**Scenario 1:** Group all customer-related procedures and functions into a package. Create a package **CustomerManagement** with procedures for adding a new customer, updating customer details, and a function to get customer balance.

**Ans :**

CREATE OR REPLACE PACKAGE CustomerManagement AS

PROCEDURE AddCustomer (

p\_name VARCHAR2,

p\_dob DATE,

p\_initial\_balance NUMBER

);

PROCEDURE UpdateCustomer (

p\_customer\_id NUMBER,

p\_name VARCHAR2,

p\_dob DATE

);

FUNCTION GetCustomerBalance (

p\_customer\_id NUMBER

) RETURN NUMBER;

END CustomerManagement;

/

CREATE OR REPLACE PACKAGE BODY CustomerManagement AS

PROCEDURE AddCustomer (

p\_name VARCHAR2,

p\_dob DATE,

p\_initial\_balance NUMBER

)

AS

BEGIN

INSERT INTO Customers (Name, DOB, Balance)

VALUES (p\_name, p\_dob, p\_initial\_balance);

END;

PROCEDURE UpdateCustomer (

p\_customer\_id NUMBER,

p\_name VARCHAR2,

p\_dob DATE

)

AS

BEGIN

UPDATE Customers

SET Name = p\_name, DOB = p\_dob

WHERE CustomerID = p\_customer\_id;

END;

FUNCTION GetCustomerBalance (

p\_customer\_id NUMBER

)

RETURN NUMBER

IS

v\_balance NUMBER;

BEGIN

SELECT Balance INTO v\_balance FROM Accounts WHERE CustomerID = p\_customer\_id;

RETURN v\_balance;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN 0; -- Or handle the exception appropriately

END;

END CustomerManagement;

**Scenario 2 :** Create a package to manage employee data. Write a package **EmployeeManagement** with procedures to hire new employees, update employee details, and a function to calculate annual salary.

**Ans :**

CREATE OR REPLACE PACKAGE EmployeeManagement AS

PROCEDURE HireEmployee (

p\_name VARCHAR2,

p\_position VARCHAR2,

p\_salary NUMBER,

p\_department VARCHAR2,

p\_hire\_date DATE

);

PROCEDURE UpdateEmployee (

p\_employee\_id NUMBER,

p\_name VARCHAR2,

p\_position VARCHAR2,

p\_salary NUMBER,

p\_department VARCHAR2

);

FUNCTION CalculateAnnualSalary (

p\_employee\_id NUMBER

) RETURN NUMBER;

END EmployeeManagement;

/

CREATE OR REPLACE PACKAGE BODY EmployeeManagement AS

PROCEDURE HireEmployee (

p\_name VARCHAR2,

p\_position VARCHAR2,

p\_salary NUMBER,

p\_department VARCHAR2,

p\_hire\_date DATE

)

AS

BEGIN

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

VALUES (p\_name, p\_position, p\_salary, p\_department, p\_hire\_date);

END;

PROCEDURE UpdateEmployee (

p\_employee\_id NUMBER,

p\_name VARCHAR2,

p\_position VARCHAR2,

p\_salary NUMBER,

p\_department VARCHAR2

)

AS

BEGIN

UPDATE Employees

SET Name = p\_name, Position = p\_position, Salary = p\_salary, Department = p\_department

WHERE EmployeeID = p\_employee\_id;

END;

FUNCTION CalculateAnnualSalary (

p\_employee\_id NUMBER

) RETURN NUMBER

IS

v\_salary NUMBER;

BEGIN

SELECT Salary INTO v\_salary FROM Employees WHERE EmployeeID = p\_employee\_id;

RETURN v\_salary \* 12;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN 0;

END;

END EmployeeManagement;

**Scenario 3 :** Group all account-related operations into a package. 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.

**Ans :**

CREATE OR REPLACE PACKAGE AccountOperations AS

PROCEDURE OpenAccount (

p\_customer\_id NUMBER,

p\_account\_type VARCHAR2

);

PROCEDURE CloseAccount (

p\_account\_id NUMBER

);

FUNCTION GetTotalCustomerBalance (

p\_customer\_id NUMBER

) RETURN NUMBER;

END AccountOperations;

/

CREATE OR REPLACE PACKAGE BODY AccountOperations AS

PROCEDURE OpenAccount (

p\_customer\_id NUMBER,

p\_account\_type VARCHAR2

)

AS

BEGIN

INSERT INTO Accounts (CustomerID, AccountType, Balance)

VALUES (p\_customer\_id, p\_account\_type, 0);

END;

PROCEDURE CloseAccount (

p\_account\_id NUMBER

)

AS

BEGIN

DELETE FROM Accounts WHERE AccountID = p\_account\_id;

END;

FUNCTION GetTotalCustomerBalance (

p\_customer\_id NUMBER

) RETURN NUMBER

IS

v\_total\_balance NUMBER := 0;

BEGIN

SELECT SUM(Balance) INTO v\_total\_balance

FROM Accounts

WHERE CustomerID = p\_customer\_id;

RETURN v\_total\_balance;

END;

END AccountOperations**;**