

RÉPUBLIQUE TUNISIENNE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA
RECHERCHE SCIENTIFIQUE

DOSSIER DE CANDIDATURE

Soumis à

La commission Nationale de
Recrutement des Maîtres assistants

Dans la discipline

Informatique

Travaux Pratiques Corrigés

Bases de Données

Elaboré par :

Dr. Ghassen HAMDI

Laboratoire MARS, Université de Sousse


Directeur Département
Génie Informatique Industrielle
ROUZOUITA Badreddine



Travaux Pratiques

Bases de Données

Niveau : 1^{ère} Informatique Appliquée

Préparé Par : HAMDI Ghassen

Docteur en Sciences Informatique.

Année universitaire: 2017/2018

Travaux pratiques
« Base de données »

Objectif(s) :

- ☐ Créer des bases de données en utilisant le SGBD Oracle et du langage SQL.
- ☐ S'habituer à utiliser le langage SQL et interroger des bases de données relationnelles à travers les requêtes SQL SELECT.
- ☐ Etre capable à créer de nouvelles tables et de modifier les structures des tables existes déjà en adoptant le langage de définition des données LDD.
- ☐ Taper des requêtes d'ajout, de modifications et de suppression des données mémorisées dans une base de données relationnelles à travers le langage de manipulation des données LMD.
- ☐ La gestion des utilisateurs et des rôles utilisant la base de données à travers le langage de contrôle des données LCD.

Bibliographie :**▪ Sources Internet :**

- [odile.papini.perso.luminy.univ-amu.fr](mailto:odile.papini.perso@luminy.univ-amu.fr).
- docs.oracle.com.
- 123dok.net.

▪ Supports de cours :

- Support de cours : Odile PAPINI, POLYTECH, Université d'Aix-Marseille.

Sommaire

TP 1 : LANGAGE DE DEFINITION DE DONNEES (LDD)	3	2
Exercice 1	3	
Exercice 2	7	
TP2 : LANGAGE DE MANIPULATION DES DONNEES (LMD).....	12	
Exercice	12	
TP3 : LANGAGE DE CONTROLE DE DONNEES	25	
Exercice	25	

TP 1 : LANGAGE DE DEFINITION DE DONNEES (LDD)

3

Objectif : le TP sur le LDD en SQL vise à enseigner aux étudiants comment faire la conception de la structure d'une base de données (BD) relationnelle. Le LDD consiste à créer, modifier et supprimer des structures de la BD à savoir les tables, les contraintes, etc.

Exercice 1

Pour réaliser ce TP, nous avons besoin de la version ORACLE 11 g Express Edition et de l'éditeur SQL Developer.

Les tables suivantes correspondent à une base de données d'une université :

Etudiant (numetu, nom, prenom, datenaiss, rue, cp, ville)

Matiere (codemat, libelle, coef)

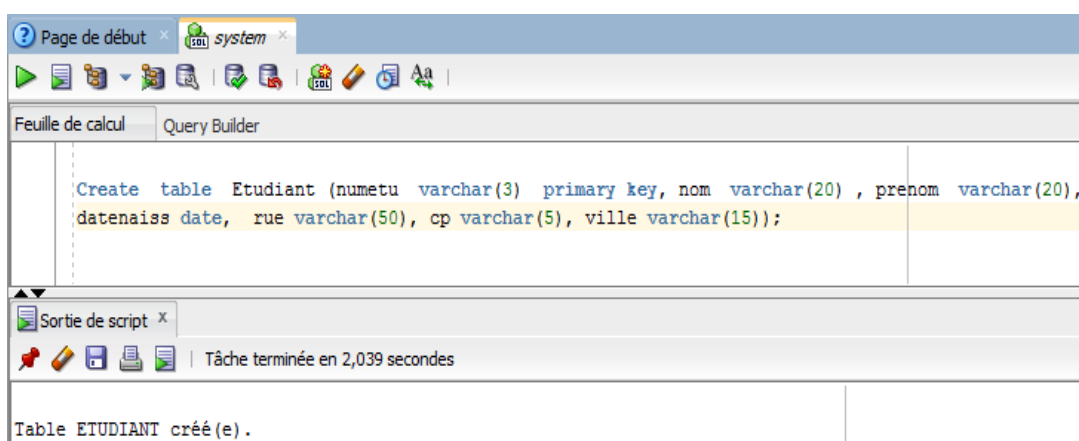
Epreuve (numepreuve, lieu, codemat)

Notation (numetu, numepreuve, note)

En élaborant les tables explorées dans le modèle relationnel ci- dessus, le but de cet exercice est de s'entraîner à utiliser les commandes : CREATE, DROP, ALTER,

1. Pour montrer les étudiants de l'université, créer la table **Etudiant**. Cette table comprend une clé primaire **numetu** de 3 caractères, un nom de l'étudiant **nom** (20 caractères), un prénom **prenom** (20 caractères), une date de naissance **datenaiss** (date) et des attributs **rue** (50 caractères), code postal **cp** (5 caractères) et nom de **ville** (15 caractères) correspond à une adresse.

Correction :

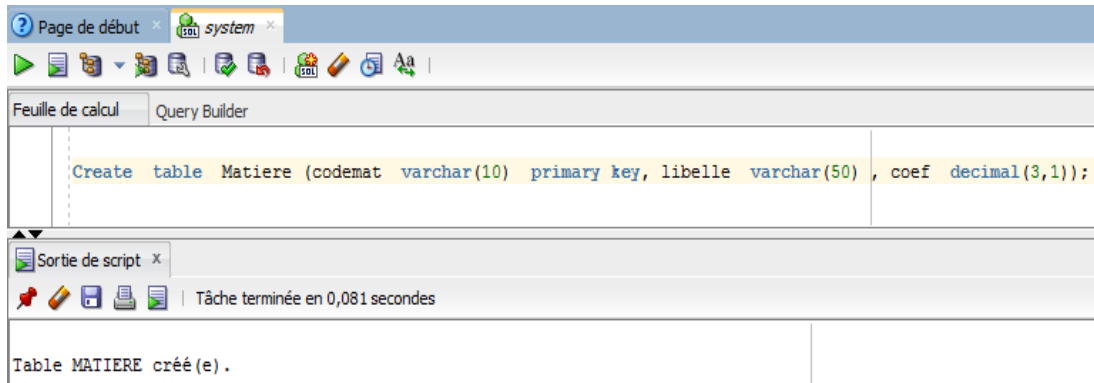


2. Pour montrer les matières étudiées à l'université, créer la table **Matiere**. Cette table comprend une clé primaire **codemat** formée de 10 caractères max, un libellé **libelle** (50

caractères), et un coefficient numérique **coef** (deux chiffres et un décimal) afin de préserver le coefficient de la matière.

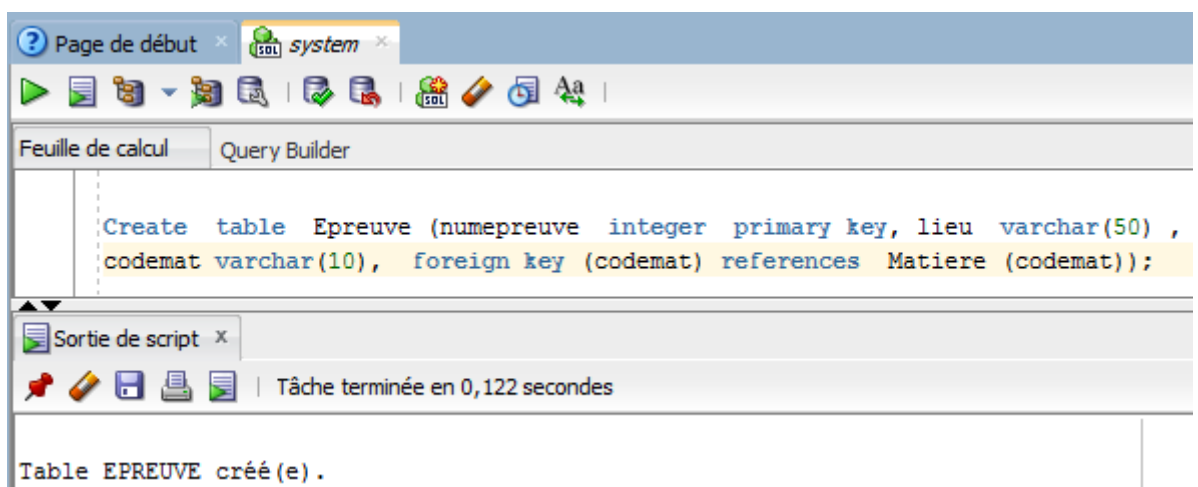
4

Correction



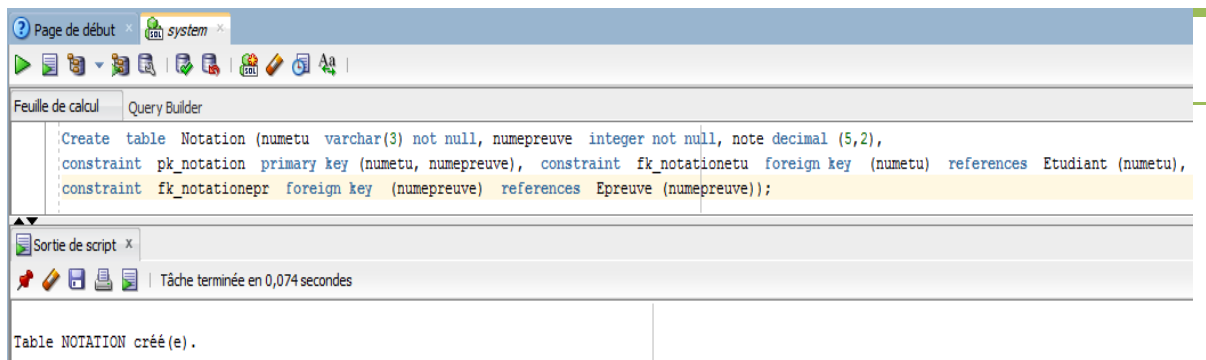
3. Pour montrer les épreuves réalisées à l'université, créer la table **Epreuve**. Pour distinguer les épreuves, on utilise la clé primaire **numepreuve** de type entier. Pour déterminer le lieu de l'épreuve, on utilise le champ **lieu**. Pour distinguer la matière impliquée par l'épreuve, on utilise la clé étrangère **codemat**.

Correction



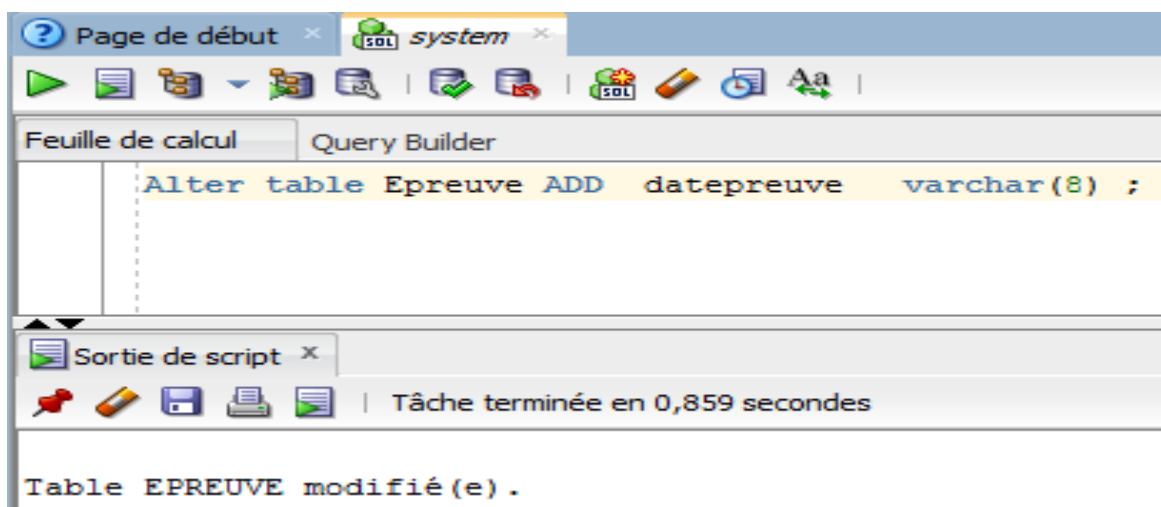
4. Pour mémoriser les notes des étudiants, créer la table **Notation**. Cette table comprend deux clés étrangères **numetu** et **numepreuve** contenant la valeur non nulle par défaut. La clé primaire de cette table contient les deux attributs **numetu** et **numepreuve**. Le champ **note** de type numérique (trois chiffres et deux décimales) est ajouté à cette table.

