

TP n°3 : Les curseurs PL/SQL :

EXERCICE N°1 : requêtes simples et exception :

1) J'ai créé la table DEPT avec le script SQL 'dept.sql'

```
1 ligne créée.
SQL> select * from dept;

   DEPTNO DNAME                LOC
-----
      10 ACCOUNTING            NEW-YORK
      20 RESEARCH              DALLAS
      30 SALES                  CHICAGO
      40 OPERATIONS            BOSTON

SQL> desc dept;
  Nom                                NULL ?    Type
-----
DEPTNO                                NOT NULL  NUMBER(2)
DNAME                                CHAR(20)
LOC                                  CHAR(20)
```

2) J'ai créé la table EMP avec le script SQL 'emp.sql'.

```
SQL> desc emp;
  Nom                                NULL ?    Type
-----
EMPNO                                NOT NULL  NUMBER(4)
ENAME                                VARCHAR2(10)
JOB                                  VARCHAR2(9)
MGR                                  NUMBER(4)
HIREDATE                             DATE
SAL                                  NUMBER(7,2)
COMM                                NUMBER(7,2)
DEPTNO                              NUMBER(2)

SQL> select * from emp;

   EMPNO ENAME      JOB              MGR HIREDATE          SAL          COMM
-----
   DEPTNO
-----
      7839 KING        PRESIDENT              17/11/81          5000
      10
      7566 JONES        MANAGER                7839 02/04/81          2975
      20
      7902 FORD        ANALYST                7566 03/12/81          3000
      20
```

3)

```
SQL> declare
  2  p number;
  3  m number;
  4  n number;
  5  begin
  6  select count (*) into n from EMP;
  7  dbms_output.put_line('le nombre total de n-uplets est : ' ||n);
  8  select count (*) into m from EMP where JOB='MANAGER';
  9  dbms_output.put_line('le nombre de manager est : ' ||m);
 10  p:=(m/n)*100;
 11  dbms_output.put_line('le pourcentage est : ' ||p ||'%');
 12  end;
 13  /
le nombre total de n-uplets est : 12
le nombre de manager est : 3
le pourcentage est : 25%

Procédure PL/SQL terminée avec succès.
```

4)

```
SQL> declare
  2  e exception;
  3  p number;
  4  m number;
  5  n number;
  6  begin
  7  select count (*) into n from EMP;
  8  if (n=0) then
  9  RAISE e;
 10  end if;
 11  dbms_output.put_line('le nombre total de n-uplets est : ' ||n);
 12  select count (*) into m from EMP where JOB='MANAGER';
 13  dbms_output.put_line('le nombre de manager est : ' ||m);
 14  p:=(m/n)*100;
 15  dbms_output.put_line('le pourcentage est : ' ||p ||'%');
 16  exception
 17  when e then raise_application_error(-20500,'ERREUR! ON NE PEUT PAS DIVISER PAR ZERO! ');
 18  end;
 19  /
le nombre total de n-uplets est : 12
le nombre de manager est : 3
le pourcentage est : 25%

Procédure PL/SQL terminée avec succès.
```

1+2+3+4 :

```
SQL> commit;

Validation effectu e.

SQL> delete from emp;

12 ligne(s) supprim e(s).

SQL> declare
  2  e exception;
  3  p number;
  4  m number;
  5  n number;
  6  begin
  7    select count (*) into n from EMP;
  8    if (n=0) then
  9      RAISE e;
 10    end if;
 11    dbms_output.put_line('le nombre total de n-uplets est : ' ||n);
 12    select count (*) into m from EMP where JOB='MANAGER';
 13    dbms_output.put_line('le nombre de manager est : ' ||m);
 14    p:=(m/n)*100;
 15    dbms_output.put_line('le pourcentage est : ' ||p ||'%');
 16  exception
 17  when e then raise_application_error(-20500,'ERREUR! ON NE PEUT PAS DIVISER PAR ZERO! ');
 18  end;
 19  /
declare
*
ERREUR   la ligne 1 :
ORA-20500: ERREUR! ON NE PEUT PAS DIVISER PAR ZERO!
ORA-06512:   ligne 17

SQL> rollback;

Annulation (rollback) effectu e.
```

