

**ASGBD**

**Rapport du TP N°1**

**CHIKH Khadidja**

Master 1 SII  
Groupe 4

## PARTIE 1:

### 1. Creation des tablespaces :

```
SQL> CREATE TABLESPACE hopital_tbs DATAFILE 'c:\tbs_hopital.dat' SIZE 100M AUTOEXTEND ON ONLINE;  
Tablespace created.  
  
SQL> CREATE TEMPORARY TABLESPACE hopital_Temptbs TEMPFILE 'c:\temptbs_hopital.dat' SIZE 100M AUTOEXTEND ON;  
Tablespace created.
```

### 2. Creation de l'utilisateur DBAHOPITAL en lui attribuant les deux tablespaces précédents:

```
SQL> Create User dbahopital Identified by chikh Default Tablespace hopital_tbs  
2 Temporary Tablespace hopital_Temptbs;  
  
User created.
```

### 3. Attribuer tous les privilèges à DBAHOPITAL:

```
SQL> grant all privileges to dbahopital;  
  
Grant succeeded.
```

## PARTIE 2:

### 4. Toutes les clés étrangères:

#### La relation «Chambre»:

- ✓ «code\_service» est une clé étrangère référençant «Service» parce qu'elle représente le code du service où est situé la chambre.
- ✓ «surveillant» est une clé étrangère référençant «Infirmier» puisqu'elle représente le numéro d'un infirmier.

#### La relation «Medecin»:

- ✓ «num\_med» est une clé étrangère référençant le numero d'employé dans «Employe».

#### La relation «Infirmier»:

- ✓ «code\_service» est une clé étrangère référençant «Service» parce qu'elle représente le code du service où est affecté un infirmier.
- ✓ «num\_inf» est une clé étrangère référençant le numero d'employé dans «Employe».

#### La relation «Hospitalisation»:

- ✓ «num\_patient» est une clé étrangère référençant «Patient» représentant le numéro d'un patient hospitalisé.
- ✓ «code\_service,num\_chambre» est une clé étrangère référençant la «chambre» où un patient est hospitalisé, par le code du service et le numéro de la chambre concernée.

#### La relation «Soigne»:

- ✓ «num\_patient» est une clé étrangère référençant «Patient» représentant le numéro d'un patient suivi par un medecin.
- ✓ «num\_med» est une clé étrangère référençant «Medecin» représentant le numéro d'un medecin suivant un patient.

### 5. Les autres contraintes d'intégrité:

Sur la relation «Service»: Une contrainte de type «unique»garantissant l'unicité du nom de chaque service.

Sur la relation «Medecin»: Une contrainte de type «check» spécifiant les valeurs que peut prendre l'attribut «spécialité».

Sur la relation «Infirmier»: Une contrainte de type «check» spécifiant les valeurs que peut prendre l'attribut «rotation».

Sur la relation «Hospitalisation»:Une contrainte de type «unique»garantissant l'unicité du numéro de chaque lit.

### Service:

```
SQL> create table service(code_service varchar2(3),nom_service varchar2(50),batiment varchar2(1),directeur number(3),constraint cpservice primary key(code_service),constraint uservice unique(nom_service),constraint ceservice foreign key(directeur) references medecin(num_med));
```

Table created.

### Chambre:

```
SQL> create table chambre(code_service varchar2(3),num_chambre number(3),surveillant number(3),nb_lits number(1),constraint cpchambre primary key(code_service,num_chambre),constraint ce1chambre foreign key(code_service) references service,constraint ce2chambre foreign key(surveillant) references infirmier(num_inf));
```

Table created.

### Employe:

```
SQL> create table employe(num_emp integer,nom_emp varchar2(20),prenom_emp varchar2(20),adresse_emp varchar2(60),tel_emp number(10),constraint cpemploye primary key(num_emp));
```

Table created.

### Medecin:

```
SQL> conn dbahospital/chikh
```

Connected.

```
SQL> create table medecin(num_med integer,specialite varchar2(20),constraint cpmedecin primary key(num_med), constraint chmedecin check (specialite in('orthopediste','cardiologue','traumatologue','anesthesiste','pneumologue','radiologue')));
```

Table created.

```
SQL> alter table medecin add constraint cemedecin foreign key(num_med) references employe(num_emp);
```

Table altered.

