PL/SQL

PL-SQL (Procedural - SQL)

$$\begin{tikzcd}
 & & \text{PL-SQL} \arrow[dll] \arrow[dl] \arrow[d] \arrow[dr] & \\
\text{Function} & \text{* Procedure} & \text{Trigger} & \text{Cursor}
\end{tikzcd}$$

Block.
Code.

Declaration	a int

Executable Code	begin
	end

Exception handling (error)	

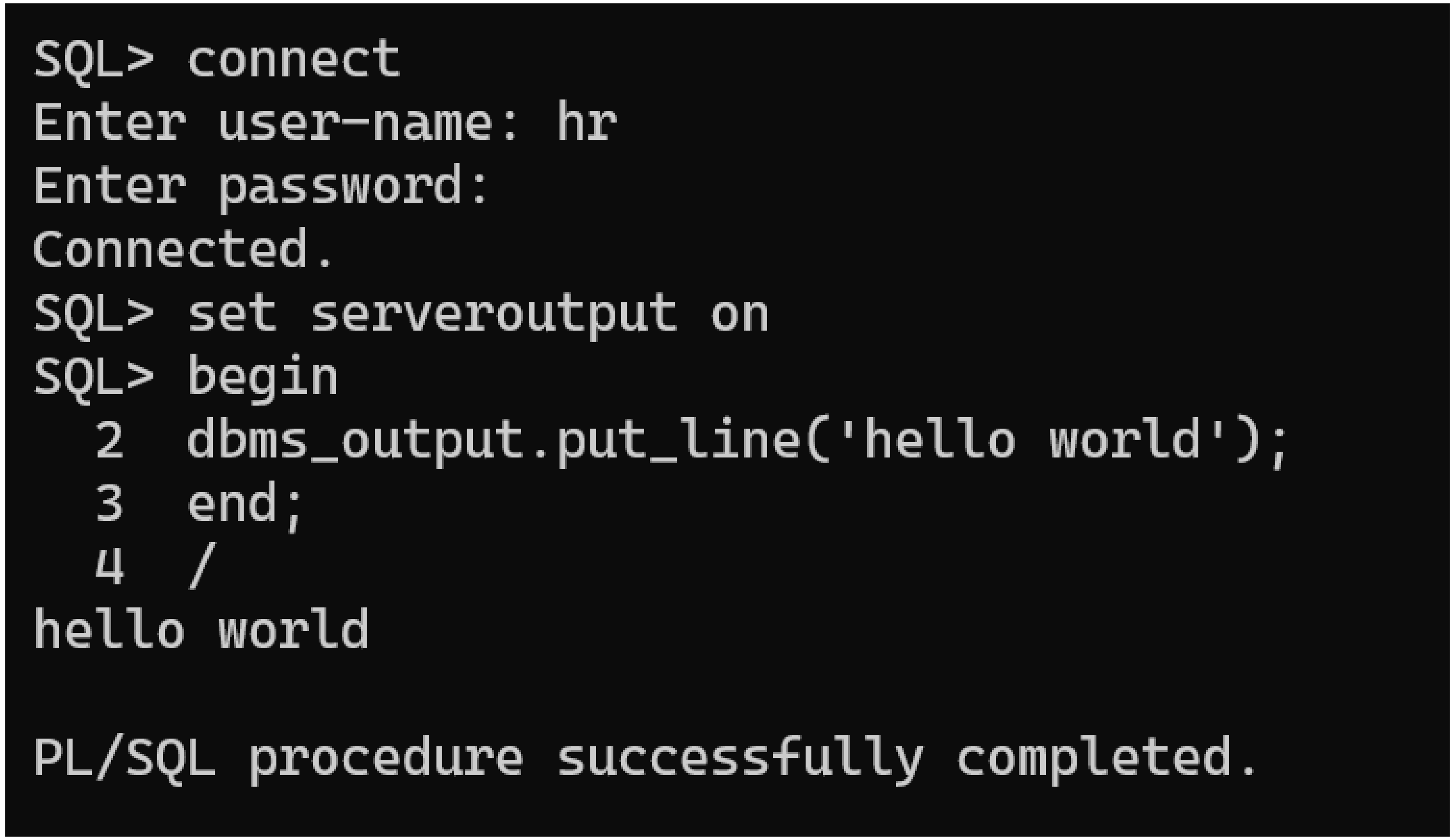
=

SQL (write a query)
↓
Declarative.
'What to do'
How to do ✗

{
}

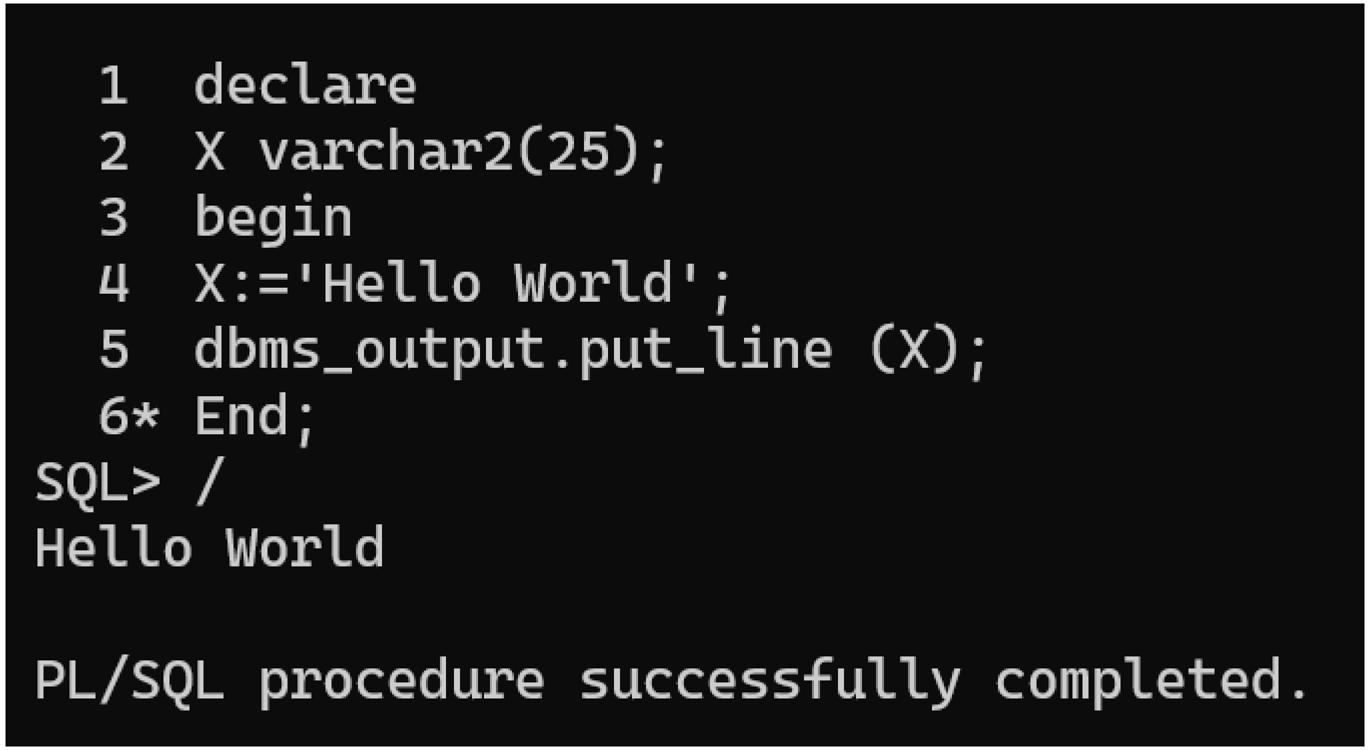
```
declare
a int
b int
c int
begin
a:=10;
b:=20
c:=a+b;
end;
```

write a simple program for printing "Hello World"

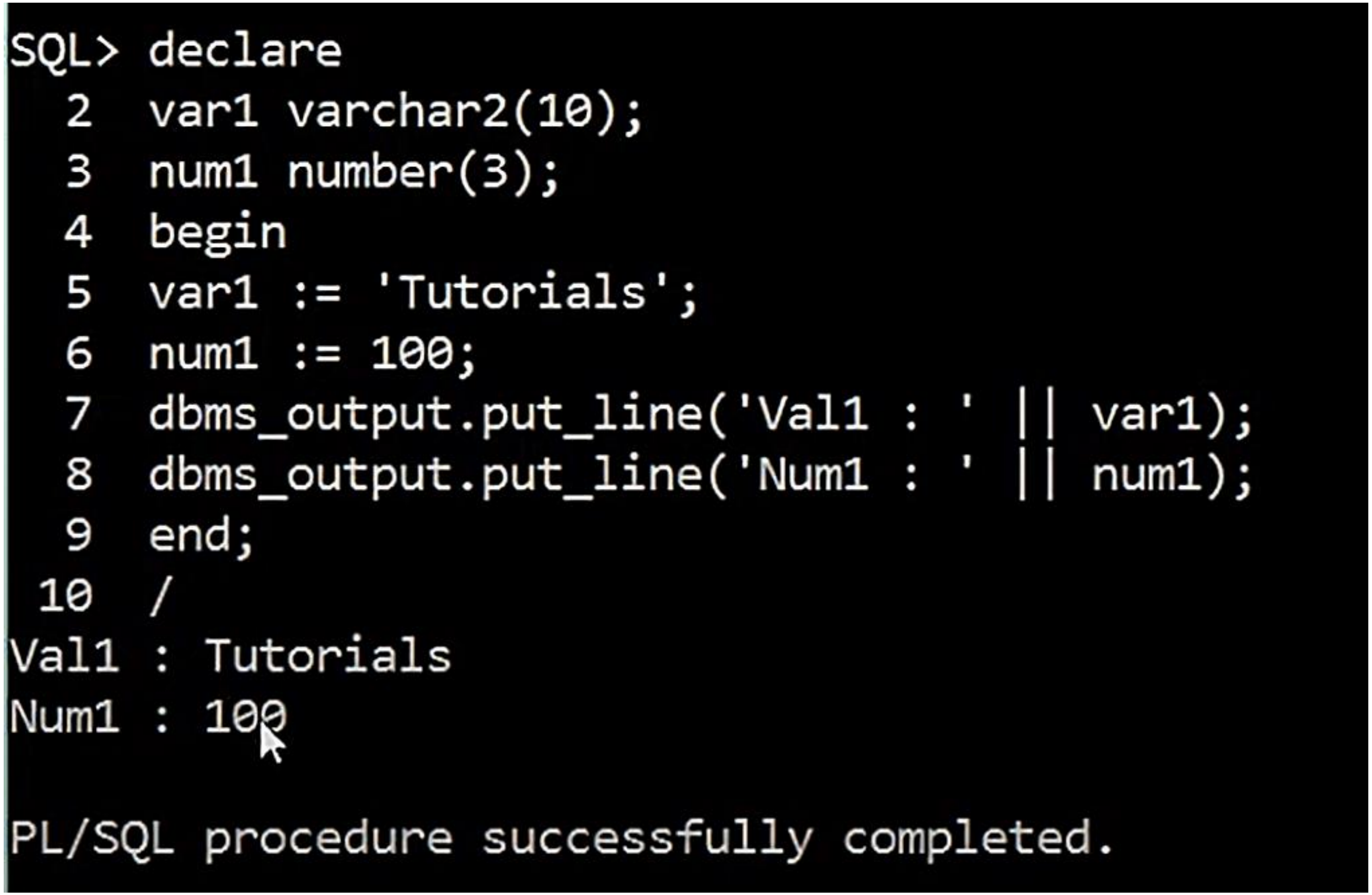


```
SQL> connect
Enter user-name: hr
Enter password:
Connected.
SQL> set serveroutput on
SQL> begin
  2  dbms_output.put_line('hello world');
  3  end;
  4  /
hello world

PL/SQL procedure successfully completed.
```

