

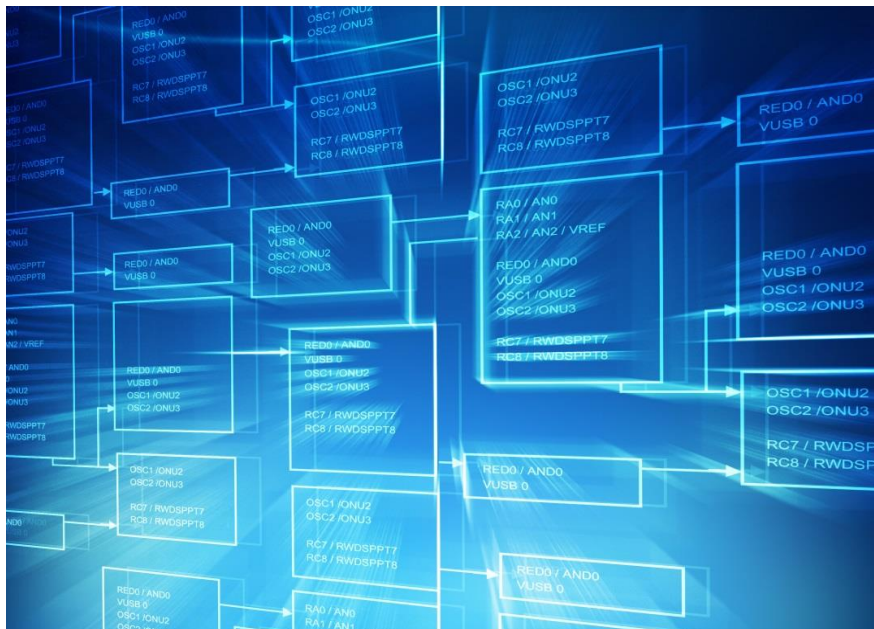
Département Informatique

MASTER EN SCIENCES ET TECHNIQUES



SYSTÈMES D'INFORMATION DÉCISIONNELS ET IMAGERIE

FACULTÉ DES SCIENCES ET TECHNIQUES ERRACHIDIA



TP2 - DEVELOPPEMENT AVEC PL/SQL

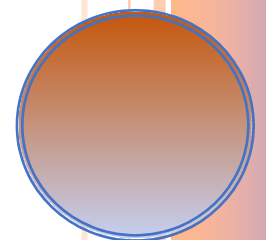
GESTION D'UN PARC DE VEHICLES (PARCVEH)

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

Contenu

REMERCIEMENT	2
INTRODUCTION	4
I. Création de la base.....	5
II. Remplissage des tables	7
1. On commence par la table VOITURE :	7
2. La table CH :	7
3. La table v_ch :	8
4. La table réparation :	8
5. La table Trajet :	9
6. La table tr_nov :	10
III. PL /SQL	11
1. Script1	11
2. Script2	12
3. Script3	12
4. Script4	13
5. Script5	14
6. Script6	16
7. Script7	17
8. Script8	18
9. Script9	19
10. Script10	21
FIN.....	22

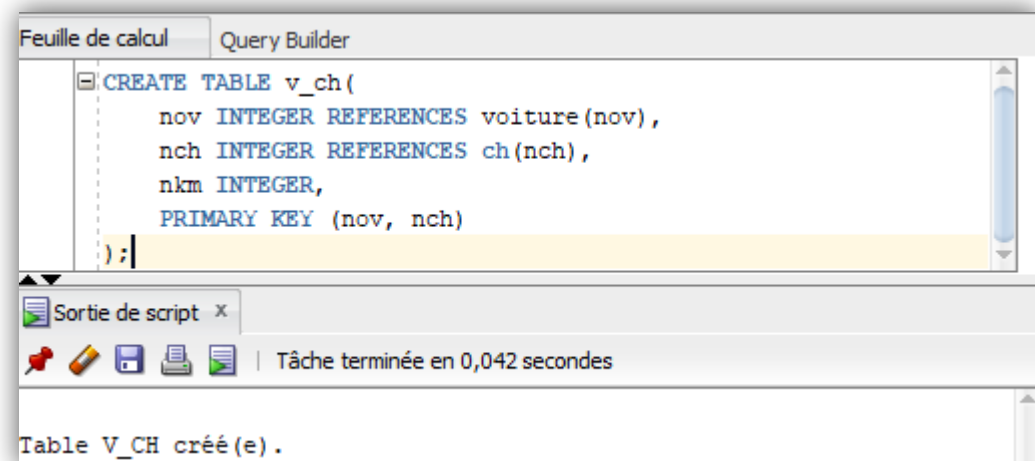
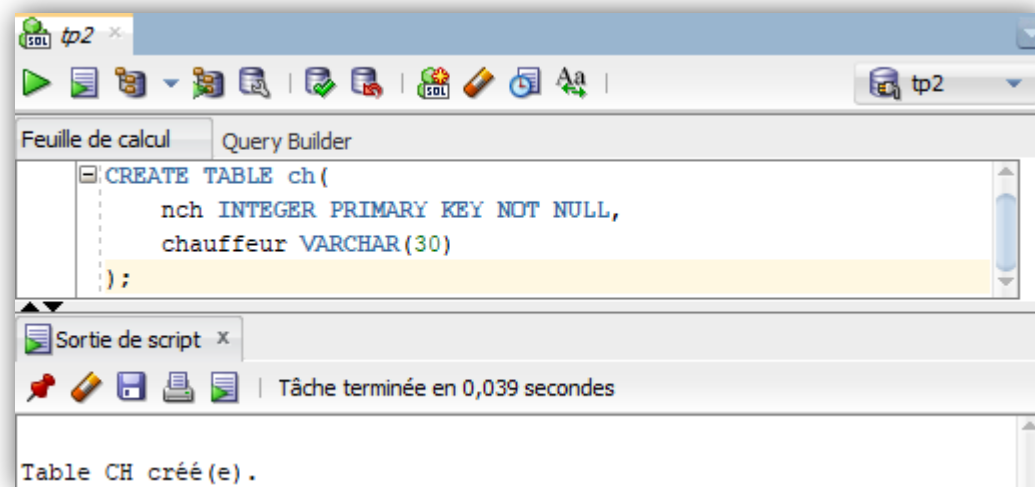
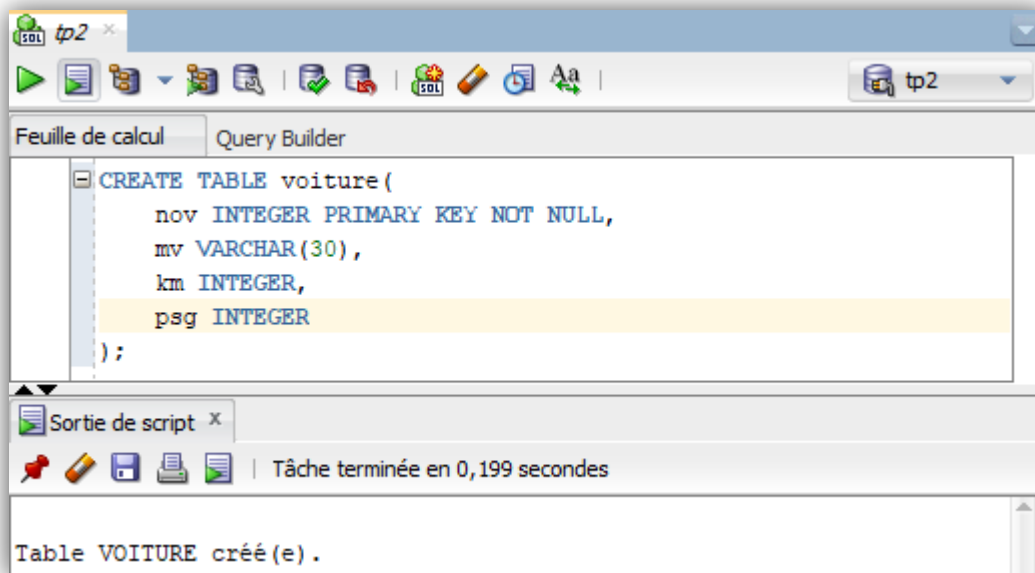
INTRODUCTION

➤ 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 !

I. Création de la base

Tout d'abord, nous allons créer le schéma de notre base (création des tables)!



