**Section10**

**PL/SQL 10:**

CREATE OR REPLACE PACKAGE manage\_jobs\_pkg

IS

g\_todays\_date DATE := SYSDATE;

CURSOR jobs\_curs IS

SELECT employee\_id, job\_id FROM employees

ORDER BY employee\_id;

PROCEDURE update\_job

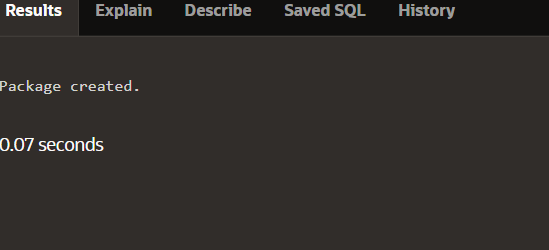
(p\_emp\_id IN employees.employee\_id%TYPE);

PROCEDURE fetch\_emps

(p\_job\_id IN employees.job\_id%TYPE,

p\_emp\_id OUT employees.employee\_id%TYPE);

END manage\_jobs\_pkg;



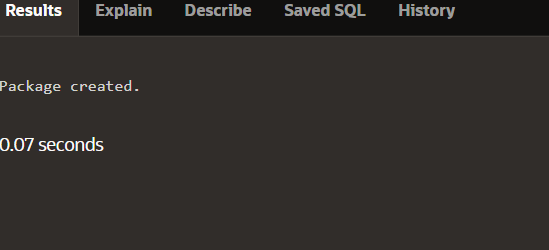
CREATE [OR REPLACE] PACKAGE BODY package\_name IS|AS

private type and variable declarations

subprogram bodies

[BEGIN initialization statements]

END [package\_name];



CREATE OR REPLACE PACKAGE BODY check\_emp\_pkg IS

PROCEDURE chk\_hiredate

(p\_date IN employees.hire\_date%TYPE)

IS BEGIN

IF MONTHS\_BETWEEN(SYSDATE, p\_date) >

g\_max\_length\_of\_service \* 12 THEN

RAISE\_APPLICATION\_ERROR(-20200, 'Invalid Hiredate');

END IF;

END chk\_hiredate;

PROCEDURE chk\_dept\_mgr

(p\_empid IN employees.employee\_id%TYPE,

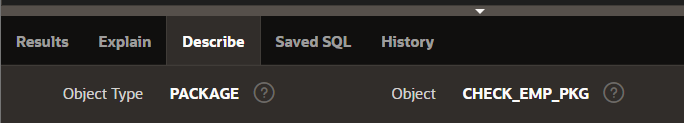
p\_mgr IN employees.manager\_id%TYPE)

IS BEGIN ...

END chk\_dept\_mgr;

END check\_emp\_pkg;

DESCRIBE check\_emp\_pkg



CREATE OR REPLACE PACKAGE salary\_pkg

IS

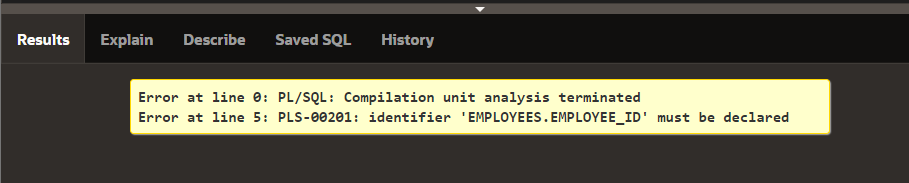
g\_max\_sal\_raise CONSTANT NUMBER := 0.20;

PROCEDURE update\_sal

(p\_employee\_id employees.employee\_id%TYPE,

p\_new\_salary employees.salary%TYPE);

END salary\_pkg;



PROCEDURE update\_sal -- public procedure

(p\_employee\_id employees.employee\_id%TYPE,

p\_new\_salary employees.salary%TYPE)

IS v\_old\_salary employees.salary%TYPE; -- local variable

BEGIN

SELECT salary INTO v\_old\_salary FROM employees

WHERE employee\_id = p\_employee\_id;

IF validate\_raise(v\_old\_salary, p\_new\_salary) THEN

UPDATE employees

SET salary = p\_new\_salary

WHERE employee\_id = p\_employee\_id;