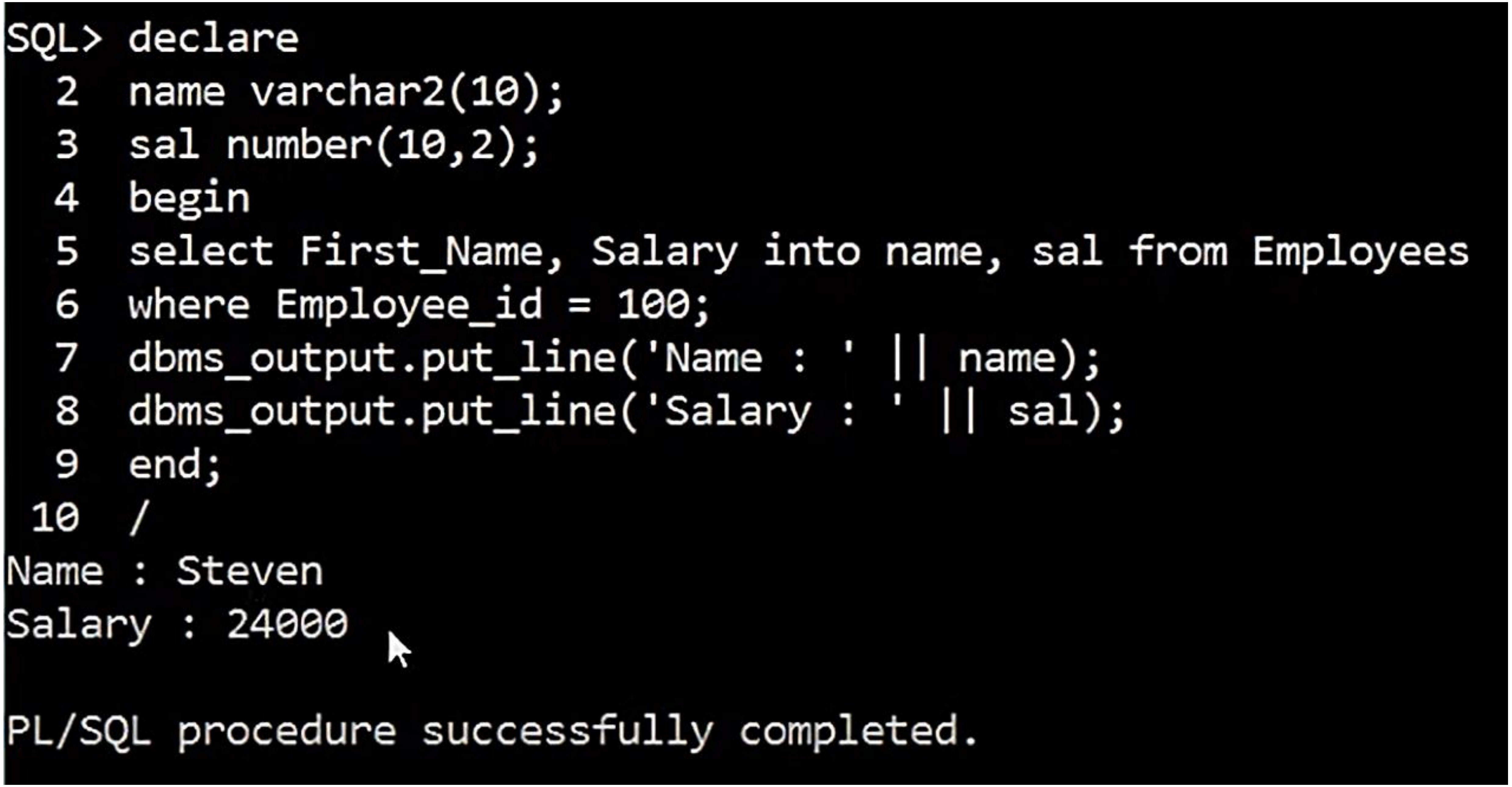
Declaring and usage of variables in the program



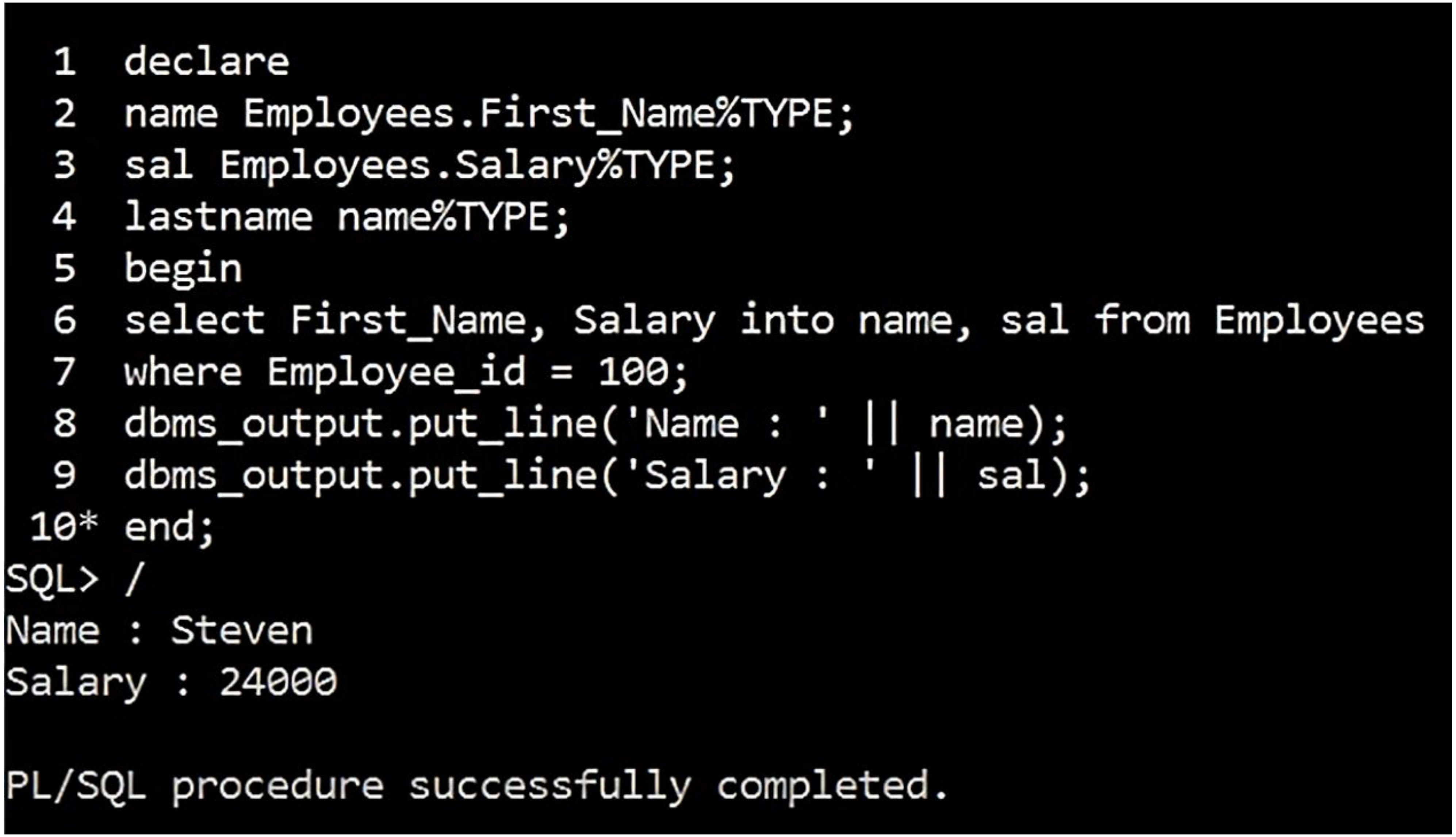
```
declare
X varchar2(25);
begin
X:='Hello World';
dbms_output.put_line (X);
End;
SQL> /
Hello World

PL/SQL procedure successfully completed.
```



```
SQL> declare
  2  var1 varchar2(10);
  3  num1 number(3);
  4  begin
  5  var1 := 'Tutorials';
  6  num1 := 100;
  7  dbms_output.put_line('Val1 : ' || var1);
  8  dbms_output.put_line('Num1 : ' || num1);
  9  end;
 10  /
Val1 : Tutorials
Num1 : 100

PL/SQL procedure successfully completed.
```

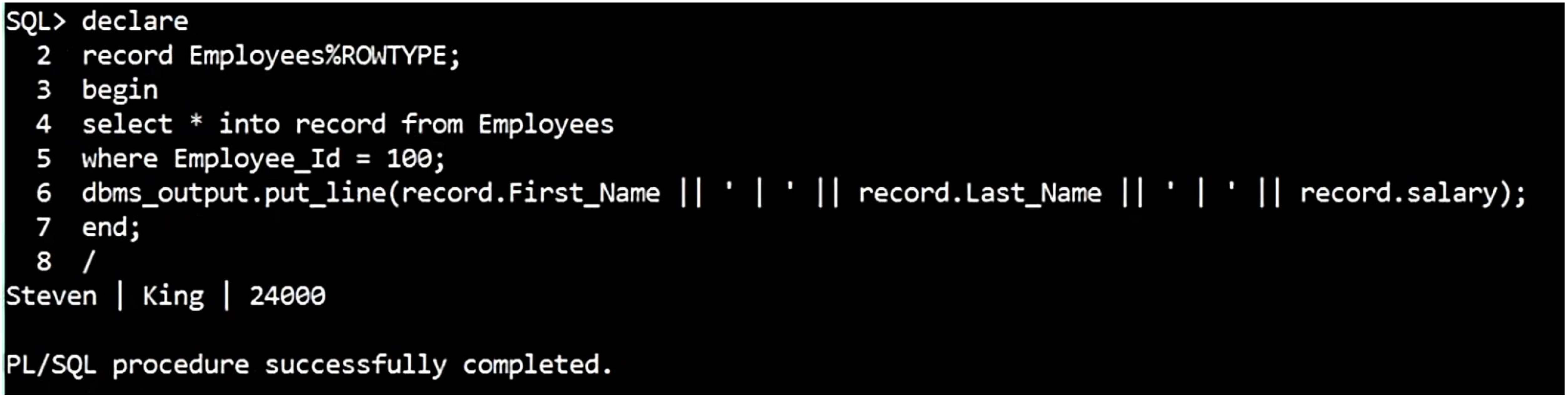


```
SQL> declare
     name varchar2(10);
     sal number(10,2);
     begin
     select First_Name, Salary into name, sal from Employees
     where Employee_id = 100;
     dbms_output.put_line('Name : ' || name);
     dbms_output.put_line('Salary : ' || sal);
     end;
     /
Name : Steven
Salary : 24000

PL/SQL procedure successfully completed.
```



