**Imama Kainat**

**22F3661**

**SE-4A**

Question 1

CREATE OR REPLACE PACKAGE employee\_count\_pkg AS

FUNCTION get\_employee\_count(p\_job\_title VARCHAR2) RETURN NUMBER;

END employee\_count\_pkg;

CREATE OR REPLACE PACKAGE BODY employee\_count\_pkg AS

FUNCTION get\_employee\_count(p\_job\_title VARCHAR2) RETURN NUMBER IS

v\_employee\_count NUMBER := 0;

BEGIN

SELECT COUNT(\*) INTO v\_employee\_count

FROM emp

WHERE job = CLERK;

RETURN v\_employee\_count;

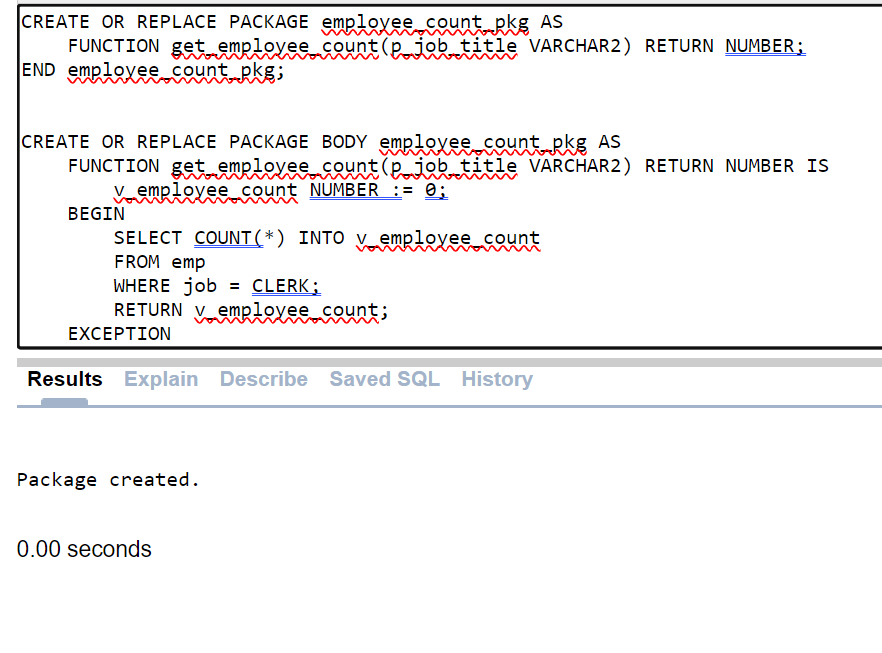
EXCEPTION

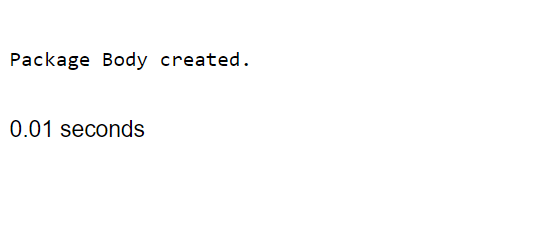
WHEN OTHERS THEN

RETURN NULL;

END get\_employee\_count;

END employee\_count\_pkg;





**Question 2**

CREATE OR REPLACE PACKAGE EmpPackage AS

TYPE EmployeeRecord IS RECORD (

emp\_id NUMBER,

emp\_name VARCHAR2(100),

emp\_salary NUMBER

);

TYPE EmployeeTable IS TABLE OF EmployeeRecord;

FUNCTION GetTopEmployees(N IN NUMBER)

RETURN EmployeeTable;

END EmpPackage;

CREATE OR REPLACE PACKAGE BODY EmpPackage AS

FUNCTION GetTopEmployees(N IN NUMBER) RETURN EmployeeTable IS

top\_employees EmployeeTable;

BEGIN

SELECT empno, ename,sal

BULK COLLECT INTO top\_employees

FROM (SELECT empno, ename, sal

FROM emp

ORDER BY sal DESC)

WHERE ROWNUM <= N;

RETURN top\_employees;

END GetTopEmployees;

