



Master Internet des Objets et Systèmes Mobiles
2021-2022

Module: S.I et Bases de données

Rapport des Travaux Pratiques
Série n° 3

Réalisé par :

ZAROUAL Mohammed

Encadré par :

Pr. EL AKKAD Nabil

On dispose d'une table BUDGET qui a été créée de la manière suivante :

```
CREATE TABLE BUDGET
( NUM_OPERATION  NUMBER (4,2) PRIMARY KEY,
  NOM_OPERATION   VARCHAR2 (15),
  CATEGORIE       VARCHAR2 (10),
  DATE_OPERATION  DATE,
  MONTANT         NUMBER (6, 2),
  SOLDE          NUMBER (7, 2) );
```

Remarque : les montants des opérations effectuées ainsi que les soldes sont exprimés en Dirhams.

```
SQL> select * from budget;
```

NUM_OPERATION	NOM_OPERATION	CATEGORIE	DATE_OPE	MONTANT	SOLDE
1	Divers	Debit	15/10/01	220	5500
2	Courses	Debit	16/10/01	455	5045
3	Virement	Debit	26/10/01	2000	3045
4	Salaire	Credit	27/10/01	8500	11545
5	Cadeau	Debit	29/10/01	350	11195
6	Loyer	Debit	03/11/01	3000	8195
7	Remboursement	Debit	10/11/01	2500	5695
8	Courses	Debit	15/11/01	670	5025
9	Divers	Debit	20/11/01	300	4725
10	Sortie	Debit	24/11/01	600	4125
11	Courses	Debit	26/11/01	555	3570
12	Virement	Debit	28/11/01	2000	1570
13	Salaire	Credit	27/11/01	8500	10070
14	Facture	Debit	02/12/01	1000	9070
15	Courses	Debit	05/12/01	750	8320
16	Loyer	Debit	06/12/01	3000	5320
17	Cadeau	Debit	10/12/01	500	4820
18	Cadeau	Debit	11/12/01	700	4120
19	Sortie	Debit	15/12/01	500	3620
20	Courses	Debit	19/12/01	660	2960
21	Cadeau	Debit	21/12/01	350	2610
22	Facture	Debit	22/12/01	550	2060
23	Virement	Debit	26/12/01	2000	60
24	Salaire	Credit	27/12/01	8500	8560
25	Courses	Debit	30/12/01	1100	7460
26	Cadeau	Credit	03/01/02	750	8210
27	Sortie	Debit	04/01/02	460	7750
28	Courses	Debit	06/01/02	630	7120
29	Divers	Debit	10/01/02	700	6420
30	Facture	Debit	13/01/02	890	5530

```
30 rows selected.
```

Exercice 1 :

On souhaite insérer un nouvel enregistrement dans la table budget sans avoir à saisir le numéro de l'opération et le nouveau solde. Écrire un programme PL/SQL qui ajoute un nouvel enregistrement à la table BUDGET avec les valeurs suivantes :

- Num_Operation : calculé automatiquement en ajoutant 1 à la plus grande valeur de Num_Operation présent dans la table BUDGET.
- Nom_Operation : Courses
- Categorie : Débit
- Date_Operation : 14/01/2002
- Montant : 500 (en dirhams)
- Solde : calculé automatiquement en fonction du dernier solde de la table BUDGET (celui pour lequel Num_Operation est le plus grand) : « par exemple, le dernier solde -500 ».

PL/SQL :

```
declare
vNum_operation budget.num_operation%type;
v_solde budget.solde%type;
begin
select max(num_operation) into vNum_operation from budget;
select solde into v_solde from budget where num_operation=vNum_operation;
insert into budget values (vNum_operation+1,'courses','Debit','14/01/2002',500,v_solde-500);
end;
/
```

L’Affichage :

22	Facture	Débit	22/12/01	550	2000
NUM_OPERATION	NOM_OPERATION	CATEGORIE	DATE_OPE	MONTANT	SOLDE
23	Virement	Débit	26/12/01	2000	60
24	Salaire	Crédit	27/12/01	8500	8560
25	Courses	Débit	30/12/01	1100	7460
26	Cadeau	Crédit	03/01/02	750	8210
27	Sortie	Débit	04/01/02	460	7750
28	Courses	Débit	06/01/02	630	7120
29	Divers	Débit	10/01/02	700	6420
30	Facture	Débit	13/01/02	890	5530
31	courses	Debit	14/01/02	500	5030

Exercice 2 : Utilisation des curseurs

On souhaite à présent effectuer des traitements sur plusieurs colonnes en même temps. L'objectif de cet exercice est de construire une table **COURSES**, qui contient toutes les opérations pour lesquelles le nom de l'opération est 'Courses', à partir de la table **BUDGET**.

La table **COURSES** doit contenir les champs suivants : Num_operation, Nom_operation, Categorie, Date_operation et Montant.

Pour cela, il faut définir un curseur qui va parcourir toute la table « **BUDGET** » et insérer tous ses enregistrements dans la table « **Courses** ».