Correction



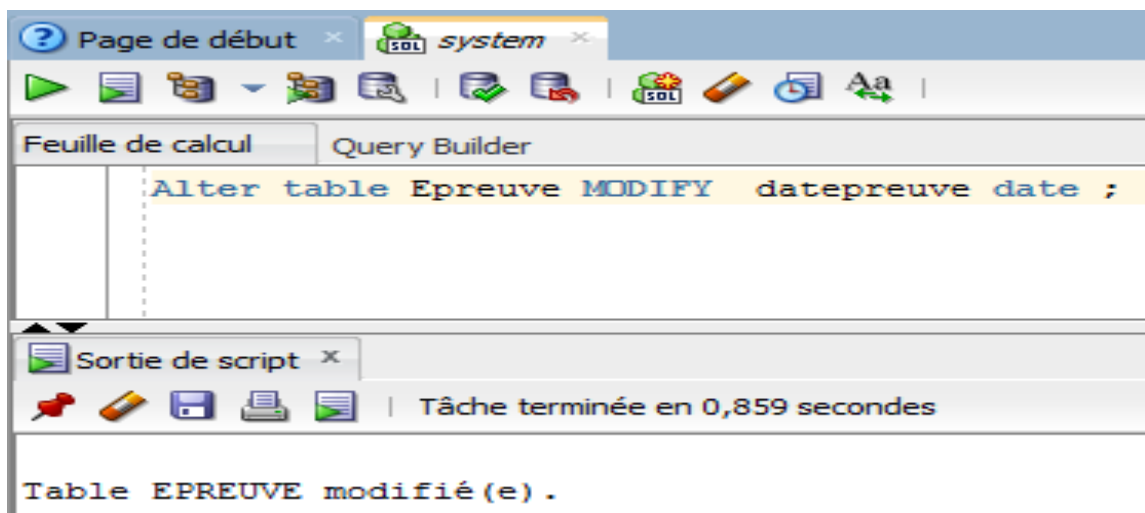
5. Pour mémoriser la date de l'épreuve de cette façon « 05122013 », ajouter un attribut **datepreuve** constitué de 8 caractères () dans la table Epreuve.

Correction



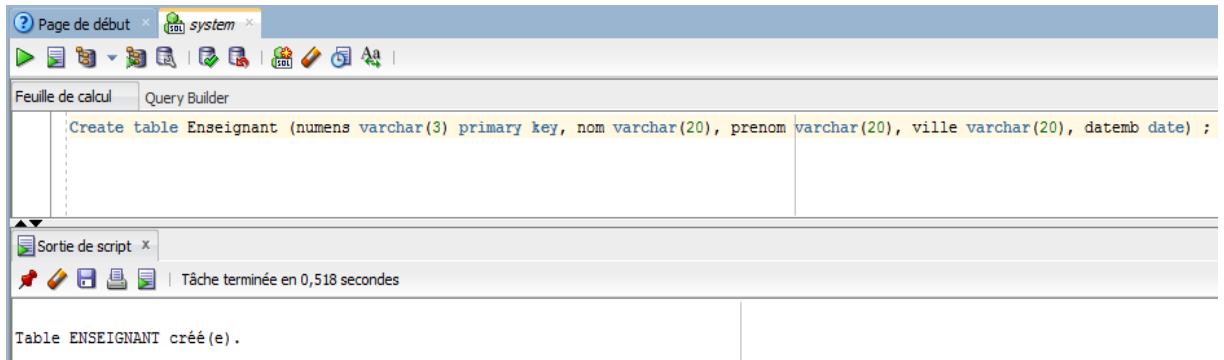
6. Modifier le type de l'attribut **datepreuve** de la table Epreuve au type date car l'attribution du type char n'est pas efficace.

Correction

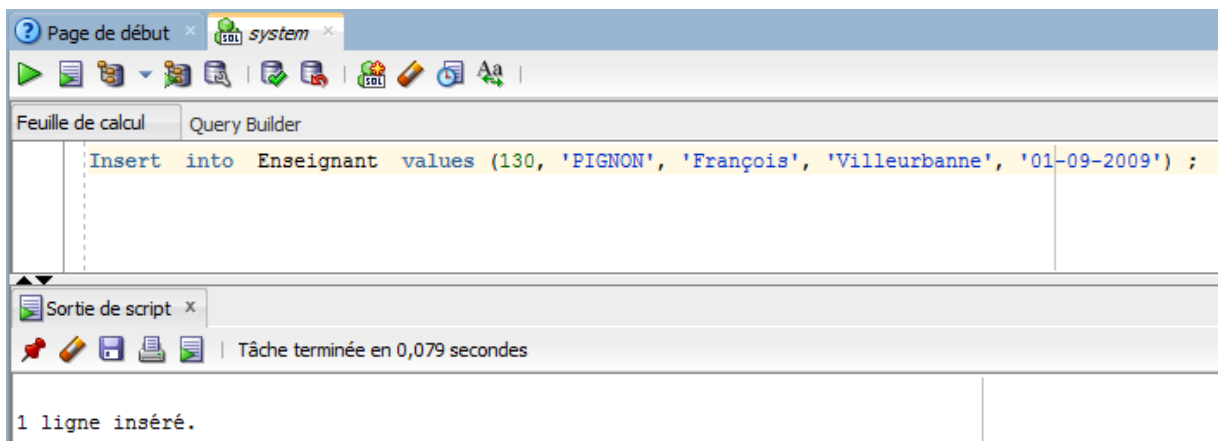


7. Pour montrer les enseignants de l'université, créer la table **Enseignant**. Cette table est identifiée par l'attribut **numens** formée de 3 caractères, un **nom** (20 caractères), un **prénom** (20 caractères) une **ville** représentant son adresse (20 caractères) et une date d'embauche **datemb** (date).

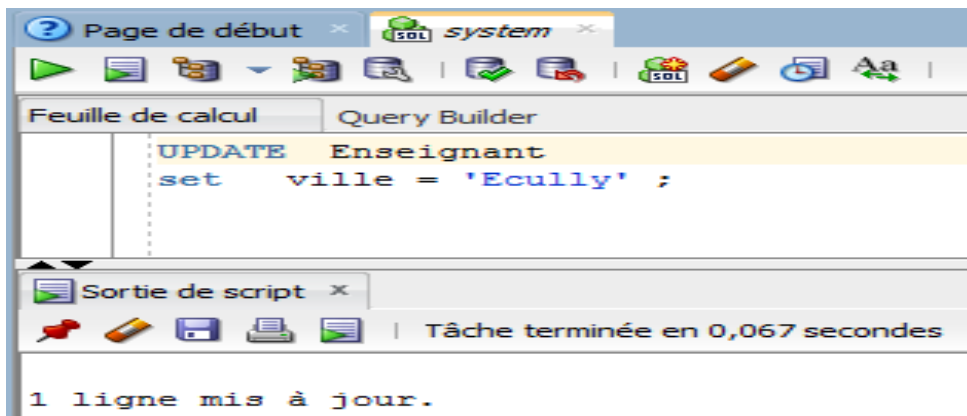
6

Correction

8. Dans la table **Enseignant**, ajouter l'enseignant François PIGNON. François est recruté le 01/09/2009 et habite à Villeurbanne et immatriculé par le numéro 130.

Correction

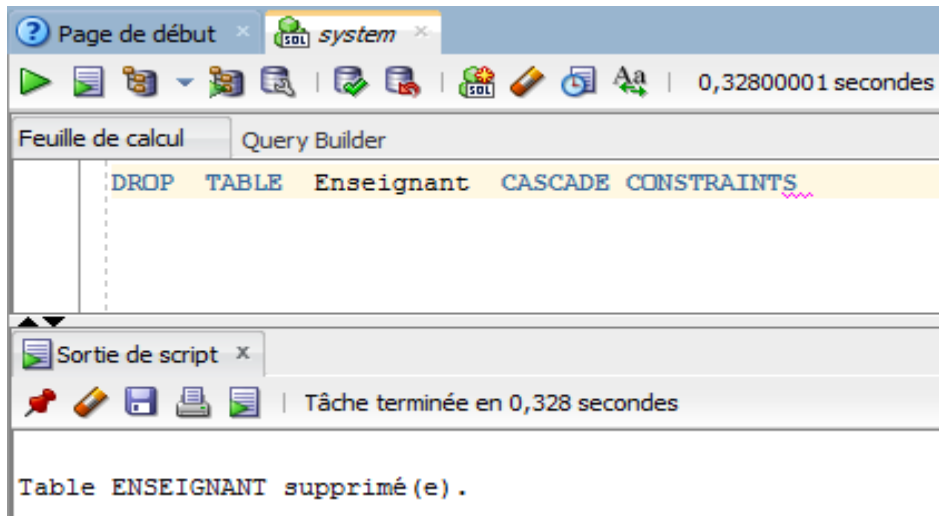
9. En annonçant son nouvelle adresse à Ecully, rectifier l'adresse de l'enseignant PIGNON.

Correction

10. Supprimer la table Enseignant.

Correction

7



Exercice 2

Considérons le schéma relationnel :

VILLE(No_Ville, NameV, CodePostal).

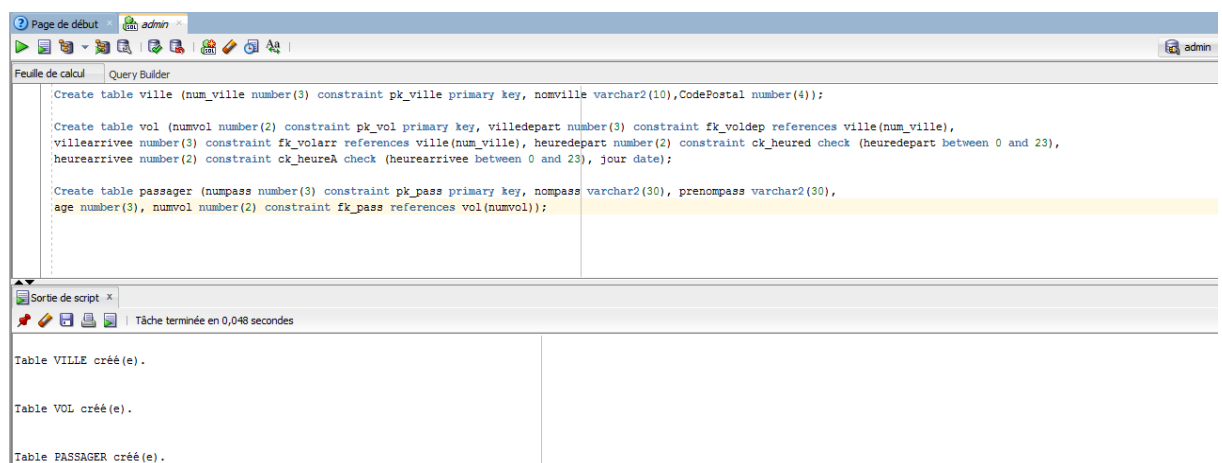
VOL(No_Vol, #VilleDepart, #VilleArrivee, HeureDepart, HeureArrivee, Jour).

PASSAGER(No_Pass, NamePass, PrenomPass, Age, #No_Vo).

NB : représenter les heures en utilisant un nombre de 0 à 23 sans minutes.

1- Créer les tables sans oublier les contraintes d'intégrité.

Correction



2- Intégrer les données suivantes dans la table ville : (15, SFAX, 3000), (16, PARIS, 75000).

