



Base de Données avancées

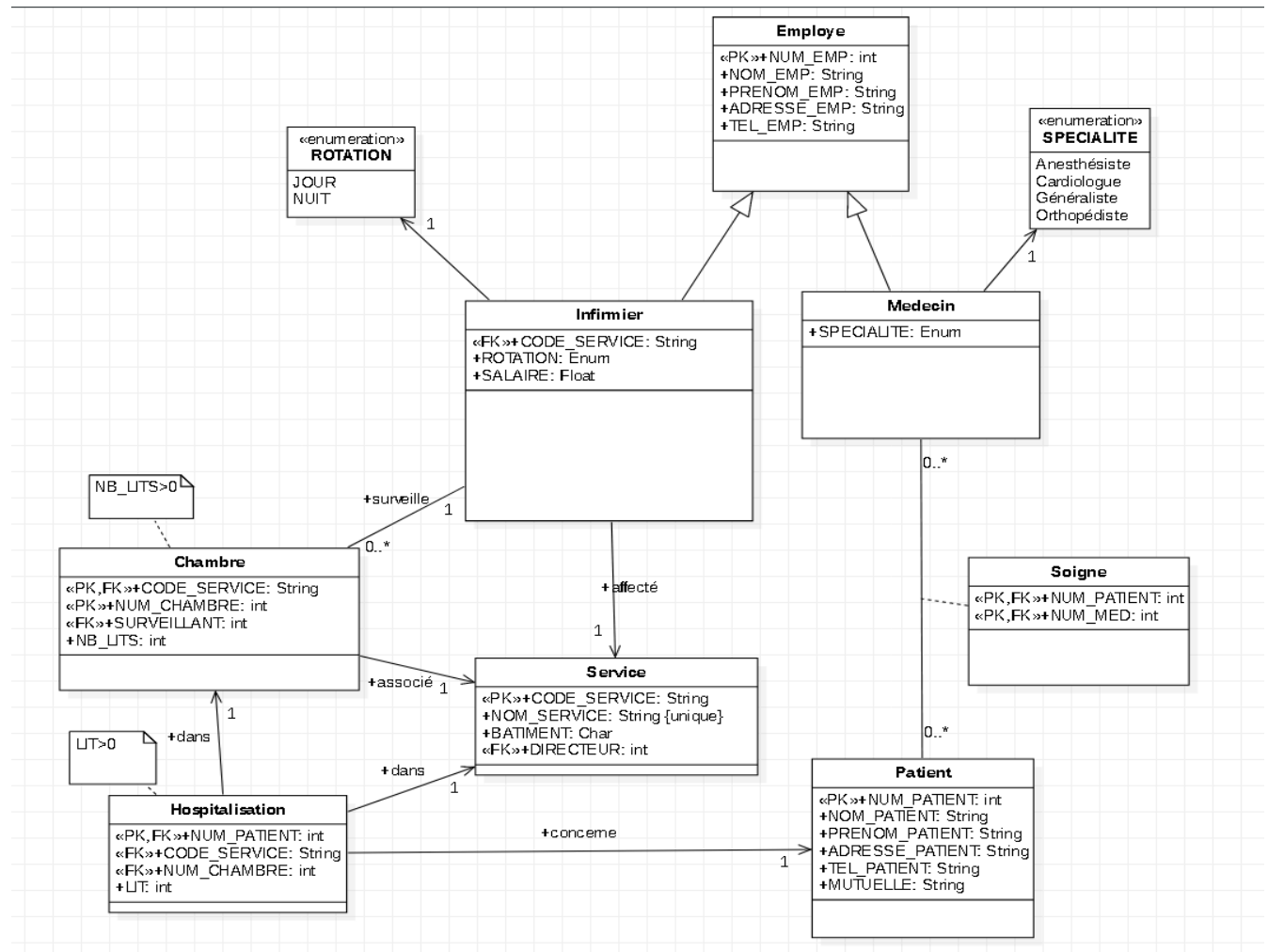
RAPPORT DU Projet 01 SQL3-Oracle

Binôme:

MATRICULE	NOM	PRENOM
181831072145	ZAÏT	FOUAD
181832028311	HANANI	FOUAD

Partie I : Modélisation orientée objet

- 1- Transformez ce schéma relationnel en schéma Objet (diagramme de classes):



Partie II : Création des TableSpaces et utilisateur :

2- Créer deux TableSpaces SQL3_TBS et SQL3_TempTBS :

```
CREATE TABLESPACE SQL3_TBS DATAFILE'C:\SQL3_TBS.dat' size 100M  
AUTOEXTEND ON ONLINE;
```

```
CREATE TEMPORARY TABLESPACE SQL_TempTBS    TEMPFILE  
'C:\SQL_TempTBS.dat' SIZE 100M AUTOEXTEND ON;
```

```
SQL> CREATE TABLESPACE SQL3_TBS DATAFILE'C:\SQL3_TBS.dat' size 100M AUTOEXTEND ON ONLINE;  
Tablespace created.  
  
SQL> CREATE TEMPORARY TABLESPACE SQL_TempTBS TEMPFILE 'C:\SQL_TempTBS.dat' SIZE 100M AUTOEXTEND ON;  
Tablespace created.
```

