**Scenario 1: Customer Management Package**

**Package Specification:**

sql

CopyEdit

CREATE OR REPLACE PACKAGE CustomerManagement AS

PROCEDURE AddNewCustomer(p\_name VARCHAR2, p\_dob DATE, p\_balance NUMBER);

PROCEDURE UpdateCustomerDetails(p\_customer\_id NUMBER, p\_name VARCHAR2, p\_balance NUMBER);

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

END CustomerManagement;

/

**Package Body:**

sql

CopyEdit

CREATE OR REPLACE PACKAGE BODY CustomerManagement AS

PROCEDURE AddNewCustomer(p\_name VARCHAR2, p\_dob DATE, p\_balance NUMBER) IS

BEGIN

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

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

END AddNewCustomer;

PROCEDURE UpdateCustomerDetails(p\_customer\_id NUMBER, p\_name VARCHAR2, p\_balance NUMBER) IS

BEGIN

UPDATE Customers

SET Name = p\_name, Balance = p\_balance, LastModified = SYSDATE

WHERE CustomerID = p\_customer\_id;

END UpdateCustomerDetails;

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

v\_balance NUMBER;

BEGIN

SELECT Balance INTO v\_balance FROM Customers WHERE CustomerID = p\_customer\_id;

RETURN v\_balance;

END GetCustomerBalance;

END CustomerManagement;

/

**Expected Output**:

1. **Add a new customer**:

sql

CopyEdit

EXEC CustomerManagement.AddNewCustomer('John Doe', TO\_DATE('1985-08-15', 'YYYY-MM-DD'), 1000);

1. **Update customer details**:

sql

CopyEdit

EXEC CustomerManagement.UpdateCustomerDetails(1, 'John Smith', 1200);

1. **Get customer balance**:

sql

CopyEdit

SELECT CustomerManagement.GetCustomerBalance(1) FROM dual;

Output:

plaintext

CopyEdit

1200

**Scenario 2: Employee Management Package**

**Package Specification:**

sql

CopyEdit

CREATE OR REPLACE PACKAGE EmployeeManagement AS

PROCEDURE HireEmployee(p\_name VARCHAR2, p\_salary NUMBER, p\_department\_id NUMBER);

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

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

END EmployeeManagement;

/

**Package Body:**

sql

CopyEdit

CREATE OR REPLACE PACKAGE BODY EmployeeManagement AS

PROCEDURE HireEmployee(p\_name VARCHAR2, p\_salary NUMBER, p\_department\_id NUMBER) IS

BEGIN

INSERT INTO Employees (Name, Salary, DepartmentID, HireDate)

VALUES (p\_name, p\_salary, p\_department\_id, SYSDATE);

END HireEmployee;

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

BEGIN

UPDATE Employees

SET Name = p\_name, Salary = p\_salary

WHERE EmployeeID = p\_employee\_id;

END UpdateEmployeeDetails;

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

v\_salary NUMBER;

BEGIN

SELECT Salary INTO v\_salary FROM Employees WHERE EmployeeID = p\_employee\_id;

RETURN v\_salary \* 12; -- Annual Salary

END CalculateAnnualSalary;

END EmployeeManagement;

/

**Expected Output**:

1. **Hire new employee**:

sql

CopyEdit

EXEC EmployeeManagement.HireEmployee('Alice Green', 50000, 3);

1. **Update employee details**:

sql

CopyEdit

EXEC EmployeeManagement.UpdateEmployeeDetails(1, 'Alice Green', 55000);

1. **Calculate annual salary**:

sql

CopyEdit

SELECT EmployeeManagement.CalculateAnnualSalary(1) FROM dual;

Output:

plaintext

CopyEdit

660000

**Scenario 3: Account Operations Package**

**Package Specification:**

sql

CopyEdit

CREATE OR REPLACE PACKAGE AccountOperations AS

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

PROCEDURE CloseAccount(p\_account\_id NUMBER);

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

END AccountOperations;

/

**Package Body:**

sql

CopyEdit

CREATE OR REPLACE PACKAGE BODY AccountOperations AS

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

BEGIN

INSERT INTO Accounts (CustomerID, Balance, AccountCreationDate)

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

END OpenNewAccount;

PROCEDURE CloseAccount(p\_account\_id NUMBER) IS

BEGIN

DELETE FROM Accounts WHERE AccountID = p\_account\_id;

END CloseAccount;

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

v\_total\_balance NUMBER := 0;

BEGIN

SELECT SUM(Balance) INTO v\_total\_balance

FROM Accounts

WHERE CustomerID = p\_customer\_id;

RETURN v\_total\_balance;

END GetTotalBalance;

END AccountOperations;

/

**Expected Output**:

1. **Open a new account**:

sql

CopyEdit

EXEC AccountOperations.OpenNewAccount(1, 1500);

1. **Close an account**:

sql

CopyEdit

EXEC AccountOperations.CloseAccount(1);

1. **Get total balance of a customer across all accounts**:

sql

CopyEdit

SELECT AccountOperations.GetTotalBalance(1) FROM dual;

Output:

plaintext

CopyEdit

1500