

## Oracle PL-SQL Fundamentals for Oracle 11g

**Duration: 30 hrs**

### Audience:

- Data Warehouse Administrator
- Forms Developer
- System Analysts
- Business Analysts
- Developer
- Application Developers
- PL/SQL Developer

### Suggested Prerequisites:

- None

#### Introduction to PLSQL

- History of PL/SQL language
- The anonymous PLSQL block
- Variables & Data types
- PLSQL operators, Nested blocks
- Implicit data conversion
- SELECT ... INTO statement
- DBMS\_OUTPUT.PUT\_LINE

#### Iterative and Conditional control structures

- What is an explicit cursor?
- Implicit and explicit cursors
- The simple loop (basic loop)
- Working with a cursor row-by-row.
- Using cursors in procedures
- OPEN..FETCH CLOSE construct
- The %NOTFOUND operator.
- FOR UPDATE cursors

#### The IF-THEN-ELSE statement

- Nesting of these statements.
- PL/SQL CASE statement
- Use of FOR Loop & WHILE Loop
- How to label a loop and how to nest a loop.

#### Exceptions and Exception handling

- What are Exceptions?
- Need for handling exceptions
- Predefined exceptions
- The *raise\_application\_error* built-in procedure & User defined exceptions
- Non predefined exceptions
- Raising & Trapping Exceptions
- Exception propagation in nested calls
- The SQLCODE & SQLERRM built-in PL/SQL functions.

#### Creating Functions

- Syntax of CREATE FUNCTION
- Defining of user defined functions
- Restrictions on function calls
- Dropping functions

#### Creating Procedures (Stored sub-programs)

- Stored sub-programs (Procedure/Functions)
- Need for Procedures – create syntax
- Use of Parameters ( IN , OUT & IN OUT)
- Executing procedures
- Exceptions in procedures

#### Creating Packages (PL/SQL API Development)

- What are packages and what is an API ?
- The Need for packages
- Defining Package Body & Specification
- Private and Public Package level data
- Calling package functions, procedures
- Referencing package-level constants & variables

#### Package Features

- Overloading package subprograms
- Forward declaration of subprograms
- Creating bodiless package
- Persistent state of package variables
- Persistent state of package cursor

#### Oracle Supplied Packages

- DBMS\_OUTPUT
- DBMS\_UTILITY
- UTL\_FILE
- DBMS\_RANDOM
- UTL\_MAIL
- EXECUTE IMMEDIATE

#### Oracle Utilities

- SQL\*loader (sqlldr)

- The spool and spool append commands.
- Profiling PL/SQL code using DBMS\_HPROF supplied package.

### Composite data types

- Using %ROWTYPE
- PLSQL Records
- PLSQL Table of Records
- INDEX BY Table
- INDEX BY Table of Records

### Triggers

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

### Bulk Data Processing with PL/SQL

- The PL/SQL & SQL Run Time engine
- Context switching between PL/SQL Run Time engine and SQL Run Time engine.
- BULK COLLECT operation in PL/SQL
- FOR ALL operation in PL/SQL
- Exception Handling during Bulk Data Processing.

### 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
- How to invoke DBMS\_SQL Package?
- Implement DBMS\_SQL with a Parameterized DML Statement
- Dynamic SQL Functional Completeness

### Creating Compound, DDL, and Event Database Triggers

- What are Compound Triggers?
- Identify the Timing-Point Sections of a Table Compound Trigger
- Understand the Compound Trigger Structure for Tables and Views

- Implement a Compound Trigger to Resolve the Mutating Table Error
- Database Triggers & Stored Procedures
- Create Triggers on DDL Statements
- Create Database & System-Events Triggers
- System Privileges Required for Triggers

### Manage Dependencies

- Overview of Schema Object Dependencies
- Query Direct Object Dependencies using the USER\_DEPENDENCIES View
- Query an Object's Status
- Invalidation of Dependent Objects
- Display the Direct and Indirect Dependencies
- Fine-Grained Dependency Management in Oracle Database 12c
- Understand Remote Dependencies
- Recompile a PL/SQL Program Unit

### PL/SQL Compiler

- What is the PL/SQL Compiler?
- Describe Initialization Parameters for PL/SQL Compilation
- List new PL/SQL Compile Time Warnings
- Overview of PL/SQL Compile Time Warnings for Subprograms
- List the benefits of Compiler Warnings
- List the PL/SQL Compile Time Warning Messages Categories
- Setting the Warning Messages Levels: Using SQL Developer, PLSQL\_WARNINGS Initialization Parameter, and the DBMS\_WARNING Package Subprograms
- View Compiler Warnings: Using SQL Developer, SQL\*Plus, or the Data Dictionary Views

### Design Considerations for PL/SQL Code

- Standardize Constants and Exceptions
- Understand Local Subprograms
- Write Autonomous Transactions
- Implement the NOCOPY Compiler Hint
- Invoke the PARALLEL\_ENABLE Hint
- The Cross-Session PL/SQL Function Result Cache
- The DETERMINISTIC Clause with Functions
- Usage of Bulk Binding to Improve Performance

**Certification Exam: Oracle Database 11g: Program with PL/SQL**

<b>Exam Number : 1Z0-144</b>	<b>1Z0-144</b>
Exam Product Version: Oracle Database 11g	This exam has been validated against Oracle Database 10g, Oracle Database 11g, Oracle Database 11g Release 2, and Oracle Database 12c Release 1.  <a href="#">Oracle PL/SQL Developer Certified Associate</a> <a href="#">Oracle Advanced PL/SQL Developer Certified Professional</a> ,
Exam Price: Rs 9604	
Format: MultipleChoice	
Duration: 90 minutes	
Number of Questions: 63	
Passing Score 65%	

Step 1 - Pass this exam.

Pass one SQL Exam:

- [Oracle Database 12c: SQL Fundamentals 1Z0-061](#) or
- [Oracle Database 11g: SQL Fundamentals I 1Z0-051](#) or
- [Oracle Database SQL Expert 1Z0-047](#)

Step 2 - Pass this exam.

[Oracle Database 11g: Program with PL/SQL 1Z0-144](#)