Oracle Database 19c: PL/SQL Workshop

Oracle Database

DURATION	MODULES	COURSE CODE
5 Days	**22 Lectures**	**—**

Course Overview

This is a complete course on PL/SQL including an introduction to Programming and Managing the PL/SQL code.

What You Will Learn

Introduction

- Lesson Objectives
- Lesson Agenda
- Course Objectives
- Course Road Map
- Lesson Agenda
- Human Resources (HR) Schema for This Course
- Course Agenda
- Class Account Information
- Appendixes and Practices Used in This Course
- Lesson Agenda
- Oracle Database 19c: Focus Areas
- Oracle Database 19c
- Lesson Agenda
- PL/SQL Development Environments
- Oracle SQL Developer
- Specifications of SQL Developer
- SQL Developer Interface
- Coding PL/SQL in SQL*Plus
- SQL Developer Web
- Lesson Agenda
- Oracle SQL and PL/SQL Documentation
- Additional Resources
- Summary

- Practice 1 Overview: Getting Started

Introduction to PL/SQL

- Course Road Map
- Objectives
- Agenda
- Limitations of SQL
- Why PL/SQL?
- Why PL/SQL
- About PL/SQL
- Benefits of PL/SQL
- PL/SQL Runtime Architecture
- PL/SQL Block Structure
- Agenda
- Block Types
- Examining an Anonymous Block
- Executing an Anonymous Block
- Agenda
- Enabling Output of a PL/SQL Block
- Viewing the Output of a PL/SQL Block
- Quiz
- Summary
- Practice 2: Overview

Declaring PL/SQL Variables

- Course Road Map
- Objectives
- Agenda
- Variables
- Variables in PL/SQL
- Requirements for Variable Names
- Using Variables in PL/SQL
- Declaring and Initializing PL/SQL Variables
- Agenda
- Declaring and Initializing PL/SQL Variables
- Initializing Variables Through a SELECT Statement
- Types of Variables
- Declaring Variables
- Guidelines for Declaring and Initializing PL/SQL Variables
- Guidelines for Declaring PL/SQL Variables
- Naming Conventions of the PL/SQL Structures Used in This Course
- Data Types for Strings
- Delimiters in String Literals
- Data Types for Numeric values
- Data Types for Date and Time values

- Data Type Conversion
- Agenda
- The %TYPE Attribute
- Declaring Variables with the %TYPE Attribute
- Declaring Boolean Variables
- LOB Data Type Variables
- Composite Data Types: Records and Collections
- Agenda
- Bind Variables
- Bind Variables: Examples
- Using AUTOPRINT with Bind Variables
- Quiz
- Summary
- Practice 3: Overview

Writing Executable Statements

- Course Road Map
- Objectives
- Agenda
- Lexical Units in a PL/SQL Block
- PL/SQL Block Syntax and Guidelines
- Commenting Code
- SQL Functions in PL/SQL
- SQL Functions in PL/SQL: Examples
- Using Sequences in PL/SQL blocks
- Using Sequences in PL/SQL Blocks
- Agenda
- Nested blocks
- Nested Blocks: Example
- Variable Scope and Visibility
- Using a Qualifier with Nested Blocks
- Challenge: Determining the Variable Scope
- Agenda
- Operators in PL/SQL
- Operators in PL/SQL: Examples
- Programming Guidelines
- Indenting Code
- Quiz
- Summary
- Practice 4: Overview

Using SQL Statements Within a PL/SQL Block

- Course Road Map
- Objectives
- Agenda

- SQL Statements in PL/SQL
- SELECT Statements in PL/SQL
- Retrieving Data in PL/SQL: Example
- Retrieving Data in PL/SQL
- Naming Ambiguities
- Avoiding Naming Ambiguities
- Agenda
- Using PL/SQL to Manipulate Data
- Insert Data: Example
- Update Data: Example
- Delete Data: Example
- Merging Rows
- Agenda
- SQL Cursor
- SQL Cursor Attributes for Implicit Cursors
- Quiz
- Summary
- Practice 5: Overview