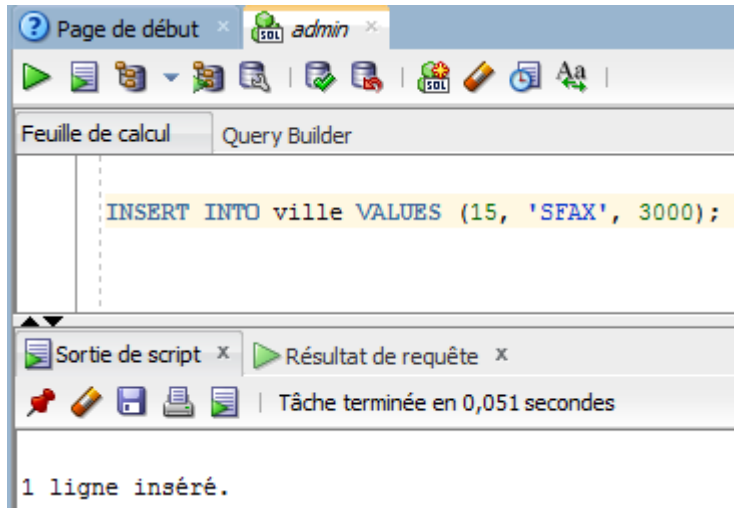
Intégrer les données suivantes dans la table VOL : (123, 15, 16, 9, 23, 31/12/2017).

Intégrer les données suivantes dans la table PASSAGER : (120, MHIRI, HEDI, 50, 123), (121, MAHMOUDI, ALI, 80, 123), (122, MASMOUDI, AHMED, 20, 123).

8

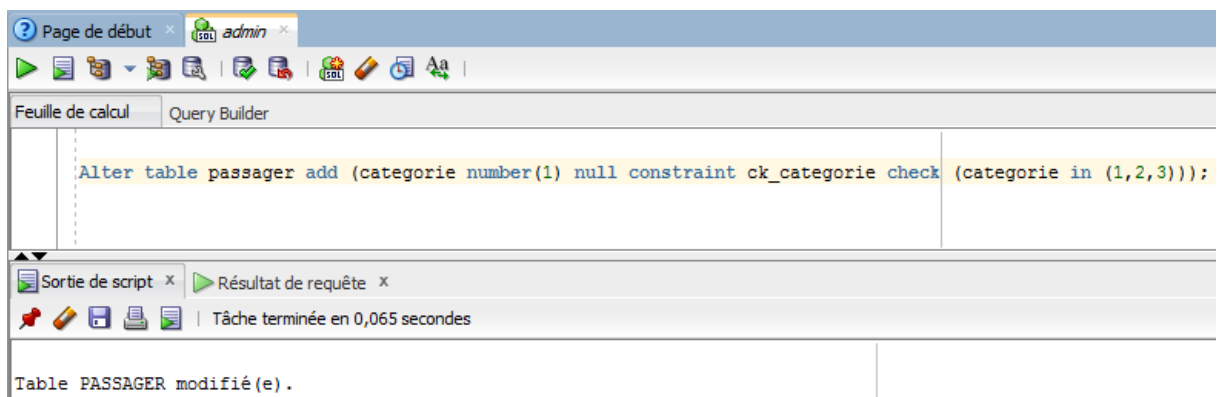
Correction

Pour l'insertion on va faire un exemple et les autres sont identiques.



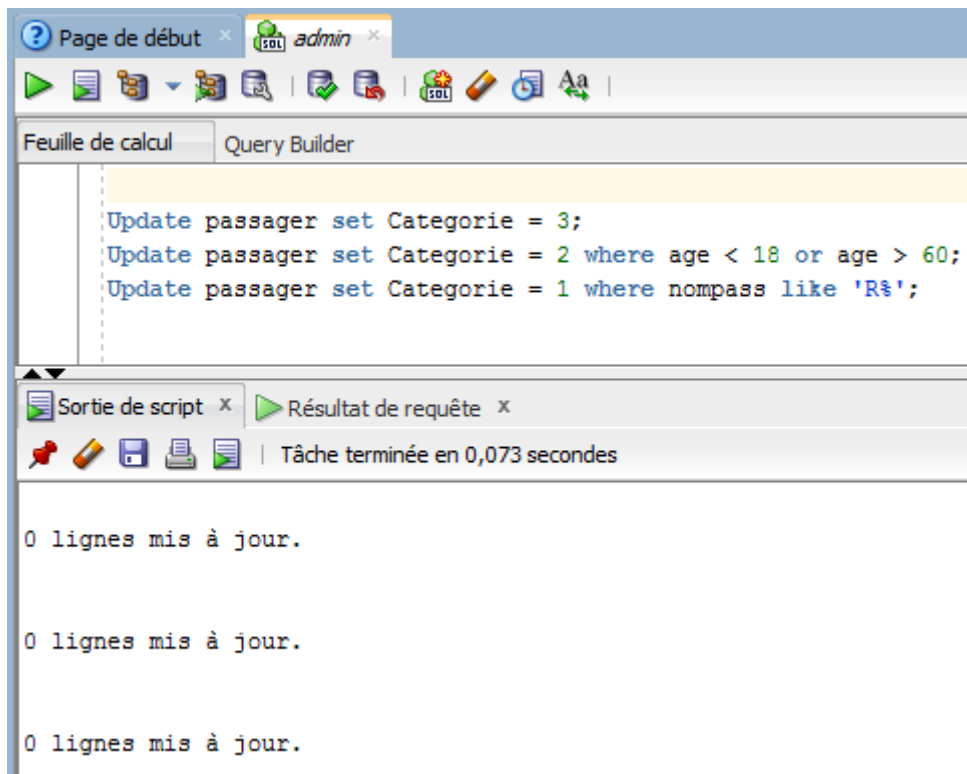
- 3- Ajouter l'attribut « Catégorie » contenant la valeur NULL par défaut pour bien développer la relation PASSAGER. On distingue 3 catégories de passagers : 1, 2 et 3.

Correction



- 4- En tapant les commandes SQL suivantes, modifier la table PASSAGER :
- Mettre en catégorie 2 tous les passagers dont l'âge est inférieur à 18 ans ou supérieur à 60 ans.
 - Mettre en catégorie 1 tous les passagers dont le nom débute par R.
 - Mettre en catégorie 3 le reste des passagers.

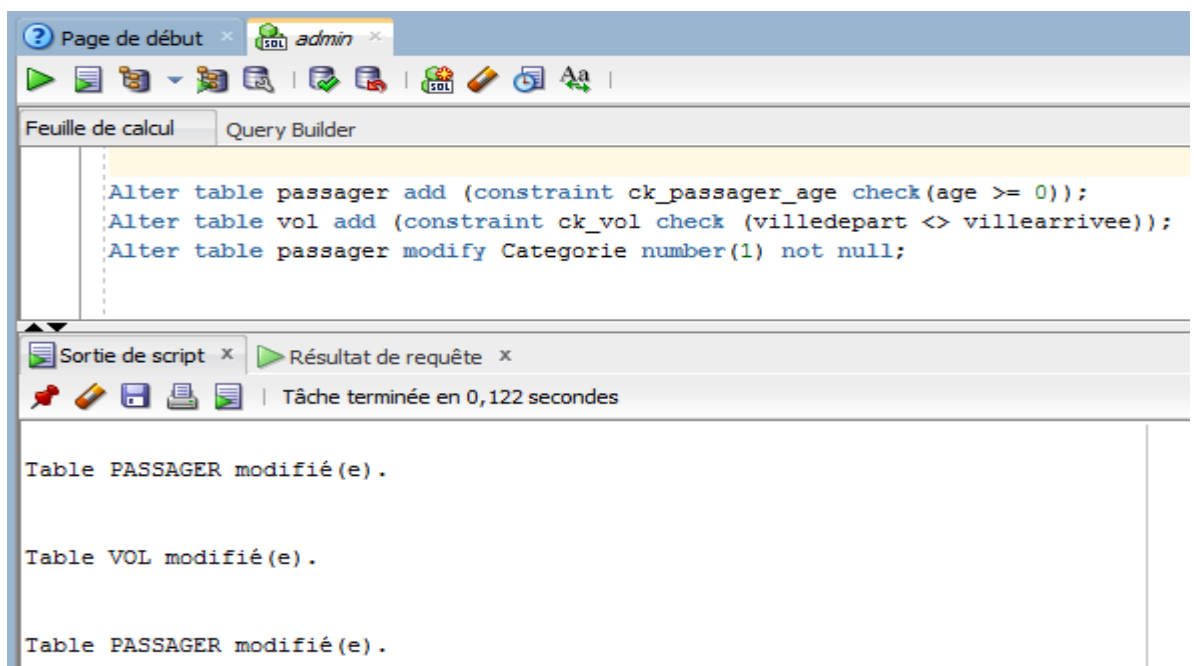
Correction



5- Sans oublier les contraintes d'intégrité mentionnées ci-dessous, mettre à jour les tables précédemment élaborées :

- L'âge d'un passager est supérieur à 0.
- Pas de vol entre une même ville.
- Il y a que trois catégories de passagers : 1, 2 et 3.

Correction



- 6- Augmenter votre modèle par l'ajout de la relation Catégorie (NumCat, Designation) et mettre à chaque catégorie un tuple. Comment répercuter intelligemment cet ajout sur les définitions des tables PASSAGER et VOL ?

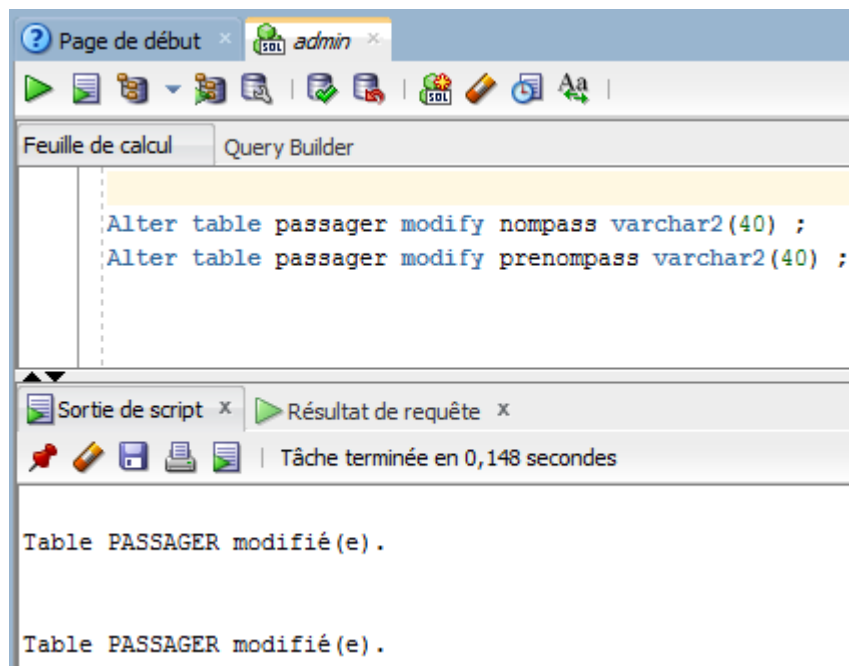
10

Correction

Creation table Categorrie contenant un cle primaire numcat et un autre colonne designation, puis insertion de trois lignes dans ce tableau.

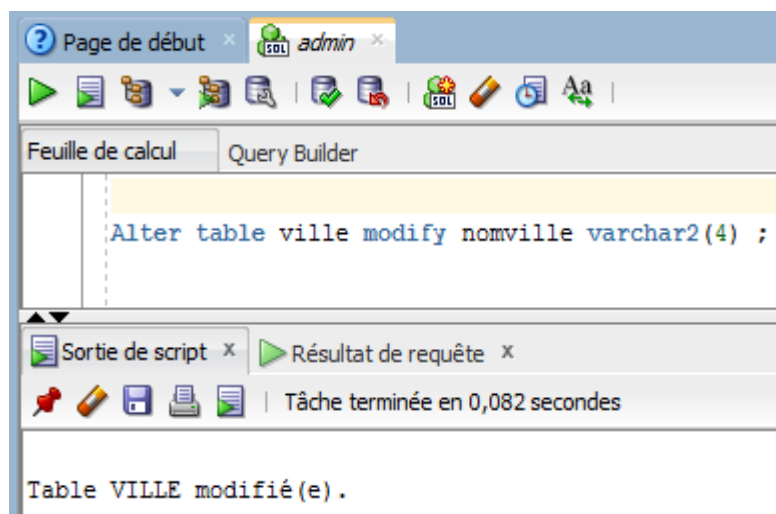
- 7- Ajouter 10 caractères pour élargir le type des attributs NomPass et PrenomPass.

