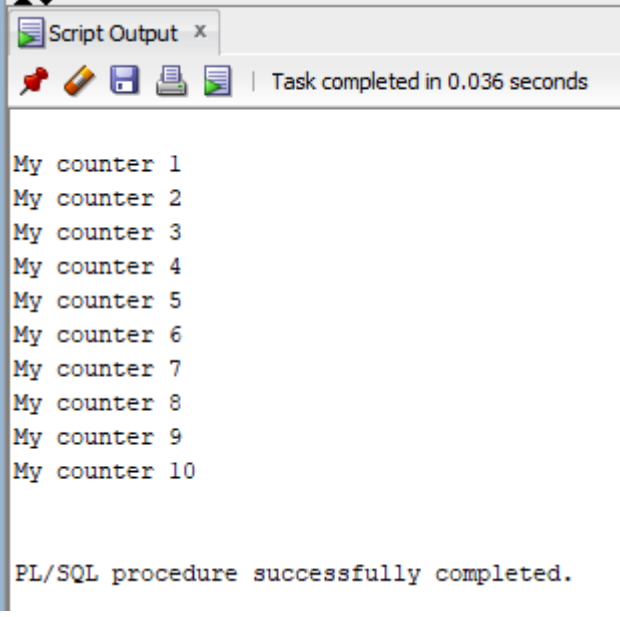


For Loops:

Example:

```
set serveroutput on;
begin
for i in 1..10 loop
  dbms_output.put_line('My counter '||i);
end loop;
end;
```

A screenshot of a 'Script Output' window from a database client. The window title is 'Script Output' with a close button. Below the title bar is a toolbar with icons for a red pin, a pencil, a save icon, a print icon, and a document icon. To the right of the toolbar, it says 'Task completed in 0.036 seconds'. The main area of the window displays the output of a PL/SQL script. It lists 'My counter' followed by numbers 1 through 10, each on a new line. At the bottom, it says 'PL/SQL procedure successfully completed.'

```
My counter 1
My counter 2
My counter 3
My counter 4
My counter 5
My counter 6
My counter 7
My counter 8
My counter 9
My counter 10

PL/SQL procedure successfully completed.
```

```
set serveroutput on;
```

```
declare
```

```
v_start number(2):=1;
```

```
v_end number(2):=15;
```

```
begin
```

```
for i in v_start..v_end loop
```

```
  dbms_output.put_line('Counter: '||i);
```

```
end loop;
```

```
end;
```

```
PL/SQL procedure successfully completed.
```

```
Counter: 1  
Counter: 2  
Counter: 3  
Counter: 4  
Counter: 5  
Counter: 6  
Counter: 7  
Counter: 8  
Counter: 9  
Counter: 10  
Counter: 11  
Counter: 12
```

Nested Loops:

```
set serveroutput on;
```

```
declare  
v_inner number(2):=1;  
begin  
  <<outer_loop>>  
  for v_outer in 1..4 loop  
    dbms_output.put_line('Outer Count: '||v_outer);  
    v_inner:=1;  
    <<inner_loop>>  
    loop  
      v_inner := v_inner+1;  
      dbms_output.put_line('Inner Count: '||v_inner);  
      exit when v_inner*v_outer>=15;  
    end loop;  
  end loop;  
end;
```

```
Outer Count: 1  
Inner Count: 2  
Inner Count: 3  
Inner Count: 4  
Inner Count: 5  
Inner Count: 6  
Inner Count: 7  
Inner Count: 8
```

Inner Count: 9
Inner Count: 10
Inner Count: 11
Inner Count: 12
Inner Count: 13
Inner Count: 14
Inner Count: 15
Outer Count: 2
Inner Count: 2
Inner Count: 3
Inner Count: 4
Inner Count: 5
Inner Count: 6
Inner Count: 7
Inner Count: 8
Outer Count: 3
Inner Count: 2
Inner Count: 3
Inner Count: 4
Inner Count: 5
Outer Count: 4
Inner Count: 2
Inner Count: 3
Inner Count: 4

PL/SQL procedure successfully completed.

```
set serveroutput on;
declare
v_inner number(2):=1;
begin
for s in 1..10 loop
  dbms_output.put_line('Outer Value: '||s);
  v_inner:=1;
  while v_inner*s <15 loop
    v_inner:=v_inner+1;
    continue when mod(v_inner*s,3)=0;
    dbms_output.put_line('Inner Value: '||v_inner);
  end loop;
end loop;
end;
```

Outer Value: 1
Inner Value: 2
Inner Value: 4
Inner Value: 5
Inner Value: 7

Inner Value: 8
Inner Value: 10
Inner Value: 11
Inner Value: 13
Inner Value: 14
Outer Value: 2
Inner Value: 2
Inner Value: 4
Inner Value: 5
Inner Value: 7
Inner Value: 8
Outer Value: 3
Outer Value: 4
Inner Value: 2
Inner Value: 4
Outer Value: 5
Inner Value: 2
Outer Value: 6
Outer Value: 7
Inner Value: 2
Outer Value: 8
Inner Value: 2
Outer Value: 9
Outer Value: 10
Inner Value: 2

PL/SQL procedure successfully completed.

```
set serveroutput on;
declare
v_searchnum number:=22;
is_prime boolean:=true;
s number:=2;

begin
<<start_point>>
if v_searchnum mod s=0 then
dbms_output.put_line('Not Prime Number');

is_prime:=false;
goto end_point;
end if;
s:=s+1;

if s=v_searchnum then
goto prime_point;
end if;
```

```
goto start_point;
```

```
<<prime_point>>
```

```
if is_prime then
```

```
dbms_output.put_line('It is a Prime Number');
```

```
end if;
```

```
<<end_point>>
```

```
dbms_output.put_line('Check Complete');
```

```
end;
```

```
*Action:
```

```
Not Prime Number
```

```
Check Complete
```

```
PL/SQL procedure successfully completed.
```