END EmpPackage;

DECLARE

top\_employees EmpPackage.EmployeeTable;

BEGIN

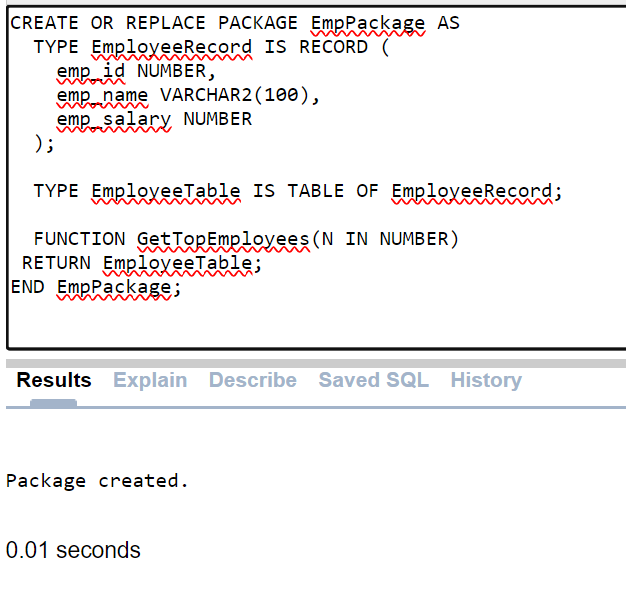
top\_employees := EmpPackage.GetTopEmployees(5);

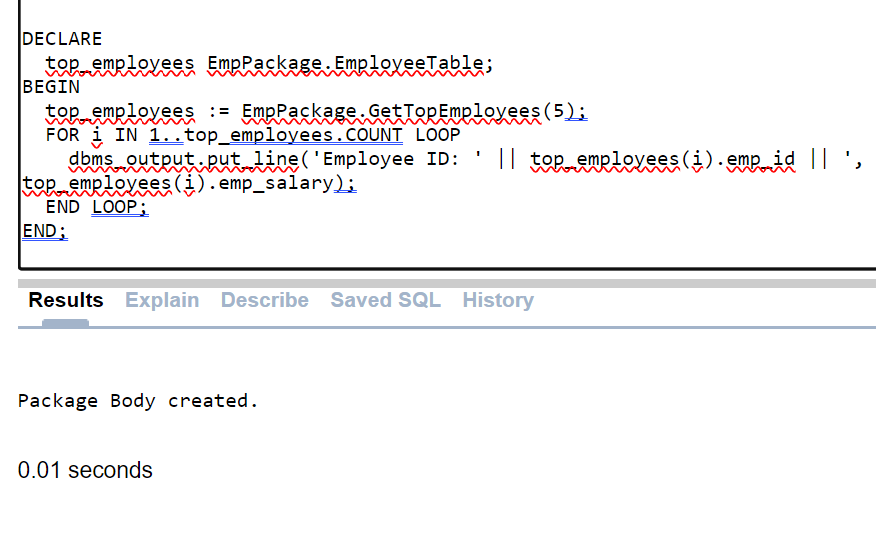
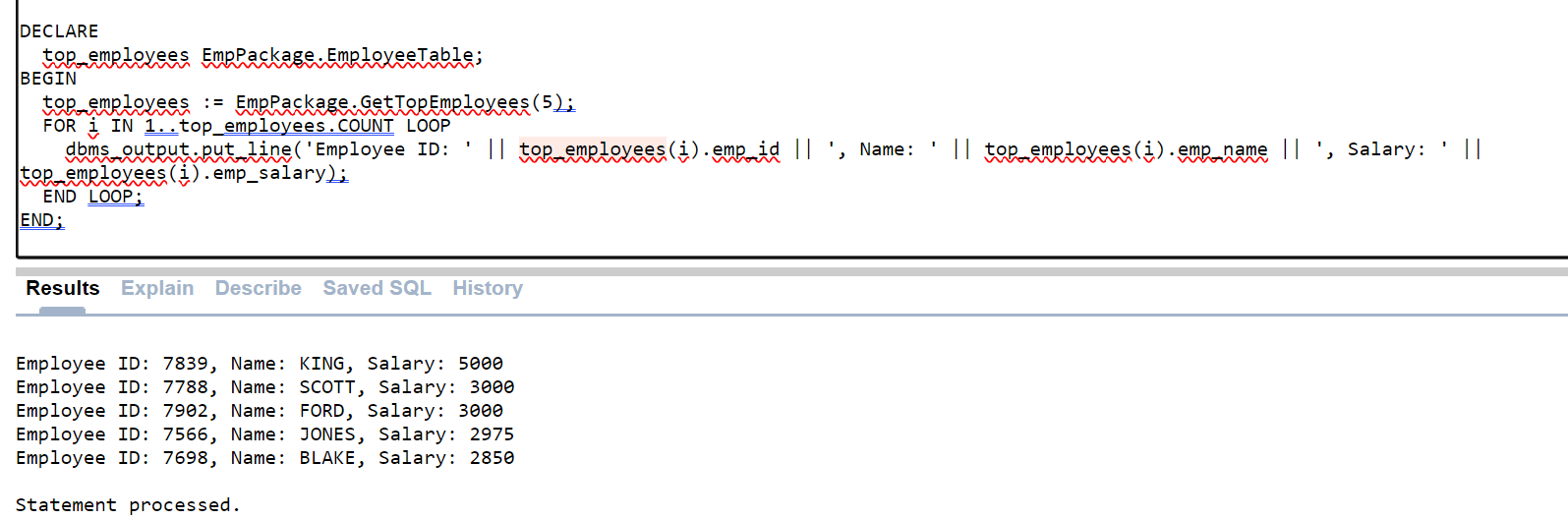
FOR i IN 1..top\_employees.COUNT LOOP

