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

CREATE PACKAGE CustomerManagement AS

    PROCEDURE AddNewCustomer(customer\_id NUMBER, name VARCHAR2, dob DATE);

    PROCEDURE UpdateCustomerDetails(customer\_id NUMBER, name VARCHAR2, dob DATE);

    FUNCTION GetCustomerBalance(customer\_id NUMBER) RETURN NUMBER;

END CustomerManagement;

CREATE PACKAGE BODY CustomerManagement AS

    PROCEDURE AddNewCustomer(customer\_id NUMBER, name VARCHAR2, dob DATE) AS

    BEGIN

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

        VALUES (customer\_id, name, dob, 0, SYSDATE);

    END;

    PROCEDURE UpdateCustomerDetails(customer\_id NUMBER, name VARCHAR2, dob DATE) AS

    BEGIN

        UPDATE Customers

        SET Name = name, DOB = dob

        WHERE CustomerID = customer\_id;

    END;

    FUNCTION GetCustomerBalance(customer\_id NUMBER) RETURN NUMBER AS

        balance NUMBER;

    BEGIN

        SELECT Balance INTO balance

        FROM Customers

        WHERE CustomerID = customer\_id;

        RETURN balance;

    END;

END CustomerManagement;

-- Scenario 2: Create a package to manage employee data

CREATE PACKAGE EmployeeManagement AS

    PROCEDURE HireNewEmployee(employee\_id NUMBER, name VARCHAR2, position VARCHAR2, salary NUMBER, department VARCHAR2, hire\_date DATE);

    PROCEDURE UpdateEmployeeDetails(employee\_id NUMBER, name VARCHAR2, position VARCHAR2, salary NUMBER, department VARCHAR2);

    FUNCTION CalculateAnnualSalary(employee\_id NUMBER) RETURN NUMBER;

END EmployeeManagement;

CREATE PACKAGE BODY EmployeeManagement AS

    PROCEDURE HireNewEmployee(employee\_id NUMBER, name VARCHAR2, position VARCHAR2, salary NUMBER, department VARCHAR2, hire\_date DATE) AS

    BEGIN

        INSERT INTO Employees (EmployeeID, Name, Position, Salary, Department, HireDate)

        VALUES (employee\_id, name, position, salary, department, hire\_date);

    END;

    PROCEDURE UpdateEmployeeDetails(employee\_id NUMBER, name VARCHAR2, position VARCHAR2, salary NUMBER, department VARCHAR2) AS

    BEGIN

        UPDATE Employees

        SET Name = name, Position = position, Salary = salary, Department = department

        WHERE EmployeeID = employee\_id;

    END;

    FUNCTION CalculateAnnualSalary(employee\_id NUMBER) RETURN NUMBER AS

        salary NUMBER;

    BEGIN

        SELECT Salary INTO salary

        FROM Employees

        WHERE EmployeeID = employee\_id;

        RETURN salary \* 12;

    END;

END EmployeeManagement;

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

CREATE PACKAGE AccountOperations AS

    PROCEDURE OpenNewAccount(account\_id NUMBER, customer\_id NUMBER, account\_type VARCHAR2, initial\_balance NUMBER);

    PROCEDURE CloseAccount(account\_id NUMBER);

    FUNCTION GetTotalBalance(customer\_id NUMBER) RETURN NUMBER;

END AccountOperations;

CREATE PACKAGE BODY AccountOperations AS

    PROCEDURE OpenNewAccount(account\_id NUMBER, customer\_id NUMBER, account\_type VARCHAR2, initial\_balance NUMBER) AS

    BEGIN

        INSERT INTO Accounts (AccountID, CustomerID, AccountType, Balance, LastModified)

        VALUES (account\_id, customer\_id, account\_type, initial\_balance, SYSDATE);

    END;

    PROCEDURE CloseAccount(account\_id NUMBER) AS

    BEGIN

        DELETE FROM Accounts

        WHERE AccountID = account\_id;

    END;

    FUNCTION GetTotalBalance(customer\_id NUMBER) RETURN NUMBER AS

        total\_balance NUMBER;

    BEGIN

        SELECT SUM(Balance) INTO total\_balance

        FROM Accounts

        WHERE CustomerID = customer\_id;

        RETURN total\_balance;

    END;

END AccountOperations;