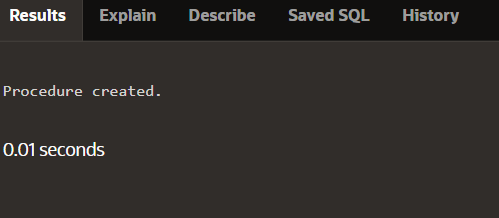
ELSE

RAISE\_APPLICATION\_ERROR(-20210, 'Raise too high');

END IF;

END update\_sal;

END salary\_pkg;



DECLARE

v\_bool BOOLEAN;

v\_number NUMBER;

BEGIN

salary\_pkg.update\_sal(100,25000); -- 1

update\_sal(100,25000); -- 2

v\_bool := salary\_pkg.validate\_raise(24000,25000); -- 3

v\_number := salary\_pkg.g\_max\_sal\_raise; -- 4

v\_number := salary\_pkg.v\_old\_salary; -- 5

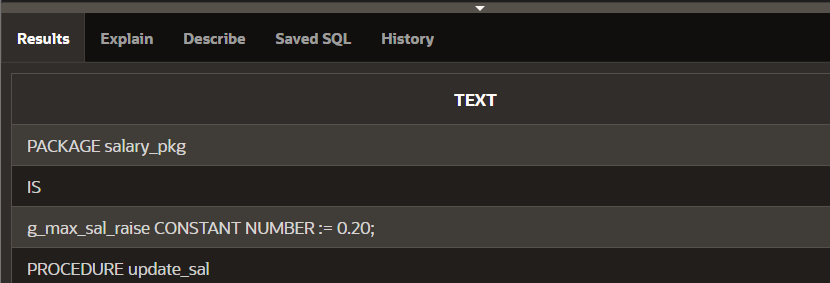
END;

SELECT text

FROM user\_source

WHERE name = 'SALARY\_PKG' AND type = 'PACKAGE'

ORDER BY line;

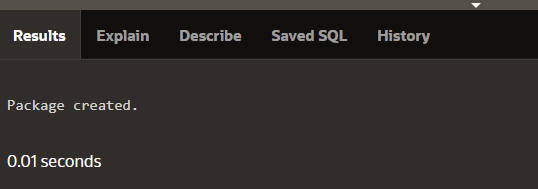


CREATE PACKAGE sample\_pack IS

PROCEDURE sample\_proc (p\_char\_param IN CHAR);

PROCEDURE sample\_proc (p\_varchar\_param IN VARCHAR2);

END sample\_pack;



CREATE OR REPLACE PACKAGE dept\_pkg IS

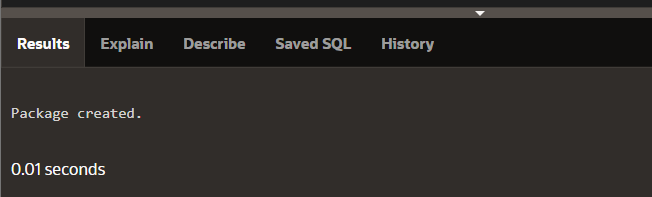
PROCEDURE add\_department(p\_deptno NUMBER,

p\_name VARCHAR2 := 'unknown', p\_loc NUMBER := 1700);

PROCEDURE add\_department(

p\_name VARCHAR2 := 'unknown', p\_loc NUMBER := 1700);

END dept\_pkg;



CREATE OR REPLACE PACKAGE BODY dept\_pkg IS

PROCEDURE add\_department (p\_deptno NUMBER,

p\_name VARCHAR2:='unknown', p\_loc NUMBER:=1700) IS

BEGIN

INSERT INTO departments(department\_id,

department\_name, location\_id)

VALUES (p\_deptno, p\_name, p\_loc);

END add\_department;

PROCEDURE add\_department (

p\_name VARCHAR2:='unknown', p\_loc NUMBER:=1700) IS

BEGIN

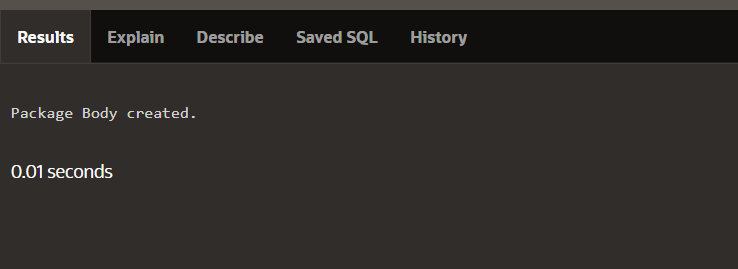
INSERT INTO departments (department\_id,