### Infirmier:

```
SQL> create table infirmier(num_inf number(3),code_service varchar2(3),rotation varchar2(4),salaire integer, constraint cpinfirmier primary key(num_inf),constraint ceinfirmier foreign key(code_service) references service, constraint chinfirmier check(rotation in('jour','nuit')));
```

Table created.

```
SQL> alter table infirmier add constraint ce2infirmier foreign key(num_inf) references employe(num_emp);
```

Table altered.

### Patient:

```
SQL> create table patient(num_patient number(3),nom_patient varchar2(20),prenom_patient varchar2(20),adresse_patient varchar2(60),tel_patient integer,mutuelle varchar2(7),constraint cppatient primary key(num_patient));
```

Table created.

### Hospitalisation:

```
SQL> create table hospitalisation(num_patient number(3),code_service varchar2(3),
num_chambre number(3),lit number(1),constraint cphospitalisation primary key(num_
patient),constraint ce1hospitalisation foreign key(num_patient) references patien
t,constraint ce2hospitalisation foreign key(code_service,num_chambre) references
chambre,constraint uhospitalisation unique(code_service,num_chambre,lit));

Table created.
```

### Soigne:

```
SQL> create table soigne(num_patient number(3),num_med number(3),constraint cpsoigne primary key(num_patient,num_med),constraint ce1soigne foreign key(num_patient) refe
rences patient,constraint ce2soigne foreign key(num_med) references medecin);

Table created.
```

#### 6. L'ajout de l'attribut Date Host de type date dans la relation «Hospitalisation»:

```
SQL> alter table hospitalisation add (date_host date);

Table altered.
```

#### 7. L'ajoute de la contrainte not null pour les attributs «salaire» et «mutuelle»:

```
SQL> alter table infirmier modify salaire not null;

Table altered.

SQL> alter table patient modify mutuelle not null;

Table altered.
```

#### 8. Agrandir et réduire la taille de l'attribut «prenom\_patient»:

```
SQL> desc patient;
Name                               Null?    Type
-----
NUM_PATIENT                        NOT NULL NUMBER(3)
NOM_PATIENT                        VARCHAR2(20)
PRENOM_PATIENT                     VARCHAR2(20)
ADRESSE_PATIENT                     VARCHAR2(60)
TEL_PATIENT                         NUMBER(38)
MUTUELLE                           NOT NULL VARCHAR2(7)

SQL> alter table patient modify prenom_patient varchar2(40);

Table altered.

SQL> desc patient;
Name                               Null?    Type
-----
NUM_PATIENT                        NOT NULL NUMBER(3)
NOM_PATIENT                        VARCHAR2(20)
PRENOM_PATIENT                     VARCHAR2(40)
ADRESSE_PATIENT                     VARCHAR2(60)
TEL_PATIENT                         NUMBER(38)
MUTUELLE                           NOT NULL VARCHAR2(7)

SQL> alter table patient modify prenom_patient varchar2(20);

Table altered.

SQL> desc patient;
Name                               Null?    Type
-----
NUM_PATIENT                        NOT NULL NUMBER(3)
NOM_PATIENT                        VARCHAR2(20)
PRENOM_PATIENT                     VARCHAR2(20)
ADRESSE_PATIENT                     VARCHAR2(60)
TEL_PATIENT                         NUMBER(38)
MUTUELLE                           NOT NULL VARCHAR2(7)
```

