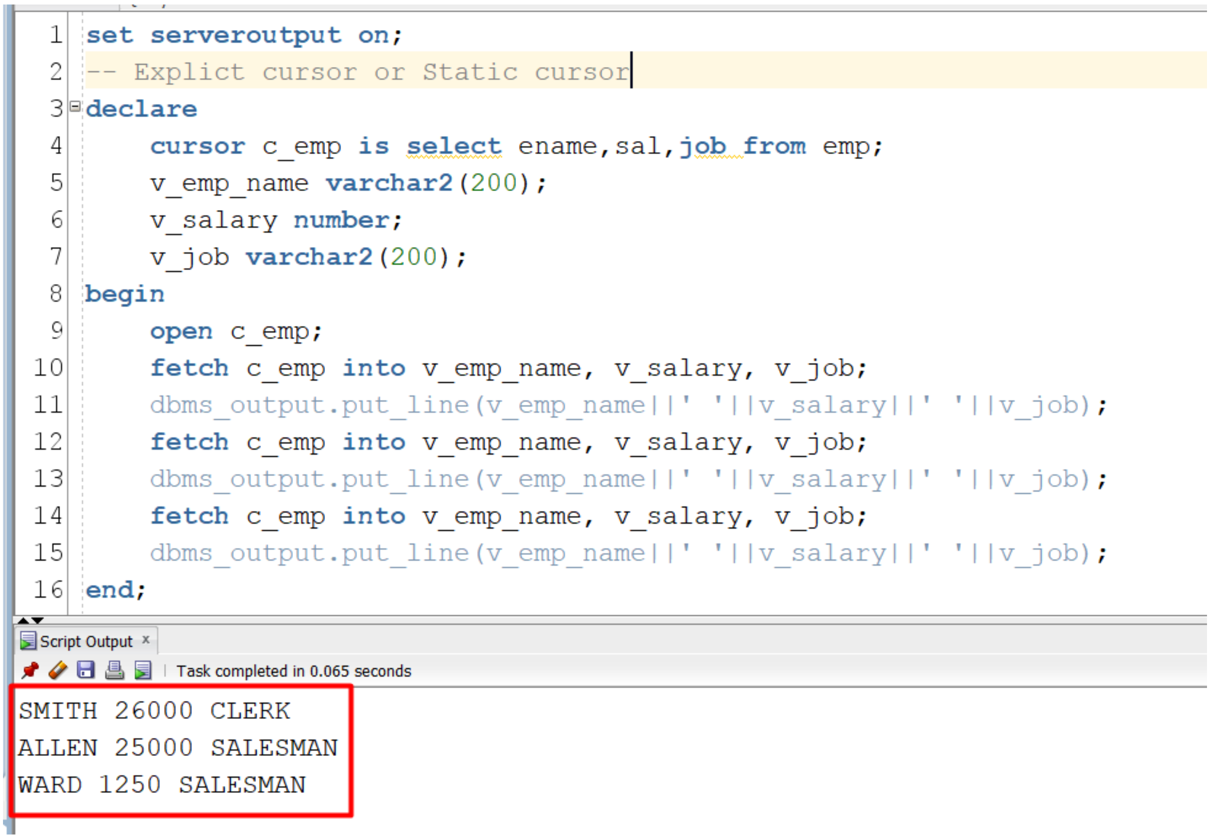
1. Static Cursor

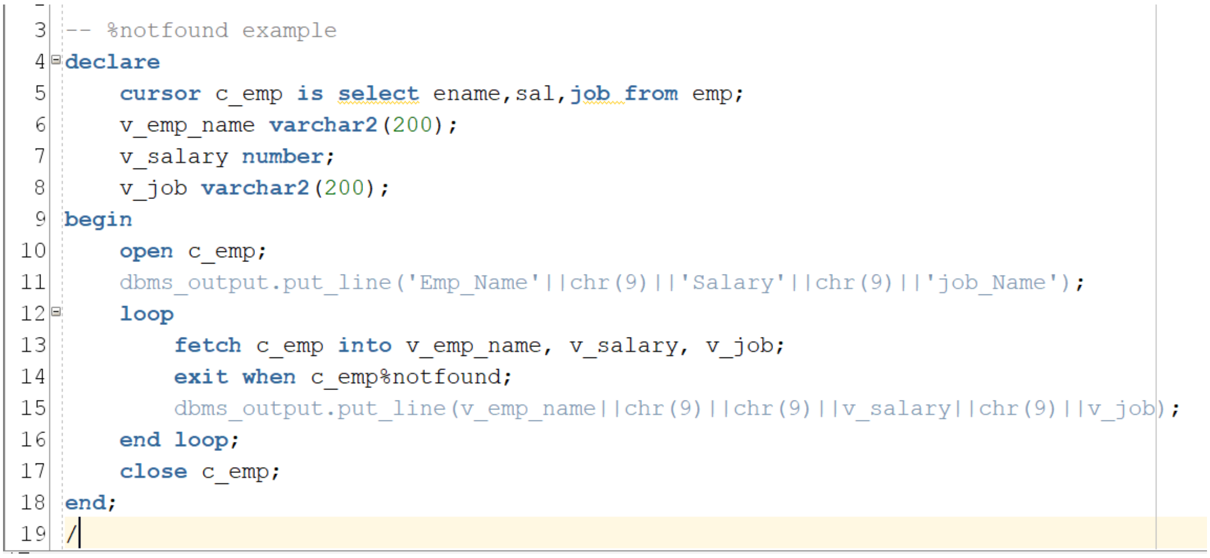


```
set serveroutput on;
-- Explicit cursor or Static cursor
declare
     cursor c_emp is select ename,sal,job from emp;
     v_emp_name varchar2(200);
     v_salary number;
     v_job varchar2(200);
begin
     open c_emp;
     fetch c_emp into v_emp_name, v_salary, v_job;
     dbms_output.put_line(v_emp_name||' '||v_salary||' '||v_job);
     fetch c_emp into v_emp_name, v_salary, v_job;
     dbms_output.put_line(v_emp_name||' '||v_salary||' '||v_job);
     fetch c_emp into v_emp_name, v_salary, v_job;
     dbms_output.put_line(v_emp_name||' '||v_salary||' '||v_job);
end;
```

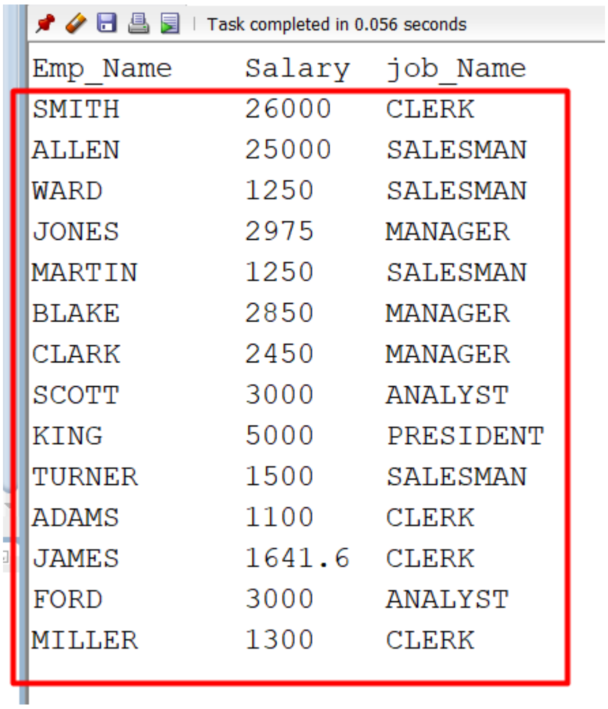
Task completed in 0.065 seconds

```
SMITH 26000 CLERK
ALLEN 25000 SALESMAN
WARD 1250 SALESMAN
```