Correction



- 8- Abaisser de 5 caractères le type de l'attribut NomVille. Imaginer le cas de troncatures.

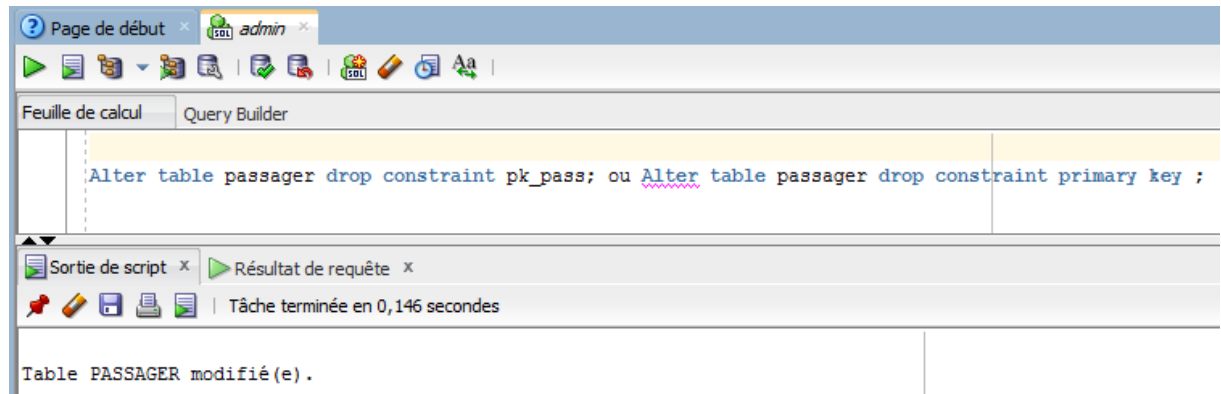
Correction



9- Sur l'attribut NumPass, retirer la contrainte de domaine.

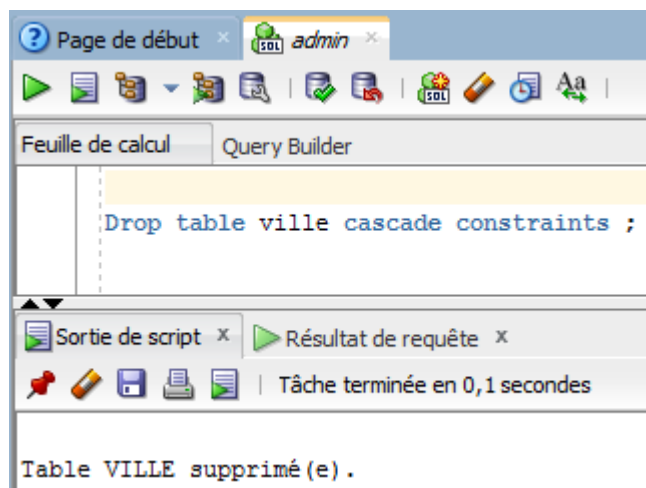
11

Correction



10- Par l'option CASCADE CONSTRAINTS, supprimer la table ville et tester son impact sur les tables VOL et PASSAGER ;

Correction



TP2 : LANGAGE DE MANIPULATION DES DONNEES (LMD)

12

Objectif : le TP sur le LMD en SQL vise à enseigner aux étudiants comment interagir avec les bases de données relationnelles.

Exercice

Considérons une partie du schéma de la base « gestion des employés » suivante :

DEPT (DEPTNO, DNAME, LOC)

EMP (EMPNO, ENAME, JOB, #MGR, HIREDATE, SAL, COMM, #DEPTNO)

SALGRADE (# DEPTNO, # EMPNO, GRADE, LOSAL, HISAL)

Le contenu de ces Tables est le suivant :

Tables des employés : EMP

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7360	SMITH	CLERK	7902	17-DEC-80	800		20
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7698	BLAKE	MANAGER	7839	01-MAY-81	2850		30
7782	CLARK	MANAGER	7839	09-JUN-81	2450		10
7788	SCOTT	ANALYST	7566	09-DEC-82	3000		20
7839	KING	PRESIDENT		17-NOV-81	5000		10
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7876	ADAMS	CLERK	7788	12-JAN-83	1100		20
7900	JAMES	CLERK	7698	03-DEC-81	950		30
7902	FORD	ANALYST	7566	03-DEC-81	3000		20
7934	MILLER	CLERK	7782	23-JAN-82	1300		10

Tables des départements : DEPT

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

Tables des grades de salaire : SALGRADE

GRADE	LOSAL	HISAL
1	700	1200
2	1201	1400
3	1401	2000
4	2001	3000
5	3001	9999

Remarque : un même domaine sémantique définit par EMP.EMPNO et EMP.MGR, MGR représente l'identifiant le supérieur hiérarchique d'un employé est identifié par MGR.

- 1- Sans oublier les contraintes d'intégrité, créer les tables en tapant les commandes LDD nécessaires.

```

-- Création de la table DEPT
CREATE TABLE DEPT (DEPTNO INT PRIMARY KEY, DNAME VARCHAR(255), LOC VARCHAR(255));

-- Création de la table EMP
CREATE TABLE EMP (EMPNO INT PRIMARY KEY, ENAME VARCHAR(255), JOB VARCHAR(255), MGR INT,
HIREDATE DATE, SAL DECIMAL(10, 2), COMM DECIMAL(10, 2), DEPTNO INT,
FOREIGN KEY (MGR) REFERENCES EMP(EMPNO),
FOREIGN KEY (DEPTNO) REFERENCES DEPT(DEPTNO)
);

-- Création de la table SALGRADE
CREATE TABLE SALGRADE (DEPTNO INT, EMPNO INT, GRADE INT, LOSAL DECIMAL(10, 2),
HISAL DECIMAL(10, 2),
PRIMARY KEY (DEPTNO, EMPNO),
FOREIGN KEY (DEPTNO) REFERENCES DEPT(DEPTNO),
FOREIGN KEY (EMPNO) REFERENCES EMP(EMPNO)
);

```

Sortie de script x

Tâche terminée en 1,145 secondes

```

Table DEPT créé(e) .

Table EMP créé(e) .

Table SALGRADE créé(e) .

```

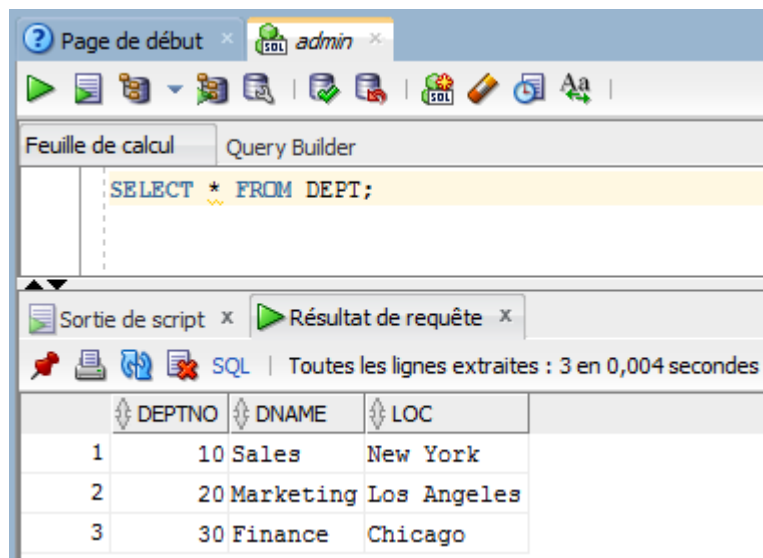
- 2- En adoptant SQL, exprimer les requêtes suivantes en SQL :

14

Interrogation d'une seule relation

1. Donner toutes les informations de la table DEPT.

Correction

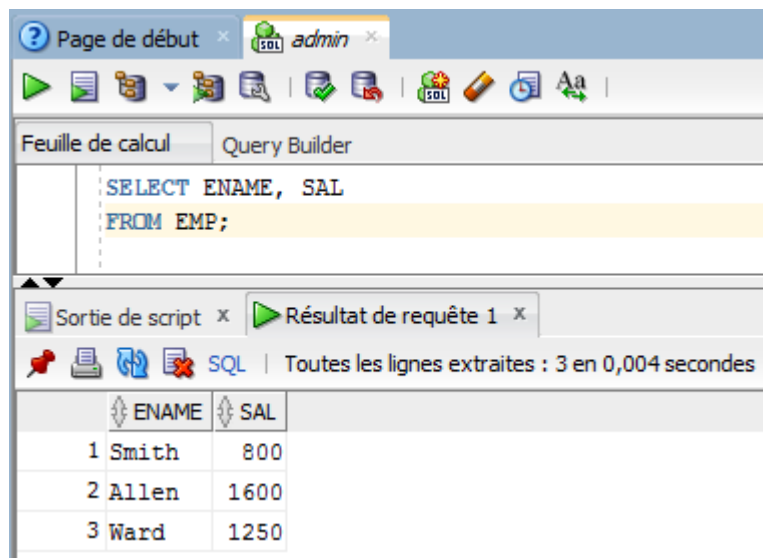


The screenshot shows the SQL Developer interface. The 'Query Builder' tab is active, displaying the SQL query: `SELECT * FROM DEPT;`. Below the query, the 'Résultat de requête' (Query Result) tab shows the data extracted from the DEPT table. The status bar indicates 'Toutes les lignes extraites : 3 en 0,004 secondes' (All lines extracted: 3 in 0.004 seconds).

	DEPTNO	DNAME	LOC
1	10	Sales	New York
2	20	Marketing	Los Angeles
3	30	Finance	Chicago

2. Afficher les noms et les salaires des employés.

Correction

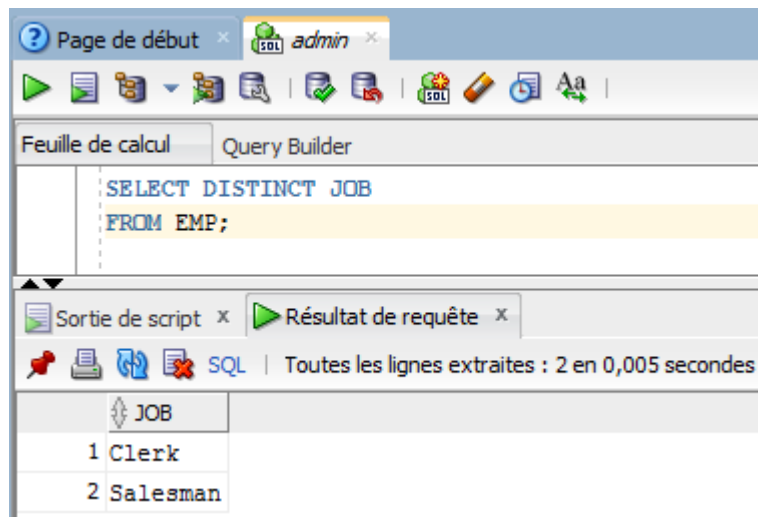


The screenshot shows the SQL Developer interface. The 'Query Builder' tab is active, displaying the SQL query: `SELECT ENAME, SAL FROM EMP;`. Below the query, the 'Résultat de requête 1' (Query Result 1) tab shows the data extracted from the EMP table. The status bar indicates 'Toutes les lignes extraites : 3 en 0,004 secondes' (All lines extracted: 3 in 0.004 seconds).

	ENAME	SAL
1	Smith	800
2	Allen	1600
3	Ward	1250

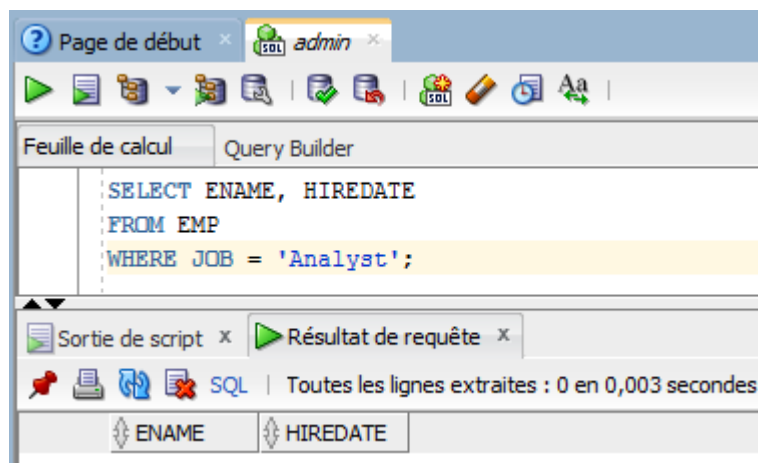
3. Après élimination des doublons, afficher les professions des employés.