EXERCICE N 2 : Curseur implicite :

```
SQL> create or replace view EMPDIR as select EMPNO,ENAME from EMP;

Vue cr  e.

SQL> select * from TAB
  2  ;

TNAME                                TABTYPE  CLUSTERID
-----                                -
DEPT                                TABLE
EMP                                TABLE
EMPDIR                             VIEW

SQL> declare
  2  cursor c_cat is select TNAME,TABTYPE from TAB;
  3  v_cat c_cat%ROWTYPE ;
  4  begin
  5  for v_cat in c_cat loop
  6    dbms_output.put_line('L"objet ' ||v_cat.TNAME || ' est de type '
|| v_cat.TABTYPE);
  7  end loop;
  8  end;
  9  /
L"objet DEPT est de type TABLE
L"objet EMP est de type TABLE
L"objet EMPDIR est de type VIEW

Proc dure PL/SQL termin e avec succ s.
```

EXERCICE N°3 : Curseur explicite :

```
SQL> start C:\Users\Admin\Desktop\demo_product_info;  
Table créée.  
  
1 ligne créée.  
  
1 ligne créée.  
  
1 ligne créée.  
  
1 ligne créée.  
  
1 ligne créée.  
  
1 ligne créée.  
  
1 ligne créée.  
  
1 ligne créée.  
  
1 ligne créée.
```

```
SQL> declare  
2  cursor c_pro is select PRODUCT_NAME,LIST_PRICE from DEMO_PRODUCT  
_INFO ORDER BY LIST_PRICE DESC;  
3  v_pro c_pro%rowtype;  
4  begin  
5  open c_pro ;  
6  fetch c_pro into v_pro;  
7  while c_pro%FOUND and c_pro%ROWCOUNT<=5 loop  
8  dbms_output.put_line(c_pro%ROWCOUNT || ' c'est le produit ' || v_p  
ro.product_name || ' , son prix est = ' || v_pro.list_price);  
9  fetch c_pro into v_pro ;  
10 end loop;  
11 close c_pro;  
12 end;  
13 /  
1 c'est le produit 54' Plasma Flat Screen , son prix est = 3995  
2 c'est le produit Ultra Slim Laptop , son prix est = 1999  
3 c'est le produit 3.2 GHz Desktop PC , son prix est = 1200  
4 c'est le produit Portable DVD Player , son prix est = 500  
5 c'est le produit PDA Cell Phone , son prix est = 250  
  
Procédure PL/SQL terminée avec succès.
```

EXERCICE N°4 : Liste et curseur implicite :

C'est le résultat de l'exécution du bloc PL/SQL :

```
53  end loop;
54
55  end;
56  /
notes(1) = 15
notes(2) = 16
notes(3) = 12
notes(4) = 16
notes(5) = 12
notes(6) = 7
notes(7) = 17,5
notes(8) = 7
notes(9) = 8
notes(10) = 10
notes(11) = 5
notes(12) = 9
notes(13) = 15
notes(14) = 16
notes(15) = 12
notes(16) = 16
moyenne = 12,09375
AVG(notecc) = 12,09375
indice du premier élément de la collection : 1
indice du dernier élément de la collection : 16
nombre des éléments dans la collection: 16
indice du premier élément de la collection : 2
indice du dernier élément de la collection : 15
nombre des éléments dans la collection: 13
les éléments de la collection sont :
note: 16
note: 12
note: 16
note: 12
note: 7
note: 17,5
note: 7
note: 8
note: 5
note: 9
note: 15
note: 16
note: 12

Procédure PL/SQL terminée avec succès.
```

EXERCICE N°5 : enregistrement et curseur paramétré :

```
SQL> declare
  2
  3 --question1 :
  4 type rec is record (nom emp.ename%TYPE,
  5 fonction emp.job%TYPE,
  6 salaire_tot emp.sal%TYPE);
  7
  8 --question2 :
  9 r rec;
10
11 --question3:
12 CURSOR c(n NUMBER) IS SELECT ename, job, sal + NVL(comm, 0) from emp where deptno = n ;
13
14 begin
15
16 --question4:
17 open c(30);
18 fetch c into r;
19 while c%FOUND LOOP
20 DBMS_OUTPUT.PUT_LINE(r.nom || ' (' || r.fonction || ') : ' || r.salaire_tot || ' DT');
21 fetch c into r;
22 end loop;
23 close c;
24 end;
25 /
BLAKE (MANAGER) : 2850 DT
ALLEN (SALESMAN) : 1900 DT
WARD (SALESMAN) : 1750 DT
MARTIN (SALESMAN) : 2650 DT
TURNER (SALESMAN) : 1500 DT
JAMES (CLERK) : 950 DT

Procédure PL/SQL terminée avec succès.
```