**Exercises 12-15**

**Database programming**

**Name: Sakshi Jain Student Id: c0753352**

**Instruction: Provide FULL Screenshot of your code with the result on SQL Developer**

**Exercise 12 (1.2%)**

Write a PL/SQL block that declares and uses two cursors—one without a parameter and one with a parameter. The first cursor retrieves the department number and the department name from the departments table for all departments whose ID number is less than 100. The second cursor receives the department number as a parameter, and retrieves employee details for those who work in that department and whose employee\_id is less than 120.

SET SERVEROUTPUT ON

DECLARE CURSOR c\_dept\_cursor IS

SELECT department\_id, department\_name

FROM departments

WHERE department\_id < 100

ORDER BY

department\_id;

CURSOR c\_emp\_cursor(v\_deptno NUMBER)IS

SELECT last\_name,job\_id,hire\_date,salary FROM

employees

WHERE department\_id =v\_deptno

AND employee\_id < 120;

v\_current\_deptno departments.department\_id%TYPE;

v\_current\_dname departments.department\_name%TYPE;

v\_ename employees.last\_name%TYPE;

v\_job employees.job\_id%TYPE;

v\_hiredate employees.hire\_date%TYPE;

v\_sal employees.salary%TYPE;

BEGIN

OPEN c\_dept\_cursor;

LOOP

FETCH c\_dept\_cursor INTO v\_current\_deptno,v\_current\_dname;

EXIT WHEN c\_dept\_cursor%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE('DEPARTMENT NUMBER:' || v\_current\_deptno || ' ,DEPARTMENT NAME:' || v\_current\_dname);

IF c\_emp\_cursor%ISOPEN THEN

CLOSE c\_emp\_cursor;

END IF;

OPEN c\_emp\_cursor(v\_current\_deptno);

LOOP

FETCH c\_emp\_cursor INTO v\_ename,v\_job,v\_hiredate,v\_sal;

EXIT WHEN c\_emp\_cursor%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE('EMPLOYEE NAME:' || v\_ename || ' ,JOB:' || v\_job || ' ,HIRE DATE:' ||v\_hiredate || ' ,SALARY: '||v\_sal);

END LOOP;

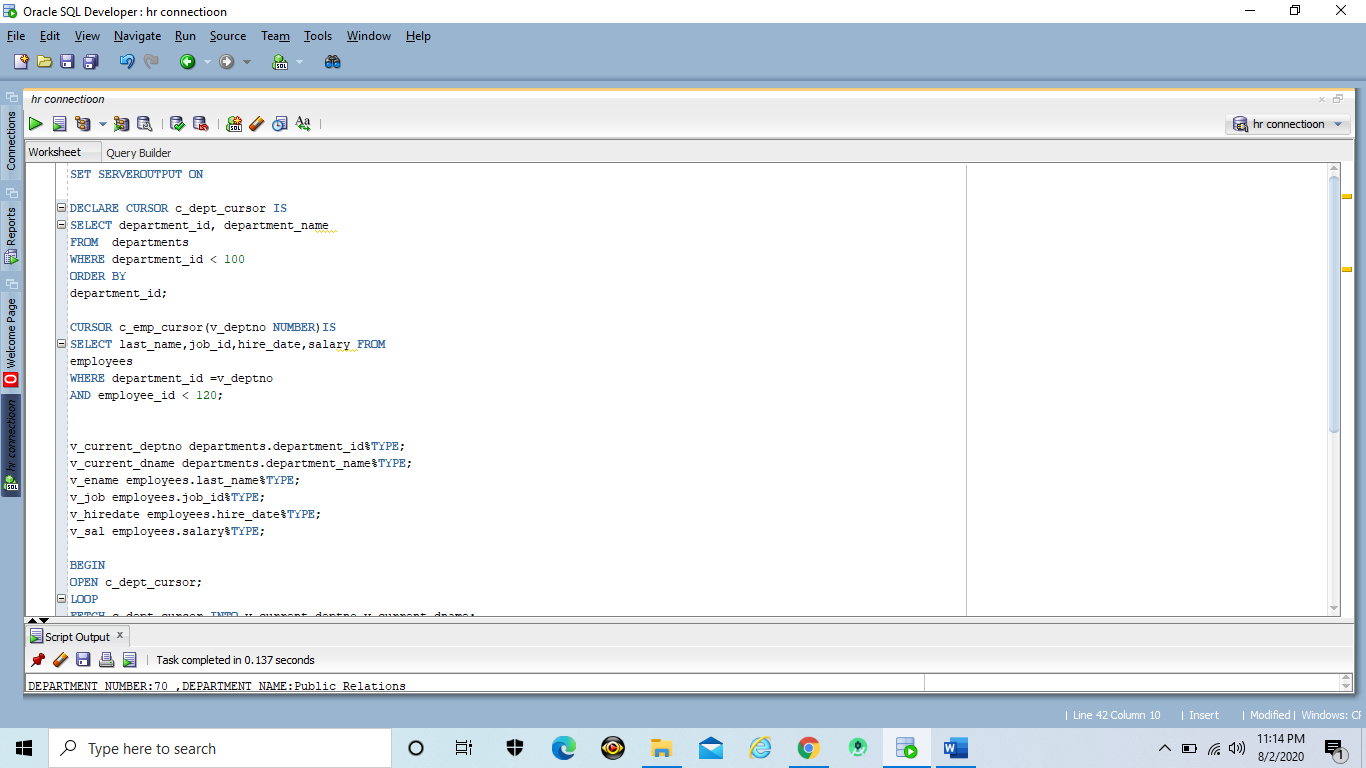
DBMS\_OUTPUT.PUT\_LINE('---------------------------------------------');

