



SASTRA
ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION
DEEMED TO BE UNIVERSITY
(U/S 3 of the UGC Act, 1956)



THINK MERIT | THINK TRANSPARENCY | THINK SASTRA
T H A N J A V U R | K U M B A K O N A M | C H E N N A I

Low Code Programming with Oracle APEX
(Semi Theory Semi Lab Course)

Course Code: INT327

Semester: VII

Lab Manual

2026- 2027

SHANMUGHA ARTS, SCIENCE, TECHNOLOGY AND RESEARCH ACADEMY
(SASTRA Deemed to be) University
Tirumalaisamudram, Thanjavur-613 401

School of Computing

Course Objectives:

This course will help the learner to infer the fundamentals of low code programming tools by using Oracle APEX. A background in database design and SQL is recommended. Construct data models in Oracle, design Oracle APEX applications in the cloud and develop a real-world application that is deployed on Oracle Cloud at the end of the project.

COURSE LEARNING OUTCOMES

Upon successful completion of this course, the learner will be able to

- Illustrate the architecture of Oracle APEX on the cloud
- Experiment with SQL to construct queries for the given problem
- Use PL/SQL to implement business logic
- Connect Oracle APEX to other cloud systems (for email, telephony, AI etc.) to build integrated applications
- Design and develop a complete Oracle APEX application
- Produce a real-world application that can be used by others in the capstone project
- Be capable of getting an Oracle APEX certification from Oracle

List of Experiments:

Ex. No.	Exercise List
1.	a. Create an Oracle APEX workspace account and explore SQL Workshop and App Builder b. Case study on themes in gallery
2.	a. Basic DDL and DML operations on student achievement schema using constraints (Primary Key, Foreign Key, Unique, Check, Not Null) b. Creating a database application from scratch and spreadsheet
3.	Using SQL Workshop to execute SQL queries a. Other schema objects - joins, indexes, views, sequences, synonyms, Common Table Expressions (WITH), CASE, MERGE and JSON_OBJECT b. Advanced SQL statement - Window function and Model clause
4.	Develop and execute PL/SQL programs in Oracle APEX SQL Workshop using functions, procedures, triggers, packages and exception handling for the Student Achievement database.
5.	Create two different types of forms a. Form on table b. Master detail forms c. Manage regions and navigation controls Designing Forms, Regions, and Navigation Controls in Oracle APEX.
6.	Add and update page items and buttons for the student achievement application a. Creating and Customizing Page Items and Buttons in Oracle APEX b. Session State Management in Oracle APEX
7.	Create and customize the branches, validations, computations, and process controls for the student achievement application
8.	Developing various types of report for student achievement applications - interactive report, interactive grid, classic report, report with form, faceted search page
9.	Create shared components, dynamic action and dynamic SQL on student achievement application
10.	Individually design and implement a complete, responsive Oracle APEX application.

Student achievement application -Tables

Student (
Regno NUMBER PRIMARY KEY,
Name VARCHAR2(255) NOT NULL,
Year NUMBER NOT NULL,
Qualification VARCHAR2(255),
Expertise VARCHAR2(255),
Willingness CHAR (1));

CertificationProgram(
ProgramID NUMBER PRIMARY KEY,
ProgramName VARCHAR2(255) NOT NULL,
Description CLOB);

Faculty (
FacultyID NUMBER PRIMARY KEY,
Name VARCHAR2(255) NOT NULL,
SchoolID NUMBER,
Department VARCHAR2(255));

School (
SchoolID NUMBER PRIMARY KEY,
SchoolName VARCHAR2(255) NOT NULL,
DeanName VARCHAR2(255));

AcademicAchievement (
AchievementID NUMBER PRIMARY KEY,
Regno NUMBER,
GPA NUMBER (3, 2),
CONSTRAINT FK_Student_Achievement FOREIGN KEY (Regno) REFERENCES
Student (Regno));

Note: Create additional tables, if required

Ex. No 1: Exploring the Oracle APEX Development Environment

Aim

To create an Oracle APEX workspace account and explore the Oracle APEX development environment.

Exercises

1. Create an **Oracle APEX Workspace** account.
2. Sign in to the Oracle APEX Workspace.
3. Explore the **Oracle APEX Home Page**.
4. Navigate to **SQL Workshop** and explore:
 - SQL Commands
 - SQL Scripts
 - Object Browser
 - Utilities
5. Navigate to **App Builder** and:
 - Create a sample application.
 - Explore application pages and components.
6. Explore **Shared Components** and identify:
 - Navigation Menu
 - Lists of Values (LOVs)
 - Authentication
 - Authorization
 - Templates
 - Breadcrumbs
7. Document the purpose of each Oracle APEX component with screenshots.

