

Rapport TP01

BD1

NAAJI Dorian & ARMANET Nathan – 3A INFO GROUPE 2
POLYTECH LYON
DORIAN.NAAJI@ETU.UNIV-LYON1.FR
NATHAN.ARMANET@ETU.UNIV-LYON1.FR

1. Partie 1

1.1. Création de commande

1.1.1. Question 1

On ouvre le fichier commandes.sql et on exécute le script. On obtient un message de validation dans la console.

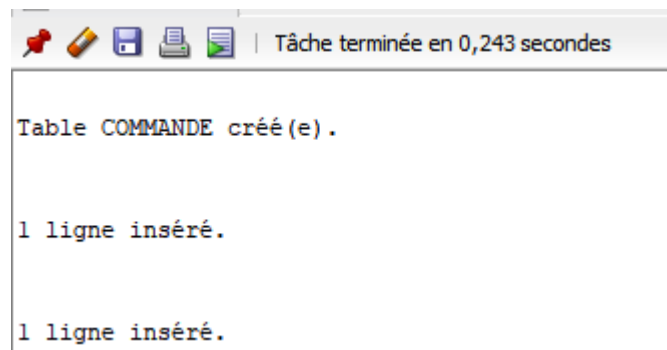


FIGURE 1 : RESULTAT DE LA CREATION DE TABLE

1.1.2. Question 2

Il suffit d'utiliser la commande : `SELECT * FROM USER_TABLES;`

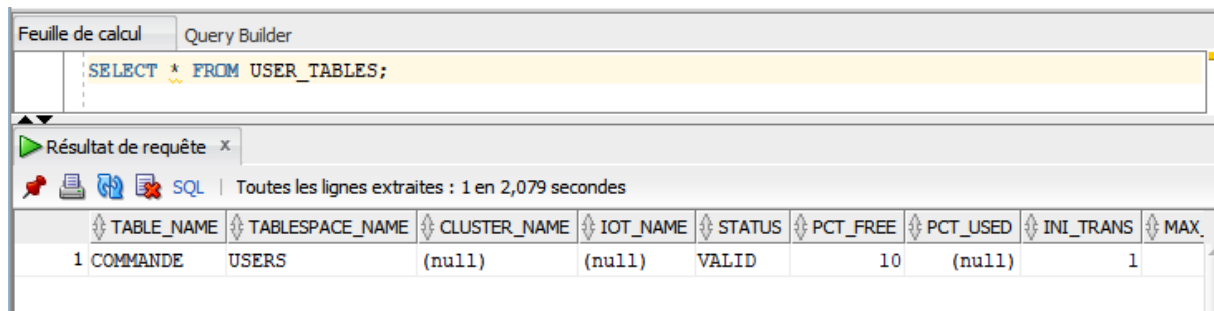
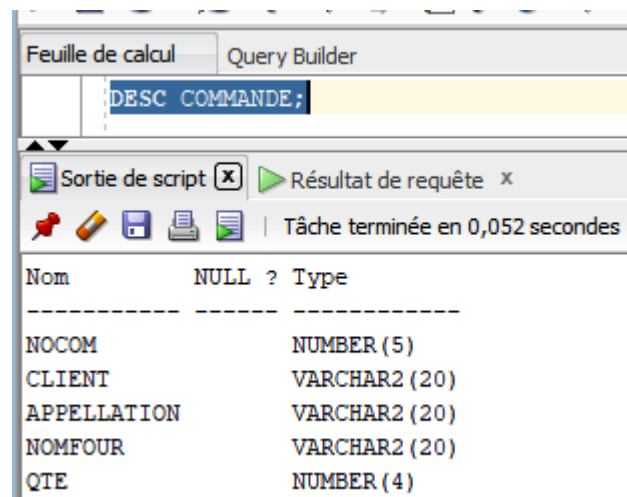


FIGURE 2 : COMMANDE QUESTION 2

1.1.3. Question 3

On exécute *DESC COMMANDE* ;



