

<b>CLASS: B. Sc (Information technology)</b>		<b>Semester – III</b>	
<b>COURSE: Advanced SQL</b>			
<b>Periods per week</b> <b>1 Period is 50 minutes</b>	<b>Lecture</b>	<b>5</b>	
	<b>TW/Tutorial/Practical</b>	<b>3</b>	
		<b>Hours</b>	<b>Marks</b>
<b>Evaluation System</b>	<b>Theory Examination</b>	<b>3</b>	<b>100</b>
	<b>TW/Tutorial/Practical</b>	<b>--</b>	<b>50</b>

Unit-I	<b>Structured Query Language:</b> Writing Basic SQL Select Statements, Restricting and Sorting Data, Single-Row Functions, Joins (Displaying Data from Multiple Tables), Aggregating Data using Group Functions, Subqueries, Manipulating Data, Creating and Managing Tables, Including Constraints, Creating Views, Creating Other Database Objects(Sequences, Indexes and Synonyms)
Unit-II	<b>Advanced SQL:</b> Controlling User Access, Using SET operators, DateTime Functions, Enhancements to Group by clause(cube, Rollup and Grouping), Advanced Subqueries (Multiple column subqueries, Subqueries in FROM clause, Scalar and correlated subqueries), WITH Clause, Hierarchical retrieval,
Unit-III	<b>PLSQL:</b> Introduction, Overview and benefits of PL/SQL, Subprograms, types of PL/SQL blocks, Simple Anonymous Block, Identifiers, types of Identifiers, Declarative Section, variables, Scalar Data Types, The %TYPE Attribute, Bind Variables, Sequences in PL/SQL Expressions, Executable Statements, PL/SQL Block Syntax, Comment the Code, Deployment of SQL Functions in PL/SQL, Convert Data Types, Nested Blocks, Operators. Interaction with the Oracle Server, Invoke SELECT Statements in PL/SQL, SQL Cursor concept, Data Manipulation in the Server using PL/SQL, SQL Cursor Attributes to Obtain Feedback on DML, Save and Discard Transactions.
Unit-IV	<b>Control Structures:</b> Conditional processing using IF Statements and CASE Statements, Loop Statement, While Loop Statement, For Loop Statement, the Continue Statement, <b>Composite Data Types:</b> PL/SQL Records, The %ROWTYPE Attribute, Insert and Update with PL/SQL Records, INDEX BY Tables, INDEX BY Table Methods, Use INDEX BY Table of Records, <b>Explicit Cursors</b> , Declare the Cursor, Open the Cursor, Fetch data from the Cursor, Close the Cursor, Cursor FOR loop, The %NOTFOUND and %ROWCOUNT Attributes, the FOR UPDATE Clause and WHERE CURRENT Clause, <b>Exception Handling</b> , Handle Exceptions with PL/SQL, Trap Predefined and non-predefined Oracle Server Errors, User-Defined Exceptions, Propagate Exceptions, RAISE_APPLICATION_ERROR Procedure,
Unit-V	<b>Stored Procedures:</b> Create a Modularized and Layered Subprogram Design, the PL/SQL Execution Environment, differences between Anonymous Blocks and Subprograms, Create, Call, and Remove Stored Procedures, Implement Procedures Parameters and Parameters Modes, View Procedure Information, Stored Functions and Debugging Subprograms, Create, Call, and Remove a Stored Function, advantages of using Stored Functions, the steps to create a stored function, Invoke User-Defined Functions in SQL Statements, Restrictions when calling Functions, Control side effects when calling Functions, View Functions Information, debug Functions and Procedures, <b>Packages</b> , advantages of Packages, components of a Package, Develop a Package, enable visibility of a Package's Components, Create the Package Specification and Body using the SQL CREATE Statement and SQL Developer, Invoke the Package Constructs, View the PL/SQL Source Code using the Data Dictionary, Deploying Packages, Overloading Subprograms in PL/SQL, Use the STANDARD Package, Use Forward Declarations, 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

Unit-VI	<p><b>Dynamic SQL:</b> The Execution Flow of 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, <b>Triggers</b>, 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 and Row Level Triggers, Create Instead of and Disabled Triggers, Manage, Test and Remove Triggers. Creating Compound, DDL and Event Database Triggers, Compound Trigger Structure for Tables and Views, Compound Trigger to Resolve the Mutating Table Error, Comparison of Database Triggers and Stored Procedures, Create Triggers on DDL Statements, Create Database-Event and System-Events Triggers, System Privileges Required to Manage Triggers</p>
---------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**Books:**

*Murach's Oracle SQL and PLSQL* by Joel Murach, Murach and Associates.