PL/SQL :

```
CREATE TABLE COURSES ( NUM_OPERATION NUMBER (4) PRIMARY KEY,
                        NOM_OPERATION VARCHAR2 (15),
                        CATEGORIE VARCHAR2 (10),
                        DATE_OPERATION DATE,
                        MONTANT NUMBER (6, 2));

declare |

cursor c_courses is select num_operation,nom_operation,categorie,date_operation,montant
from budget where initcap(nom_operation)=initcap('Courses');

vnum_operation budget.num_operation%type;
vnom_operation budget.nom_operation%type;
vcategorie budget.categorie%type;
vdate_operation budget.date_operation%type;
vmontant budget.montant%type;
--var_courses courses%rowtype;

begin
    I
open c_courses;
fetch c_courses into vnum_operation,vnom_operation,vcategorie,vdate_operation,vmontant;

while c_courses %found loop
    insert into courses values (vnum_operation,vnom_operation,vcategorie,vdate_operation,vmontant);
    fetch c_courses into vnum_operation,vnom_operation,vcategorie,vdate_operation,vmontant;
end loop;
close c_courses;
end;
/
```

L’Affichage :

```
SQL> @"C:\Users\rozarot\Desktop\TP3\Exercice 2.sql.txt";
Table created.

PL/SQL procedure successfully completed.
SQL> select * from courses;

NUM_OPERATION NOM_OPERATION CATEGORIE DATE_OPE MONTANT
-----
          2 Courses      Debit      16/10/01      455
          8 Courses      Debit      15/11/01      670
         11 Courses      Debit      26/11/01      555
         15 Courses      Debit      05/12/01      750
         20 Courses      Debit      19/12/01      660
         25 Courses      Debit      30/12/01     1100
         28 Courses      Debit      06/01/02      630
         31 courses      Debit      14/01/02      500

8 rows selected.
```

Exercice 3 : Utilisation des curseurs

On souhaite créer une table BUDGET_EURO avec toutes les opérations réalisées depuis le passage à l'Euro (c'est-à-dire depuis le 01/01/2002) avec les montants des opérations exprimés en Euros et non plus en dirhams (le taux de conversion est 11).

Créez une nouvelle table BUDGET_EURO, définissez un curseur qui traite toutes les opérations réalisées depuis le 01/01/2002 et insérez dans la table BUDGET_EURO ces opérations avec le montant exprimé en Euros.

PL/SQL :

```
CREATE TABLE BUDGET_EURO ( NUM_OPERATION NUMBER (4) PRIMARY KEY,
  NOM_OPERATION VARCHAR2 (15),
  CATEGORIE VARCHAR2 (10),
  DATE_OPERATION DATE,
  MONTANT NUMBER (6, 2));

declare
cursor c_euro is select * from budget where date_operation>'01/01/2002';
new_budget budget_euro%rowtype;

begin
open c_euro;
fetch c_euro into new_budget;

while c_euro %found loop
  insert into budget_euro values (new_budget.num_operation,new_budget.nom_operation,new_budget.categorie,
    new_budget.date_operation,new_budget.montant*11,new_budget.solde);
  fetch c_euro into new_budget;

end loop;
close c_euro;
end;
```

L’Affichage :

```
SQL> select * from budget_euro;
```

NUM_OPERATION	NOM_OPERATION	CATEGORIE	DATE_OPE	MONTANT	SOLDE
26	Cadeau	Crédit	03/01/02	8250	8210
27	Sortie	Débit	04/01/02	5060	7750
28	Courses	Débit	06/01/02	6930	7120
29	Divers	Débit	10/01/02	7700	6420
30	Facture	Débit	13/01/02	9790	5530
31	courses	Debit	14/01/02	5500	5030

Exercice 4 : Curseur implicite et paramétré.

Créer une table BUDGET_SEUIL(NUM_OPERATION, DATE_OPERATION, MONTANT) dont on enregistre tous les opérations de débit (catégorie de type débit) qui contiennent un montant dépassant un seuil (saisi par l'utilisateur), en utilisant :

- Un curseur implicite.
- Un curseur paramétré.

PL/SQL : Curseur implicite

```
CREATE TABLE BUDGET_SEUIL ( NUM_OPERATION          NUMBER (4) PRIMARY KEY,
                             DATE_OPERATION DATE,
                             MONTANT NUMBER (6, 2));

declare

cursor cursor_seuil is select num_operation,DATE_OPERATION,montant from budget where categorie='Debit' and montant>500;
new_budget budget_seuil%rowtype;

begin

open cursor_seuil;

fetch cursor_seuil into new_budget;

while cursor_seuil %found loop
    insert into budget_seuil values (new_budget.num_operation,new_budget.DATE_OPERATION,new_budget.montant);
    fetch cursor_seuil into new_budget;
end loop;
close cursor_seuil;
end;
```

PL/SQL : Curseur paramétré

```
CREATE TABLE BUDGET_SEUIL ( NUM_OPERATION          NUMBER (4) PRIMARY KEY,
                             DATE_OPERATION DATE,
                             MONTANT NUMBER (6, 2));

declare

cursor cursor_seuil(seuil Number) is select num_operation,DATE_OPERATION,montant from budget
where categorie='Debit' and montant>seuil;
new_budget budget_seuil%rowtype;

begin

open cursor_seuil(500);

fetch cursor_seuil into new_budget;

while cursor_seuil %found loop
    insert into budget_seuil values (new_budget.num_operation,new_budget.DATE_OPERATION,new_budget.montant);
    fetch cursor_seuil into new_budget;
end loop;
close cursor_seuil;
end;
```

L’Affichage :

```
SQL> @"C:\Users\rozarot\Desktop\TP3\Exercice 4_Implicite.sql.txt";
Table created.

PL/SQL procedure successfully completed.
SQL> select * from budget_seuil;
```

NUM_OPERATION	DATE_OPE	MONTANT
3	26/10/01	2000
6	03/11/01	3000
7	10/11/01	2500
8	15/11/01	670
10	24/11/01	600
11	26/11/01	555
12	26/11/01	2000
14	02/12/01	1000
15	05/12/01	750
16	06/12/01	3000
18	11/12/01	700

NUM_OPERATION	DATE_OPE	MONTANT
20	19/12/01	660
22	22/12/01	550
23	26/12/01	2000
25	30/12/01	1100
28	06/01/02	630
29	10/01/02	700
30	13/01/02	890

```
18 rows selected.
```

FIN.