To work on database:

```
set serveroutput on;
```

```
declare
```

```
v_empname VARCHAR2(50);
```

```
v_empsalary number(10);
```

```
begin
```

```
select last_name,salary into v_empname,v_empsalary from hr.employees where  
employee_id=100;
```

```
dbms_output.put_line('The Salary of '||v_empname||'is: '||v_empsalary);
```

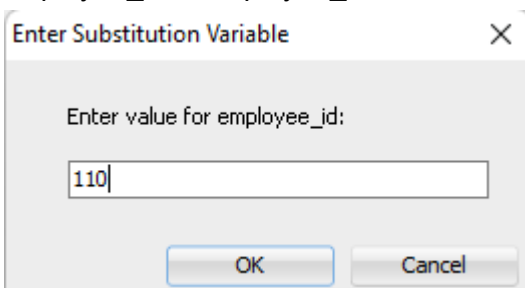
```
End;
```

```
The Salary of Kingis: 24000
```

```
PL/SQL procedure successfully completed.
```

To get input from user

```
select last_name,salary into v_empname,v_empsalary from hr.employees where  
employee_id=&employee_id;
```



```
The Salary of Chenis: 8200
```

```
PL/SQL procedure successfully completed.
```

DML Operations:


Insert,update,delete,modify.

Queries as follows:

--- to create table copy

```
create table employee_copy as select * from hr.employees;
begin
for s in 120..125 loop
insert into employee_copy(employee_id,first_name,last_name,email,hire_date,job_id,salary)
values
(&employee_id,&first_name,&last_name,&email',sysdate,&job_id',&salary);
end loop;
end;
```

Script Output x Query Result x

 All Rows Fetched: 7 in 0.006 seconds

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID	
1	121	Adam	Fripp	AFRIPP	650.123.2234	10-APR-05	ST_MAN	8200	(null)	100	50
2	121	mahesh	vaithi	mahesh@gmail.com	(null)	21-DEC-23	IT_PROG	10000	(null)	(null)	(null)
3	121	mahesh	vaithi	mahesh@gmail.com	(null)	21-DEC-23	IT_PROG	10000	(null)	(null)	(null)
4	121	mahesh	vaithi	mahesh@gmail.com	(null)	21-DEC-23	IT_PROG	10000	(null)	(null)	(null)
5	121	mahesh	vaithi	mahesh@gmail.com	(null)	21-DEC-23	IT_PROG	10000	(null)	(null)	(null)
6	121	mahesh	vaithi	mahesh@gmail.com	(null)	21-DEC-23	IT_PROG	10000	(null)	(null)	(null)
7	121	mahesh	vaithi	mahesh@gmail.com	(null)	21-DEC-23	IT_PROG	10000	(null)	(null)	(null)

--to drop that value

```
delete from employee_copy where employee_id=121;
```

***Action:**

7 rows deleted.

Sequence in PLSQL:

```
create sequence seq_empid start with 80 increment by 1;
```

To create sequence

```
begin
for s in 1..10 loop
insert into emp(employee_id,first_name,last_name,email,hire_date,job_id,salary) values
(seq_empid.nextval,&first_name',&last_name',&email',sysdate,&job_id',&salary);
end loop;
end;
```

```
select * from emp;
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
81	mahesh	vaithi	mahes...	(null)	21-DEC-23	WEB_DEV	20000	(null)	(null)	(null)
82	mahesh	vaithi	mahes...	(null)	21-DEC-23	WEB_DEV	20000	(null)	(null)	(null)
83	mahesh	vaithi	mahes...	(null)	21-DEC-23	WEB_DEV	20000	(null)	(null)	(null)
84	mahesh	vaithi	mahes...	(null)	21-DEC-23	WEB_DEV	20000	(null)	(null)	(null)
85	mahesh	vaithi	mahes...	(null)	21-DEC-23	WEB_DEV	20000	(null)	(null)	(null)
86	mahesh	vaithi	mahes...	(null)	21-DEC-23	WEB_DEV	20000	(null)	(null)	(null)
87	mahesh	vaithi	mahes...	(null)	21-DEC-23	WEB_DEV	20000	(null)	(null)	(null)
88	mahesh	vaithi	mahes...	(null)	21-DEC-23	WEB_DEV	20000	(null)	(null)	(null)
89	mahesh	vaithi	mahes...	(null)	21-DEC-23	WEB_DEV	20000	(null)	(null)	(null)

Here the employee id from 81 to 89

– records in plsql:

```

declare
r_emp emp%rowtype;

begin
select * into r_emp from emp where employee_id=120;

dbms_output.put_line('The Name from the Record: '||r_emp.first_name||
'||r_emp.last_name);

end;

```

```

*Action:
The Name from the Record: Matthew Weiss

PL/SQL procedure successfully completed.

```

--- record type in plsql

```

declare
type typ_emp is record (first_name varchar(50),last_name varchar(30),salary
number(10),hire_date date);
r_emp typ_emp;
begin

select first_name,last_name,salary,hire_date into r_emp from emp where
employee_id='120';
dbms_output.put_line(r_emp.first_name||' '||r_emp.last_name||' is earning '||r_emp.salary||'
per year'||' hired at '||r_emp.hire_date);
End;

```