Correction



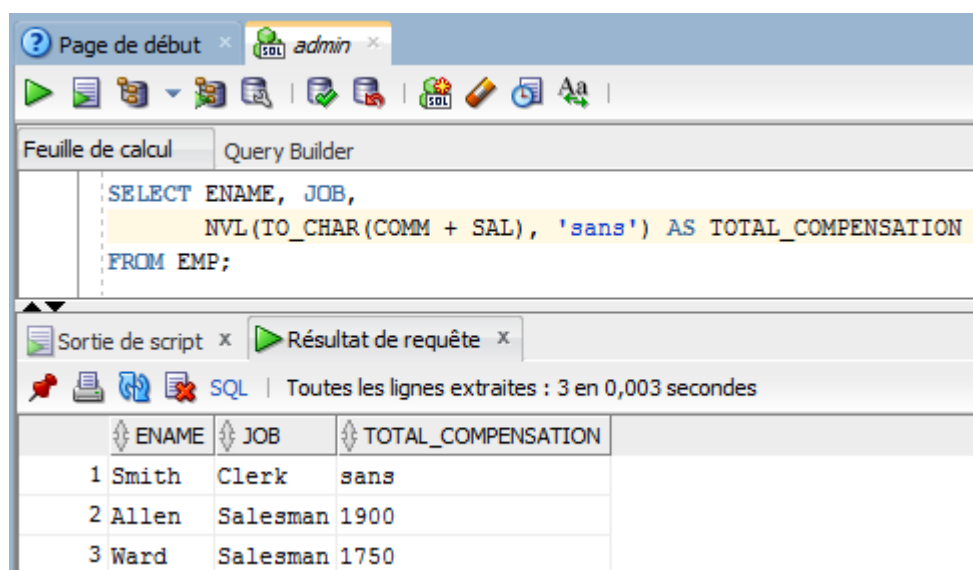
4. Pour les analystes, afficher Les dates de recrutement.

Correction



5. En marquant le mot 'sans' pour ceux qui n'ont pas de commission, afficher ENAME, JOB, COMM, COMM+SAL pour tout employé.

Correction



6. Dans le département 10, lister tous les employés en changeant les noms par « **** ».

16

Correction

The screenshot shows the SQL Developer interface. The Query Builder tab is active, displaying the following SQL query:

```
SELECT
  DECODE(DEPTNO, 10, '****', ENAME) AS ENAME,
  JOB,
  MGR,
  HIREDATE,
  SAL,
  COMM,
  DEPTNO
FROM EMP;
```

The Results window shows the output of the query, with 3 rows extracted in 0.005 seconds:

	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	Smith	Clerk	(null)	1980-12-17	800	(null)	20
2	Allen	Salesman	7369	1981-02-20	1600	300	30
3	Ward	Salesman	7499	1981-02-22	1250	500	30

7. Donner le nom, la fonction et le salaire des employés en changeant respectivement par « *** » et « ### » le salaire des président et manager.

Correction

The screenshot shows the SQL Developer interface. The Query Builder tab is active, displaying the following SQL query:

```
SELECT
  ENAME,
  JOB,
  DECODE(JOB, 'President', '***', 'Manager', '###', SAL) AS SALARY
FROM EMP;
```

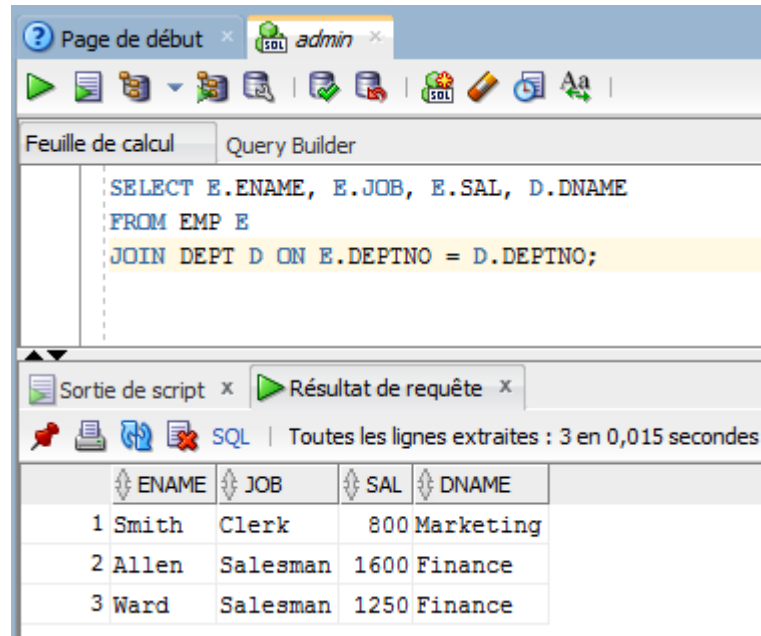
The Results window shows the output of the query, with 3 rows extracted in 0.004 seconds:

	ENAME	JOB	SALARY
1	Smith	Clerk	800
2	Allen	Salesman	1600
3	Ward	Salesman	1250

Jointures et imbrications

17

1. Afficher les employés et les noms de leurs départements.

Correction


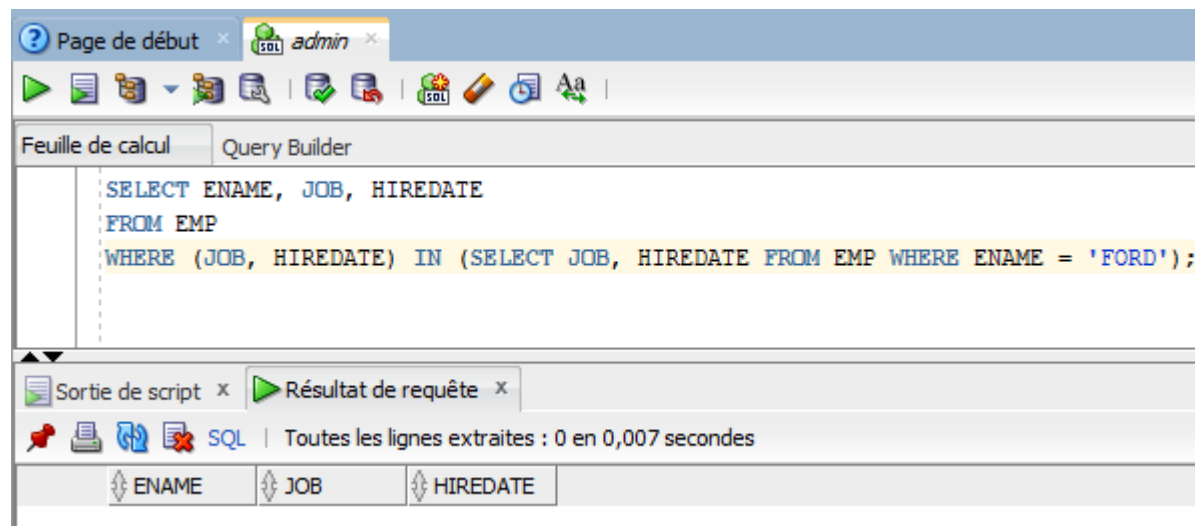
The screenshot shows the SQL Developer interface. The 'Query Builder' tab is active, displaying the following SQL query:

```
SELECT E.ENAME, E.JOB, E.SAL, D.DNAME
FROM EMP E
JOIN DEPT D ON E.DEPTNO = D.DEPTNO;
```

Below the query, the 'Résultat de requête' (Query Result) tab shows the results of the query. The status bar indicates 'Toutes les lignes extraites : 3 en 0,015 secondes' (All lines extracted: 3 in 0.015 seconds).

	ENAME	JOB	SAL	DNAME
1	Smith	Clerk	800	Marketing
2	Allen	Salesman	1600	Finance
3	Ward	Salesman	1250	Finance

2. Afficher les employés qui possèdent le même poste et la même date d'embauche que « FORD ».

Correction


The screenshot shows the SQL Developer interface. The 'Query Builder' tab is active, displaying the following SQL query:

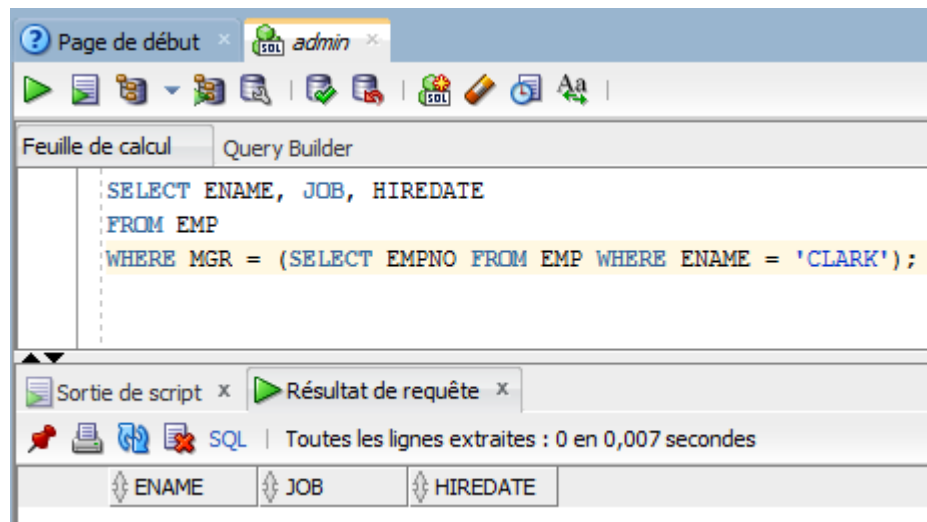
```
SELECT ENAME, JOB, HIREDATE
FROM EMP
WHERE (JOB, HIREDATE) IN (SELECT JOB, HIREDATE FROM EMP WHERE ENAME = 'FORD');
```

Below the query, the 'Résultat de requête' (Query Result) tab shows the results of the query. The status bar indicates 'Toutes les lignes extraites : 0 en 0,007 secondes' (All lines extracted: 0 in 0.007 seconds).

ENAME	JOB	HIREDATE
-------	-----	----------

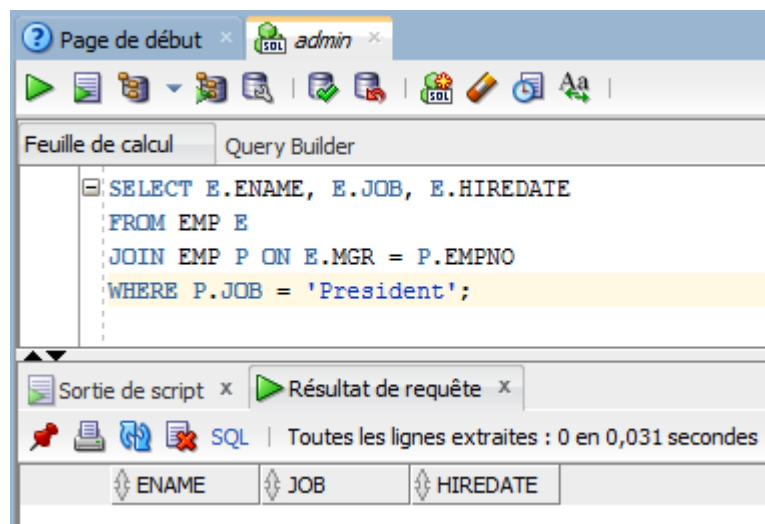
3. Afficher les employés qui possèdent le même supérieur que 'CLARK'.