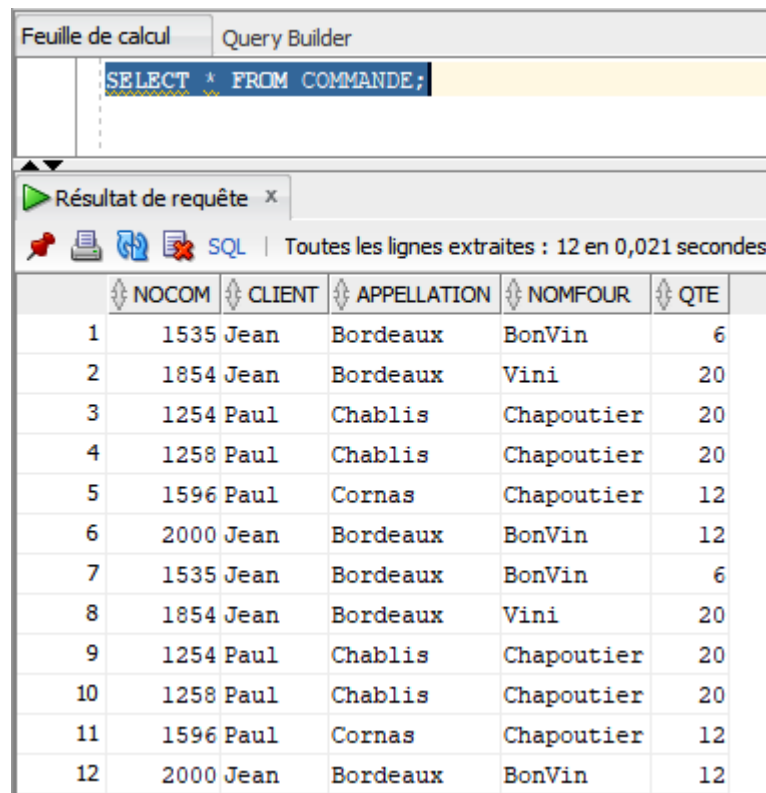
The screenshot shows a 'Query Builder' window with the SQL command 'DESC COMMANDE;' entered. Below the command bar, there is a toolbar with icons for saving, printing, and other functions. A status bar indicates 'Tâche terminée en 0,052 secondes'. The main area displays the table structure with columns: Nom, NULL, and Type.

| Nom | NULL | Type |
|-------------|------|---------------|
| NOCOM | | NUMBER (5) |
| CLIENT | | VARCHAR2 (20) |
| APPELLATION | | VARCHAR2 (20) |
| NOMFOUR | | VARCHAR2 (20) |
| QTE | | NUMBER (4) |

FIGURE 3 : RESULTAT DE LA REQUETE

1.1.4. Question 4

On exécute *SELECT * FROM COMMANDE* ; afin de vérifier que les tuples ont bien été insérés.



The screenshot shows a 'Query Builder' window with the SQL command 'SELECT * FROM COMMANDE;' entered. Below the command bar, there is a toolbar with icons for saving, printing, and other functions. A status bar indicates 'Toutes les lignes extraites : 12 en 0,021 secondes'. The main area displays a table with 12 rows and 6 columns: NOCOM, CLIENT, APPELLATION, NOMFOUR, and QTE.

| | NOCOM | CLIENT | APPELLATION | NOMFOUR | QTE |
|----|-------|--------|-------------|------------|-----|
| 1 | 1535 | Jean | Bordeaux | BonVin | 6 |
| 2 | 1854 | Jean | Bordeaux | Vini | 20 |
| 3 | 1254 | Paul | Chablis | Chapoutier | 20 |
| 4 | 1258 | Paul | Chablis | Chapoutier | 20 |
| 5 | 1596 | Paul | Cornas | Chapoutier | 12 |
| 6 | 2000 | Jean | Bordeaux | BonVin | 12 |
| 7 | 1535 | Jean | Bordeaux | BonVin | 6 |
| 8 | 1854 | Jean | Bordeaux | Vini | 20 |
| 9 | 1254 | Paul | Chablis | Chapoutier | 20 |
| 10 | 1258 | Paul | Chablis | Chapoutier | 20 |
| 11 | 1596 | Paul | Cornas | Chapoutier | 12 |
| 12 | 2000 | Jean | Bordeaux | BonVin | 12 |