```
Matthew Weiss is earning 8000 per year hired at 18-JUL-04
```

```
PL/SQL procedure successfully completed.
```

--- easy dml with records

```
create table retired_emp as select * from hr.employees where 1 = 2;
```

```
select * from retired_emp;
```

```
declare
```

```
r_emp emp%rowtype;
```

```
begin
```

```
select * into r_emp from emp where employee_id=104;
```

```
r_emp.salary:=100;
```

```
r_emp.commission_pct:=0;
```

```
--- insert into retired_emp values r_emp;
```

```
update retired_emp set row=r_emp where employee_id=104;
```

```
end;
```

--- select * from retired_emp;

Script Output x Query Result x

SQL | All Rows Fetched: 1 in 0.001 seconds

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID	
1	104	Bruce	Ernst	BERNST	590.423.4568	21-MAY-07	IT_PROG	100	0	103	60

--- v arrays

```
declare
```

```
type emp_list is varray(5) of varchar(50);
```

```
emp emp_list;
```

```
begin
```

```
emp :=emp_list('Alice','Shyam','Ashi','Durga','Vijay');
```

```
for i in 1..5 loop
```

```
if emp.exists(i) then
```

```
dbms_output.put_line(emp(i));
```

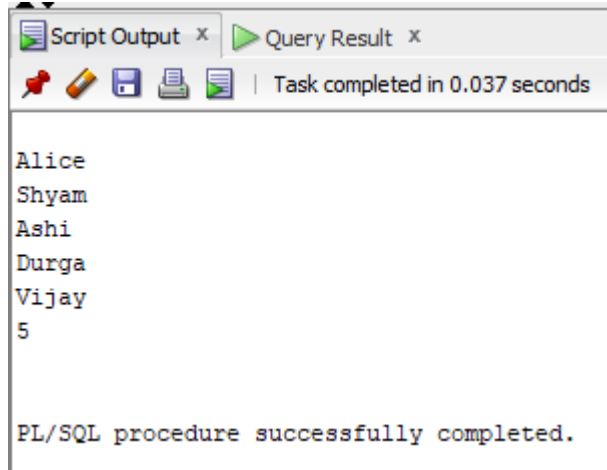


```
end if;
```

```
end loop;
```

```
dbms_output.put_line(emp.limit());
```

```
end;
```



```
--- v arrays
```

```
declare
```

```
type emp_list is varray(15) of varchar(50);
```

```
empl emp_list :=emp_list();
```

```
idx number:=1;
```

```
begin
```

```
for i in 120..125 loop
```

```
empl.extend;
```

```
select first_name into empl(idx) from emp where employee_id=i;
```

```
idx:=idx+1;
```

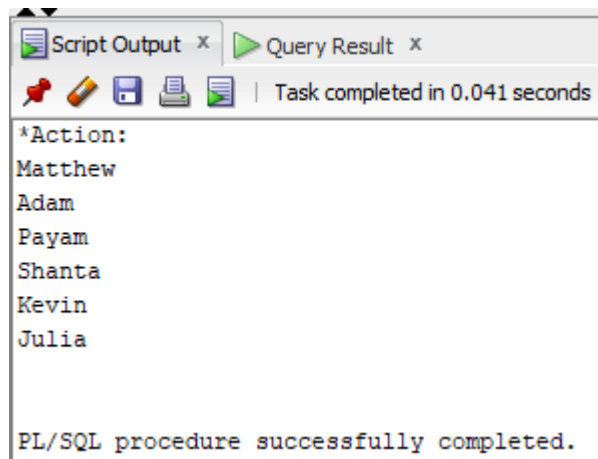
```
end loop;
```

```
for x in 1..empl.count() loop
```

```
dbms_output.put_line(empl(x));
```

```
end loop;
```

```
end;
```



Script Output x Query Result x

Task completed in 0.041 seconds

```
*Action:
Matthew
Adam
Payam
Shanta
Kevin
Julia

PL/SQL procedure successfully completed.
```

--- nested tables

declare

```
type e_lst is table of varchar(50);
```

```
emps e_lst:=e_lst();
```

```
idx pls_integer:=1;
```

```
begin
```

```
for x in 110..114 loop
```

```
emps.extend;
```

```
select first_name into emps(idx) from emp where employee_id=x;
```

```
idx:=idx+1;
```

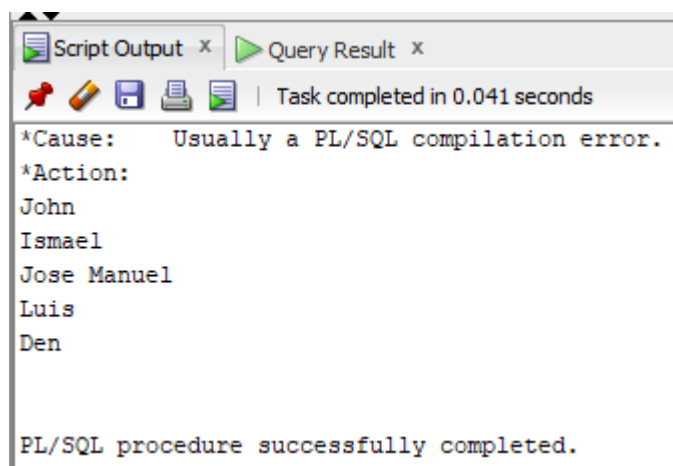
```
end loop;
```

```
for i in 1..emps.count() loop
```

```
dbms_output.put_line(emps(i));
```

```
end loop;
```

```
end;
```



Script Output x Query Result x

Task completed in 0.041 seconds

```
*Cause: Usually a PL/SQL compilation error.
*Action:
John
Ismael
Jose Manuel
Luis
Den

