

Oracle 12c PL/SQL Programming

Course Summary

Description

PL/SQL allows developers to extend the basic data query and manipulation of SQL into complete applications and shared program units. This class provides the technical expertise necessary to utilize this powerful component of Oracle.

This class combines the content of the Oracle 12c PL/SQL Introduction and Oracle 12c PL/SQL Program Units classes.

The content of this course applies to developing PL/SQL blocks for stand-alone use, to be embedded in other programming environments (Java, C, C#, etc.) or to augment the functionality of Oracle Application Express (APEX) applications or in Oracle Forms or Reports.

One of the most powerful features of the Oracle RDBMS product is the ability to create named 'blocks' of PL/SQL code that are stored within the database. This allows for robust solutions to be developed that can be shared and re-used. The four types of these program units: procedures, functions, packages and triggers are each covered in depth. Students will learn to write, debug and manage each.

Students use the Oracle PL/SQL IDE, SQL Developer, throughout the course as the presentation tool and as their development environment. Features of this tool that enhance productivity along with its integral source level debugger are presented.

The performance of PL/SQL is highly dependent on the selection of suitable data types for local variables. In particular computationally intensive PL/SQL is highly influenced by the numeric types that are used. The differences in and the costs and benefits of the various PL/SQL numeric types are compared.

PL/SQL supports quite complex composite datatypes. Understanding when such a variable is called for and understanding how to choose among them requires in depth knowledge of their characteristic, strengths and limitations. The implementation details of the various kinds of composites are explained carefully.

Built-in exception handling is a feature of the PL/SQL programming language that provides for commonality in the approach to handle any sort of run-time errors, both those due to Oracle errors (for example, duplicate primary key indexes) and application specific (customer spending over a pre-assigned credit limit).

Oracle provides dozens of PL/SQL packages with the core RDBMS product. Utilization of some samples of these presupplied PL/SQL packages is covered allowing developers to accomplish sophisticated tasks such as job scheduling, interprocess communication and utilizing Oracle's FLASHBACK capabilities.

The content of this course is applicable to Oracle 12c with relevance to versions back to 10g.

This class provides preparation for the OCP Program with PL/SQL 1Z0-147 and Oracle Database: Program with PL/SQL 1Z0-144 tests. Oracle Corporation suggests that candidates also have six months of hands-on experience before testing.

Oracle 12c PL/SQL Programming

Course Summary (cont'd)

Topics

- PL/SQL language fundamentals
- Creating anonymous PL/SQL blocks using SQL Developer
- The Petsaver database
- PL/SQL scalar variables
- SELECT statements in PL/SQL
- DML statements in PL/SQL
- Transaction control in PL/SQL
- The SQL Developer interface
- PL/SQL control structures
- PL/SQL composite variables
- PL/SQL cursors
- PL/SQL block hierarchies
- PL/SQL exception handling
- Advanced exception handling
- Basic Oracle supplied packages
- Optimizing PL/SQL performance
- Creating stored procedures
- Writing stored procedures using SQL Developer
- Creating stored functions
- Advanced features of procedures and functions
- 12c PL/SQL new features
- Debugging PL/SQL with SQL Developer
- Creating packages
- Writing packages with SQL Developer
- Advanced features of packages
- Creating database triggers
- Advanced features of triggers
- Maintaining program units
- Dynamic SQL in PL/SQL

Audience

This course is ideal for:

- Oracle SQL developers and DBAs
- Oracle PL/SQL developers and DBAs wishing to 'back fill' gaps in their expertise
- Oracle APEX, Forms and Reports developers
- Technical managers needing Oracle expertise for project administration

Prerequisites

Prior to taking this course, students should have skill with GUI interfaces and data processing background. A basic understanding of SQL is required to succeed in this class. The SQL used in this class is as simple as possible, but persons with absolutely no SQL experience will likely experience difficulty