Writing Control Structures

- Course Road Map
- Objectives
- PL/SQL Control Structures
- Agenda
- IF Statement
- IF-ELSIF Statements
- Simple IF Statement
- IF THEN ELSE Statement
- IF ELSIF ELSE Clause
- NULL Value an in IF Statement
- Agenda
- CASE Expressions
- Searched CASE Expressions
- CASE Statement
- Handling Nulls
- Logic Tables
- Boolean Expression or Logical Expression?
- Agenda
- Iterative Control: LOOP Statements
- Basic Loops
- Basic Loop: Example
- WHILE Loops
- WHILE Loops: Example
- FOR Loops
- FOR Loops: Example

- FOR Loop Rules
- Suggested Use of Loops
- Nested Loops and Labels
- Nested Loops and Labels: Example
- PL/SQL CONTINUE Statement
- PL/SQL CONTINUE Statement: Example 1
- PL/SQL CONTINUE Statement: Example 2
- Quiz
- Summary
- Practice 6: Overview

Working with Composite Data Types

- Course Road Map
- Objectives
- Agenda
- Composite Data Types
- PL/SQL Records Versus Collections
- Agenda
- PL/SQL Records
- Creating a PL/SQL Record
- Creating a PL/SQL Record: Example
- PL/SQL Record Structure
- %ROWTYPE Attribute
- Creating a PL/SQL Record: Example
- Advantages of Using the %ROWTYPE Attribute
- Another %ROWTYPE Attribute: Example
- Inserting a Record by Using %ROWTYPE
- Updating a Row in a Table by Using a Record
- Agenda
- Associative Arrays (INDEX BY Tables)
- Associative Array Structure
- Steps to Create an Associative Array
- Creating and Accessing Associative Arrays
- Associative Arrays with Record values
- Using Collection Methods
- Using Collection Methods with Associative Arrays
- Nested Tables
- Nested Tables: Syntax and Usage
- Variable-Sized Arrays (Varrays)
- VARRAYs: Syntax and Usage
- Summarizing Collections
- Quiz
- Summary
- Practice 7: Overview

Using Explicit Cursors

- Course Road Map
- Objectives
- Agenda
- Cursors
- Implicit Cursors
- Explicit Cursor
- Controlling Explicit Cursors
- Agenda
- Declaring the Cursor
- Opening the Cursor
- Fetching Data from the Cursor
- Closing the Cursor
- Cursors and Records
- Cursor FOR Loops
- Explicit Cursor Attributes
- %ISOPEN Attribute
- %ROWCOUNT and %NOTFOUND: Example
- Cursor FOR Loops Using Subqueries
- Agenda
- Cursors with Parameters
- Agenda
- FOR UPDATE Clause
- WHERE CURRENT OF Clause
- WHERE CURRENT OF Clause: Example
- Quiz
- Summary
- Practice 8: Overview

Handling Exceptions

- Course Road Map
- Objectives
- Agenda
- What Is an Exception?
- Handling an Exception: Example
- Understanding Exceptions with PL/SQL
- Handling Exceptions
- Exception Types
- Agenda
- Syntax to Trap Exceptions
- Guidelines for Trapping Exceptions
- Trapping Internally Predefined Exceptions
- Internally Defined Exception Trapping: Example
- Trapping Predefined Exceptions

- Functions for Trapping Exceptions
- Trapping User-Defined Exceptions
- RAISE Statement
- Trapping User-Defined Exceptions
- Propagating Exceptions in a Sub-Block
- The RAISE_APPLICATION_ERROR Procedure
- Quiz
- Summary
- Practice 9: Overview

Introducing Stored Procedures and Functions

- Course Road Map
- Objectives
- Agenda
- What Are PL/SQL Subprograms?
- Differences Between Anonymous Blocks and Subprograms
- Agenda
- Procedure: Syntax
- Creating a Procedure
- Invoking a Procedure
- Agenda
- Function: Syntax
- Creating a Function
- Invoking a Function
- Passing a Parameter to the Function
- Invoking the Function with a Parameter
- Quiz
- Summary
- Practice 10: Overview