9. Suppression de la colonne «tel\_emp» dans «Employe», vérification de la suppression ensuite la recréation de la colonne «tel\_emp»:

```
SQL> desc employe;
Name                               Null?    Type
-----
NUM_EMP                            NOT NULL NUMBER(38)
NOM_EMP                            VARCHAR2(20)
PRENOM_EMP                         VARCHAR2(20)
ADRESSE_EMP                        VARCHAR2(60)
TEL_EMP                            NUMBER(10)

SQL> alter table employe drop column tel_emp;

Table altered.

SQL> desc employe;
Name                               Null?    Type
-----
NUM_EMP                            NOT NULL NUMBER(38)
NOM_EMP                            VARCHAR2(20)
PRENOM_EMP                         VARCHAR2(20)
ADRESSE_EMP                        VARCHAR2(60)

SQL> alter table employe add(tel_emp number(10));

Table altered.

SQL> desc employe;
Name                               Null?    Type
-----
NUM_EMP                            NOT NULL NUMBER(38)
NOM_EMP                            VARCHAR2(20)
PRENOM_EMP                         VARCHAR2(20)
ADRESSE_EMP                        VARCHAR2(60)
TEL_EMP                            NUMBER(10)
```

10. Renommer la colonne «adresse\_patient» dans la table «Patient» par «adr\_pat»:

```
SQL> desc patient
Name                               Null?    Type
-----
NUM_PATIENT                        NOT NULL NUMBER(3)
NOM_PATIENT                        VARCHAR2(20)
PRENOM_PATIENT                     VARCHAR2(20)
ADRESSE_PATIENT                     VARCHAR2(60)
TEL_PATIENT                        NUMBER(10)
MUTUELLE                           NOT NULL VARCHAR2(7)

SQL> alter table patient rename column adresse_patient to adr_pat;

Table altered.

SQL> desc patient
Name                               Null?    Type
-----
NUM_PATIENT                        NOT NULL NUMBER(3)
NOM_PATIENT                        VARCHAR2(20)
PRENOM_PATIENT                     VARCHAR2(20)
ADR_PAT                            VARCHAR2(60)
TEL_PATIENT                        NUMBER(10)
MUTUELLE                           NOT NULL VARCHAR2(7)
```

11. La contrainte spécifiant que le salaire d'un infirmier doit être entre 10000 DA et 30000 DA:

```
SQL> desc infirmier
Name                                         Null?    Type
-----
NUM_INF                                     NOT NULL NUMBER(3)
CODE_SERVICE                               VCHAR2(3)
ROTATION                                   VCHAR2(4)
SALAIRE                                     NOT NULL NUMBER(38)

SQL> alter table infirmier add constraint chsalaireinfirmier check (salaire between 10000
and 30000);

Table altered.
```

12. La contrainte spécifiant que chaque medecin doit avoir une spécialité:

```
SQL> desc medecin
Name                                         Null?    Type
-----
NUM_MED                                     NOT NULL NUMBER(3)
SPECIALITE                                VCHAR2(20)

SQL> alter table medecin modify specialite not null;

Table altered.

SQL> desc medecin
Name                                         Null?    Type
-----
NUM_MED                                     NOT NULL NUMBER(3)
SPECIALITE                                NOT NULL VCHAR2(20)
```