dbms\_output.put\_line('Employee ID: ' || top\_employees(i).emp\_id || ', Name: ' || top\_employees(i).emp\_name || ', Salary: ' || top\_employees(i).emp\_salary);

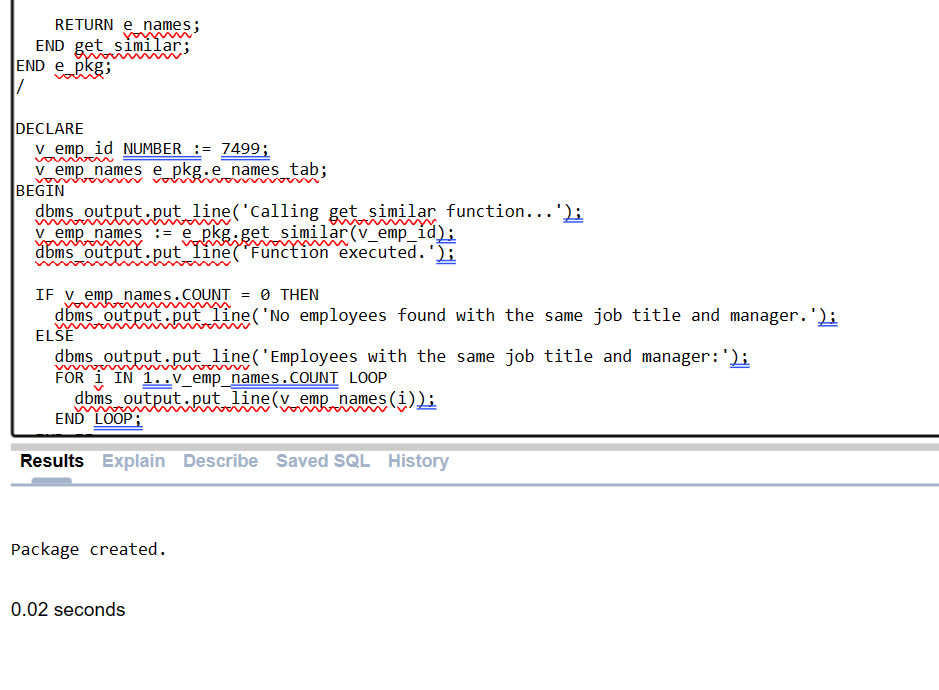
END LOOP;