PL/SQL procedure successfully completed.
```

--- associative array

```
declare
```

```
type elist is table of emp.first_name%type index by pls_integer;
```

```
emps elist;
```

```
begin
```

```
for x in 130..134 loop
```

```
select first_name into emps(x) from emp where employee_id=x and department_id=60;
```

```
end loop;
```

```
for s in 1..emps.count() loop
```

```
dbms_output.put_line(emps(s));
```

```
end loop;
```

```
end;
```

```
declare
```

```
type elist is table of emp.first_name%type index by pls_integer;
```

```
emps elist;
```

```
idx pls_integer;
```

```
begin
```

```
emps(100):='Bob';
```

```
emps(102):='Mob';
```

```
idx:=emps.first();
```

```
---select first_name into emps(x) from emp where employee_id=x and department_id=60;
```

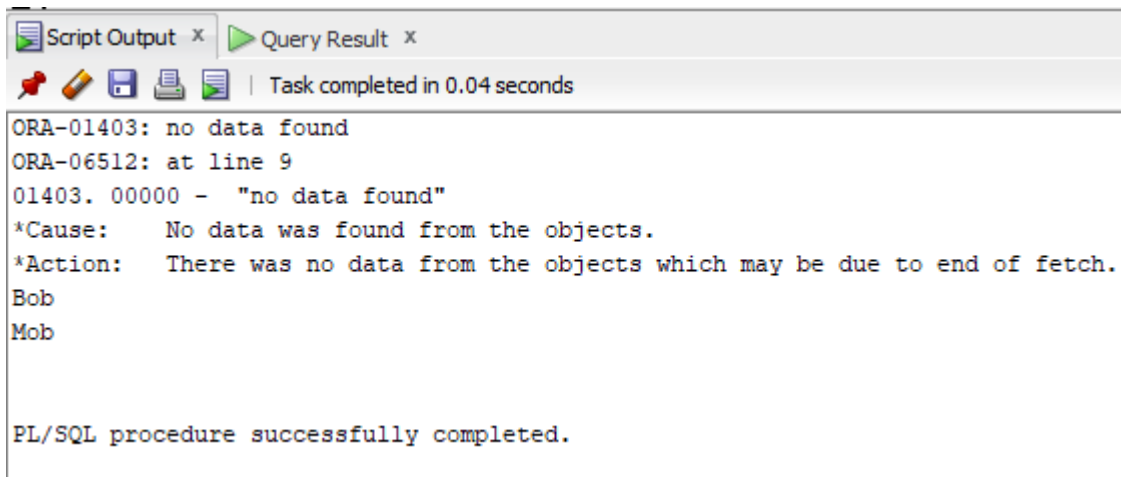
```
while idx is not null loop
```

```
dbms_output.put_line(emps(idx));
```

```
idx:=emps.next(idx);
```

```
end loop;
```

```
end;
```



The screenshot shows a window with two tabs: "Script Output" and "Query Result". Below the tabs is a toolbar with icons for a red pin, a pencil, a save icon, a print icon, and a refresh icon. To the right of the toolbar, it says "Task completed in 0.04 seconds". The main area of the window displays the following text:

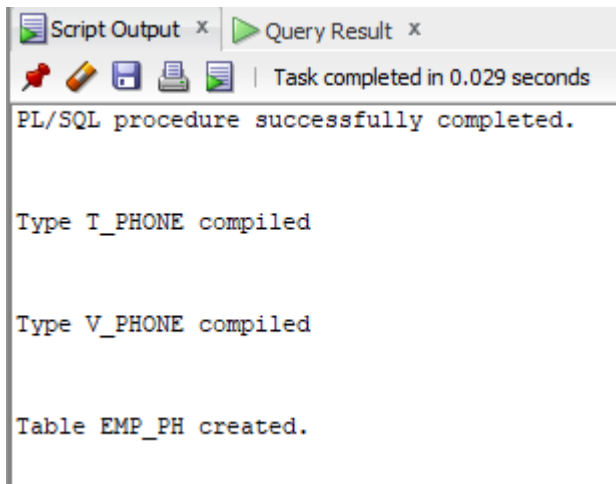
```
ORA-01403: no data found
ORA-06512: at line 9
01403. 00000 - "no data found"
*Cause:      No data was found from the objects.
*Action:     There was no data from the objects which may be due to end of fetch.
Bob
Mob

PL/SQL procedure successfully completed.
```

```
create or replace type t_phone as object(ph_type varchar(20),ph_no varchar(30));
```

```
create or replace type v_phone as varray(3) of t_phone;
```

```
create table emp_ph(employee_id number,first_name varchar(30),last_name  
varchar(30),phone_num v_phone);
```



```
select * from emp_ph;
```

The screenshot shows the 'Query Result' window in SQL Developer. It displays the results of the query 'select * from emp_ph;'. The results are shown in a table with 4 columns: EMPLOYEE_ID, FIRST_NAME, LAST_NAME, and PHONE_NUM. There are 3 rows of data. The status bar indicates 'All Rows Fetched: 3 in 0.009 seconds'.

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	PHONE_NUM
1	101 Mahesh	vaithi	SYSTEM.V_PHONE ([SYS...
2	161 Nijan	Jain	SYSTEM.V_PHONE ([SYS...
3	121 Ashi	Udhay	SYSTEM.V_PHONE ([SYS...

Cursor and its Types:

```
declare  
cursor c_emps is select first_name,last_name from emp;  
fname emp.first_name%type;  
lname emp.last_name%type;  
begin  
open c_emps;  
fetch c_emps into fname,lname;  
dbms_output.put_line('The Name in the record: '||fname||' '||lname);  
close c_emps;  
end;
```

```
*Cause:    Usually a PL/SQL compilation error.
*Action:
The Name in the record: Steven King
```

PL/SQL procedure successfully completed.

--- using cursor with records

declare

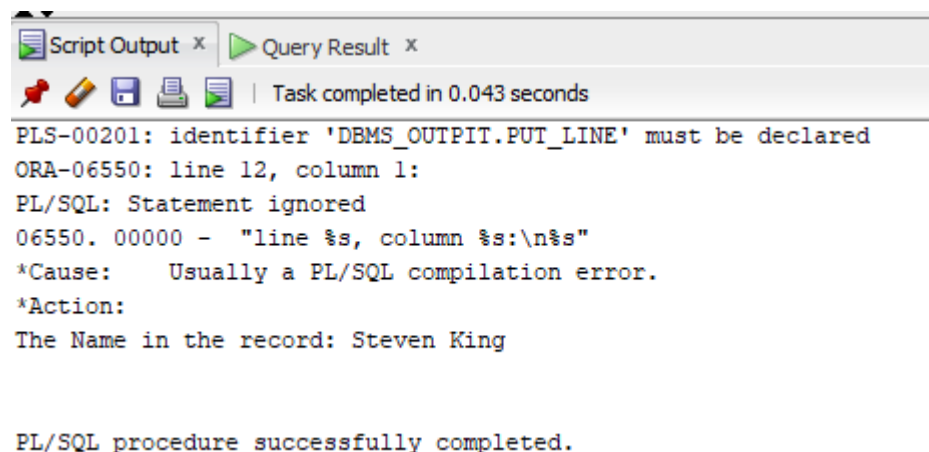
```
type t_emp is record(fname emp.first_name%type,lname emp.last_name%type);
v_emp t_emp;
cursor c_emp is select first_name,last_name from emp;
```

begin

open c_emp;

fetch c_emp into v_emp;

```
dbms_output.put_line('The Name in the record: '||v_emp.fname||' '||v_emp.lname);
close c_emp;
end;
```



The screenshot shows a window titled 'Script Output' with a sub-tab 'Query Result'. Below the title bar, there are icons for a red pin, a yellow pencil, a blue folder, a printer, and a green document. To the right of these icons, it says 'Task completed in 0.043 seconds'. The main area of the window displays the following text:

```
PLS-00201: identifier 'DBMS_OUTPUT.PUT_LINE' must be declared
ORA-06550: line 12, column 1:
PL/SQL: Statement ignored
06550. 00000 - "line %s, column %s:\n%s"
*Cause:    Usually a PL/SQL compilation error.
*Action:
The Name in the record: Steven King

PL/SQL procedure successfully completed.
```

--- loops in cursor

declare

```
cursor c_emp is select first_name,last_name from emp where department_id=30;
v_emp c_emp%rowtype;
```

begin

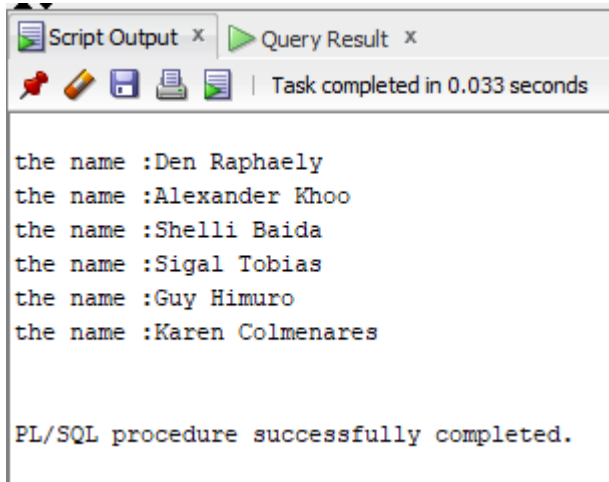
open c_emp;

loop

```

fetch c_emp into v_emp;
exit when c_emp%notfound;
dbms_output.put_line('the name :'||v_emp.first_name||' '||v_emp.last_name);
end loop;
close c_emp;
end;

```



```

Script Output x Query Result x
Task completed in 0.033 seconds

the name :Den Raphaely
the name :Alexander Khoo
the name :Shelli Baida
the name :Sigal Tobias
the name :Guy Himuro
the name :Karen Colmenares

PL/SQL procedure successfully completed.

```

--- for loops in cursor

```

declare
cursor c_emp is select first_name,last_name from emp where department_id=60;

```

```

begin

```

```

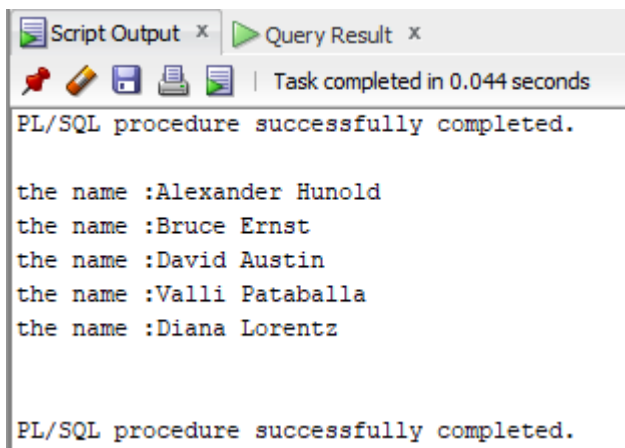
for i in c_emp loop
dbms_output.put_line('the name :'||i.first_name||' '||i.last_name);
end loop;

```