Creating Procedures

- Course Road Map
- Objectives
- Lesson Agenda
- Modularized Program Design
- Modularizing Code with PL/SQL
- Benefits of Modularization
- What Are PL/SQL Subprograms?
- Lesson Agenda
- Procedures
- What Are Procedures?
- Creating Procedures: Overview
- Creating Procedures
- Creating Procedures Using SQL Developer
- Compiling Procedures

- Calling Procedures
- Calling Procedures Using SQL Developer
- Procedures
- What Are Parameters and Parameter Modes?
- Formal and Actual Parameters
- Procedural Parameter Modes
- Comparing the Parameter Modes
- Using the IN Parameter Mode: Example
- Using the OUT Parameter Mode: Example
- Using the IN OUT Parameter Mode: Example
- Passing Parameters to Procedures
- Passing Actual Parameters: Creating the raise_sal Procedure
- Passing Actual Parameters: Examples
- Using the DEFAULT Option for the Parameters
- Lesson Agenda
- Handled Exceptions
- Handled Exceptions: Example
- Exceptions Not Handled
- Exceptions Not Handled: Example
- Removing Procedures: Using the DROP SQL Statement or SQL Developer
- Viewing Procedure Information Using the Data Dictionary Views
- Viewing Procedures Information Using SQL Developer
- Quiz
- Summary
- Practice 11-1 Overview: Creating a New Database Connection
- Practice 11-2 Overview: Creating, Compiling, and Calling Procedures

Creating Functions

- Course Road Map
- Objectives
- Lesson Agenda
- Functions
- Creating Functions syntax
- Tax Calculation
- The Difference Between Procedures and Functions
- Creating Functions: Overview
- Invoking a Stored Function: Example
- Using Different Methods for Executing Functions
- Creating and Compiling Functions Using SQL Developer
- Lesson Agenda
- Using a Function in a SQL Expression: Example
- Calling User-Defined Functions in SQL Statements
- Restrictions When Calling Functions from SQL Expressions
- Side Effects of Function Execution
- Controlling Side Effects

- Guidelines to Control Side Effects
- Lesson Agenda
- Passing Parameters to Functions
- Named and Mixed Notation from SQL: Example
- Viewing Functions Using Data Dictionary Views
- Viewing Functions Information Using SQL Developer
- Lesson Agenda
- Removing Functions: Using the DROP SQL Statement or SQL Developer
- Quiz
- Summary
- Practice 12: Overview

Debugging Subprograms

- Course Road Map
- Objectives
- Lesson Agenda
- Before Debugging PL/SQL Subprograms
- Lesson Agenda
- Debugging a Subprogram: Overview
- Lesson Agenda
- The Debugging – Log Tab Toolbar
- Tracking Data and Execution
- Lesson Agenda
- Debugging a Procedure Example: Creating a New emp_list Procedure
- Debugging a Procedure Example: Creating a New get_location Function
- Setting Breakpoints and Compiling emp_list for Debug Mode
- Compiling the get_location Function for Debug Mode
- Debugging emp_list and Entering Values for the PMAXROWS Parameter
- Debugging emp_list: Step Into (F7) the Code
- Viewing the Data
- Modifying the Variables While Debugging the Code
- Debugging emp_list: Step Over Versus Step Into
- Debugging emp_list: Step Out of the Code (Shift + F7)
- Debugging emp_list: Step to End of Method
- Debugging a Subprogram Remotely: Overview
- Summary
- Practice 13 Overview: Introduction to the SQL Developer Debugger

Creating Packages

- Course Road Map
- Objectives
- Lesson Agenda
- DBMS_OUTPUT.PUT_LINE
- What Is a Package?
- Advantages of Packages

- How Do You Create PL/SQL Packages?
- Components of a PL/SQL Package
- Application Program Interface
- Lesson Agenda
- Creating the Package Specification: Using the CREATE PACKAGE Statement
- Creating Package Specification: Using SQL Developer
- Creating the Package Body: Using SQL Developer
- Example of a Package Specification: comm_pkg
- Creating the Package Body
- Example of a Package Body: comm_pkg
- Invoking the Package Subprograms: Examples
- Invoking Package Subprograms: Using SQL Developer
- Creating and Using Bodiless Packages
- Viewing Packages by Using the Data Dictionary
- Viewing Packages by Using SQL Developer
- Removing Packages
- Removing Package Bodies
- Guidelines for Writing Packages
- Quiz
- Summary
- Practice 14 Overview: Creating and Using Packages