### PARTIE 3:

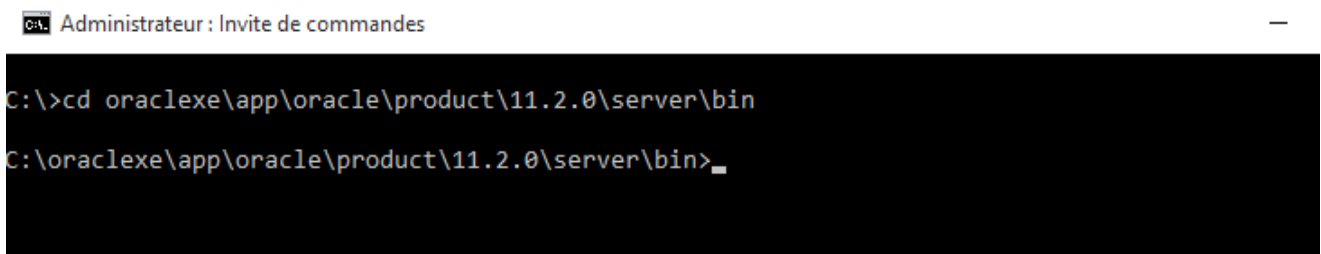
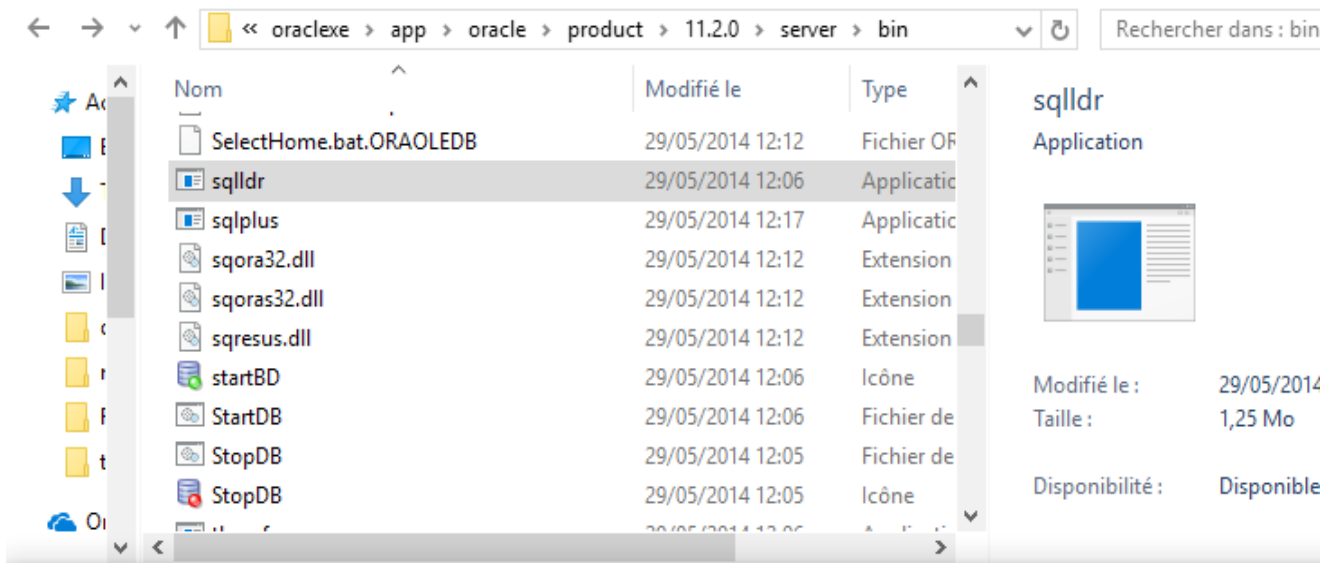
13. Le remplissage de toutes les tables en utilisant le SQL LOADER:

Nous créons d'abord pour chaque table, un fichier (.csv) depuis un classeur (excel par exemple) contenant les tuples donnés (exemple dans la figure suivante de la table «Employe»), ensuite un fichier(.ctl) de contrôle contenant une instruction dans laquelle est spécifié le chemin du fichier qui contient les tuples et la table qui va contenir ces tuples (nous changeons à chaque fois le nom et les colonnes de la table concernée).

Remarque: On peut choisir le séparateur des colonnes('; ' par exemple) lors de l'enregistrement du fichier csv.

