**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.

**Package:**

SQL> CREATE OR REPLACE PACKAGE CustomerManagement AS

2 PROCEDURE AddCustomer(p\_CustomerID NUMBER, p\_Name VARCHAR2, p\_DOB DATE, p\_Balance NUMBER);

3 PROCEDURE UpdateCustomer(p\_CustomerID NUMBER, p\_Name VARCHAR2, p\_DOB DATE, p\_Balance NUMBER);

4 FUNCTION GetCustomerBalance(p\_CustomerID NUMBER) RETURN NUMBER;

5 END CustomerManagement;

6 /

**Package Body:**

SQL> CREATE OR REPLACE PACKAGE BODY CustomerManagement AS

2 PROCEDURE AddCustomer(p\_CustomerID NUMBER, p\_Name VARCHAR2, p\_DOB DATE, p\_Balance NUMBER) IS

3 BEGIN

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

5 VALUES (p\_CustomerID, p\_Name, p\_DOB, p\_Balance, SYSDATE);

6 END AddCustomer;

7

8 PROCEDURE UpdateCustomer(p\_CustomerID NUMBER, p\_Name VARCHAR2, p\_DOB DATE, p\_Balance NUMBER) IS

9 BEGIN

10 UPDATE Customers

11 SET Name = p\_Name, DOB = p\_DOB, Balance = p\_Balance, LastModified = SYSDATE

12 WHERE CustomerID = p\_CustomerID;

13 END UpdateCustomer;

14

15 FUNCTION GetCustomerBalance(p\_CustomerID NUMBER) RETURN NUMBER IS

16 v\_Balance NUMBER;

17 BEGIN

18 SELECT Balance INTO v\_Balance

19 FROM Customers

20 WHERE CustomerID = p\_CustomerID;

21 RETURN v\_Balance;

22 END GetCustomerBalance;

23 END CustomerManagement;

24 /

**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.

**Package:**

SQL> CREATE OR REPLACE PACKAGE EmployeeManagement AS

2 PROCEDURE HireEmployee(p\_EmployeeID NUMBER, p\_Name VARCHAR2, p\_Position VARCHAR2, p\_Salary NUMBER, p\_Department VARCHAR2, p\_HireDate DATE);

3 PROCEDURE UpdateEmployee(p\_EmployeeID NUMBER, p\_Name VARCHAR2, p\_Position VARCHAR2, p\_Salary NUMBER, p\_Department VARCHAR2);

4 FUNCTION CalculateAnnualSalary(p\_EmployeeID NUMBER) RETURN NUMBER;

5 END EmployeeManagement;

6 /

**Package Body:**

SQL> CREATE OR REPLACE PACKAGE BODY EmployeeManagement AS

2 PROCEDURE HireEmployee(p\_EmployeeID NUMBER, p\_Name VARCHAR2, p\_Position VARCHAR2, p\_Salary NUMBER, p\_Department VARCHAR2, p\_HireDate DATE) IS

3 BEGIN

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

5 VALUES (p\_EmployeeID, p\_Name, p\_Position, p\_Salary, p\_Department, p\_HireDate);

6 END HireEmployee;

7

8 PROCEDURE UpdateEmployee(p\_EmployeeID NUMBER, p\_Name VARCHAR2, p\_Position VARCHAR2, p\_Salary NUMBER, p\_Department VARCHAR2) IS

9 BEGIN

10 UPDATE Employees

11 SET Name = p\_Name, Position = p\_Position, Salary = p\_Salary, Department = p\_Department

12 WHERE EmployeeID = p\_EmployeeID;

13 END UpdateEmployee;

14

15 FUNCTION CalculateAnnualSalary(p\_EmployeeID NUMBER) RETURN NUMBER IS

16 v\_Salary NUMBER;

17 BEGIN

18 SELECT Salary \* 12 INTO v\_Salary

19 FROM Employees

20 WHERE EmployeeID = p\_EmployeeID;

21 RETURN v\_Salary;

22 END CalculateAnnualSalary;

23 END EmployeeManagement;

24 /

**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.

**Package:**

SQL> CREATE OR REPLACE PACKAGE AccountOperations AS

2 PROCEDURE OpenAccount(p\_AccountID NUMBER, p\_CustomerID NUMBER, p\_AccountType VARCHAR2, p\_Balance NUMBER);

3 PROCEDURE CloseAccount(p\_AccountID NUMBER);

4 FUNCTION GetTotalBalance(p\_CustomerID NUMBER) RETURN NUMBER;

5 END AccountOperations;

6 /

**Package Body:**

SQL> CREATE OR REPLACE PACKAGE BODY AccountOperations AS

2 PROCEDURE OpenAccount(p\_AccountID NUMBER, p\_CustomerID NUMBER, p\_AccountType VARCHAR2, p\_Balance NUMBER) IS

3 BEGIN

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

5 VALUES (p\_AccountID, p\_CustomerID, p\_AccountType, p\_Balance, SYSDATE);

6 END OpenAccount;

7

8 PROCEDURE CloseAccount(p\_AccountID NUMBER) IS

9 BEGIN

10 DELETE FROM Accounts

11 WHERE AccountID = p\_AccountID;

12 END CloseAccount;

13

14 FUNCTION GetTotalBalance(p\_CustomerID NUMBER) RETURN NUMBER IS

15 v\_TotalBalance NUMBER;

16 BEGIN

17 SELECT SUM(Balance) INTO v\_TotalBalance

18 FROM Accounts

19 WHERE CustomerID = p\_CustomerID;

20 RETURN v\_TotalBalance;

21 END GetTotalBalance;

22 END AccountOperations;

23 /