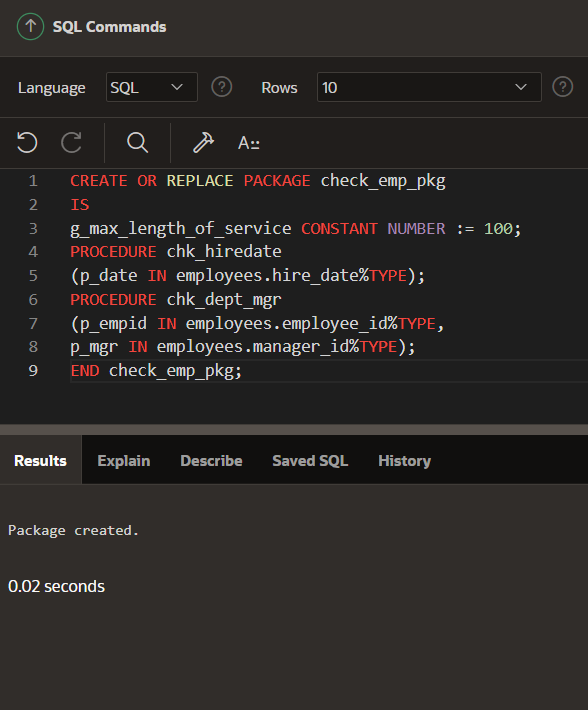
department\_name, location\_id)

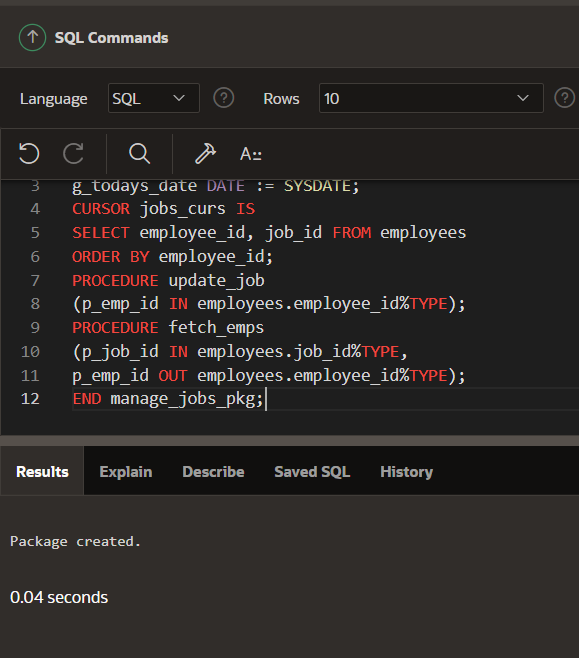
VALUES (departments\_seq.NEXTVAL, p\_name, p\_loc);

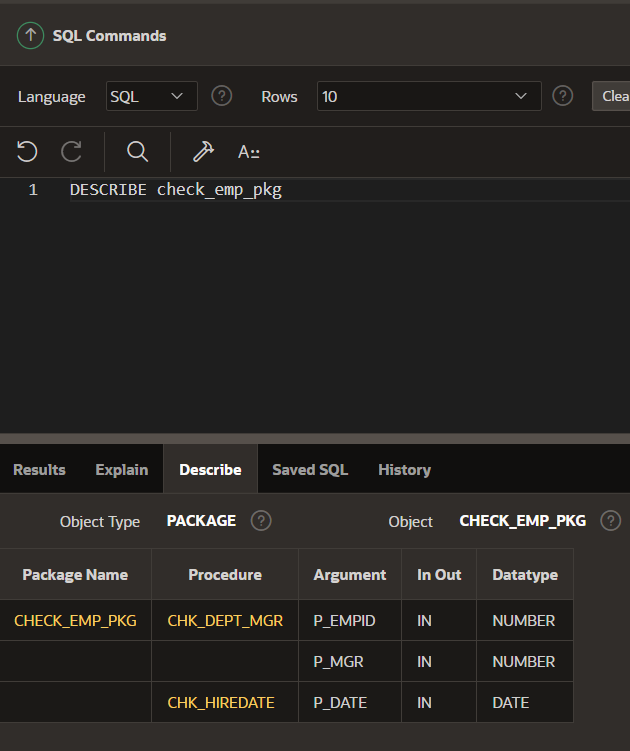
END add\_department;