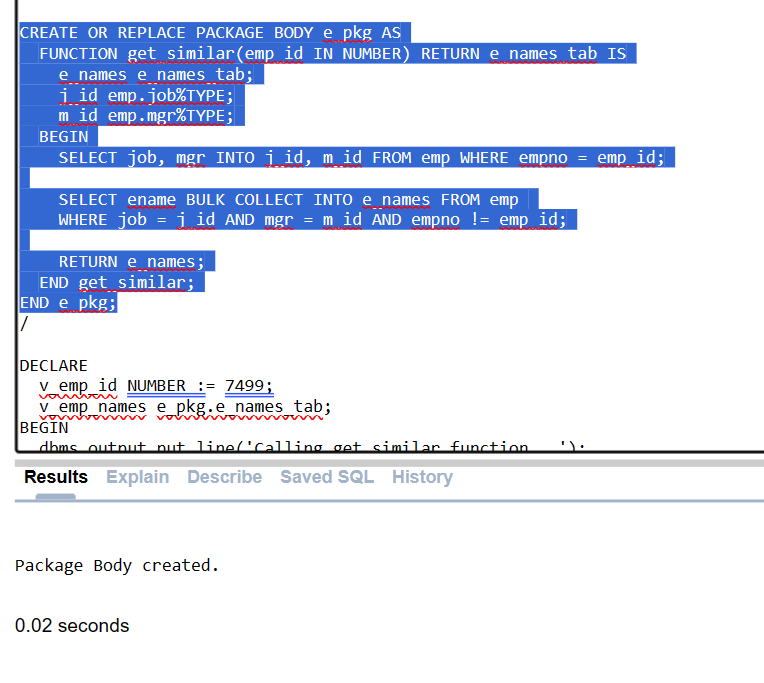
END;

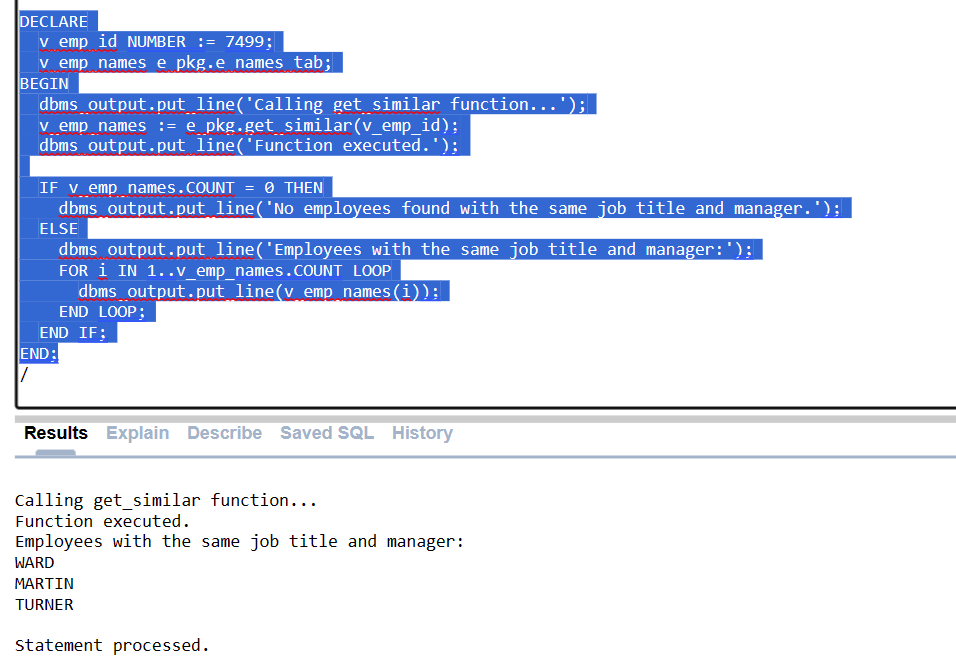


**Question 3**







CREATE OR REPLACE PACKAGE e\_pkg AS

TYPE e\_names\_tab IS TABLE OF VARCHAR2(100) INDEX BY PLS\_INTEGER;

