

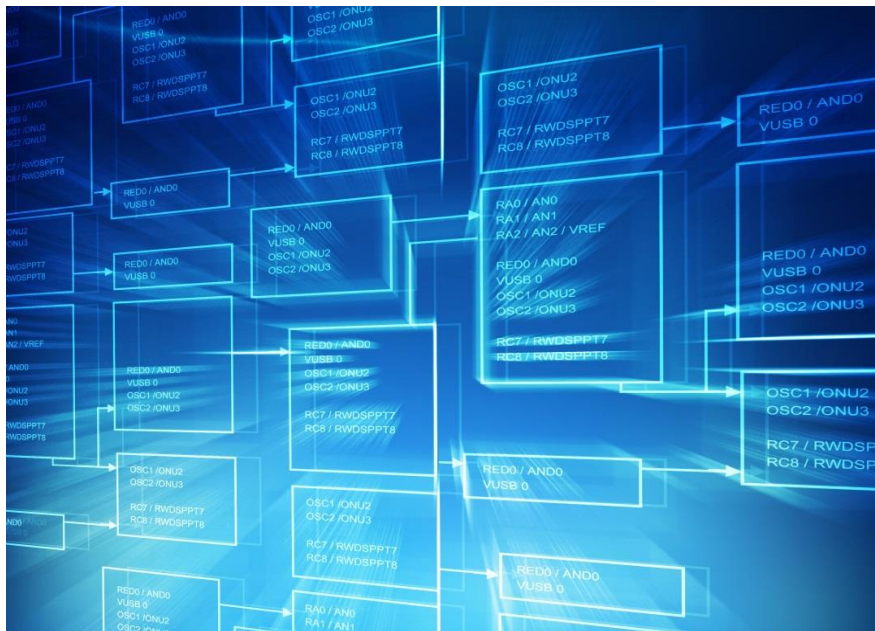
Département Informatique

MASTER EN SCIENCES ET TECHNIQUES



SYSTÈMES D'INFORMATION DÉCISIONNELS ET IMAGERIE

FACULTÉ DES SCIENCES ET TECHNIQUES ERRACHIDIA



Bases de données réparties TP3 - Curseurs

ZEKKOURI Hassan

Vendredi 06 Decembre 2019

Responsable du module :

Prof. BAATAOUI

REMERCIEMENT

Nous tenons à vous remercier monsieur **BAATAOUI** pour votre formation et vos services.

Nous sommes également reconnaissant de nous avoir donné l'occasion de s'ouvrir sur un nouvel aspect de technologie qui est le **PL/SQL** et y mettre en œuvre nos compétences dont nous avons obtenu au cours du module.

Hassan ZEKKOURI

PLAN

Contents

REMERCIEMENT	2
INTRODUCTION	4
I. EXERCICE 1.....	5
II. EXERCICE 2.....	7
III. EXERCICE 3.....	8
IV. EXERCICE 4.....	10
1. <i>Curseur implicite</i>	10
2. <i>Curseur explicite paramétré</i>	11
FIN.....	11