Working with Packages

- Course Road Map
- Objectives
- Lesson Agenda
- Why Overload Subprograms?
- Overloading Subprograms in PL/SQL
- Overloading Procedures Example: Creating the Package Specification
- Overloading Procedures Example: Creating the Package Body
- Restrictions on Overloading
- STANDARD package
- Overloading and the STANDARD Package
- Lesson Agenda
- Package Instantiation and Initialization
- Initializing Packages in Package Body
- Using User-Defined Package Functions in SQL
- User-Defined Package Function in SQL: Example
- Lesson Agenda
- Package State
- Serially Reusable Packages
- Memory Architecture
- Serially Reusable Packages
- Persistent State of Packages
- Persistent State of Package Variables: Example

- Persistent State of a Package Cursor: Example
- Executing the CURS_PKG Package
- Quiz
- Summary
- Practice 15 Overview: Working with Packages

Using Oracle-Supplied Packages in Application Development

- Course Road Map
- Objectives
- Lesson Agenda
- Using Oracle-Supplied Packages
- Examples of Some Oracle-Supplied Packages
- Lesson Agenda
- How the DBMS_OUTPUT Package Works
- Using the UTL_FILE Package
- Some of the UTL_FILE Procedures and Functions
- File Processing Using the UTL_FILE Package: Overview
- Using the Available Declared Exceptions in the UTL_FILE Package
- FOPEN and IS_OPEN Functions: Example
- Using UTL_FILE: Example
- What Is the UTL_MAIL Package?
- Setting Up and Using the UTL_MAIL: Overview
- Summary of UTL_MAIL Subprograms
- Installing and Using UTL_MAIL
- The SEND Procedure Syntax
- The SEND_ATTACH_RAW Procedure
- Sending Email with a Binary Attachment: Example
- The SEND_ATTACH_VARCHAR2 Procedure
- Sending Email with a Text Attachment: Example
- Quiz
- Summary
- Practice 16 Overview: Using Oracle-Supplied Packages in Application Development

Using Dynamic SQL

- Course Road Map
- Objectives
- Lesson Agenda
- What is Dynamic SQL?
- When do you use Dynamic SQL?
- Using Dynamic SQL
- Execution Flow of SQL Statements
- Dynamic SQL implementation
- Lesson Agenda
- Native Dynamic SQL (NDS)
- Using the EXECUTE IMMEDIATE Statement

- Dynamic SQL with a DDL Statement: Examples
- Dynamic SQL with DML Statements
- Dynamic SQL with a Single-Row Query: Example
- Executing a PL/SQL Anonymous Block Dynamically
- BULK COLLECT INTO clause
- OPEN FOR clause
- Using BULK COLLECT and OPEN FOR clause
- Summarizing Methods for Using Dynamic SQL
- Lesson Agenda
- Using the DBMS_SQL Package
- Using the DBMS_SQL Package Subprograms
- Using DBMS_SQL with a DML Statement: Deleting Rows
- Using DBMS_SQL with a Parameterized DML Statement
- Quiz
- Summary
- Practice 17 Overview: Using Dynamic SQL

Creating Triggers

- Course Road Map
- Objectives
- Lesson Agenda
- What are Triggers?
- Defining Triggers
- Why do you use Triggers?
- Trigger Event Types
- Available Trigger Types
- Trigger Event Types and Body
- Lesson Agenda
- Creating DML Triggers by Using the CREATE TRIGGER Statement
- Creating DML Triggers by Using SQL Developer
- Specifying the Trigger Execution Time
- Creating a DML Statement Trigger Example: SECURE_EMP
- Testing Trigger SECURE_EMP
- Using Conditional Predicates
- Multiple Triggers of the Same Type
- CALL Statements in Triggers
- Lesson Agenda
- Statement-Level Triggers Versus Row-Level Triggers
- Creating a DML Row Trigger
- Correlation names and Pseudorecords
- Correlation Names and Pseudorecords
- Using OLD and NEW Qualifiers
- Using OLD and NEW Qualifiers: Example
- Using the WHEN Clause to Fire a Row Trigger Based on a Condition
- Trigger-Firing Sequence: Single-Row Manipulation