FUNCTION get\_similar(emp\_id IN NUMBER) RETURN e\_names\_tab;

END e\_pkg;

/

CREATE OR REPLACE PACKAGE BODY e\_pkg AS

FUNCTION get\_similar(emp\_id IN NUMBER) RETURN e\_names\_tab IS

e\_names e\_names\_tab;

j\_id emp.job%TYPE;

m\_id emp.mgr%TYPE;

BEGIN

SELECT job, mgr INTO j\_id, m\_id FROM emp WHERE empno = emp\_id;

SELECT ename BULK COLLECT INTO e\_names FROM emp

WHERE job = j\_id AND mgr = m\_id AND empno != emp\_id;

RETURN e\_names;

END get\_similar;

END e\_pkg;

/

DECLARE

v\_emp\_id NUMBER := 7499;

v\_emp\_names e\_pkg.e\_names\_tab;

BEGIN

dbms\_output.put\_line('Calling get\_similar function...');

v\_emp\_names := e\_pkg.get\_similar(v\_emp\_id);

dbms\_output.put\_line('Function executed.');

IF v\_emp\_names.COUNT = 0 THEN

dbms\_output.put\_line('No employees found with the same job title and manager.');

ELSE

dbms\_output.put\_line('Employees with the same job title and manager:');

FOR i IN 1..v\_emp\_names.COUNT LOOP

dbms\_output.put\_line(v\_emp\_names(i));

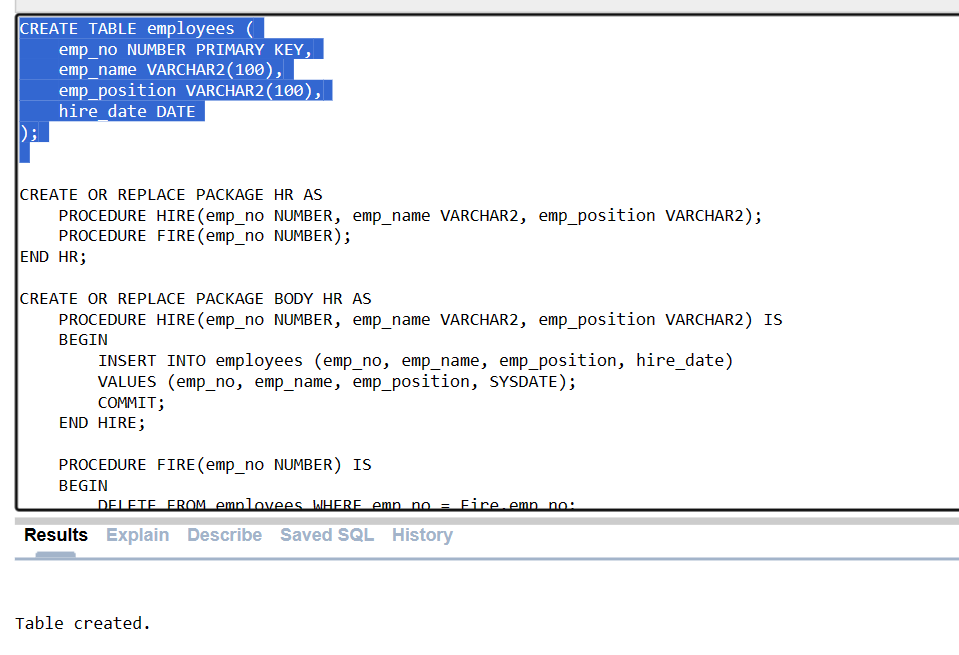
END LOOP;