Expected Outcome

Students will be able to:

- Create and access an Oracle APEX Workspace.
- Navigate the Oracle APEX development environment.
- Understand the purpose of SQL Workshop, App Builder, Team Development, and Shared Components.

Ex. No 1.(b). Case Study on Oracle APEX Themes and Theme Gallery

Aim

To explore and customize themes using the Oracle APEX Theme Gallery.

Exercises

1. Open an existing Oracle APEX application.
2. Navigate to the **Theme Gallery**.
3. Explore the available themes and Universal Theme components.
4. Apply different themes to the application.
5. Customize the application appearance by modifying:
 - Theme Style
 - Color Palette
 - Navigation Menu
 - Page Templates
 - Region Templates
 - Button Templates

Outcome

Students will be able to:

- Explore the Oracle APEX Theme Gallery.
- Apply and customize themes.
- Compare themes based on usability and appearance.

Ex. No 2(a): Basic DDL and DML Operations on Student Achievement Database

Aim

To create a Student Achievement database and perform basic DDL and DML operations using integrity constraints in Oracle Database.

Exercises

1. Design the Student Achievement database schema.
2. Create the following tables:
 - Student
 - Course
 - Achievement
3. Apply the following constraints:
 - PRIMARY KEY
 - FOREIGN KEY
 - UNIQUE
 - CHECK
 - NOT NULL
4. Perform DDL operations:
 - CREATE
 - ALTER
 - TRUNCATE
 - DROP
5. Perform DML operations:
 - INSERT
 - UPDATE
 - DELETE
 - SELECT
6. Verify the behavior of integrity constraints by executing valid and invalid SQL statements.
7. Display and analyze the contents of each table.

Outcome

- Design and create relational database tables.
- Apply integrity constraints to maintain data consistency.
- Perform DDL and DML operations.
- Verify relational integrity using SQL queries.

Ex. No 2(b): Creating a Database Application from Scratch and Spreadsheet

Aim

To create a database application in Oracle APEX using both manual table creation and spreadsheet data import.

Create an Application from Spreadsheet

1. Prepare a spreadsheet containing student achievement records.
2. Import the spreadsheet into Oracle APEX.
3. Generate database tables automatically from the spreadsheet.
4. Create an application using the imported data.
5. Verify the generated reports and forms.
6. Perform CRUD operations on the imported data.
7. Compare manual application creation with spreadsheet-based application generation.

Outcome

- Create Oracle APEX applications from scratch.
- Import spreadsheet data into Oracle Database.
- Automatically generate applications from imported data.
- Perform CRUD operations on application data.
- Compare manual and spreadsheet-based application development approaches.

Ex.No 3(a): Using SQL Workshop to Create and Manage Schema Objects

Aim

To use Oracle APEX SQL Workshop for creating and managing schema objects and executing advanced SQL queries on the Student Achievement database.

Exercises

1. Open **SQL Workshop** and execute SQL statements using **SQL Commands**.
2. Retrieve data using different types of **JOIN** operations:
 - INNER JOIN
 - LEFT OUTER JOIN
 - RIGHT OUTER JOIN
 - FULL OUTER JOIN
3. Create and use an **INDEX** to improve query performance.
4. Create a **VIEW** to simplify complex queries.
5. Create and use a **SEQUENCE** to generate unique values for primary keys.
6. Create a **SYNONYM** for a database object and access it using the synonym.
7. Write SQL queries using **Common Table Expressions (WITH)** to simplify complex queries.
8. Use the **CASE** expression to classify students based on marks, CGPA, or achievement level.
9. Perform **MERGE** operations to synchronize data between source and target tables.
10. Generate JSON data using **JSON_OBJECT**.
11. Execute and verify all SQL statements using SQL Workshop.
12. Record the SQL queries, outputs, and observations.

Outcome

- Execute SQL queries using Oracle APEX SQL Workshop.
- Retrieve data using different types of JOIN operations.
- Create and manage schema objects such as INDEXES, VIEWS, SEQUENCES, and SYNONYMS.
- Simplify complex SQL queries using Common Table Expressions (WITH).
- Apply conditional logic using the CASE expression.
- Synchronize database records using the MERGE statement.
- Generate and process JSON data using JSON_OBJECT, JSON_ARRAYAGG, and JSON_TABLE.
- Develop advanced SQL skills required for Oracle APEX application development.