INTRODUCTION

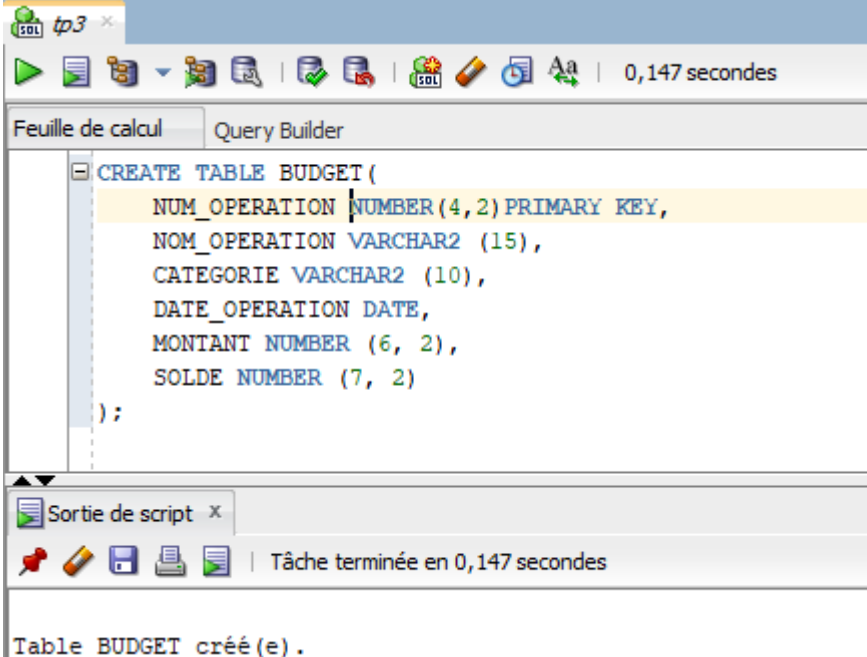
➤ Objectif :

Dans ce TP nous allons découvrir la programmation avec **PL/SQL** qui est une extension de langage **SQL** permettant d'interroger les bases de données efficacement !

En particulier, pendant ce TP nous allons découvrir les curseurs et leurs manipulations !

I. EXERCICE 1

ON crée la table **BUDGET** :

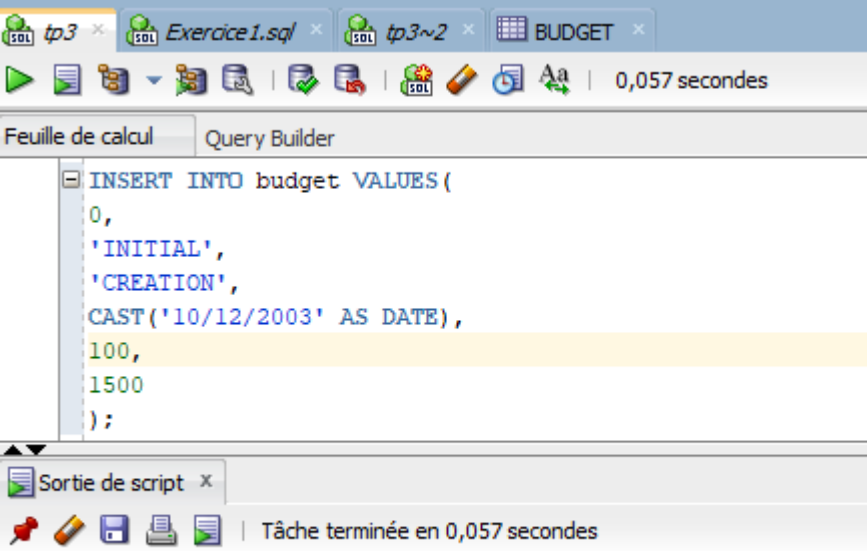


The screenshot shows the SQL Developer interface. The top toolbar includes icons for running queries, saving, and other database operations. The main window displays the 'Query Builder' tab with the following SQL code:

```
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)  
);
```

Below the code editor, the 'Sortie de script' (Script Output) window shows the message: 'Table BUDGET créé(e) .' (Table BUDGET created).

On insère le premier enregistrement :



The screenshot shows the SQL Developer interface with the 'BUDGET' table selected in the top toolbar. The main window displays the 'Query Builder' tab with the following SQL code:

```
INSERT INTO budget VALUES (  
  0,  
  'INITIAL',  
  'CREATION',  
  CAST('10/12/2003' AS DATE),  
  100,  
  1500  
);
```

Below the code editor, the 'Sortie de script' (Script Output) window shows the message: '1 ligne inséré.' (1 line inserted).

