


Exercice 1 :

 ex1.txt - Bloc-notes

Fichier Edition Format Affichage Aide

ACCEPT numEmploye PROMPT 'Donner le numero d''employe : '

ACCEPT pourcentage PROMPT 'Donner le pourcentage d''augmentation : '

DECLARE

nomEmploye emp.ENAME%type;

salaire emp.SAL%type;

nvSalaire emp.SAL%type;

BEGIN

SELECT ENAME , SAL INTO nomEmploye , salaire FROM emp WHERE EMPNO=&numEmploye;

nvSalaire := salaire + (salaire * (&pourcentage*0.5));

DBMS_OUTPUT.PUT_LINE('L''employé : ' || nomEmploye || ', Ancien salaire : ' || salaire || ', Nouveau salaire : ' || nvSalaire);

END;

/


- Résultat d'exécution :

```
SQL> @C:\Users\WIN\Desktop\Tps\tp1\ex1.txt
Donner le numero d'employe : 7698
Donner le pourcentage d'augmentation : 10
old 7:      SELECT ENAME , SAL INTO nomEmploye , salaire FROM emp WHERE EMPNO=&numEmploye;
new 7:      SELECT ENAME , SAL INTO nomEmploye , salaire FROM emp WHERE EMPNO=7698;
old 8:      nvSalaire := salaire + (salaire * (&pourcentage*0.5));
new 8:      nvSalaire := salaire + (salaire * (10*0.5));
L'employé® : BLAKE, Ancien salaire : 2850, Nouveau salaire : 17100

PL/SQL procedure successfully completed.
```

Exercice 2 :

Q1 :

 ex2.txt - Bloc-notes

Fichier Edition Format Affichage Aide

DECLARE

numDept dept.DEPTNO%type := 50;

nomDept dept.DNAME%type := 'COMPTABILITÉ';

locDept dept.LOC%type := 'Paris';

BEGIN

INSERT INTO DEPT VALUES (numDept, nomDept, locDept);

END ;|

/

- Résultat d'exécution :

```
SQL> @C:\Users\WIN\Desktop\Tps\tp1\ex2.txt  
  
PL/SQL procedure successfully completed.  
  
SQL>
```

Q2 :

ex2-q2.txt - Bloc-notes

Fichier Edition Format Affichage Aide

```
DECLARE  
    dept_sup DEPT%ROWTYPE;  
  
BEGIN  
    SELECT * INTO dept_sup FROM DEPT WHERE DEPT.DEPTNO=50;  
    DELETE FROM DEPT WHERE DEPTNO=50;  
    DBMS_OUTPUT.PUT_LINE('Le departement supprimé : ' || dept_sup.dname || ', a le numéro : ' || dept_sup.deptno || ', et la localisation : ' || dept_sup.loc);  
  
END;  
/
```

- Résultat d'exécution :

```
SQL> @C:\Users\WIN\Desktop\Tps\tp1\ex2-q2.txt  
Le departement supprimé : COMPTABILITE , a le numéro : 50 , et la  
localisation : Paris  
  
PL/SQL procedure successfully completed.
```

Exercice 3 :

Q1 :

```
SQL> select tablespace_name from dba_tablespaces where tablespace_name = 'USERS';  
  
TABLESPACE_NAME  
-----  
USERS
```

Q2 :

```
SQL> create user utilisateur1 identified by utilisateur1 default tablespace users;  
  
User created.
```

Q3:

```
SQL> Grant create session to utilisateur1;

Grant succeeded.

SQL> Grant create table to utilisateur1;

Grant succeeded.

SQL> connect utilisateur1/utilisateur1
Connected.
```

```
SQL> CREATE TABLE dep (
2   ndep          INT PRIMARY KEY,
3   libelle       VARCHAR(50),
4   local         VARCHAR(50),
5   nbre_empls    INT,
6   budget        DECIMAL(10, 2)
7 );

Table created.

SQL> CREATE TABLE empl (
2   nemp          INT PRIMARY KEY,
3   nom           VARCHAR(50),
4   prenom        VARCHAR(50),
5   adresse       VARCHAR(100),
6   salaire       DECIMAL(10, 2),
7   ndept         INT,
8   CONSTRAINT fk_emp_dep FOREIGN KEY (ndept) REFERENCES dep(ndep)
9 );

Table created.
```

```
SQL> INSERT INTO dep (ndep, libelle, local, nbre_empls, budget) VALUES (1, 'Informatique', 'Paris', 10, 50000.00);

1 row created.

SQL> INSERT INTO dep (ndep, libelle, local, nbre_empls, budget) VALUES (2, 'Comptabilité', 'Lyon', 8, 30000.00);

1 row created.
```

```
SQL> INSERT INTO empl (nemp, nom, prenom, adresse, salaire, ndept) VALUES (101, 'Dupont', 'Jean', '12 rue de la Paix, Paris', 2500.00, 1);

1 row created.


SQL> INSERT INTO empl (nemp, nom, prenom, adresse, salaire, ndept)VALUES (102, 'Martin', 'Claire', '5 avenue des Champs, Lyon', 2800.00, 2);

1 row created.

SQL> INSERT INTO empl (nemp, nom, prenom, adresse, salaire, ndept)VALUES (103, 'Bernard', 'Luc', '3 boulevard Saint-Germain, Paris', 3000.00, 1);

1 row created.
```

