**Exercise 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.

**Code:**

CREATE OR REPLACE PACKAGE BODY Customer\_Pkg AS

  PROCEDURE AddCustomer (

    p\_id NUMBER,

    p\_name VARCHAR2,

    p\_dob DATE,

    p\_balance NUMBER

  ) IS

  BEGIN

    INSERT INTO Customers (CustomerID, Name, DOB, Balance, LastModified)

    VALUES (p\_id, p\_name, p\_dob, p\_balance, SYSDATE);

    DBMS\_OUTPUT.PUT\_LINE('✅ Customer ' || p\_name || ' added.');

  EXCEPTION

    WHEN DUP\_VAL\_ON\_INDEX THEN

      DBMS\_OUTPUT.PUT\_LINE('❌ Error: Customer ID already exists.');

    WHEN OTHERS THEN

      DBMS\_OUTPUT.PUT\_LINE('⚠️ ' || SQLERRM);

  END;

  FUNCTION GetBalance (p\_id NUMBER) RETURN NUMBER IS

    v\_balance NUMBER;

  BEGIN

    SELECT Balance INTO v\_balance

    FROM Customers

    WHERE CustomerID = p\_id;

    RETURN v\_balance;

  EXCEPTION

    WHEN NO\_DATA\_FOUND THEN

      DBMS\_OUTPUT.PUT\_LINE('❌ Customer not found.');

      RETURN NULL;

    WHEN OTHERS THEN

      DBMS\_OUTPUT.PUT\_LINE('⚠️ ' || SQLERRM);

      RETURN NULL;

  END;

END Customer\_Pkg;

/

SET SERVEROUTPUT ON;

BEGIN

  Customer\_Pkg.AddCustomer(10, 'Divya', TO\_DATE('1993-03-05','YYYY-MM-DD'), 20000);

END;

/

DECLARE

  bal NUMBER;

BEGIN

  bal := Customer\_Pkg.GetBalance(10);

  IF bal IS NOT NULL THEN

    DBMS\_OUTPUT.PUT\_LINE('Balance for Customer 10: ₹' || bal);

  END IF;

END;

/

**Output:**



**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.

**Code:**

SET SERVEROUTPUT ON;

BEGIN

  Customer\_Pkg.AddCustomer(10, 'Divya', TO\_DATE('1993-03-05','YYYY-MM-DD'), 20000);

END;

/

DECLARE

  bal NUMBER;

BEGIN

  bal := Customer\_Pkg.GetBalance(10);

  IF bal IS NOT NULL THEN

    DBMS\_OUTPUT.PUT\_LINE('Balance for Customer 10: ₹' || bal);

  END IF;

END;

/

CREATE OR REPLACE PACKAGE CustomerBalancePkg AS

  PROCEDURE UpdateBalance (

    p\_id NUMBER,

    p\_new\_amount NUMBER

  );

END CustomerBalancePkg;

/

CREATE OR REPLACE PACKAGE BODY CustomerBalancePkg AS

  PROCEDURE UpdateBalance (

    p\_id NUMBER,

    p\_new\_amount NUMBER

  ) IS

  BEGIN

    UPDATE Customers

    SET Balance = p\_new\_amount,

        LastModified = SYSDATE

    WHERE CustomerID = p\_id;

    IF SQL%ROWCOUNT = 0 THEN

      DBMS\_OUTPUT.PUT\_LINE('❌ No customer found with ID: ' || p\_id);

    ELSE

      DBMS\_OUTPUT.PUT\_LINE('✅ Balance updated for Customer ID: ' || p\_id);

    END IF;

  EXCEPTION

    WHEN OTHERS THEN

      DBMS\_OUTPUT.PUT\_LINE('⚠️ Error: ' || SQLERRM);

  END;

END CustomerBalancePkg;

/

BEGIN

  CustomerBalancePkg.UpdateBalance(1, 20000);

END;

/

**Output:**



**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.

**Code:**

CREATE OR REPLACE PACKAGE CustomerDeletePkg AS

  PROCEDURE DeleteCustomer (

    p\_id NUMBER

  );

END CustomerDeletePkg;

/

CREATE OR REPLACE PACKAGE BODY CustomerDeletePkg AS

  PROCEDURE DeleteCustomer (

    p\_id NUMBER

  ) IS

  BEGIN

    DELETE FROM Customers

    WHERE CustomerID = p\_id;

    IF SQL%ROWCOUNT = 0 THEN

      DBMS\_OUTPUT.PUT\_LINE('❌ Customer ID not found.');

    ELSE

      DBMS\_OUTPUT.PUT\_LINE('🗑️ Customer ID ' || p\_id || ' deleted.');

    END IF;

  EXCEPTION

    WHEN OTHERS THEN

      DBMS\_OUTPUT.PUT\_LINE('⚠️ Error: ' || SQLERRM);

  END;

END CustomerDeletePkg;

/

BEGIN

  CustomerDeletePkg.DeleteCustomer(1);

END;

/

**Output:**