On va insérer un nouvel enregistrement dans la table budget sans avoir à saisir le numéro de l'opération et le nouveau solde.

```

DECLARE
AncienSolde Budget.Solde%TYPE;
NouveauSolde NUMBER(7, 2);
DernierNumero NUMBER(4, 2);
BEGIN
-- Récupérer le dernier solde
SELECT Solde INTO AncienSolde
FROM budget
WHERE Num_Operation IN (
    SELECT max(Num_Operation) FROM BUDGET
);
-- Récupérer le numéro de dernier ligne
SELECT max(Num_Operation) INTO DernierNumero
FROM budget;
-- Insertion
NouveauSolde := AncienSolde - 500;
INSERT INTO BUDGET VALUES (
    DernierNumero + 1,
    'Courses',
    'Débit',
    '14/01/2002',
    500,
    NouveauSolde);
END;
/

```

Sortie de script x

Tâche terminée en 0,069 secondes

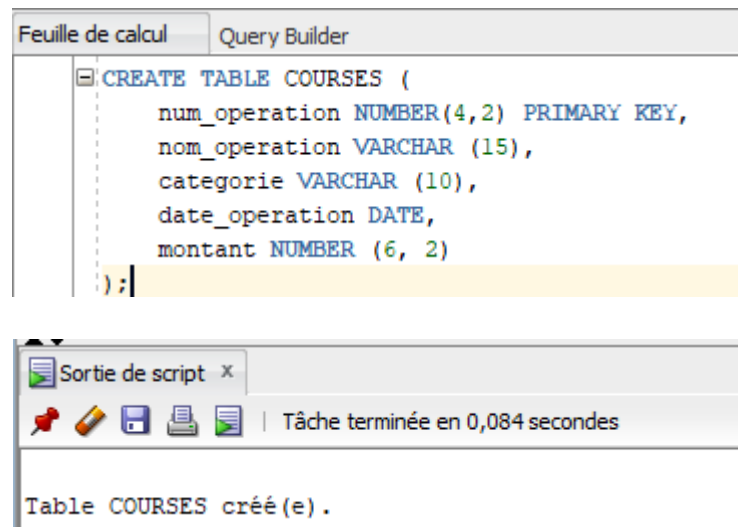
Procédure PL/SQL terminée.

	NUM_OPERATION	NOM_OPERATION	CATEGORIE	DATE_OPERATION	MONTANT	SOLDE
1	0	INITIAL	CREATION	10/12/03	100	1500
2	1	Courses	Débit	14/01/02	500	1000

Le script a effectué l'opération d'insertion avec succès !

II. EXERCICE 2

Création de la table **Courses** :