1.2. Création de produit

1.2.1. Question 1

On effectue la commande « *CREATE TABLE* » suivante :

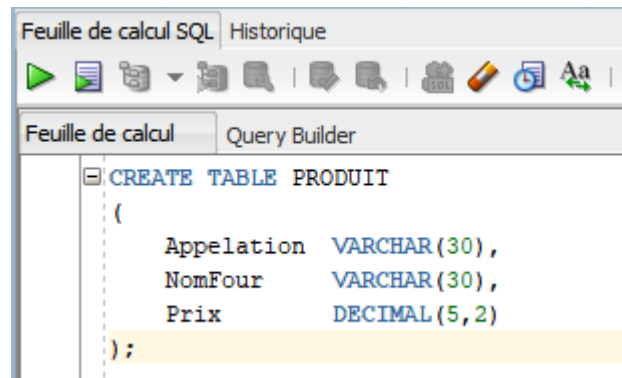


FIGURE 4 : CREATION DE LA TABLE PRODUIT

1.2.2. Question 2

On insère les tuples grâce à une commande « *INSERT INTO* »

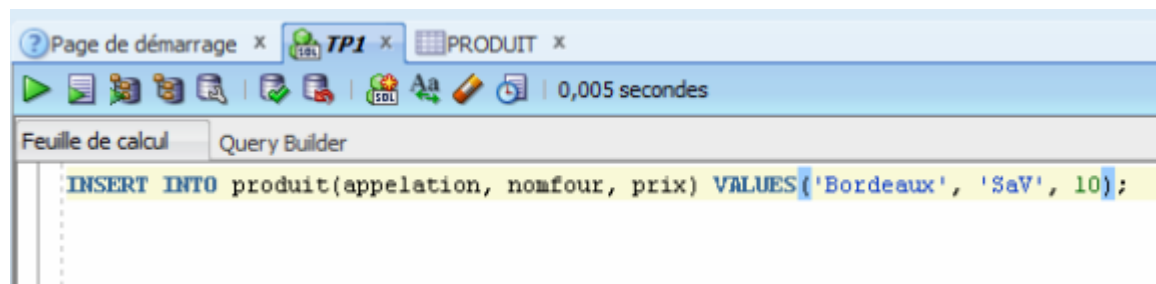


FIGURE 5 : INSERTION DES DONNEES

1.2.3. Question 3

On effectue les vérifications d'usage

| Feuille de calcul | | Query Builder | |
|-------------------|------------|--|------|
| | | <pre>SELECT * FROM USER_TABLES; DESC PRODUIT; SELECT * FROM PRODUIT;</pre> | |
| | | Résultat de requête x Sortie de script x Résultat de requête 1 x | |
| | | Toutes les lignes extraites : 8 en 0,012 secondes | |
| | APPELATION | NOMFOUR | PRIX |
| 1 | Bordeaux | BonVin | 20 |
| 2 | Corvas | Chapoutier | 18 |
| 3 | Bordeaux | Vini | 8,2 |
| 4 | Boudes | Vini | 4,3 |
| 5 | Bordeaux | Chapoutier | 18,5 |
| 6 | Chapoutier | Chapoutier | 5,1 |
| 7 | Chablis | Chapoutier | 5 |
| 8 | Bordeaux | SaV | 10 |

FIGURE 6 : VERIFICATIONS D'USAGE POUR LA TABLE PRODUIT

1.3. Création de fournisseur

1.3.1. Question 1

Pour visualiser le contenu, on effectue un « *SELECT* ».