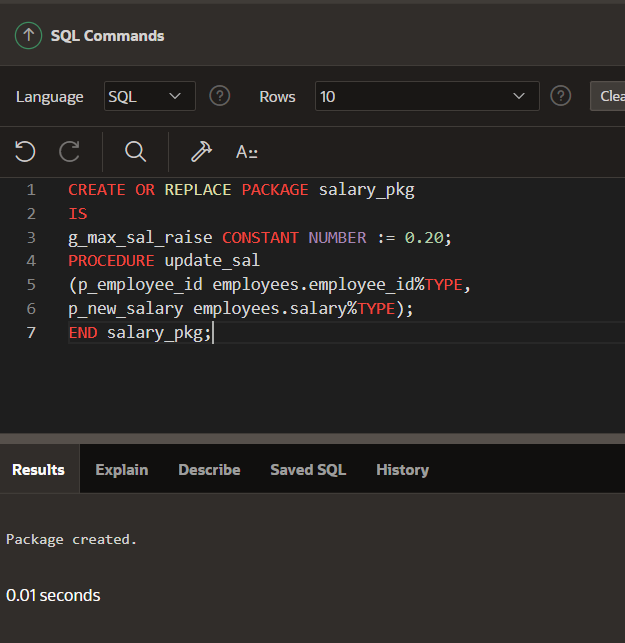
END dept\_pkg;