Correction



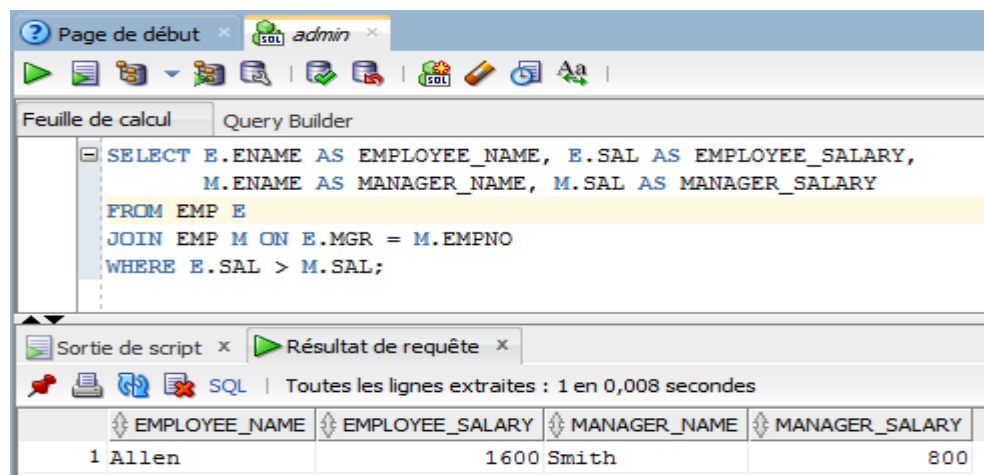
4. Afficher Les employés qui possèdent directement comme chef le 'Président'.

Correction



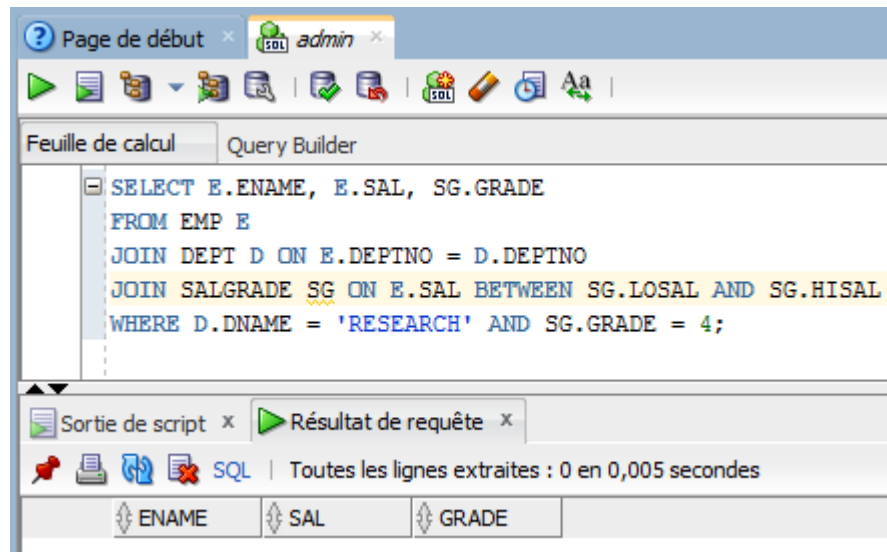
5. Afficher les noms et salaires des employés qui possèdent un salaire supérieur à celui de leurs chefs ainsi que les noms et les salaires de leurs chefs.

Correction



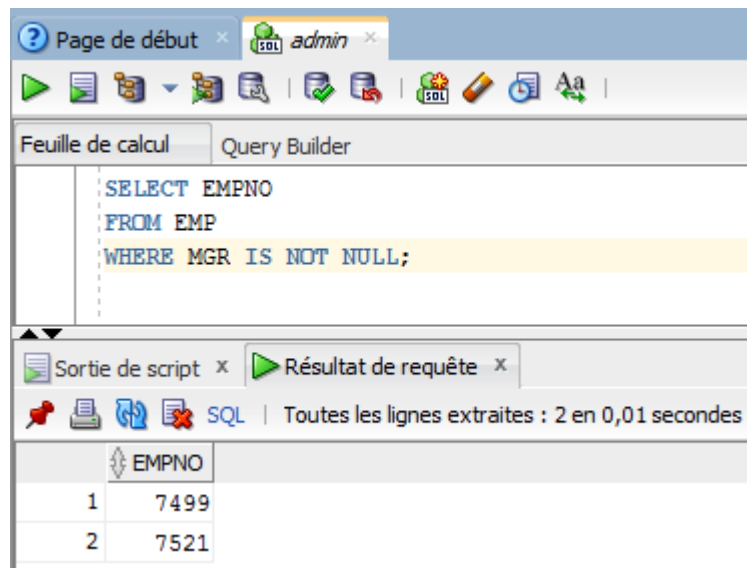
6. dans le département 'RESEARCH', afficher les employés qui possèdent un grade de salaire = 4.

Correction



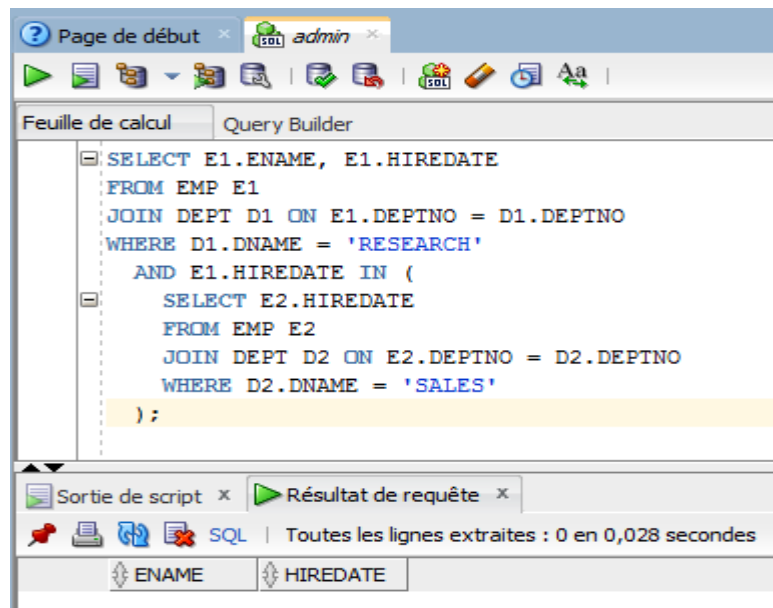
7. Afficher les employés (Numéros) qui possèdent des chefs.

Correction



8. Dans le département 'RESEARCH', afficher les employés embauchés dans la même date qu'un employé dans le département 'SALES'.

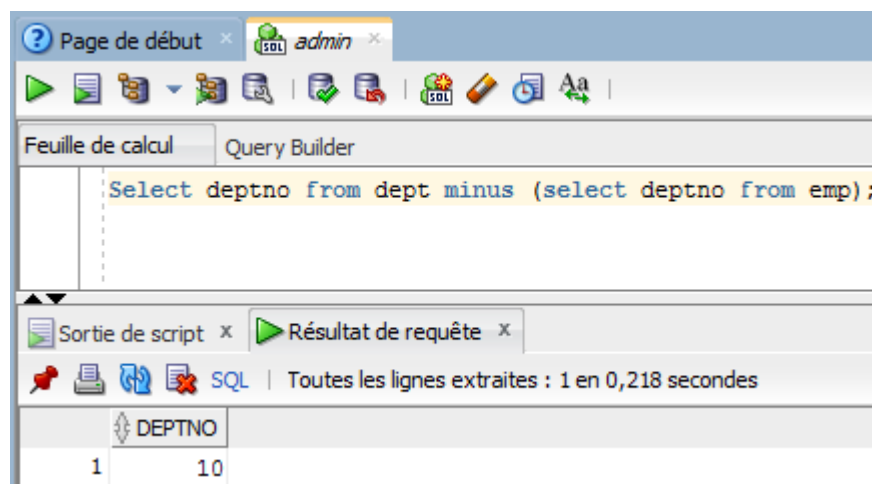
Correction



Opérateurs ensemblistes

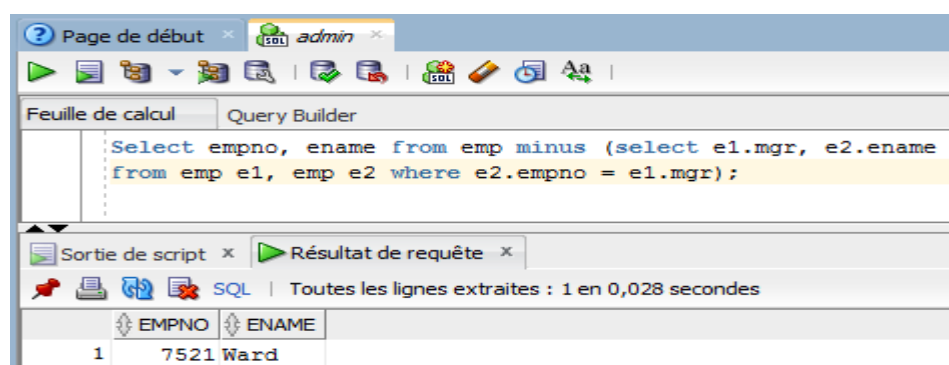
1. Afficher Les numéros des départements où il n'existe pas d'employés.

Correction



2. Afficher les numéros et les noms des employés qui ne possèdent pas de chefs.

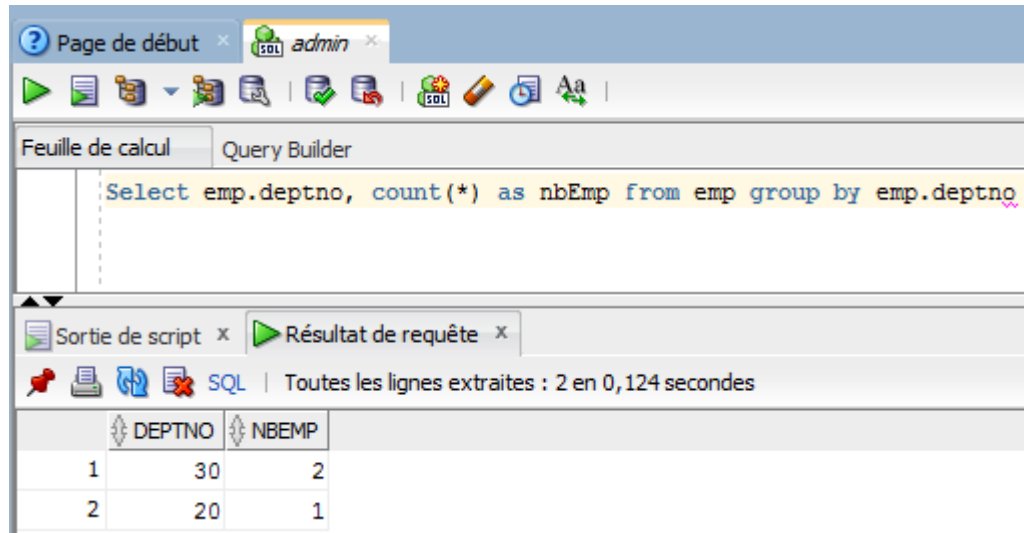
Correction



Tris, Groupes et agrégats

21

1. Pour chaque département, afficher le nombre d'employés.

Correction


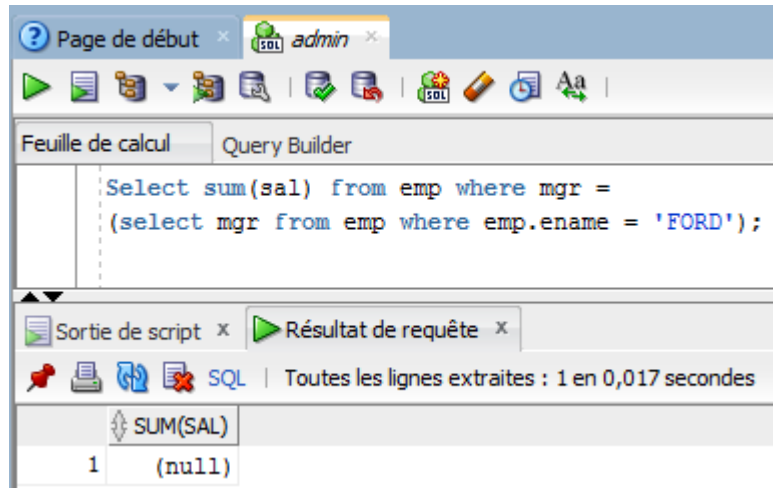
The screenshot shows the SQL Developer interface. The 'Query Builder' tab is active, displaying the following SQL query:

```
Select emp.deptno, count(*) as nbEmp from emp group by emp.deptno
```

Below the query editor, the 'Résultat de requête' (Query Result) tab shows the results of the query. The status bar indicates 'Toutes les lignes extraites : 2 en 0,124 secondes'.