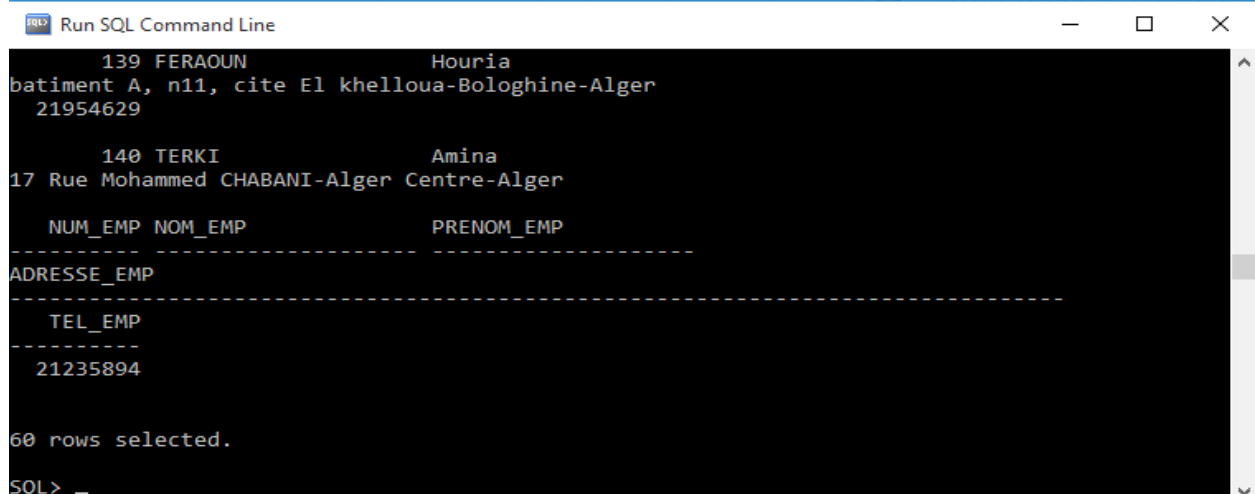
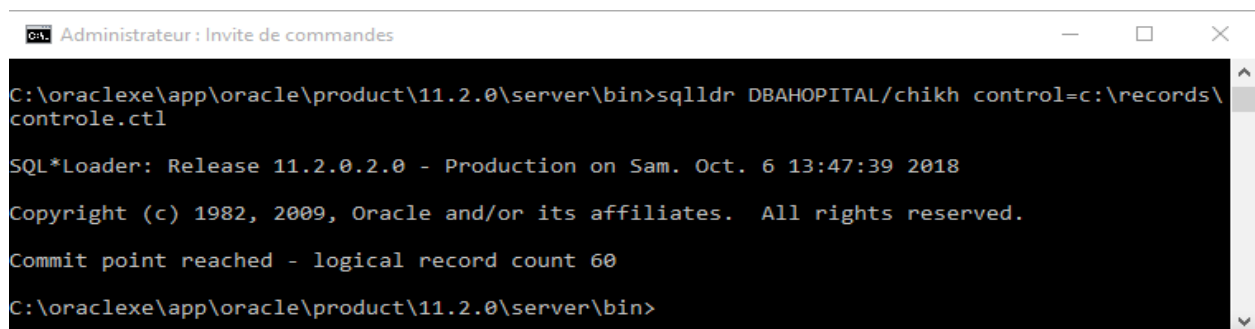
The image shows two windows. The top window is OpenOffice Calc with a spreadsheet titled 'employe.csv'. It has columns A, B, C, D, and E. The data includes employee names, addresses, and phone numbers. The bottom window is Sublime Text (UNREGISTERED) showing a control file named 'controle.ctl'. The file contains four lines of SQL Loader instructions: 1. 'load data infile 'c:\records\employe.csv'', 2. 'into table employe', 3. 'fields terminated by ';' optionally enclosed by ''', and 4. '(num\_emp,nom\_emp,prenom\_emp,adresse\_emp,tel\_emp)'. The file is saved in 'C:\Records\controle.ctl'.

Puis nous utilisons l'invite de commande (CMD) pour charger les tuples depuis le fichier classeur vers la base de données. Nous devons se déplacer vers le dossier contenant l'application «SQLldr».



Nous executons en fin la commande qui a la syntaxe suivante: SQLldr userid/password control=chemin de fichier de contrôle

Le userid/password représente le nom et le mot de passe de l'utilisateur de la base de donnée à qui appartient la table que nous voulons remplir; pour notre cas c'est «dbahopital»; pour «control» c'est le fichier «contrôle.ctl» qui a été précédemment créé.





14. Nous modifions la valeur de «directeur» en recherchant la valeur dans la table «Employe» par le nom et le prenom donnés.

```
SQL> update service set directeur=(select num_emp from employe where nom_emp='BOU
LARAS' and prenom_emp='Fatima') where nom_service='Cardiologie';

1 row updated.

SQL> select * from service;
```

| COD | NOM_SERVICE                  | B | DIRECTEUR |
|-----|------------------------------|---|-----------|
| CAR | Cardiologie                  | B | 152       |
| CHG | Chirurgie generale           | A | 34        |
| REA | Reanimation et Traumatologie | A | 19        |

