**EXERCISE 7-PACKAGES**

**SCENARIO 1: CustomerManagement Package**

**PACKAGE SPECIFICATION:**

CREATE OR REPLACE PACKAGE CustomerManagement AS

PROCEDURE AddCustomer (

p\_custid NUMBER,

p\_name VARCHAR2,

p\_dob DATE,

p\_balance NUMBER

);

PROCEDURE UpdateCustomer (

p\_custid NUMBER,

p\_name VARCHAR2,

p\_balance NUMBER

);

FUNCTION GetCustomerBalance (

p\_custid NUMBER

) RETURN NUMBER;

END CustomerManagement;

/

**PACKAGE BODY:**

CREATE OR REPLACE PACKAGE BODY CustomerManagement AS

PROCEDURE AddCustomer (

p\_custid NUMBER,

p\_name VARCHAR2,

p\_dob DATE,

p\_balance NUMBER

) IS

BEGIN

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

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

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Customer added successfully.');

EXCEPTION

WHEN DUP\_VAL\_ON\_INDEX THEN

DBMS\_OUTPUT.PUT\_LINE('Error: Customer ID already exists.');

END;

PROCEDURE UpdateCustomer (

p\_custid NUMBER,

p\_name VARCHAR2,

p\_balance NUMBER

) IS

BEGIN

UPDATE Customers

SET Name = p\_name,

Balance = p\_balance,

LastModified = SYSDATE

WHERE CustomerID = p\_custid;

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Customer updated successfully.');

END;

FUNCTION GetCustomerBalance (

p\_custid NUMBER

) RETURN NUMBER IS

v\_balance NUMBER;

BEGIN

SELECT Balance INTO v\_balance

FROM Customers

WHERE CustomerID = p\_custid;

RETURN v\_balance;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

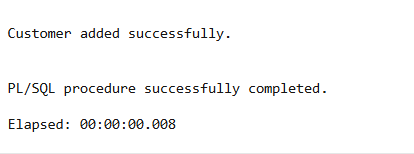
RETURN NULL;

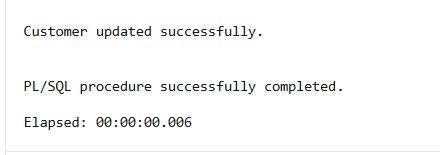
END;

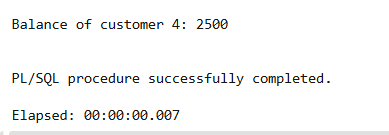
END CustomerManagement;

/

**OUTPUT:**







**SCENARIO 2:** **EmployeeManagement Package**

**PACKAGE SPECIFICATION:**

CREATE OR REPLACE PACKAGE EmployeeManagement AS

PROCEDURE HireEmployee (

p\_empid NUMBER,

p\_name VARCHAR2,

p\_position VARCHAR2,

p\_salary NUMBER,

p\_department VARCHAR2

);

PROCEDURE UpdateEmployee (

p\_empid NUMBER,

p\_salary NUMBER

);

FUNCTION AnnualSalary (

p\_empid NUMBER

) RETURN NUMBER;

END EmployeeManagement;

/

**PACKAGE BODY:**

CREATE OR REPLACE PACKAGE BODY EmployeeManagement AS

PROCEDURE HireEmployee (

p\_empid NUMBER,

p\_name VARCHAR2,

p\_position VARCHAR2,

p\_salary NUMBER,

p\_department VARCHAR2

) IS

BEGIN

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

VALUES (p\_empid, p\_name, p\_position, p\_salary, p\_department, SYSDATE);

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Employee hired successfully.');

EXCEPTION

WHEN DUP\_VAL\_ON\_INDEX THEN

DBMS\_OUTPUT.PUT\_LINE('Error: Employee ID already exists.');

END;

PROCEDURE UpdateEmployee (

p\_empid NUMBER,

p\_salary NUMBER

) IS

BEGIN

UPDATE Employees

SET Salary = p\_salary

WHERE EmployeeID = p\_empid;

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Employee salary updated.');

END;

FUNCTION AnnualSalary (

p\_empid NUMBER

) RETURN NUMBER IS

v\_salary NUMBER;

BEGIN

SELECT Salary INTO v\_salary

FROM Employees

WHERE EmployeeID = p\_empid;

RETURN v\_salary \* 12;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

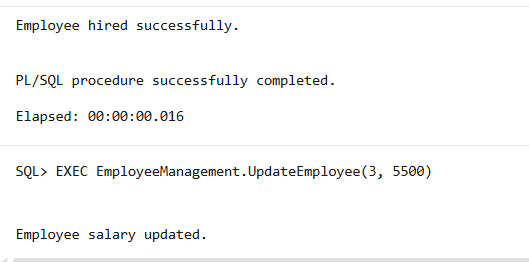
RETURN NULL;

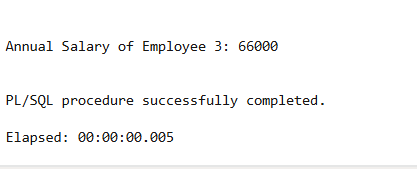
END;

END EmployeeManagement;

/

**OUTPUT:**

****

****

**SCENARIO 3:** **AccountOperations Package**

**PACKAGE SPECIFICATION:**

CREATE OR REPLACE PACKAGE AccountOperations AS

PROCEDURE OpenAccount (

p\_accid NUMBER,

p\_custid NUMBER,

p\_type VARCHAR2,

p\_balance NUMBER

);

PROCEDURE CloseAccount (

p\_accid NUMBER

);

FUNCTION TotalBalance (

p\_custid NUMBER

) RETURN NUMBER;

END AccountOperations;

/

**PACKAGE BODY:**

CREATE OR REPLACE PACKAGE BODY AccountOperations AS

PROCEDURE OpenAccount (

p\_accid NUMBER,

p\_custid NUMBER,

p\_type VARCHAR2,

p\_balance NUMBER

) IS

BEGIN

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

VALUES (p\_accid, p\_custid, p\_type, p\_balance, SYSDATE);

COMMIT;

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

END;

PROCEDURE CloseAccount (

p\_accid NUMBER

) IS

BEGIN

DELETE FROM Accounts

WHERE AccountID = p\_accid;

COMMIT;

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

END;

FUNCTION TotalBalance (

p\_custid NUMBER

) RETURN NUMBER IS

v\_total NUMBER;

BEGIN

SELECT SUM(Balance) INTO v\_total

FROM Accounts

WHERE CustomerID = p\_custid;

RETURN v\_total;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN 0;

END;

END AccountOperations;

/

**OUTPUT:**