1.3.2. Question 2

Pour créer la table à partir de la requête précédente, on utilise l'opérateur « AS » :

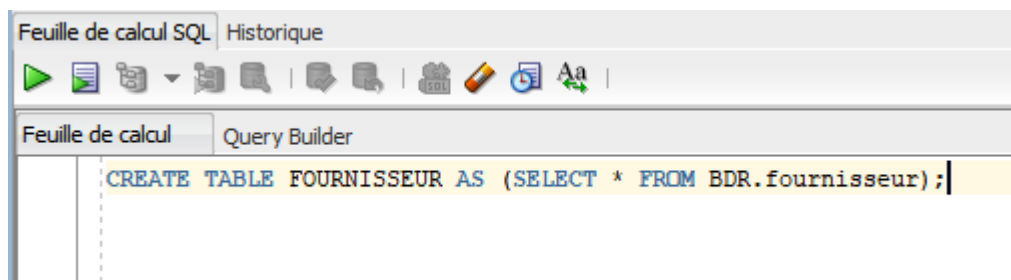


FIGURE 8 : CREATION DE LA TABLE A PARTIR D'UN SELECT

1.3.3. Question 3

On effectue finalement les vérifications d'usage

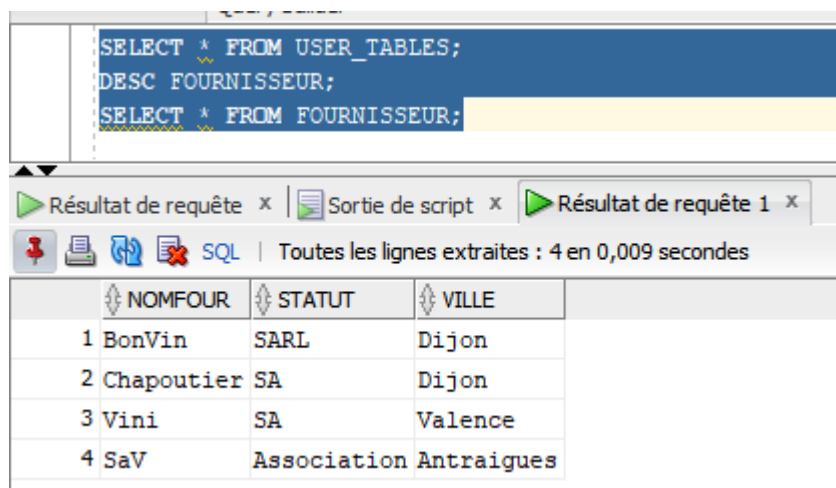
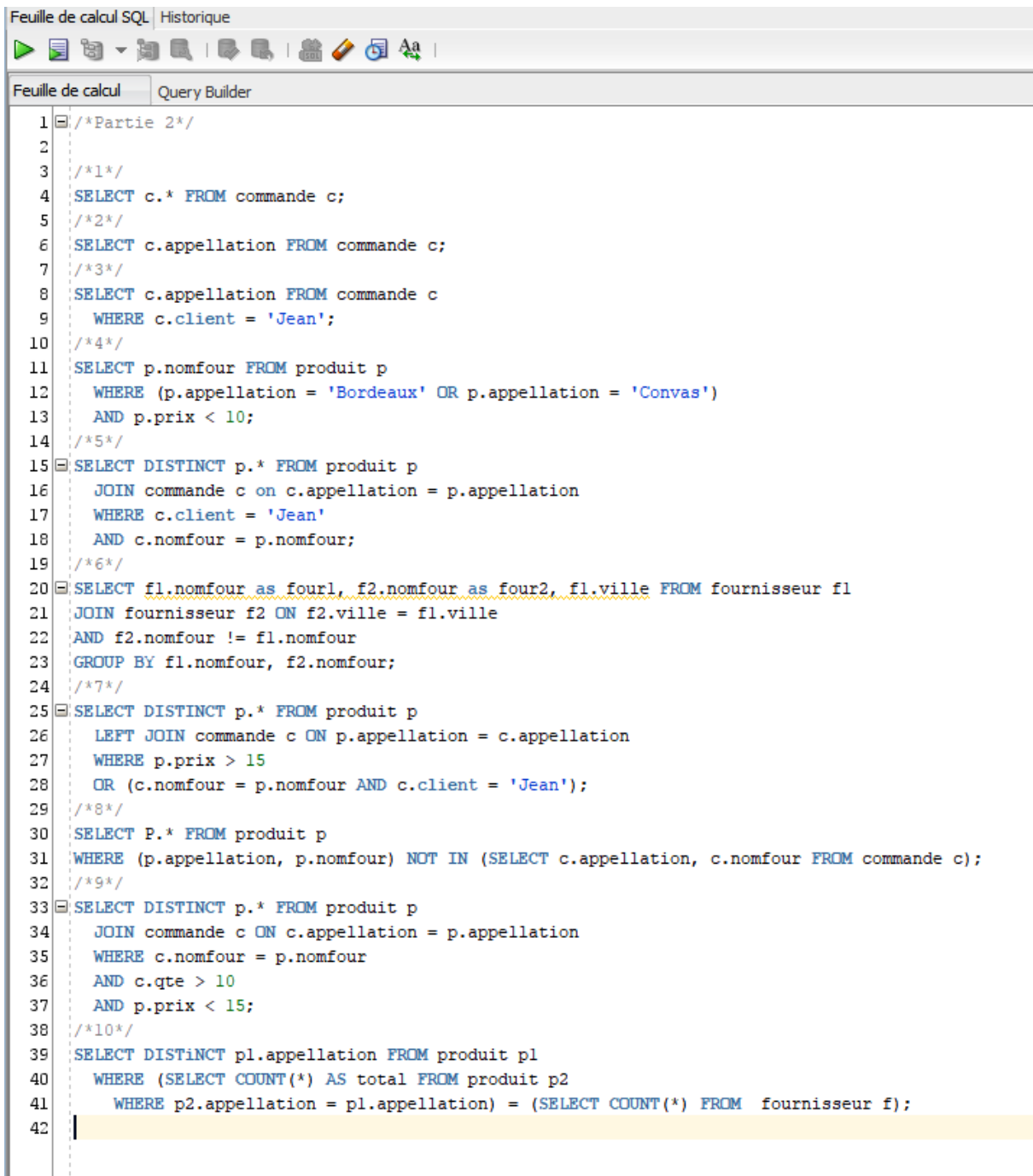