2. %notfound example

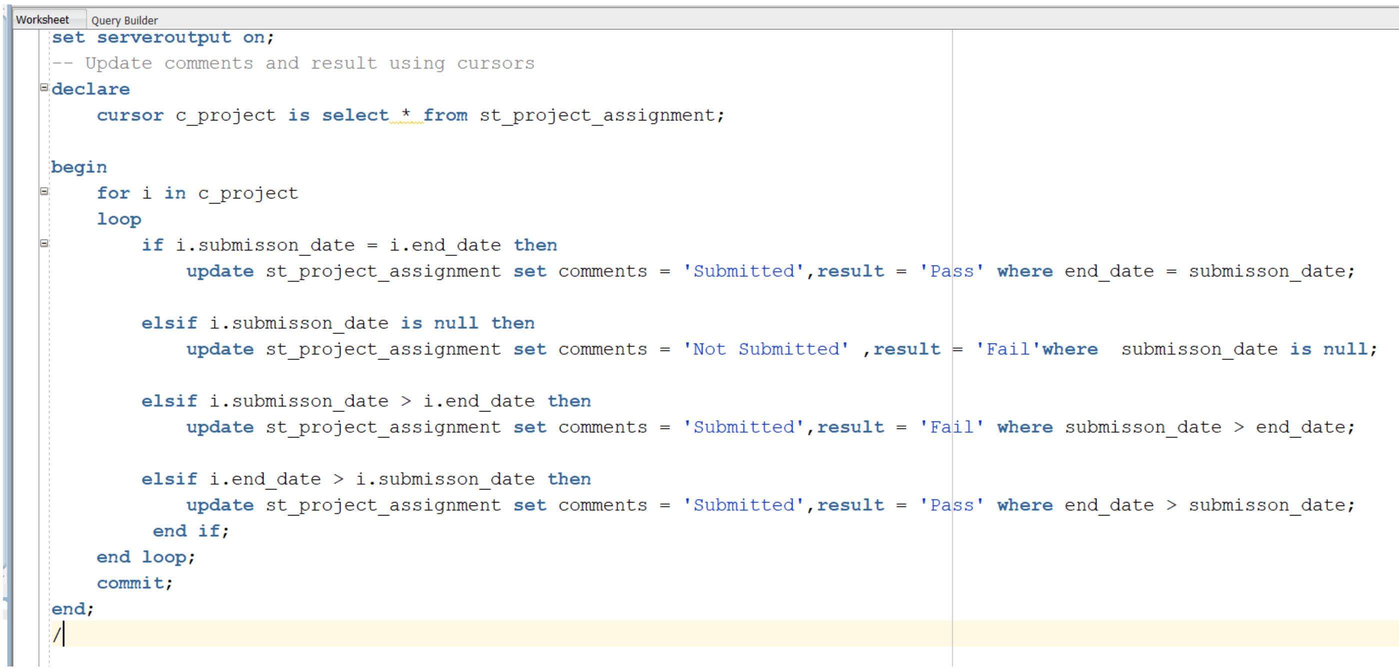


```
-- %notfound example
declare
     cursor c_emp is select ename,sal,job from emp;
     v_emp_name varchar2(200);
     v_salary number;
     v_job varchar2(200);
begin
     open c_emp;
     dbms_output.put_line('Emp_Name'||chr(9)||'Salary'||chr(9)||'job_Name');
     loop
         fetch c_emp into v_emp_name, v_salary, v_job;
         exit when c_emp%notfound;
         dbms_output.put_line(v_emp_name||chr(9)||chr(9)||v_salary||chr(9)||v_job);
     end loop;
     close c_emp;
end;
/
```

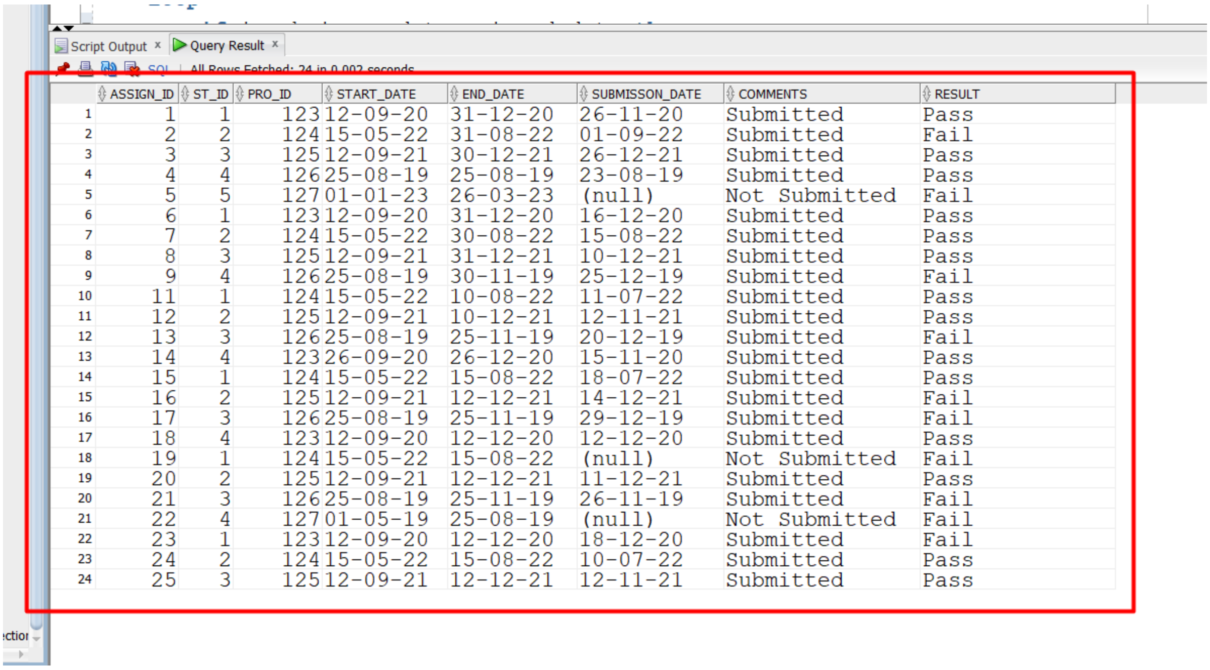


```
Emp_Name     Salary  job_Name
SMITH        26000   CLERK
ALLEN        25000   SALESMAN
WARD         1250    SALESMAN
JONES        2975    MANAGER
MARTIN       1250    SALESMAN
BLAKE        2850    MANAGER
CLARK        2450    MANAGER
SCOTT        3000    ANALYST
KING         5000    PRESIDENT
TURNER       1500    SALESMAN
ADAMS        1100    CLERK
JAMES        1641.6  CLERK
FORD         3000    ANALYST
MILLER       1300    CLERK
```

