**Question - 2**

Scenario 2: Create a package to manage employee data.

o Question: Write a package EmployeeManagement with procedures to hire new employees, update employee details, and a function to calculate annual salary.

PL/SQL Code ----

--Create the Package Specification

CREATE OR REPLACE PACKAGE EmployeeManagement AS

-- Procedure to hire a new employee

PROCEDURE HireEmployee(p\_name VARCHAR2, p\_position VARCHAR2, p\_salary NUMBER, p\_department VARCHAR2, p\_hiredate DATE);

-- Procedure to update employee details

PROCEDURE UpdateEmployee(p\_employeeid NUMBER, p\_name VARCHAR2, p\_position VARCHAR2, p\_salary NUMBER, p\_department VARCHAR2);

-- Function to calculate annual salary

FUNCTION CalculateAnnualSalary(p\_employeeid NUMBER) RETURN NUMBER;

END EmployeeManagement;

/

--Create the Package Body

CREATE OR REPLACE PACKAGE BODY EmployeeManagement AS

PROCEDURE HireEmployee(p\_name VARCHAR2, p\_position VARCHAR2, p\_salary NUMBER, p\_department VARCHAR2, p\_hiredate DATE) IS

BEGIN

INSERT INTO Employees (NAME, POSITION, SALARY, DEPARTMENT, HIREDATE)

VALUES (p\_name, p\_position, p\_salary, p\_department, p\_hiredate);

END HireEmployee;

PROCEDURE UpdateEmployee(p\_employeeid NUMBER, p\_name VARCHAR2, p\_position VARCHAR2, p\_salary NUMBER, p\_department VARCHAR2) IS

BEGIN

UPDATE Employees

SET NAME = p\_name,

POSITION = p\_position,

SALARY = p\_salary,

DEPARTMENT = p\_department

WHERE EMPLOYEEID = p\_employeeid;

END UpdateEmployee;

FUNCTION CalculateAnnualSalary(p\_employeeid NUMBER) RETURN NUMBER IS

v\_annual\_salary NUMBER;

BEGIN

SELECT SALARY \* 12 INTO v\_annual\_salary

FROM Employees

WHERE EMPLOYEEID = p\_employeeid;

RETURN v\_annual\_salary;

END CalculateAnnualSalary;

END EmployeeManagement;

/

-- Test the Package

SET SERVEROUT ON;

BEGIN

EmployeeManagement.HireEmployee('Bruce Banner', 'Scientist', 85000, 'Research', TO\_DATE('22-DEC-20', 'DD-MON-YY'));

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

EXCEPTION

WHEN OTHERS THEN

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

END;

/

BEGIN

EmployeeManagement.UpdateEmployee(1, 'Alice Johnson', 'Senior Manager', 90000, 'HR');

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

EXCEPTION

WHEN OTHERS THEN

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

END;

/

DECLARE

v\_annual\_salary NUMBER;

BEGIN

v\_annual\_salary := EmployeeManagement.CalculateAnnualSalary(1);

DBMS\_OUTPUT.PUT\_LINE('Annual salary: ' || v\_annual\_salary);

EXCEPTION

WHEN OTHERS THEN

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

END;

/

-- Check the OUTPUT

-- Verify the employee hired

SELECT \* FROM Employees WHERE NAME = 'Bruce Banner';

-- Verify the employee updated

SELECT \* FROM Employees WHERE EMPLOYEEID = 1;

-- Verify the annual salary

SELECT SALARY \* 12 AS ANNUAL\_SALARY FROM Employees WHERE EMPLOYEEID = 1;