The screenshot shows a 'Query Builder' window with a 'Feuille de calcul' tab. The SQL script in the editor is:

```
CREATE TABLE reparation(
  norep INTEGER PRIMARY KEY NOT NULL,
  nov INTEGER REFERENCES voiture(nov),
  nog INTEGER,
  typerep VARCHAR(30),
  px INTEGER,
  kmcpt INTEGER
);
```

Below the editor, a status bar indicates 'Tâche terminée en 0,049 secondes'. At the bottom, a message states 'Table REPARATION créé(e)'.

The screenshot shows a 'Query Builder' window with a 'Feuille de calcul' tab. The SQL script in the editor is:

```
CREATE TABLE Trajet(
  notraj INTEGER PRIMARY KEY NOT NULL,
  villearr VARCHAR(20),
  Villedep VARCHAR(20),
  dateTrajet DATE,
  nbkm INTEGER
);
```

Below the editor, a status bar indicates 'Tâche terminée en 0,039 secondes'. At the bottom, a message states 'Table TRAJET créé(e)'.

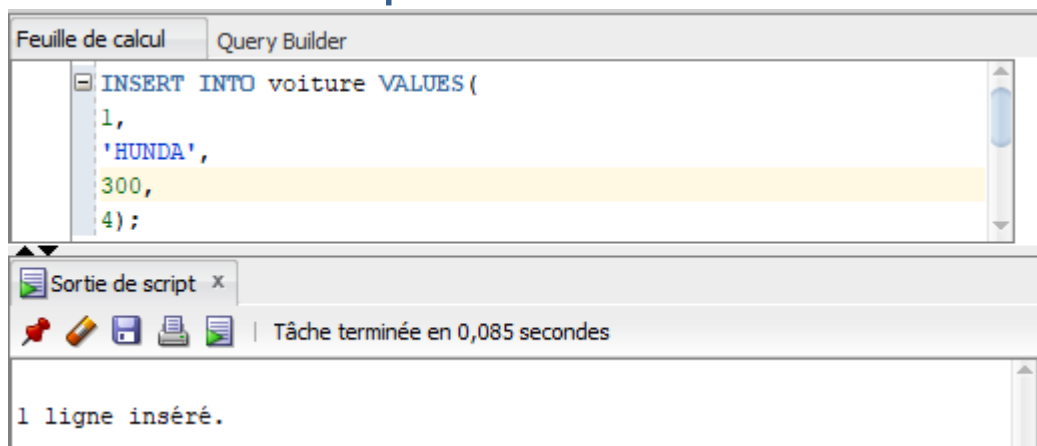
The screenshot shows a 'Query Builder' window with a 'Feuille de calcul' tab. The SQL script in the editor is:

```
CREATE TABLE tr_nov(
  notraj INTEGER NOT NULL,
  nov INTEGER NOT NULL,
  nch INTEGER NOT NULL,
  nbperstr INTEGER,
  FOREIGN KEY (nov) REFERENCES voiture(Nov),
  FOREIGN KEY (notraj) REFERENCES trajet(notraj),
  FOREIGN KEY (nch) REFERENCES ch(nch)
);
```

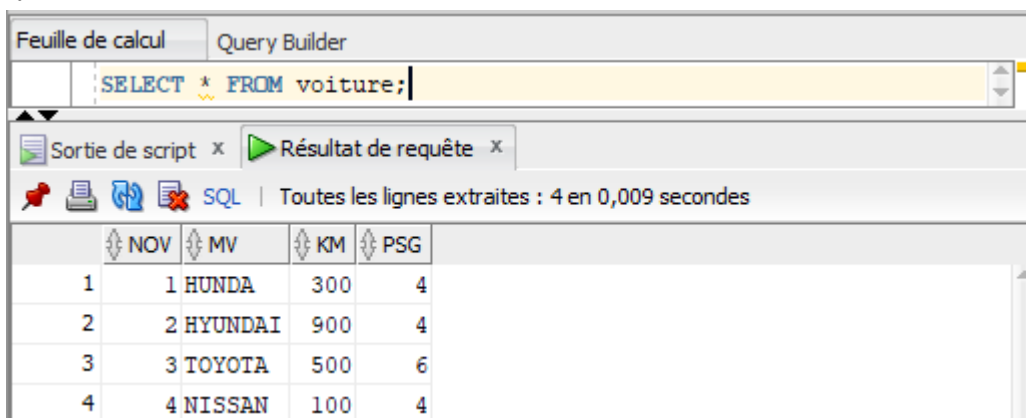
Maintenant, notre schéma est prêt à être exploiter ! Par la suite nous allons peupler chaque table avec des enregistrements !

II. Remplissage des tables

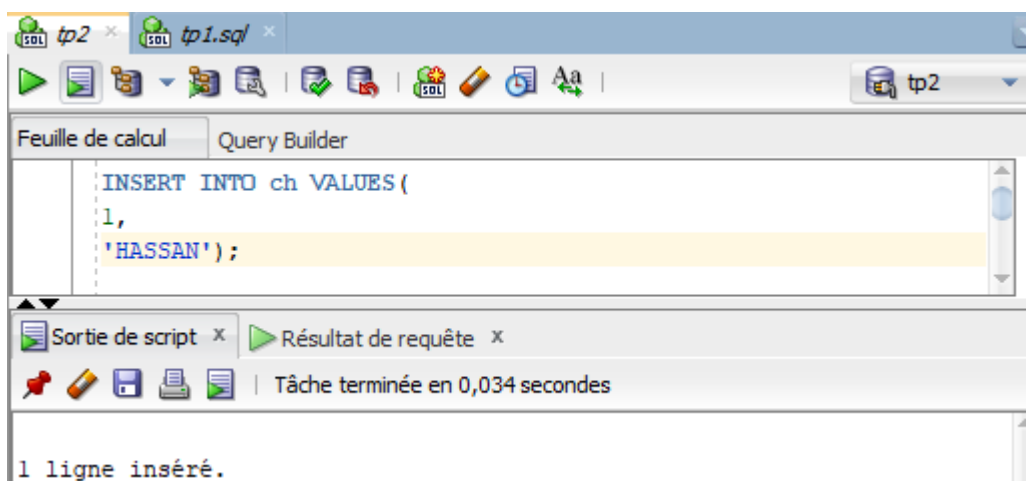
1. On commence par la table VOITURE :



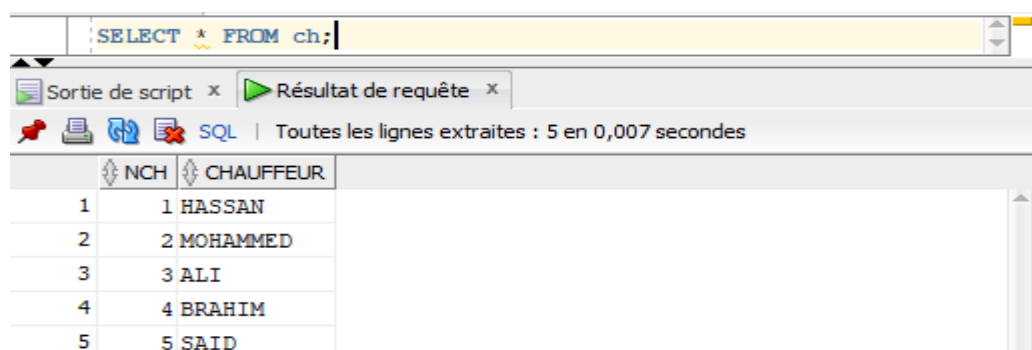
Visualisation :



2. La table CH :



Visualisation :



3. La table v_ch :

Feuille de calcul Query Builder

```
INSERT INTO v_ch VALUES (
1,
2,
1000);
```

Sortie de script x Résultat de requête x

Tâche terminée en 0,027 secondes

1 ligne inséré.

Feuille de calcul Query Builder

```
SELECT * FROM v_ch;
```

Sortie de script x Résultat de requête x

Toutes les lignes extraites : 6 en 0,005 secondes

	NOV	NCH	NKM
1	1	2	1000
2	2	1	1400
3	2	3	20000
4	3	1	2230
5	3	4	2530
6	4	4	2530

4. La table réparation :

tp2 tp1.sql x

Feuille de calcul Query Builder

```
INSERT INTO reparation VALUES (
2,
1,
1,
1,
'Portes',
100,
2123);
```

Sortie de script x Résultat de requête x

Tâche terminée en 0,025 secondes

1 ligne inséré.

