

المحاضرة السادسة

الحزم PACKAGES

ال **Package** عبارة عن وعاء أو مغلف يحتوي على مجموعة من الدوال والإجراءات المتعلقة بتطبيق معين، ويتم عمل الترجمة **Compilation** له ثم يحفظ في قاعدة البيانات ، وهو يساعدنا في تنظيم التطبيقات داخل قاعدة البيانات وتحسين الأداء لأنه يقوم بتحميل مجموعة من الأشياء إلى الذاكرة مرة واحدة. وهو يتكون من جزأين هما: -

◆ **Package Specification** ويتم فيه تعريف الدوال والإجراءات والمتغيرات.

◆ **Package Body** ويتم فيه كتابة الدوال والإجراءات كاملة.

◆

Package specification Syntax: -

CREATE [OR REPLACE] PACKAGE *package_name* IS

Declaring procedures & functions & cursors & variables (الاسم والوسائط فقط)

END *package_name*;

مثال: إنشاء Package Specification :-

SQL> CREATE OR REPLACE PACKAGE my_first_pack IS

2 procedure pack_del_record(del_rec_no number);

3 function pack_to_dolar(reyal number,d_val number)

4 return number;

5 end;

6 /

Package created.

Package Body Syntax:-

CREATE PACKAGE BODY *package_name* IS (نفس الاسم السابق)

Declaration section;

Procedures bodies;

Function bodies;

Initialization section; (يتم تنفيذه مرة واحدة فقط ، عند الاستدعاء)

END *package_name*;

مثال: إنشاء Package Body :-

```
SQL> CREATE OR REPLACE PACKAGE BODY my_first_pack is
2  PROCEDURE pack_del_record(del_rec_no number) IS
3  BEGIN
4  delete from my_emp
5  where
6  empno = del_rec_no;
7  END;
8  --*****
9  function pack_to_dolar(reyal number,d_val number)
10  return number is
11  dolar number;
12  begin
13  if d_val != 0 then
14      dolar := reyal / d_val;
15  else
16      dolar := 0;
17  end if;
18  return dolar;
19  end;
20  end;
21 /
```

Package body created.

استدعاء الدوال والإجراءات من الـ Package :-

يتم استدعاء الدوال والإجراءات من داخل الوحدات النمطية وذلك بذكر اسم الـ Package يليه نقطة ثم اسم الدالة أو الإجراء المراد استدعاءه. كما في الأمثلة التالية.

مثال: استدعاء الدالة Pack to dolar :-

```
SQL> declare
2  m number;
3  begin
4  dbms_output.enable;
5  m := my_first_pack.pack_to_dolar(4500,3.75);
6  dbms_output.put_line('The result is :- '||m);
7  end;
8  /
```

The result is :- 1200

مثال: استدعاء الإجراء Pack del record :-

```
SQL> begin
2 my_first_pack.pack_del_record(9999);
3 end;
4 /
```

PL/SQL procedure successfully completed.

مثال:

أنشي Package تحتوي على

- a. إجراء يقوم بطباعة الأعداد من 1 إلى n
- b. دالة تقوم بإيجاد حاصل جمع الأعداد من 1 إلى n

الحل:

أولاً: Package Specification

```
SQL> CREATE OR REPLACE PACKAGE summation IS
2 procedure print_numbers(n number);
3 function numbers_sum(m number)
4 return number;
5 end;
6 /
```

Package created.

ثانياً: Package Body

```
SQL>CREATE OR REPLACE PACKAGE BODY summation is
2 i number;
3 procedure print_numbers(n number) IS
4 BEGIN
5 dbms_output.enable;
6 for i in 1..n loop
7 dbms_output.put_line('i='||i);
8 end loop;
9 END;
10 --*****
11 function numbers_sum(m number)
12 return number is
13 s number:=0;
14 begin
15 dbms_output.enable;
16 for i in 1..m loop
```

```
17 s:= s +i;
18 end loop;
19 return s;
20 end;
21 end;
22 /
```

Package body created.

