

PLSQL CONTENTS(CORE+ADVANCE)

1. Introduction to PL/SQL

Objectives

What is PL/SQL?

About PL/SQL

PL/SQL Environment

Benefits of PL/SQL

PL/SQL Block Structure

Block Types

Programs Constructs

PL/SQL Programming Environments

iSQL*Plus Architecture

Create an Anonymous Block

Execute an Anonymous Block

Test the Output of a PL/SQL Block

Summary

Practice 1: Overview

2. Declaring PL/SQL Variables

Objectives

Use of Variables

Identifiers

Handling Variables in PL/SQL

Declaring and Initializing PL/SQL Variables

Delimiters in String Literals

Types of Variables

Guidelines for Declaring and Initializing PL/SQL Variables

Guidelines for Declaring PL/SQL Variables

Scalar Data Types

Base Scalar Data Types

BINARY_FLOAT and BINARY_DOUBLE

Declaring Scalar Variables

%TYPE Attribute

Declaring Variables with the %TYPE Attribute

Declaring Boolean Variables

- Bind Variables
- Printing Bind Variables
- Substitution Variables
- Prompt for Substitution Variables
- Using DEFINE for User Variable
- Composite Data Types
- LOB Data Type Variables
- Summary
- Practice 2: Overview

3. Writing Executable Statements

- Objectives
- 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
- Data Type Conversion
- Nested Blocks
- Variable Scope and Visibility
- Qualify an Identifier
- Determining Variable Scope
- Operators in PL/SQL
- Programming Guidelines
- Indenting Code

4. Interacting with the Oracle Server

- SQL Statements in PL/SQL
- SELECT Statements in PL/SQL
- Retrieving Data in PL/SQL
- Naming Conventions
- Manipulating Data Using PL/SQL
- Inserting Data
- Updating Data
- Deleting Data
- Merging Rows
- SQL Cursor
- SQL Cursor Attributes for Implicit Cursors

5. CONTROL STRUCTURES

Objectives

Controlling Flow of Execution

IF Statements

Simple IF Statement

IF THEN ELSE Statement

IF ELSIF ELSE Clause

NULL Values in IF Statements

CASE Expressions

CASE Expressions: Example

Searched CASE Expressions

CASE Statement

Handling Nulls

Logic Tables

Boolean Conditions

Iterative Control: LOOP Statements

Basic Loops

WHILE Loops

FOR Loops

Guidelines for Loops

Nested Loops and Labels

Summary

Practice 5: Overview

6. Working with Composite Data Types

Objectives

Composite Data Types

PL/SQL Records

Creating a PL/SQL Record

PL/SQL Record Structure

%ROWTYPE Attribute

Advantages of Using %ROWTYPE

%ROWTYPE Attribute

Inserting a Record by Using %ROWTYPE

Updating a Row in a Table by Using a Record

INDEX BY Tables or Associative Arrays

- Creating an INDEX BY Table
- INDEX BY Table Structure
- Creating an INDEX BY Table
- Using INDEX BY Table Methods
- INDEX BY Table of Records
- INDEX BY Table of Records: Example
- Nested Tables
- VARRAY

7. Using Explicit Cursors

- Objectives
- Cursors
- Explicit Cursor Operations
- Controlling Explicit Cursors
- 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
- Cursors with Parameters
- FOR UPDATE Clause
- WHERE CURRENT OF Clause
- Cursors with Subqueries
- Summary
- Practice 7: Overview

8. Handling Exceptions

- Objectives
- Example of an Exception
- Handling Exceptions with PL/SQL
- Handling Exceptions
- Exception Types

- Trapping Exceptions
- Guidelines for Trapping Exceptions
- Trapping Predefined Oracle Server Errors
- Trapping Non-Predefined Oracle Server Errors
- Non-Predefined Error
- Functions for Trapping Exceptions
- Trapping User-Defined Exceptions
- Calling Environments
- Propagating Exceptions in a Subblock
- RAISE_APPLICATION_ERROR Procedure

9. Creating Stored Procedures and Functions

- Objectives
- Procedures and Functions
- Differences Between Anonymous Blocks and Subprograms
- Procedure: Syntax
- Procedure: Example
- Invoking the Procedure
- Function: Syntax
- Function: Example
- Invoking the Function
- Passing Parameter to the Function
- Invoking the Function with a Parameter
- Summary

10. Creating Stored Procedures

- What Is a Procedure?
- Syntax for Creating Procedures
- Developing Procedures
- What Are Parameters?
- Formal and Actual Parameters
- Procedural Parameter Modes
- Using IN Parameters: Example
- Using OUT Parameters: Example
- Viewing OUT Parameters with iSQL*Plus
- Calling PL/SQL Using Host Variables

Using IN OUT Parameters: Example
Syntax for Passing Parameters
Parameter Passing: Examples
Using the DEFAULT Option for Parameters
Summary of Parameter Modes
Invoking Procedures
Handled Exceptions
Handled Exceptions: Example
Exceptions Not Handled
Exceptions Not Handled: Example
Removing Procedures
Viewing Procedures in the Data Dictionary
Benefits of Subprograms

11. Creating Stored Functions

Overview of Stored Functions
Syntax for Creating Functions
Developing Functions
Stored Function: Example
Ways to Execute Functions
Advantages of User-Defined Functions in SQL Statements
Function in SQL Expressions: Example
Locations to Call User-Defined Functions
Restrictions on Calling Functions from SQL Expressions
Controlling Side Effects When Calling Functions from SQL Expressions
Restrictions on Calling Functions from SQL: Example

