**WEEK-2**

**MODULE -3**

**PL/SQL PROGRAMMING**

**SUPERSET ID:6407550**

**Exercise:7-Packages**

**Scenario:1**

**Package Specification**

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;

/

**Package Body**

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);

DBMS\_OUTPUT.PUT\_LINE('✅ Customer added: ' || p\_Name);

EXCEPTION

WHEN DUP\_VAL\_ON\_INDEX THEN

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

END;

PROCEDURE UpdateCustomer(p\_ID NUMBER, p\_Name VARCHAR2) IS

BEGIN

UPDATE Customers SET Name = p\_Name, LastModified = SYSDATE

WHERE CustomerID = p\_ID;

DBMS\_OUTPUT.PUT\_LINE('🔄 Customer updated: ' || p\_Name);

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;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

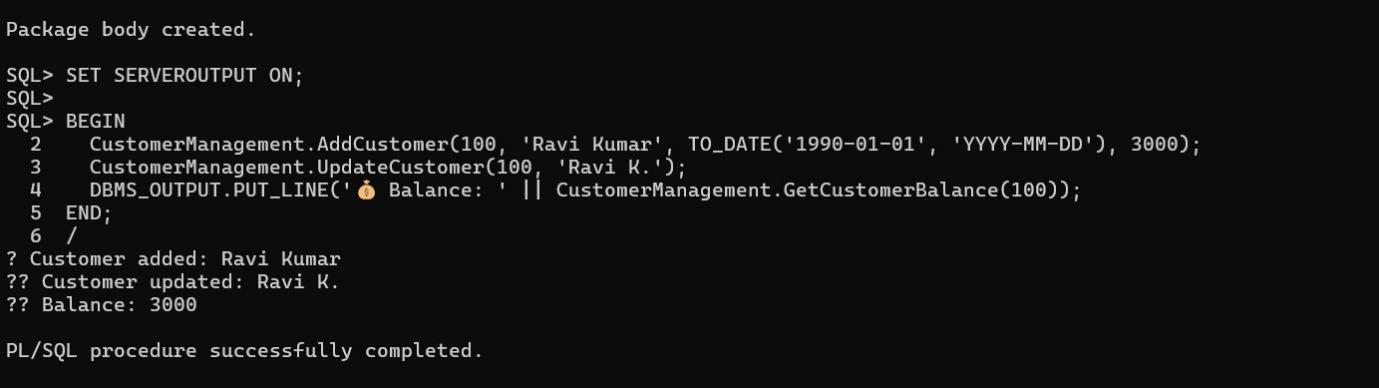
RETURN NULL;

END;

END CustomerManagement;

/

**OUTPUT:**

****

**Scenario:2**

**Pacakage specification**

CREATE OR REPLACE PACKAGE EmployeeManagement IS

PROCEDURE HireEmployee(p\_ID NUMBER, p\_Name VARCHAR2, p\_Position VARCHAR2, p\_Salary NUMBER, p\_Dept VARCHAR2);

PROCEDURE UpdateEmployee(p\_ID NUMBER, p\_Salary NUMBER);

FUNCTION GetAnnualSalary(p\_ID NUMBER) RETURN NUMBER;

END EmployeeManagement;

/

**Package Body**

CREATE OR REPLACE PACKAGE BODY EmployeeManagement IS

PROCEDURE HireEmployee(p\_ID NUMBER, p\_Name VARCHAR2, p\_Position VARCHAR2, p\_Salary NUMBER, p\_Dept VARCHAR2) IS

BEGIN

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

VALUES (p\_ID, p\_Name, p\_Position, p\_Salary, p\_Dept, SYSDATE);

DBMS\_OUTPUT.PUT\_LINE('✅ Employee hired: ' || p\_Name);

END;

PROCEDURE UpdateEmployee(p\_ID NUMBER, p\_Salary NUMBER) IS

BEGIN

UPDATE Employees SET Salary = p\_Salary WHERE EmployeeID = p\_ID;

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

END;

FUNCTION GetAnnualSalary(p\_ID NUMBER) RETURN NUMBER IS

v\_Salary NUMBER;

BEGIN

SELECT Salary \* 12 INTO v\_Salary FROM Employees WHERE EmployeeID = p\_ID;

