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.**

| CREATE OR REPLACE PACKAGE CustomerManagement AS  PROCEDURE AddCustomer(p\_CustomerID NUMBER, p\_Name VARCHAR2, p\_DOB DATE, p\_Balance NUMBER);  PROCEDURE UpdateCustomer(p\_CustomerID NUMBER, p\_Name VARCHAR2, p\_DOB DATE, p\_Balance NUMBER);  FUNCTION GetCustomerBalance(p\_CustomerID NUMBER) RETURN NUMBER; END CustomerManagement;   CREATE OR REPLACE PACKAGE BODY CustomerManagement AS  PROCEDURE AddCustomer(p\_CustomerID NUMBER, p\_Name VARCHAR2, p\_DOB DATE, p\_Balance NUMBER) IS  BEGIN  INSERT INTO Customers (CustomerID, Name, DOB, Balance, LastModified)  VALUES (p\_CustomerID, p\_Name, p\_DOB, p\_Balance, SYSDATE);  EXCEPTION  WHEN DUP\_VAL\_ON\_INDEX THEN  DBMS\_OUTPUT.PUT\_LINE('Customer with this ID already exists.');  END AddCustomer;    PROCEDURE UpdateCustomer(p\_CustomerID NUMBER, p\_Name VARCHAR2, p\_DOB DATE, p\_Balance NUMBER) IS  BEGIN  UPDATE Customers  SET Name = p\_Name, DOB = p\_DOB, Balance = p\_Balance, LastModified = SYSDATE  WHERE CustomerID = p\_CustomerID;  IF SQL%ROWCOUNT = 0 THEN  DBMS\_OUTPUT.PUT\_LINE('Customer not found.');  END IF;  END UpdateCustomer;    FUNCTION GetCustomerBalance(p\_CustomerID NUMBER) RETURN NUMBER IS  v\_balance NUMBER;  BEGIN  SELECT Balance INTO v\_balance  FROM Customers  WHERE CustomerID = p\_CustomerID;  RETURN v\_balance;  EXCEPTION  WHEN NO\_DATA\_FOUND THEN  RETURN NULL;  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.**

| CREATE OR REPLACE PACKAGE EmployeeManagement AS  PROCEDURE HireEmployee(p\_EmployeeID NUMBER, p\_Name VARCHAR2, p\_Position VARCHAR2, p\_Salary NUMBER, p\_Department VARCHAR2, p\_HireDate DATE);  PROCEDURE UpdateEmployee(p\_EmployeeID NUMBER, p\_Name VARCHAR2, p\_Position VARCHAR2, p\_Salary NUMBER, p\_Department VARCHAR2);  FUNCTION CalculateAnnualSalary(p\_EmployeeID NUMBER) RETURN NUMBER; END EmployeeManagement;   CREATE OR REPLACE PACKAGE BODY EmployeeManagement AS  PROCEDURE HireEmployee(p\_EmployeeID NUMBER, p\_Name VARCHAR2, p\_Position VARCHAR2, p\_Salary NUMBER, p\_Department VARCHAR2, p\_HireDate DATE) IS  BEGIN  INSERT INTO Employees (EmployeeID, Name, Position, Salary, Department, HireDate)  VALUES (p\_EmployeeID, p\_Name, p\_Position, p\_Salary, p\_Department, p\_HireDate);  EXCEPTION  WHEN DUP\_VAL\_ON\_INDEX THEN  DBMS\_OUTPUT.PUT\_LINE('Employee with this ID already exists.');  END HireEmployee;    PROCEDURE UpdateEmployee(p\_EmployeeID NUMBER, p\_Name VARCHAR2, p\_Position VARCHAR2, p\_Salary NUMBER, p\_Department VARCHAR2) IS  BEGIN  UPDATE Employees  SET Name = p\_Name, Position = p\_Position, Salary = p\_Salary, Department = p\_Department  WHERE EmployeeID = p\_EmployeeID;  IF SQL%ROWCOUNT = 0 THEN  DBMS\_OUTPUT.PUT\_LINE('Employee not found.');  END IF;  END UpdateEmployee;    FUNCTION CalculateAnnualSalary(p\_EmployeeID NUMBER) RETURN NUMBER IS  v\_salary NUMBER;  BEGIN  SELECT Salary INTO v\_salary  FROM Employees  WHERE EmployeeID = p\_EmployeeID;  RETURN v\_salary \* 12;  EXCEPTION  WHEN NO\_DATA\_FOUND THEN  RETURN NULL;  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.**

| CREATE OR REPLACE PACKAGE AccountOperations AS  PROCEDURE OpenAccount(p\_AccountID NUMBER, p\_CustomerID NUMBER, p\_AccountType VARCHAR2, p\_Balance NUMBER);  PROCEDURE CloseAccount(p\_AccountID NUMBER);  FUNCTION GetTotalBalance(p\_CustomerID NUMBER) RETURN NUMBER; END AccountOperations;   CREATE OR REPLACE PACKAGE BODY AccountOperations AS  PROCEDURE OpenAccount(p\_AccountID NUMBER, p\_CustomerID NUMBER, p\_AccountType VARCHAR2, p\_Balance NUMBER) IS  BEGIN  INSERT INTO Accounts (AccountID, CustomerID, AccountType, Balance, LastModified)  VALUES (p\_AccountID, p\_CustomerID, p\_AccountType, p\_Balance, SYSDATE);  EXCEPTION  WHEN DUP\_VAL\_ON\_INDEX THEN  DBMS\_OUTPUT.PUT\_LINE('Account with this ID already exists.');  END OpenAccount;    PROCEDURE CloseAccount(p\_AccountID NUMBER) IS  BEGIN  DELETE FROM Accounts  WHERE AccountID = p\_AccountID;  IF SQL%ROWCOUNT = 0 THEN  DBMS\_OUTPUT.PUT\_LINE('Account not found.');  END IF;  END CloseAccount;    FUNCTION GetTotalBalance(p\_CustomerID NUMBER) RETURN NUMBER IS  v\_totalBalance NUMBER;  BEGIN  SELECT SUM(Balance) INTO v\_totalBalance  FROM Accounts  WHERE CustomerID = p\_CustomerID;  RETURN v\_totalBalance;  EXCEPTION  WHEN NO\_DATA\_FOUND THEN  RETURN 0;  END GetTotalBalance; END AccountOperations; |
| --- |