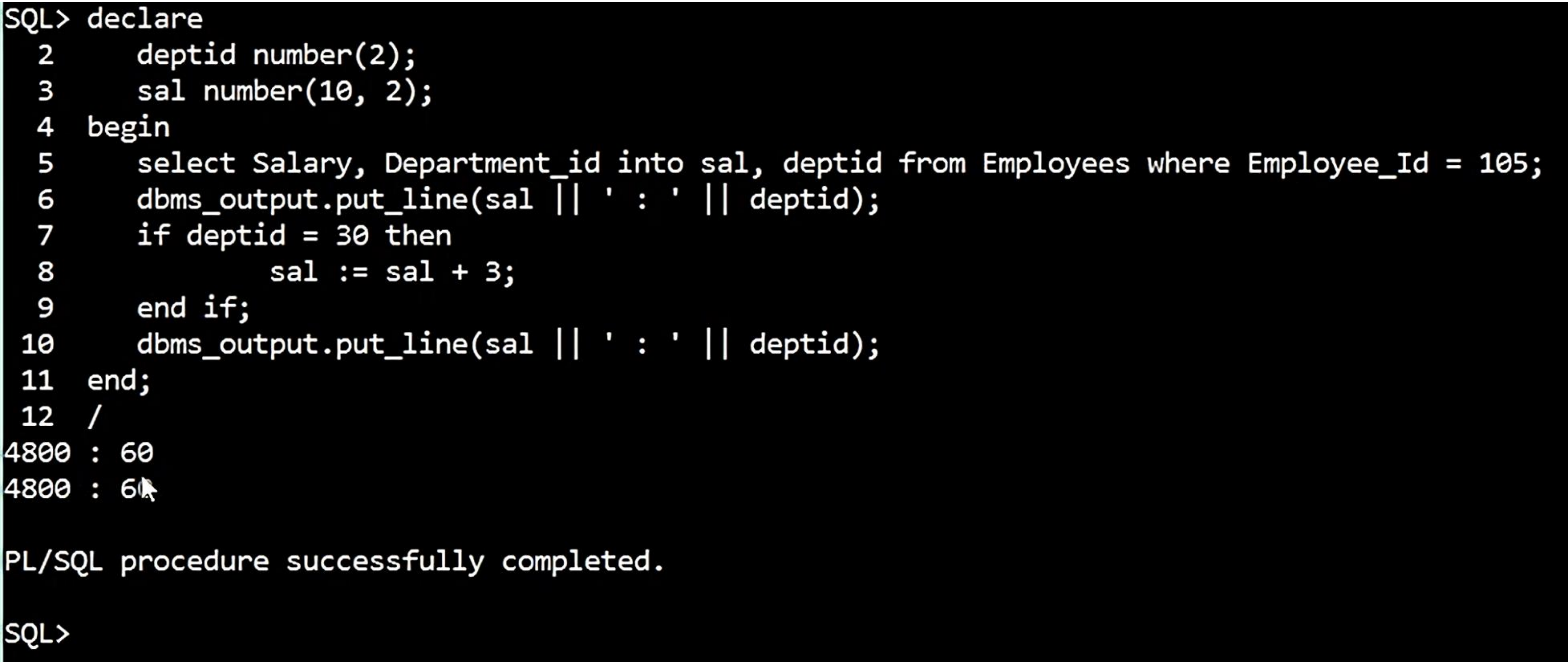
```
declare
name Employees.First_Name%TYPE;
sal Employees.Salary%TYPE;
lastname name%TYPE;
begin
select First_Name, Salary into name, sal from Employees
where Employee_id = 100;
dbms_output.put_line('Name : ' || name);
dbms_output.put_line('Salary : ' || sal);
end;
SQL> /
Name : Steven
Salary : 24000

PL/SQL procedure successfully completed.
```

If we don’t know the size of the column as exists in the table, employees then we can use the % Type for the declaration of the data type of the attributes in the PL/ SQL block.

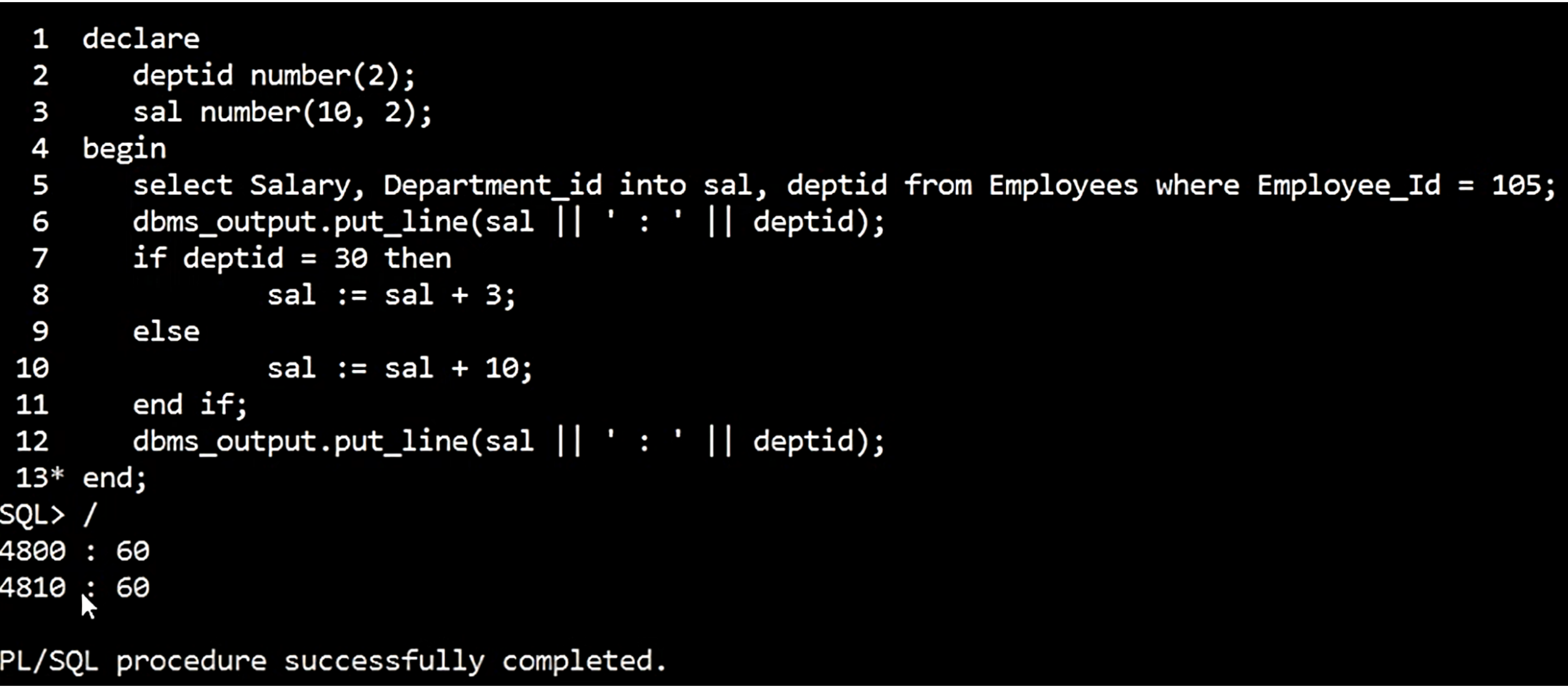


```
SQL> declare
  2  record Employees%ROWTYPE;
  3  begin
  4  select * into record from Employees
  5  where Employee_Id = 100;
  6  dbms_output.put_line(record.First_Name || ' | ' || record.Last_Name || ' | ' || record.salary);
  7  end;
  8  /
Steven | King | 24000

PL/SQL procedure successfully completed.
```

