



Oracle Database 12c: PL/SQL Fundamentals

The Oracle Database: PL/SQL Fundamentals Ed 2 course introduces students to the PL/SQL programming language. In the class, students learn to create PL/SQL blocks of application code.

Learn To

The Oracle Database: PL/SQL Fundamentals Ed 2 course introduces students to the PL/SQL programming language. In the class, students learn to create PL/SQL blocks of application code.

Learn To:

- Use various features of PL/SQL.
- Write anonymous blocks of code in PL/SQL.
- Use various PL/SQL conditional constructs and loops in program blocks.
- Interface the PL/SQL code with the database.
- Use stored procedures and functions in their program blocks.
- Handle Exceptions in PL/SQL code.

Benefits To You

Students will benefit from this course as it enables programmers with the skill of using this programming language. In the class students will learn to create anonymous PL/SQL blocks, to create procedures, functions and triggers. Students will learn to use explicit cursors in their program blocks. Demonstrations and hands-on practice reinforce the fundamental concepts. Students use Oracle SQL Developer to develop these program units. SQL*Plus is introduced as optional tools.

Prerequisites

Audience

- Analyst
- Developer
- Implementer

Course Objectives

- Design PL/SQL anonymous block that execute efficiently
- Use PL/SQL programming constructs and conditionally control code flow (loops, control structures, and explicit cursors)

- Handle exceptions in the program units
- Create and execute simple stored procedures and functions
- Write PL/SQL code to interface with the database
- Describe the features and syntax of PL/SQL

Course Topics

Introduction

- Course Objectives, Course Agenda and Class Account Information
- The Human Resources (HR) Schema
- Appendices Used in this Course
- PL/SQL Development Environments

Introduction to PL/SQL

- Understanding the benefits and structure of PL/SQL
- Understanding PL/SQL Blocks
- Generating output messages in PL/SQL

Declaring PL/SQL Variables

- Identifying valid and invalid identifiers
- Declaring and initializing variables
- Various data types
- Identifying the benefits of using the %TYPE attribute
- Using bind variables

Writing Anonymous PL/SQL blocks

- Lexical Units in a PL/SQL Block
- Using SQL Functions in PL/SQL
- Data Type Conversion
- Using Nested Blocks as Statements
- Referencing an Identifier Value in a Nested Block
- Qualifying an Identifier with a Label
- Operators in PL/SQL
- Using Sequences in PL/SQL Expressions

Using SQL Statements within a PL/SQL Block

- Using SQL Statements in PL/SQL

- Retrieving Data in PL/SQL with the SELECT statement
- Using Naming Conventions in DML Statements and data retrieval
- Manipulating Data on the Server Using PL/SQL
- SQL Cursors
- Using SQL Cursor Attributes to Obtain Feedback on DML
- Saving and Discarding Transactions

Writing Control Structures

- Controlling PL/SQL Flow of Execution
- Using IF and CASE Statements for conditional processing
- Handling Nulls
- Building Boolean Conditions with Logical Operators
- Using Iterative Control with Loop Statements

Working with Composite Data Types

- Composite Data Types of PL/SQL Records and Tables
- Using PL/SQL Records
- Inserting and Updating PL/SQL Records
- Using INDEX BY Tables
- Using Associative arrays
- Declaring and Using VArrays

Using Explicit Cursors

- Cursors
- Explicit Cursor Operations
- Controlling Explicit Cursors
- Using Explicit Cursors to Process Rows
- Cursors and Records
- Cursor FOR Loops Using Subqueries
- Explicit Cursor Attributes
- %NOTFOUND and %ROWCOUNT Attributes

Handling Exceptions

- Concept of Exception
- Handling Exceptions with PL/SQL

- Predefined Exceptions
- Trapping Predefined and Non-predefined Oracle Server Errors
- Functions that Return Information on Exceptions
- Usage of PRAGMA keyword
- Trapping User-Defined Exceptions
- Propagating Exceptions

Creating Stored Procedures and Functions

- Overview of Stored Procedures and Functions
- Differentiating between anonymous blocks and subprograms
- CREATE OR REPLACE PROCEDURE | FUNCTION
- Understanding the Header Area of a Stored Procedure and Function
- Creating Simple Procedures and Functions
- Creating a Simple Procedure with an IN Parameter
- Executing a Procedure and a Function