3- Créer un utilisateur SQL3 en lui attribuant les deux tablespaces créés précédemment :

Create User SQL3 Identified by psw Default Tablespace SQL3_TBS Temporary Tablespace SQL_TempTBS;

```
SQL> Create User SQL3 Identified by psw Default Tablespace SQL3_TBS Temporary Tablespace SQL_TempTBS;  
User created.
```

4- Donner tous les privilèges à cet utilisateur:

grant all privileges to SQL3;

```
SQL> grant all privileges to SQL3;  
  
Grant succeeded.
```

Partie III : Langage de définition de données

5. En se basant sur le diagramme de classes fait, définir tous les types nécessaires. Prendre en compte toutes associations qui existent.

- Type employe :

Requette:

```
CREATE TYPE employe as object(NUM_EMP Number(7),NOM_EMP  
varchar2(30),PRENOM_EMP varchar2(30),ADRESSE_EMP varchar2(100),TEL_EMP  
varchar2(10)) Not Final;
```

/

```
SQL> desc employe  
employe is NOT FINAL  
Name                               Null?      Type  
-----  
NUM_EMP                            NUMBER(7)  
NOM_EMP                            VARCHAR2(30)  
PRENOM_EMP                         VARCHAR2(30)  
ADRESSE_EMP                        VARCHAR2(100)  
TEL_EMP                            VARCHAR2(10)
```

-Type tmedecin :

Requette:

```
CREATE TYPE tmedecin under employe(SPECIALITE varchar2(40));
```

/

```
SQL> desc tmedecin  
tmedecin extends SQL3.EMPLOYEE  
Name                               Null?      Type  
-----  
NUM_EMP                            NUMBER(7)  
NOM_EMP                            VARCHAR2(30)  
PRENOM_EMP                         VARCHAR2(30)  
ADRESSE_EMP                        VARCHAR2(100)  
TEL_EMP                            VARCHAR2(10)  
SPECIALITE                        VARCHAR2(40)
```

-Type tservice :

Requette:

```
CREATE TYPE tservice as object(CODE_SERVICE char(3),NOM_SERVICE  
varchar2(40),BATIMENT char,directeur ref employe);
```

/

```
SQL> desc tservice
Name                                     Null?   Type
-----
CODE_SERVICE                           CHAR(3)
NOM_SERVICE                            VARCHAR2(40)
BATIMENT                               CHAR(1)
DIRECTEUR                              REF OF EMPLOYE
```

-Type tinfirmer :

Requette :

```
CREATE TYPE tinfirmer under employe (NUM_INF ref employe,infirmer_service ref  
tservice,ROTATION char(4),SALAIRE number(10,2)) ;
```

/

```
SQL> desc tinfirmer
tinfirmer extends SQL3.EMPLOYE
Name                                     Null?   Type
-----
NUM_EMP                                NUMBER(7)
NOM_EMP                                VARCHAR2(30)
PRENOM_EMP                             VARCHAR2(30)
ADRESSE_EMP                             VARCHAR2(100)
TEL_EMP                                 VARCHAR2(10)
INFIRMIER_SERVICE                       REF OF TSERVICE
ROTATION                                CHAR(4)
SALAIRE                                 NUMBER(10,2)
```

-Type tpatient:

Requette:

```
CREATE TYPE tpatient as object(NUM_PATIENT Number(7),NOM_PATIENT  
varchar2(30),PRENOM_PATIENT varchar2(30),  
ADRESSE_PATIENT varchar2(100),TEL_PATIENT varchar2(10),MUTUELLE varchar2(10));
```

/

```
SQL> desc tpatient
Name                                     Null?      Type
-----
NUM_PATIENT                             NUMBER(7)
NOM_PATIENT                             VARCHAR2(30)
PRENOM_PATIENT                           VARCHAR2(30)
ADRESSE_PATIENT                           VARCHAR2(100)
TEL_PATIENT                              VARCHAR2(10)
MUTUELLE                                 VARCHAR2(10)
PATIENT_SOIGNE                           T_SET_REF_SOIGNE
```

-Type tchambre :

Requette:

```
CREATE TYPE tchambre as object (chambre_service ref tservice,NUM_CHAMBRE Number(4),
chambre_infirmier ref temploie,NB_LITS int);
```

