

€ TRAINING

ORACLE PL/SQL Fundamentals Training Course



16 - 20 December 2024
Casablanca (Morocco)
New Hotel



ORACLE PL/SQL Fundamentals Training Course

REF: B1614 DATE: 16 - 20 December 2024 Venue: Casablanca (Morocco) - New Hotel Fee: 4290 Euro

Introduction:

This 5-day course gives an introduction to ORACLE PL/SQL, an application development environment that enables the writing of stored procedures, functions, and triggers using both SQL and PL/SQL commands. The course takes the format of a workshop, with a mix of lectures, working examples, and practical exercises.

Course Objectives:

At the end of this course, the participants will be able to:

- Handle runtime errors
- Describe stored procedures and functions
- Use cursors to process rows
- Design PL/SQL anonymous block that executes efficiently
- Describe the features and syntax of PL/SQL,
- Use PL/SQL programming constructs and conditionally control code flow loops, control structures, and explicit cursors
- Check index fragmentation and maintain them.

Targeted Audience:

- Anyone who wants to start a career in SQL database administration
- Anyone who wants to have complete SQL database training to get a job in IT
- Anyone who wants to advance his/her career
- Anyone who wants to master the SQL command-line skills
- IT professionals

Course Outlines:

Unit 1: Introduction

- Aims and Objectives
- Course Schedule
- Introductions
- Pre-requisites
- Responsibilities

SQL Tools

- Objectives
- SQL Developer
- SQL Developer - Connection
- Viewing Table Information
- Using SQL, SQL Developer - Query
- SQL*Plus Login
- Direct Connection

- Using SQL*Plus
- Ending the Session
- SQL*Plus Commands
- SQL*Plus Environment
- SQL*Plus Prompt
- Finding Information about Tables
- Getting Help
- Using SQL Files
- iSQL*Plus, Entity Models
- The ORDERS Tables
- The FILM Tables
- Course Tables Handout
- SQL Statement Syntax
- SQL*Plus Commands

Unit 2: What is PL/SQL?

- What is PL/SQL?
- Why Use PL/SQL?
- Block Structure
- Displaying a Message
- Sample Code
- Setting SERVER OUTPUT
- Update Example, Style Guide

Variables

- Variables
- Datatypes
- Setting Variables
- Constants
- Local and Global Variables
- %Type Variables
- Substitution Variables
- Comments with &
- Verify Option
- && Variables
- Define and Undefine

Unit 3: SELECT Statement

- SELECT Statement
- Populating Variables
- %Rowtype Variables
- CHR Function
- Self Study
- PL/SQL Records
- Example Declarations

Conditional Statement

- IF Statement
- SELECT Statement
- Self Study
- Case Statement

Trapping Errors

- Exception
- Internal Errors
- Error Code and Message
- Using No Data Found
- User Exceptions
- Raise Application Error
- Trapping Non-defined Errors
- Using PRAGMA EXCEPTION_INIT
- Commit and Rollback
- Self Study
- Nested Blocks
- Workshop

Unit 4: Iteration - Looping

- Loop Statement
- While Statement
- For Statement
- Goto Statement and Labels

Cursors

- Cursors
- Cursor Attributes
- Explicit Cursors
- Explicit Cursor Example
- Declaring the Cursor
- Declaring the Variable
- Open, Fetching the First Row
- Fetching the Next Row
- Exit When %Notfound
- Close
- For Loop I
- For Loop II
- Update Example
- FOR UPDATE
- FOR UPDATE OF
- WHERE CURRENT OF
- Commit with Cursors
- Validation Example I
- Validation Example II
- Cursor Parameters,
- Workshop
- Workshop Solution

Procedures, Functions and Packages

- Create Statement
- Parameters
- Procedure Body
- Showing Errors
- Describe a Procedure
- Calling Procedures
- Calling Procedures in SQL*Plus
- Using Output Parameters
- Calling with Output Parameters
- Creating Functions
- Example Function
- Showing Errors
- Describe a Function
- Calling Functions
- Calling Functions in SQL*Plus
- Modular Programming
- Example Procedure
- Calling Functions
- Calling Functions In An IF Statement
- Creating Packages
- Package Example
- Reasons for Packages
- Public and Private Sub-programs
- Showing Errors
- Describe a Package
- Calling Packages in SQL*Plus
- Calling Packages From Sub-Programs
- Dropping a Sub-Program
- Finding Sub-programs
- Creating a Debug Package
- Calling the Debug Package
- Positional and Named Notation
- Parameter Default Values
- Recompiling Procedures and Functions
- Workshop

Unit 5: Triggers

- Creating Triggers
- Statement Triggers
- Row Level Triggers
- WHEN Restriction
- Selective Triggers - IF
- Showing Errors
- Commit in Triggers
- Restrictions
- Mutating Triggers
- Finding Triggers
- Dropping a Trigger
- Generating an Auto-number

- Disabling Triggers
- Enabling Triggers
- Trigger Names

Sample Data

- ORDER Tables
- FILM Tables
- EMPLOYEE Tables

Dynamic SQL

- SQL in PL/SQL
- Binding
- Dynamic SQL
- Native Dynamic SQL
- DDL and DML
- DBMS_SQL Package
- Dynamic SQL - SELECT
- Dynamic SQL - SELECT Procedure

Using Files

- Using Text Files
- UTL_FILE Package
- Write/Append Example
- Read Example
- Trigger Example
- DBMS_ALERT Packages
- DBMS_JOB Package

COLLECTIONS

- %Type Variables
- Record Variables
- Collection Types
- Index-By Tables
- Setting Values
- Nonexistent Elements
- Nested Tables
- Nested Table Initialisation
- Using the Constructor
- Adding to a Nested Table
- Varrays
- Varray Initialization
- Adding Elements to a Varray
- Multilevel Collections
- Bulk Bind
- Bulk Bind Example
- Transactional Issues
- BULK COLLECT Clause

- RETURNING INTO