Feuille de calcul Query Builder

```
SELECT
*
FROM reparation;
```

Sortie de script x Résultat de requête x

SQL | Toutes les lignes extraites : 6 en 0,005 secondes

	NOREP	NOV	NOG	TYPEREPA	PX	KMCPT
1	1	1	1	Roues	1234	2123
2	2	1	1	Portes	100	2123
3	3	2	2	Portes	100	2123
4	4	2	2	Roues	100	2123
5	5	3	3	Roues	200	2000
6	6	3	3	Portes	200	2000

5. La table Trajet :

tp2 x tp1.sql x

Feuille de calcul Query Builder

```
INSERT INTO trajet VALUES(
1,
'ERRACHIDIA',
'ALNIF',
'16/11/2019',
400);
```

Sortie de script x Résultat de requête x

Tâche terminée en 0,039 secondes

1 ligne inséré.

Feuille de calcul Query Builder

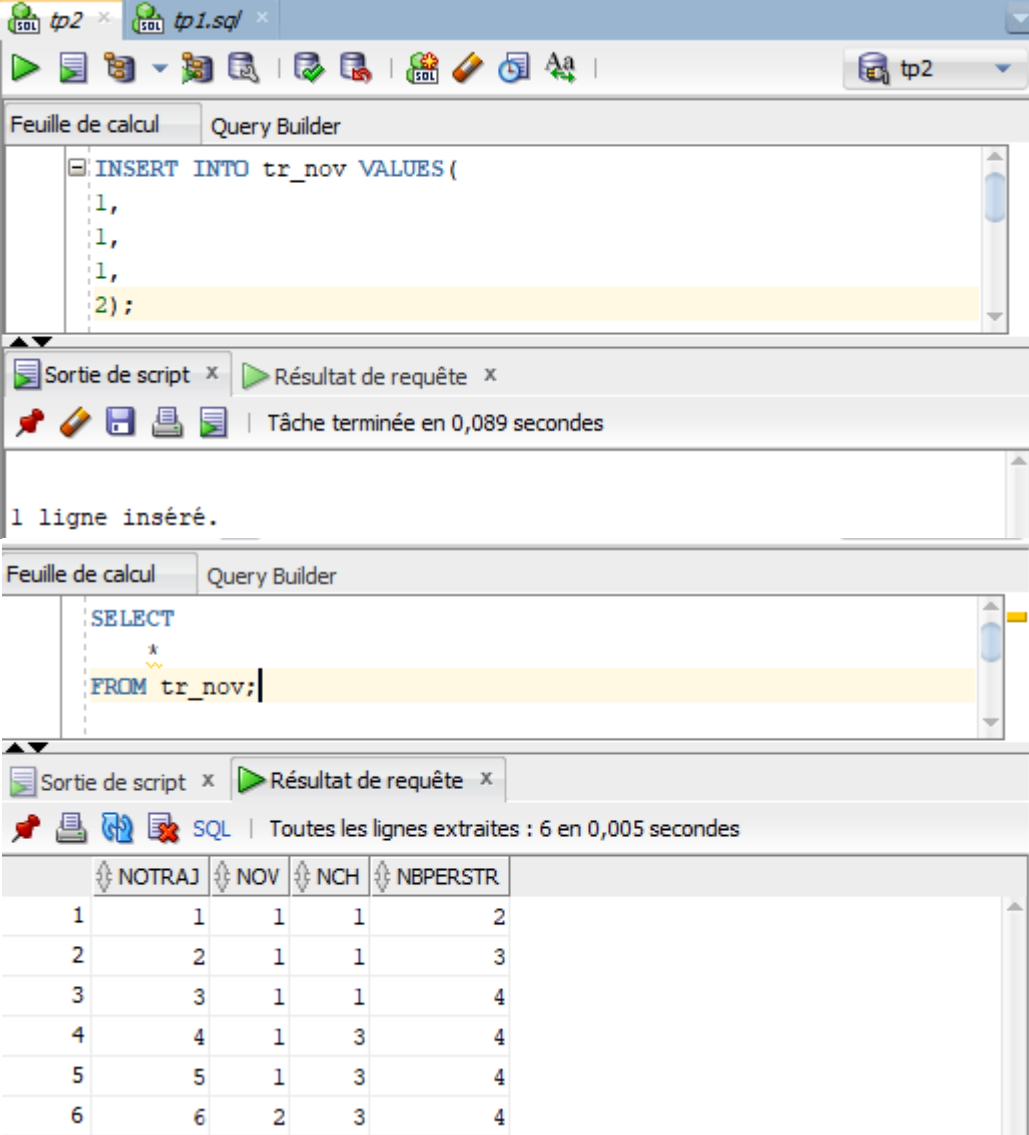
```
SELECT
*
FROM trajet;
```

Sortie de script x Résultat de requête x

SQL | Toutes les lignes extraites : 6 en 0,007 secondes

	NOTRAJ	VILLEARR	VILLEDEP	DATETRAJET	NBKM
1	1	ERRACHIDIA	ALNIF	16/11/19	400
2	2	ERRACHIDIA	MCISSI	16/11/19	370
3	3	NADOR	MCISSI	10/11/19	800
4	4	RABAT	MCISSI	10/11/19	650
5	5	RABAT	ALNIF	10/11/19	900
6	6	RABAT	KENITRA	10/12/19	80

6. La table tr_nov :



The screenshot shows the SQL Developer interface with two panels. The top panel displays the execution of an INSERT statement into the tr_nov table. The bottom panel shows the result of a SELECT query, displaying the data inserted into the table.

Top Panel: Query Execution

```
INSERT INTO tr_nov VALUES (
1,
1,
1,
2);
```

Sortie de script x Résultat de requête x
Tâche terminée en 0,089 secondes

1 ligne inséré.

Bottom Panel: Query Result

```
SELECT
*
FROM tr_nov;
```

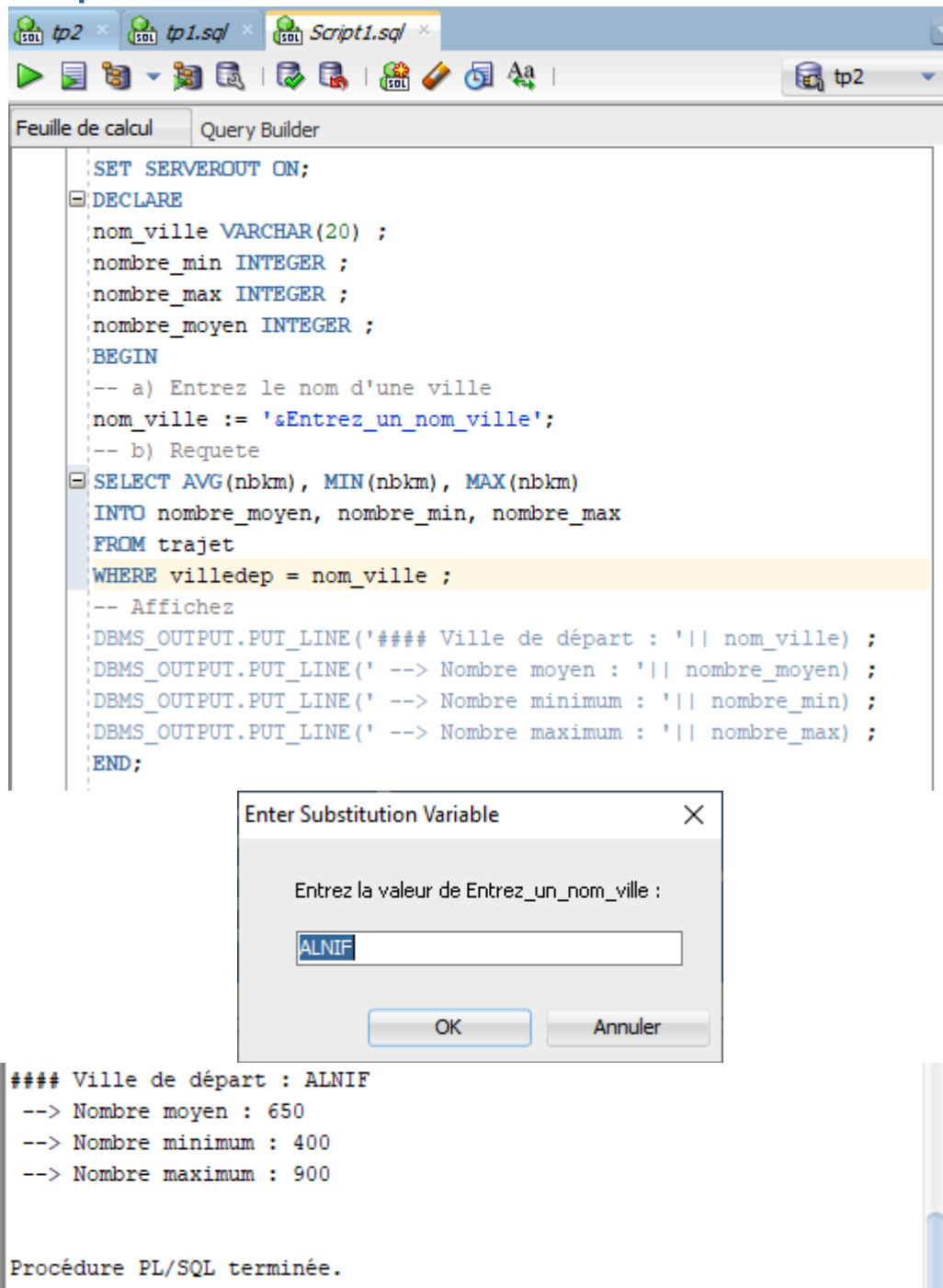
Sortie de script x Résultat de requête x
Toutes les lignes extraites : 6 en 0,005 secondes

