Scenario 1: Customer Management Package

CREATE OR REPLACE PACKAGE CustomerManagement AS

PROCEDURE AddCustomer (

p\_customer\_id IN NUMBER,

p\_name IN VARCHAR2,

p\_dob IN DATE,

p\_balance IN NUMBER

);

PROCEDURE UpdateCustomer (

p\_customer\_id IN NUMBER,

p\_name IN VARCHAR2,

p\_dob IN DATE,

p\_balance IN NUMBER

);

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

END CustomerManagement;

/

CREATE OR REPLACE PACKAGE BODY CustomerManagement AS

PROCEDURE AddCustomer (

p\_customer\_id IN NUMBER,

p\_name IN VARCHAR2,

p\_dob IN DATE,

p\_balance IN NUMBER

)

IS

BEGIN

-- Insert customer into Customers table

END AddCustomer;

PROCEDURE UpdateCustomer (

p\_customer\_id IN NUMBER,

p\_name IN VARCHAR2,

p\_dob IN DATE,

p\_balance IN NUMBER

)

IS

BEGIN

-- Update customer details in Customers table

END UpdateCustomer;

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

IS

v\_balance NUMBER;

BEGIN

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

RETURN v\_balance;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN 0; -- Or handle the exception appropriately

END;

END CustomerManagement;

/

Scenario 2: Employee Management Package

CREATE OR REPLACE PACKAGE EmployeeManagement AS

PROCEDURE HireEmployee (

p\_employee\_id IN NUMBER,

p\_name IN VARCHAR2,

p\_position IN VARCHAR2,

p\_salary IN NUMBER,

p\_department IN VARCHAR2,

p\_hire\_date IN DATE

);

PROCEDURE UpdateEmployee (

p\_employee\_id IN NUMBER,

p\_name IN VARCHAR2,

p\_position IN VARCHAR2,

p\_salary IN NUMBER,

p\_department IN VARCHAR2

);

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

END EmployeeManagement;

/

CREATE OR REPLACE PACKAGE BODY EmployeeManagement AS

PROCEDURE HireEmployee (

-- ... procedure body ...

)

IS

BEGIN

-- Insert employee into Employees table

END HireEmployee;

PROCEDURE UpdateEmployee (

-- ... procedure body ...

)

IS

BEGIN

-- Update employee details in Employees table

END UpdateEmployee;

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

IS

v\_salary NUMBER;

BEGIN

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

RETURN v\_salary \* 12; -- Assuming 12 months in a year

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN 0; -- Or handle the exception appropriately

END;

END EmployeeManagement;

/

Scenario 3: Account Operations Package

CREATE OR REPLACE PACKAGE AccountOperations AS

PROCEDURE OpenAccount (

p\_customer\_id IN NUMBER,

p\_account\_type IN VARCHAR2,

p\_initial\_balance IN NUMBER

);

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

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

END AccountOperations;

/

CREATE OR REPLACE PACKAGE BODY AccountOperations AS

PROCEDURE OpenAccount (

-- ... procedure body ...

)

IS

BEGIN

-- Insert new account into Accounts table

END OpenAccount;

PROCEDURE CloseAccount (

-- ... procedure body ...

)

IS

BEGIN

-- Update account status to closed in Accounts table

END CloseAccount;

FUNCTION GetCustomerTotalBalance (p\_customer\_id IN 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;

END AccountOperations;

/