

Introduction to Oracle 11g – SQL

RDBMS

- Data, Database Concept
- RDBMS Concept
- Codd's Rules
- Entity, Relation
- Normalization

Retrieving Data Using the SQL SELECT Statement

- List the capabilities of SQL SELECT statements
- Projection and selection of the rows
- Execute a basic SELECT statement

Restricting and Sorting Data

- Limit the rows that are retrieved by a query
- Use ampersand substitution to restrict and sort output at runtime
- Sort the rows that are retrieved by a query

Using Single-Row Functions to Customize Output

- Describe various types of functions available in SQL
- Use character, number, and date functions in SELECT statements

Using Conversion Functions and Conditional Expressions

- Describe various types of conversion functions that are available in SQL
- Apply conditional expressions in a SELECT statement (CASE, DECODE)
- Use the TO_CHAR, TO_NUMBER, and TO_DATE conversion functions

Reporting Aggregated Data Using the Group Functions

- Identify the available group functions
- Group data by using the GROUP BY clause
- Describe the use of group functions
- Include or exclude grouped rows by using the HAVING clause

Displaying Data from Multiple Tables

- Write SELECT statements to access data from more than one table using equijoins and nonequijoins
- View data that generally does not meet a join condition by using outer joins
- Join a table to itself by using a self-join
- Generate a Cartesian product of all rows from two or more tables

Using Subqueries to Solve Queries

Palasia Centre: 20/1 Modern Mansion, Palasia square, Indore Ph: 0731- 4066541, 4091658

Bhanwarkuan Centre: 18- Professor Colony, Wadhvani Chambers, Bhanwarkuan square, Indore.

Ph. 0731- 4289083/ 84

Email: info@ssiedu.in

- Define subqueries
- Describe the types of problems that the subqueries can solve
- List the types of subqueries
- Write single-row and multiple-row subqueries

Producing Readable outputs with ISQL Plus

- Substitution variables
- Define, Undefined, Redefined variable
- Set Options
- Creating and Executing Script file

Using the Set Operators

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

Manipulating Data

- Describe each data manipulation language (DML) statement
- Insert rows into a table
- Update rows in a table
- Delete rows from a table
- Control transactions
- Advance DML statement:- multiple Insert, Insert all

Using DDL Statements to Create and Manage Tables

- Categorize the main database objects
- Review the table structure
- List the data types that are available for columns
- Create a simple table
- Explain how constraints are created at the time of table creation
- Flashback
- Functionality of recycle Bin
- Purge
- Describe how schema objects work

Creating Other Schema Objects (Views)

- Create simple and complex views
- Retrieve data from views
- Inline views
- Top –N Analysis
- Create, maintain, and use sequences
- Create and maintain indexes
- Create private and public synonyms

Database security and controlling user access

- System security and data security
- user creation and management
- grant, revoke, with grant option
- System Privileges
- Objects Privileges
- Public synonyms

Duration: 30 Days

Palasia Centre: 20/1 Modern Mansion, Palasia square, Indore Ph: 0731- 4066541, 4091658

Bhanwarkuan Centre: 18- Professor Colony, Wadhvani Chambers, Bhanwarkuan square, Indore.

Ph. 0731- 4289083/ 84

Email: info@ssiedu.in

Programming with PL/SQL (11g)

Introduction to PL/SQL

- Explain the need for PL/SQL
- Identify the different types of PL/SQL blocks
- Explain the benefits of PL/SQL
- Output messages in PL/SQL

Declaring PL/SQL Variables

- Recognize valid and invalid identifiers
- List and describe various data types using the %TYPE attribute
- List the uses of variables, declare and initialize variables, use bind variables

Writing Executable Statements

- Identify lexical units in a PL/SQL block
- Describe when implicit conversions take place and when explicit conversions have to be dealt with
- Write readable code with appropriate indentation
- Use built-in SQL functions in PL/SQL and sequences in PL/SQL expressions
- Write nested blocks and qualify variables with labels

Interacting with the Oracle Database Server

- Create PL/SQL executable blocks using DML and transaction control statements
- Make use of the INTO clause to hold the values returned by a SQL statement

Writing Control Structures

- Identify the uses and types of control structures (IF, CASE statements and expressions)
- Apply guidelines when using conditional control structures
- Construct and identify loop statements

Working with Composite Data Types

- Create user-defined PL/SQL records
- Create an INDEX BY table and INDEX BY table of records
- Create a record with the %ROWTYPE attribute
- Describe the differences among records, tables, and tables of records

Using Explicit Cursors

- Distinguish between usage of implicit and explicit cursors, use SQL cursor attributes
- Declare and use cursors with parameters
- Declare and control explicit cursors, use simple loops and cursor FOR loops to fetch data
- Lock rows with the FOR UPDATE clause and reference the current row with the WHERE CURRENT OF clause

Handling Exceptions

- Define PL/SQL exceptions
- Recognize unhandled exceptions

Palasia Centre: 20/1 Modern Mansion, Palasia square, Indore Ph: 0731- 4066541, 4091658

Bhanwarkuan Centre: 18- Professor Colony, Wadhvani Chambers, Bhanwarkuan square, Indore.

Ph. 0731- 4289083/ 84

Email: info@ssiedu.in

- Handle different types of exceptions (pre-defined exceptions, non-predefined exceptions and user-defined exceptions)
- Propagate exceptions in nested blocks and call applications

Creating Stored Procedures and Functions

- Differentiate between anonymous blocks and subprograms
- Create a simple function
- Differentiate between procedures and functions
- Create a simple procedure and invoke it from an anonymous block
- Create a simple function that accepts a parameter

Creating Procedures

- Differentiate between anonymous blocks and subprograms, use a modularized and layered subprogram design, and identify the benefits of subprograms
- Work with procedures
- Create a simple procedure and invoke it from an anonymous block
- Handle exceptions in procedures, remove a procedure, and display a procedure's information

Creating Functions

- Differentiate between a procedure and a function
- Work with functions (create, invoke and remove functions)
- Describe the uses of functions

Creating Packages

- Identify the benefits and the components of packages
- Work with packages (create package specification and body, invoke package subprograms, remove a package and display package information)

Working with Packages

- Overload package subprograms, use forward declarations
- Manage persistent package data states for the life of a session and use PL/SQL tables and records in packages
- Create an initialization block in a package body

Using Oracle-Supplied Packages in Application Development

- Describe how the DBMS_OUTPUT package works
- Describe the main features of UTL_MAIL
- Use UTL_FILE to direct output to operating system files

Palasia Centre: 20/1 Modern Mansion, Palasia square, Indore Ph: 0731- 4066541, 4091658

Bhanwarkuan Centre: 18- Professor Colony, Wadhvani Chambers, Bhanwarkuan square, Indore.

Ph. 0731- 4289083/ 84

Email: info@ssiedu.in

Using Dynamic SQL

- Describe the execution flow of SQL statements
- Use the DBMS_SQL package
- Use Native Dynamic SQL (NDS)

Design Considerations for PL/SQL Code

- Create standard constants and exceptions
- Control the run-time privileges of a subprogram
- Use NOCOPY hint, PARALLEL ENABLE hint and DETERMINISTIC clause
- Write and call local subprograms
- Perform autonomous transactions
- Use bulk binding and the RETURNING clause with DML

Creating Triggers

- Describe different types of triggers and their uses
- Manage triggers
- Create database triggers

Database Triggers

- Create triggers on DDL statements
- Create triggers on system events

Managing PL/SQL Code

- Describe and use conditional compilation
- Hide PL/SQL source code using dynamic obfuscation and the Wrap utility

Managing Dependencies

- Track and manage procedural dependencies