Focus Training Services
#301, 3rd floor, Nagasuri Plazza,
Satyam theatre Road Ameerpet
Above bank of India (BOI)

Oracle PLSQL Syllabus

Introduction
- Course Objectives
- Course Agenda
- Human Resources (HR) Schema
- Introduction to SQL Developer

Introduction to PL/SQL
- PL/SQL Overview
- Benefits of PL/SQL Subprograms
- Overview of the Types of PL/SQL blocks
- Create a Simple Anonymous Block
- Generate Output from a PL/SQL Block

PL/SQL Identifiers
- List the different Types of Identifiers in a PL/SQL subprogram
- Usage of the Declarative Section to define Identifiers
- Use variables to store data
- Identify Scalar Data Types
- The %TYPE Attribute
- What are Bind Variables?
- Sequences in PL/SQL Expressions

Write Executable Statements
- Describe Basic PL/SQL Block Syntax Guidelines
- Comment Code
- Deployment of SQL Functions in PL/SQL
- How to convert Data Types?
- Nested Blocks
- Identify the Operators in PL/SQL

Interaction with the Oracle Server
- Invoke SELECT Statements in PL/SQL to Retrieve data
- Data Manipulation in the Server Using PL/SQL
- SQL Cursor concept
- Usage of SQL Cursor Attributes to Obtain Feedback on DML
- Save and Discard Transactions

Control Structures
- Conditional processing Using IF Statements
- Conditional processing Using CASE Statements
- Use simple Loop Statement
- Use While Loop Statement
- Use For Loop Statement
- Describe the Continue Statement

Composite Data Types
- Use PL/SQL Records
- The %ROWTYPE Attribute
- Insert and Update with PL/SQL Records
- Associative Arrays (INDEX BY Tables)
- Examine INDEX BY Table Methods
- Use INDEX BY Table of Records

Explicit Cursors
- What are Explicit Cursors?
- Declare the Cursor
- Open the Cursor
- Fetch data from the Cursor
- Close the Cursor
- Cursor FOR loop
- Explicit Cursor Attributes
- FOR UPDATE Clause and WHERE CURRENT Clause

Exception Handling
- Understand Exceptions
- Handle Exceptions with PL/SQL
- Trap Predefined Oracle Server Errors
- Trap Non-Predefined Oracle Server Errors
- Trap User-Defined Exceptions
- Propagate Exceptions
- RAISE_APPLICATION_ERROR Procedure

Stored Procedures and Functions
- Understand Stored Procedures and Functions
- Differentiate between anonymous blocks and subprograms
- Create a Simple Procedure
- Create a Simple Procedure with IN parameter
- Create a Simple Function
- Execute a Simple Procedure
- Execute a Simple Function

Create Stored Procedures
- Create a Modularized and Layered Subprogram Design
- Modularize Development With PL/SQL Blocks
- Describe the PL/SQL Execution Environment
- Identity the benefits of Using PL/SQL Subprograms
- List the differences Between Anonymous Blocks and Subprograms
- Create, Call, and Remove Stored Procedures Using the CREATE Command and SQL Developer
- Implement Procedures Parameters and Parameters Modes
- View Procedures Information Using the Data Dictionary Views and SQL Developer

Create Stored Functions
- Create, Call, and Remove a Stored Function Using the CREATE Command and SQL Developer
- Identity the advantages of Using Stored Functions in SQL Statements
- List the steps to create a stored function
- Implement User-Defined Functions in SQL Statements
- Identity the restrictions when calling Functions from SQL statements
- Control Side Effects when calling Functions from SQL Expressions
- View Functions Information

Create Packages
- Identity the advantages of Packages
- Describe Packages
- List the components of a Package
- Develop a Package
- How to enable visibility of a Package€ ™s components?
- Create the Package Specification and Body Using the SQL CREATE Statement and SQL Developer
- Invoke Package Constructs
- View PL/SQL Source Code Using the Data Dictionary

Focus Training Services

Focus Training Services
#301, 3rd floor, Nagasuri Plazza,
Satyam theatre Road Ameerpet
Above bank of India (BOI)

Oracle PLSQL Course Content

Packages

- Overloading Subprograms in PL/SQL
- Use the STANDARD Package
- Use Forward Declarations to Solve Illegal Procedure Reference
- Implement Package Functions in SQL and Restrictions
- Persistent State of Packages
- Persistent State of a Package Cursor
- Control Side Effects of PL/SQL Subprograms
- Invoke PL/SQL Tables of Records in Packages

Dynamic SQL

- The Execution Flow of SQL
- What is Dynamic SQL?
- Declare Cursor Variables
- Dynamically executing a PL/SQL Block
- Configure Native Dynamic SQL to Compile PL/SQL Code
- Invoke DBMS_SQL Package
- Implement DBMS_SQL with a Parameterized DML Statement
- Dynamic SQL Functional Completeness

Triggers

- Describe Triggers
- Identify the Trigger Event Types and Body
- Business Application Scenarios for Implementing Triggers
- Create DML Triggers Using the CREATE TRIGGER Statement and SQL Developer
- Identify the Trigger Event Types, Body, and Firing (Timing)
- Statement Level Triggers Versus Row Level Triggers
- Create Instead of and Disabled Triggers
- How to Manage, Test, and Remove Triggers?

Designing PL/SQL Code

- Describe the predefined data types
- Create subtypes based on existing types for an application
- List the different guidelines for cursor design
- Cursor variables

Using Collections

- Overview of collections
- Use Associative arrays
- Use Nested tables
- Use VARRAYs
- Compare nested tables and VARRAYs
- Write PL/SQL programs that use collections
- Use Collections effectively

Focus Training Services