RETURN v\_Salary;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

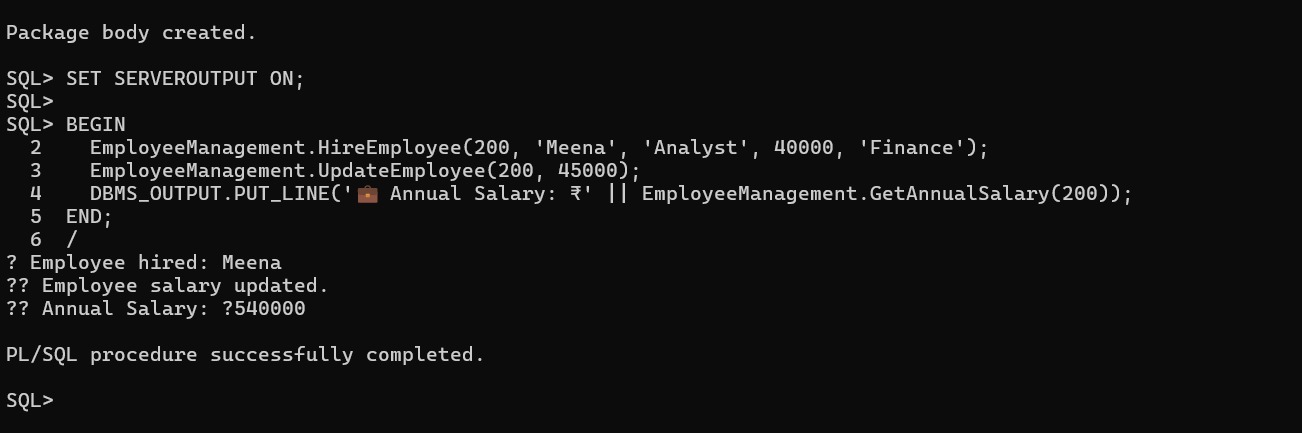
RETURN NULL;

END;

END EmployeeManagement;

/

**OUTPUT:**



**Scenario:3**

**Package Specification**

CREATE OR REPLACE PACKAGE AccountOperations IS

PROCEDURE OpenAccount(p\_ID NUMBER, p\_CustomerID NUMBER, p\_Type VARCHAR2, p\_Balance NUMBER);

PROCEDURE CloseAccount(p\_ID NUMBER);

FUNCTION GetTotalBalance(p\_CustomerID NUMBER) RETURN NUMBER;

END AccountOperations;

/

**Package Body**

CREATE OR REPLACE PACKAGE BODY AccountOperations IS

PROCEDURE OpenAccount(p\_ID NUMBER, p\_CustomerID NUMBER, p\_Type VARCHAR2, p\_Balance NUMBER) IS

BEGIN

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

VALUES (p\_ID, p\_CustomerID, p\_Type, p\_Balance, SYSDATE);

DBMS\_OUTPUT.PUT\_LINE('🏦 Account opened: ' || p\_ID);

END;

PROCEDURE CloseAccount(p\_ID NUMBER) IS

BEGIN

DELETE FROM Accounts WHERE AccountID = p\_ID;

DBMS\_OUTPUT.PUT\_LINE('❌ Account closed: ' || p\_ID);

END;

FUNCTION GetTotalBalance(p\_CustomerID NUMBER) RETURN NUMBER IS

v\_Total NUMBER;

BEGIN

SELECT SUM(Balance) INTO v\_Total FROM Accounts WHERE CustomerID = p\_CustomerID;

RETURN v\_Total;

EXCEPTION

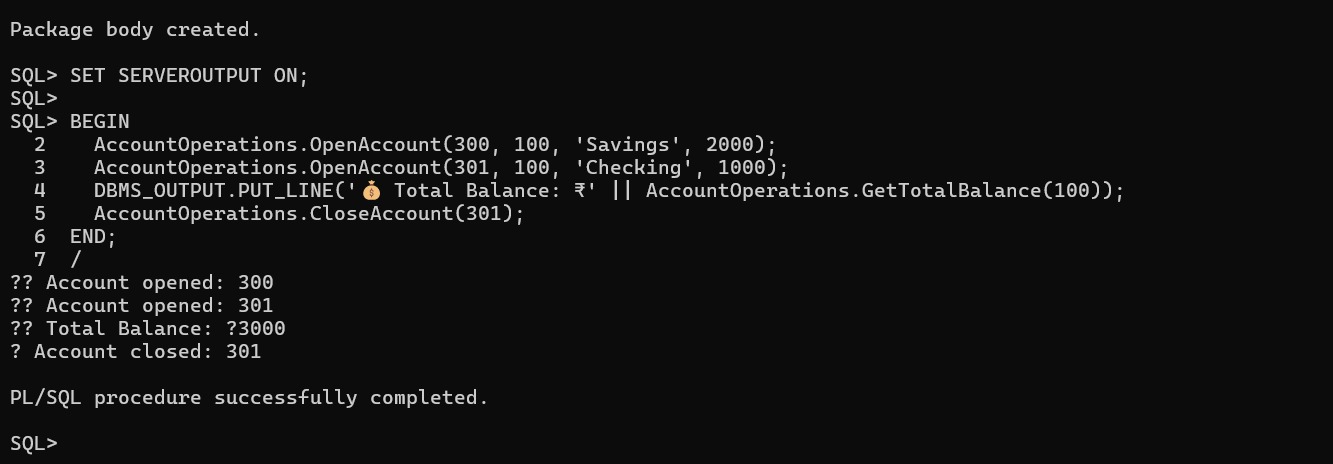
WHEN NO\_DATA\_FOUND THEN RETURN 0;

END;

END AccountOperations;

/

**OUTPUT:**

****