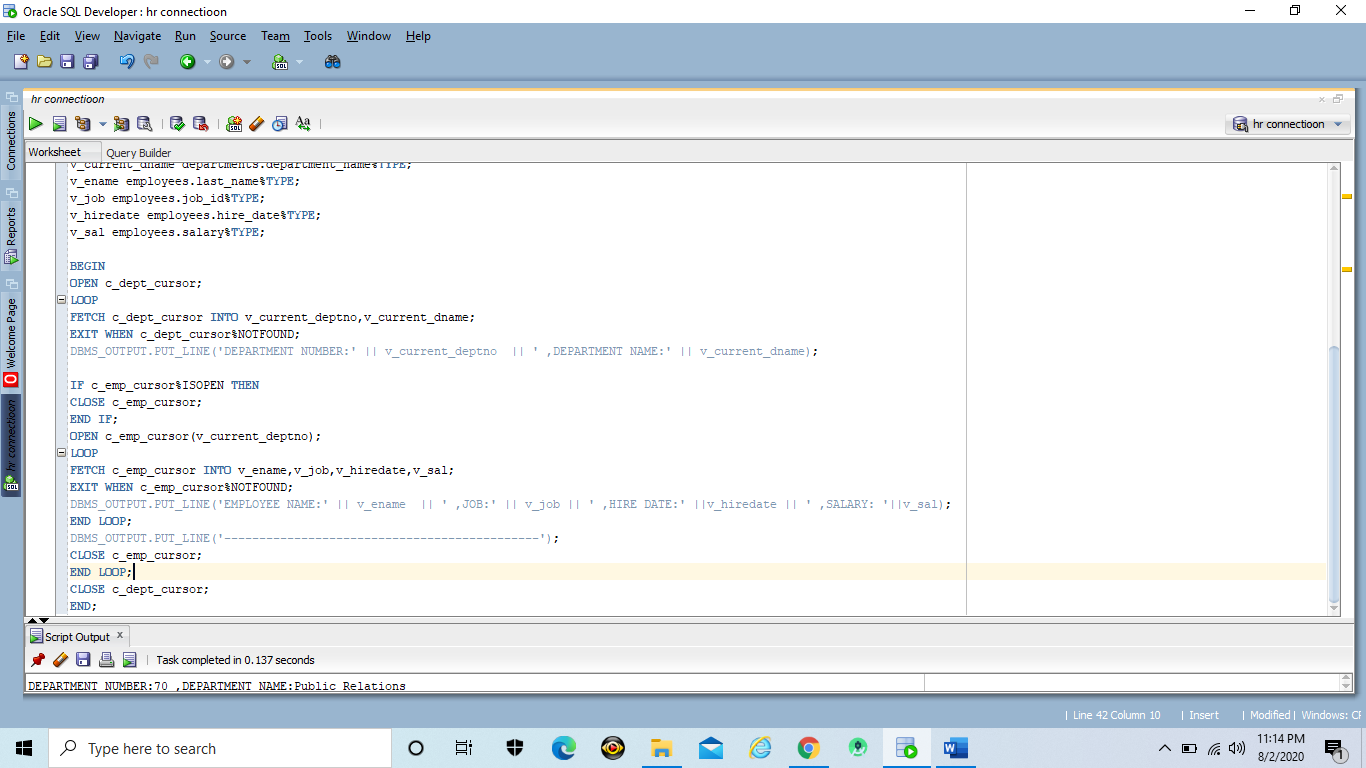
CLOSE c\_emp\_cursor;

