**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.**

**CODE:**

CREATE OR REPLACE PACKAGE CustomerManagement AS

PROCEDURE AddCustomer (

p\_custid NUMBER,

p\_name VARCHAR2,

p\_dob DATE

);

PROCEDURE UpdateCustomer (

p\_custid NUMBER,

p\_name VARCHAR2

);

FUNCTION GetCustomerDOB (

p\_custid NUMBER

) RETURN DATE;

END CustomerManagement;

/

CREATE OR REPLACE PACKAGE BODY CustomerManagement AS

PROCEDURE AddCustomer (

p\_custid NUMBER,

p\_name VARCHAR2,

p\_dob DATE

) IS

BEGIN

INSERT INTO Customers (Customer\_ID, Name, DOB, LastModified)

VALUES (p\_custid, p\_name, p\_dob, 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

) IS

BEGIN

UPDATE Customers

SET Name = p\_name,

LastModified = SYSDATE

WHERE Customer\_ID = p\_custid;

COMMIT;

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

END;

FUNCTION GetCustomerDOB (

p\_custid NUMBER

) RETURN DATE IS

v\_dob DATE;

BEGIN

SELECT DOB INTO v\_dob

FROM Customers

WHERE Customer\_ID = p\_custid;

RETURN v\_dob;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

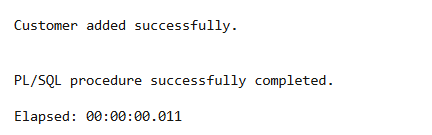
RETURN NULL;

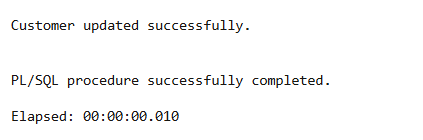
END;

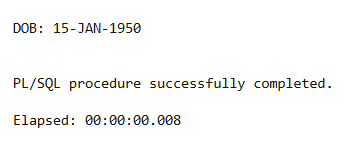
END CustomerManagement;

/

**OUTPUT:**

****

****

****

**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.**

**CODE:**

CREATE TABLE Employees (

Employee\_ID NUMBER PRIMARY KEY,

Name VARCHAR2(100),

Salary NUMBER(10,2),

Department VARCHAR2(50),

HireDate DATE

);

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;

/

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 (Employee\_ID, Name, Salary, Department, HireDate)

VALUES (p\_empid, p\_name, 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 Employee\_ID = 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 Employee\_ID = p\_empid;

RETURN v\_salary \* 12;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN NULL;

END;

END EmployeeManagement;

/

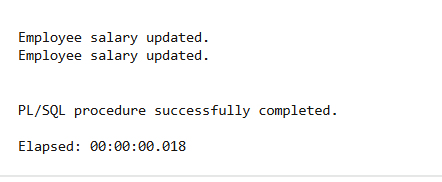
**OUTPUT:**

**BEGIN**

**EmployeeManagement.UpdateEmployee(3, 45000);**

**END;**

**/**

****

**BEGIN**

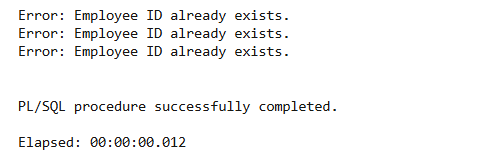
**EmployeeManagement.HireEmployee(1, 'Meena', 'Manager', 60000, 'HR');**

**EmployeeManagement.HireEmployee(2, 'Sri', 'Analyst', 50000, 'Finance');**

**EmployeeManagement.HireEmployee(3, 'Harini', 'Clerk', 40000, 'Admin');**

**END;**

**/**

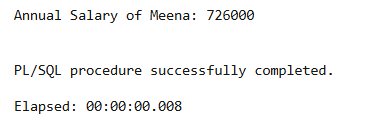
****

**BEGIN**

**DBMS\_OUTPUT.PUT\_LINE(EmployeeManagement.AnnualSalary(1));**

**END;**

**/**

****

**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.**

**CODE:**

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;

/

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:**