	DEPTNO	NBEMP
1	30	2
2	20	1

2. Pour les employés qui possèdent le même chef direct que FORD, afficher le total des salaires.

Correction


The screenshot shows the SQL Developer interface. The 'Query Builder' tab is active, displaying the following SQL query:

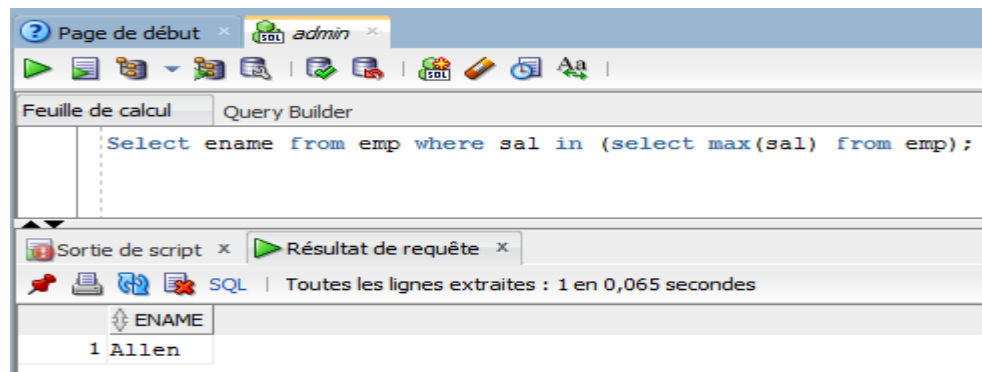
```
Select sum(sal) from emp where mgr =  
(select mgr from emp where emp.ename = 'FORD');
```

Below the query editor, the 'Résultat de requête' (Query Result) tab shows the results of the query. The status bar indicates 'Toutes les lignes extraites : 1 en 0,017 secondes'.

	SUM(SAL)
1	(null)

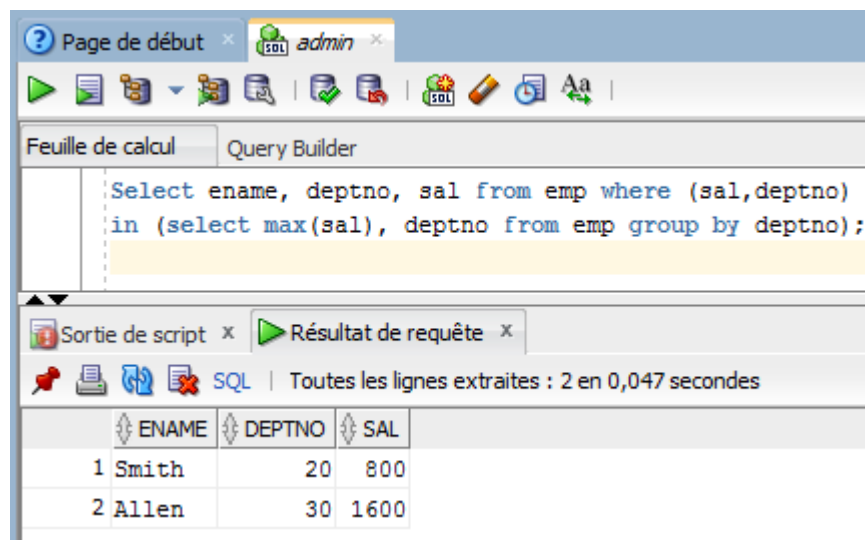
3. Pour les employés qui possèdent le plus gros salaire, afficher leurs Nom.

Correction



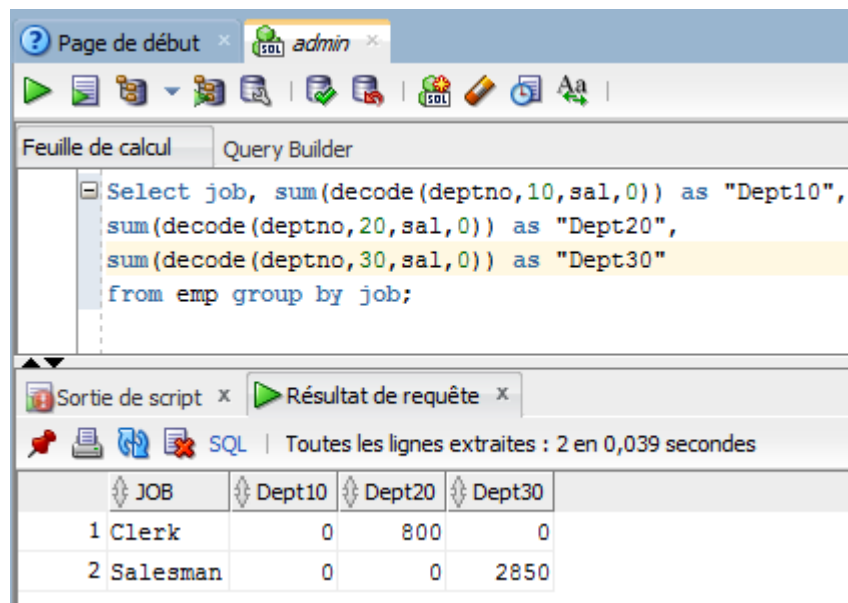
4. Pour chaque département, afficher les noms des employés ayant le plus gros salaire.

Correction



5. En utilisant la façon présentée dans le tableau, afficher les totaux des salaires par poste et par département.

Poste	Dept 10	Dept 20	Dept 30
Secrétaire	100000	20000	150000
Ingénieur	80000	130000	200000

Correction


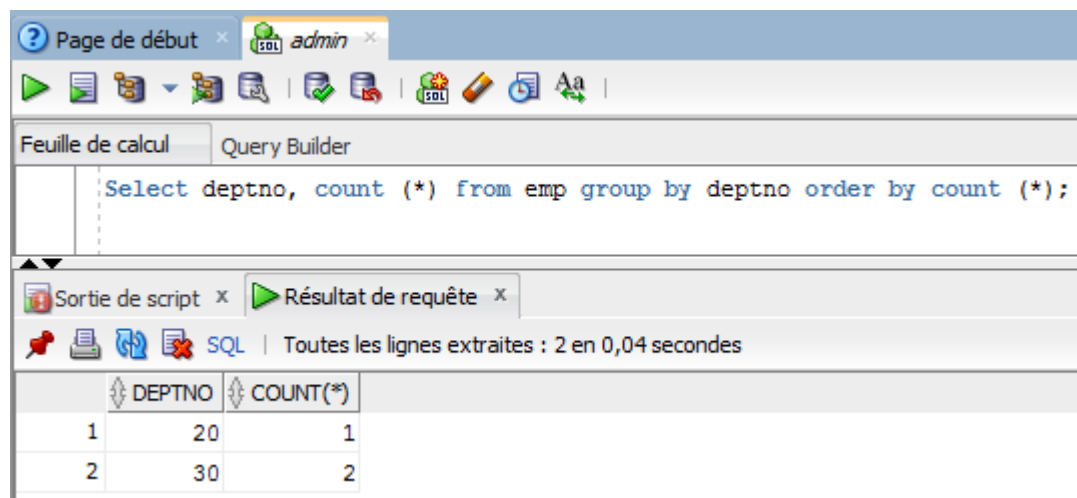
The screenshot shows the SQL Developer interface. The 'Query Builder' tab is active, displaying the following SQL query:

```
Select job, sum(decode(deptno,10,sal,0)) as "Dept10",
sum(decode(deptno,20,sal,0)) as "Dept20",
sum(decode(deptno,30,sal,0)) as "Dept30"
from emp group by job;
```

The 'Résultat de requête' (Query Result) tab shows the results of the query, with 2 rows extracted in 0.039 seconds. The results are as follows:

	JOB	Dept10	Dept20	Dept30
1	Clerk	0	800	0
2	Salesman	0	0	2850

6. En respectant le tri selon le nombre d'employés, afficher les numéros de département.

Correction


The screenshot shows the SQL Developer interface. The 'Query Builder' tab is active, displaying the following SQL query:

```
Select deptno, count (*) from emp group by deptno order by count (*);
```

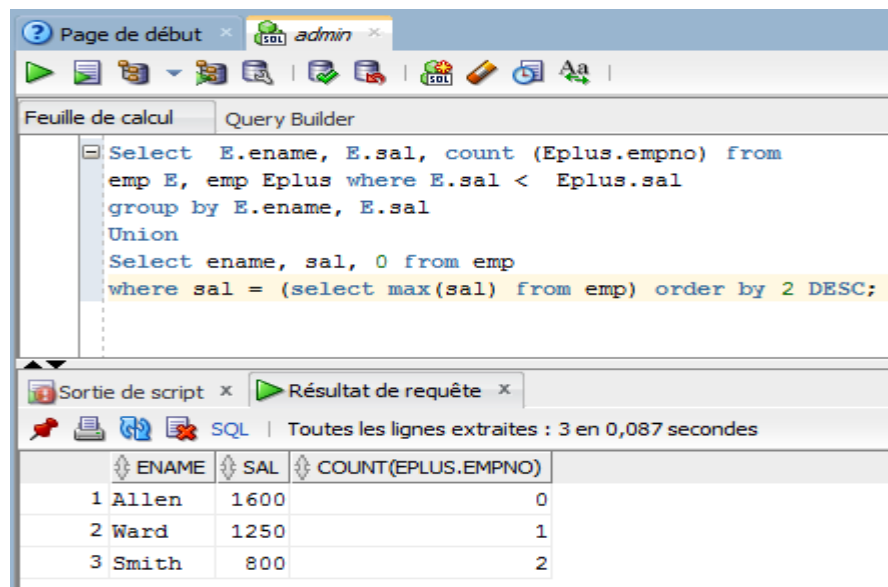
The 'Résultat de requête' (Query Result) tab shows the results of the query, with 2 rows extracted in 0.04 seconds. The results are as follows:

	DEPTNO	COUNT(*)
1	20	1
2	30	2

7. En respectant le tri par salaire décroissant, afficher la liste des employés, avec, pour chaque employé le nombre d'employés ayant le plus gros salaire.

Correction

24



The screenshot shows a SQL query editor window with a tab labeled 'admin'. The query is as follows:

```
Select E.ename, E.sal, count (Eplus.empno) from  
emp E, emp Eplus where E.sal < Eplus.sal  
group by E.ename, E.sal  
Union  
Select ename, sal, 0 from emp  
where sal = (select max(sal) from emp) order by 2 DESC;
```

Below the query editor, the 'Résultat de requête' (Query Result) tab is active, displaying the results of the query. The results are shown in a table with 4 columns: an index, ENAME, SAL, and COUNT(EPLUS.EMPNO). The table contains 3 rows of data.