END LOOP;

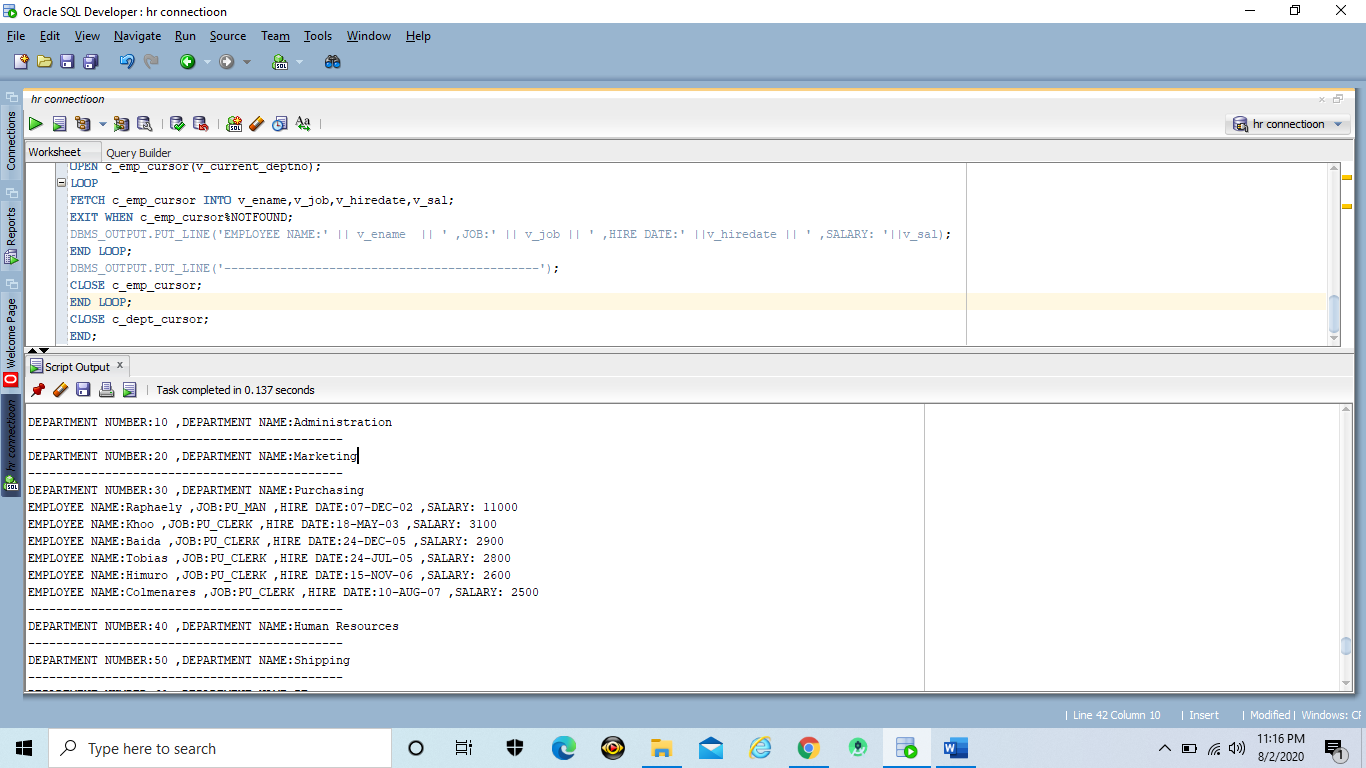
CLOSE c\_dept\_cursor;