	NOTRAJ	NOV	NCH	NBPERSTR
1	1	1	1	2
2	2	1	1	3
3	3	1	1	4
4	4	1	3	4
5	5	1	3	4
6	6	2	3	4

Par la suite nous allons programmer des blocs PL/SQL.

III. PL /SQL

1. Script1



```

SET SERVEROUT ON;
DECLARE
nom_ville VARCHAR(20) ;
nombre_min INTEGER ;
nombre_max INTEGER ;
nombre_moyen INTEGER ;
BEGIN
-- a) Entrez le nom d'une ville
nom_ville := '&Entrez_un_nom_ville';
-- b) Requete
SELECT AVG(nbkkm), MIN(nbkkm), MAX(nbkkm)
INTO nombre_moyen, nombre_min, nombre_max
FROM trajet
WHERE villedep = nom_ville ;
-- Affichez
DBMS_OUTPUT.PUT_LINE('#### Ville de départ : '|| nom_ville) ;
DBMS_OUTPUT.PUT_LINE(' --> Nombre moyen : '|| nombre_moyen) ;
DBMS_OUTPUT.PUT_LINE(' --> Nombre minimum : '|| nombre_min) ;
DBMS_OUTPUT.PUT_LINE(' --> Nombre maximum : '|| nombre_max) ;
END;

```

Enter Substitution Variable

Entrez la valeur de Entrez_un_nom_ville :

ALNIF

OK Annuler

```

#### Ville de départ : ALNIF
--> Nombre moyen : 650
--> Nombre minimum : 400
--> Nombre maximum : 900

Procédure PL/SQL terminée.

```

Comme nous pouvons remarquer, le script affiche le nombre moyen, le nombre minimum, le nombre maximum de kilomètres des trajets qui sont partis d'une ville saisie.

2. Script2

The screenshot shows the Oracle SQL Developer interface. The top toolbar includes icons for running, saving, and other database operations. The main window displays a PL/SQL script named 'Script2.sql'. The script declares two variables: 'un_ans' of type VARCHAR(4) and 'm_nombre_trajets' of type INTEGER. It begins with a 'BEGIN' block containing two prompts: 'a) Entrez un ans' and 'b) Calculez le nombre de trajets'. The script then uses a SELECT statement to count the number of trips for the entered year. The output window shows the results of the script execution, including the year 2019 and the number of trips, 6.

```

DECLARE
un_ans VARCHAR(4);
m_nombre_trajets INTEGER;
BEGIN
-- a) Entrez un ans
un_ans := '&Entrez_un_ans';
-- b) Calculez le nombre de trajets
SELECT COUNT(notraj) INTO m_nombre_trajets
FROM trajet tr
WHERE to_char(datetrajet, 'YYYY') = un_ans;
DBMS_OUTPUT.PUT_LINE('L'annee : ' || un_ans);
DBMS_OUTPUT.PUT_LINE(' - le nombre de trajets : ' || m_nombre_trajets);
END ;
/

```

Sortie de script x

Tâche terminée en 2,861 secondes

END ;
L'annee : 2019
- le nombre de trajets : 6

Procédure PL/SQL terminée.

Comme nous pouvons remarquer, le script affiche aussi son nombre de trajets effectués pendant une année donnée.

3. Script3

The screenshot shows the Oracle SQL Developer interface. The main window displays a PL/SQL script named 'Script3.sql'. The script declares two variables: 'un_ans' of type VARCHAR(4) and 'un_jour' of type VARCHAR(2). It begins with a 'BEGIN' block containing two prompts: 'a) Entrez un ans' and 'b) Calculez le nombre de trajets'. The script then uses a SELECT statement to count the number of trips for the entered year. The output window shows the results of the script execution, including the year 2019 and the number of trips, 6.

```

DECLARE
un_ans VARCHAR(4);
un_jour VARCHAR(2);
BEGIN
un_ans := to_char(sysdate, 'YYYY');
un_jour := to_char(sysdate, 'D');
DBMS_OUTPUT.PUT_LINE('Aujourd'hui est ' ||
to_char(sysdate, 'DAY') || ', ' || un_jour || ' ' || to_char(sysdate,
|| ' ' || un_ans);
END ;
/

```

Sortie de script x

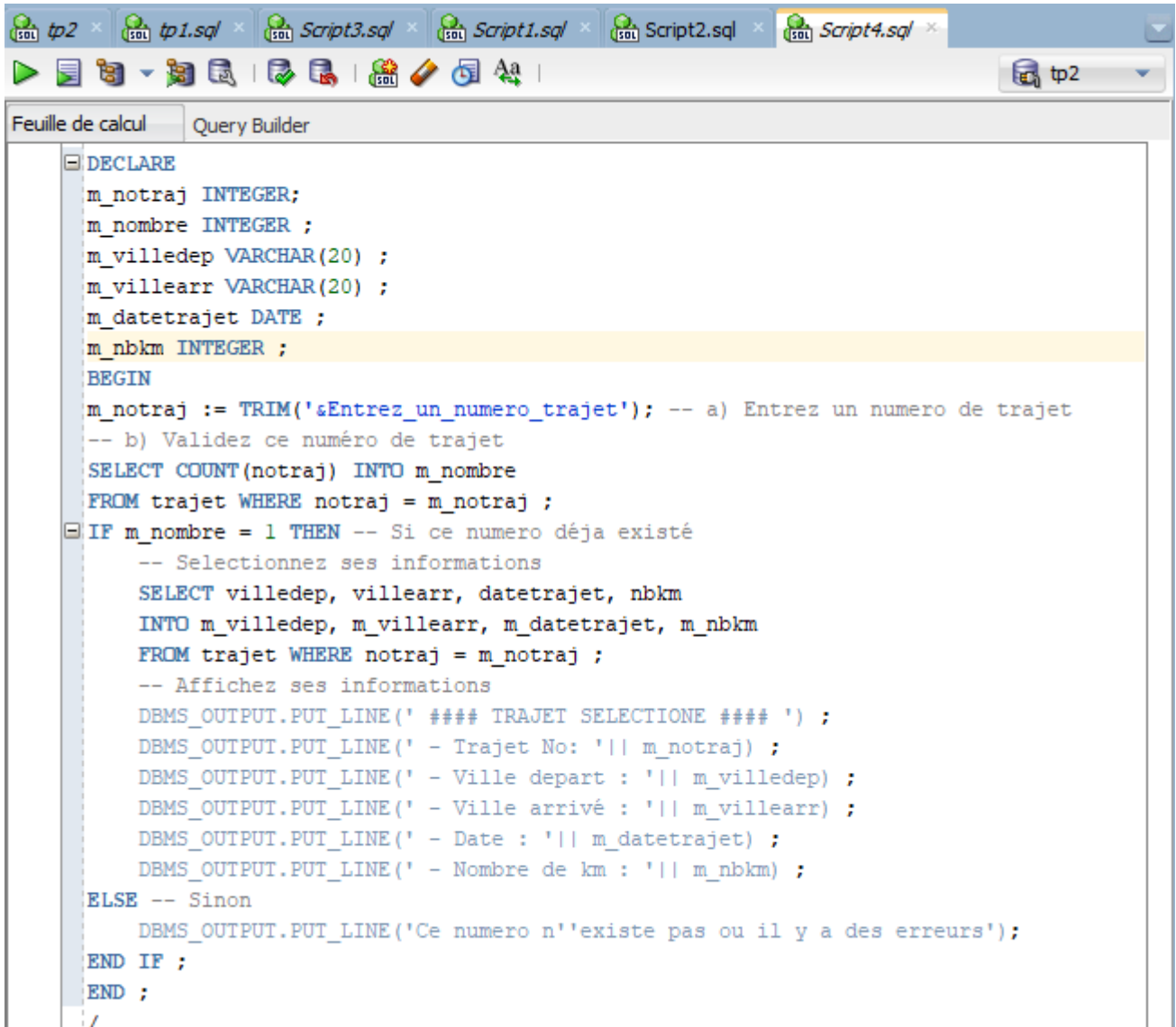
Tâche terminée en 0,036 secondes

Aujourd'hui est DIMANCHE, 7 DÉCEMBRE 2019

Procédure PL/SQL terminée.

