

المحاضرة الثالثة

متابعة الحلقات التكرارية وإظهار رسائل المخرجات

3. التكرار المحدد FOR LOOP :-

Syntax: -

```
FOR loop variable IN [ REVERSE ] Lower value . . Upper value LOOP
    statements ;
END LOOP ;
```

مثال: استخدام FOR LOOP :-

```
FOR i IN 1 .. 100 LOOP
    i := i + 1 ;
    Insert into my_table
    (std_no)
    values
    ( i );
END LOOP ;
```

مثال:

1. انشئ جدول باسم student مكون من حقل واحد باسم st_no

2. ثم قم بانشاء وحدة نمطية تقوم باضافة 50 سجل في الجدول بالاعداد من 1 الى 50

```
create table student( st_no number );
```

```
SQL> declare
2 i number ;
3 begin
4 for i in 1..50 loop
5 insert into student(st_no)
6 values
7 (i);
8 end loop;
9 end;
10 /
```

PL/SQL procedure successfully completed.

مثال: انشئ وحدة نمطية تقوم بطباعة الاعداد من 50 الى 1 بشكل تنازلي:-

```
SQL> Set serveroutput on;
SQL> declare
  2 i number;
  3 begin
  4 dbms_output.enable;
  5 for i in reverse 1 ..50 loop
  6 dbms_output.put_line('i='||i);
  7 end loop;
  8 end;
  9 /
```

PL/SQL procedure successfully completed.

هام جداً :-

يمكننا استخدام For – Loop بصيغة أخرى مع جملة Select كالتالي :-
 FOR variable IN (SELECT one_column FROM table_name [WHILE
 condition]
 مع ملاحظة أنه ليس من الضروري تعريف متغير FOR



مثال: يوضح استخدام حلقة for مع جملة select:-

```
SQL> begin
  2 dbms_output.enable;
  3 dbms_output.put_line('empno sal');
  4 for i in (select empno,sal from emp where sal>=2500) loop
  5 dbms_output.put_line(i.empno||' '||i.sal);
  6 end loop;
  7 end;
  8 /
```

```
empno sal
7566 2975
7698 2850
7788 3000
7839 5000
7902 3000
```

PL/SQL procedure successfully completed.

مثال : يوضح استخدام حلقة for مع جملة select :-

```
SQL> begin
2 dbms_output.enable;
3 dbms_output.put_line('empno ename sal');
4 for i in (select * from emp ) loop
5 dbms_output.put_line(i.empno||' '||i.ename || ' '||i.sal);
6 end loop;
7 end;
8 /
```

```
empno ename sal
7369 SMITH 800
7499 ALLEN 1600
7521 WARD 1250
7566 JONES 2975
7654 MARTIN 1250
7698 BLAKE 2850
7782 CLARK 2450
7788 SCOTT 3000
7839 KING 5000
7844 TURNER 1500
7876 ADAMS 1100
7900 JAMES 950
7902 FORD 3000
7934 MILLER 1300
```

PL/SQL procedure successfully completed.

4. التفرع باستخدام GOTO :-

توفر لغة **PL / SQL** إمكانية التفرع باستخدام **GOTO** ، وبالرغم من أننا قد نحتاج لها أحياناً إلا أننا نقول بأنها من الأوامر الخطرة . لأنه عند استخدامها في البرمجة قد نفقد التسلسل الصحيح للبرنامج لذلك يجب أن نكون حريصين جداً عند استخدامها.

Syntax: -

```
GOTO label name ;
.
.
.
<< the label name >>
```

هام :-	
<input checked="" type="checkbox"/>	يجب أن نكتب اسم ال Label Name بين قوسين كالتالي >> << حتى يستطيع الأوراكل أن يميز أنها اسم ل Label Name وليست جملة أو أمر.