FIGURE 9 : VERIFICATIONS D'USAGE SUR LA TABLE FOURNISSEUR

2. Partie 2 : Interrogation des tables

2.1. Exécution des requêtes



```
1  /*Partie 2*/
2
3  /*1*/
4  SELECT c.* FROM commande c;
5  /*2*/
6  SELECT c.appellation FROM commande c;
7  /*3*/
8  SELECT c.appellation FROM commande c
9     WHERE c.client = 'Jean';
10 /*4*/
11 SELECT p.nomfour FROM produit p
12     WHERE (p.appellation = 'Bordeaux' OR p.appellation = 'Convas')
13     AND p.prix < 10;
14 /*5*/
15 SELECT DISTINCT p.* FROM produit p
16     JOIN commande c ON c.appellation = p.appellation
17     WHERE c.client = 'Jean'
18     AND c.nomfour = p.nomfour;
19 /*6*/
20 SELECT f1.nomfour as four1, f2.nomfour as four2, f1.ville FROM fournisseur f1
21     JOIN fournisseur f2 ON f2.ville = f1.ville
22     AND f2.nomfour != f1.nomfour
23     GROUP BY f1.nomfour, f2.nomfour;
24 /*7*/
25 SELECT DISTINCT p.* FROM produit p
26     LEFT JOIN commande c ON p.appellation = c.appellation
27     WHERE p.prix > 15
28     OR (c.nomfour = p.nomfour AND c.client = 'Jean');
29 /*8*/
30 SELECT P.* FROM produit p
31     WHERE (p.appellation, p.nomfour) NOT IN (SELECT c.appellation, c.nomfour FROM commande c);
32 /*9*/
33 SELECT DISTINCT p.* FROM produit p
34     JOIN commande c ON c.appellation = p.appellation
35     WHERE c.nomfour = p.nomfour
36     AND c.qte > 10
37     AND p.prix < 15;
38 /*10*/
39 SELECT DISTINCT pl.appellation FROM produit pl
40     WHERE (SELECT COUNT(*) AS total FROM produit p2
41         WHERE p2.appellation = pl.appellation) = (SELECT COUNT(*) FROM fournisseur f);
42
```

