-- Exercise 7: Packages

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

CREATE OR REPLACE PACKAGE CustomerManagement AS

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

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

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

END CustomerManagement;

CREATE OR REPLACE PACKAGE BODY CustomerManagement AS

PROCEDURE AddNewCustomer(p\_customer\_id NUMBER, p\_name VARCHAR2, p\_dob DATE, p\_initial\_balance NUMBER)

IS

BEGIN

INSERT INTO Customers (CustomerID, Name, DateOfBirth)

VALUES (p\_customer\_id, p\_name, p\_dob);

INSERT INTO Accounts (AccountID, CustomerID, Balance)

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

END AddNewCustomer;

PROCEDURE UpdateCustomerDetails(p\_customer\_id NUMBER, p\_name VARCHAR2, p\_dob DATE)

IS

BEGIN

UPDATE Customers

SET Name = p\_name, DateOfBirth = p\_dob

WHERE CustomerID = p\_customer\_id;

END UpdateCustomerDetails;

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

IS

v\_balance NUMBER;

BEGIN

SELECT SUM(Balance)

INTO v\_balance

FROM Accounts

WHERE CustomerID = p\_customer\_id;

RETURN v\_balance;

END GetCustomerBalance;

END CustomerManagement;

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

CREATE OR REPLACE PACKAGE EmployeeManagement AS

PROCEDURE HireNewEmployee(p\_employee\_id NUMBER, p\_name VARCHAR2, p\_dob DATE, p\_salary NUMBER);

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

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

END EmployeeManagement;

CREATE OR REPLACE PACKAGE BODY EmployeeManagement AS

PROCEDURE HireNewEmployee(p\_employee\_id NUMBER, p\_name VARCHAR2, p\_dob DATE, p\_salary NUMBER)

IS

BEGIN

INSERT INTO Employees (EmployeeID, Name, DateOfBirth, Salary)

VALUES (p\_employee\_id, p\_name, p\_dob, p\_salary);

END HireNewEmployee;

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

IS

BEGIN

UPDATE Employees

SET Name = p\_name, DateOfBirth = p\_dob, 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;

END CalculateAnnualSalary;

END EmployeeManagement;

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

CREATE OR REPLACE PACKAGE AccountOperations AS

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

PROCEDURE CloseAccount(p\_account\_id NUMBER);

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

END AccountOperations;

CREATE OR REPLACE PACKAGE BODY AccountOperations AS

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

IS

BEGIN

INSERT INTO Accounts (AccountID, CustomerID, Balance)

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

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\_balance NUMBER;

BEGIN

SELECT SUM(Balance)

INTO v\_balance

FROM Accounts

WHERE CustomerID = p\_customer\_id;

RETURN v\_balance;

END GetTotalBalance;

END AccountOperations;