**Question - 1**

Scenario 1: Group all customer-related procedures and functions into a package.

o Question: Create a package CustomerManagement with procedures for adding a new customer, updating customer details, and a function to get customer balance.

* PL/SQL Code---

CREATE OR REPLACE PACKAGE CustomerManagement AS

-- Procedure to add a new customer

PROCEDURE AddCustomer(p\_name VARCHAR2, p\_dob DATE, p\_balance NUMBER, p\_isvip VARCHAR2);

-- Procedure to update customer details

PROCEDURE UpdateCustomer(p\_customerid NUMBER, p\_name VARCHAR2, p\_dob DATE, p\_balance NUMBER, p\_isvip VARCHAR2);

-- Function to get customer balance

FUNCTION GetCustomerBalance(p\_customerid NUMBER) RETURN NUMBER;

END CustomerManagement;

/

--CREATE the PACKAGE --Create a new Package Body

CREATE OR REPLACE PACKAGE BODY CustomerManagement AS

PROCEDURE AddCustomer(p\_name VARCHAR2, p\_dob DATE, p\_balance NUMBER, p\_isvip VARCHAR2) IS

BEGIN

INSERT INTO Customers (NAME, DOB, BALANCE, LASTMODIFIED, ISVIP)

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

END AddCustomer;

PROCEDURE UpdateCustomer(p\_customerid NUMBER, p\_name VARCHAR2, p\_dob DATE, p\_balance NUMBER, p\_isvip VARCHAR2) IS

BEGIN

UPDATE Customers

SET NAME = p\_name,

DOB = p\_dob,

BALANCE = p\_balance,

LASTMODIFIED = SYSDATE,

ISVIP = p\_isvip

WHERE CUSTOMERID = p\_customerid;

END UpdateCustomer;

FUNCTION GetCustomerBalance(p\_customerid NUMBER) RETURN NUMBER IS

v\_balance NUMBER;

BEGIN

SELECT BALANCE INTO v\_balance

FROM Customers

WHERE CUSTOMERID = p\_customerid;

RETURN v\_balance;

END GetCustomerBalance;

END CustomerManagement;

/

-- Test adding a new customer

BEGIN

CustomerManagement.AddCustomer('Bruce Banner', TO\_DATE('22-DEC-69', 'DD-MON-YY'), 1200, 'NO');

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

EXCEPTION

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('Error: ' || SQLERRM);

END;

/

-- Test updating a customer

SET SERVEROUT ON;

BEGIN

CustomerManagement.UpdateCustomer(1, 'John Doe', TO\_DATE('15-MAY-85', 'DD-MON-YY'), 1000, 'NO');

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

EXCEPTION

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('Error: ' || SQLERRM);

END;

/

-- Test getting customer balance

DECLARE

v\_balance NUMBER;

BEGIN

v\_balance := CustomerManagement.GetCustomerBalance(1);

DBMS\_OUTPUT.PUT\_LINE('Customer balance: ' || v\_balance);

EXCEPTION

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('Error: ' || SQLERRM);

END;

/