```

end;

```



```

Script Output x Query Result x
Task completed in 0.044 seconds

PL/SQL procedure successfully completed.

the name :Alexander Hunold
the name :Bruce Ernst
the name :David Austin
the name :Valli Pataballa
the name :Diana Lorentz

PL/SQL procedure successfully completed.

```

```

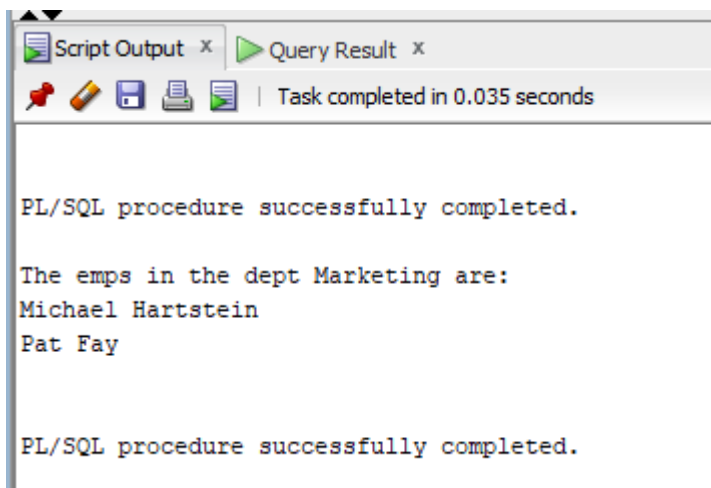
--- cursor with parameters
declare
cursor c_emp (dept_id number) is select first_name,last_name,department_name from emp
join dept using (department_id) where department_id=dept_id;

v_emp c_emp%rowtype;

begin
open c_emp(:dept_id);
fetch c_emp into v_emp;
dbms_output.put_line('The emps in the dept '||v_emp.department_name||' are: ');
close c_emp;

open c_emp(:dept_id);
loop
fetch c_emp into v_emp;
exit when c_emp%notfound;
dbms_output.put_line(v_emp.first_name||' '||v_emp.last_name);
end loop;
close c_emp;
end;

```



```

--- cursor with parameters
declare
cursor c_emp (dept_id number) is select first_name,last_name,department_name from emp
join dept using (department_id) where department_id=dept_id;

v_emp c_emp%rowtype;

begin
open c_emp(:dept_id);

```

```
fetch c_emp into v_emp;
dbms_output.put_line('The emps in the dept '||v_emp.department_name||' are: ');
close c_emp;
```

```
open c_emp(:dept_id2);
loop
fetch c_emp into v_emp;
exit when c_emp%notfound;
dbms_output.put_line(v_emp.first_name||' '||v_emp.last_name);
end loop;
close c_emp;
end;
```

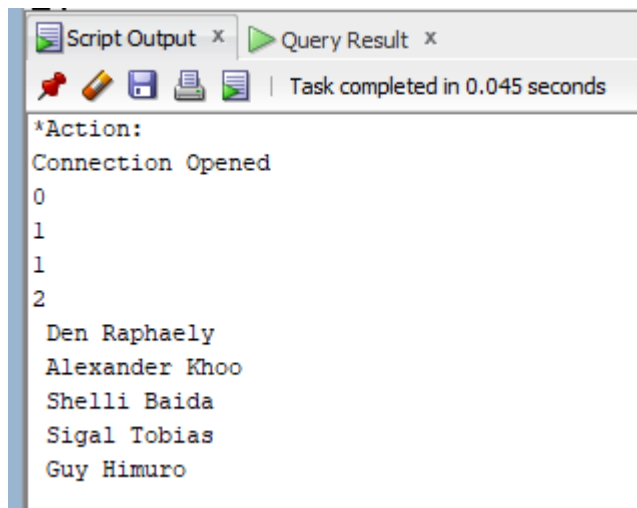
--- cursor attribute types

```
declare
cursor c_emps is select * from emp where department_id=30;
```

```
v_emps c_emps%rowtype;
```

```
begin
if not c_emps%isopen then
open c_emps;
dbms_output.put_line('Connection Opened');
end if;
dbms_output.put_line(c_emps%rowcount);
fetch c_emps into v_emps;
dbms_output.put_line(c_emps%rowcount);
dbms_output.put_line(c_emps%rowcount);
fetch c_emps into v_emps;
dbms_output.put_line(c_emps%rowcount);
close c_emps;
```

```
open c_emps;
loop
fetch c_emps into v_emps;
exit when c_emps%notfound or c_emps%rowcount>5;
dbms_output.put_line(' '||v_emps.first_name||' '||v_emps.last_name);
end loop;
close c_emps;
end;
```

--- Reference cursor

```
declare
type t_emps is ref cursor return emp%rowtype;