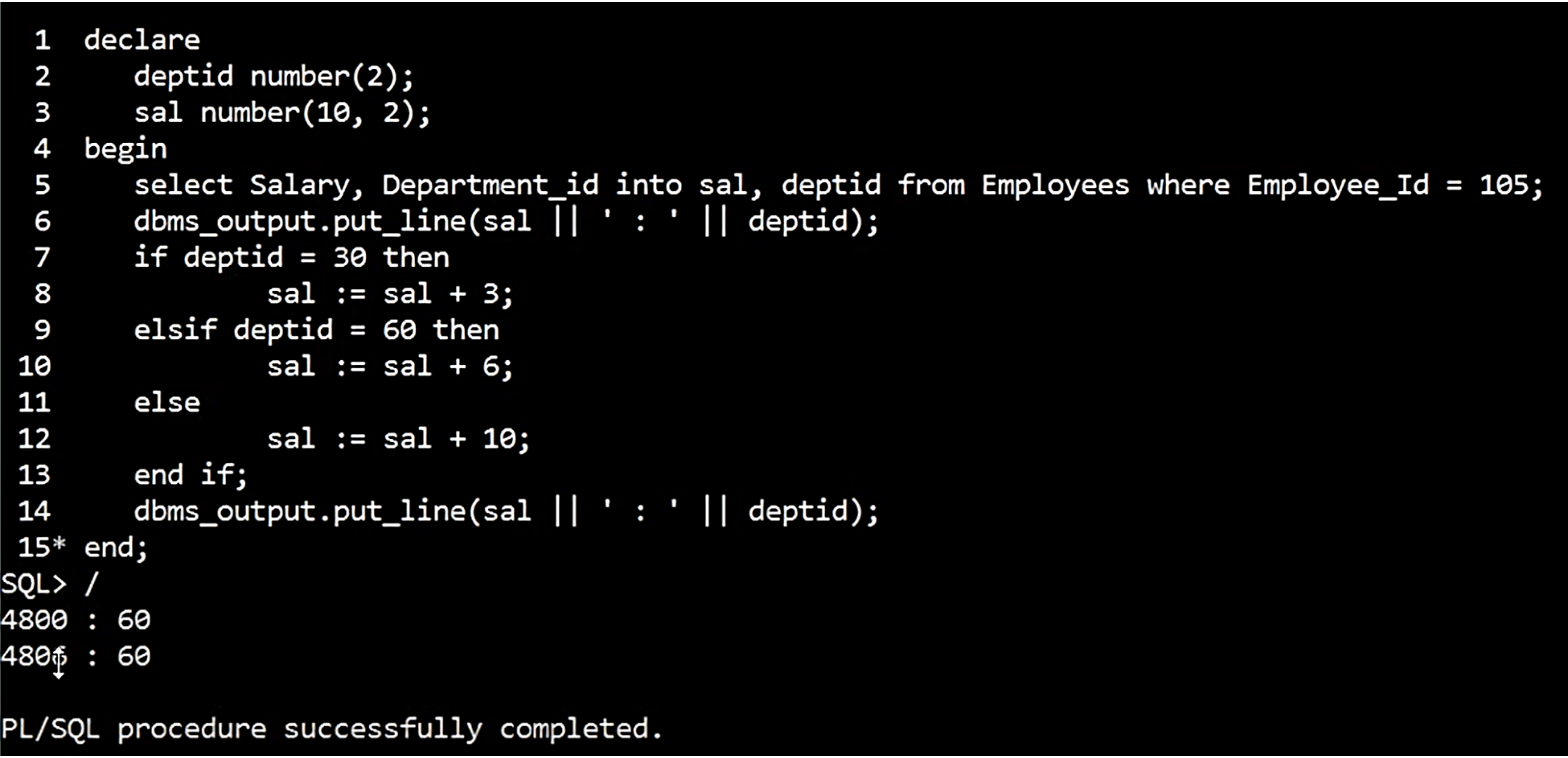
PL/SQL - Conditional Statements



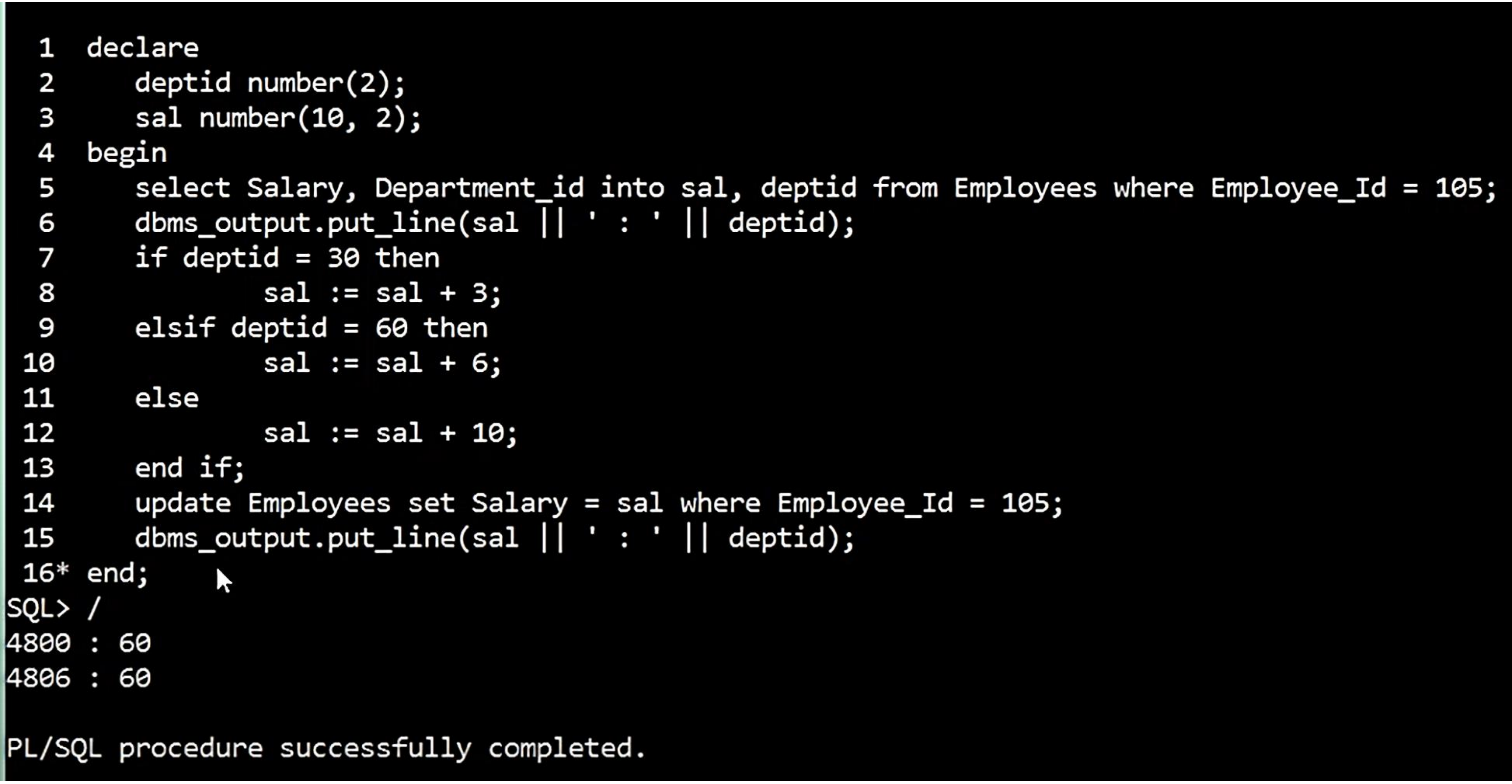
```
SQL> declare
         deptid number(2);
         sal number(10, 2);
     begin
         select Salary, Department_id into sal, deptid from Employees where Employee_Id = 105;
         dbms_output.put_line(sal || ' : ' || deptid);
         if deptid = 30 then
                 sal := sal + 3;
         end if;
         dbms_output.put_line(sal || ' : ' || deptid);
     end;
     /
4800 : 60
4800 : 60

PL/SQL procedure successfully completed.

SQL>
```



```
declare
   deptid number(2);
   sal number(10, 2);
begin
   select Salary, Department_id into sal, deptid from Employees where Employee_Id = 105;
   dbms_output.put_line(sal || ' : ' || deptid);
   if deptid = 30 then
           sal := sal + 3;
   else
           sal := sal + 10;
   end if;
   dbms_output.put_line(sal || ' : ' || deptid);
end;
SQL> /
4800 : 60
4810 : 60

PL/SQL procedure successfully completed.
```

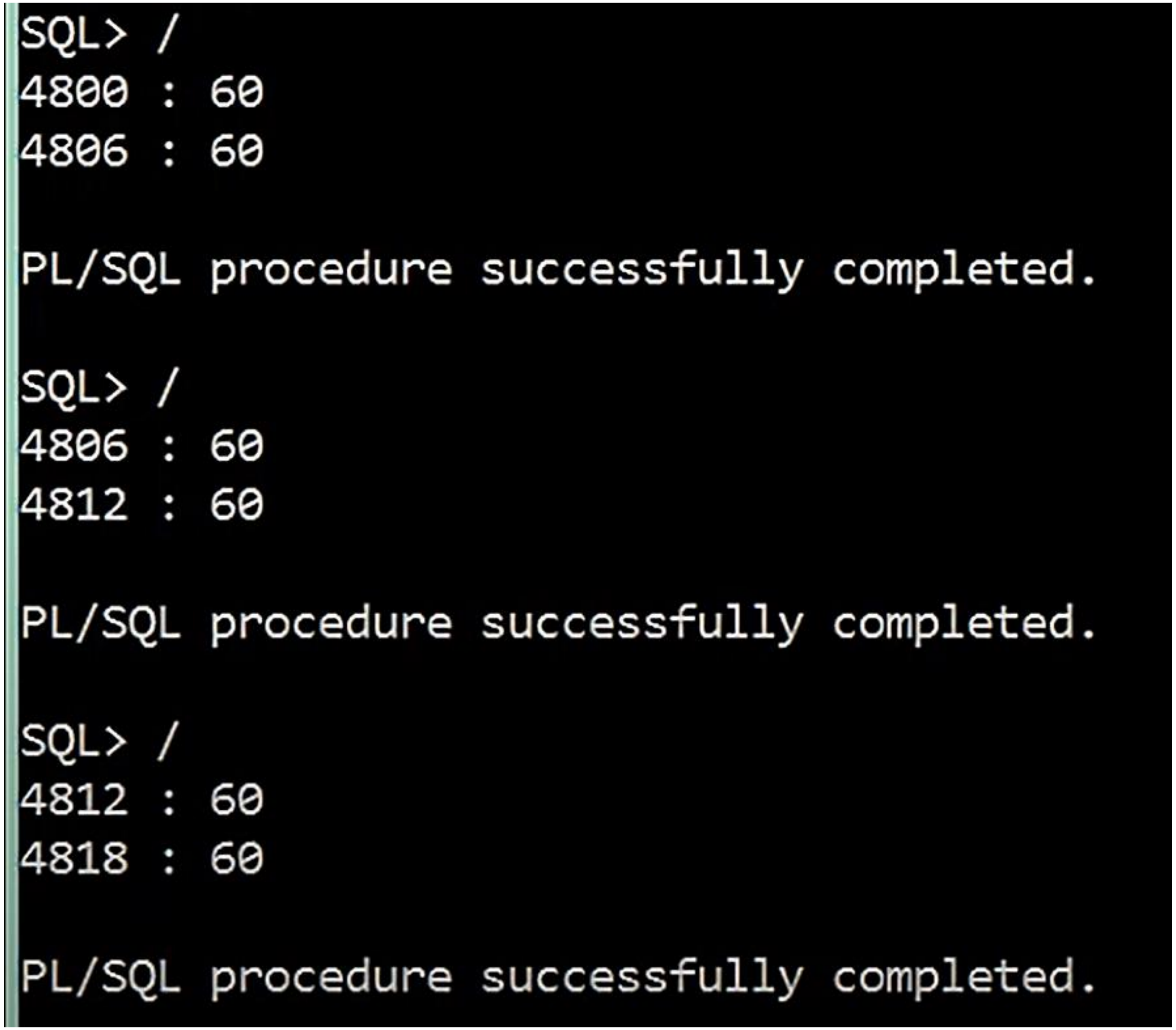


```
declare
   deptid number(2);
   sal number(10, 2);
begin
   select Salary, Department_id into sal, deptid from Employees where Employee_Id = 105;
   dbms_output.put_line(sal || ' : ' || deptid);
   if deptid = 30 then
           sal := sal + 3;
   elsif deptid = 60 then
           sal := sal + 6;
   else
           sal := sal + 10;
   end if;
   dbms_output.put_line(sal || ' : ' || deptid);
end;
SQL> /
4800 : 60
4806 : 60

PL/SQL procedure successfully completed.
```