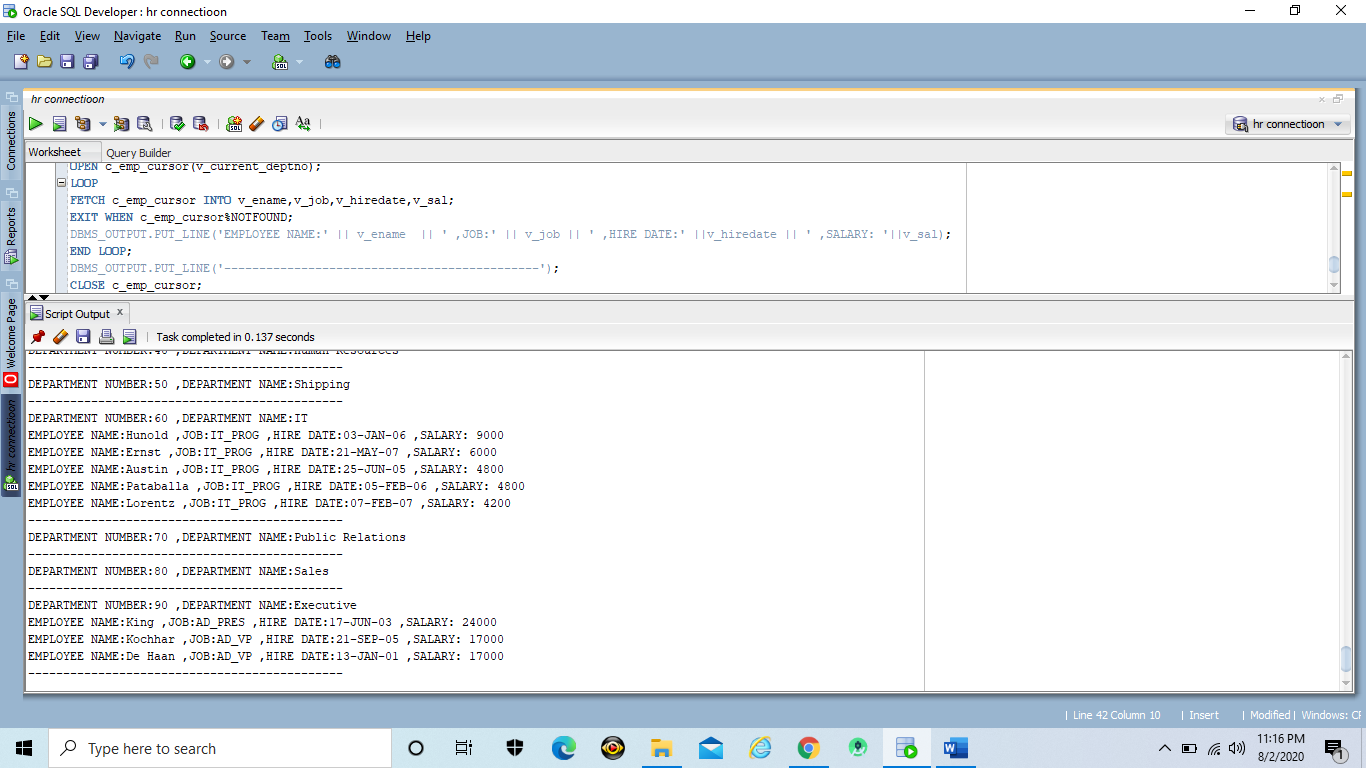
END;





Output





**Exercise 13 (1.2%)**

Declare a cursor dept\_cursor to retrieve department\_id and department\_name for those departments with department\_id less than 100. Order by department\_id. Open dept\_cursor and use a simple loop to fetch values into the variables declared. Display the department number and department name. Use the appropriate cursor attribute to exit the loop.

