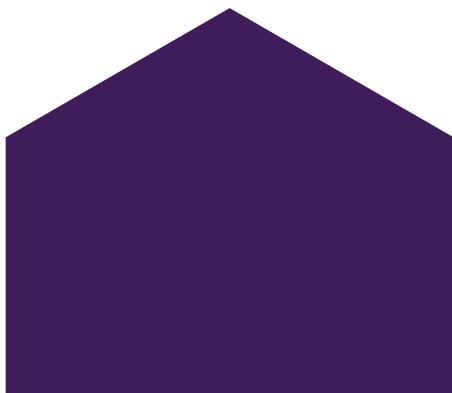
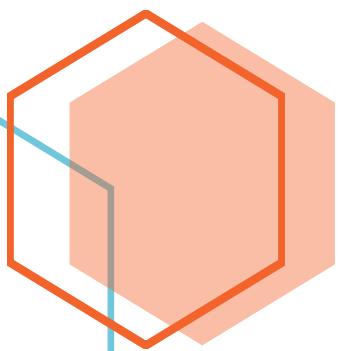




Advanced Oracle PLSQL Programming

We Make Training Affordable

Learn by Doing



Advanced Oracle PLSQL Programming

Course Overview

PL/SQL programming training is an intense course that is designed to give the student hands-on exposure to Oracle PL/SQL. The student learns by coding, and this class has dozens of in-class exercises and the student will be guided from very simple PL/SQL programs to increasingly complex PL/SQL coding techniques. The topics start with the basics of PL/SQL and progress into increasingly complex programmatic structures including array manipulation cursor management and bulking operations.

The PL/SQL training class begins with simple concepts and the student gradually masters PL/SQL through increasingly challenging classroom exercises. In this course, students learn to use the advanced features of PL/SQL in order to design and tune PL/SQL to interface with the database and other applications in the most efficient manner. Using advanced features of program design, packages, cursors, extended interface methods, and collections, students learn to write powerful PL/SQL programs.

Duration & Module Coverage

Duration: 3 Days (24 hrs)

Time: 9.30 am to 5.30 pm

Per Day Schedule	Module Coverage
Session 1: 09.30 - 11.00 Theory 11.00 - 11.10 Break 11.10 - 13.10 Practical 13.10 - 13.50 Break	Day 1 – Modules 1 to 6
Session 2: 13.50 - 15.50 Theory 15.50 - 16.00 Break 16.00 - 17.30 Practical	Day 2 - Modules 7 to 12
Day 3 – Modules 13 to 16	
Click here to view full detailed module list	

Learning Goals

By the end of this course, participants will gain a strong foundation in PL/SQL architecture and programming fundamentals. They will be able to design, develop, test, and debug PL/SQL programs using procedures, functions, triggers, and effective exception handling. Learners will also apply performance optimization techniques and best practices to build secure, efficient, and maintainable PL/SQL solutions.

Pre-Requisites

Participants are expected to have a basic understanding of SQL and relational database concepts. Familiarity with Oracle database tools such as SQL Developer or SQL*Plus is recommended. Prior exposure to programming logic is optional but will be beneficial for better comprehension of the course content.

Teaching Methodology

This is a very hands-on course where participants carry out practical exercises in the classroom. The concepts are taught through implementation of real-world use-cases. Our exercises have been carefully designed to replicate scenarios participants will face in real life work conditions. We have adopted a 'Learn by Doing' pedagogical approach - Class room training with practical. Our trainings are not only PPT based.

Who Should Take This Course?

This course is ideal for database developers and application programmers working with Oracle databases, as well as SQL developers seeking to strengthen their procedural programming skills. It is also well suited for data analysts or engineers involved in managing or automating database processes, and IT professionals preparing for Oracle PL/SQL certification or database development roles.

Course Content

Day 1

Learning Outcomes:

- ❖ Understand PL/SQL architecture, execution model, and its role in Oracle database programming.
- ❖ Write and execute structured PL/SQL blocks using variables, data types, and control structures.
- ❖ Implement effective exception handling to build reliable and maintainable PL/SQL programs.

