المحاضرة الثامنة

استرجاع البيانات باستخدام المؤشرات Cursors

يمكننا في الـ **SQL-Plus** استرجاع عدة سجلات باستخدام **Select** من أي جدول بسهولة ويسر ولكن هذه الطريقة غير مقبولة في لغة **PL / SQL** حيث جملة **Select** إذا أرجعت أكثر من سجل فسوف يعطي الأوراكل رسالة خطاء **too_many_rows** لذلك توفر لنا لغة الـ **PL / SQL** طريقة للتعامل مع إرجاع عدة بيانات وهي تعرف بـ **Cursor** حيث يمكننا اعتبارها مثل إطار أو نافذة تعرض سجلاً واحداً من البيانات المسترجعة في كل مرة (أنظر الرسم أدناه).

Empno	Ename	Jop
7369	SMITH	CLERK
7499	ALLEN	SALESMAN
7521	WARD	SALESMAN
7566	JONES	MANAGER
7654	MARTIN	SALESMAN
7698	BLAKE	MANAGER

Cursor

ويوجد نوعان من المؤشرات **Cursors** مستخدمة في الأوراكل هما :-

- المؤشر الصريح **Explicit Cursors**
- المؤشر الضمني **Implicit Cursor**

النوع الأول: المؤشر الصريح Explicit Cursors :-

وهي الـ **Cursor** التي يقوم المبرمج أو المستخدم بتعريفها لتخدم له غرض إرجاع عدة سجلات، ويمكن أن نمرر للـ **Cursor** متغيرات لتستخدم في جملة الشرط **Where** ولتعريف الـ **Cursor** نحتاج للخطوات التالية:-

(1) - Declare the cursor.
(2) - Open the cursor.
(3) - Fetch data from the cursor.
(4) - Close the cursor.

Syntax of declaration: -

CURSOR *cursor_name* **(***parameter1 datat_type* [:= *default_value*] , …….**)** IS
SELECT *select_statements* ;

Syntax of opening: -

OPEN *cursor_name* [*parameters_values*] ;

Syntax of fetching: -

```
LOOP
        FETCH cursor_name INTO mem_variables ;
        EXIT When cursor_name%NOTFOUND ;
        …
        …
END LOOP
```

Syntax of closing: -

CLOSE *cursor_name* ;

☑	**هام:-** توجد طريقة أخرى لفتح الـ Explicit Cursors باستخدام For – Loops سوف ندرسها لاحقاً.

مثال: استخدام Cursor بدون متغيرات :-

```
SQL> DECLARE
  2  mem_empno emp.empno%type;
  3  mem_ename emp.ename%type;
  4  mem_job   emp.job%type;
  5  CURSOR my_first_cursor is
  6  select empno,ename,job from emp
  7  where sal >=1500;
  8  BEGIN
  9  open my_first_cursor;
 10  dbms_output.enable;
 11  loop
 12  fetch my_first_cursor into
 13  mem_empno,mem_ename,mem_job;
 14  exit when my_first_cursor%notfound;
 15  dbms_output.put_line(mem_ename||' : '||mem_job);
 16  end loop;
 17  close my_first_cursor;
 18  end;
 19  /


ALLEN : SALESMAN
JONES : MANAGER
BLAKE : MANAGER
CLARK : MANAGER
SCOTT : ANALYST
KING : PRESIDENT
TURNER : SALESMAN
FORD : ANALYST

PL/SQL procedure successfully completed.
```

مثال: استخدام Cursor **بمتغيرات :-**

```
SQL> DECLARE
  2  mem_empno emp.empno%type;
  3  mem_ename emp.ename%type;
  4  mem_job   emp.job%type;
  5  mem_sal   emp.sal%type;
  6  CURSOR my_first_cursor (sal_val number) is
  7  select empno,ename,job from emp
  8  where sal >=sal_val;
  9  BEGIN
 10  dbms_output.enable;
 11  mem_sal := 1500;
 12  open my_first_cursor(mem_sal);
 13  loop
 14  fetch my_first_cursor into
 15  mem_empno,mem_ename,mem_job;
 16  exit when my_first_cursor%notfound;
 17  dbms_output.put_line(mem_ename||' : '||mem_job);
 18  end loop;
 19  close my_first_cursor;
 20  end;
 21  /


ALLEN : SALESMAN
JONES : MANAGER
BLAKE : MANAGER
CLARK : MANAGER
SCOTT : ANALYST
KING : PRESIDENT
TURNER : SALESMAN
FORD : ANALYST

PL/SQL procedure successfully completed.
```

