

**Oracle PL/SQL Programming**

**Course Number:** ORC-332
**Duration:** 5 days

**Overview**

Accelebrate's Oracle PL/SQL training teaches attendees beginning and advanced Oracle® PL/SQL database programming skills.

**Prerequisites**

Prior Oracle SQL experience is presumed.

**Materials**

All students receive comprehensive courseware and a related textbook.

**Software Needed on Each Student PC**

* At least 8 GB RAM
* A complete installation of Oracle 19 or higher
* Oracle SQL Developer
* Please contact us for detailed setup instructions if you have purchased this class.

**Objectives**

* Understand the environment and context in which PL/SQL operates.
* Consider the advantages and benefits of PL/SQL within a database environment.
* Declare program variables and complex data types.
* Develop logic within PL/SQL program blocks.
* Fetch data from the database into program variables.
* Return program output to users.
* Handle program exceptions.
* Use explicit and implicit database cursors.
* Maintain and implement triggers.

**Outline**

* Introduction
* PL/SQL Basics
	+ The HR Schema
	+ What is PL/SQL?
	+ Blocks
	+ Outputting Information
	+ Variables and Constants
	+ Constants
	+ Data Types
	+ Naming Variables and Other Elements
	+ Embedding SQL in PL/SQL
	+ SELECT…INTO and RETURNING…INTO
	+ Using Variables (Exercise)
	+ PL/SQL Features
* Subprograms
	+ Introduction to Subprograms
	+ Procedures
	+ Variable Declarations
	+ Parameters
	+ Parameters with Default Values
	+ Parameter Modes
	+ IN Mode
	+ OUT Mode
	+ IN OUT Mode
	+ Named Notation
	+ Using SQL in a Subprogram
	+ %TYPE
	+ Creating a Procedure (Exercise)
	+ Functions
	+ Creating a Function (Exercise)
	+ Using PL/SQL Functions in SQL Queries
	+ Dropping a Subprogram
* Conditional Processing
	+ Conditions and Booleans
	+ IF-ELSIF-ELSE Conditions
	+ Creating a get\_age() Function (Exercise)
	+ ELSIF
	+ Creating a check\_rights() Procedure (Exercise)
	+ Creating an is\_manager() Function (Exercise)
	+ BOOLEAN Values and Standard SQL
	+ The CASE Statement
	+ CASE Expressions
	+ Replacing the Head Honcho (Exercise)
* Exceptions
	+ Introduction to Exceptions
	+ Predefined Exceptions
	+ The EXCEPTION Part of the Block
	+ Catching NO\_DATA\_FOUND Exception (Exercise)
	+ User-defined Exceptions
	+ User-defined Exceptions in Subprograms
	+ Re-raising Exceptions
	+ Replacing the Head Honcho (revisited) (Exercise)
	+ Adding Exceptions to update\_employee\_manager() (Exercise)
	+ Naming Unnamed Predefined Exceptions
	+ WHILE Loops
	+ When to Use Exceptions
* Cursors
	+ Implicit Cursors
	+ Using Implicit Cursor Attributes (Exercise)
	+ Explicit Cursors
	+ %ROWTYPE
	+ Explicit Cursor Use Case
	+ Cursor FOR LOOP
	+ Using an Explicit Cursor (Exercise)
	+ Cursor Parameters
* Packages
	+ Package Basics
	+ The Package Specification
	+ The Package Body
	+ Modifying the Package (Exercise)
	+ Building an Employee Package
	+ Adding a get\_manager() Function (Exercise)
	+ Overloading Subprograms
	+ Adding Overloaded Functions to the Package (Exercise)
	+ Auditing
	+ Validation Procedures
	+ Adding a Validation Procedure (Exercise)
	+ Package Cursors
	+ Adding a Cursor to the Package (Exercise)
	+ Benefits of Packages
* Triggers
	+ What are Triggers?
	+ Trigger Parts
	+ Validation Triggers
	+ Creating a Trigger on the jobs Table (Exercise)
	+ The WHEN Clause
	+ Using the WHEN Clause (Exercise)
	+ Audit Triggers
	+ Statement-level Triggers
	+ Compound Triggers
	+ Trigger Warning
* Conclusion