المدرسة الوطنية للعلوم التطبيقية
ⵜⵉⵎⵙⵙⵉⵍⵜ ⵜⴰⵏⴰⵎⵓⵔⵜ ⵏ ⵜⵙⴻⵏⵙⵉⵡⵉⵏ ⵜⵉⵎⵙⵙⵉⵍⵜⵉⵏ
Ecole Nationale des Sciences Appliquées de Fès

جامعة سيدي محمد بن عبد الله
ⵜⴰⵙⴷⴰⵡⵉⵜ ⵙⵉⴷⵉ ⵎⵓⵃⴻⵎⵎⴷ ⴱⵏ ⵄⴱⴷⵍⵍⴰⵀ
Université Sidi Mohamed Ben Abdellah

Master Internet des Objets et Systèmes Mobiles
2021-2022

Module: S.I et Bases de données

Rapport des Travaux Pratiques
Série n° 4

Réalisé par :

ZAROUAL Mohammed

Encadré par :

Pr. EL AKKAD Nabil

Exercice 1:

1. Écrire une fonction qui permet de retourner le nom du département, prenant en paramètre le numéro du département.

```
SQL> create or replace function get_dept(num IN dept.deptno%type)
  2
  3      return dept.dname%type
  4
  5      is
  6
  7      v_dname dept.dname%type;
  8
  9      begin
 10
 11      select dname into v_dname from dept where deptno=num;
 12
 13      return (v_dname);
 14
 15   end;
 16   /

Function created.
```

Affichage:

```
SQL> set serveroutput on;
SQL> begin
  2   dbms_output.put_line(get_dept(10));
  3   end;
  4   /
ACCOUNTING

PL/SQL procedure successfully completed.
```

2. Refaire la même question en utilisant une procédure, prenant en paramètre le numéro du département et le nom du département.

```
create or replace procedure affiche_dept(num IN dept.deptno%type,nom OUT dept.dname%type)
    is
    begin

    select dname into nom from dept where deptno=num;

    dbms_output.put_line('le nom de departement ' || nom);

end;
/
```

Affichage:

```
SQL> set serveroutput on;
SQL> declare
  2   nom emp.ename%type;
  3   begin
  4   affiche_dept(20,nom);
  5   end;
  6   /
le nom de departement RESEARCH

PL/SQL procedure successfully completed.
```

Exercice 2 :

1. Ecrire une procédure (en utilisant les curseurs et la fonction de l'exercice 1) qui permet d'afficher les noms des employés sous la forme suivante:

   L'employé <<Nom employé>> a la profession <<Profession>> dans le département <<Nom département >>.

```
create or replace procedure affiche_emp
    is
    cursor curseur is select ename,job,deptno from emp;
    v_ename emp.ename%type;
    v_job emp.job%type;
    v_deptno emp.deptno%type;

    begin
    open curseur;
    fetch curseur into v_ename,v_job,v_deptno;
    while curseur %found loop
        dbms_output.put_line('l employe '|| v_ename || 'a la profession '|| v_job ||' dans le dept '|| get_dept(v_deptno));
        fetch curseur into v_ename,v_job,v_deptno;
    end loop;
    close curseur;
end;
/
```

Affichage:

```
SQL> set serveroutput on;
SQL> begin
  2  affiche_emp;
  3  end;
  4  /
l employe SMITHa la profession CLERK dans le dept RESEARCH
l employe ALLENa la profession SALESMAN dans le dept SALES
l employe WARDa la profession SALESMAN dans le dept SALES
l employe JONESa la profession MANAGER dans le dept RESEARCH
l employe MARTINa la profession SALESMAN dans le dept SALES
l employe BLAKEa la profession MANAGER dans le dept SALES
l employe CLARKa la profession MANAGER dans le dept ACCOUNTING
l employe KINGa la profession PRESIDENT dans le dept ACCOUNTING
l employe TURNERa la profession SALESMAN dans le dept SALES
l employe JAMESa la profession CLERK dans le dept SALES
l employe FORDa la profession ANALYST dans le dept RESEARCH
l employe MILLERa la profession CLERK dans le dept ACCOUNTING

PL/SQL procedure successfully completed.
```

2. Ecrire une fonction, prenant en paramètre le numéro de l'employé, qui permet d'afficher et de retourner le nom de l'employé (l'affichage est de même forme que la question 1).

```
create or replace function get_emp(num in emp.empno%type)

    return emp.ename%type

    is

    v_ename emp.ename%type;

    begin

    select ename into v_ename from emp where empno=num;

    dbms_output.put_line('le nom de l employe: ' || v_ename);

    return (v_ename);
end;
/
```

Affichage:

```
SQL> set serveroutput on;
SQL> declare
  2  emp_name varchar(30);
  3  begin
  4  emp_name:=get_emp(7521);
  5  dbms_output.put_line(emp_name);
  6  end;
  7  /
WARD

PL/SQL procedure successfully completed.
```

Exercice 3:

Ecrire une fonction prenant en paramètre le numéro de l'employé et retournant le nom de son chef. S'il n'a pas du chef, la fonction va retourner "Aucun".

NB: utiliser la question 2 de l'exercice 2.

```
create or replace function get_chef(num emp.empno%type)
    return emp.ename%type

    is

    v_chef emp.mgr%type;

    begin

    select mgr into v_chef from emp where empno=num;

    if v_chef is null then
        return ('aucun');
    else
        return (get_emp(v_chef));
    end if;
end;
/
```

Affichage :

```
SQL> set serveroutput on;
SQL> declare
  2  chef varchar(30);
  3  begin
  4  chef:=get_chef(7566);
  5  dbms_output.put_line(chef);
  6  end;
  7  /
KING

PL/SQL procedure successfully completed.
```

Exercice 4:

Ecrire une procédure qui permet d'augmenter les salaires (sans tenir compte la commission) des employés selon leurs grades.

- 10% pour le grade 1 et grade 2.
- 15% pour le grade 3.
- 20% pour le grade 4 et grade 5.

```
create or replace procedure aug_sal
    is
    cursor curseur is select e.empno,e.sal,s.grade from emp e,salgrade s where e.sal between s.losal and s.hisal;

    v_sal emp.sal%type;
    v_grade salgrade.grade%type;
    v_empno emp.empno%type;

    begin
    open curseur;
    fetch curseur into v_empno,v_sal,v_grade;
    while curseur%found loop
        if (v_grade=1 or v_grade=2) then
            update emp set sal=v_sal*1.1 where empno=v_empno;
        elsif (v_grade=3) then
            update emp set sal=v_sal*1.15 where empno=v_empno;
        else
            update emp set sal=v_sal*1.2 where empno=v_empno;
        end if;
    fetch curseur into v_empno,v_sal,v_grade;
    end loop;
    close curseur;
end;
/
```

Affichage :

```
SQL> set serveroutput on;
SQL> begin
  2  aug_sal;
  3  end;
  4  /

PL/SQL procedure successfully completed.
```

Exercice 5:

Ecrire un package « gestion_emp » permettant de regrouper les procédures et les fonctions créées dans les quatre exercices précédents.

```
SQL> create or replace package gestion_emp
  2      is
  3      function get_dept(num IN dept.deptno%type) return dept.dname%type;
  4
  5      procedure affiche_dept(num IN dept.deptno%type,nom OUT dept.dname%type);

  6
  7      procedure affiche_emp;
  8
  9      function get_emp(num in emp.empno%type) return emp.ename%type;
 10
 11      function get_chef(num emp.empno%type) return emp.ename%type;
 12
 13      procedure aug_sal;
 14  end gestion_emp;
 15  /

Package created.
```

FIN.