FIGURE 10 : ENSEMBLE DES REQUETES

3. Exercice 2

3.1. Création des tables

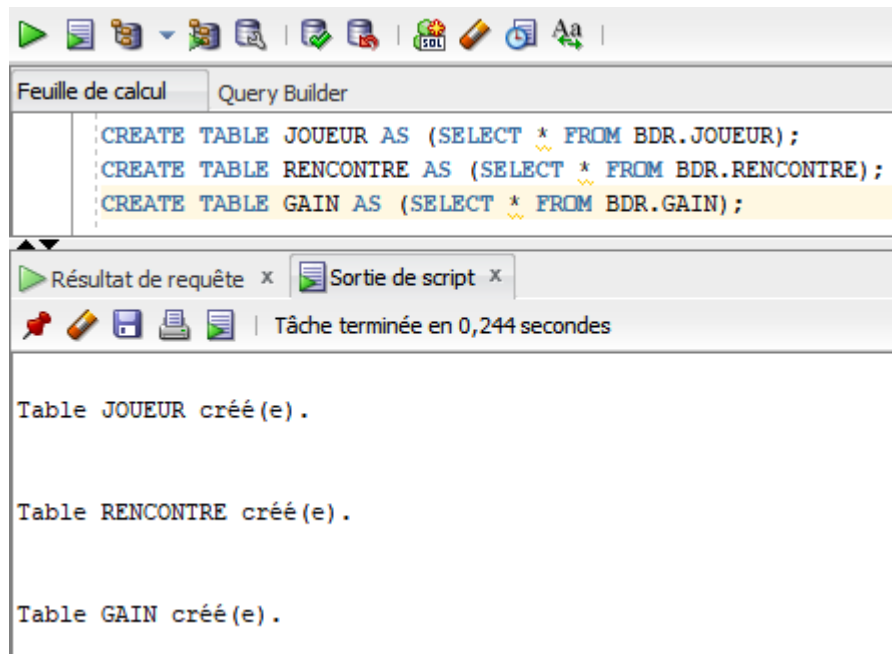
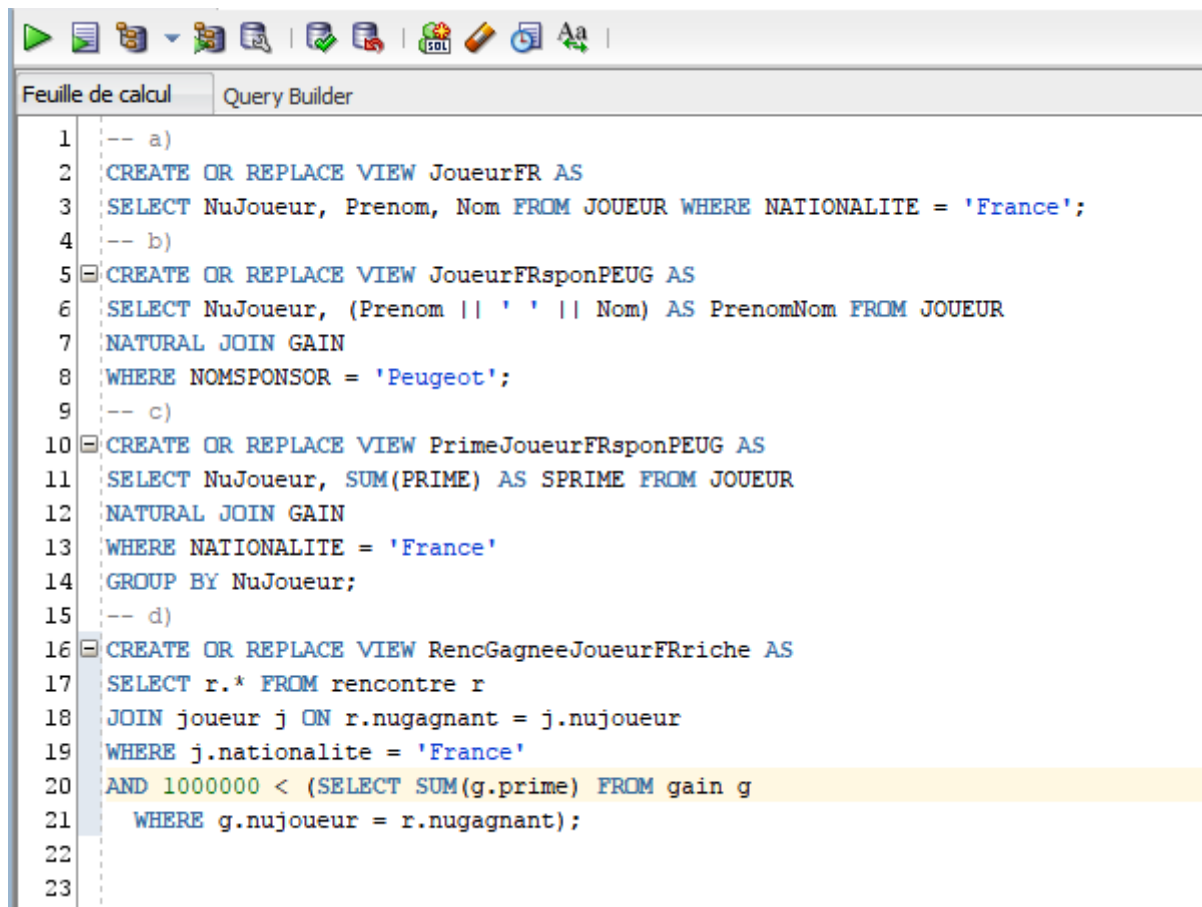