Le script affiche des informations de la journée.

4. Script4



```

DECLARE
m_notraj INTEGER;
m_nombre INTEGER ;
m_villedep VARCHAR(20) ;
m_villearr VARCHAR(20) ;
m_datetrajet DATE ;
m_nbkm INTEGER ;
BEGIN
m_notraj := TRIM('&Entrez_un_numero_trajet'); -- a) Entrez un numero de trajet
-- b) Validez ce numéro de trajet
SELECT COUNT(notraj) INTO m_nombre
FROM trajet WHERE notraj = m_notraj ;
IF m_nombre = 1 THEN -- Si ce numero déjà existé
-- Selectionnez ses informations
SELECT villedep, villearr, datetrajet, nbkm
INTO m_villedep, m_villearr, m_datetrajet, m_nbkm
FROM trajet WHERE notraj = m_notraj ;
-- Affichez ses informations
DBMS_OUTPUT.PUT_LINE(' #### TRAJET SELECTIONNE #### ');
DBMS_OUTPUT.PUT_LINE(' - Trajet No: ' || m_notraj) ;
DBMS_OUTPUT.PUT_LINE(' - Ville depart : ' || m_villedep) ;
DBMS_OUTPUT.PUT_LINE(' - Ville arrivé : ' || m_villearr) ;
DBMS_OUTPUT.PUT_LINE(' - Date : ' || m_datetrajet) ;
DBMS_OUTPUT.PUT_LINE(' - Nombre de km : ' || m_nbkm) ;
ELSE -- Sinon
DBMS_OUTPUT.PUT_LINE('Ce numero n'existe pas ou il y a des erreurs');
END IF ;
END ;
/

```

Sortie de script x

Tâche terminée en 3,223 secondes

```

DBMS_OUTPUT.PUT_LINE('Ce numero n'existe pas ou il y a des erreurs ');
END IF ;
END ;
#### TRAJET SELECTIONNE ####
- Trajet No: 2
- Ville depart : MCISSE
- Ville arrivé : ERRACHIDIA
- Date : 16/11/19
- Nombre de km : 370

Procédure PL/SQL terminée.

```

Le script affiche toutes les informations d'un trajet dont le numéro est saisi. Si on saisit un numéro inexistant, il affiche :

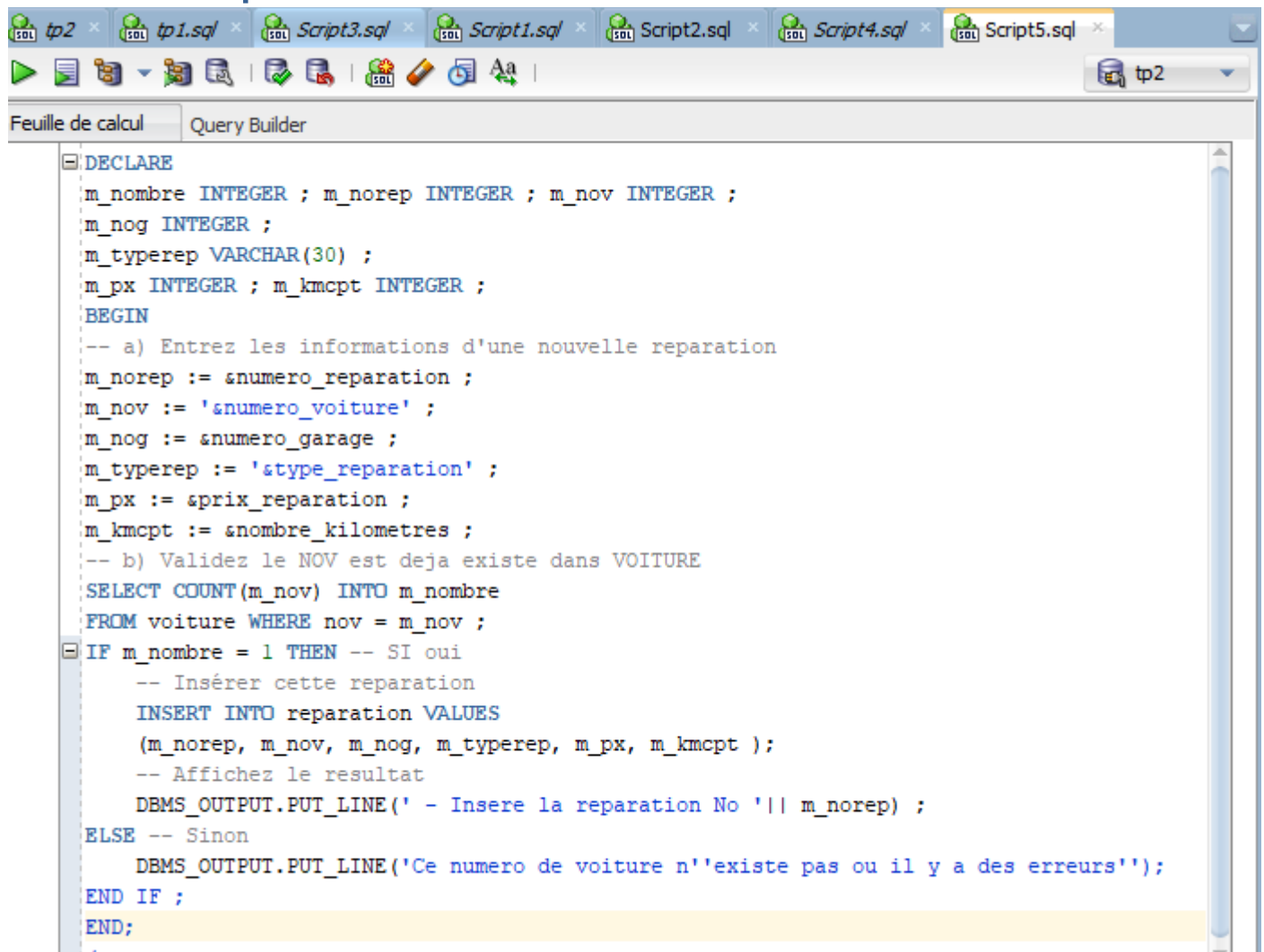
```

END ;
Ce numero n'existe pas ou il y a des erreurs

Procédure PL/SQL terminée.