3. Update comments and result using cursors

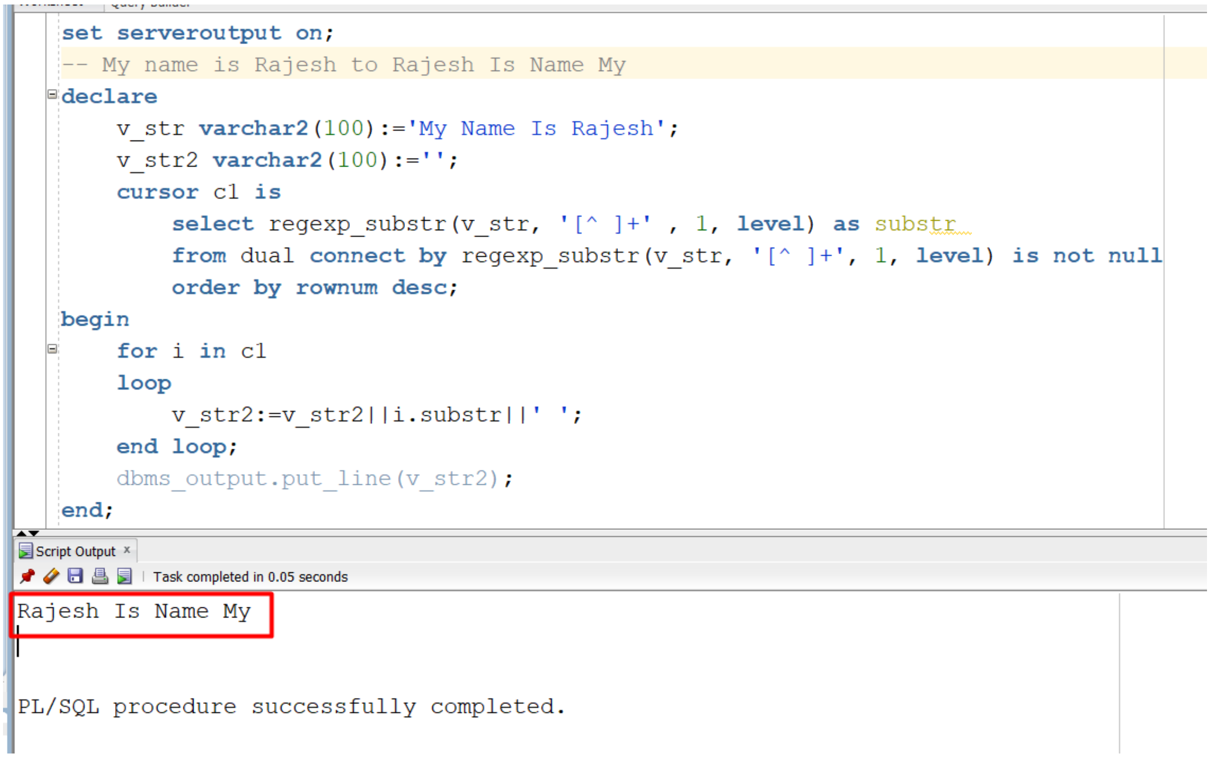


```
set serveroutput on;
-- Update comments and result using cursors
declare
    cursor c_project is select * from st_project_assignment;

begin
    for i in c_project
    loop
        if i.submisson_date = i.end_date then
            update st_project_assignment set comments = 'Submitted',result = 'Pass' where end_date = submisson_date;

        elsif i.submisson_date is null then
            update st_project_assignment set comments = 'Not Submitted' ,result = 'Fail'where  submisson_date is null;

        elsif i.submisson_date > i.end_date then
            update st_project_assignment set comments = 'Submitted',result = 'Fail' where submisson_date > end_date;

        elsif i.end_date > i.submisson_date then
            update st_project_assignment set comments = 'Submitted',result = 'Pass' where end_date > submisson_date;
         end if;
    end loop;
    commit;
end;
/
```



