

# Oracle SQL & PL/SQL Course Syllabus

---

80% Practical training & 20% Theory

Experts Available 24x7 for your queries

**30 to 40 Hrs. of training**

## Trainer Profile

Our Trainers provide complete freedom to the students, to explore the subject and learn based on real-time examples. Our trainers help the candidates in completing their projects and even prepare them for interview questions and answers. Candidates are free to ask any questions at any time.

- More than **7+ Years** of Experience
- Expert level Subject Knowledge and fully up-to-date on real-world industry applications
- Trainers have experienced on multiple real-time projects in their Industries

Learn how to use Oracle from beginner level to advanced techniques which is taught by experienced working professionals. With our Oracle Training you'll learn concepts in expert level with practical manner.

## **Introduction to Oracle Database**

- List the features of Oracle Database 11g
- Discuss the basic design, theoretical, and physical aspects of a relational database
- Categorize the different types of SQL statements
- Describe the data set used by the course
- Log on to the database using SQL Developer environment
- Save queries to files and use script files in SQL Developer

## **Retrieve Data using the SQL SELECT Statement**

- List the capabilities of SQL SELECT statements
- Generate a report of data from the output of a basic SELECT statement
- Select All Columns
- Select Specific Columns
- Use Column Heading Defaults
- Use Arithmetic Operators
- Understand Operator Precedence
- Learn the DESCRIBE command to display the table structure

## **Learn to Restrict and Sort Data**

- Write queries that contain a WHERE clause to limit the output retrieved
- List the comparison operators and logical operators that are used in a WHERE clause
- Describe the rules of precedence for comparison and logical operators
- Use character string literals in the WHERE clause
- Write queries that contain an ORDER BY clause to sort the output of a SELECT statement
- Sort output in descending and ascending order

## **Usage of Single-Row Functions to Customize Output**

- Describe the differences between single row and multiple row functions
- Manipulate strings with character function in the SELECT and WHERE clauses
- Manipulate numbers with the ROUND, TRUNC, and MOD functions
- Perform arithmetic with date data
- Manipulate dates with the DATE functions

## **Invoke Conversion Functions and Conditional Expressions**

- Describe implicit and explicit data type conversion
- Use the TO\_CHAR, TO\_NUMBER, and TO\_DATE conversion functions

- Nest multiple functions
- Apply the NVL, NULLIF, and COALESCE functions to data
- Use conditional IF THEN ELSE logic in a SELECT statement

### **Aggregate Data Using the Group Functions**

- Use the aggregation functions in SELECT statements to produce meaningful reports
- Divide the data into groups by using the GROUP BY clause
- Exclude groups of data by using the HAVING clause

### **Display Data From Multiple Tables Using Joins**

- Write SELECT statements to access data from more than one table
- View data that generally does not meet a join condition by using outer joins
- Join a table by using a self-join

### **Use Sub-Queries to Solve Queries**

- Describe the types of problem that sub-queries can solve
- Define sub-queries
- List the types of sub-queries
- Write single-row and multiple-row sub-queries

### **The SET Operators**

- Describe the SET operators
- Use a SET operator to combine multiple queries into a single query
- Control the order of rows returned

### **Data Manipulation Statements**

- Describe each DML statement
- Insert rows into a table
- Change rows in a table by the UPDATE statement
- Delete rows from a table with the DELETE statement
- Save and discard changes with the COMMIT and ROLLBACK statements
- Explain read consistency

### **Use of DDL Statements to Create and Manage Tables**

- Categorize the main database objects
- Review the table structure
- List the data types available for columns
- Create a simple table
- Decipher how constraints can be created at table creation

- Describe how schema objects work

### **Other Schema Objects**

- Create a simple and complex view
- Retrieve data from views
- Create, maintain, and use sequences
- Create and maintain indexes
- Create private and public synonyms

### **Control User Access**

- Differentiate system privileges from object privileges
- Create Users
- Grant System Privileges
- Create and Grant Privileges to a Role
- Change Your Password
- Grant Object Privileges
- How to pass on privileges?
- Revoke Object Privileges

### **Management of Schema Objects**

- Add, Modify and Drop a Column
- Add, Drop and Defer a Constraint
- How to enable and Disable a Constraint?
- Create and Remove Indexes
- Create a Function-Based Index
- Perform Flashback Operations
- Create an External Table by Using ORACLE\_LOADER and by Using ORACLE\_DATAPUMP
- Query External Tables