Ex. No 3(b) : Executing Advanced SQL Statements Using SQL Workshop

Aim

To execute advanced SQL statements using **Window Functions** and the **MODEL Clause** in Oracle Database through Oracle APEX SQL Workshop.

Exercises

1. Open **SQL Workshop** → **SQL Commands** in Oracle APEX.
2. Execute SQL queries using the following **Window Functions**:
 - ROW_NUMBER()
 - RANK()
 - DENSE_RANK()
 - LAG()
 - LEAD()
 - FIRST_VALUE()
 - LAST_VALUE()
 - SUM() OVER()
 - AVG() OVER()
 - PARTITION BY
 - ORDER BY
 - MODEL Clause
 - Analytical SQL

Spreadsheet-like Data Modeling

3. Apply **PARTITION BY** and **ORDER BY** clauses to perform analytical operations on student achievement data.
4. Generate rankings based on marks, CGPA, or achievement scores using window functions.
5. Calculate running totals, cumulative averages, and moving averages using analytical functions.
6. Write SQL queries using the **MODEL Clause** to perform spreadsheet-like calculations and data modeling.
7. Use the MODEL Clause to:
 - Forecast student performance.
 - Compute derived values.
 - Apply business rules across multiple rows.
8. Execute and verify the SQL statements using SQL Workshop.
9. Record the SQL queries, outputs, and observations.

Outcome

- Execute advanced analytical SQL queries using Oracle SQL.
- Apply Window Functions for ranking, aggregation, and comparative analysis.
- Use **PARTITION BY** and **ORDER BY** with analytical functions.

- Perform spreadsheet-like calculations using the **MODEL Clause**.
- Analyze and manipulate relational data efficiently using advanced SQL techniques.

Ex. No 4. Developing and Executing PL/SQL Programs Using SQL Workshop

Aim

To develop and execute PL/SQL programs in Oracle APEX SQL Workshop using functions, procedures, triggers, packages, cursors, and exception handling for the Student Achievement database.

Exercises

1. Open **SQL Workshop** → **SQL Commands** in Oracle APEX.
2. Write and execute anonymous PL/SQL blocks to perform database operations.
3. Create and execute a **Procedure** to insert, update, or delete student achievement records.
4. Create and execute a **Function** to calculate and return the student's grade, percentage, or achievement status.
5. Create **Database Triggers** to:
 - Validate data before insertion or update.
 - Automatically maintain audit information.
 - Log changes to student achievement records.
6. Create a **Package** containing related procedures and functions for managing student achievement data.
7. Write PL/SQL programs using:
 - **Implicit Cursor**
 - **Explicit Cursor**
 to retrieve and process multiple records.
8. Implement **Exception Handling** to handle runtime errors, including:
 - NO_DATA_FOUND
 - TOO_MANY_ROWS
 - ZERO_DIVIDE
9. Execute and test each PL/SQL program using SQL Workshop.
10. Verify the program outputs and document the observations.

Outcome

- Develop and execute PL/SQL programs in Oracle APEX SQL Workshop.
- Create and use PL/SQL procedures, functions, triggers, and packages.
- Retrieve and process data using implicit and explicit cursors.
- Handle runtime exceptions using Oracle PL/SQL exception handling.
- Develop reusable and modular database programs for Oracle APEX applications.

Ex. No 5: Designing Forms, Regions, and Navigation Controls in Oracle APEX

Aim

To design and customize forms, regions, and navigation controls for the Student Achievement application using Oracle APEX.

Exercises

1. Open the **Student Achievement** application in Oracle APEX App Builder.
2. Create the following types of forms:
 - Form on a Table
 - Form with Report
 - Master–Detail Form
3. Design and customize page **Regions** by creating:
 - Static Content Region
 - Interactive Report Region
 - Interactive Grid Region
 - Cards Region
 - Chart Region
4. Configure page layouts and region templates.
5. Create and customize **Navigation Controls**, including:
 - Navigation Menu
 - Breadcrumb Navigation
 - Navigation Bar
 - Tabs
 - Lists
6. Add and configure page items, buttons, and labels within forms.
7. Apply validations and dynamic actions to form controls.
8. Test navigation between pages and verify application flow.
9. Execute CRUD (Create, Read, Update, Delete) operations using the designed forms.
10. Document the application design with screenshots and observations.

Outcome

- Design and customize forms in Oracle APEX.
- Create and manage different types of regions.
- Configure navigation menus, breadcrumbs, tabs, and lists.
- Develop user-friendly interfaces for database applications.
- Implement and test CRUD operations using Oracle APEX forms.