Removing Functions
Viewing Functions in the Data Dictionary
Procedures Versus Functions

12. Creating Packages

Objectives
PL/SQL Packages: Overview

Components of a PL/SQL Package
Visibility of Package Components
Developing PL/SQL Packages
Creating the Package Specification
Example of Package Specification: comm_pkg
Creating the Package Body
Example of Package Body: comm_pkg
Invoking Package Subprograms
Creating and Using Bodiless Packages
Removing Packages
Viewing Packages in the Data Dictionary
Guidelines for Writing Packages
Advantages of Using Packages

13. Using More Package Concepts

Overloading Subprograms
Overloading: Example
Overloading and the STANDARD Package
Using Forward Declarations
Package Initialization Block
Using Package Functions in SQL and Restrictions
Package Function in SQL: Example
Persistent State of Packages
Persistent State of Package Variables: Example
Persistent State of a Package Cursor
Executing CURS_PKG
Using PL/SQL Tables of Records in Packages
PL/SQL Wrapper
Running the Wrapper
Results of Wrapping
Guidelines for Wrapping 4-21
Summary 4-22
Practice 4: Overview 4-23

14. Using Oracle-Supplied Packages in Application Development

Objectives
Using Oracle-Supplied Packages

List of Some Oracle-Supplied Packages
How the DBMS_OUTPUT Package Works
Interacting with Operating System Files
File Processing Using the UTL_FILE Package
Exceptions in the UTL_FILE Package
Using UTL_FILE: Example
Generating Web Pages with the HTP Package
Using the HTP Package Procedures
Creating an HTML File with iSQL*Plus
Using UTL_MAIL
Installing and Using UTL_MAIL
Sending E-Mail with a Binary Attachment
Sending E-Mail with a Text Attachment
DBMS_SCHEDULER Package
Creating a Job
Creating a Job with In-Line Parameters
Creating a Job Using a Program
Creating a Job for a Program with Arguments
Creating a Job Using a Schedule
Setting the Repeat Interval for a Job
Creating a Job Using a Named Program and Schedule
Managing Jobs
Data Dictionary Views

15. Design Considerations for PL/SQL Code

Objectives
Standardizing Constants and Exceptions
Standardizing Exceptions
Standardizing Exception Handling
Standardizing Constants
Local Subprograms
Definer's Rights Versus Invoker's Rights
Specifying Invoker's Rights
Autonomous Transactions
Features of Autonomous Transactions
Using Autonomous Transactions
RETURNING Clause
Bulk Binding
Using Bulk Binding

- Bulk Binding FORALL: Example
- Using BULK COLLECT INTO with Queries
- Using BULK COLLECT INTO with Cursors
- Using BULK COLLECT INTO with a RETURNING Clause
- Using the NOCOPY Hint
- Effects of the NOCOPY Hint
- NOCOPY Hint Can Be Ignored
- PARALLEL_ENABLE Hint

16. Manipulating Large Objects

- Objectives
- What Is a LOB?
- Contrasting LONG and LOB Data Types
- Anatomy of a LOB
- Internal LOBs
- Managing Internal LOBs
- What Are BFILES?
- A New Database Object: DIRECTORY
- Guidelines for Creating DIRECTORY Objects
- Managing BFILES
- Preparing to Use BFILES
- Populating BFILE Columns with SQL
- Populating a BFILE Column with PL/SQL
- Using DBMS_LOB Routines with BFILES
- Migrating from LONG to LOB
- DBMS_LOB Package
- DBMS_LOB.READ and DBMS_LOB.WRITE
- Initializing LOB Columns Added to a Table
- Populating LOB Columns
- Updating LOB by Using DBMS_LOB in PL/SQL
- Selecting CLOB Values by Using SQL
- Selecting CLOB Values by Using DBMS_LOB
- Selecting CLOB Values in PL/SQL
- Removing LOBs
- Temporary LOBs
- Creating a Temporary LOB

17 Creating Triggers

Types of Triggers

Guidelines for Designing Triggers

Creating DML Triggers

Types of DML Triggers

Trigger Timing

Trigger-Firing Sequence

Trigger Event Types and Body

Creating a DML Statement Trigger

Testing SECURE_EMP

Using Conditional Predicates

Creating a DML Row Trigger

Using OLD and NEW Qualifiers

Using OLD and NEW Qualifiers: Example Using AUDIT_EMP

Restricting a Row Trigger: Example

Summary of the Trigger Execution Model

Implementing an Integrity Constraint with a Trigger

INSTEAD OF Triggers

Creating an INSTEAD OF Trigger

Comparison of Database Triggers and Stored Procedures

Comparison of Database Triggers and Oracle Forms Triggers

Managing Triggers

Removing Triggers

Testing Triggers

19. Applications for Triggers

Creating Database Triggers

Creating Triggers on DDL Statements

Creating Triggers on System Events

LOGON and LOGOFF Triggers: Example CALL Statements

Reading Data from a Mutating Table

Mutating Table: Example

Benefits of Database Triggers

Managing Triggers

Business Application Scenarios for Implementing Triggers

Viewing Trigger Information

Using USER_TRIGGERS

Listing the Code of Triggers