SET SERVEROUTPUT ON

DECLARE CURSOR c\_dept\_cursor IS

SELECT department\_id, department\_name

FROM departments

WHERE department\_id < 100

ORDER BY

department\_id;

v\_current\_deptno departments.department\_id%TYPE;

v\_current\_dname departments.department\_name%TYPE;

BEGIN

OPEN c\_dept\_cursor;

LOOP

FETCH c\_dept\_cursor INTO v\_current\_deptno,v\_current\_dname;

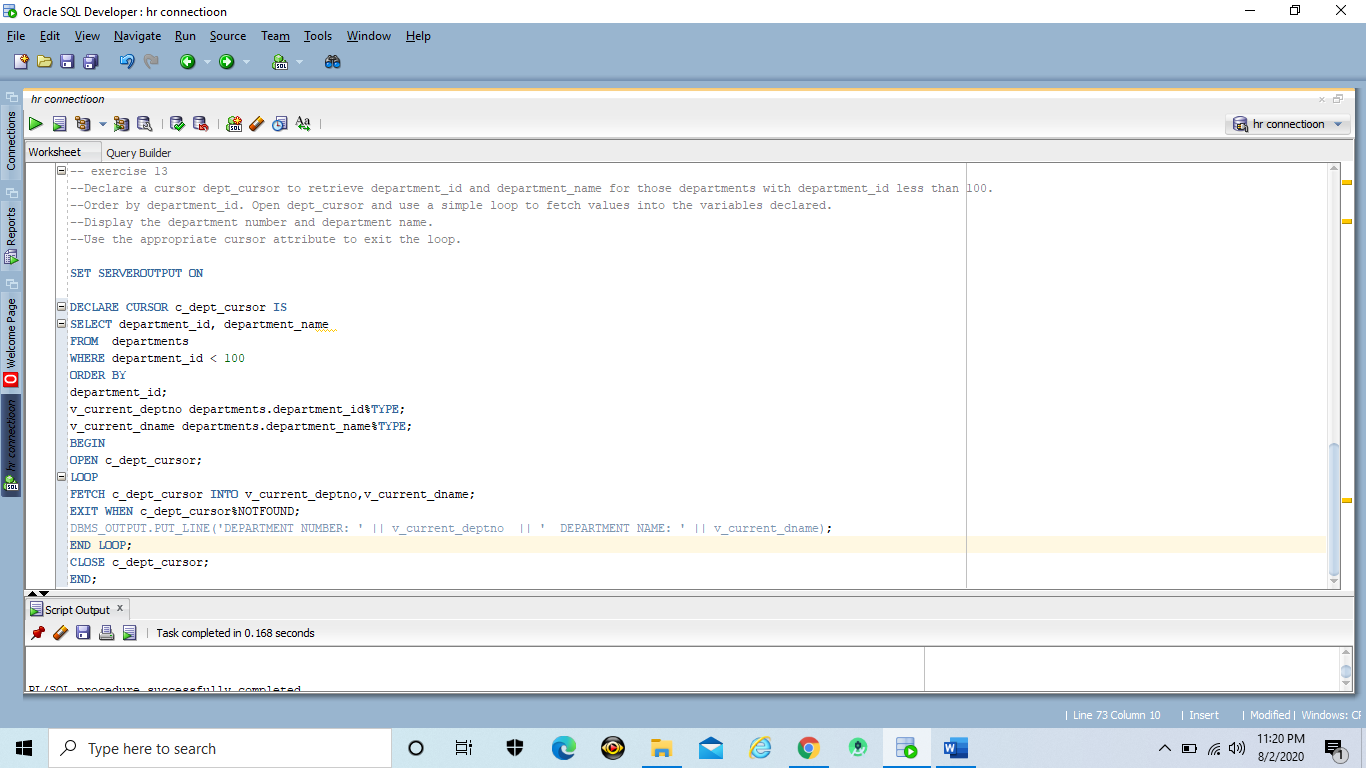
EXIT WHEN c\_dept\_cursor%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE('DEPARTMENT NUMBER: ' || v\_current\_deptno || ' DEPARTMENT NAME: ' || v\_current\_dname);