Ex. No 6(a): Creating and Customizing Page Items and Buttons in Oracle APEX

Aim

To create and customize page items and buttons for the Student Achievement application using Oracle APEX.

Exercises

1. Open the **Student Achievement** application in Oracle APEX App Builder.
2. Create a new page or modify an existing form page.
3. Add and configure different **Page Items**, including:
 - Text Field
 - Text Area
 - Number Field
 - Date Picker
 - Select List
 - Radio Group
 - Checkbox Group
 - Switch
 - Display Only
 - Hidden Item
 - File Browse (optional)
4. Configure page item properties such as:
 - Labels
 - Default Values
 - Required Attribute
 - Format Masks
 - Help Text
 - Lists of Values (LOVs)
5. Create and customize **Buttons** for:
 - Create
 - Save
 - Apply Changes
 - Delete
 - Cancel
 - Reset
6. Configure button properties:
 - Position
 - Template
 - Appearance
 - Icon
 - Hot Button

- Conditions
- 7. Associate buttons with appropriate **Processes** and **Branches**.
- 8. Apply validations to page items and test form submission.
- 9. Execute the application and verify the functionality of page items and buttons.
- 10. Record observations with screenshots.

Outcome

- Create and configure various Oracle APEX page items.
- Customize page item properties and Lists of Values (LOVs).
- Design and configure action buttons.
- Associate buttons with page processes and navigation.
- Develop user-friendly data entry forms for Oracle APEX applications.

Ex.No 6(b): Session State Management in Oracle APEX

Aim

To understand and implement Session State Management in Oracle APEX by managing page items, session variables, navigation, and cache for the Student Achievement application.

Exercises

1. Open the **Student Achievement** application in Oracle APEX.
2. Display and observe the **Session ID** for the current user session.
3. Create page items and examine how their values are stored in **Session State**.
4. Compare the behavior of:
 - **Submit Page**
 - **Redirect to Page**
5. Pass values between pages using:
 - Page Items
 - Branches
 - URL Parameters
 - Application Items
6. Create and use **Application Items** to maintain data across multiple pages.
7. Configure **Session State Protection** for selected page items and test its behavior.
8. Clear page cache using:
 - Clear Cache property in Branches
 - URL syntax
 - Session management options
9. Reset the application session and verify that session values are cleared.
10. Use the **Developer Toolbar** to inspect Session State and debug page processing.
11. Record observations with screenshots of Session State before and after page submissions.

Outcome

- Understand the concept and importance of Session State in Oracle APEX.
- Manage page and application item values during user sessions.
- Differentiate between page submission and page redirection.
- Pass data between application pages using different techniques.
- Clear cache and reset application sessions.
- Use Session State information to debug Oracle APEX applications.

Ex. No 7. Creating and Customizing Branches, Validations, Computations, and Processes in Oracle APEX

Aim

To create and customize **Branches, Validations, Computations, and Processes** for the Student Achievement application using Oracle APEX.

Exercises

1. Open the **Student Achievement** application in Oracle APEX App Builder.
2. Create and configure **Validations** to ensure data integrity:
 - Value Required
 - Unique Value
 - Value in Specified Range (Marks/CGPA)
 - PL/SQL Expression Validation
 - SQL Query Validation
3. Create **Computations** to automatically:
 - Calculate Total Marks
 - Calculate Percentage
 - Determine Grade/CGPA
 - Assign default values to page items
4. Create **Processes** to:
 - Insert new student achievement records
 - Update existing records
 - Delete records
 - Execute PL/SQL after page submission
5. Create and configure **Branches** to:
 - Navigate to a report page after successful form submission
 - Redirect users based on button actions
 - Display success or error messages
6. Apply **Conditional Processing** to execute validations, computations, processes, and branches based on user actions.
7. Test the complete page processing flow by submitting valid and invalid data.
8. Observe and document the execution sequence of:
 - Validations
 - Computations
 - Processes
 - Branches
9. Record the outputs and application behavior with screenshots.

Outcome

- Create and customize validations to enforce business rules.
- Perform automatic calculations using computations.
- Develop page processes to execute database operations.
- Configure branches to control page navigation.

- Apply conditional logic during page processing.
- Understand the execution order of page processing components in Oracle APEX.

Ex. No 8. Developing Various Types of Reports in Oracle APEX

Aim

To develop and customize various types of reports for the Student Achievement application using Oracle APEX.