مثال: تعريف Cursor بمتغيرات مع عدم تمرير قيمة للمتغيرات :-

```
SQL> DECLARE
  2  mem_empno emp.empno%type;
  3  mem_ename emp.ename%type;
  4  mem_job   emp.job%type;
  5  mem_sal   emp.sal%type;
  6  CURSOR my_first_cursor (sal_val number) is
  7  select empno,ename,job from emp
  8  where sal >=sal_val;
  9  BEGIN
 10  dbms_output.enable;
 11  mem_sal := 1500;
 12  open my_first_cursor;
 13  loop
 14  fetch my_first_cursor into
 15  mem_empno,mem_ename,mem_job;
 16  exit when my_first_cursor%notfound;
 17  dbms_output.put_line(mem_ename||' : '||mem_job);
 18  end loop;
 19  close my_first_cursor;
 20  end;
 21  /


DECLARE
*
ERROR at line 1:
ORA-06550: line 12, column 1:
PLS-00306: wrong number or types of arguments in call to 'MY_FIRST_CURSOR'
ORA-06550: line 12, column 1:
PL/SQL: SQL Statement ignored
```

عند تعريف **Cursor** تحتوي على متغيرات لابد من تمرير قيم لهذه المتغيرات عند فتح الـ **Cursor** وإلا فسوف يعطي الأوراكل رسالة الخطاء السابقة. ولتلافي رسالة الخطاء السابقة يمكننا أن نعطي المتغيرات قيمة ابتدائية عند تعريف الـ **Cursor** كما سنرى في المثال التالي.

مثال: إعطاء قيم ابتدائية للمتغيرات عند تعريف الـ Cursor :-

```
SQL> DECLARE
  2  mem_empno emp.empno%type;
  3  mem_ename emp.ename%type;
  4  mem_job   emp.job%type;
  5  mem_sal   emp.sal%type;
  6  CURSOR my_first_cursor (sal_val number:=1000) is
  7  select empno,ename,job from emp
  8  where sal >=sal_val;
  9  BEGIN
 10  dbms_output.enable;
 11  mem_sal := 1500;
 12  open my_first_cursor;
 13  loop
 14  fetch my_first_cursor into
 15  mem_empno,mem_ename,mem_job;
 16  exit when my_first_cursor%notfound;
 17  dbms_output.put_line(mem_ename||' : '||mem_job);
 18  end loop;
 19  close my_first_cursor;
 20  end;
 21  /
ALLEN : SALESMAN
WARD : SALESMAN
JONES : MANAGER
MARTIN : SALESMAN
BLAKE : MANAGER
CLARK : MANAGER
SCOTT : ANALYST
KING : PRESIDENT
TURNER : SALESMAN
ADAMS : CLERK
FORD : ANALYST
MILLER : CLERK

PL/SQL procedure successfully completed.
```

مثال:إعادة فتح الـ Cursor بقيم مختلفة :-

```
SQL> DECLARE
  2  mem_empno emp.empno%type;
  3  mem_ename emp.ename%type;
  4  mem_job   emp.job%type;
  5  mem_sal   emp.sal%type;
  6  CURSOR my_first_cursor (sal_val number:=1500) is
  7  select empno,ename,job from emp
  8  where sal >=sal_val;
  9  BEGIN
 10  dbms_output.enable;
 11  mem_sal := 1500;
 12  open my_first_cursor;
 13  loop
 14  fetch my_first_cursor into
 15  mem_empno,mem_ename,mem_job;
 16  exit when my_first_cursor%notfound;
 17  dbms_output.put_line(mem_ename||' : '||mem_job);
 18  end loop;
 19  close my_first_cursor;
 20  open my_first_cursor(2500);
 21  loop
 22  fetch my_first_cursor into
 23  mem_empno,mem_ename,mem_job;
 24  exit when my_first_cursor%notfound;
 25  dbms_output.put_line('The 2nd opening is: '||mem_ename||' : '||mem_job);
 26  end loop;
 27  end;
 28  /
ALLEN : SALESMAN
JONES : MANAGER
BLAKE : MANAGER
CLARK : MANAGER
SCOTT : ANALYST
KING : PRESIDENT
TURNER : SALESMAN
FORD : ANALYST
The 2nd opening is: JONES : MANAGER
The 2nd opening is: BLAKE : MANAGER
The 2nd opening is: SCOTT : ANALYST
The 2nd opening is: KING : PRESIDENT
The 2nd opening is: FORD : ANALYST

PL/SQL procedure successfully completed.
```