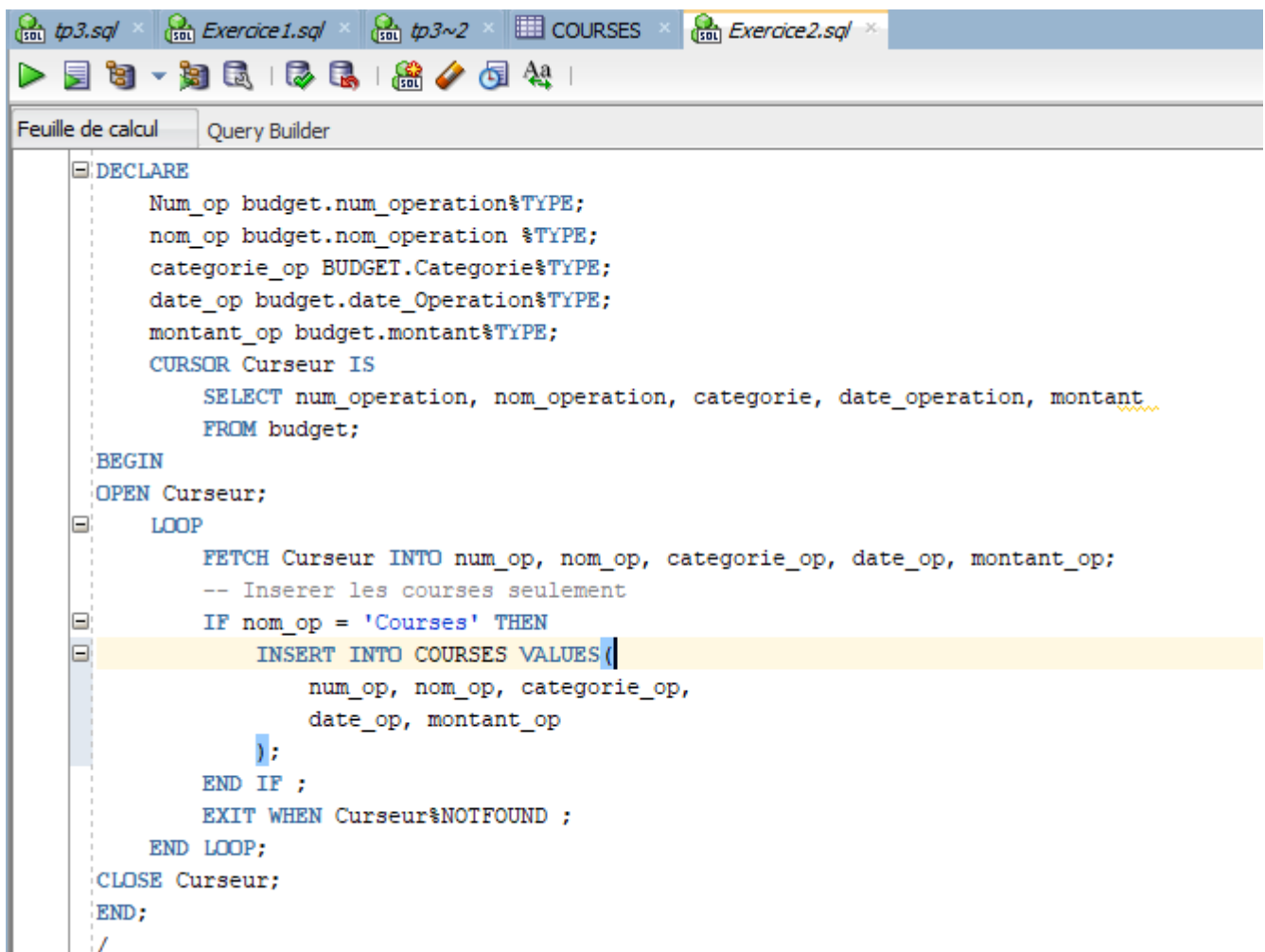


The screenshot shows a 'Query Builder' window with the following SQL code:

```
CREATE TABLE COURSES (
    num_operation NUMBER(4,2) PRIMARY KEY,
    nom_operation VARCHAR (15),
    categorie VARCHAR (10),
    date_operation DATE,
    montant NUMBER (6, 2)
);
```

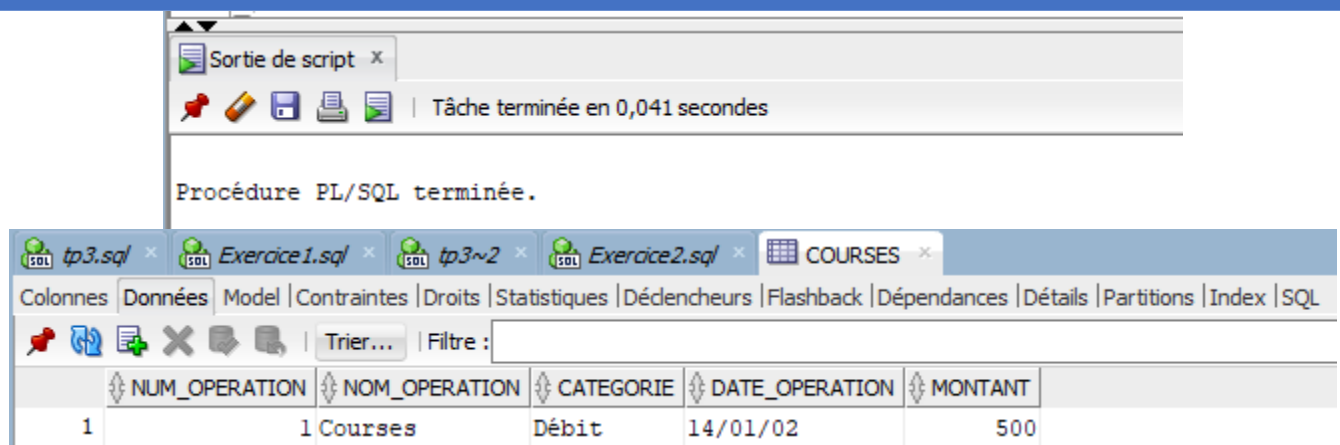
Below the code, a status bar indicates 'Tâche terminée en 0,084 secondes' and a message 'Table COURSES créé(e)'.