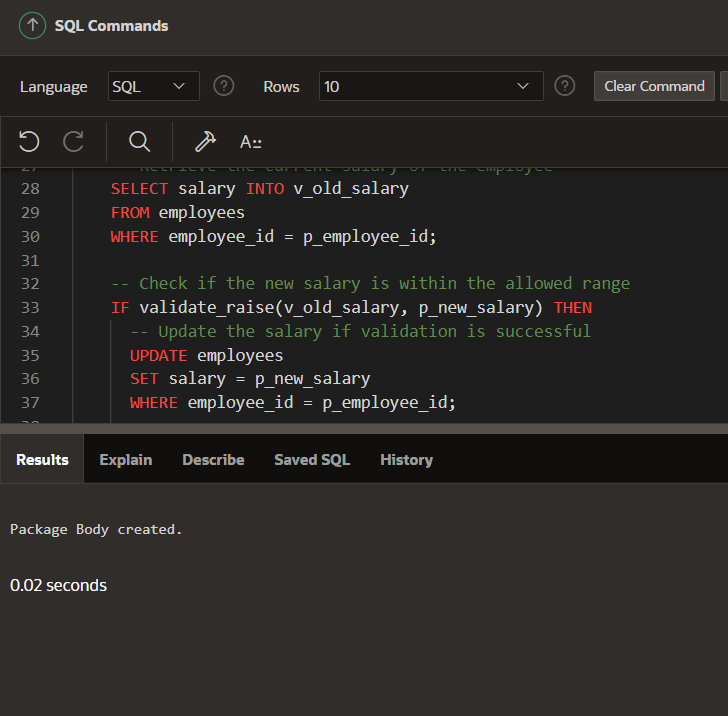


****

****

****

****

****

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer program

Description automatically generated**

**A screenshot of a computer program

Description automatically generated**

**A screenshot of a computer program

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer program

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer program

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer program

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

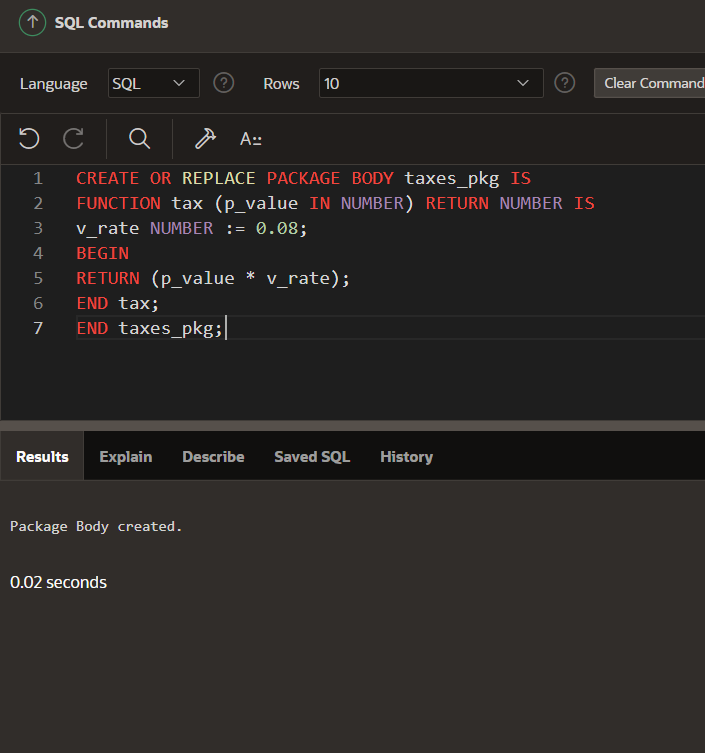
Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

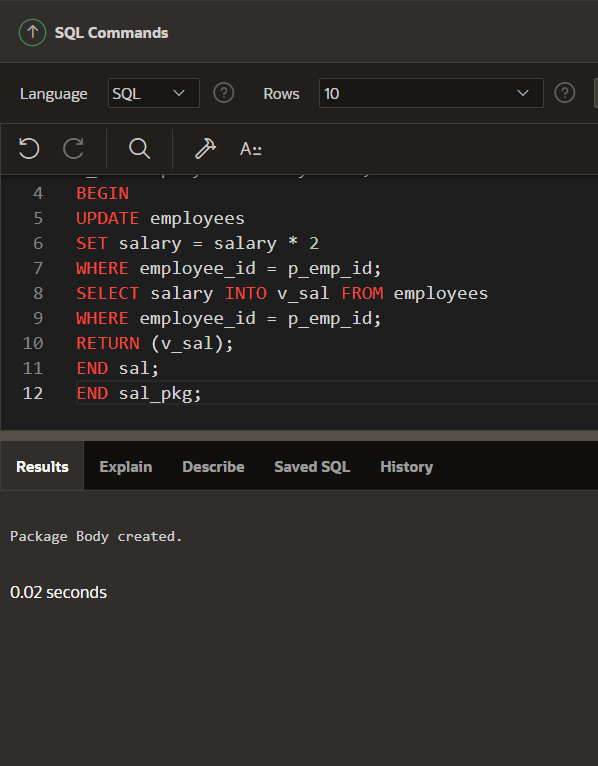
****

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

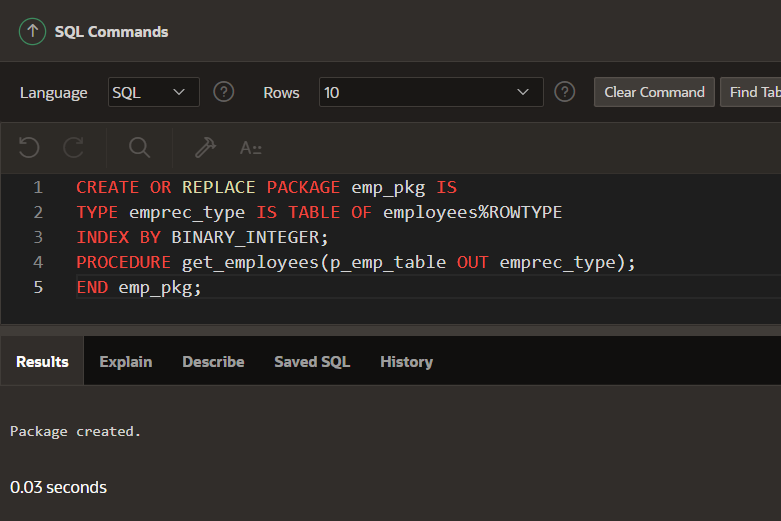
****

**A screenshot of a computer program

Description automatically generated**

**A screenshot of a computer program

Description automatically generated**

****

**A screenshot of a computer

Description automatically generated**