```

5. Script5

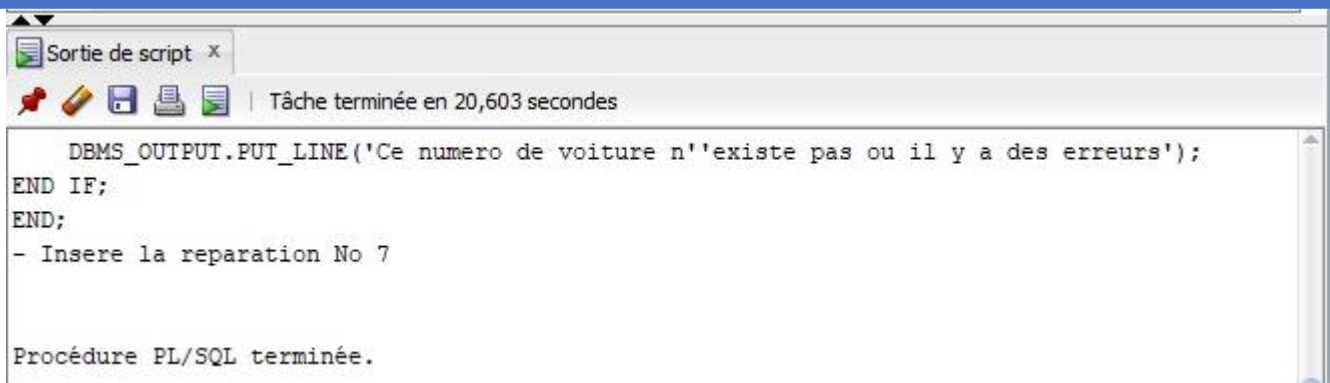


```

DECLARE
m_nombre INTEGER ; m_norep INTEGER ; m_nov INTEGER ;
m_nog INTEGER ;
m_typerep VARCHAR(30) ;
m_px INTEGER ; m_kmcpt INTEGER ;
BEGIN
-- a) Entrez les informations d'une nouvelle reparation
m_norep := &numero_reparation ;
m_nov := '&numero_voiture' ;
m_nog := &numero_garage ;
m_typerep := '&type_reparation' ;
m_px := &prix_reparation ;
m_kmcpt := &nombre_kilometres ;
-- b) Validez le NOV est deja existe dans VOITURE
SELECT COUNT(m_nov) INTO m_nombre
FROM voiture WHERE nov = m_nov ;
IF m_nombre = 1 THEN -- SI oui
-- Insérer cette reparation
INSERT INTO reparation VALUES
(m_norep, m_nov, m_nog, m_typerep, m_px, m_kmcpt ) ;
-- Affichez le resultat
DBMS_OUTPUT.PUT_LINE(' - Insere la reparation No '|| m_norep) ;
ELSE -- Sinon
DBMS_OUTPUT.PUT_LINE('Ce numero de voiture n'existe pas ou il y a des erreurs');
END IF ;
END;

```

<p>Enter Substitution Variable</p> <p>Entrez la valeur de numero_reparation :</p> <input type="text" value="7"/> <p>OK Annuler</p>	<p>Enter Substitution Variable</p> <p>Entrez la valeur de numero_voiture :</p> <input type="text" value="4"/> <p>OK Annuler</p>
<p>Enter Substitution Variable</p> <p>Entrez la valeur de numero_garage :</p> <input type="text" value="3"/> <p>OK Annuler</p>	<p>Enter Substitution Variable</p> <p>Entrez la valeur de type_reparation :</p> <input type="text" value="Portes"/> <p>OK Annuler</p>
<p>Enter Substitution Variable</p> <p>Entrez la valeur de prix_reparation :</p> <input type="text" value="150"/> <p>OK Annuler</p>	<p>Enter Substitution Variable</p> <p>Entrez la valeur de nombre_kilometres :</p> <input type="text" value="12340"/> <p>OK Annuler</p>



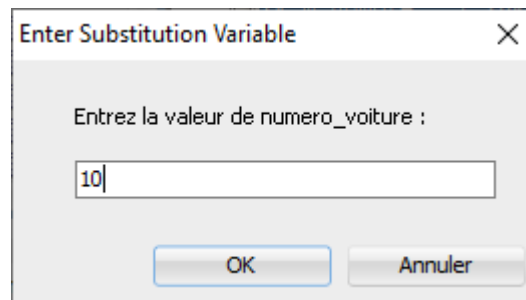
Sortie de script x

Tâche terminée en 20,603 secondes

```
DBMS_OUTPUT.PUT_LINE('Ce numero de voiture n''existe pas ou il y a des erreurs');  
END IF;  
END;  
- Insere la reparation No 7
```

Procédure PL/SQL terminée.

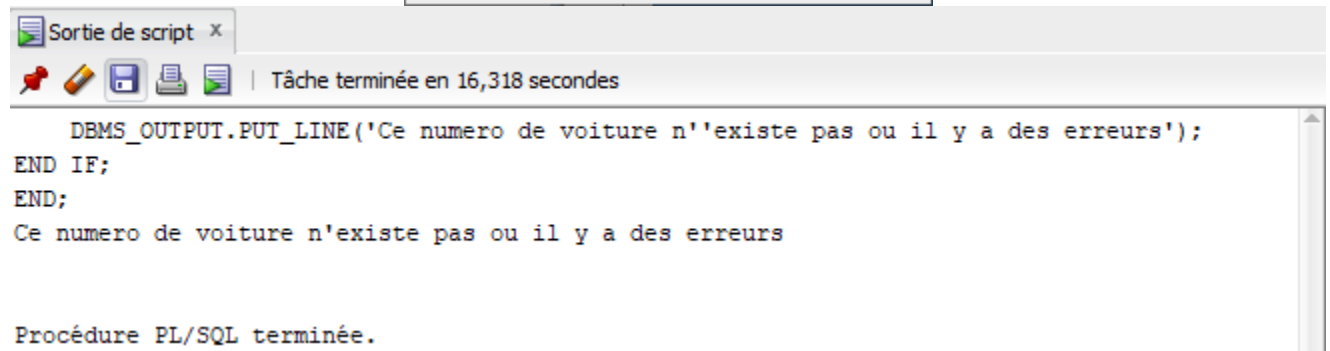
L'insertion est effectuée avec succès ! Mais si on saisit un numéro de voiture inexistant :



Enter Substitution Variable

Entrez la valeur de numero_voiture :

OK Annuler



Sortie de script x

Tâche terminée en 16,318 secondes

```
DBMS_OUTPUT.PUT_LINE('Ce numero de voiture n''existe pas ou il y a des erreurs');  
END IF;  
END;  
Ce numero de voiture n'existe pas ou il y a des erreurs
```

Procédure PL/SQL terminée.

Donc le script vérifie bien si ce numéro est déjà existant dans la table VOITURE ou pas.

6. Script6

```

DECLARE
m_notraj INTEGER := &Entrez_numero_trajet;
m_datetrajet DATE;
m_villedep VARCHAR(30);
m_villearr VARCHAR(30);
BEGIN
-- a) Sélectionnez les informations de ce trajet:
SELECT datetrajet, villedep, villearr
INTO m_datetrajet, m_villedep, m_villearr
FROM trajet WHERE m_notraj = notraj ;
IF SQL%FOUND THEN -- S'il existe
-- Affichez les informations
DBMS_OUTPUT.PUT_LINE ('### Trajet sélectionné ###');
DBMS_OUTPUT.PUT_LINE ('No trajet: '||m_notraj);
DBMS_OUTPUT.PUT_LINE ('Date : '||m_datetrajet);
DBMS_OUTPUT.PUT_LINE ('Départ : '||m_villedep);
DBMS_OUTPUT.PUT_LINE ('Arrivée : '||m_villearr);
END IF;
EXCEPTION -- Exception non trouvé
WHEN NO_DATA_FOUND THEN
DBMS_OUTPUT.PUT_LINE('ERREUR: Il n''y a pas ce trajet');
END ;
/

```

Sortie de script x

Tâche terminée en 3,018 secondes

```

EXCEPTION -- Exception non trouvé
WHEN NO_DATA_FOUND THEN
DBMS_OUTPUT.PUT_LINE('ERREUR: Il n''y a pas ce trajet');
END ;
### Trajet sélectionné ###
No trajet: 3
Date : 10/11/19
Départ : MCISSI
Arrivée : NADOR

```

Procédure PL/SQL terminée.

Alors le script affiche le trajet dont le numéro est saisi. S'il n'existe pas il affiche :

Sortie de script x

Tâche terminée en 3,07 secondes

```

DBMS_OUTPUT.PUT_LINE('ERREUR: Il n''y a pas ce trajet');
END ;
ERREUR: Il n'y a pas ce trajet

```

Procédure PL/SQL terminée.