	ASSIGN_ID	ST_ID	PRO_ID	START_DATE	END_DATE	SUBMISSON_DATE	COMMENTS	RESULT
1	1	1	123	12-09-20	31-12-20	26-11-20	Submitted	Pass
2	2	2	124	15-05-22	31-08-22	01-09-22	Submitted	Fail
3	3	3	125	12-09-21	30-12-21	26-12-21	Submitted	Pass
4	4	4	126	25-08-19	25-08-19	23-08-19	Submitted	Pass
5	5	5	127	01-01-23	26-03-23	(null)	Not Submitted	Fail
6	6	1	123	12-09-20	31-12-20	16-12-20	Submitted	Pass
7	7	2	124	15-05-22	30-08-22	15-08-22	Submitted	Pass
8	8	3	125	12-09-21	31-12-21	10-12-21	Submitted	Pass
9	9	4	126	25-08-19	30-11-19	25-12-19	Submitted	Fail
10	11	1	124	15-05-22	10-08-22	11-07-22	Submitted	Pass
11	12	2	125	12-09-21	10-12-21	12-11-21	Submitted	Pass
12	13	3	126	25-08-19	25-11-19	20-12-19	Submitted	Fail
13	14	4	123	26-09-20	26-12-20	15-11-20	Submitted	Pass
14	15	1	124	15-05-22	15-08-22	18-07-22	Submitted	Pass
15	16	2	125	12-09-21	12-12-21	14-12-21	Submitted	Fail
16	17	3	126	25-08-19	25-11-19	29-12-19	Submitted	Fail
17	18	4	123	12-09-20	12-12-20	12-12-20	Submitted	Pass
18	19	1	124	15-05-22	15-08-22	(null)	Not Submitted	Fail
19	20	2	125	12-09-21	12-12-21	11-12-21	Submitted	Pass
20	21	3	126	25-08-19	25-11-19	26-11-19	Submitted	Fail
21	22	4	127	01-05-19	25-08-19	(null)	Not Submitted	Fail
22	23	1	123	12-09-20	12-12-20	18-12-20	Submitted	Fail
23	24	2	124	15-05-22	15-08-22	10-07-22	Submitted	Pass
24	25	3	125	12-09-21	12-12-21	12-11-21	Submitted	Pass

