**Exercise 7: Packages**

CREATE OR REPLACE PACKAGE CustomerManagement IS

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

  PROCEDURE UpdateCustomer(p\_id NUMBER, p\_name VARCHAR2);

  FUNCTION GetCustomerBalance(p\_id NUMBER) RETURN NUMBER;

END CustomerManagement;

/

CREATE OR REPLACE PACKAGE BODY CustomerManagement IS

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

  BEGIN

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

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

  END;

  PROCEDURE UpdateCustomer(p\_id NUMBER, p\_name VARCHAR2) IS

  BEGIN

    UPDATE Customers

    SET Name = p\_name, LastModified = SYSDATE

    WHERE CustomerID = p\_id;

  END;

  FUNCTION GetCustomerBalance(p\_id NUMBER) RETURN NUMBER IS

    v\_balance NUMBER;

  BEGIN

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

    RETURN v\_balance;

  END;

END CustomerManagement;

/

CREATE OR REPLACE PACKAGE EmployeeManagement IS

  PROCEDURE HireEmployee(p\_id NUMBER, p\_name VARCHAR2, p\_position VARCHAR2, p\_salary NUMBER, p\_dept VARCHAR2, p\_hiredate DATE);

  PROCEDURE UpdateEmployee(p\_id NUMBER, p\_salary NUMBER);

  FUNCTION CalculateAnnualSalary(p\_id NUMBER) RETURN NUMBER;

END EmployeeManagement;

/

CREATE OR REPLACE PACKAGE BODY EmployeeManagement IS

  PROCEDURE HireEmployee(p\_id NUMBER, p\_name VARCHAR2, p\_position VARCHAR2, p\_salary NUMBER, p\_dept VARCHAR2, p\_hiredate DATE) IS

  BEGIN

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

    VALUES (p\_id, p\_name, p\_position, p\_salary, p\_dept, p\_hiredate);

  END;

  PROCEDURE UpdateEmployee(p\_id NUMBER, p\_salary NUMBER) IS

  BEGIN

    UPDATE Employees

    SET Salary = p\_salary

    WHERE EmployeeID = p\_id;

  END;

  FUNCTION CalculateAnnualSalary(p\_id NUMBER) RETURN NUMBER IS

    v\_salary NUMBER;

  BEGIN

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

    RETURN v\_salary \* 12;

  END;

END EmployeeManagement;

/

CREATE OR REPLACE PACKAGE AccountOperations IS

  PROCEDURE OpenAccount(p\_id NUMBER, p\_customer\_id NUMBER, p\_type VARCHAR2, p\_balance NUMBER);

  PROCEDURE CloseAccount(p\_id NUMBER);

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

END AccountOperations;

/

CREATE OR REPLACE PACKAGE BODY AccountOperations IS

  PROCEDURE OpenAccount(p\_id NUMBER, p\_customer\_id NUMBER, p\_type VARCHAR2, p\_balance NUMBER) IS

  BEGIN

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

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

  END;

  PROCEDURE CloseAccount(p\_id NUMBER) IS

  BEGIN

    DELETE FROM Accounts WHERE AccountID = p\_id;

  END;

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

    v\_total NUMBER;

  BEGIN

    SELECT NVL(SUM(Balance), 0)

    INTO v\_total

    FROM Accounts

    WHERE CustomerID = p\_customer\_id;

    RETURN v\_total;

  END;

END AccountOperations;

/

BEGIN

  CustomerManagement.AddCustomer(3, 'New Person', TO\_DATE('1995-02-10', 'YYYY-MM-DD'), 3000);

END;

/

SELECT CustomerManagement.GetCustomerBalance(3) FROM dual;

BEGIN

  EmployeeManagement.HireEmployee(3, 'Charlie', 'Tester', 40000, 'QA', SYSDATE);

END;

/

BEGIN

  AccountOperations.OpenAccount(3, 3, 'Savings', 2000);

END;

/

SELECT \* FROM Customers;

SELECT CustomerManagement.GetCustomerBalance(1) AS Balance FROM dual;

SELECT \* FROM Employees;

SELECT EmployeeManagement.CalculateAnnualSalary(1) AS Annual\_Salary FROM dual;

SELECT \* FROM Accounts;

SELECT AccountOperations.GetTotalBalance(1) AS Total\_Balance FROM dual;

