On va construire une table **COURSES**, qui contient toutes les opérations pour lesquelles le nom de l'opération est 'Courses', à partir de la table **BUDGET**.



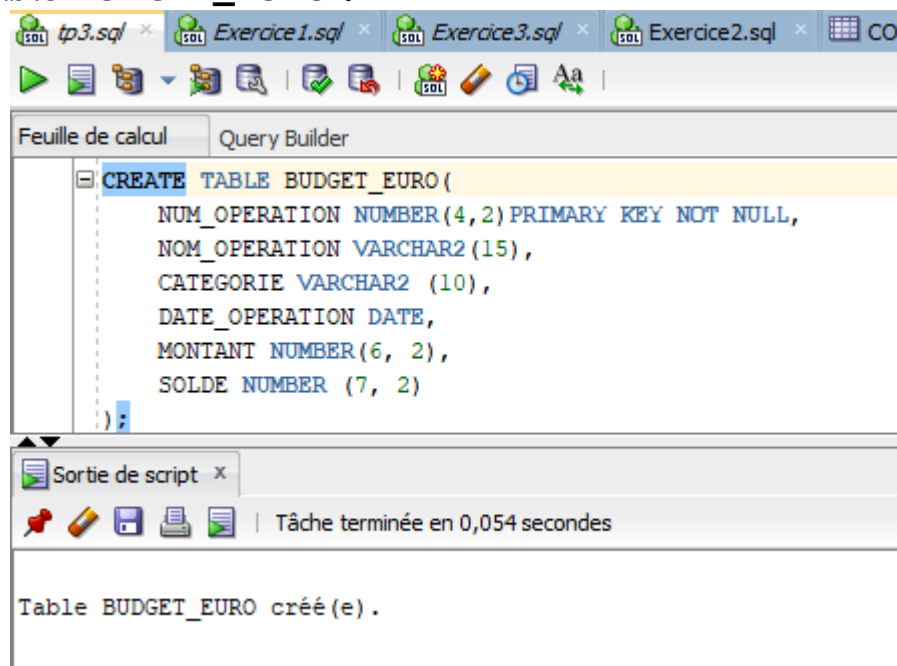
The screenshot shows a 'Query Builder' window with the following PL/SQL code:

```
DECLARE
    Num_op budget.num_operation%TYPE;
    nom_op budget.nom_operation %TYPE;
    categorie_op BUDGET.Categorie%TYPE;
    date_op budget.date_Operation%TYPE;
    montant_op budget.montant%TYPE;
    CURSOR Curseur IS
        SELECT num_operation, nom_operation, categorie, date_operation, montant
        FROM budget;
BEGIN
    OPEN Curseur;
    LOOP
        FETCH Curseur INTO num_op, nom_op, categorie_op, date_op, montant_op;
        -- Insérer les courses seulement
        IF nom_op = 'Courses' THEN
            INSERT INTO COURSES VALUES(
                num_op, nom_op, categorie_op,
                date_op, montant_op
            );
        END IF ;
        EXIT WHEN Curseur%NOTFOUND ;
    END LOOP;
    CLOSE Curseur;
END;
```