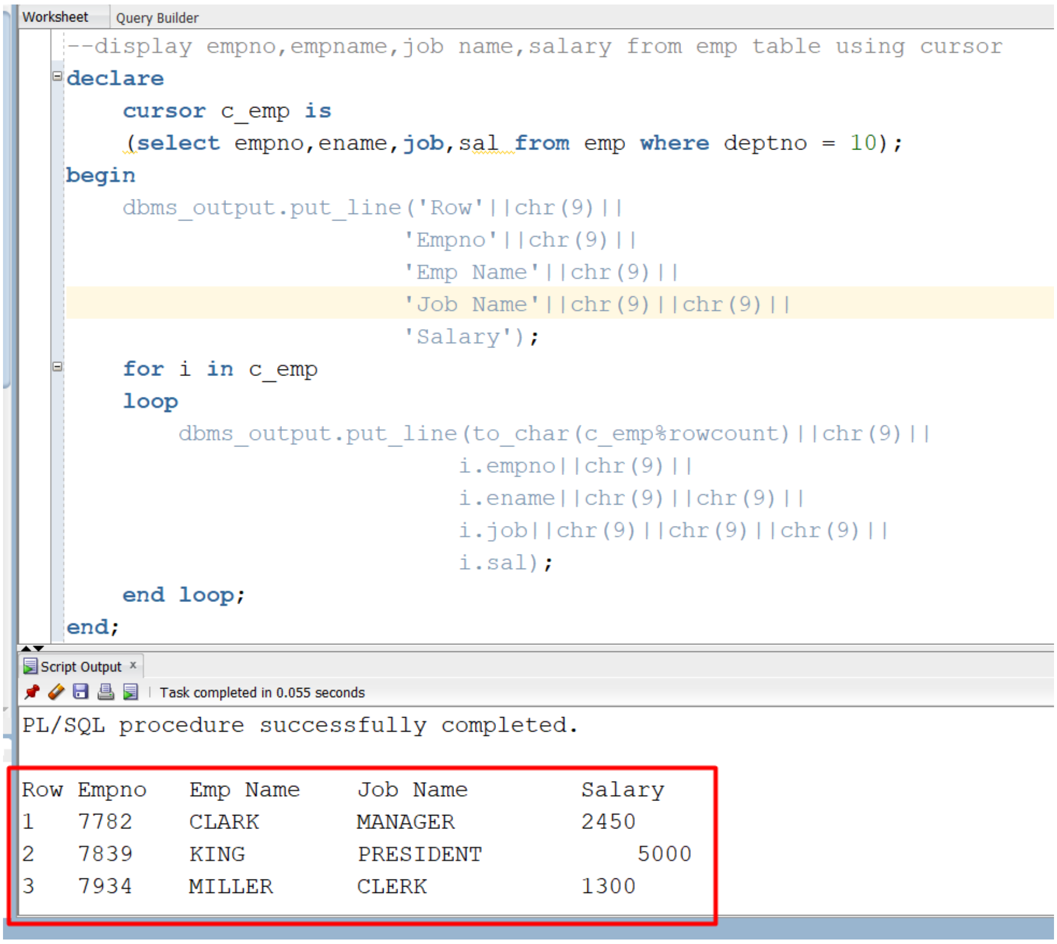
4. Reverse a string (Words)



```
set serveroutput on;
-- My name is Rajesh to Rajesh Is Name My
declare
    v_str varchar2(100):='My Name Is Rajesh';
    v_str2 varchar2(100):='';
    cursor c1 is
        select regexp_substr(v_str, '[^ ]+' , 1, level) as substr
        from dual connect by regexp_substr(v_str, '[^ ]+', 1, level) is not null
        order by rownum desc;
begin
    for i in c1
    loop
        v_str2:=v_str2||i.substr||' ';
    end loop;
    dbms_output.put_line(v_str2);
end;
```

```
Rajesh Is Name My


PL/SQL procedure successfully completed.
```

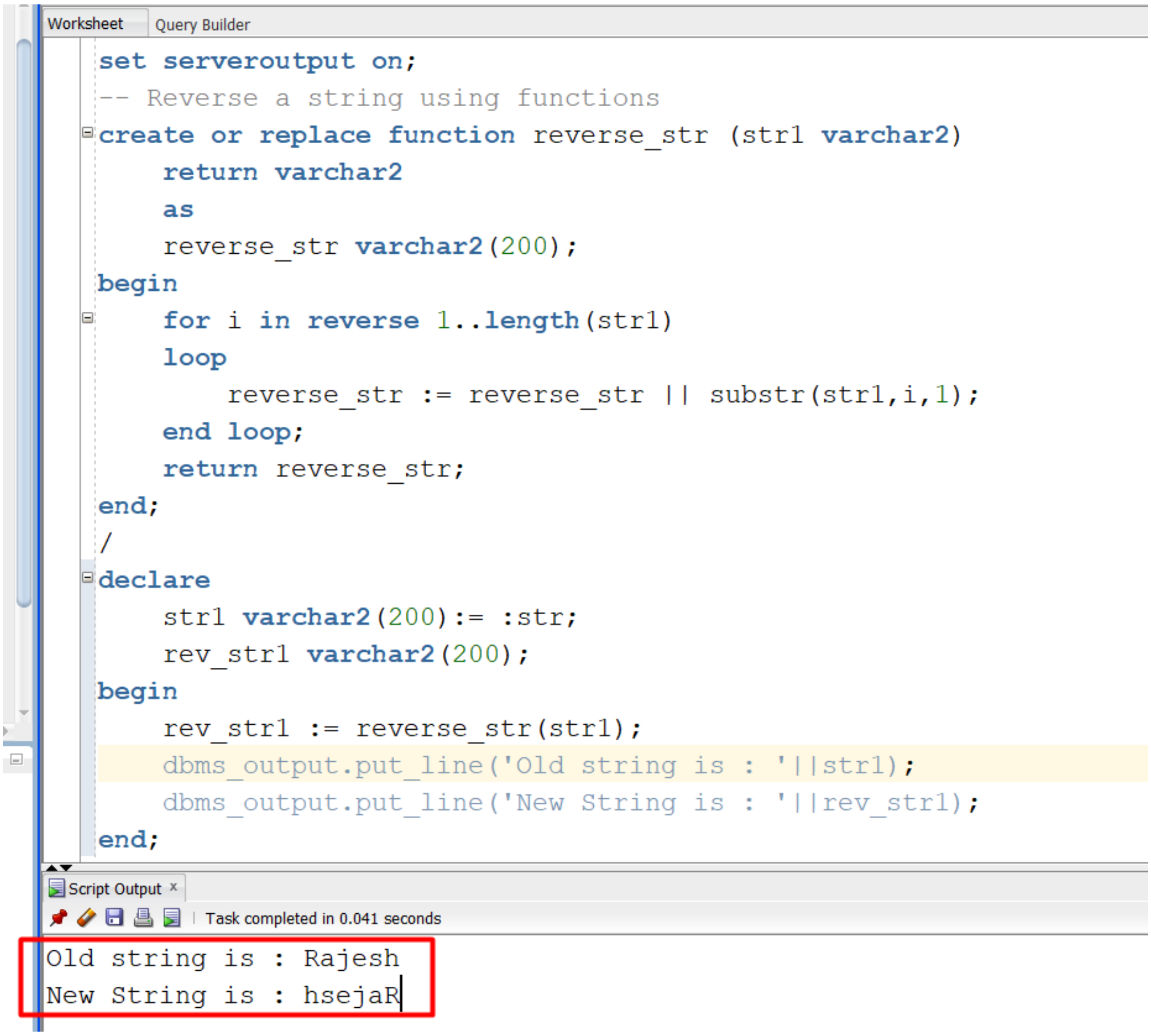
5. Display empno, empname, jobname, salary from emp table using cursor



```
--display empno,empname,job name,salary from emp table using cursor
declare
    cursor c_emp is
    (select empno,ename,job,sal from emp where deptno = 10);
begin
    dbms_output.put_line('Row'||chr(9)||
                        'Empno'||chr(9)||
                        'Emp Name'||chr(9)||
                        'Job Name'||chr(9)||chr(9)||
                        'Salary');
    for i in c_emp
    loop
        dbms_output.put_line(to_char(c_emp%rowcount)||chr(9)||
                            i.empno||chr(9)||
                            i.ename||chr(9)||chr(9)||
                            i.job||chr(9)||chr(9)||chr(9)||
                            i.sal);
    end loop;
end;
```