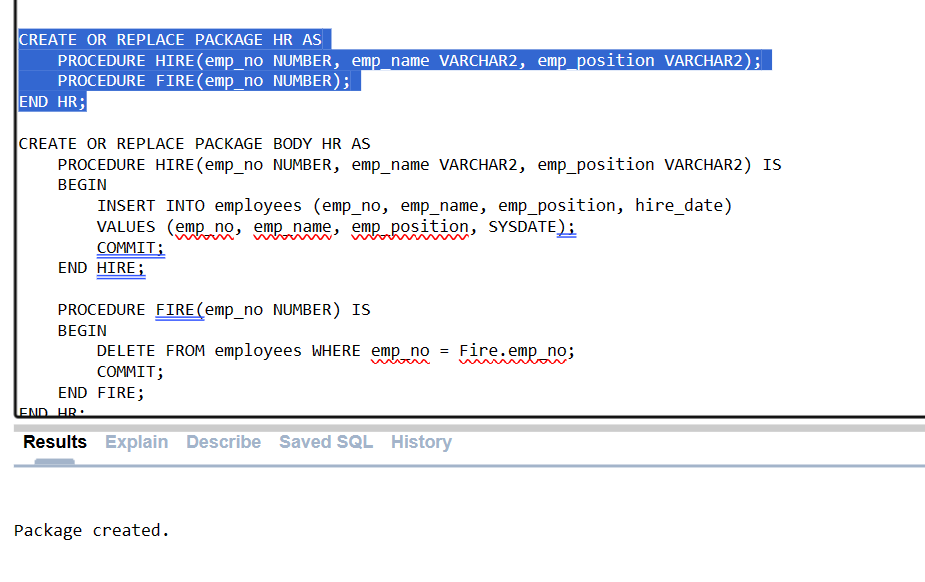
END IF;

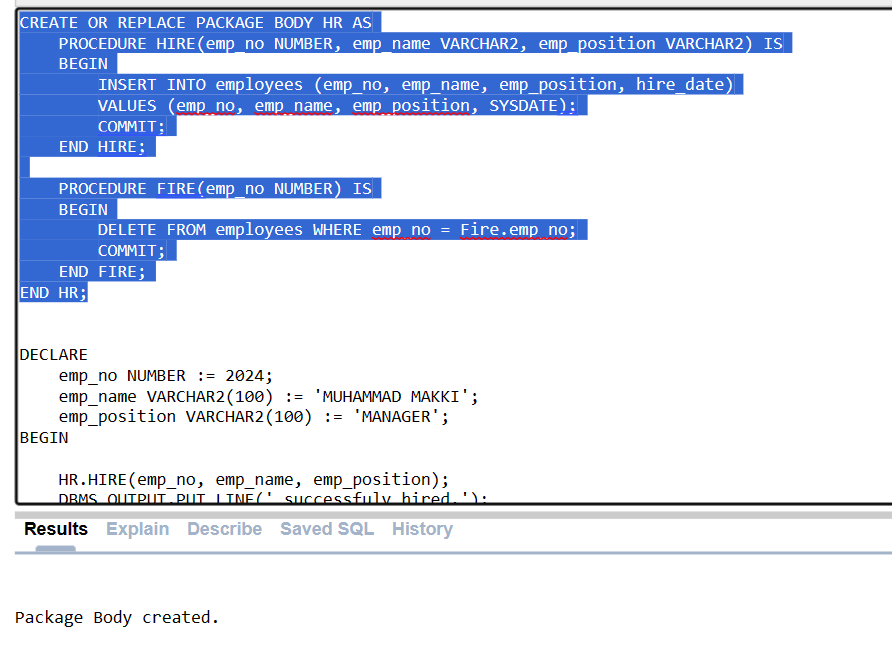
END;

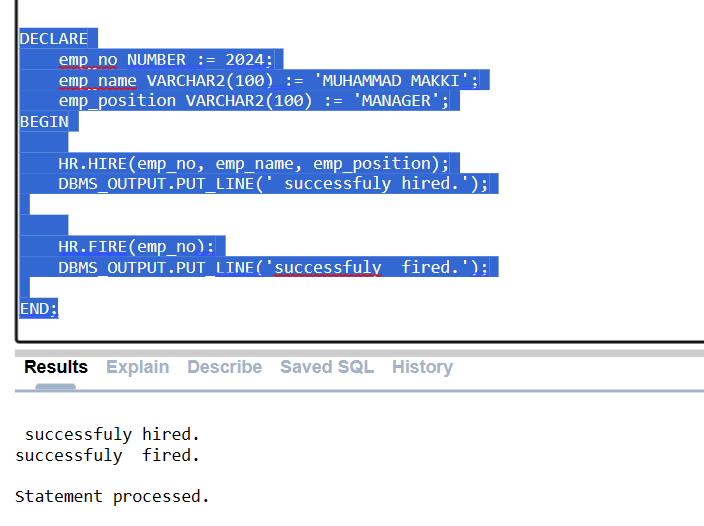
/

**Question 4**









CREATE TABLE employees (

emp\_no NUMBER PRIMARY KEY,

emp\_name VARCHAR2(100),

emp\_position VARCHAR2(100),

hire\_date DATE

);