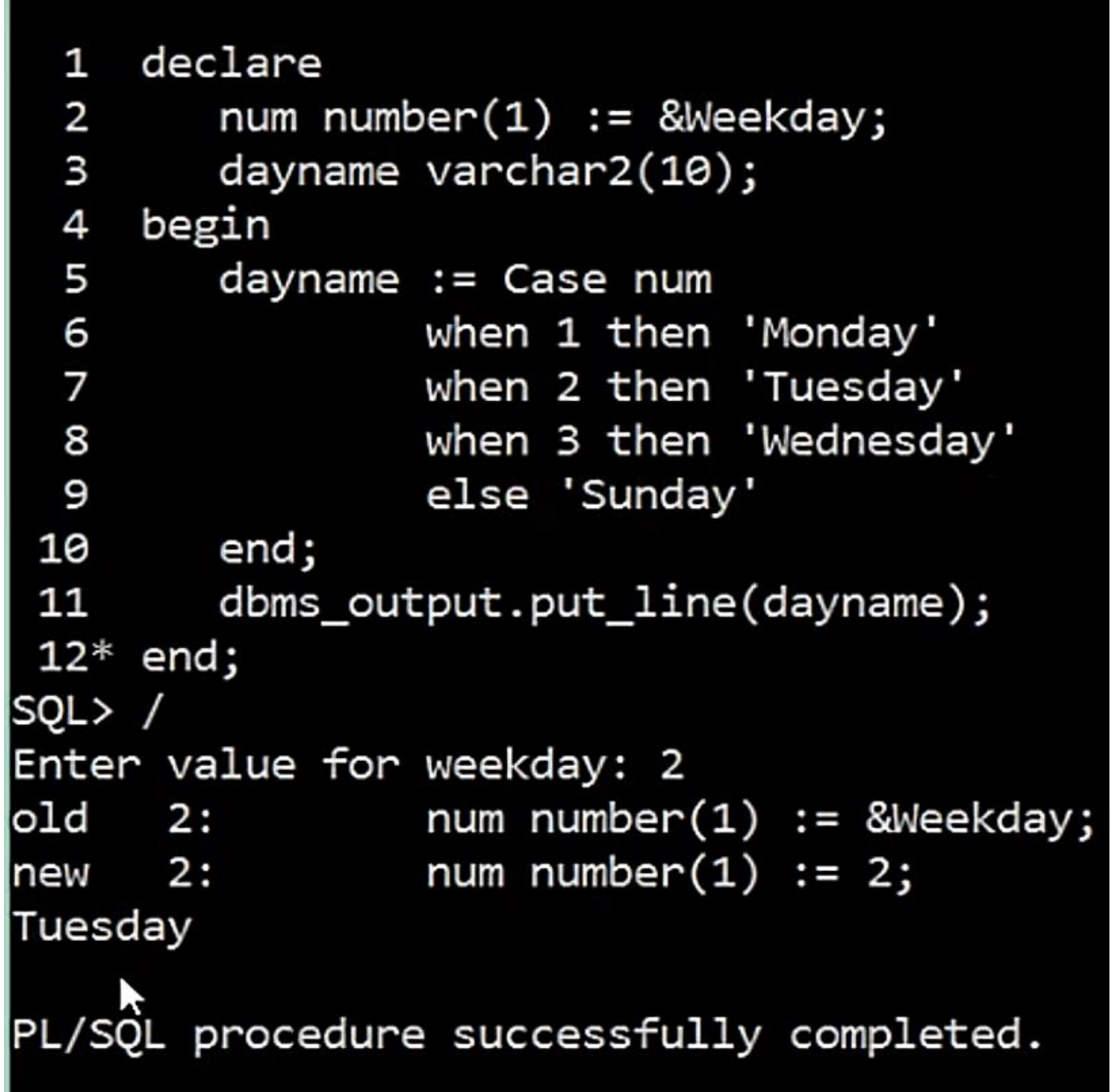
```
declare
   deptid number(2);
   sal number(10, 2);
begin
   select Salary, Department_id into sal, deptid from Employees where Employee_Id = 105;
   dbms_output.put_line(sal || ' : ' || deptid);
   if deptid = 30 then
           sal := sal + 3;
   elsif deptid = 60 then
           sal := sal + 6;
   else
           sal := sal + 10;
   end if;
   update Employees set Salary = sal where Employee_Id = 105;
   dbms_output.put_line(sal || ' : ' || deptid);
end;
SQL> /
4800 : 60
4806 : 60

PL/SQL procedure successfully completed.
```

Discussion on the results of the previous Code



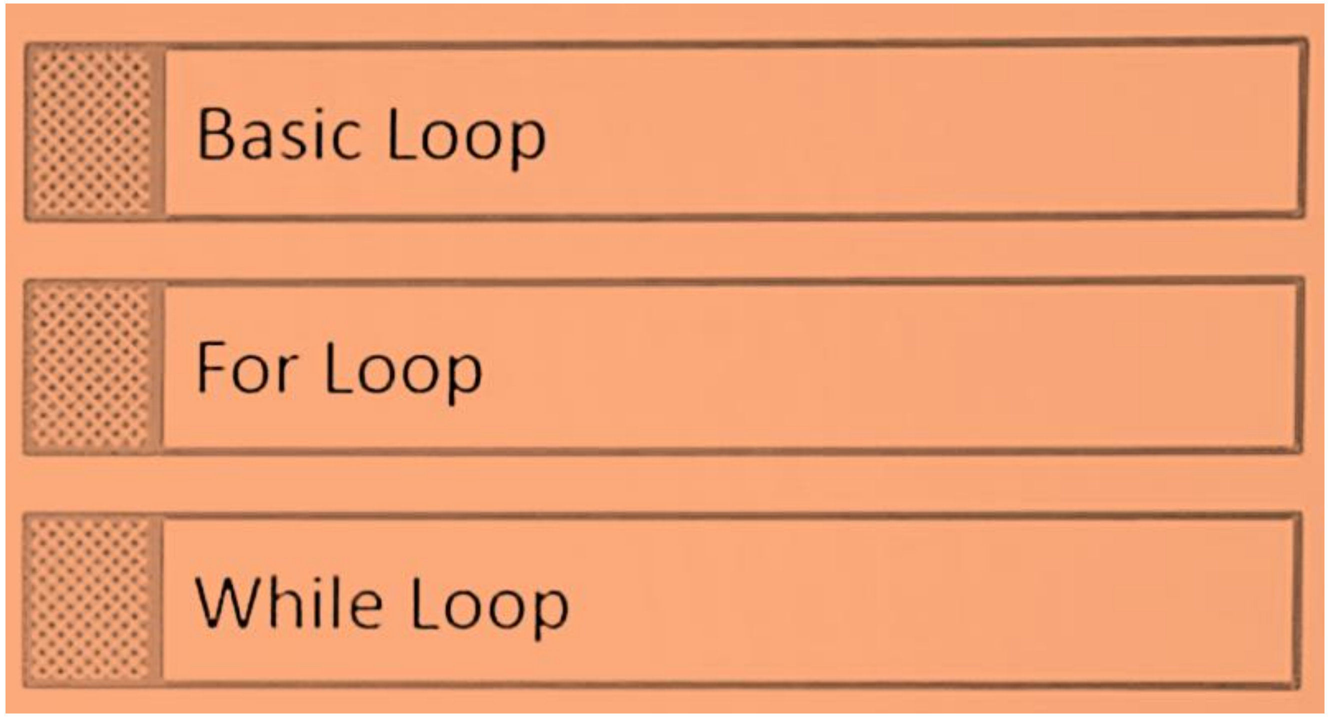
```
SQL> /
4800 : 60
4806 : 60

PL/SQL procedure successfully completed.

SQL> /
4806 : 60
4812 : 60

PL/SQL procedure successfully completed.

SQL> /
4812 : 60
4818 : 60

PL/SQL procedure successfully completed.
```

Case statement



```
declare
    num number(1) := &Weekday;
    dayname varchar2(10);
begin
    dayname := Case num
                when 1 then 'Monday'
                when 2 then 'Tuesday'
                when 3 then 'Wednesday'
                else 'Sunday'
    end;
    dbms_output.put_line(dayname);
end;
SQL> /
Enter value for weekday: 2
old   2:         num number(1) := &Weekday;
new   2:         num number(1) := 2;
Tuesday


PL/SQL procedure successfully completed.
```