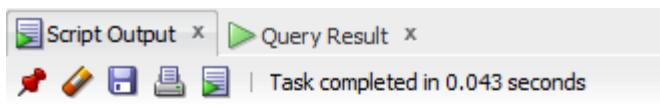
rc_emps t_emps;

r_emps emp%rowtype;
begin

open rc_emps for select * from retired_emp;

loop
fetch rc_emps into r_emps;
exit when rc_emps%notfound;
dbms_output.put_line(r_emps.first_name||' '||r_emps.last_name);
end loop;
close rc_emps;
dbms_output.put_line('-----');
open rc_emps for select * from emp;

loop
fetch rc_emps into r_emps;
exit when rc_emps%notfound;
dbms_output.put_line(r_emps.first_name||' '||r_emps.last_name);
end loop;
close rc_emps;
end;
```



PL/SQL procedure successfully completed.

Bruce Ernst

Steven King

Neena Kochhar

Lex De Haan

Alexander Hunold

Bruce Ernst

David Austin

--- exceptions in plsql

declare

fname varchar(30);

begin

select first_name into fname from emp where employee_id=40;

dbms_output.put_line('Updated');

exception

when no_data_found then

dbms_output.put_line('There is No Id in the table');

end;

There is No Id in the table

PL/SQL procedure successfully completed.

--- Non Predefined Exceptions

begin

update emp set email = null where employee_id=100;

pragma exception_init(cannot_update_to_null,-01407);

exception

when cannot_update_to_null then

dbms_output.put_line('You cannot update the value');

end;

```

pragma exception_init(cannot_update_to_null,-01407);
exception
when cannot_update_to_null then
dbms_output.put_line('You cannot update the value');
end;
Error report -
ORA-06550: line 3, column 23:
PLS-00109: unknown exception name 'CANNOT_UPDATE_TO_NULL' in PRAGMA EXCEPTION_INIT
06550. 00000 - "line %s, column %s:\n%s"
*Cause:      Usually a PL/SQL compilation error.
*Action:

```

---- user defined exception

declare

too_high_salary exception;

v_sal pls_integer;

begin

select salary into v_sal from emp where employee_id=100;

if v_sal>20000 then

raise too_high_salary;

end if;

dbms_output.put_line('The Salary is Acceptable Range');

exception

when too_high_salary then

dbms_output.put_line('The Salary is Too High');

end;

```

The Salary is Too High

```

```

PL/SQL procedure successfully completed.

```

---- user defined exception

declare

too_high_salary exception;

v_sal pls_integer;

begin

select salary into v_sal from emp where employee_id=100;

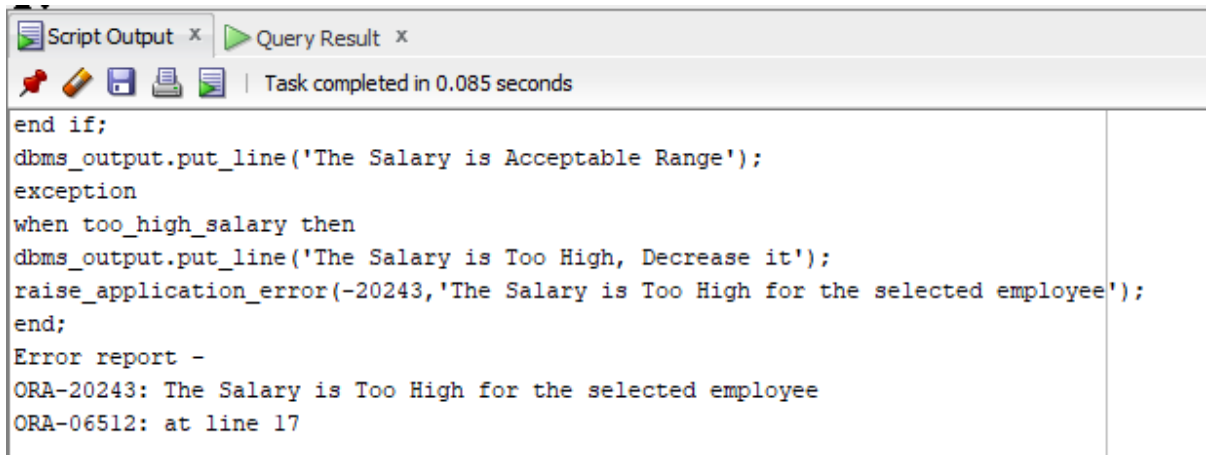
if v_sal>20000 then

raise too_high_salary;

```

end if;
dbms_output.put_line('The Salary is Acceptable Range');
exception
when too_high_salary then
dbms_output.put_line('The Salary is Too High, Decrease it');
raise_application_error(-20243,'The Salary is Too High for the selected employee');
end;

```



The screenshot shows a window titled 'Script Output' and 'Query Result'. It indicates 'Task completed in 0.085 seconds'. The script content is the same as above. The output shows an error report: 'ORA-20243: The Salary is Too High for the selected employee' and 'ORA-06512: at line 17'.

```

end if;
dbms_output.put_line('The Salary is Acceptable Range');
exception
when too_high_salary then
dbms_output.put_line('The Salary is Too High, Decrease it');
raise_application_error(-20243,'The Salary is Too High for the selected employee');
end;
Error report -
ORA-20243: The Salary is Too High for the selected employee
ORA-06512: at line 17

```

STORED PROCEDURES in PLSQL:

--- stored Procedures

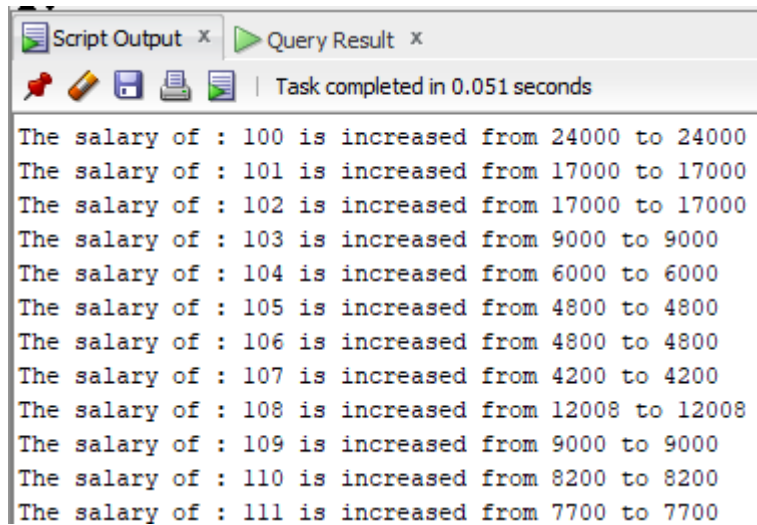
```