CREATE OR REPLACE PACKAGE HR AS

PROCEDURE HIRE(emp\_no NUMBER, emp\_name VARCHAR2, emp\_position VARCHAR2);

PROCEDURE FIRE(emp\_no NUMBER);

END HR;

CREATE OR REPLACE PACKAGE BODY HR AS

PROCEDURE HIRE(emp\_no NUMBER, emp\_name VARCHAR2, emp\_position VARCHAR2) IS

BEGIN

INSERT INTO employees (emp\_no, emp\_name, emp\_position, hire\_date)

VALUES (emp\_no, emp\_name, emp\_position, SYSDATE);

s COMMIT;

END HIRE;

PROCEDURE FIRE(emp\_no NUMBER) IS

BEGIN

DELETE FROM employees WHERE emp\_no = Fire.emp\_no;

COMMIT;

END FIRE;

END HR;

DECLARE

emp\_no NUMBER := 2024;

emp\_name VARCHAR2(100) := 'MUHAMMAD MAKKI';

emp\_position VARCHAR2(100) := 'MANAGER';

BEGIN

HR.HIRE(emp\_no, emp\_name, emp\_position);

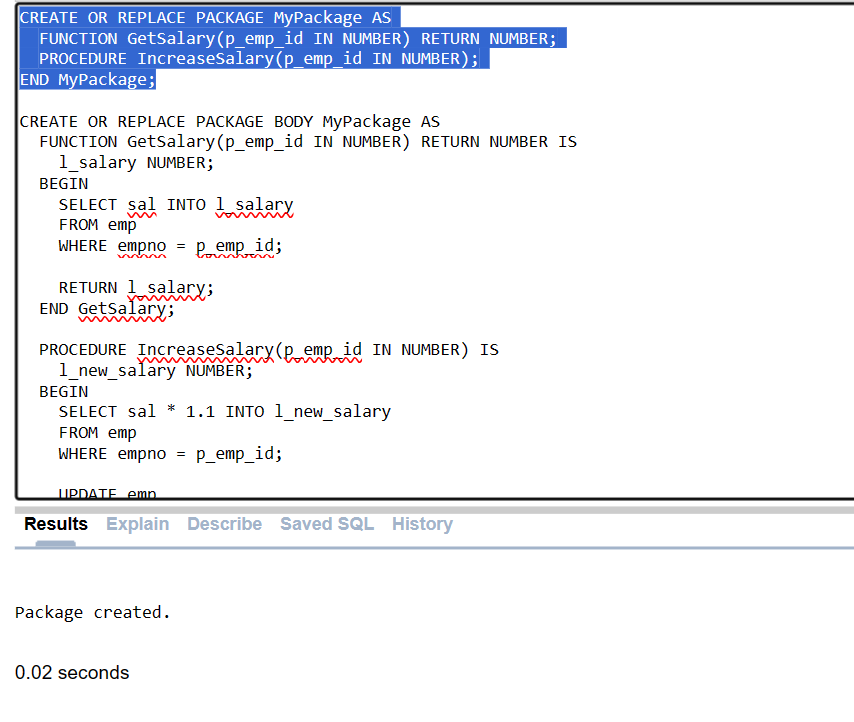
DBMS\_OUTPUT.PUT\_LINE(' successfuly hired.');