7. Script7

```

Feuille de calcul  Query Builder
-- DECLARE
-- a) Entrer les deux dates
m_date_commencement DATE := CAST('&date_de_commencement' AS DATE);
m_date_termination DATE := CAST('&date_de_termination' AS DATE);
m_nombre_voitures INTEGER; m_nombre_personnes INTEGER;
-- b) On définit un cursor à afficher plus tard
CURSOR cur_trajet IS
    SELECT notraj, datetraj, villedep, villearr
    FROM trajet WHERE datetraj
    BETWEEN m_date_commencement AND m_date_termination ;
BEGIN
-- Le début du rapport
DBMS_OUTPUT.PUT_LINE('##### LISTE DE TRAJETS: #####') ;
DBMS_OUTPUT.PUT_LINE(' Période, De: ' || m_date_commencement || ' à ' || m_date_termination) ;
DBMS_OUTPUT.PUT_LINE('-----') ;
DBMS_OUTPUT.PUT_LINE(' : ' || 'Ligne ' || ' : ' || 'No Trajet' || ' : ' || 'Départ ' || ' : ' || 'Arrivée ' || ' : ' ||
    ' Date : ' || ' : ' || 'Nbr voitures' || ' : ' || 'Nbr personnes' || ' : ');
DBMS_OUTPUT.PUT_LINE('-----') ;
FOR rec_trajet IN cur_trajet LOOP
-- Calculez les nombres des voitures participées et les nombres des personnes transportées par chaque trajet
SELECT COUNT(*), SUM(tr_nov.nbperstr)
INTO m_nombre_voitures, m_nombre_personnes
FROM tr_nov
WHERE tr_nov.notraj = rec_trajet.notraj ;
-- Une ligne du rapport:
DBMS_OUTPUT.PUT_LINE(' : ' || TO_CHAR(cur_trajet%ROWCOUNT, '999999') || ' : ' || TO_CHAR(rec_trajet.notraj, '999999999') ||
    ' : ' || SUBSTR(rec_trajet.villedep, 0, 15) || ' : ' || SUBSTR(rec_trajet.villearr, 0, 15) || ' : ' || TO_CHAR(rec_trajet.datetraj, 'DD/MM/YY') ||
    ' : ' || TO_CHAR(m_nombre_voitures, '999999999') || ' : ' || TO_CHAR(m_nombre_personnes, '999999999') || ' : ');
END LOOP ;
-- La fin du rapport:
DBMS_OUTPUT.PUT_LINE('-----') ;
END ;

```

Il affiche pour une période donnée:

```

Sortie de script x
Tâche terminée en 8,345 secondes
##### LISTE DE TRAJETS: #####
Période, De: 12/12/12 à 24/12/19
-----
: Ligne : No Trajet : Départ : Arrivée : Date : Nbr voitures : Nbr personnes :
-----
: 1 : 1 : ALNIF : ERRACHIDIA : 16/11/19 : 1 : 2 :
: 2 : 2 : MCISSI : ERRACHIDIA : 16/11/19 : 1 : 3 :
: 3 : 3 : MCISSI : NADOR : 10/11/19 : 1 : 4 :
: 4 : 4 : MCISSI : RABAT : 10/11/19 : 1 : 4 :
: 5 : 5 : ALNIF : RABAT : 10/11/19 : 1 : 4 :
: 6 : 6 : KENITRA : RABAT : 10/12/19 : 1 : 4 :
-----
Procédure PL/SQL terminée.

```

8. Script8

```

-- tp2 x tp1.sql x Script5.sql x Script6.sql x Script7.sql x Script8.sql x
6,97700024 secondes

Feuille de calcul Query Builder

DECLARE
    m_trnov tr_nov%ROWTYPE;
    m_nch tr_nov.nch%TYPE;
    m_count INTEGER;
    plus_q_un_chauffeur EXCEPTION;
BEGIN
    m_trnov.notraj := &Numero_trajet;
    m_trnov.nov := '&Numero_Voiture';
    m_trnov.nch := &Numero_Chauffeur;
    m_trnov.nbperstr:= &Nom_Personnes;
    DBMS_OUTPUT.PUT_LINE('Insérez ce tuple');
    INSERT INTO tr_nov VALUES (m_trnov.notraj, m_trnov.nov, m_trnov.nch, m_trnov.nbperstr);
    /* Validez la règle:
    « il y a un seul chauffeur qui conduit une voiture au cours d'un trajet » */
    -- Compter le nombre de chauffeurs pour une voiture au cours d'un trajet
    SELECT COUNT(*) INTO m_count
    FROM tr_nov
    WHERE (m_trnov.notraj = tr_nov.notraj) AND
    (m_trnov.nov = tr_nov.nov);
    -- Tester
    IF m_count >= 2 THEN
        -- Lancement d'Exception
        RAISE plus_q_un_chauffeur ;
    END IF ;
    EXCEPTION -- Capture d'Exception
    WHEN plus_q_un_chauffeur THEN
        DBMS_OUTPUT.PUT_LINE ('Exception');
        ROLLBACK;
END ;

```

On a le contenu de la table **tr_nov** est :

	NOTRAJ	NOV	NCH	NBPERSTR
1	1	1	1	2
2	2	1	1	3
3	3	1	1	4
4	4	1	3	4
5	5	1	3	4
6	6	2	3	4

On insère les mêmes informations de première ligne mais on changeant le numéro de chauffeur :

The image shows four 'Enter Substitution Variable' dialog boxes arranged in a 2x2 grid. Each dialog box has a title bar with a close button (X). The top-left dialog box prompts 'Entrez la valeur de Numero_trajet :' with a text input field containing '1' and 'OK'/'Annuler' buttons. The top-right dialog box prompts 'Entrez la valeur de Numero_Voiture :' with a text input field containing '1' and 'OK'/'Annuler' buttons. The bottom-left dialog box prompts 'Entrez la valeur de Numero_Chauffeur :' with a text input field containing '3' and 'OK'/'Annuler' buttons. The bottom-right dialog box prompts 'Entrez la valeur de Nom_Personnes :' with a text input field containing '2' and 'OK'/'Annuler' buttons. Below these dialog boxes is a 'Sortie de script' window with a title bar and a close button. It contains the following text: 'Tâche terminée en 64,382 secondes', 'ROLLBACK;', 'END ;', 'Insérez ce tuple', 'Exception', and 'Procédure PL/SQL terminée.'

Impossible d'insérer, il génère une exception car « il y a un seul chauffeur qui conduit une voiture au cours d'un trajet pas deux »!

9. Script9

Le script 9 affiche tous les trajets et les voitures qui transgressent la règle (les numéros de trajet et les numéros de voiture) suivante :

«le nombre de passagers transportés pendant une journée doit être inférieur à 50»

Feuille de calcul Query Builder

```

DECLARE
    max_personnes CONSTANT INTEGER := 50 ;
    m_nombre_personnes INTEGER; m_notraj INTEGER; m_datetrajet DATE;
    CURSOR cur_les_jours IS -- Cursor pour les journées
        SELECT DISTINCT datetrajet FROM trajet;
    CURSOR cur_les_trajets IS -- Cursor pour les trajets de la journées
        SELECT DISTINCT notraj FROM trajet
        WHERE datetrajet = m_datetrajet;
    CURSOR cur_les_voitures IS -- Cursor pour les voiture de trajets
        SELECT nov FROM tr_nov
        WHERE (notraj = m_notraj);
BEGIN
    FOR rec_les_jours IN cur_les_jours LOOP -- Traitez la liste des jours
        m_datetrajet := rec_les_jours.datetrajet; -- Traitez une journée de la liste des jours
        -- Validez la règle pour cette journée
        SELECT SUM(nbperstr) INTO m_nombre_personnes
        FROM tr_nov, trajet
        WHERE (tr_nov.notraj = trajet.notraj) AND (trajet.datetrajet = m_datetrajet);
        IF m_nombre_personnes > max_personnes THEN -- Trop de personnes (supérieur au max)
            DBMS_OUTPUT.PUT_LINE('On a trop de personnes: ' || m_nombre_personnes || ' personnes pendant la journée de, Date: ' || m_datetrajet);
            -- Affichage des trajets et voitures correspondantes (no. voiture et no. trajet)
            FOR rec_les_trajets IN cur_les_trajets LOOP
                m_notraj := rec_les_trajets.notraj; -- afficher trajet
                DBMS_OUTPUT.PUT_LINE(' >>> Trajet numéro: ' || m_notraj);
                FOR rec_les_voitures IN cur_les_voitures LOOP
                    DBMS_OUTPUT.PUT_LINE(' >>> Numéro de voiture: ' || rec_les_voitures.nov); -- Afficher voiture
                END LOOP;
            END LOOP;
        END IF;
    END LOOP;
END;

```

Sortie de script x

Tâche terminée en 0,074 secondes

Procédure PL/SQL terminée.