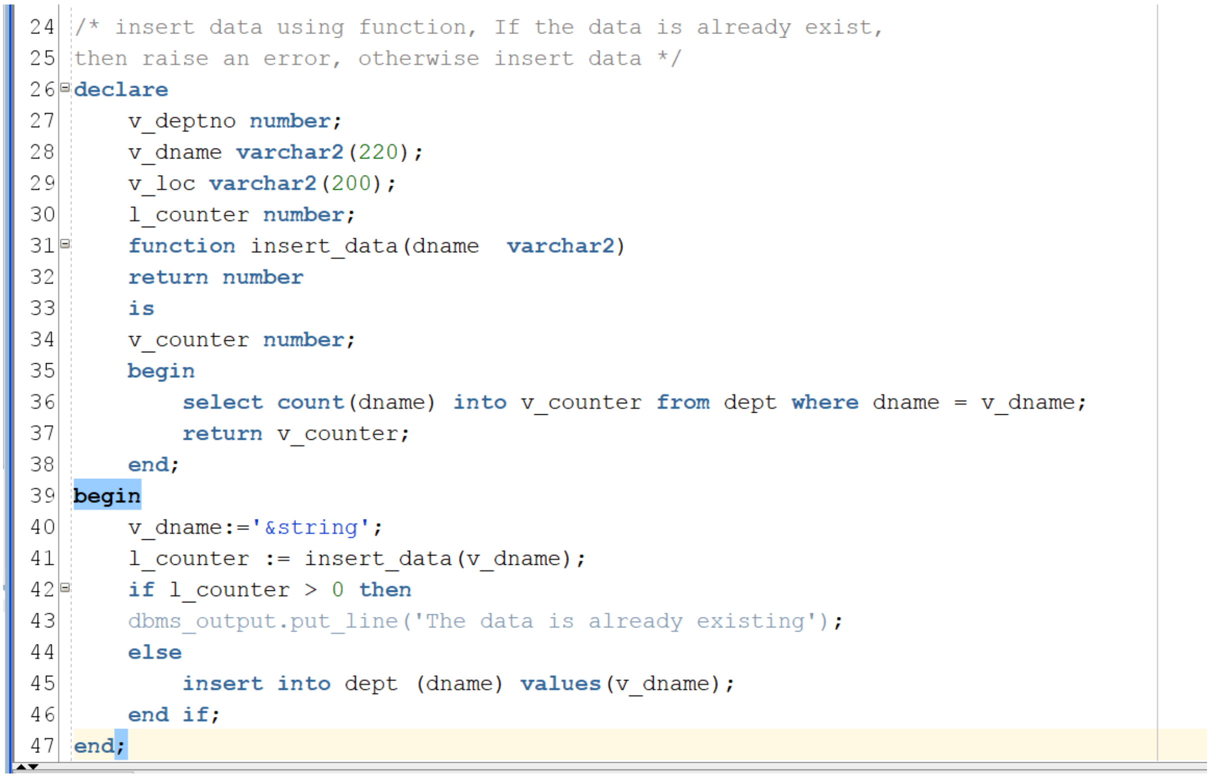
```
PL/SQL procedure successfully completed.

Row Empno   Emp Name    Job Name        Salary
1   7782    CLARK       MANAGER         2450
2   7839    KING        PRESIDENT           5000
3   7934    MILLER      CLERK           1300
```