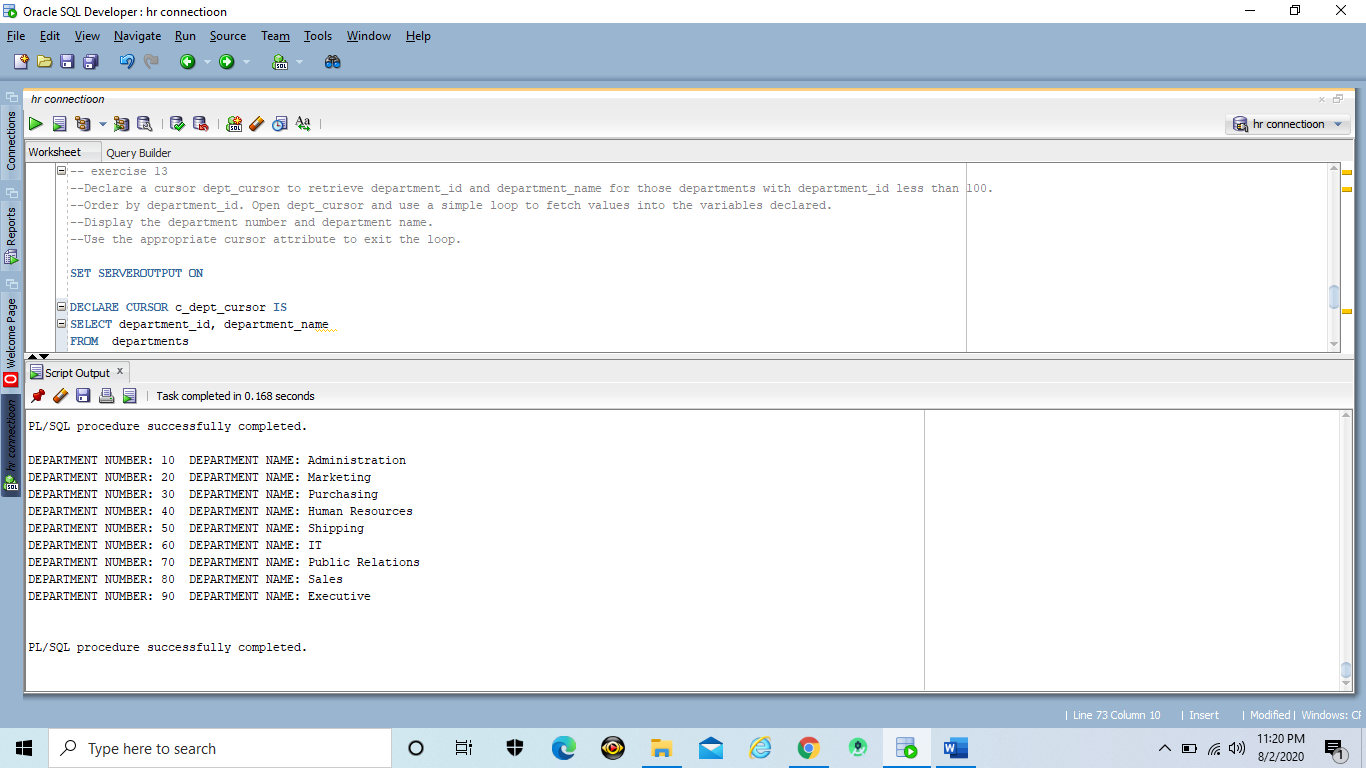
END LOOP;

CLOSE c\_dept\_cursor;

END;



Output



**Exercise 14 (1.2%)**

write a PL/SQL block that declares an exception for the Oracle Server error ORA-02292(integrity constraint violated – child record found). The block tests for the exception and outputs the error message.