ثالثاً : استدعاء الاجراء print_numbers

```
SQL> begin
  2 summation.print_numbers (10);
  3 end;
  4 /
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.

رابعاً : استدعاء الدالة numbers_sum

```
SQL> declare
  2 total number;
  3 begin
  4 total := summation.numbers_sum (12);
  5 dbms_output.put_line('the sum is :'|| total);
  6 end;
  7 /
```

the sum is :78

PL/SQL procedure successfully completed.

انواع البيانات Tables & Records

لغة ال PL / SQL تدعم نوعين آخرين من أنواع البيانات هما Table & Record وكل منهما يتم تعريفه على أنه نوع جديد من أنواع البيانات ثم نقوم بتعريف متغيرات أخرى تتبع لهذا النوع الجديد.

Table Datatype :-

يمكن تشبيه نوع البيانات Table على أنه مصفوفة ذات بعد واحد One Dimension Array وهو يحتوي على مجموعة من البيانات من نفس النوع يتم ترتيبها ترتيب رقمي.

Syntax: -

```
TYPE type_name IS TABLE OF  
existing_table_name.column_name%type  
INDEX BY BINARY_INTEGER ;
```

مثال: استخدام نوع البيانات Table :-

```
SQL> declare  
2 TYPE empno_type IS TABLE OF my_emp.empno%type  
3 INDEX BY BINARY_INTEGER;  
4 my_array empno_type;  
5 i number :=0;  
6 my_rec_no number;  
7 BEGIN  
8 dbms_output.enable;  
9 for j in (select empno from my_emp) loop  
10 i := i + 1;  
11 my_array(i) := j.empno;  
12 end loop;  
13 my_rec_no := i;  
14 for i in 1..my_rec_no loop  
15 dbms_output.put_line('empno('||i||') = '||my_array(i));  
16 end loop;  
17 end;  
18 /
```

(أنظر النتيجة بالصفحة التالية)

عند التنفيذ سوف تظهر النتيجة بالشكل التالي:-

```
empno(1) = 7369
empno(2) = 7499
empno(3) = 7521
empno(4) = 7566
empno(5) = 7654
empno(6) = 7698
empno(7) = 7782
empno(8) = 7788
empno(9) = 7839
empno(10) = 7844
empno(11) = 7876
empno(12) = 7900
empno(13) = 7902
empno(14) = 7934
empno(15) = 7777
```

PL/SQL procedure successfully completed.

Record Datatype :-

نوع البيانات Record عبارة عن مجموعة من الحقول المنفصلة بحيث يمكن أن تكون سجلاً كاملاً في جدول أو جزء من سجل أو حتى جزء من سجل ومعه حقول من خارج الجدول. حيث يمكننا من تجميع عدة بيانات في هيكل تنظيمي واحد. مثله مثل نوع البيانات ROWTYPE % إلا أن الأخير يشترط أن يكون سجلاً كاملاً من جدول واحد فقط ، ويمكننا أيضاً أن نشير إلى أي عنصر داخل نوع البيانات Record باسم ال Record ثم النقطة يليها اسم الحقل الموجود داخل السجل.

Syntax: -

```
TYPE record_name IS RECORD
(col-1 datatype [ not null ] [ initial_value ] , [ ... ] ) ;
```

إذا تم تعريف الحقل Not Null فلا بد من أن نعطيه قيمة ابتدائية.	<input checked="" type="checkbox"/>
---	-------------------------------------

مثال: استخدام نوع البيانات Record :-

```
SQL> declare
2  TYPE rxr_arguments_type IS RECORD
3  (my_empno my_emp.empno%type,my_ename my_emp.ename%type,
4  my_job my_emp.job%type,my_sal my_emp.sal%type,
5  my_deptno my_emp.deptno%type);
6
6  wal_arg rxr_arguments_type;
7
7  procedure rxr_pro (wal_pro_arg rxr_arguments_type) is
8  dolar_sal number(8,2);
9  begin
10   dolar_sal := wal_pro_arg.my_sal / 3.75;
11   insert into my_emp
12   (empno,ename,job,sal,deptno)
13   values
14   (wal_pro_arg.my_empno,wal_pro_arg.my_ename,wal_pro_arg.my_job,
15   wal_pro_arg.my_sal,wal_pro_arg.my_deptno);
16   commit;
17   end rxr_pro;
18
18  BEGIN
19   wal_arg.my_empno := 1122;
20   wal_arg.my_ename := 'RXR';
21   wal_arg.my_job := 'MANAGER';
22   wal_arg.my_sal := 3750;
23   wal_arg.my_deptno := 30;
24   -- تمرير المتغيرات للإجراء
25   rxr_pro(wal_arg);
26   end;
27   /
```

PL/SQL procedure successfully completed.