6. Reverse a string using function

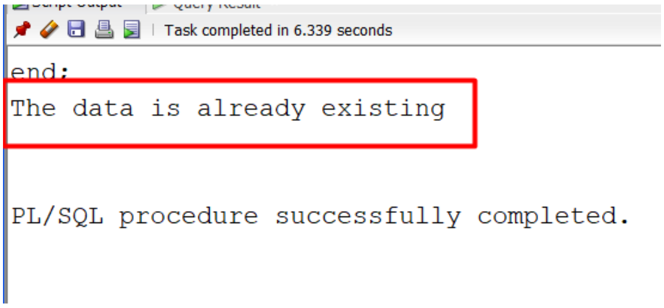


```
set serveroutput on;
-- Reverse a string using functions
create or replace function reverse_str (str1 varchar2)
    return varchar2
    as
    reverse_str varchar2(200);
begin
    for i in reverse 1..length(str1)
    loop
        reverse_str := reverse_str || substr(str1,i,1);
    end loop;
    return reverse_str;
end;
/
declare
    str1 varchar2(200):= :str;
    rev_str1 varchar2(200);
begin
    rev_str1 := reverse_str(str1);
    dbms_output.put_line('Old string is : '||str1);
    dbms_output.put_line('New String is : '||rev_str1);
end;
```

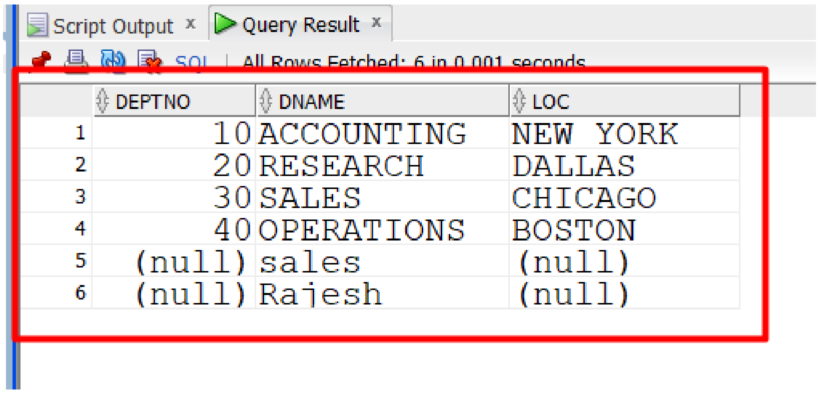
```
Old string is : Rajesh
New String is : hsejaR
```



```
/* insert data using function, If the data is already exist,
then raise an error, otherwise insert data */
declare
    v_deptno number;
    v_dname varchar2(220);
    v_loc varchar2(200);
    l_counter number;
    function insert_data(dname  varchar2)
    return number
    is
    v_counter number;
    begin
        select count(dname) into v_counter from dept where dname = v_dname;
        return v_counter;
    end;
begin
    v_dname:='&string';
    l_counter := insert_data(v_dname);
    if l_counter > 0 then
    dbms_output.put_line('The data is already existing');
    else
        insert into dept (dname) values(v_dname);
    end if;
end;
```



```
end;
The data is already existing


PL/SQL procedure successfully completed.
```



	DEPTNO	DNAME	LOC
1	10	ACCOUNTING	NEW YORK
2	20	RESEARCH	DALLAS
3	30	SALES	CHICAGO
4	40	OPERATIONS	BOSTON
5	(null)	sales	(null)
6	(null)	Rajesh	(null)