In the executable section, display “You are not allowed to delete department number 60.” Include a DELETE statement to delete the department with department\_id 60.

SET SERVEROUTPUT ON

DECLARE e\_childrecord\_exists EXCEPTION;

PRAGMA EXCEPTION\_INIT(e\_childrecord\_exists,-02292);

BEGIN

DBMS\_OUTPUT.PUT\_LINE('DEPARTMENT is deleted whose id is 60');

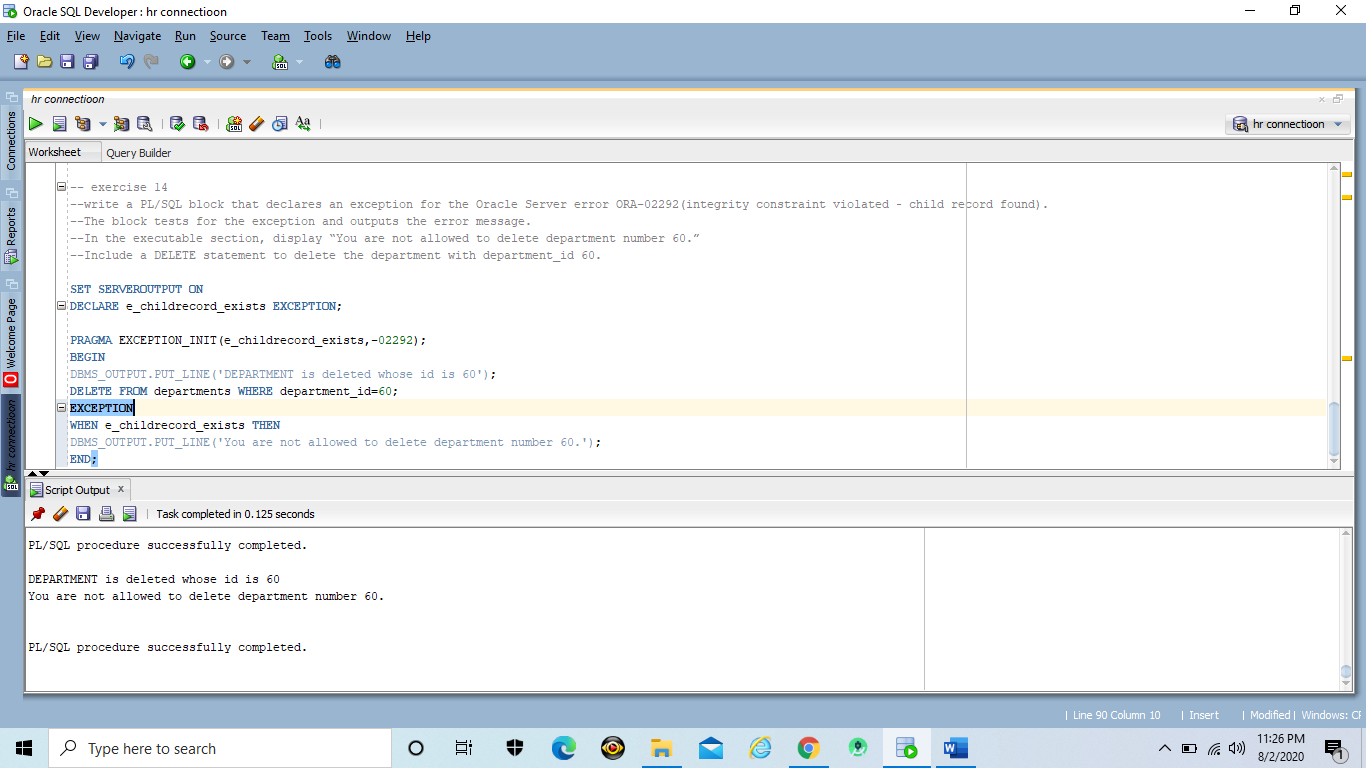
DELETE FROM departments WHERE department\_id=60;