- Trigger-Firing Sequence: Multirow Manipulation
- Summary of the Trigger Execution Model
- Lesson Agenda
- INSTEAD OF Triggers
- Creating an INSTEAD OF Trigger: Example
- Creating an INSTEAD OF Trigger to Perform DML on Complex Views
- Lesson Agenda
- The Status of a Trigger
- System Privileges Required to Manage Triggers
- Managing Triggers by Using the ALTER and DROP SQL Statements
- Managing Triggers by Using SQL Developer
- Viewing Trigger Information
- Using USER_TRIGGERS
- Testing Triggers
- Quiz
- Summary
- Practice 18 Overview: Creating Statement and Row Triggers

Creating Compound, DDL, and Event Database Triggers

- Course Road Map
- Objectives
- Lesson Agenda
- What is a Compound Trigger?
- Working with Compound Triggers
- Why Compound Triggers?
- Compound Trigger Structure
- Compound Trigger Structure for Views
- Compound Trigger Restrictions
- Lesson Agenda
- Mutating Tables
- Mutating Table: Example
- Using a Compound Trigger to Resolve the Mutating Table Error
- Lesson Agenda
- Creating Triggers on DDL Statements
- Creating Triggers on DDL Statements -Example
- Lesson Agenda
- Creating Database Triggers
- Creating Triggers on System Events
- LOGON and LOGOFF Triggers: Example
- Lesson Agenda
- Guidelines for Designing Triggers
- Quiz
- Summary
- Practice 19 Overview: Creating Compound, DDL, and Event Database Triggers

Design Considerations for the PL/SQL Code

- Course Road Map
- Objectives
- Lesson Agenda
- Standardizing Constants and Exceptions
- Standardizing Exceptions
- Standardizing Exception Handling
- Standardizing Constants
- Local Subprograms
- Lesson Agenda
- Definer’s and Invoker’s Rights
- Specifying Invoker’s Rights: Setting AUTHID to CURRENT_USER
- Granting Privileges to Invoker’s Rights Unit
- Lesson Agenda
- Autonomous Transactions
- Features of Autonomous Transactions
- Using Autonomous Transactions: Example
- Lesson Agenda
- Using the NOCOPY Hint
- Effects of the NOCOPY Hint
- When Does the PL/SQL Compiler Ignore the NOCOPY Hint?
- Using the PARALLEL_ENABLE Hint
- Using the Cross-Session PL/SQL Function Result Cache
- Declaring and Defining a Result-Cached Function: Example
- Using the DETERMINISTIC Clause with Functions
- Using the RETURNING Clause
- Lesson Agenda
- Using Bulk Binding
- Bulk Binding: Syntax and Keywords
- Bulk Binding FORALL: Example
- Using BULK COLLECT INTO with Queries
- Using BULK COLLECT INTO with Cursors
- Using BULK COLLECT INTO with a RETURNING Clause
- Quiz
- Summary
- Practice 20 Overview: Design Considerations for PL/SQL Code

Tuning the PL/SQL Compiler

- Course Road Map
- Objectives
- Lesson Agenda
- Optimizing PL/SQL Compiler Performance
- Initialization Parameters for PL/SQL Compilation
- Using the Initialization Parameters for PL/SQL Compilation