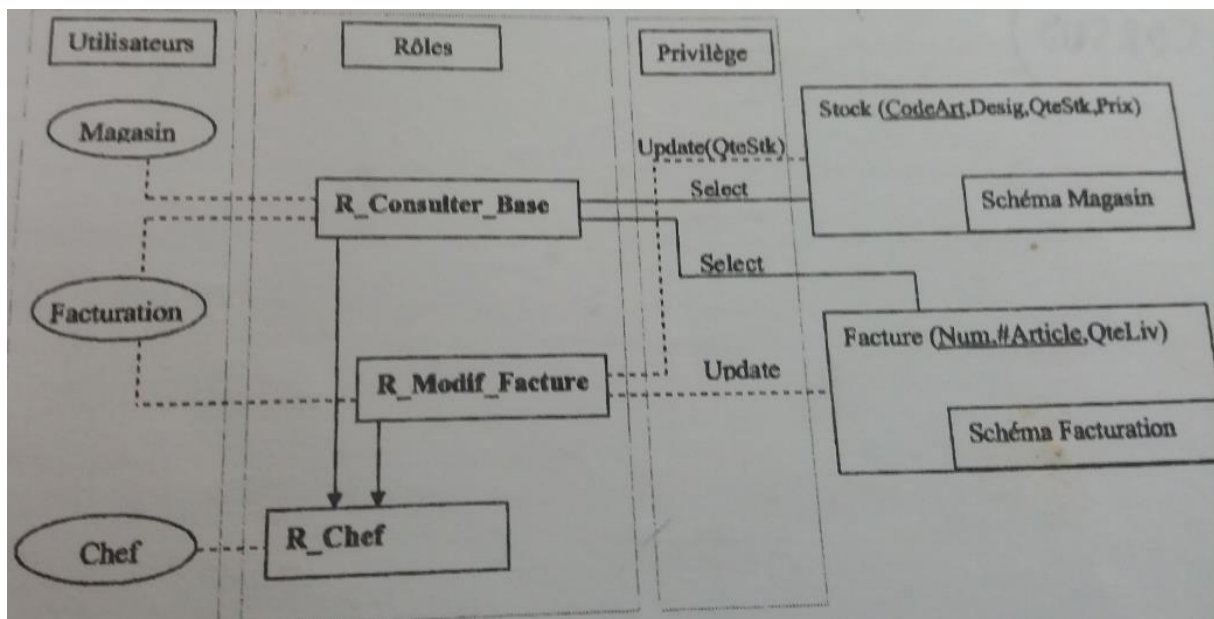
	ENAME	SAL	COUNT(EPLUS.EMPNO)
1	Allen	1600	0
2	Ward	1250	1
3	Smith	800	2

TP3 : LANGAGE DE CONTROLE DE DONNEES (LCD)

25

Objectif : le TP sur le LCD en SQL vise à enseigner aux étudiants comment créer des utilisateurs et des rôles et affecter des droits d'accès aux utilisateurs.

Exercice



1. Elaborer les utilisateurs Magasin, Facturation et Chef.

Correction

Page de début x admin x

Feuille de calcul Query Builder

```
Create user Magasin identified by 123;
Create user Facturation identified by 200;
Create user Chef identified by 210;
```

Sortie de script x Résultat de requête x

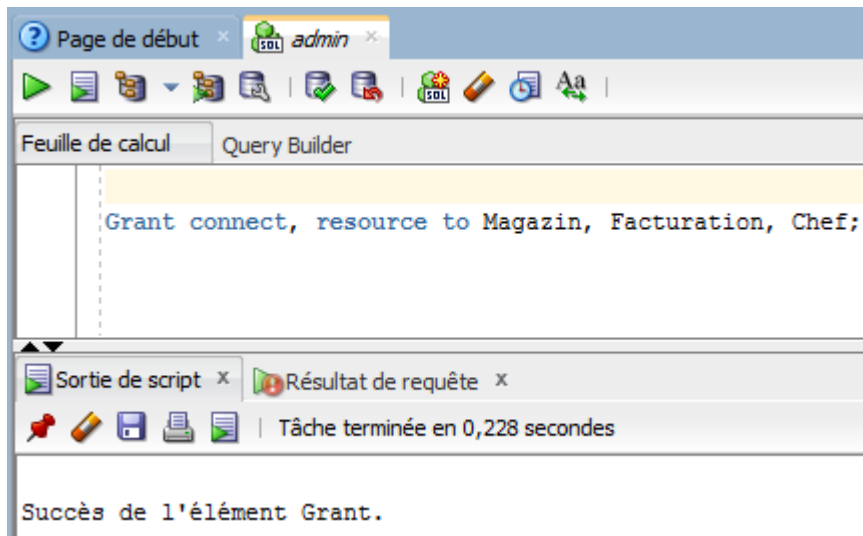
Tâche terminée en 0,41 secondes

```
User MAGAZIN créé(e) .
User FACTURATION créé(e) .
User CHEF créé(e) .
```

2. Faire la modification des privilèges des utilisateurs précédemment élaborés pour qu'ils disposent les rôles de CONNECT et RESOURCE.

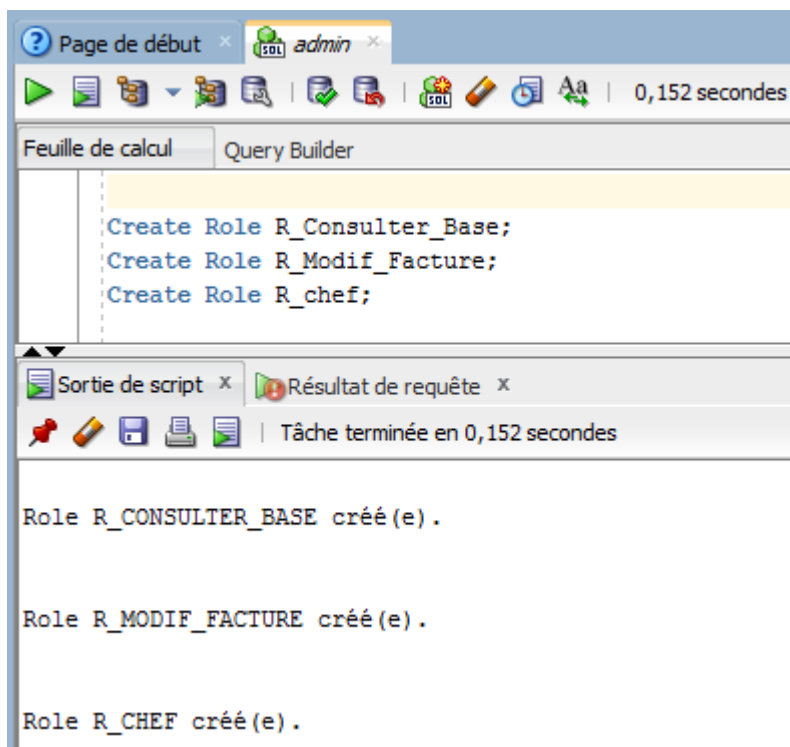
26

Correction



3. Elaborer les rôles R_Consulter_Base, R_Modif_Facture et R_Chef.

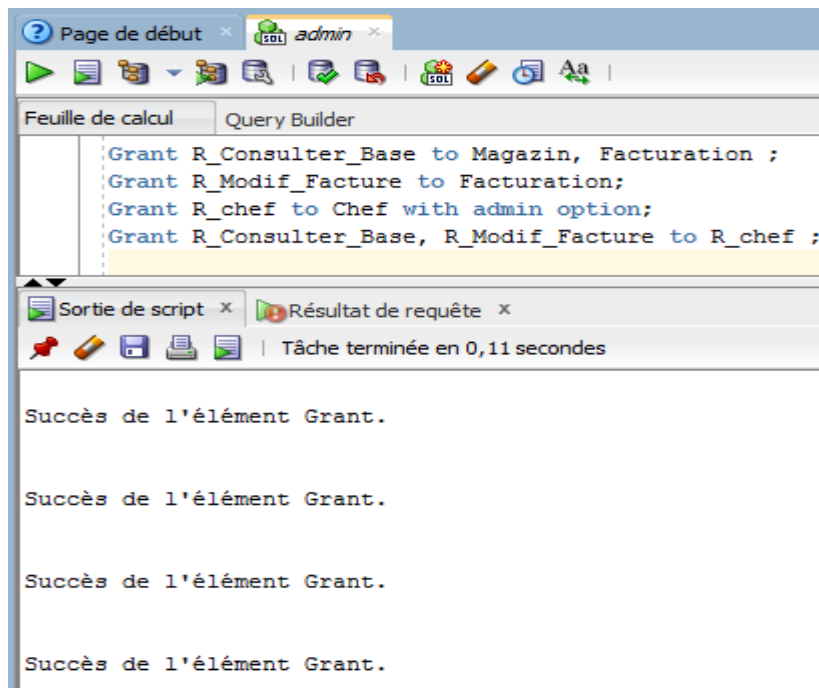
Correction



4. Affecter à chaque utilisateur les rôles associés :
- Rôles utilisateur **Magasin** : R_Consulter_Base,
 - Rôles utilisateur **Facturation** : R_Consulter_Base et R_Modif_Facture,
 - Rôles utilisateur **Chef** : R_Chef,
 - Le rôle **R_Chef** possède les rôles R_Consulter_Base, R_Modif_Facture.

Correction

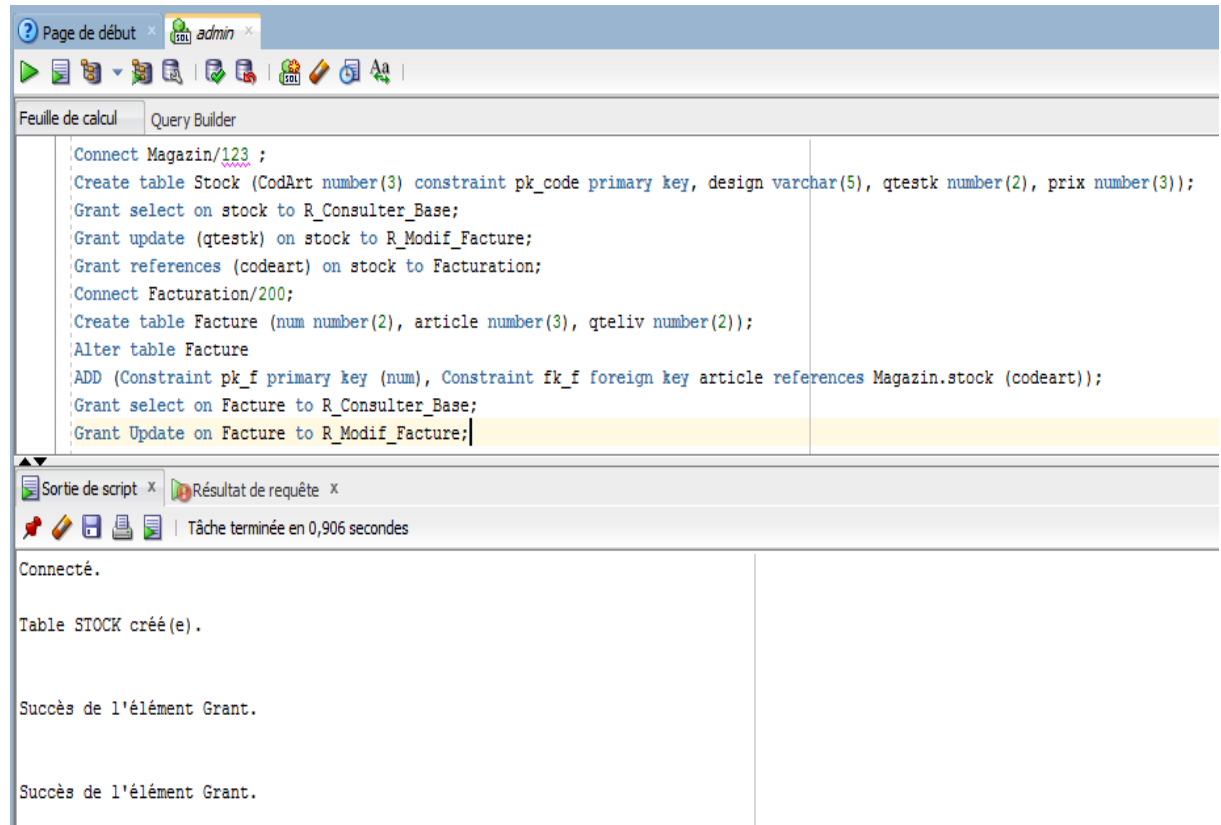
27



5. Elaborer les deux tables **Stock** dans le schéma de **Magazin** et **Facture** dans le schéma de **Facturation**,

Donner à **R_Consumer_Base**, **R_Modif_Facture** et **R_Chef** les privilèges concernés :

- Privilège **Select** appliqué à la table **Stock** et à la table **Facture** pour le rôle **R_Consumer_Base**,
- Privilège **Update** de l'attribut (**QteStk**) appliqué à la table **Stock** et **Update** appliqué à la table **Facture** pour le rôle **R_Modif_Facture**).

Correction

6. Par l'option CASCADE, Supprimer l'utilisateur Magazin et tester l'impact sur les rôles R_Consumer_Base, R_Modif_Facture et R_Chef.

NB : L'utilisateur Chef, seulement, admet les droits et capable de les transférer à son tour.

Correction