FIGURE 11 : CREATION DES TABLES

On crée les tables grâce à un « *CREATE TABLE [] AS* ».

3.2. Création de vues



```
1  -- a)
2  CREATE OR REPLACE VIEW JoueurFR AS
3  SELECT NuJoueur, Prenom, Nom FROM JOUEUR WHERE NATIONALITE = 'France';
4  -- b)
5  CREATE OR REPLACE VIEW JoueurFRsponPEUG AS
6  SELECT NuJoueur, (Prenom || ' ' || Nom) AS PrenomNom FROM JOUEUR
7  NATURAL JOIN GAIN
8  WHERE NOMSPONSOR = 'Peugeot';
9  -- c)
10 CREATE OR REPLACE VIEW PrimeJoueurFRsponPEUG AS
11 SELECT NuJoueur, SUM(PRIME) AS SPRIME FROM JOUEUR
12 NATURAL JOIN GAIN
13 WHERE NATIONALITE = 'France'
14 GROUP BY NuJoueur;
15 -- d)
16 CREATE OR REPLACE VIEW RencGagneeJoueurFRriche AS
17 SELECT r.* FROM rencontre r
18 JOIN joueur j ON r.nugagnant = j.nujoueur
19 WHERE j.nationalite = 'France'
20 AND 1000000 < (SELECT SUM(g.prime) FROM gain g
21               WHERE g.nujoueur = r.nugagnant);
22
23
```