- Displaying the PL/SQL Initialization Parameters
- Displaying and Setting PL/SQL Initialization Parameters
- Changing PL/SQL Initialization Parameters: Example
- Lesson Agenda
- PL/SQL Compile-Time Warnings
- Benefits of Compiler Warnings
- Categories of PL/SQL Compile-Time Warning Messages
- Enabling Warning Messages
- Setting Compiler Warning Levels: Using PLSQL_WARNINGS, Examples
- Enabling Compiler Warnings: Using PLSQL_WARNINGS in SQL Developer
- Viewing the Current Setting of PLSQL_WARNINGS
- Viewing Compiler Warnings
- SQL*Plus Warning Messages: Example
- Defining PLSQL_WARNINGS for Program Units
- Lesson Agenda
- Using the DBMS_WARNINGS Package
- Using the DBMS_WARNING Package Subprograms
- The DBMS_WARNING Procedures: Syntax, Parameters, and Allowed Values
- The DBMS_WARNING Procedures: Example
- The DBMS_WARNING Functions: Syntax, Parameters, and Allowed Values
- The DBMS_WARNING Functions: Example
- Using DBMS_WARNING: Example
- Quiz
- Summary
- Practice 21 Overview: Tuning PL/SQL Compiler

Managing Dependencies

- Course Road Map
- Objectives
- Lesson Agenda
- What are Dependencies in a Schema?
- How Dependencies Work?
- Dependent and Referenced Objects
- Querying Object Dependencies: Using the USER_DEPENDENCIES View
- Querying an Object’s Status
- Categorizing Dependencies
- Lesson Agenda
- Direct Dependencies
- Indirect Dependencies
- Displaying Direct and Indirect Dependencies
- Lesson Agenda
- Fine-Grained Dependency Management
- Fine-Grained Dependency Management: Example 1
- Fine-Grained Dependency Management: Example 2
- Guidelines for Reducing Invalidation
- Object Revalidation

- Lesson Agenda
- Remote Dependencies
- Managing Remote Procedure Dependencies
- Setting the REMOTE_DEPENDENCIES_MODE Parameter
- Timestamp Checking
- Signature Checking
- Lesson Agenda
- Revalidating PL/SQL Program Units
- Unsuccessful Recompilation
- Successful Recompilation
- Recompiling Procedures
- Lesson Agenda
- Packages and Dependencies: Subprogram References the Package
- Packages and Dependencies: Package Subprogram References Procedure
- Quiz
- Summary
- Practice 22 Overview: Managing Dependencies in Your Schema
- A Using SQL Developer
- Objectives A-2
- What Is Oracle SQL Developer? A-3
- Specifications of SQL Developer A-4
- SQL Developer Interface A-5
- Creating a Database Connection A-7
- Browsing Database Objects A-10
- Displaying the Table Structure A-11
- Browsing Files A-12
- Creating a Schema Object A-13
- Creating a New Table: Example A-14
- Using the SQL Worksheet A-15
- Executing SQL Statements A-19
- Saving SQL Scripts A-20
- Executing Saved Script Files: Method 1 A-21
- Executing Saved Script Files: Method 2 A-22
- Formatting the SQL Code A-23
- Using Snippets A-24
- Using Snippets: Example A-25
- Debugging Procedures and Functions A-26
- Database Reporting A-27
- Creating a User-Defined Report A-28
- Search Engines and External Tools A-29
- Setting Preferences A-30
- Resetting the SQL Developer Layout A-32
- Data Modeler in SQL Developer A-33
- Summary A-34
- B Using SQL*Plus
- Objectives B-2

- SQL and SQL*Plus Interaction B-3
- SQL Statements Versus SQL*Plus Commands B-4
- Overview of SQL*Plus B-5
- Logging In to SQL*Plus B-6
- Displaying the Table Structure B-7
- SQL*Plus Editing Commands B-9
- Using LIST, n, and APPEND B-11
- Using the CHANGE Command B-12
- SQL*Plus File Commands B-13
- Using the SAVE, START Commands B-14
- SERVEROUTPUT Command B-15
- Using the SQL*Plus SPOOL Command B-16
- Using the AUTOTRACE Command B-17
- Summary B-18
- C Commonly Used SQL Commands
- Objectives C-2
- Basic SELECT Statement C-3
- SELECT Statement C-4
- WHERE Clause C-5
- ORDER BY Clause C-6
- GROUP BY Clause C-7
- Data Definition Language C-8
- CREATE TABLE Statement C-9
- ALTER TABLE Statement C-10
- DROP TABLE Statement C-11
- GRANT Statement C-12
- Privilege Types C-13
- REVOKE Statement C-14
- TRUNCATE TABLE Statement C-15
- Data Manipulation Language C-16
- INSERT Statement C-17
- UPDATE Statement Syntax C-18
- DELETE Statement C-19
- Transaction Control Statements C-20
- COMMIT Statement C-21
- ROLLBACK Statement C-22
- SAVEPOINT Statement C-23
- Joins C-24
- Types of Joins C-25
- Qualifying Ambiguous Column Names C-26
- Natural Join C-27
- Equijoins C-28
- Retrieving Records with Equijoins C-29
- Additional Search Conditions Using the AND and WHERE Operators C-30
- Retrieving Records with Nonequijoins C-31
- Retrieving Records by Using the USING Clause C-32