Exercises

1. Open the **Student Achievement** application in Oracle APEX App Builder.
2. Create an **Interactive Report** to display student achievement records with features such as:
 - Search and Filter
 - Sort
 - Control Break
 - Aggregate Functions
 - Highlighting
 - Download Report
3. Create an **Interactive Grid** to enable users to:
 - View records
 - Insert new records
 - Update existing records
 - Delete records
 - Perform inline editing
4. Create a **Classic Report** to display formatted student achievement data.
5. Create a **Report with Form** to:
 - Display student records in a report
 - Edit selected records using a linked form
 - Perform CRUD operations
6. Create a **Faceted Search Page** to enable users to search and filter records using multiple criteria such as:
 - Department
 - Course
 - Academic Year
 - Achievement Category
 - Grade/CGPA
7. Customize report attributes, including:
 - Columns
 - Formatting
 - Pagination
 - Sorting
 - Filtering
 - Actions Menu
8. Execute the application and verify the functionality of each report type.
9. Compare the features and use cases of the different report types.
10. Document the implementation with screenshots and observations.

Outcome

- Create and customize different report types in Oracle APEX.
- Develop interactive reports with advanced data analysis features.
- Implement interactive grids for inline data manipulation.

- Design classic reports for structured data presentation.
- Create report-with-form pages for CRUD operations.
- Develop faceted search pages for efficient multi-criteria filtering.
- Select the appropriate report type based on application requirements.

Ex. No 9: Creating Shared Components, Dynamic Actions, and Dynamic SQL in Oracle APEX

Aim

To create and configure **Shared Components**, implement **Dynamic Actions**, and develop **Dynamic SQL** for the Student Achievement application using Oracle APEX.

Exercises

Part A: Shared Components

1. Open the **Student Achievement** application in Oracle APEX App Builder.
2. Create and configure the following **Shared Components**:
 - Lists of Values (LOVs)
 - Navigation Menu
 - Breadcrumbs
 - Navigation Bar
 - Lists
 - Authentication Scheme
 - Authorization Scheme
 - Build Options (Introduction)
3. Associate the shared components with application pages.

Part B: Dynamic Actions

4. Create **Dynamic Actions** to perform client-side operations without page submission, such as:
 - Show/Hide page items
 - Enable/Disable page items
 - Set or Clear item values
 - Display success or warning messages
 - Refresh a report or region
 - Execute JavaScript or PL/SQL on user events
5. Configure dynamic actions based on events such as:
 - Click
 - Change
 - Page Load
 - Selection Change

Part C: Dynamic SQL

6. Create SQL queries that dynamically retrieve student achievement data based on user inputs.
7. Implement dynamic filtering and sorting using page item values.
8. Use PL/SQL to construct and execute dynamic SQL statements where appropriate.
9. Test and validate the generated reports and retrieved data.

Outcome

- Create and configure reusable Shared Components.
- Implement Dynamic Actions to improve application interactivity without full page refresh.
- Develop Dynamic SQL queries for flexible data retrieval.
- Build responsive and user-friendly Oracle APEX applications.
- Integrate reusable components and dynamic features into database-driven applications.

Ex. No 10: Design and Implementation of a Complete Responsive Oracle APEX Application

Aim

To design, develop, and deploy a complete responsive Oracle APEX application by integrating database design, SQL, PL/SQL, and Oracle APEX development features.

Exercises

1. Select a real-world application domain (e.g., Student Achievement, Library Management, Hospital Management, Inventory Management, Event Management, or Online Examination System).
2. Design the database schema by identifying entities, relationships, and integrity constraints.
3. Create the database objects using Oracle SQL:
 - Tables
 - Constraints
 - Views
 - Sequences
4. Populate the database with sample data.
5. Create an Oracle APEX application from the database schema.
6. Design responsive user interfaces by creating:
 - Forms
 - Interactive Reports
 - Interactive Grids
 - Dashboards
 - Faceted Search Pages
7. Implement application features using:
 - Page Items
 - Buttons
 - Shared Components
 - Dynamic Actions
 - Validations
 - Computations
 - Processes
 - Branches
8. Develop PL/SQL components such as procedures, functions, packages, triggers, and exception handling where required.
9. Implement authentication and authorization to secure the application.
10. Test the application on desktop and mobile devices to ensure responsive design.
11. Debug and optimize the application by resolving errors and improving usability.
12. Demonstrate the completed application and submit the project documentation.

Outcome

- Design and develop a complete Oracle APEX application from database design to deployment.
- Integrate SQL, PL/SQL, and Oracle APEX features into a unified application.
- Develop responsive, secure, and user-friendly database applications.
- Apply best practices in application design, testing, debugging, and deployment.
- Demonstrate competency in low-code application development using Oracle APEX.