*Oracle Database 11g PL/SQL Programming Workbook*, ISBN: 9780070702264, By: Michael McLaughlin, John Harper, TATAMCGRAW-HILL

**Reference:**

*Oracle PL/SQL Programming*, Fifth Edition By Steven Feuerstein, Bill Pribyl

*Oracle 11g: SQL Reference* Oracle press

*Oracle 11g: PL/SQL Reference* Oracle Press.

*Expert Oracle PL/SQL*, By: Ron Hardman, Michael McLaughlin, TATAMCGRAW-HILL

*Oracle database 11g: hands on SQL/PL SQL* by Satish Asnani (PHI) EEE edition

**Term Work:** Should contain at least 6 assignments (one per unit) covering the syllabus.

**Suggested Practical:**

Should contain at least 10 programs. Some sample practicals are listed below.

**1. Practical 1: Select queries and joins**

- a. Select queries on single table using alias, where and order by clause.
- b. Select queries on single table using aggregate functions and group by clause.
- c. Querying data from multiple tables using all types of joins.

**2. Practical 2: Subqueries, DML and DDL**

- a. Querying single and multiple tables using subqueries.
- b. Manipulating data (Insert, update and delete)
- c. Creating simple tables and tables with constraints.

**3. Practical 3: Creating database objects, Controlling user access and using set operators**

- a. Creating Views, Sequences, Indexes and synonyms.
- b. Granting and revoking privileges on user objects.
- c. Using set operators, date-time functions, roll up, cube and grouping sets.

**4. Practical 4: Working with advanced subqueries and WITH clause**

- a. Multiple column subqueries, subqueries in from clause,
- b. Scalar subqueries and correlated subqueries,
- c. WITH Clause and hierarchical retrieval.

**5. Practical 5: Basic PL/SQL, INDEX BY tables, PL/SQL Record and FOR loop.**

- a. Creating anonymous PL/SQL blocks.
- b. Define, create, and use INDEX BY tables and a PL/SQL record.
- c. Process a number of rows from a table and populate another table with the results using a cursor FOR loop.

**6. Practical 6: Cursors, Exceptions and procedures issuing DML and query commands.**

- a. Cursors with parameters to process a number of rows from multiple tables.
- b. Create exception handlers for specific situations.
- c. Create procedures that issue DML and query commands.

**7. Practical 7: Functions and Stored Procedures**

- a. Creating and invoking functions from SQL statements.
- b. Creating and invoking stored procedures.
- c. Re-create the source code for a procedure and a function.

**8. Practical 8: Working with packages**

- a. Create package specifications and package bodies. Invoke the constructs in the packages.
- b. Create a package containing an overloaded function.
- c. Create a one-time-only procedure within a package to populate a PL/SQL table.

**9. Practical 9: Working with Large Objects and triggers**

- a. Create a table with both BLOB and CLOB columns. Use the DBMS\_LOB package to populate the table and manipulate the data.
- b. Create statement and row triggers.
- c. Create procedures that will be invoked from the triggers.

**10. Practical 10: Working with INSTEAD OF triggers, business rules and recompiling procedures, functions, packages and views.**

- a. Create instead of triggers for views.
- b. Implement a number of business rules. Create triggers for those rules that should be implemented as triggers. The triggers must execute procedures that are placed in a package.
- c. Use the DEPTREE\_FILL procedure and the IDEPTREE view to investigate dependencies in your schema. Recompile invalid procedures, functions, packages, and views.