

المحاضرة الرابعة

الاجراءات والدوال

Procedures and Functions

تعريف الإجراءات Declaring Procedures -:

الإجراء هو برنامج جزئي من البرنامج العام يتكون من مجموعة من التعليمات أو الأوامر تقوم بتنفيذ شيء معين مثل أن نمرر إلى الإجراء بيانات الموظف ومن ضمنها الراتب بالريال فيقوم الإجراء بتحويل الراتب من ريال إلى دولار ويحفظ بيانات الموظف في الجدول. كما هو معروف فإن الإجراء يمكن أن نمرر له قيم ولكن لا يرجع قيمة وهذا هو الفرق الأساسي بينه وبين الدالة .

Syntax: -

```
PROCEDURE pro_name [ (argument1 [IN |OUT] arg_datatype [,.....] ) ] IS
Local variables declaration ;
BEGIN
    statements ;
END [ pro_name ] ;
```

مثال: إجراء يقوم بتحويل راتب الموظف إلى دولار ويحفظه :-

```
SQL> declare
2 my_empno my_emp.empno%type;
3 my_ename my_emp.ename%type;
4 my_job my_emp.job%type;
5 my_sal my_emp.sal%type;
6 my_deptno my_emp.deptno%type;
7 procedure to_dolar (pro_empno number,pro_ename varchar2,pro_job
varchar2,
8 pro_sal number,pro_deptno number) is
9 dolar_sal number(8,2);
10 begin
11 dolar_sal := pro_sal / 3.75;
12 insert into my_emp
13 (empno,ename,job,sal,deptno)
14 values
```

```

15 (pro_empno,pro_ename,pro_job,dolar_sal,pro_deptno);
16 commit;
17 end to_dolar;
18 BEGIN
19 my_empno := 7777;
20 my_ename := 'AHMED';
21 my_job := 'MANAGER';
22 my_sal := 3750;
23 my_deptno := 30;
24 -- تمرير المتغيرات للإجراء
25 to_dolar (my_empno,my_ename,my_job,my_sal,my_deptno);
26 end;
27 /
    
```

PL/SQL procedure successfully completed.

تعريف الدوال Declaring Functions :-

الدالة هي أيضا برنامج جزئي من البرنامج العام تتكون من مجموعة من التعليمات أو الأوامر ، والدالة شبيهة بالإجراء إلا أنها تعيد قيمة أي بمعنى أنه يتم تمرير قيم لها فتعالجها وترجع النتيجة إلى الوحدة النمطية المعينة التي تم مناداة الدالة منها. وفي الدالة يجب أن نحدد لها فقط نوع بيانات القيمة التي سترجعها.

Syntax: -

```

FUNCTION fun_name [ (argument-1 [IN |OUT] arg_datatype [,.....] ) ]
RETURN function_datatype IS
Local variables declaration ;
BEGIN
    statements ;
.
END [ fun_name ] ;
    
```

مثال: تحويل المثال السابق من إجراء إلى دالة :-

في هذا المثال سنقوم بتمرير قيمة الراتب بالريال إلى الدالة فتقوم بتحويله إلى دولار وإرجاع الناتج.

```

SQL> declare
2 my_empno my_emp.empno%type;
3 my_ename my_emp.ename%type;
4 my_job my_emp.job%type;
5 my_sal my_emp.sal%type;
6 my_deptno my_emp.deptno%type;
    
```

7 dolar_sal number(8,2); (تم تعريف المتغير خارج الدالة لأنه سيتم إرجاعه)

8 function to_dolar (fun_sal in number)

9 return number is

10 begin

11 dolar_sal := fun_sal / 3.75;

12 return dolar_sal;

13 end to_dolar;

14 BEGIN

15 my_empno := 9999;

16 my_ename := 'RXR';

17 my_job := 'MANAGER';

18 my_sal := 3750;

19 my_deptno := 30;

20 insert into my_emp

21 (empno,ename,job,sal,deptno)

22 values

23 (my_empno,my_ename,my_job,dolar_sal,my_deptno);

24 commit;

25 END;

26 /

PL/SQL procedure successfully completed.