15. Diminuer 5000da à tous les infirmiers dont la rotation égale 'jour'.

```
SQL> update infirmier set salaire=salaire-5000 where rotation='jour';
update infirmier set salaire=salaire-5000 where rotation='jour'
*
ERROR at line 1:
ORA-02290: check constraint (DBAHOPITAL.CHSALAIREINFIRMIER) violated

SQL> alter table infirmier disable constraint chsalaireinfirmier;

Table altered.

SQL> update infirmier set salaire=salaire-5000 where rotation='jour';

14 rows updated.

SQL> alter table infirmier enable constraint chsalaireinfirmier;
alter table infirmier enable constraint chsalaireinfirmier
*
ERROR at line 1:
ORA-02293: cannot validate (DBAHOPITAL.CHSALAIREINFIRMIER) - check constraint
violated
```

Au début, la contrainte que nous avons définie précédemment n'a pas permis cette opération, donc pour s'y faire nous avons dû la désactiver. Ensuite nous avons écrit une requête pour la réactiver, ce qui a mené à une incohérence entre certains tuples (modifiés) et les conditions de la contrainte.

16. Suppression de tous les médecins de spécialité «Cardiologie»:

Suppression non permise suite à la présence des enregistrements référençant la table «Medecin».

```
SQL> Delete from medecin where specialite='cardiologue';
Delete from medecin where specialite='cardiologue'
*
ERROR at line 1:
ORA-02292: integrity constraint (DBAHOPITAL.CE2SOIGNE) violated - child record
found
```

17. Si la majorité des accès à la table «Medecin» se font par l'attribut «specialite», nous pouvons opter pour un index secondaire sur cet attribut.

```
SQL> create index index_medecin on medecin(specialite);

Index created.
```



#### PARTIE 4:

18. Donner la liste des patients (Prénom et nom) affiliés à la mutuelle « MAAF »

```
SQL> select nom_patient, prenom_patient from patient where mutuelle like '%MAAF%';
```

| NOM_PATIENT  | PRENOM_PATIENT |
|--------------|----------------|
| DIAF AMROUNI | Ghania         |
| LAAOUAR      | Ali            |
| MEDJAHED     | Ahmed          |
| HALFAOUI     | Redouane       |
| MATI         | Djamel         |
| HABABB       | khadra         |
| MAHBOUBA     | Cherifa        |
| BOUDJELAL    | Salim          |

8 rows selected.

19. Donner pour chaque lit occupé du bâtiment « B » de l'hôpital occupé par un patient affilié à une mutuelle dont le nom commence par « MN... », le numéro du lit, le numéro de la chambre, le nom du service ainsi que le prénom, le nom et la mutuelle du patient l'occupant.

```
SQL> select h.lit, h.num_chambre, s.nom_service, p.nom_patient, p.prenom_patient, p.mutuelle from hospitalisation h, service s, patient p where h.num_patient=p.num_patient and h.code_service=s.code_service and p.mutuelle like 'MN%';
```

|       | LIT | NUM_CHAMBRE | NOM_SERVICE                  |
|-------|-----|-------------|------------------------------|
| SERIR | 1   | 101         | Cardiologie                  |
| TAHMI | 3   | 101         | Cardiologie                  |
| DRIZI | 1   | 105         | Reanimation et Traumatologie |

| NOM_PATIENT | PRENOM_PATIENT | MUTUELL |
|-------------|----------------|---------|
| SERIR       | Mustapha       | MNAM    |
| TAHMI       | Lamia          | MNH     |
| DRIZI       | Djamel         | MNAM    |

|        | LIT | NUM_CHAMBRE | NOM_SERVICE                  |
|--------|-----|-------------|------------------------------|
| GROUDA | 1   | 107         | Reanimation et Traumatologie |
| KOUBA  | 1   | 401         | Chirurgie generale           |
| SAADI  | 1   | 402         | Chirurgie generale           |

| NOM_PATIENT | PRENOM_PATIENT | MUTUELL |
|-------------|----------------|---------|
| GROUDA      | Houda          | MNH     |
| KOUBA       | Mohamed        | MNFTC   |
| SAADI       | Med Tayeb      | MNH     |