III. EXERCICE 3

Création de la table **BUDGET_EURO** :



On va remplir la 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).


```

DECLARE
  i NUMBER;
  Num_op BUDGET.Num_operation%TYPE := 0 ;
  nom_op BUDGET.nom_operation%TYPE ;
  Categorie_op BUDGET.Categorie%TYPE;
  Date_op BUDGET.Date_Operation%TYPE ;
  Montant_op BUDGET.Montant%TYPE := 500;
  Solde_op BUDGET.Solde%TYPE;
  CURSOR Curseur IS
    SELECT nom_operation, Categorie, Date_operation, Montant, Solde
    FROM BUDGET WHERE Date_operation > '01/01/2002';
BEGIN
  i := 0;
  OPEN Curseur;
  LOOP i := i+1;
    FETCH Curseur INTO nom_op, Categorie_op, Date_op, Montant_op, Solde_op;
    INSERT INTO BUDGET_EURO VALUES (
      Num_op, nom_op, Categorie_op,
      Date_op, Montant_op/11,
      Solde_op/11
    );
    Num_op := Num_op + 1 ;
    EXIT WHEN Curseur%NOTFOUND ;
  END LOOP;
  CLOSE Curseur;
END;

```

Sortie de script x

Tâche terminée en 0,057 secondes

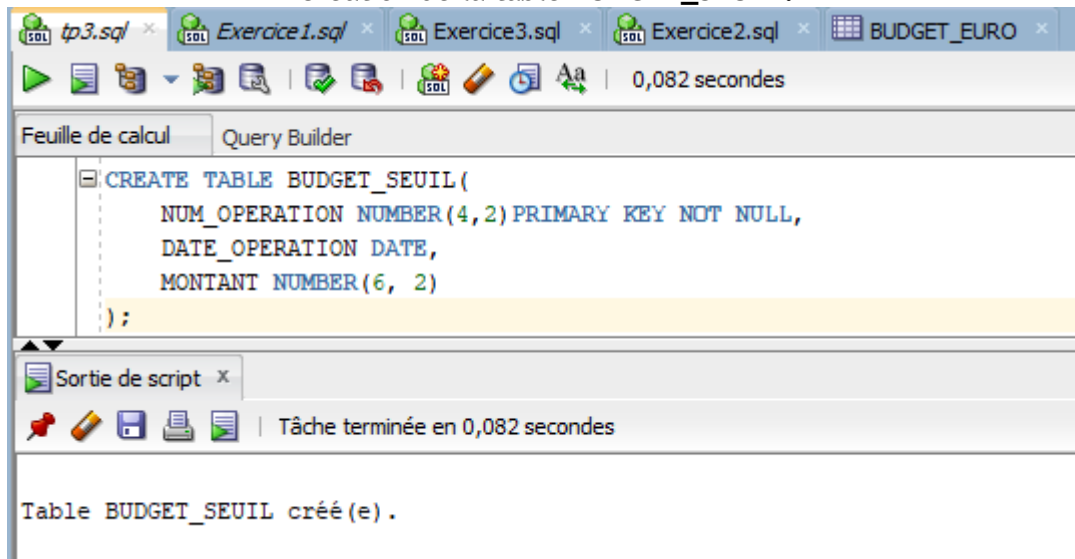
Procédure PL/SQL terminée.

	NUM_OPERATION	NOM_OPERATION	CATEGORIE	DATE_OPERATION	MONTANT	SOLDE
1	0	INITIAL	CREATION	10/12/03	9,09	136,36
2	1	Courses	Débit	14/01/02	45,45	90,91

Le passage à réussit !

IV. EXERCICE 4

Création de la table **BUDGET_SEUIL** :



1. Curseur implicite