FIGURE 12 : CREATION DES VUES

3.3. Interrogation des tables

| Feuille de calcul | Query Builder |
|-------------------|--|
| 1 | --1 |
| 2 | select * from joueur; |
| 3 | select * from gain; |
| 4 | select * from rencontre; |
| 5 | |
| 6 | --2 |
| 7 | select g.nujoueur, g.lieutournoi, g.annee from gain g |
| 8 | where g.nomsponsor = 'Peugeot' |
| 9 | and g.annee >=1990 |
| 10 | and g.annee <= 1994; |
| 11 | |
| 12 | --3 |
| 13 | SELECT DISTINCT j.nom, j.annais from joueur j |
| 14 | WHERE j.nujoueur in (select rl.nugagnant from rencontre rl |
| 15 | where rl.lieutournoi = 'Roland Garros' |
| 16 | and rl.annee = 1994) |
| 17 | OR j.nujoueur in (select r2.nuperdant from rencontre r2 |
| 18 | where r2.lieutournoi = 'Roland Garros' |
| 19 | and r2.annee = 1994); |
| 20 | |
| 21 | --4 |
| 22 | select DISTINCT j.nujoueur, j.nom, j.nationalite from joueur j |
| 23 | join gain g on g.nujoueur = j.nujoueur |
| 24 | where g.nomsponsor = 'Peugeot' |
| 25 | and j.nujoueur in (select r.nugagnant from rencontre r |
| 26 | where r.lieutournoi = 'Roland Garros'); |
| 27 | |
| 28 | --5 |
| 29 | select j.nom, j.nationalite from joueur j |
| 30 | where (j.nujoueur in (SELECT rl.nugagnant from rencontre rl |
| 31 | where rl.lieutournoi = 'Roland Garros' |
| 32 | and rl.annee = 1992) or j.nujoueur in (SELECT r2.nuperdant from rencontre r2 |
| 33 | where r2.lieutournoi = 'Roland Garros' |
| 34 | and r2.annee = 1992)) |
| 35 | and (j.nujoueur in (SELECT r3.nugagnant from rencontre r3 |
| 36 | where r3.lieutournoi = 'Wimbledon' |
| 37 | and r3.annee = 1992) or j.nujoueur in (SELECT r4.nuperdant from rencontre r4 |
| 38 | where r4.lieutournoi = 'Wimbledon' |
| 39 | and r4.annee = 1992)); |
| 40 | |
| 41 | --6 |
| 42 | select count(*) as nbJoueur from joueur j |
| 43 | where (j.nujoueur in (SELECT r3.nugagnant from rencontre r3 |
| 44 | where r3.lieutournoi = 'Wimbledon' |
| 45 | and r3.annee = 1993) or j.nujoueur in (SELECT r4.nuperdant from rencontre r4 |
| 46 | where r4.lieutournoi = 'Wimbledon' |
| 47 | and r4.annee = 1993)); |
| 48 | |

FIGURE 13 : REQUETES

```

49 | --7
50 | select g.annee, avg(g.prime) as MoyPrime from gain g
51 | group by g.annee;
52 |

```

FIGURE 15 : REQUETES

```

53 | --8
54 | select distinct j.nujoueur, j.nom from joueur j
55 |   where j.nujoueur not in (select g1.nujoueur from gain g
56 |     where g1.lieutournoi = 'Roland Garros'
57 |     and g1.prime < 1000000)
58 |   and j.nujoueur in (select g2.nujoueur from gain g2
59 |     where g2.lieutournoi = 'Roland Garros');
60 |

```

FIGURE 14 : REQUETES

Requête 9 :

```

select j.nom from joueur j
  where j.nujoueur in (SELECT r1.nuperdant from rencontre r1
    where r1.lieutournoi = 'Wimbledon')
 and j.nujoueur not in (SELECT r2.nugagnant from rencontre r2
    where r2.lieutournoi = 'Wimbledon')
 and j.nujoueur in (SELECT r3.nugagnant from rencontre r3
    where r3.lieutournoi = 'Roland Garros')
 and j.nujoueur not in (SELECT r4.nuperdant from rencontre r4
    where r4.lieutournoi = 'Roland Garros');

```