وسائط الدوال والإجراءات Arguments :-

كما سبق وعرفنا من الصيغة العامة لتعريف الوسائط Arguments في الدوال والإجراءات فإنه يتم تعريفها بواحد من الطرق التالي :-

IN	عند تعريف الوسيط Argument على إنه IN فهذا يعني بأنه سيتم تمريره إلى الدالة أو الإجراء ولكن لن يعيد قيمة . مع ملاحظة أننا لا يمكننا إسناد قيمة للوسيط المعرف IN بل نمرر له قيمة فقط.
OUT	عند تعريف الوسيط Argument على إنه OUT فهذا يعني بأن الدالة أو الإجراء لن تستخدم أي قيمة ممررة إلى هذا الوسيط ولكنها تعيد قيمة إلى الوحدة النمطية التي تم مناداتها منها. مع ملاحظة أن الوسيط المعرف OUT يمكننا فقط إسناد قيمة له.
IN OUT	عند تعريف الوسيط Argument على إنه IN OUT فهذا يعني بأننا يمكن أن نعامله بكلتا الطريقتين السابقتين.

مثال: التحقق من أنواع الوسائط:-

```
SQL> declare
2 arg_in number := 0;
3 arg_out number := 0;
4 arg_inout number := 0;
5 procedure test_arg (arg1 in number,arg2 out number,arg3 in out number) is
6 begin
7 arg2 := arg1 * 5;
8 arg3 := arg3 + 10;end;
9 end test_arg;
10 begin
11 dbms_output.enable;
12 arg_in := 5;
13 arg_out:=3;
14 test_arg(arg_in,arg_out,arg_inout);
15 dbms_output.put_line("The IN argument value is:- "||arg_in);
16 dbms_output.put_line("The OUT argument value is:- "||arg_out);
17 dbms_output.put_line("The IN OUT argument value is:- "||arg_inout);
18 end;
19 /
The IN argument value is:- 5
The OUT argument value is:- 25
The IN OUT argument value is:- 10
```

PL/SQL procedure successfully completed.

مثال متكامل عن الدوال :-

إذا كان لدينا جدول باسم **employee** ومخزنة فيه البيانات التالية

serial	emp_no	emp_name	emp_job	salary
1		ahmed	manager	7000
2		omar	programmer	8500
3		ali	teacher	6200
4		saad	programmer	8300
5		fahad	System analyst	7100

1. اكتب وحدة نمطية تقوم بتخزين أرقام الموظفين emp_no وأرقامهم هي (10,20,30,40,50) على التوالي

```
SQL> declare
2 i number;
3 x number;
4 begin
5 x:=0;
6 for i in 1..5 loop
7 x:=x+10;
8 update employee
9 set
10 emp_no=x
11 where
12 serial=i;
13 end loop;
14 end;
15 /
```

PL/SQL procedure successfully completed.

الجدول employee بعد تنفيذ الوحدة النمطية

```
SQL> select * from employee;
```

SERIAL	EMP_NO	EMP_NAME	EMP_JOB	SALARY
1	10	ahmed	manager	7000
2	20	omar	programmer	8500
3	30	ali	teacher	6200
4	40	saad	programmer	8300
4	40	saad	programmer	8300

2. اكتب دالة تقوم بتحويل راتب الموظف salary من دولار إلى ريال لكافة الموظفين في الجدول

```
SQL> declare
2 i number;
3 s employee.salary %type;
4 riyal_sal number(8,2);
5 function to_riyal(fun_sal in number)
6 return number is
7 begin
```

```

8  riyal_sal:=fun_sal*3.75;
9  return riyal_sal;
10 end to_riyal;
11 begin
12 for i in 1..5 loop
13 select salary into  s from employee where serial=i;
14 riyal_sal:=to_riyal(s);
15 update employee
16 set
17 salary=riyal_sal
18 where
19 serial=i;
20 end loop;
21 end;
22 /

```

PL/SQL procedure successfully completed.

SQL> select * from employee;

SERIAL	EMP_NO	EMP_NAME	EMP_JOB	SALARY
-----	-----	-----	-----	-----
1	10	ahmed	manager	26250
2	20	omar	programmer	31875
3	30	ali	teacher	23250
4	40	saad	programmer	31125
5	50	fahad	system analyst	26625