مثال: استخدام GOTO :-

```
LOOP
    i := i + 1 ;
    if i > 100 then
        goto other_processing ;
    end if ;
END LOOP
<< other_processing >>
i := 1
```

مثال آخر توضيحي لاستخدام GOTO :-

```
SQL> declare
2  i number:=0;
3  begin
4  dbms_output.enable;
5  loop
6  i:=i+1;
7  dbms_output.put_line('i'||i);
8  if i>=10 then
9  goto lab;
10 end if ;
11 end loop;
12 <<lab >>
13 dbms_output.put_line('number greater than 10 ');
14 end;
15 /
```

```
i=1  
i=2  
i=3  
i=4  
i=5  
i=6  
i=7  
i=8  
i=9  
i=10  
number greater than 10
```

PL/SQL procedure successfully completed.

كتابة الملاحظات في PL / SQL :-

كما هو معروف فإن كتابة الملاحظات **Comments** تعتبر من الأشياء الضرورية في توثيق البرنامج **Documentation** حتى يسهل قرأته وتعديله عند الحاجة.

لذلك توفر لنا ال **PL / SQL** طريقتين في التعامل مع كتابة الملاحظات **Comments** داخل البرنامج وهما:-

- استخدام الشرطتين --.
- استخدام طريقة لغة C++ /* */ .

مثال: استخدام الشرطتين :-

```
SQL> declare  
2 i number :=1;  
3 begin  
4 i:=i+1;  
5 --Increment Counter  
6 end;  
7 /
```

PL/SQL procedure successfully completed.

مثال: استخدام طريقة لغة السي :-

```
SQL> declare
2 i number :=1;
3 begin
4 i:=i+1;
5 /* Increment
6 Counter */
7 end;
8 /
```

PL/SQL procedure successfully completed.

إظهار رسائل المخرجات :-

عندما تم تصميم لغة **PL / SQL** في الأساس لم يكن لها المقدرة على إظهار المخرجات (تذكر أن ال **PL SQL** ليست لغة متكاملة **Not Standalone Language**) وكان يتم التعامل مع المخرجات بواسطة البرامج والأدوات المساعدة التي تستخدم مع ال **PL / SQL** .
حالياً قامت شركة الأوراكل بتضمين حزمة **DBMS_OUTPUT** مع ال **PL / SQL** لتوفر بعض الإمكانيات المحدودة في التعامل مع المخرجات. ويمكننا أن نستخدم الإجراء **DBMS_OUTPUT.put_line** كما سنوضح في المثال التالي.

Syntax: -

```
Dbms_output.enable ;
.
.
.
dbms_output.put_line ( 'any message' || variable ) ;
```

هام :-

حتى يستطيع الأوراكل إظهار الرسالة يجب أن ننبهه لذلك باستخدام الأمر
SQL> SET SERVEROUTPUT ON
مع ملاحظة أننا نحتاج هذا الأمر مرة واحدة فقط في كل **جلسة عمل**.



مثال لاستخدام dbms_output :-

```
SQL> set serveroutput on
SQL> declare
  2 i number :=1;
  3 begin
  4 dbms_output.enable;
  5 loop
  6 i := i + 1;
  7 exit when i=100;
  8 end loop;
  9 dbms_output.put_line('The max number is :- '||i);
 10 end;
 11 /
The max number is :- 100
```

PL/SQL procedure successfully completed.

مثال اخر لاستخدام dbms_output :-

انشئ وحدة نمطية تقوم بطباعة الاعداد من 1 الى 10 ثم حساب مجموعها
ومتوسطها ثم طباعتها على الشاشة :-

```
SQL> declare
  2 i number;
  3 s number:=0;
  4 avgerage number(6,2);
  5 begin
  6 dbms_output.enable;
  7 for i in 1..10 loop
  8 s:=s+i;
  9 dbms_output.put_line('the number is :'||i);
 10 end loop;
 11 dbms_output.put_line('the sum is :'||s);
 12 avgerage:=s/10;
 13 dbms_output.put_line('the average is :'||avgerage);
 14 end;
 15 /
```

the number is :1
the number is :2
the number is :3
the number is :4
the number is :5
the number is :6
the number is :7
the number is :8
the number is :9
the number is :10
the sum is :55
the average is :5.5

PL/SQL procedure successfully completed.