20. Pour chaque patient soigné par plus de 3 médecins donner le nombre total de ses médecins ainsi que le nombre correspondant de spécialités médicales concernées:

D'abord, nous créons une vue qui nous donne, pour chaque patient, le nombre de ses medecins:

```
SQL> create view patient_nbrmed(num_patient,nbr_medecins) as
  2 select num_patient,count(num_med) from soigne group by num_patient;

View created.

SQL> select * from patient_nbrmed;

NUM_PATIENT  NBR_MEDECINS
-----
           1             2
           3             3
           6             4
          13             3
          14             1
          21             3
          23             3
          33             2
          35             2
          36             3
          37             1

NUM_PATIENT  NBR_MEDECINS
-----
          41             3
          43             2
          44             2
          46             2
```

Ensuite une vue qui nous donne pour chaque patient le nombre de spécialités concernées:

```
SQL> create view patient_spec(num_patient,specialite) as
  2 select s.num_patient,m.specialite from soigne s,medecin m where s.num_med=
m.num_med
  3 group by num_patient,specialite;

View created.

SQL> select * from patient_spec;

NUM_PATIENT  SPECIALITE
-----
          14 cardiologue
          33 anesthesiste
          36 radiologue
          43 orthopediste
          78 orthopediste
         104 cardiologue
         105 anesthesiste
         117 orthopediste
         128 orthopediste
         145 orthopediste
```

En fin, nous executons une requête qui joint les deux vues pour obtenir le résultat voulu:

```
SQL> select v1.num_patient,v1.nbr_medecins,count(specialite) from patient_nbrme
d v1,patient_spec v2 where v1.num_patient=v2.num_patient and v1.nbr_medecins>3
group by v1.num_patient,v1.nbr_medecins order by v1.num_patient;
```

| NUM_PATIENT | NBR_MEDECINS | COUNT(SPECIALITE) |
|-------------|--------------|-------------------|
| 6           | 4            | 3                 |
| 76          | 4            | 3                 |
| 108         | 4            | 4                 |
| 117         | 4            | 3                 |
| 145         | 4            | 4                 |
| 147         | 4            | 4                 |
| 159         | 4            | 4                 |
| 172         | 4            | 3                 |

8 rows selected.

21. La moyenne des salaires des infirmiers(ères) par service:

```
SQL> select code_service,avg(salaire) from infirmier group by code_service;
```

| COD | AVG(SALAIRE) |
|-----|--------------|
| REA | 10930,8571   |
| CHG | 12463,5385   |
| CAR | 12568,5      |

22. Pour chaque service, le rapport entre le nombre d'infirmier(ères) affecté(es) au service et le nombre de patients hospitalisés dans le service:

```
SQL> select i.code_service,count(code_service) as nbr_infirmiers,(select count(nu
m_patient) from hospitalisation h where i.code_service=h.code_service) as nbr_pat
ients from infirmier i group by i.code_service;
```

| COD | NBR_INFIRMIERS | NBR_PATIENTS |
|-----|----------------|--------------|
| REA | 7              | 10           |
| CHG | 13             | 20           |
| CAR | 8              | 9            |

23. la liste des médecins (Prénom et nom) ayant un patient hospitalisé dans chaque service

```
SQL> select s.num_med,e.nom_emp,e.prenom_emp,count(distinct h.code_service) from
soigne s,employe e,hospitalisation h where e.num_emp=s.num_med and s.num_patient=
h.num_patient group by s.num_med,e.nom_emp,e.prenom_emp having count(distinct h.c
ode_service)=(select count (*) from service);
```

| NUM_MED | NOM_EMP   | PRENOM_EMP | COUNT(DISTINCTH.CODE_SERVICE) |
|---------|-----------|------------|-------------------------------|
| 179     | MOHAMMEDI | Mustapha   | 3                             |
| 126     | BELGHALI  | Mohammed   | 3                             |
| 144     | BENDALI   | Hacine     | 3                             |
| 85      | BAALI     | Souad      | 3                             |
| 196     | TEBIBEL   | Nabila     | 3                             |
| 31      | ABDELAZIZ | Ahmed      | 3                             |