/

```
SQL> desc tchambre
Name                                     Null?      Type
-----
CHAMBRE_SERVICE                         REF OF TSERVICE
NUM_CHAMBRE                             NUMBER(4)
CHAMBRE_INFIRMIER                       REF OF TEMPLOIE
NB_LITS                                 NUMBER(38)
```

-Type thospitalisation:

Requette:

```
CREATE TYPE thospitalisation as object(hospitalisation_patient ref
tpatient,hospitalisation_service ref tservice,hospitalisation_chambre ref tchambre,LIT int);
```

/

```
SQL> desc thospitalisation
Name                                     Null?      Type
-----
HOSPITALISATION_PATIENT                 REF OF TPATIENT
HOSPITALISATION_SERVICE                 REF OF TSERVICE
HOSPITALISATION_CHAMBRE                 REF OF TCHAMBRE
LIT                                     NUMBER(38)
```

-Type tsoigne :

Requette:

create or replace type tsoigne as object(soigne_patient ref tpatient,soigne_medecin ref
temploye);

/

```
SQL> desc tsoigne
```

Name	Null?	Type
SOIGNE_PATIENT		REF OF TPATIENT
SOIGNE_MEDECIN		REF OF EMPLOYE

```
SQL> CREATE TYPE temploye as object(NUM_EMP Number(7),NOM_EMP varchar2(30),PRENOM_EMP varchar2(30),ADRESSE_EMP varchar2(100),TEL_EMP varchar2(10)) Not Final;  
2 /
```

Type created.

```
SQL> CREATE TYPE tmedecin under temploye(SPECIALITE varchar2(40));  
2 /
```

Type created.

```
SQL> CREATE TYPE tservice as object(CODE_SERVICE char(3),NOM_SERVICE varchar2(40),BATIMENT char,directeur ref temploye);  
2 /
```

Type created.

```
SQL> CREATE TYPE tinfirmier under temploye (infirmier_service ref tservice,ROTATION char(4),SALAIRE number(10,2)) ;  
2 /
```

Type created.

```
SQL> CREATE TYPE tpatient as object(NUM_PATIENT Number(7),NOM_PATIENT varchar2(30),PRENOM_PATIENT varchar2(30),  
2 ADRESSE_PATIENT varchar2(100),TEL_PATIENT varchar2(10),MUTUELLE varchar2(10));  
3 /
```

Type created.

```
SQL> CREATE TYPE tchambre as object (chambre_service ref tservice,NUM_CHAMBRE Number(4), chambre_infirmier ref temploye,NB_LITS int);  
2 /
```

Type created.

```
SQL> CREATE TYPE thospitalisation as object(hospitalisation_patient ref tpatient,hospitalisation_service ref tservice,hospitalisation_chambre ref tchambre,LIT int);  
2 /
```

Type created.

```
SQL> create or replace type tsoigne as object( soigne_patient ref tpatient,soigne_medecin ref temploye);  
2 /
```

Type created.

-Prendre en compte les associations qui existent :

-association infirmier chambre :

L'infirmier surveille 0 ou plusieurs chambres donc on ajoute une table de chambres à tinfirmier , une chambre peut être surveiller par un seul infirmier on a ajouter dans la déclaration du type tchambre un attribut (ref) vers infirmier (chambre_infirmier ref tinfirmier) :

```
CREATE TYPE t_set_ref_chambre as table of ref tchambre;
```

```
/
```

```
ALTER TYPE tinfirmier add attribute infirmier_chambre t_set_ref_chambre cascade;
```

```
SQL> CREATE TYPE t_set_ref_chambre as table of ref tchambre;
2 /
Type created.
SQL> ALTER TYPE tinfirmier add attribute infirmier_chambre t_set_ref_chambre cascade;
Type altered.
```

-association chambre service :

Une chambre est associé à un service ,on a rajouter dans la déclaration du type tchambre un attribut (ref) vers tservice (chambre_service ref tservice).

-association hospitalisation chambre :

Une hospitalisation est dans une chambre ,on a rajouter dans la déclaration du type thospitalisation un attribut (ref) vers tchambre (hospitalisation_chambre ref tchambre)

-association hospitalisation patient :

Une hospitalisation concerne un seul patient , on a ajouter dans la déclaration du type thospitalisation un attribut (ref) vers tpatient (hospitalisation_patient ref tpatient)

-association hospitalisation service :

Une hospitalisation se déroule dans un seul service , on a ajouter dans la déclaration du type thospitalisation un attribut (ref) vers tservice(hospitalisation_service ref tservice)

-association infirmier service :

Un infirmier est affecté à un seul service , on a ajouter dans la déclaration du type tinfirmier un attribut (ref) vers tservice (infirmier_service ref tservice).