خصائص الـ Cursor :-

الـ **Cursor** بنوعيها **Explicit & Implicit** تمتلك أربع خصائص هي:-

(1)- %isopen

وهي تستخدم لمعرفة ما إذا كانت الـ **Cursor** مفتوحة أم لا. لأننا إذا حاولنا أن نفتح **Cursor** تم فتحها مسبقاً فسوف يعطينا الأوراكل رسالة خطاء **cursor_already_open** لذلك يمكننا أن نفتحها كالتالي:-

```
If   my_first_cursor%isopen   Then
        Process_data                الذهاب لتنفيذ بقية الإجراء
Else
        Open   my_first_cursor;
        Process_data;
End if;
<< process_data >>
…
…
```

(2)- %rowcount :-

وهي تستخدم لمعرفة عدد السجلات **Records** التي استرجعتها جملة **Select** الموجودة بالـ **Cursor**.

```
loop
fetch my_first_cursor into
mem_empno,mem_ename,mem_job;
exit when my_first_cursor%notfound;
dbms_output.put_line(mem_ename||' : '||mem_job);
end loop;
dbms_output.put_line(my_first_cursor%rowcount);
```

(3)- %notfound :-

كما سبق وعرفناها فهي تستخدم لمعرفة ما إذا كانت توجد بيانات داخل الـ **Cursor** أم لا.

(4)- %found :-

هي عكس السابقة وتستخدم لمعرفة هل تم إيجاد بيانات أم لا كما سيتضح لنا في المثال التالي.

مثال: استخدام %found :-

```
SQL> DECLARE
  2  mem_empno emp.empno%type;
  3  mem_ename emp.ename%type;
  4  mem_job   emp.job%type;
  5  mem_sal   emp.sal%type;
  6  CURSOR my_first_cursor (sal_val number:=10000) is
  7  select empno,ename,job from emp
  8  where sal >=sal_val;
  9  BEGIN
 10  dbms_output.enable;
 11  open my_first_cursor;
 12  if my_first_cursor%found then
 13  goto process_data;
 14  else
 15  dbms_output.put_line('No Data Found');
 16  goto finish;
 17  end if;
 18  <<process_data>>
 19  loop
 20  fetch my_first_cursor into
 21  mem_empno,mem_ename,mem_job;
 22  exit when my_first_cursor%notfound;
 23  dbms_output.put_line(mem_ename||' : '||mem_job);
 24  end loop;
 25  goto finish;
 26  <<finish>>
 27  close my_first_cursor;
 28  end;
 29  /

No Data Found

PL/SQL procedure successfully completed.
```

مثال: استخدام %rowcount :-

```
SQL> DECLARE
  2  mem_empno emp.empno%type;
  3  mem_ename emp.ename%type;
  4  mem_job   emp.job%type;
  5  mem_sal   emp.sal%type;
  6  CURSOR my_first_cursor (sal_val number:=1000) is
  7  select empno,ename,job from emp
  8  where sal >=sal_val;
  9  BEGIN
 10  dbms_output.enable;
 11  mem_sal := 1500;
 12  open my_first_cursor;
 13  loop
 14  fetch my_first_cursor into
 15  mem_empno,mem_ename,mem_job;
 16  exit when my_first_cursor%notfound;
 17  end loop;
 18  dbms_output.put_line('The total no of record found is:
'||my_first_cursor%rowcount);
 19  close my_first_cursor;
 20  end;
 21  /


The total no of record found is: 12

PL/SQL procedure successfully completed.
```