PL/SQL - Loops

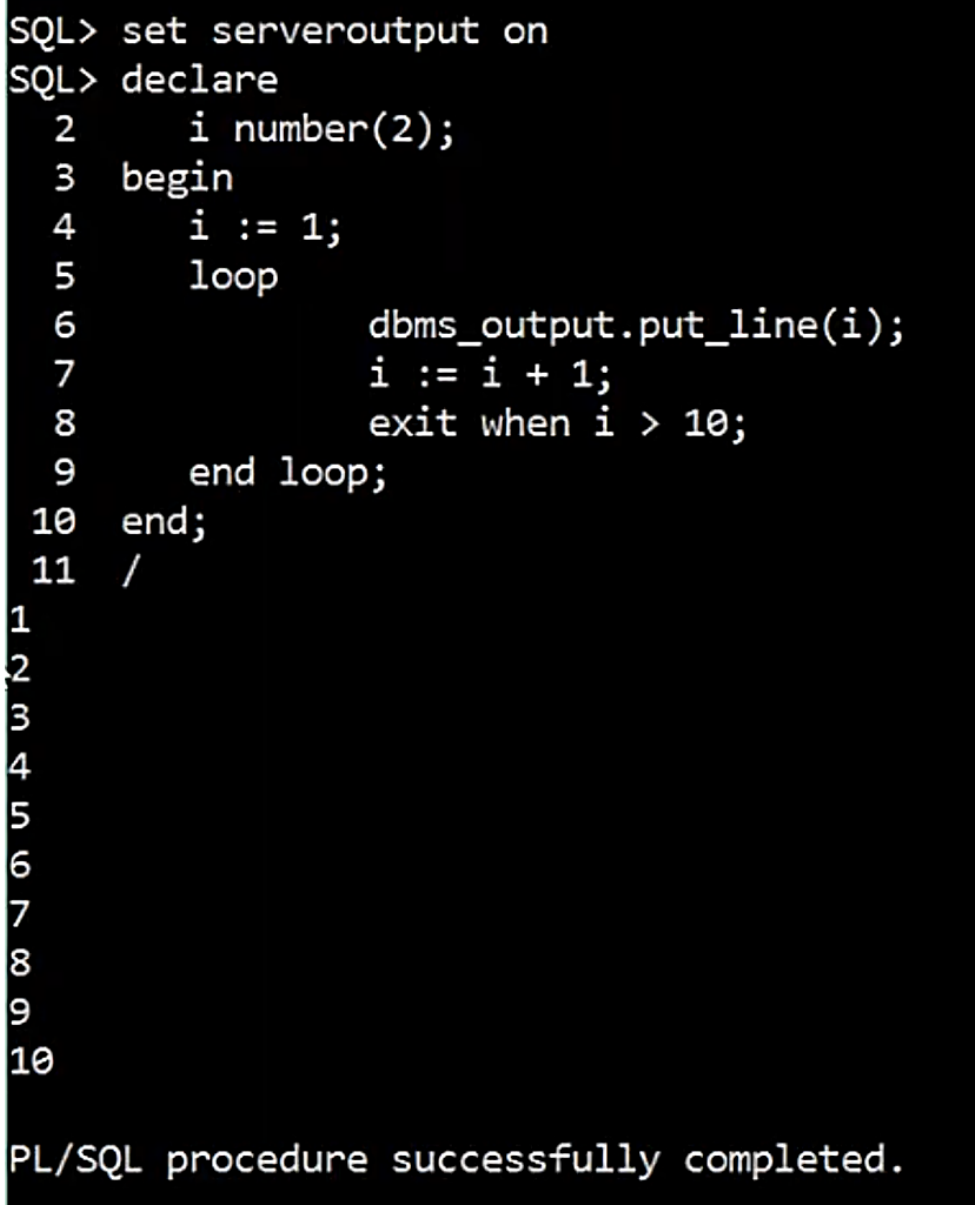


Basic Loop

For Loop

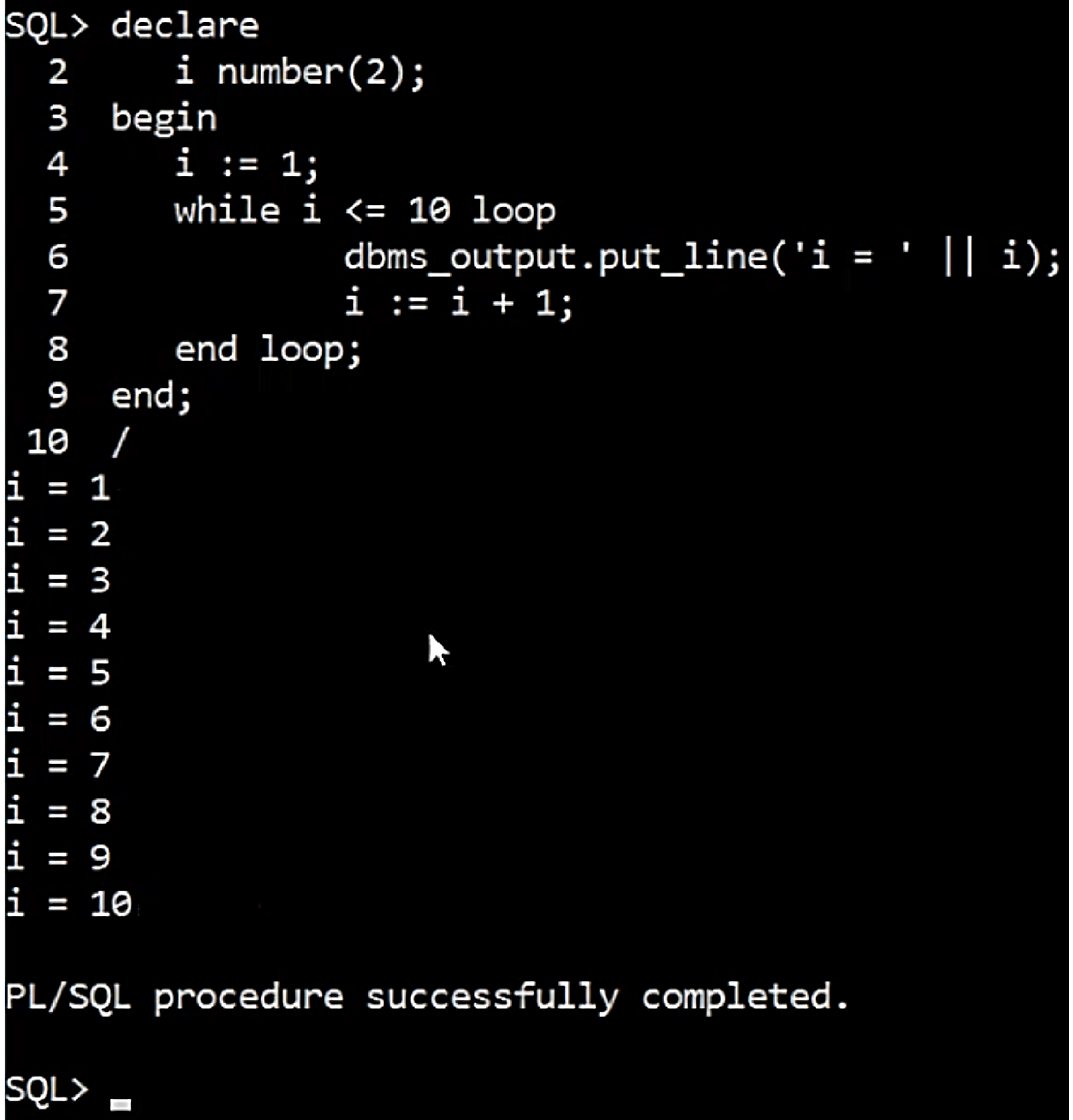
While Loop

Basic Loop



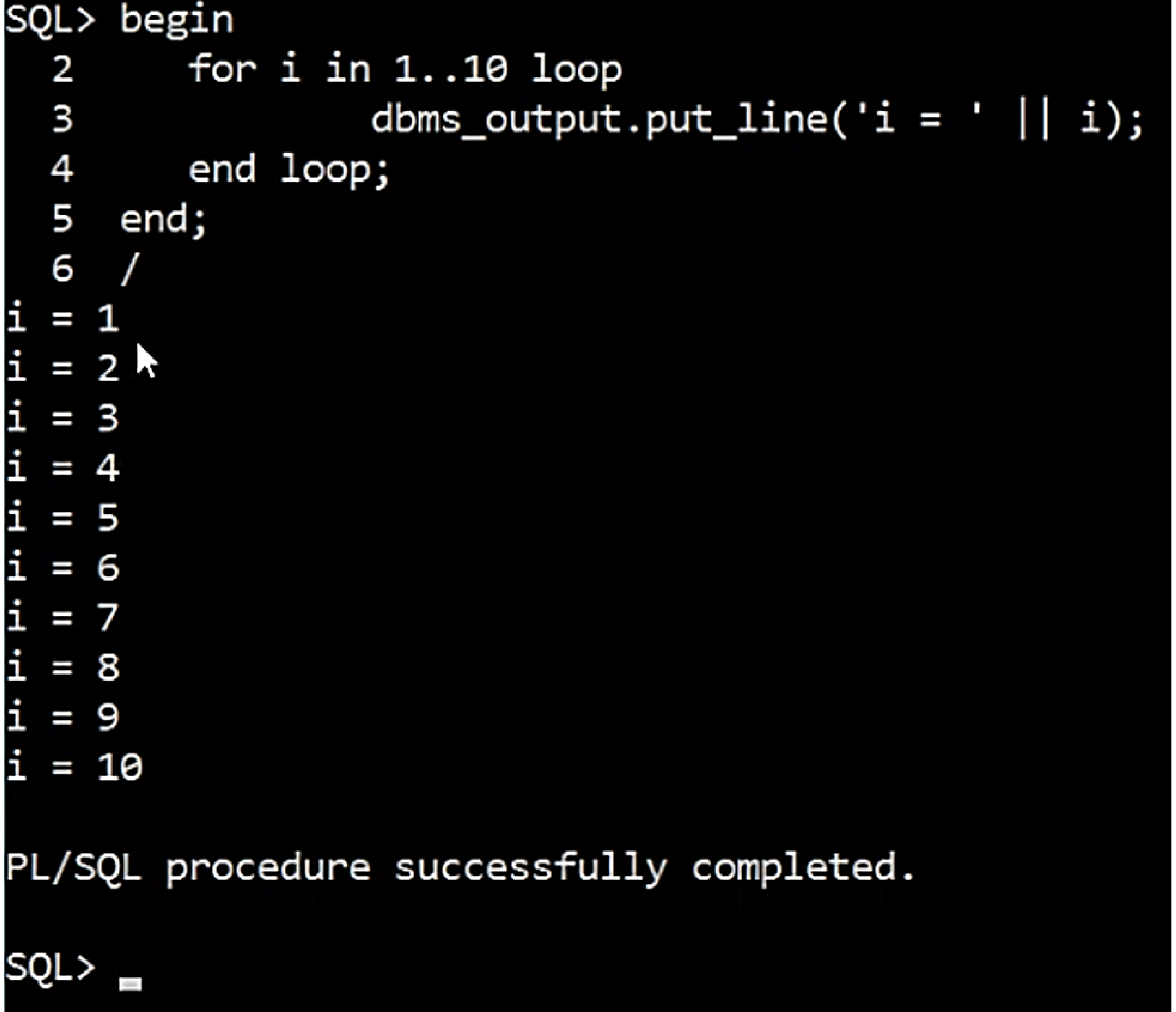
```
SQL> set serveroutput on
SQL> declare
  2      i number(2);
  3  begin
  4      i := 1;
  5      loop
  6              dbms_output.put_line(i);
  7              i := i + 1;
  8              exit when i > 10;
  9      end loop;
 10  end;
 11  /
1
2
3
4
5
6
7
8
9
10

PL/SQL procedure successfully completed.
```