### **Manage Objects with Data Dictionary Views**

- Explain the data dictionary
- Use the Dictionary Views
- USER\_OBJECTS and ALL\_OBJECTS Views
- Table and Column Information
- Query the dictionary views for constraint information
- Query the dictionary views for view, sequence, index, and synonym information

- Add a comment to a table
- Query the dictionary views for comment information

### **Manipulate Large Data Sets**

- Use Subqueries to Manipulate Data
- Retrieve Data Using a Subquery as Source
- Insert Using a Subquery as a Target
- Usage of the WITH CHECK OPTION Keyword on DML Statements
- List the types of Multitable INSERT Statements
- Use Multitable INSERT Statements
- Merge rows in a table
- Track Changes in Data over a period of time

### **Data Management in Different Time Zones**

- Time Zones
- CURRENT\_DATE, CURRENT\_TIMESTAMP, and LOCALTIMESTAMP
- Compare Date and Time in a Session's Time Zone
- DBTIMEZONE and SESSIONTIMEZONE
- Difference between DATE and TIMESTAMP
- INTERVAL Data Types
- Use EXTRACT, TZ\_OFFSET, and FROM\_TZ
- Invoke TO\_TIMESTAMP, TO\_YMINTERVAL and TO\_DSINTERVAL

### **Retrieve Data Using Sub-queries**

- Multiple-Column Subqueries
- Pairwise and Nonpairwise Comparison
- Scalar Subquery Expressions
- Solve problems with Correlated Subqueries
- Update and Delete Rows Using Correlated Subqueries
- The EXISTS and NOT EXISTS operators
- Invoke the WITH clause
- The Recursive WITH clause

### **Regular Expression Support**

- Use the Regular Expressions Functions and Conditions in SQL
- Use Meta Characters with Regular Expressions
- Perform a Basic Search using the REGEXP\_LIKE function

- Find patterns using the REGEXP\_INSTR function
- Extract Substrings using the REGEXP\_SUBSTR function
- Replace Patterns Using the REGEXP\_REPLACE function
- Usage of Sub-Expressions with Regular Expression Support
- Implement the REGEXP\_COUNT function

## **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

## **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

## **Implement Oracle-Supplied Packages in Application Development**

- What are Oracle-Supplied Packages?
- Examples of Some of the Oracle-Supplied Packages
- How Does the DBMS\_OUTPUT Package Work?
- Use the UTL\_FILE Package to Interact With Operating System Files
- Invoke the UTL\_MAIL Package
- Write UTL\_MAIL Subprograms

## **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

### **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

### **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?

### **Create Compound, DDL, and Event Database Triggers**

- What are Compound Triggers?
- Identify the Timing-Point Sections of a Table Compound Trigger
- Compound Trigger Structure for Tables and Views
- Implement a Compound Trigger to Resolve the Mutating Table Error
- Compare Database Triggers to Stored Procedures
- Create Triggers on DDL Statements
- Create Database-Event and System-Event Triggers
- System Privileges Required to Manage Triggers

### **The PL/SQL Compiler**

- What is the PL/SQL Compiler?
- Describe the Initialization Parameters for PL/SQL Compilation

- List the 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

### **Manage PL/SQL Code**

- What Is Conditional Compilation?
- Implement Selection Directives
- Invoke Predefined and User-Defined Inquiry Directives
- The PLSQL\_CCFLAGS Parameter and the Inquiry Directive
- Conditional Compilation Error Directives to Raise User-Defined Errors
- The DBMS\_DB\_VERSION Package
- Write DBMS\_PREPROCESSOR Procedures to Print or Retrieve Source Text
- Obfuscation and Wrapping PL/SQL Code

### **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 11g
- Understand Remote Dependencies
- Recompile a PL/SQL Program Unit

# Our Branches

---

## Chennai Locations

Velachery  
Tambaram  
OMR  
Porur  
Anna Nagar  
T.Nagar  
Thiruvanmiyur  
Siruseri  
Maraimalai Nagar

## Bangalore Locations

BTM Layout  
Marathahalli  
Rajaji Nagar  
Jaya Nagar  
Kalyan Nagar  
Electronic City  
Indira Nagar  
HSR Layout  
Hebbal

*Follow Us*