Q4 :

 ex3-q4.txt - Bloc-notes

Fichier Edition Format Affichage Aide

```
DECLARE
    maxSalaire empl.salaire%type;

BEGIN
    SELECT MAX(salaire) INTO maxSalaire FROM empl;
    DBMS_OUTPUT.PUT_LINE('Le salaire maximal est: ' || maxSalaire);

END;
/
```


- Résultat d'exécution :

```
SQL> @C:\Users\WIN\Desktop\Tps\tp1\ex3-q4.txt
Le salaire maximal est: 3000

PL/SQL procedure successfully completed.
```

Exercice 4 :

Q1 & Q2 :

 ex4-q1.txt - Bloc-notes

Fichier Edition Format Affichage Aide

```
VARIABLE maxSalaire NUMBER;

BEGIN
    SELECT MAX(salaire) INTO :maxSalaire FROM empl;

END;
/
```

- Résultat d'exécution :


```
SQL> @C:\Users\WIN\Desktop\Tps\tp1\ex4.txt

PL/SQL procedure successfully completed.

SQL> print maxSalaire

MAXSALAIRE
-----
          3000
```

Exercice 5 :

 ex5.txt - Bloc-notes

Fichier Edition Format Affichage Aide

```
DECLARE
TYPE type_table IS TABLE OF VARCHAR2(60);
table1 type_table := type_table('AERIS','Air France','Air Littoral','Regional');
table2 type_table := type_table('ALITALIA','Quantas','SABENA');
table_fusion type_table := type_table();
v_temp VARCHAR2(60);
v_index INTEGER := 1; -- Index pour la boucle
v_fusion_index INTEGER := 1; -- Index pour la fusion

BEGIN
-- Tri par ordre croissant de table1
WHILE v_index <= table1.COUNT - 1 LOOP
FOR i IN 1 .. table1.COUNT - v_index LOOP
IF table1(i) > table1(i + 1) THEN
-- Échange des éléments
v_temp := table1(i);
table1(i) := table1(i + 1);
table1(i + 1) := v_temp;
END IF;
END LOOP;
v_index := v_index + 1;
END LOOP;

-- Tri par ordre croissant de table2
v_index := 1;
WHILE v_index <= table2.COUNT - 1 LOOP
FOR i IN 1 .. table2.COUNT - v_index LOOP
IF table2(i) > table2(i + 1) THEN
-- Échange des éléments
v_temp := table2(i);
table2(i) := table2(i + 1);
table2(i + 1) := v_temp;
END IF;
END LOOP;
v_index := v_index + 1;
END LOOP;

-- La fusion de table1 et table2 dans table_fusion
table_fusion.EXTEND(table1.COUNT + table2.COUNT); -- Étendre table_fusion pour accueillir les deux tables

-- Ajout des éléments de table1 à table_fusion
FOR i IN 1 .. table1.COUNT LOOP
table_fusion(v_fusion_index) := table1(i);
v_fusion_index := v_fusion_index + 1;
END LOOP;

-- Ajout des éléments de table2 à table_fusion
FOR i IN 1 .. table2.COUNT LOOP
table_fusion(v_fusion_index) := table2(i);
v_fusion_index := v_fusion_index + 1;
END LOOP;

-- Tri de table_fusion par ordre croissant
v_index := 1;
WHILE v_index <= table_fusion.COUNT - 1 LOOP
FOR i IN 1 .. table_fusion.COUNT - v_index LOOP
IF table_fusion(i) > table_fusion(i + 1) THEN
-- Échange des éléments
v_temp := table_fusion(i);
table_fusion(i) := table_fusion(i + 1);
table_fusion(i + 1) := v_temp;
END IF;
END LOOP;
v_index := v_index + 1;
END LOOP;

-- Affichage des éléments de table_fusion
DBMS_OUTPUT.PUT_LINE('Le nouveau tableau : ');

FOR j IN 1 .. table_fusion.COUNT LOOP
DBMS_OUTPUT.PUT_LINE(table_fusion(j));
END LOOP;

DBMS_OUTPUT.PUT_LINE('La taille du nouveau tableau : ' || table_fusion.COUNT);

END;
```

/

- Résultat d'exécution :

```
SQL> @C:\Users\WIN\Desktop\Tps\tp1\ex5.txt
Le nouveau tableau :
AERIS
ALITALIA
Air France
Air Littoral
Quantas
Regional
SABENA
La taille du nouveau tableau : 7

PL/SQL procedure successfully completed.
```