create procedure inc_sal as
  cursor c_emps is select * from emp for update;
  v_salary_increase pls_integer:= 1.10;
  v_old_salary pls_integer;
begin
  for r_emp in c_emps loop
    v_old_salary := r_emp.salary;
    r_emp.salary := r_emp.salary*v_salary_increase + r_emp.salary *
nvl(r_emp.commission_pct,0);
    update emp set row = r_emp where current of c_emps;
    dbms_output.put_line('The salary of : '|| r_emp.employee_id
                        || ' is increased from '||v_old_salary||' to '||r_emp.salary);
  end loop;
end;

```

Procedure INC_SAL compiled

execute inc_sal;

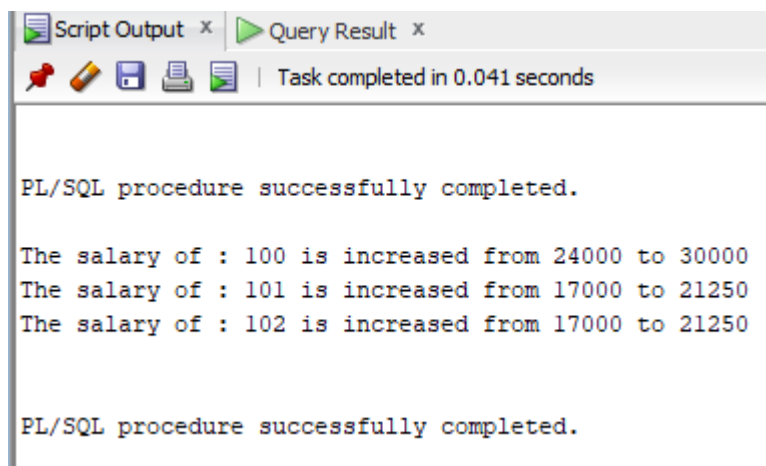


```
Script Output x Query Result x
Task completed in 0.051 seconds

The salary of : 100 is increased from 24000 to 24000
The salary of : 101 is increased from 17000 to 17000
The salary of : 102 is increased from 17000 to 17000
The salary of : 103 is increased from 9000 to 9000
The salary of : 104 is increased from 6000 to 6000
The salary of : 105 is increased from 4800 to 4800
The salary of : 106 is increased from 4800 to 4800
The salary of : 107 is increased from 4200 to 4200
The salary of : 108 is increased from 12008 to 12008
The salary of : 109 is increased from 9000 to 9000
The salary of : 110 is increased from 8200 to 8200
The salary of : 111 is increased from 7700 to 7700
```

```
create procedure incsal(v_salary_increase in number,v_department_id pls_integer) as
  cursor c_emps is select * from emp where department_id=v_department_id for update;
  ---v_salary_increase pls_integer:= 1.10;
  v_old_salary pls_integer;
begin
  for r_emp in c_emps loop
    v_old_salary := r_emp.salary;
    r_emp.salary := r_emp.salary*v_salary_increase + r_emp.salary *
nvl(r_emp.commission_pct,0);
    update emp set row = r_emp where current of c_emps;
    dbms_output.put_line('The salary of : '|| r_emp.employee_id
                        || ' is increased from '||v_old_salary||' to '||r_emp.salary);
  end loop;
end;
--- execute the procedure

begin
incsal(1.25,90);
end;
```



```
Script Output x Query Result x
Task completed in 0.041 seconds

PL/SQL procedure successfully completed.

The salary of : 100 is increased from 24000 to 30000
The salary of : 101 is increased from 17000 to 21250
The salary of : 102 is increased from 17000 to 21250

PL/SQL procedure successfully completed.
```

```

create procedure inc_sal_affected(v_salary_increase in number,v_department_id
pls_integer,v_affected_emp out number) as
  cursor c_emps is select * from emp where department_id=v_department_id for update;
  ---v_salary_increase pls_integer:= 1.10;
  v_old_salary pls_integer;
  v_sal_inc number:=0;
begin
  v_affected_emp:=0;
  for r_emp in c_emps loop
    v_old_salary := r_emp.salary;
    r_emp.salary := r_emp.salary*v_salary_increase + r_emp.salary *
nvl(r_emp.commission_pct,0);
    update emp set row = r_emp where current of c_emps;
    dbms_output.put_line('The salary of : '|| r_emp.employee_id
                        || ' is increased from '||v_old_salary||' to '||r_emp.salary);
    v_affected_emp:=v_affected_emp+1;
    v_sal_inc:=v_sal_inc+v_salary_increase+nvl(r_emp.commission_pct,0);
  end loop;
  v_salary_increase:=v_sal_inc/v_affected_emp;
end;

```

```

create procedure add_jobs(job_id varchar2,job_title varchar2,min_sal number default
1000,max_sal number default null) is

```

```

begin

insert into jobs values(job_id,job_title,min_sal,max_sal);
print('The Job :'||job_title||' is inserted..');

End;

```

--- functions in PLSQL

```

create function getsal(dept_id department_id%type) return number as
avg_sal number;
begin
return
select avg(salary) into avg_sal from emp where department_id=dept_id;
return avg_sal;

end getsal;

```

