

# Oracle Database 12c R2: Develop PL SQL Program Units

Learn via: **Classroom / Virtual Classroom / Online**

Duration: **3 Gün**

## Overview

**This Oracle Database: Develop PL/SQL Program Units course is designed for developers with basic PL/SQL and SQL language skills. You will learn to develop, execute and manage PL/SQL stored program units, which include: procedures, functions, packages and database triggers.**

### Learn To:

- Create, and execute stored procedures and functions.
- Design and use PL/SQL packages.
- Create overloaded package subprograms for more flexibility.
- Utilize Oracle supplied packages in application development.
- Create triggers to solve business challenges.
- Build and execute SQL statements dynamically.
- Manage PL/SQL subprograms and triggers.
- Understand and influence the PL/SQL compiler.
- Manage dependencies.

### Benefits to You:

Ensure fast, reliable, secure and easy to manage performance. Optimize database workloads, lower IT costs and deliver a higher quality of service by enabling consolidation onto database clouds.

### Learn Dynamic SQL, Design Considerations and More:

This course will also teach you how to use Dynamic SQL through instruction, as well as hands-on exercises. Expert Oracle instructors will also help you understand design considerations when coding using PL/SQL.

### Using Oracle SQL Developer:

In addition, you'll use Oracle SQL Developer as the main environment tool to develop these program units. SQL\*Plus is introduced as optional tools. Demonstrations and hands-on practice reinforce the fundamental concepts you've learned throughout the course.

## Prerequisites

### Required:

- Basic Knowledge of PL/SQL
- Familiarity with programming languages
- Oracle Database: Introduction to SQL
- Oracle Database: PL/SQL Fundamentals

### Required knowledge:

- Block Structure, Variables, IF constructs, Loops, Composite Data Types, Cursors, Exception Handling

### Recommended:

- Oracle SQL Tuning for Developers Workshop

## What You Will Learn

At the end of this course you will be able to:

- Create, use, and debug stored procedures and functions
- Design and use PL/SQL packages to group and contain related constructs
- Create overloaded package subprograms for more flexibility
- Use the Oracle supplied PL/SQL packages to generate screen output, file output, and mail output
- 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
- Use conditional PL/SQL compilation and obfuscate (hide) code
- Create triggers to solve business challenges
- Manage dependencies between PL/SQL subprograms

## **Outline**

### **Introduction**

- Course Objectives, Course Agenda and Appendixes Used in this Course
- Describe the full Human Resources (HR) Schema
- Review the online Oracle Database 12c SQL and PL/SQL documentation and the additional available resources
- List the PL/SQL development environments Available in this course
- Use the SQL Worksheet
- Execute SQL Statements
- Work With Script Files
- Create and Execute Anonymous Blocks

### **Creating Stored Procedures**

- Describe PL/SQL blocks and subprograms
- Describe the uses and benefits of procedures
- Create, call, and remove procedures
- Use formal and actual parameters
- Identify the available parameter-passing modes
- Pass parameters using the positional, named, or combination techniques
- Handle exceptions in procedures
- View the procedure information

### **Creating Functions and Debugging Subprograms**

- Creating Stored Functions
- The Difference Between Procedures and Functions
- Developing Functions
- Creating and Executing and Removing Functions
- Identifying the Advantages of Using Stored Functions in SQL Statements
- Using User-Defined Functions in SQL Statements
- Using a PL/SQL Function in the SQL WITH Clause
- Restrictions When Calling Functions from SQL statements

### **Creating Packages**

- Using PL/SQL Packages
- The Components of a PL/SQL Package
- The Visibility of a Package's Components
- Developing a PL/SQL Package
- Creating the Package Specification and Package Body
- Invoking the Package Constructs
- Creating and Using Bodiless Packages
- Removing a Package

### **Working With Packages**

- Overloading Subprograms
- Using Forward Declarations to Solve Illegal Procedure Reference
- Initializing Packages
- Using Package Functions in SQL and Restrictions
- Controlling Side Effects of PL/SQL Subprograms
- Persistent State of Packages
- Persistent State of Package Variables and Cursors
- Using PL/SQL Tables of Records in Packages

### **Using Oracle-Supplied Packages in Application Development**

- Using Oracle-Supplied Packages
- Examples of Some of the Oracle-Supplied Packages
- How Does the DBMS\_OUTPUT Package Work?
- Using the UTL\_FILE Package to Interact With Operating System Files
- Using the UTL\_MAIL Package

## Using Dynamic SQL

- The Execution Flow of SQL
- Working With Dynamic SQL
- When Do You Need Dynamic SQL?
- Using Native Dynamic SQL (NDS)
- Declaring Cursor Variables
- Executing a PL/SQL Block Dynamically
- Using Native Dynamic SQL to Compile PL/SQL Code

## Design Considerations for PL/SQL Code

- Standardize constants with a constant package
- Standardize exceptions with an exception package
- Write PL/SQL code that uses local subprograms
- Grant Roles to PL/SQL Packages and Standalone Stored Subprograms
- Use the NOCOPY compiler hint to pass parameters by reference
- Use the PARALLEL ENABLE hint for optimization
- Use the AUTONOMOUS TRANSACTION pragma to run independent transactions within a single transactio
- Describe the differences between invoker rights and definer rights

## Creating Triggers

- Describe different types of triggers
- Describe database triggers and their use
- Create database triggers
- Describe database trigger firing rules
- Remove database triggers

## Creating Compound, DDL, and Event Database Triggers

- Describe compound triggers
- Describe mutating tables
- Create triggers on DDL statements
- Create triggers on system events
- Display information about triggers

## Using PL/SQL compiler

- Using the PL/SQL Compiler Using the Initialization Parameters for PL/SQL Compilation
- Using the PL/SQL Compile Time Warnings
- Viewing the Current Setting of PLSQL\_WARNINGS
- Viewing the Compiler Warnings: Using SQL Developer, SQL\*Plus, or the Data Dictionary Views
- Guidelines for Using PLSQL\_WARNINGS

## Managing Dependencies

- Describe dependent and referenced objects
- Track procedural dependencies with dictionary views
- Predict the effect of changing a database object upon stored procedures and functions
- Manage local and remote procedural dependencies