- Retrieving Records by Using the ON Clause C-33
- Left Outer Join C-34
- Right Outer Join C-35
- Full Outer Join C-36
- Self-Join: Example C-37
- Cross Join C-38
- Summary C-39
- D REF Cursors
- Cursor Variables D-2
- Using Cursor Variables D-3
- Defining REF CURSOR Types D-4
- Using the OPEN-FOR, FETCH, and CLOSE Statements D-7
- Example of Fetching D-10
- E Managing PL/SQL Code
- Objectives E-2
- Agenda E-3
- Conditional Compilation E-4
- How Does Conditional Compilation Work? E-5
- Using Selection Directives E-6
- Using Predefined and User-Defined Inquiry Directives E-7
- The PLSQL_CCFLAGS Parameter and the Inquiry Directive E-8
- Displaying the PLSQL_CCFLAGS Initialization Parameter Setting E-9
- The PLSQL_CCFLAGS Parameter and the Inquiry Directive: Example E-10

Using Conditional Compilation Error Directives to Raise User-Defined Errors E-11

- Using Static Expressions with Conditional Compilation E-12
- DBMS_DB_VERSION Package: Boolean Constants E-13
- DBMS_DB_VERSION Package Constants E-14
- Using Conditional Compilation with Database Versions: Example E-15
- Using DBMS_PREPROCESSOR Procedures to Print or Retrieve Source Text E-17
- Agenda E-18
- Obfuscation E-19
- Benefits of Obfuscating E-20
- What’s New in Dynamic Obfuscating Since Oracle 10g? E-21
- Nonobfuscated PL/SQL Code: Example E-22
- Obfuscated PL/SQL Code: Example E-23
- Dynamic Obfuscation: Example E-24
- PL/SQL Wrapper Utility E-25
- Running the PL/SQL Wrapper Utility E-26
- Results of Wrapping E-27
- Guidelines for Wrapping E-28
- DBMS_DDL Package Versus wrap Utility E-29
- Summary E-30
- F Implementing Triggers
- Objectives F-2
- Controlling Security Within the Server F-3
- Controlling Security with a Database Trigger F-4

- Enforcing Data Integrity Within the Server F-5
- Protecting Data Integrity with a Trigger F-6
- Enforcing Referential Integrity Within the Server F-7
- Protecting Referential Integrity with a Trigger F-8
- Replicating a Table Within the Server F-9
- Replicating a Table with a Trigger F-10
- Computing Derived Data Within the Server F-11
- Computing Derived Values with a Trigger F-12
- Logging Events with a Trigger F-13
- Summary F-15
- G Using the DBMS_SCHEDULER and HTP Packages
- Objectives G-2
- Generating Webpages with the HTP Package G-3
- Using the HTP Package Procedures G-4
- Creating an HTML File with SQL*Plus G-5
- The DBMS_SCHEDULER Package G-6
- Creating a Job G-8
- Creating a Job with Inline Parameters G-9
- Creating a Job Using a Program G-10
- Creating a Job for a Program with Arguments G-11
- Creating a Job Using a Schedule G-12
- Setting the Repeat Interval for a Job G-13
- Creating a Job Using a Named Program and Schedule G-14
- Managing Jobs G-15
- Data Dictionary Views G-16
- Summary G-17