**Exercise 7: Packages**

**Scenario 3:**

**CODE:**

**-- Creating Accounts table**

CREATE TABLE Accounts (

AccountID NUMBER PRIMARY KEY,

CustomerID NUMBER,

AccountType VARCHAR2(20),

Balance NUMBER,

LastModified DATE,

FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)

);

**-- Inserting Values into Accounts table**

BEGIN

INSERT INTO Accounts (AccountID, CustomerID, AccountType, Balance, LastModified) VALUES (1, 1, 'Savings', 1200, SYSDATE);

INSERT INTO Accounts (AccountID, CustomerID, AccountType, Balance, LastModified) VALUES (2, 2, 'Checking', 800, SYSDATE);

INSERT INTO Accounts (AccountID, CustomerID, AccountType, Balance, LastModified) VALUES (3, 3, 'Savings', 15000, SYSDATE);

INSERT INTO Accounts (AccountID, CustomerID, AccountType, Balance, LastModified) VALUES (4, 4, 'Checking', 9500, SYSDATE);

INSERT INTO Accounts (AccountID, CustomerID, AccountType, Balance, LastModified) VALUES (5, 5, 'Savings', 20000, SYSDATE);

END;

CREATE OR REPLACE PACKAGE AccountOperations AS

PROCEDURE OpenAccount(p\_AccountID IN NUMBER,p\_CustomerID IN NUMBER,

p\_AccountType IN VARCHAR2,p\_InitialBalance IN NUMBER);

PROCEDURE CloseAccount(p\_AccountID IN NUMBER);

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

END AccountOperations;

/

CREATE OR REPLACE PACKAGE BODY AccountOperations AS

PROCEDURE OpenAccount(p\_AccountID IN NUMBER,p\_CustomerID IN NUMBER,p\_AccountType IN VARCHAR2,p\_InitialBalance IN NUMBER) IS

BEGIN

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

VALUES (p\_AccountID, p\_CustomerID, p\_AccountType, p\_InitialBalance, SYSDATE);

DBMS\_OUTPUT.PUT\_LINE('Account opened successfully.');

END OpenAccount;

PROCEDURE CloseAccount(p\_AccountID IN NUMBER) IS

BEGIN

DELETE FROM Accounts WHERE AccountID = p\_AccountID;

DBMS\_OUTPUT.PUT\_LINE('Account closed successfully.');

END CloseAccount;

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

v\_TotalBalance NUMBER := 0;

BEGIN

SELECT NVL(SUM(Balance), 0)

INTO v\_TotalBalance

FROM Accounts

WHERE CustomerID = p\_CustomerID;

RETURN v\_TotalBalance;

END GetTotalBalance;

END AccountOperations;

/

**-- Open a new account**

BEGIN

AccountOperations.OpenAccount(101, 1, 'Savings', 5000);

END;

/

**-- Close an account**

BEGIN

AccountOperations.CloseAccount(101);

END;

/

**-- Get total balance of a customer**

DECLARE

v\_balance NUMBER;

BEGIN

v\_balance := AccountOperations.GetTotalBalance(1);

DBMS\_OUTPUT.PUT\_LINE('Total Balance of customer with ID 1 is: ' || v\_balance);

END;

/

**OUTPUT:**

Output for **OpenAccount procedure** is:

A screenshot of a computer

AI-generated content may be incorrect.

Output for **CloseAccount procedure** is:

A screenshot of a computer

AI-generated content may be incorrect.

Output for **GetTotalBalance function** is:

A screenshot of a computer

AI-generated content may be incorrect.