**7)Packages**

**Scenario 1: Group all customer-related procedures and functions into a package.**

* + **Question: Create a package CustomerManagement with procedures for adding a new customer, updating customer details, and a function to get customer balance.**

**Scenario 2: Create a package to manage employee data.**

* + **Question: Write a package EmployeeManagement with procedures to hire new employees, update employee details, and a function to calculate annual salary.**

**Scenario 3: Group all account-related operations into a package.**

* + **Question: Create a package AccountOperations with procedures for opening a new account, closing an account, and a function to get the total balance of a customer across all accounts.**

i)Scenario 1-

Package Specification:

CREATE OR REPLACE PACKAGE CustomerManagement AS

PROCEDURE AddNewCustomer(p\_customer\_id IN NUMBER, p\_customer\_name IN VARCHAR2, p\_contact\_info IN VARCHAR2);

PROCEDURE UpdateCustomerDetails(p\_customer\_id IN NUMBER, p\_customer\_name IN VARCHAR2, p\_contact\_info IN VARCHAR2);

FUNCTION GetCustomerBalance(p\_customer\_id IN NUMBER) RETURN NUMBER;

END CustomerManagement;

Package Body:

CREATE OR REPLACE PACKAGE BODY CustomerManagement AS

PROCEDURE AddNewCustomer(p\_customer\_id IN NUMBER, p\_customer\_name IN VARCHAR2, p\_contact\_info IN VARCHAR2) IS

BEGIN

INSERT INTO Customers (customer\_id, customer\_name, contact\_info)

VALUES (p\_customer\_id, p\_customer\_name, p\_contact\_info);

END AddNewCustomer;

PROCEDURE UpdateCustomerDetails(p\_customer\_id IN NUMBER, p\_customer\_name IN VARCHAR2, p\_contact\_info IN VARCHAR2) IS

BEGIN

UPDATE Customers

SET customer\_name = p\_customer\_name,

contact\_info = p\_contact\_info

WHERE customer\_id = p\_customer\_id;

END UpdateCustomerDetails;

FUNCTION GetCustomerBalance(p\_customer\_id IN NUMBER) RETURN NUMBER IS

v\_balance NUMBER;

BEGIN

SELECT balance

INTO v\_balance

FROM CustomerBalances

WHERE customer\_id = p\_customer\_id;

RETURN v\_balance;

END GetCustomerBalance;

END CustomerManagement;

ii)Scenario 2-

Package Specification:

CREATE OR REPLACE PACKAGE EmployeeManagement AS

PROCEDURE HireNewEmployee(p\_employee\_id IN NUMBER, p\_employee\_name IN VARCHAR2, p\_position IN VARCHAR2, p\_salary IN NUMBER);

PROCEDURE UpdateEmployeeDetails(p\_employee\_id IN NUMBER, p\_employee\_name IN VARCHAR2, p\_position IN VARCHAR2, p\_salary IN NUMBER);

FUNCTION CalculateAnnualSalary(p\_employee\_id IN NUMBER) RETURN NUMBER;

END EmployeeManagement;

Package Body:

CREATE OR REPLACE PACKAGE BODY EmployeeManagement AS

PROCEDURE HireNewEmployee(p\_employee\_id IN NUMBER, p\_employee\_name IN VARCHAR2, p\_position IN VARCHAR2, p\_salary IN NUMBER) IS

BEGIN

INSERT INTO Employees (employee\_id, employee\_name, position, salary)

VALUES (p\_employee\_id, p\_employee\_name, p\_position, p\_salary);

END HireNewEmployee;

PROCEDURE UpdateEmployeeDetails(p\_employee\_id IN NUMBER, p\_employee\_name IN VARCHAR2, p\_position IN VARCHAR2, p\_salary IN NUMBER) IS

BEGIN

UPDATE Employees

SET employee\_name = p\_employee\_name,

position = p\_position,

salary = p\_salary

WHERE employee\_id = p\_employee\_id;

END UpdateEmployeeDetails;

FUNCTION CalculateAnnualSalary(p\_employee\_id IN NUMBER) RETURN NUMBER IS

v\_annual\_salary NUMBER;

BEGIN

SELECT salary \* 12

INTO v\_annual\_salary

FROM Employees

WHERE employee\_id = p\_employee\_id;

RETURN v\_annual\_salary;

END CalculateAnnualSalary;

END EmployeeManagement;

iii)Scenario 3-

Package Specification:

CREATE OR REPLACE PACKAGE AccountOperations AS

PROCEDURE OpenNewAccount(p\_account\_id IN NUMBER, p\_customer\_id IN NUMBER, p\_initial\_balance IN NUMBER);

PROCEDURE CloseAccount(p\_account\_id IN NUMBER);

FUNCTION GetTotalBalance(p\_customer\_id IN NUMBER) RETURN NUMBER;

END AccountOperations;

Package Body:

CREATE OR REPLACE PACKAGE BODY AccountOperations AS

PROCEDURE OpenNewAccount(p\_account\_id IN NUMBER, p\_customer\_id IN NUMBER, p\_initial\_balance IN NUMBER) IS

BEGIN

INSERT INTO Accounts (account\_id, customer\_id, balance)

VALUES (p\_account\_id, p\_customer\_id, p\_initial\_balance);

END OpenNewAccount;

PROCEDURE CloseAccount(p\_account\_id IN NUMBER) IS

BEGIN

DELETE FROM Accounts

WHERE account\_id = p\_account\_id;

END CloseAccount;

FUNCTION GetTotalBalance(p\_customer\_id IN NUMBER) RETURN NUMBER IS

v\_total\_balance NUMBER;

BEGIN

SELECT SUM(balance)

INTO v\_total\_balance

FROM Accounts

WHERE customer\_id = p\_customer\_id;

RETURN v\_total\_balance;

END GetTotalBalance;

END AccountOperations;