EXCEPTION

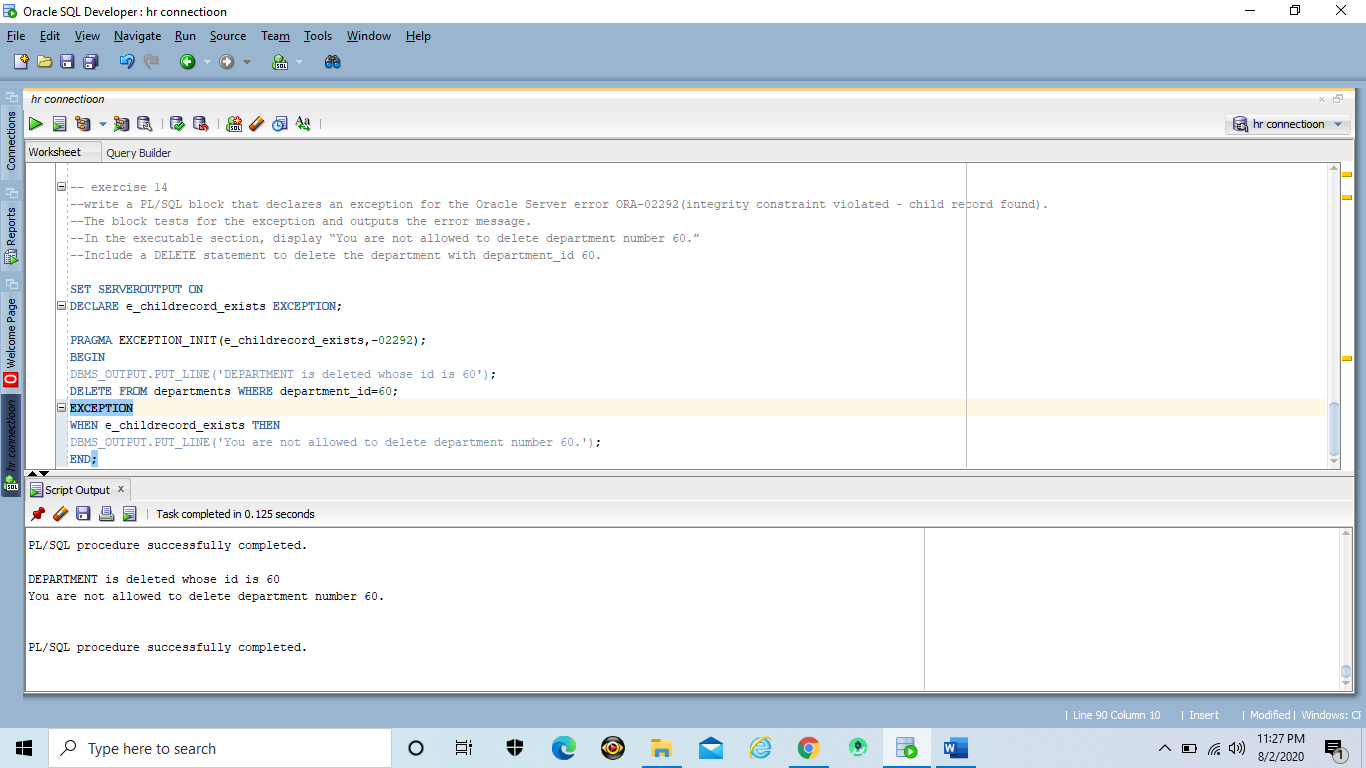
WHEN e\_childrecord\_exists THEN

DBMS\_OUTPUT.PUT\_LINE('You are not allowed to delete department number 60.');

END;



Output



**Exercise 15 (1.2%)**

Give 4 oracle pre-defined errors

CASE\_NOT\_FOUND

NO\_DATA\_FOUND

TOO\_MANY\_ROWS

ZERO\_DIVIDE

DUP\_VAL\_ON\_INDEX

CURSOR\_ALREADY\_OPEN

INVALID\_CURSOR

COLLECTION\_IS\_NULL

ROWTYPE\_MISMATCH