<u>Session 1: Modules 1 to 4</u> <u>Time: 09.30 to 13.10</u>	<u>Session 2: Modules 5 & 6</u> <u>Time: 13.50 to 17.30</u>
<p>1. Overview of PL/SQL</p> <ul style="list-style-type: none"> • What is PL/SQL? • Benefits and Features of PL/SQL • PL/SQL Architecture <p>2. Basic PL/SQL Block Structure</p> <ul style="list-style-type: none"> • Anonymous Block • Named Block (Procedures, Functions) <p>3. PL/SQL Environment Setup</p> <ul style="list-style-type: none"> • Oracle SQL Developer • Executing PL/SQL in SQL*Plus <p>4. Data Types and Variables</p> <ul style="list-style-type: none"> • Scalar Data Types • Composite Data Types (Records, Collections) • Variable Declarations and Initialization • %TYPE and %ROWTYPE Attributes 	<p>5. Control Structures</p> <ul style="list-style-type: none"> • Scalar Data Types • Composite Data Types (Records, Collections) • Variable Declarations and Initialization • %TYPE and %ROWTYPE Attributes <p>6. Error Handling</p> <ul style="list-style-type: none"> • Exception Types (Predefined, User-Defined) • Exception Handling (RAISE, RAISE_APPLICATION_ERROR) • Handling Exceptions in Nested Blocks

Day 2

Learning Outcomes:

- ❖ Use cursors, procedures, functions, packages, and triggers to build modular and reusable PL/SQL programs.
- ❖ Implement advanced exception handling, logging, and error propagation for robust database applications.
- ❖ Optimize PL/SQL and SQL performance using tuning techniques, bulk processing, and efficient data structures.

<u>Session 1: Modules 7 to 9</u> <u>Time: 09.30 to 13.10</u>	<u>Session 2: Modules 10 to 12</u> <u>Time: 13.50 to 17.30</u>
<p>7. Cursors</p> <ul style="list-style-type: none"> • Implicit Cursors • Explicit Cursors • Cursor Attributes • Cursor FOR LOOP <p>8. Procedures and Functions</p> <ul style="list-style-type: none"> • Creating Procedures and Functions • Parameter Modes (IN, OUT, IN OUT) • Calling Procedures and Functions • Recursive Functions <p>9. Packages</p> <ul style="list-style-type: none"> • Creating Packages (Specification and Body) • Package Variables and Cursors • Overloading Subprograms • Package Initialization 	<p>10. Triggers</p> <ul style="list-style-type: none"> • Creating Triggers (Before, After, Instead Of) • Row-Level and Statement-Level Triggers • INSTEAD OF Triggers for Views • Mutating Table Errors and Solutions <p>11. Advanced Error Handling</p> <ul style="list-style-type: none"> • Creating Propagating Exceptions • Using SQLCODE and SQLERRM • Creating and Using User-Defined Exceptions • Logging Errors <p>12. Performance Tuning</p> <ul style="list-style-type: none"> • Identifying Performance Issues • Optimizing SQL Queries • Bulk Processing with BULK COLLECT and FORALL • Using Collections for Performance

Day 3

Learning Outcomes:

- ❖ Build flexible PL/SQL programs using dynamic SQL for runtime DDL and DML operations.
- ❖ Leverage advanced collections and techniques to design secure, scalable, and high-performance solutions.
- ❖ Apply PL/SQL best practices for maintainable code, effective testing, and controlled deployment.

<u>Session 1: Modules 13 & 14</u> <u>Time: 09.30 to 13.10</u>	<u>Session 2: Modules 15 & 16</u> <u>Time: 13.50 to 17.30</u>
<p>13. Dynamic SQL</p> <ul style="list-style-type: none"> • Introduction to Dynamic SQL • EXECUTE IMMEDIATE • DBMS_SQL Package • Dynamic SQL for DDL and DML Operations <p>14. Advanced Collections</p> <ul style="list-style-type: none"> • Nested Tables and VARRAYs • Associative Arrays (Index-By Tables) • Collection Methods 	<p>15. Advanced Techniques</p> <ul style="list-style-type: none"> • Autonomous Transactions • Using PL/SQL Compiler Options • Invoker's Rights vs. Definer's Rights • Securing PL/SQL Code <p>16. Best Practices & Wrap-Up</p> <ul style="list-style-type: none"> • Data Writing Readable and Maintainable Code • Code Review and Testing • Documentation and Comments • Version Control and Deployment Strategies