While Loop



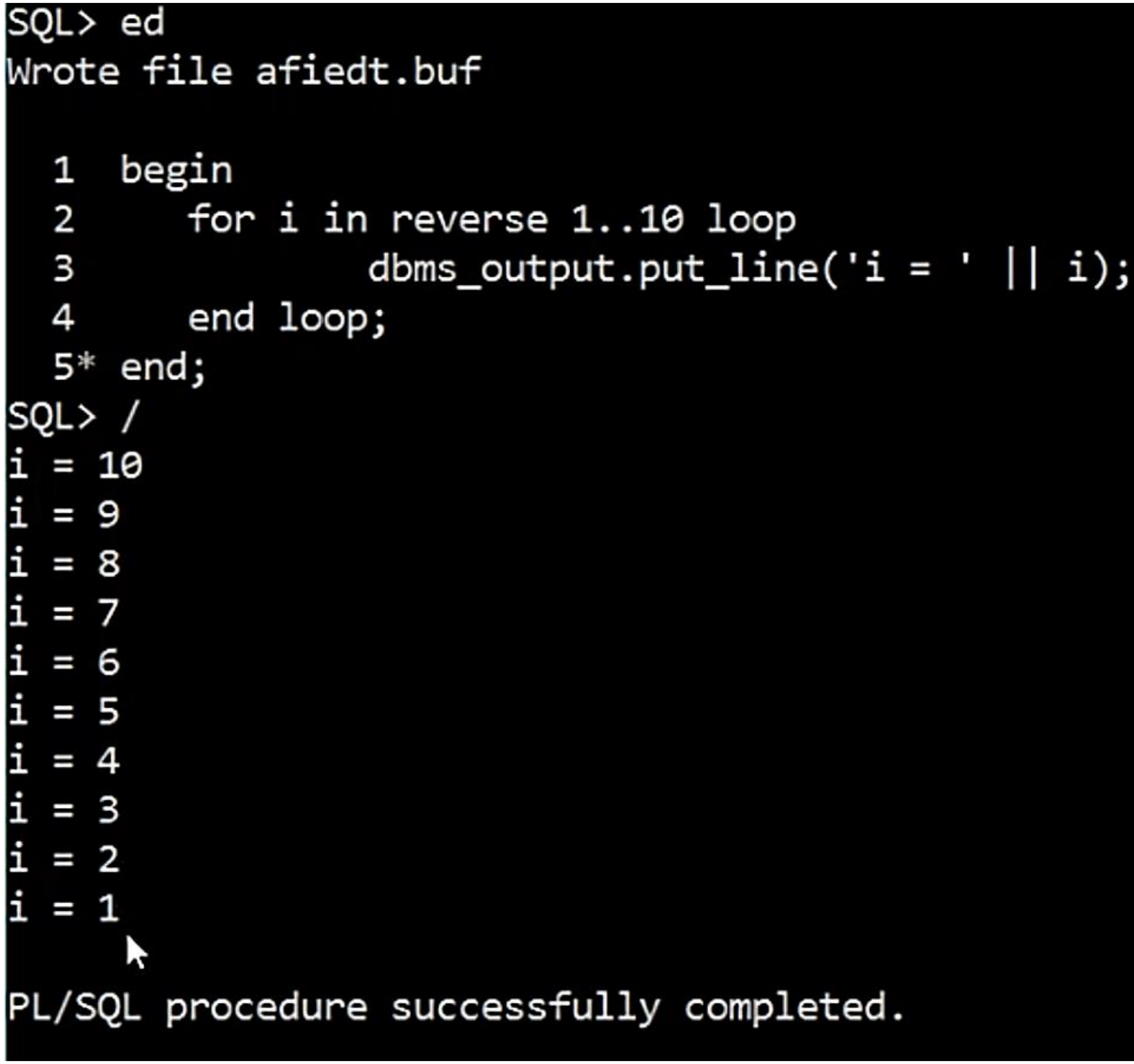
```
SQL> declare
        i number(2);
     begin
        i := 1;
        while i <= 10 loop
                dbms_output.put_line('i = ' || i);
                i := i + 1;
        end loop;
     end;
     /
i = 1
i = 2
i = 3
i = 4
i = 5
i = 6
i = 7
i = 8
i = 9
i = 10

PL/SQL procedure successfully completed.

SQL> _
```

For Loop



```
SQL> begin
  2      for i in 1..10 loop
  3              dbms_output.put_line('i = ' || i);
  4      end loop;
  5  end;
  6  /
i = 1
i = 2
i = 3
i = 4
i = 5
i = 6
i = 7
i = 8
i = 9
i = 10

PL/SQL procedure successfully completed.

SQL> _
```

Print the values in reverse order by putting the reverse keyword



```
SQL> ed
Wrote file afiedt.buf

  1  begin
  2     for i in reverse 1..10 loop
  3             dbms_output.put_line('i = ' || i);
  4     end loop;
  5* end;
SQL> /
i = 10
i = 9
i = 8
i = 7
i = 6
i = 5
i = 4
i = 3
i = 2
i = 1

PL/SQL procedure successfully completed.
```

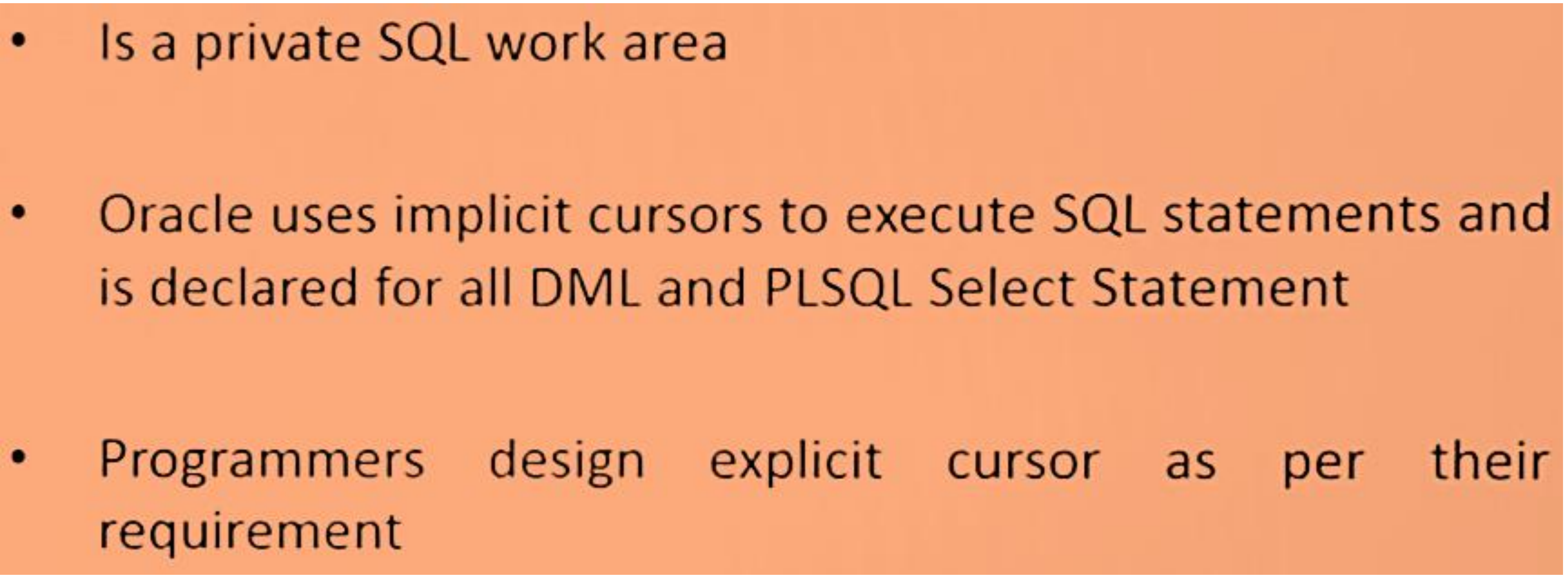
1.To find sum of two number:

```
declare
a int;
b int;
c int;
begin
a:=&a;
b:=&b;
c:=a+b;
dbms_output.put_line('sum of a and b'||c);
end;
```

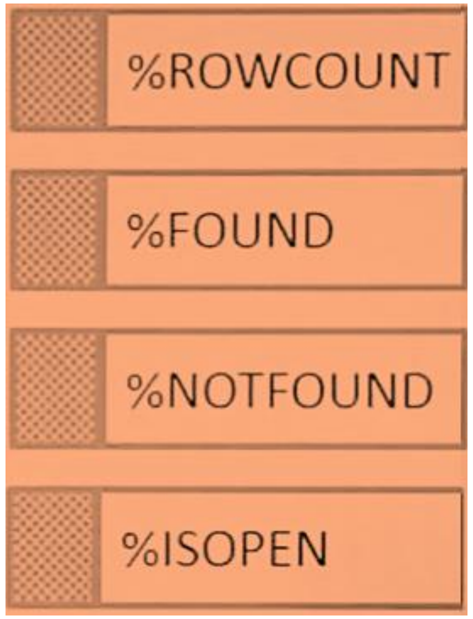
2.To find greatest number among two:

```
declare
a int;
b int;
c int;
begin
a:=&a;
b:=&b;
if(a>b)
then
dbms_output.put_line('a is greater');
else
dbms_output.put_line('b is greater');
end if;
end;
```

PL/SQL - Implicit Cursors



- Is a private SQL work area
- Oracle uses implicit cursors to execute SQL statements and is declared for all DML and PLSQL Select Statement
- Programmers design explicit cursor as per their requirement