Puisque nous ne disposons pas d'assez d'enregistrement, le script n'affiche rien. Pour cela, on fixant max comme 0 il afficher:

tp2 x tp1.sql x Script8.sql x Script9.sql x Script10.sql x

Feuille de calcul Query Builder

```

DECLARE
    max_personnes CONSTANT INTEGER := 0 ;
    m_nombre_personnes INTEGER; m_notraj INTEGER; m_datetrajet DATE;
    CURSOR cur_les_jours IS -- Cursor pour les journées
        SELECT DISTINCT datetrajet FROM trajet;
    CURSOR cur_les_trajets IS -- Cursor pour les trajets de la journées
        SELECT DISTINCT notraj FROM trajet
        WHERE datetrajet = m_datetrajet;
    CURSOR cur_les_voitures IS -- Cursor pour les voiture de trajets
        SELECT nov FROM tr_nov
        WHERE (notraj = m_notraj);
BEGIN
    FOR rec_les_jours IN cur_les_jours LOOP -- Traitez la liste des jours
        m_datetrajet := rec_les_jours.datetrajet; -- Traitez une journée de la liste des jours
        -- Validez la règle pour cette journée
        SELECT SUM(nbperstr) INTO m_nombre_personnes
        FROM tr_nov, trajet
        WHERE (tr_nov.notraj = trajet.notraj) AND (trajet.datetrajet = m_datetrajet);
        IF m_nombre_personnes > max_personnes THEN -- Trop de personnes (supérieur au max)
            DBMS_OUTPUT.PUT_LINE('On a trop de personnes: ' || m_nombre_personnes || ' personnes pendant la journée de, Date: ' || m_datetrajet);
            -- Affichage des trajets et voitures correspondantes (no. voiture et no. trajet)
            FOR rec_les_trajets IN cur_les_trajets LOOP
                m_notraj := rec_les_trajets.notraj; -- afficher trajet
                DBMS_OUTPUT.PUT_LINE(' >>> Trajet numéro: ' || m_notraj);
                FOR rec_les_voitures IN cur_les_voitures LOOP
                    DBMS_OUTPUT.PUT_LINE(' >>> Numéro de voiture: ' || rec_les_voitures.nov); -- Afficher voiture
                END LOOP;
            END LOOP;
        END IF;
    END LOOP;
END;

```

```

Sortie de script x
Tâche terminée en 0,067 secondes

On a trop de personnes: 4 personnes pendant la journée de, Date: 10/12/19
>>> Trajet numéro: 6
>>> Numéro de voiture: 2
On a trop de personnes: 12 personnes pendant la journée de, Date: 10/11/19
>>> Trajet numéro: 3
>>> Numéro de voiture: 1
>>> Trajet numéro: 4
>>> Numéro de voiture: 1
>>> Trajet numéro: 5
>>> Numéro de voiture: 1
On a trop de personnes: 7 personnes pendant la journée de, Date: 16/11/19
>>> Trajet numéro: 1
>>> Numéro de voiture: 1
>>> Trajet numéro: 2
>>> Numéro de voiture: 1

Procédure PL/SQL terminée.

```

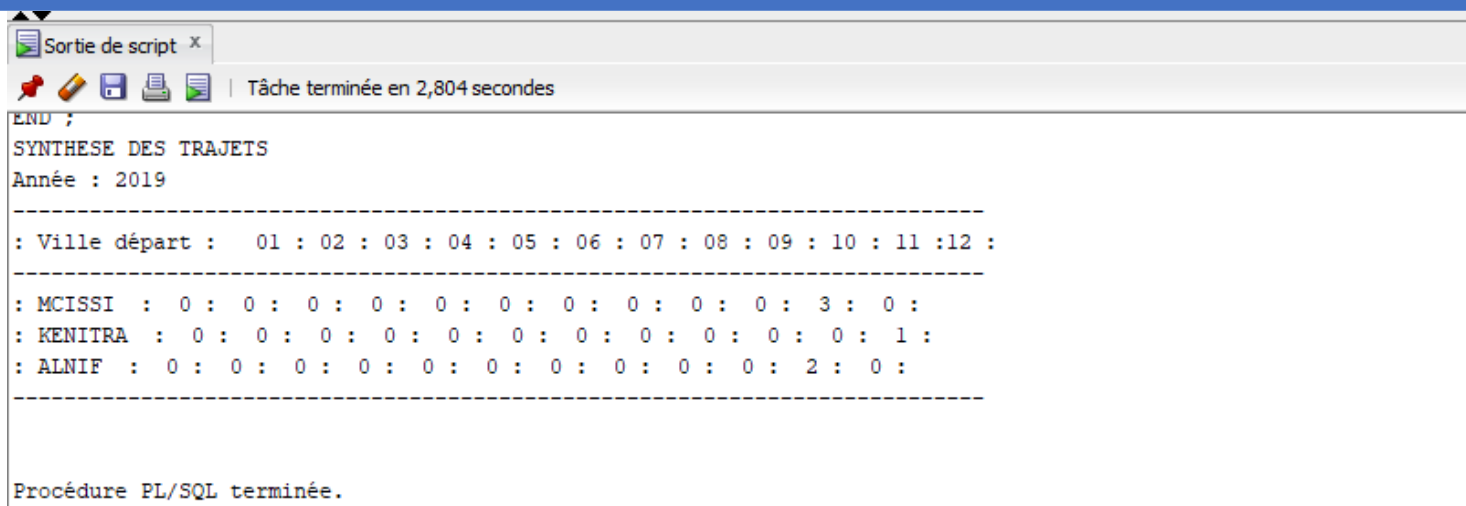
10. Script10

```

tp2 x tp1.sql x Script10.sql x
Feuille de calcul Query Builder

DECLARE
  m_annee INTEGER(4) := '<Entrez une annee'; -- a) entrer une annee
  m_villedep trajet.villedep%TYPE;
  m_index INTEGER; m_nombre_trajets INTEGER;
  CURSOR cur_les_villes IS -- liste des villes de départ de l'année saisie
    SELECT DISTINCT villedep FROM trajet
    WHERE TO_CHAR(datetrajet,'YYYY') = m_annee;
BEGIN
  -- Le début du rapport annuel
  DBMS_OUTPUT.PUT_LINE('SYNTHESE DES TRAJETS');
  DBMS_OUTPUT.PUT_LINE('Année : '||m_annee);
  DBMS_OUTPUT.PUT_LINE('-----');
  DBMS_OUTPUT.PUT_LINE(': Ville départ : 01 : 02 : 03 : 04 : 05 : 06 : 07 : 08 : 09 : 10 : 11 :12 :');
  DBMS_OUTPUT.PUT_LINE('-----');
  -- Le contenu du rapport
  FOR rec_les_villes IN cur_les_villes LOOP -- Traitez une ville
    m_villedep := rec_les_villes.villedep;
    DBMS_OUTPUT.PUT(' : '||SUBSTR(m_villedep,0,7)||' :');
    -- Traitez les mois (Pour chaque mois de 1 à 12)
    FOR m_index IN 1..12 LOOP -- Traitez un mois (m_index)
      SELECT COUNT(*) INTO m_nombre_trajets
      FROM trajet
      WHERE (villedep = m_villedep ) AND (TO_CHAR(datetrajet,'MM')= m_index) AND (TO_CHAR(datetrajet,'YYYY') = m_annee);
      DBMS_OUTPUT.PUT(TO_CHAR(m_nombre_trajets,'99')||' :');
    END LOOP; -- Les mois
    DBMS_OUTPUT.NEW_LINE; -- Nouvelle ligne
  END LOOP ; -- Les villes
  -- La fin du rapport:
  DBMS_OUTPUT.PUT_LINE('-----');
END ;
/

```



```
END ;
SYNTHESE DES TRAJETS
Année : 2019

-----
: Ville départ :   01 : 02 : 03 : 04 : 05 : 06 : 07 : 08 : 09 : 10 : 11 :12 :
-----
: MCISSI   : 0 : 0 : 0 : 0 : 0 : 0 : 0 : 0 : 0 : 0 : 3 : 0 :
: KENITRA  : 0 : 0 : 0 : 0 : 0 : 0 : 0 : 0 : 0 : 0 : 0 : 1 :
: ALNIF    : 0 : 0 : 0 : 0 : 0 : 0 : 0 : 0 : 0 : 0 : 2 : 0 :
-----

Procédure PL/SQL terminée.
```

Le script fait la synthèse des trajets pendant une année.
Pour l'analyse statistique les nombres des trajets effectués par mois et ville de départ.

FIN

Merci pour votre lecture !