# 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.

# Answer:

CREATE OR REPLACE PACKAGE CustomerManagement AS PROCEDURE AddNewCustomer(

p\_customer\_id IN NUMBER, p\_name IN VARCHAR2,

p\_dob IN DATE, p\_balance IN NUMBER

);

PROCEDURE UpdateCustomerDetails( p\_customer\_id IN NUMBER, p\_name IN VARCHAR2,

p\_dob IN DATE

);

FUNCTION GetCustomerBalance( p\_customer\_id IN NUMBER

) *RETURN* NUMBER;

END CustomerManagement;

/

CREATE OR REPLACE PACKAGE BODY CustomerManagement AS PROCEDURE AddNewCustomer(

p\_customer\_id IN NUMBER, p\_name IN VARCHAR2, p\_dob IN DATE,

p\_balance IN NUMBER

## ) IS BEGIN

INSERT INTO Customers (CustomerID, Name, DOB, Balance, LastModified) VALUES (p\_customer\_id, p\_name, p\_dob, p\_balance, SYSDATE);

## COMMIT;

END AddNewCustomer;

PROCEDURE UpdateCustomerDetails( p\_customer\_id IN NUMBER, p\_name IN VARCHAR2,

p\_dob IN DATE

## ) IS BEGIN

UPDATE Customers

SET Name = p\_name, DOB = p\_dob, LastModified = SYSDATE WHERE CustomerID = p\_customer\_id;

## COMMIT;

END UpdateCustomerDetails; 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* NULL; *-- Handle case where customer is not found*

END GetCustomerBalance; END CustomerManagement;

/

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.

# Answer:

CREATE OR REPLACE PACKAGE EmployeeManagement AS PROCEDURE HireNewEmployee(

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 UpdateEmployeeDetails( 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 HireNewEmployee(

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

## ) IS BEGIN

INSERT INTO Employees (EmployeeID, Name, Position, Salary, Department, HireDate) VALUES (p\_employee\_id, p\_name, p\_position, p\_salary, p\_department, p\_hire\_date); COMMIT;

END HireNewEmployee;

PROCEDURE UpdateEmployeeDetails( p\_employee\_id IN NUMBER, p\_name IN VARCHAR2,

p\_position IN VARCHAR2, p\_salary IN NUMBER, p\_department IN VARCHAR2

## ) IS BEGIN

UPDATE Employees

SET Name = p\_name, Position = p\_position, Salary = p\_salary, Department = p\_department

WHERE EmployeeID = p\_employee\_id; COMMIT;

END UpdateEmployeeDetails;

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; *-- Annual salary*

## EXCEPTION

*WHEN* NO\_DATA\_FOUND *THEN*

*RETURN* NULL; *-- Handle case where employee is not found*

END CalculateAnnualSalary;

END EmployeeManagement;

/

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.

# Answer:

CREATE OR REPLACE PACKAGE AccountOperations AS PROCEDURE OpenNewAccount(

p\_account\_id IN NUMBER, p\_customer\_id IN NUMBER, p\_account\_type IN VARCHAR2, p\_balance IN NUMBER

);

PROCEDURE CloseAccount( p\_account\_id IN NUMBER

);

FUNCTION GetTotalBalance( p\_customer\_id IN NUMBER

) *RETURN* NUMBER;

END AccountOperations;

/

CREATE OR REPLACE PACKAGE BODY AccountOperations AS PROCEDURE OpenNewAccount(

p\_account\_id IN NUMBER, p\_customer\_id IN NUMBER, p\_account\_type IN VARCHAR2, p\_balance IN NUMBER

## ) IS BEGIN

INSERT INTO Accounts (AccountID, CustomerID, AccountType, Balance, LastModified) VALUES (p\_account\_id, p\_customer\_id, p\_account\_type, p\_balance, SYSDATE); COMMIT;

END OpenNewAccount;

PROCEDURE CloseAccount( p\_account\_id IN NUMBER

## ) IS

BEGIN

DELETE FROM Accounts

WHERE AccountID = p\_account\_id; COMMIT;

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 CustomerID = p\_customer\_id;

*RETURN* v\_total\_balance; EXCEPTION

*WHEN* NO\_DATA\_FOUND *THEN*

*RETURN* 0; *-- Handle case where no accounts are found*

END GetTotalBalance; END AccountOperations;

/