Duration

Five days

Oracle 12c PL/SQL Programming

Course Outline

- I. PL/SQL language fundamentals**
 - A. What is PL/SQL?
 - B. Reasons to use PL/SQL
 - C. PL/SQL block structure
 - D. Types of PL/SQL blocks
 - E. Using DBMS_OUTPUT
- II. Creating anonymous blocks using SQL Developer**
 - A. The SQL Developer SQL Worksheet
 - B. Coding an anonymous block
 - C. Running PL/SQL blocks in SQL Developer
 - D. Controlling the DBMS_OUTPUT pane
- III. The Petsaver database**
 - A. The petsaver story
 - B. Important petsaver tables
 - C. Petsaver ERD
- IV. PL/SQL scalar variables**
 - A. Defining scalar variables
 - B. Anchoring to database definitions
 - C. Assigning defaults
 - D. PL/SQL value assignment
- V. SELECT statements in PL/SQL**
 - A. Single row queries
 - B. The INTO clause
 - C. Avoiding SELECT errors
- VI. DML statements in PL/SQL**
 - A. INSERT
 - B. UPDATE
 - C. MERGE
 - D. DELETE
- VII. Transaction control in PL/SQL**
 - A. COMMIT
 - B. ROLLBACK
 - C. SAVEPOINT
- VIII. The SQL Developer interface**
 - A. The Connection Navigator
 - B. Moving, pinning, closing and restoring SQL Developer panes
- IX. PL/SQL control structures**
 - A. IF THEN ELSE
 - B. LOOP
 - C. CASE
- X. PL/SQL composite variables**
 - A. Records
 - B. Associative arrays (PL/SQL tables)
 - C. Nested tables
 - D. VARRAYs
- XI. PL/SQL cursors**
 - A. Defining cursors
 - B. Cursor attributes
 - C. Cursor FOR loops
 - D. Using DML with cursors
- XII. PL/SQL block hierarchies**
 - A. Variable scope
 - B. Creating sub-blocks
 - C. Referencing block labels
- XIII. PL/SQL exception handling**
 - A. Defining exceptions
 - B. Predefined exceptions
 - C. The EXCEPTION data type
 - D. Handling OTHER Oracle exceptions
 - E. User-defined exceptions
- XIV. Advanced exception handling**
 - A. Capturing unexpected exceptions
 - B. Error-related functions
 - C. Continuing cursors after errors
- XV. Basic Oracle supplied packages**
 - A. UTL_FILE
 - B. DBMS_JOB
 - C. DBMS_PIPE
 - D. DBMS_LOCK
 - E. DBMS_FLASHBACK
- XVI. Optimizing PL/SQL performance**
 - A. The RETURN clause
 - B. Benefits of BULK operations
 - C. Using the hierarchical profiler in SQL Developer
 - D. The NOCOPY hint
 - E. Forcing memory residence using DBMS_SHARED_POOL

Oracle 12c PL/SQL Programming

Course Outline (cont'd)

XVII. Creating stored procedures

- A. Overview of procedures
- B. Client-side vs. server-side procedures
- C. Invoking procedures in SQL and PL/SQL
- D. Resolving compile errors
- E. Removing a procedure

XVIII. Writing stored procedures using SQL Developer

- A. Using the SQL Developer Create Procedure wizard
- B. Running stored procedures using SQL Developer
- C. Modifying the SQL Developer procedure testing stub

XIX. Creating stored functions

- A. Overview of functions
- B. Client-side vs. server-side functions
- C. Using the RETURN clause
- D. Invoking functions in SQL and PL/SQL
- E. Procedures and functions compared

XX. Advanced features of procedures and functions

- A. Advanced parameter topics
- B. Local modules
- C. DETERMINISTIC functions
- D. Autonomous transactions
- E. Controlling transaction visibility
- F. Definers vs. Invokers rights models

XXI. 12c PL/SQL new features

- A. Enhancements to numeric data types
- B. Enhancements to the LOOP syntax
- C. Referencing the PL/SQL result cache

XXII. Debugging PL/SQL using SQL Developer

- A. Compiling for debug
- B. Setting breakpoints

- C. Conditional breakpoints
- D. Creating a watch list

XXIII. Creating packages

- A. Package overview
- B. Creating the package specification
- C. Creating the package body
- D. Public vs. private package constructs

XXIV. Writing packages with SQL Developer

- A. Using the SQL Developer Create Package wizard
- B. Running package components in SQL Developer

XXV. Advanced features of packages

- A. Overloading
- B. Forward referencing
- C. Using 'start up' code

XXVI. Creating database triggers

- A. Trigger overview
- B. Specifying triggering events
- C. What to do about "mutating" tables

XXVII. Advanced features of triggers

- A. INSTEAD OF triggers
- B. DML trigger directives
- C. Calling procedures from triggers
- D. Autonomous triggers
- E. Compound triggers

XXVIII. Maintaining program units

- A. Program units in the data dictionary
- B. Reviewing program unit dependencies
- C. Measuring application maintenance impact analysis

XXIX. Dynamic SQL in PL/SQL

- A. Native dynamic SQL
- B. Using DBMS_SQL
- C. Cursor variables (REF cursors)