag
- Customizing Reports JSPs

Extending Functionality Using XML

- Why Use XML Report Definitions?
- Creating XML Report Definitions Partial Report Definitions: Format Modification Example
- Partial Report Definitions: Format Exception Example
- Full Report Definition: Data Model Modification Example
- Running XML Report Definitions
- Debugging XML Report Definitions

Creating a Web Report

- What Is JSP Technology?
- JSP Advantages
- Simple JSP Example
- Building a Web Report
- Using the Report Wizard
- Report Editor: Web Source View
- JSP Tags
- Web Source Example

Enhancing Reports Using the Data Model: Queries and Groups

- The Data Model Objects
- Modifying Properties of a Query
- Applying Changes
- Changing the Group Structure
- Group Hierarchy
- Ordering Data in a Group
- Query Modifications
- Filtering Data in a Group

Enhancing Reports Using the Data Model: Data Sources

- Data Source Types
- Pluggable Data Sources
- Using XML as a Data Source
- Document Type Definition File
- OLAP Data Source
- Using Text as a Data Source
- Using JDBC as a Data Source
- Using REF Cursor Queries

Enhancing Reports Using the Data Model: Creating Columns

- Data Model Columns
- Maintaining Data Source Columns
- Producing File Content Output
- Creating a Column
- Creating Summary Columns
- Displaying Subtotals
- Displaying Percentages
- Creating a Formula Column

Creating and Using Report Parameters

- Creating User Parameters
- Referencing Parameters in a Report Query
- Using Bind References
- Using Lexical References
- Hints and Tips When Referencing Parameters
- Creating a List of Values
- Referencing System Parameters
- Building a Paper Parameter Form

Embedding a Graph in a Report

- Adding a Graph to a Paper Report
- Adding a Graph to a Web Report
- Selecting the Graph Type
- Selecting the Graph Data
- Adding Options to the Graph
- Customizing Web Graphs
- The rw:graph Tag
- Customizing Graphs Using the Graph.XML File

Extending Functionality Using the SRW Package

- Contents of the SRW Package
- Outputting Messages

- Executing a Nested Report
- Restricting Data
- Initializing Fields
- Creating a Table of Contents
- Performing DDL Statements
- Setting Format Attributes

Enhancing Matrix Reports

- The Matrix Data Model
- The Matrix Paper Layout
- Creating Matrix Summaries
- Creating the Matrix Manually
- The Matrix with Group Data Model
- The Matrix with Group Layout
- Building a Nested Matrix
- Nested Matrix Paper Layout

Coding PL/SQL Triggers

- Types of Triggers in Reports
- Trigger Code
- Using Report Triggers
- Using Data Model Triggers: PL/SQL Group Filter
- Using Data Model Triggers: Parameter Validation
- Using Layout Triggers
- Using Format Triggers
- Event-Based Reporting

Maximizing Performance Using OracleAS Reports Services

- Running Reports Using OracleAS Reports Services
- Report Request Methods
- Oracle Application Server Components
- Enabling Single Sign-On Access
- Running the Web Layout: JSP Run-time Architecture
- Running the Paper Layout: Servlet Run-time Architecture
- Running a Paper Report on the Web
- Queue Manager

Building Reports: Efficiency Guidelines

- Tuning Reports
- Performance Measurement
- Non SQL Data Sources
- Investigating the Data Model
- Investigating the Paper Layout
- Running the Report
- Different Development and Run-Time Environments
- Developing Reports to Run in Different GUIs