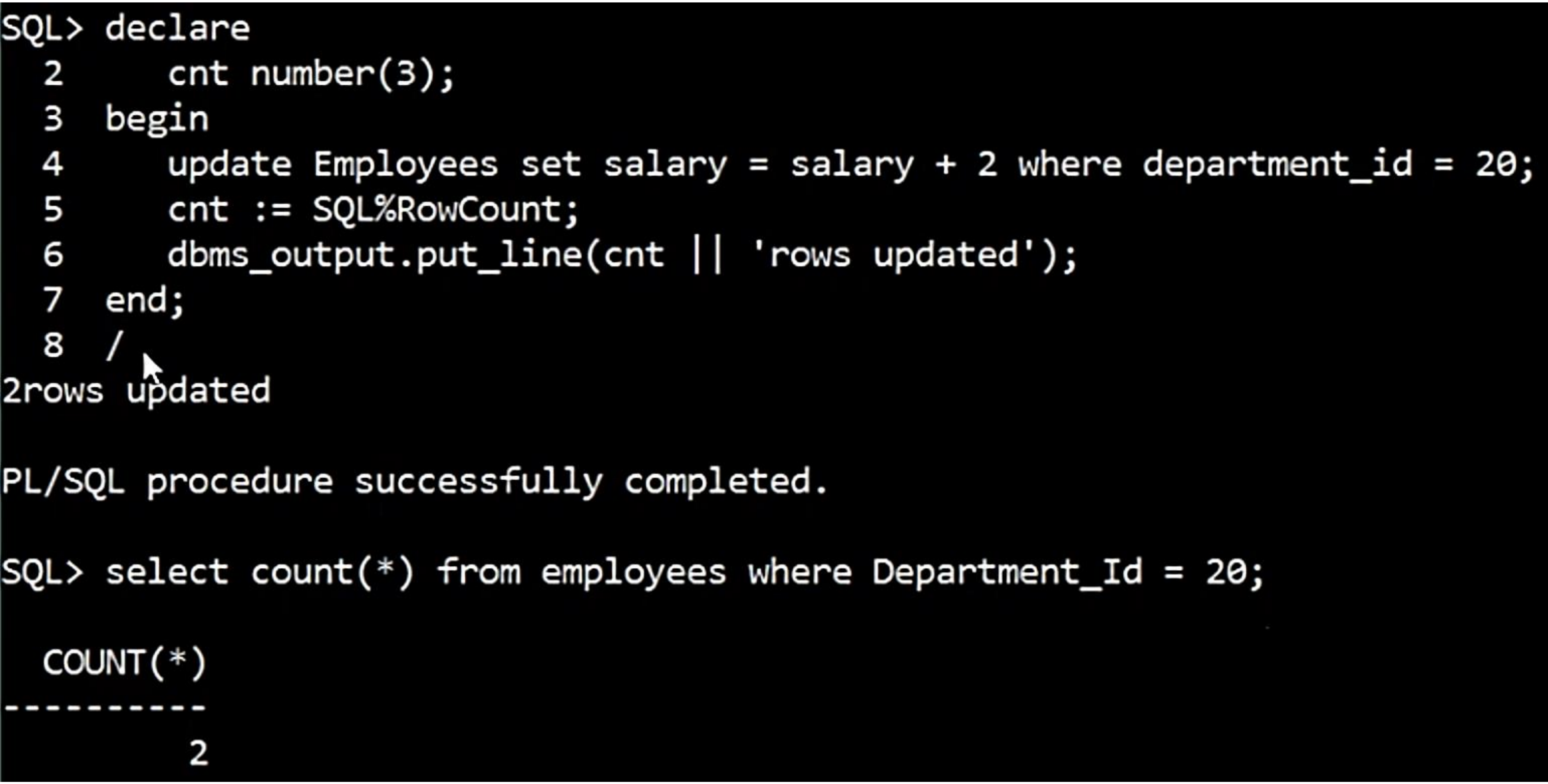


%ROWCOUNT

%FOUND

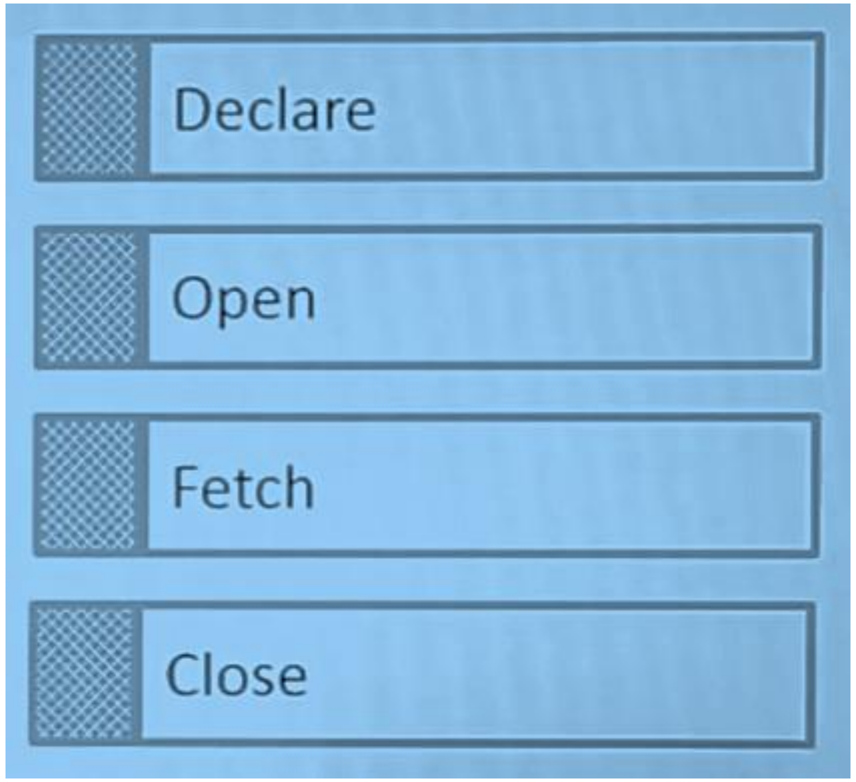
%NOTFOUND

%ISOPEN



```
SQL> declare
        cnt number(3);
     begin
        update Employees set salary = salary + 2 where department_id = 20;
        cnt := SQL%RowCount;
        dbms_output.put_line(cnt || 'rows updated');
     end;
     /
2rows updated

PL/SQL procedure successfully completed.

SQL> select count(*) from employees where Department_Id = 20;

  COUNT(*)
----------
         2
```

Steps in Explicit Cursors



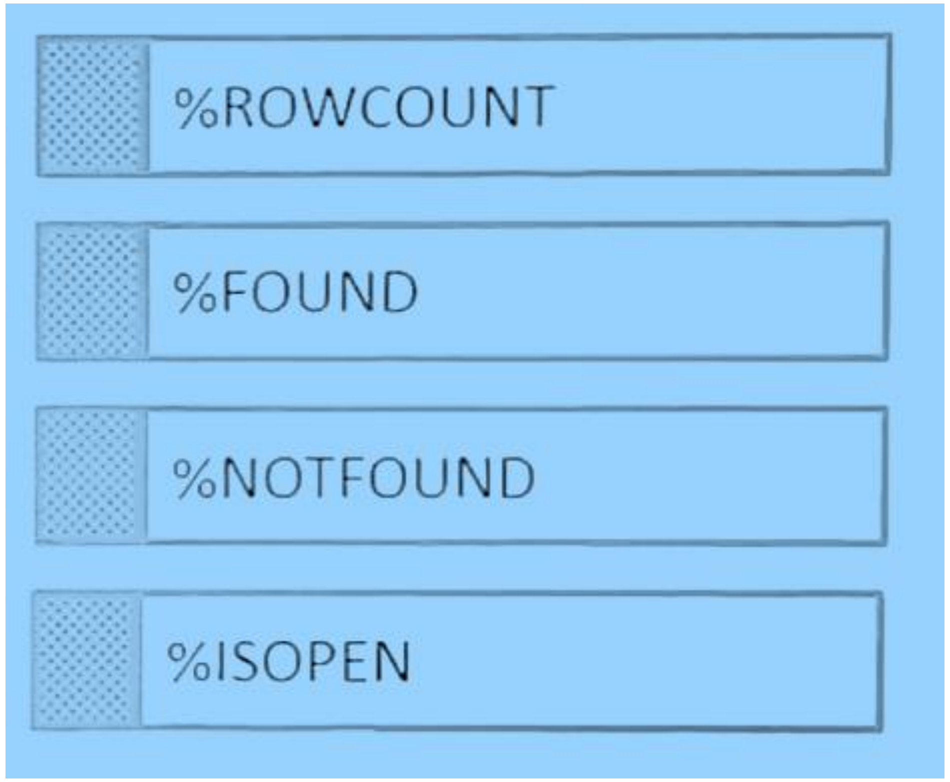
Declare

Open

Fetch

Close

Cursors Attributes

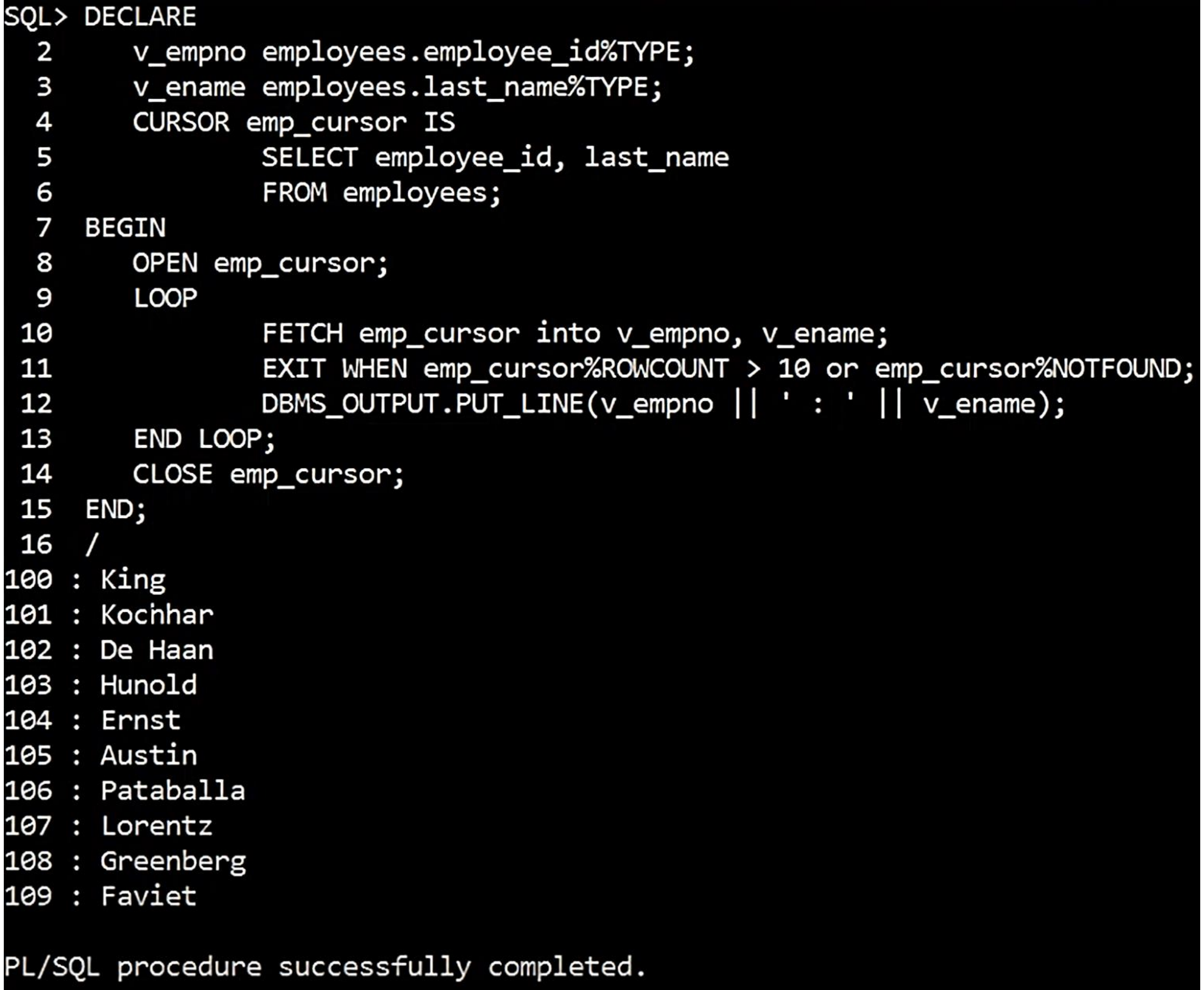


%ROWCOUNT

%FOUND

%NOTFOUND

%ISOPEN



```
SQL> DECLARE
        v_empno employees.employee_id%TYPE;
        v_ename employees.last_name%TYPE;
        CURSOR emp_cursor IS
                SELECT employee_id, last_name
                FROM employees;
     BEGIN
        OPEN emp_cursor;
        LOOP
                FETCH emp_cursor into v_empno, v_ename;
                EXIT WHEN emp_cursor%ROWCOUNT > 10 or emp_cursor%NOTFOUND;
                DBMS_OUTPUT.PUT_LINE(v_empno || ' : ' || v_ename);
        END LOOP;
        CLOSE emp_cursor;
     END;
     /
100 : King
101 : Kochhar
102 : De Haan
103 : Hunold
104 : Ernst
105 : Austin
106 : Pataballa
107 : Lorentz
108 : Greenberg
109 : Faviet

PL/SQL procedure successfully completed.
```