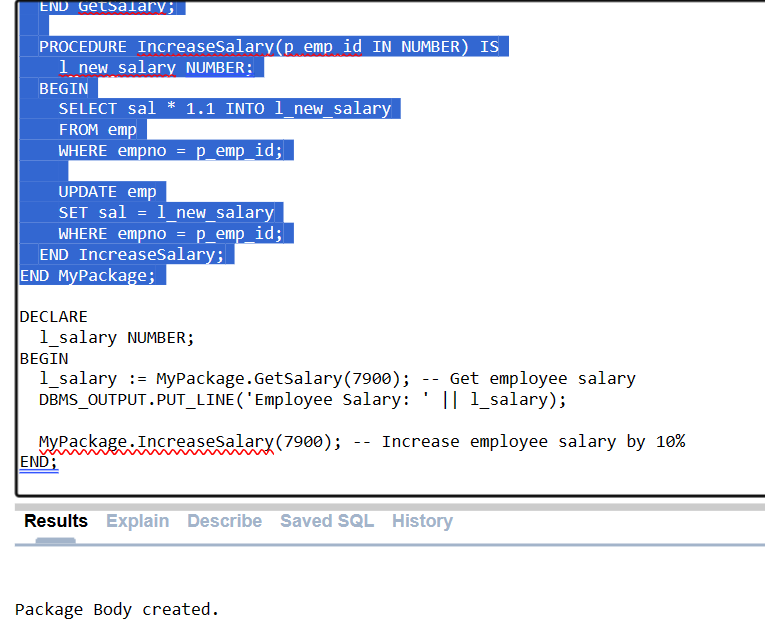
HR.FIRE(emp\_no);

DBMS\_OUTPUT.PUT\_LINE('successfuly fired.');

END;

**Question 5**







CREATE OR REPLACE PACKAGE MyPackage AS

FUNCTION GetSalary(p\_emp\_id IN NUMBER) RETURN NUMBER;

PROCEDURE IncreaseSalary(p\_emp\_id IN NUMBER);

END MyPackage;

CREATE OR REPLACE PACKAGE BODY MyPackage AS

FUNCTION GetSalary(p\_emp\_id IN NUMBER) RETURN NUMBER IS

l\_salary NUMBER;

BEGIN

SELECT sal INTO l\_salary

FROM emp

WHERE empno = p\_emp\_id;

RETURN l\_salary;

END GetSalary;

PROCEDURE IncreaseSalary(p\_emp\_id IN NUMBER) IS

l\_new\_salary NUMBER;

BEGIN

SELECT sal \* 1.1 INTO l\_new\_salary

FROM emp

WHERE empno = p\_emp\_id;

UPDATE emp

SET sal = l\_new\_salary

WHERE empno = p\_emp\_id;

END IncreaseSalary;

END MyPackage;

DECLARE

l\_salary NUMBER;

BEGIN

l\_salary := MyPackage.GetSalary(7900); -- Get employee salary

DBMS\_OUTPUT.PUT\_LINE('Employee Salary: ' || l\_salary);

MyPackage.IncreaseSalary(7900); -- Increase employee salary by 10%

END;

**MY notes**

Procedurrs

Functions

Camnadnd line variesble

;x

BIND VARIALBLE

SUB PROGRAM

PROCEDURE:ACTION

FUNCTION:COMPUTATION (LOCAL (ONLY ONE BLOCK )OR STORE(STIRED UB THE FORM OF OBJECT)

IS AND AS

PACKAGE

CREAT REPLACE

PARAMETER (IN(BAHIR SE) , OUT, IN OUT) (IN ,OUT)PASS BY REFERENCE

CAN MAKE MULTIPLE PACKAGES

IN BY DEFAULT TYPE

OUT OUT AND CAN MODIFY.CANT TAKE THE IN

IS FUNC +PROCE ( NESTED BLOCKS) LOCAL

AS STORED ,ANYONYMOUS BLOCK ,STANDALONE STORED

* FUNCITONS ALWASYS RETURN
* ASLWAYS BE TYPE OF IN
* EXECTUTABLE KO BEGIN KE UNDER HO GA

PACKAGES

PACKAGE KE THROUGH HI EXECUTE HO GA NOT LIKE FUNCITON

PACKAGE -> SIMILAR TYPE OF PROCEDURE ,FUNCITONS

WHOLE PACKAGE CANT BE EXECUTED

SPCIFICATION

BODY

PACKAGE.