-association service médecin :

Un service possède un directeur qui est un medecin , on a ajouter dans la déclaration du type tservice un attribut (ref) vers medecin qui est un employe (directeur ref temploie)

-association médecin soigne patient :

-soigne medecin : on ajoute à soigne l'attribut soigne_medecin à soigne (soigne_medecin ref tmedecin) reference un medecin qui est employe

-soigne patient : on ajoute à soigne l'attribut soigne_patient à soigne (soigne_patient ref tpatient)

-medecin soigne : on ajoute à tmedecin une table de ref de tsoigne (medecin_soigne t_set_ref_soigne)

-patient soigne : on ajoute à tpaiient une table de ref de tsoigne (patient_soigne t_set_ref_soigne)

Requette :

```
CREATE TYPE t_set_ref_soigne as table of ref tsoigne;
```

```
/
```

```
ALTER TYPE tmedecin add attribute medecin_soigne t_set_ref_soigne cascade;
```

```
ALTER TYPE tpatient add attribute patient_soigne t_set_ref_soigne cascade;
```

```
SQL> CREATE TYPE t_set_ref_soigne as table of ref tsoigne;
2 /

Type created.

SQL> ALTER TYPE tmedecin add attribute medecin_soigne t_set_ref_soigne cascade;

Type altered.

SQL> ALTER TYPE tpatient add attribute patient_soigne t_set_ref_soigne cascade;

Type altered.
```

Généralisation medecin infirmier-> employe :

Super classe : employe

Sous classe medecin et infirmier ce sont des employe avec des attributs en plus .

On a definit employe comme not final , medecin sera une sous classe de employe avec l'attribut specialite en plus et infirmier sera une sous classe de employe avec les attributs rotation et salaire en plus. et infirmier_service qui est une reference vers le service auquel il est assosiié.

6- Définir les tables nécessaires à la base de données:

1-Table employe :

CREATE TABLE EMPLOYE of temploye(primary key(NUM_EMP));

2-Table patient :

CREATE TABLE PATIENT of tpatient(primary key(NUM_PATIENT)) nested table patient_soigne store as table_soigne_patient;

4-Table service :

CREATE TABLE SERVICE of tservice(primary key(CODE_SERVICE),foreign key(directeur) references employe,UNIQUE(NOM_SERVICE));

6-Table soigne:

CREATE TABLE SOIGNE of tsoigne(foreign key(soigne_patient) references PATIENT,foreign key(soigne_medecin) references employe);

7-Table chambre :

CREATE TABLE CHAMBRE of tchambre(foreign key(chambre_service)references SERVICE,foreign key(chambre_infirmier)references employe,check(NB_LITS>0));

8-Table hospitalisation :

CREATE TABLE HOSPITALISATION of thospitalisation (foreign key(hospitalisation_patient)references PATIENT,foreign key(hospitalisation_service)references SERVICE,foreign key(hospitalisation_chambre)references CHAMBRE);

```
SQL> CREATE TABLE EMPLOYE of temploye(primary key(NUM_EMP));
Table created.

SQL> CREATE TABLE PATIENT of tpatient(primary key(NUM_PATIENT)) nested table patient_soigne store as table_soigne_patient;
Table created.

SQL> CREATE TABLE SERVICE of tservice(primary key(CODE_SERVICE),foreign key(directeur) references employe,UNIQUE(NOM_SERVICE));
Table created.

SQL> CREATE TABLE SOIGNE of tsoigne(foreign key(soigne_patient) references PATIENT,foreign key(soigne_medecin) references employe);
Table created.

SQL> CREATE TABLE CHAMBRE of tchambre(foreign key(chambre_service)references SERVICE,foreign key(chambre_infirmier)references employe,check(NB_LITS>0));
Table created.

SQL> CREATE TABLE HOSPITALISATION of thospitalisation (foreign key(hospitalisation_patient)references PATIENT,foreign key(hospitalisation_service)references SERVICE,foreign key(hospitalisation_chambre)references CHAMBRE);
Table created.
```

7-Définitions des méthodes :

1- calculer pour chaque spécialité donnée, le nombre de médecins affectés.

On ajoute d'abord la fonction au type employe:

```
alter type tmedecin add member function nbmed (spec varchar2) return numeric cascade;
```

Fonction :

```
create or replace type body tmedecin as member function nbmed(spec varchar2) return  
numeric is
```

```
nb number ;
```

```
begin
```

```
nb:=0;
```

```
SELECT count(treat(value(e) as tmedecin)) into nb from employe e where value(e) is of  
(tmedecin ) AND treat(value(e) as tmedecin).SPECIALITE=spec;
```

```
return nb;
```

```
end nbmed;
```

```
end;
```

```
/
```

```
SQL> alter type tmedecin add member function nbmed (spec varchar2) return numeric cascade;
Type altered.

SQL> create or replace type body tmedecin as member function nbmed(spec varchar2) return numeric is
 2 nb number ;
 3 begin
 4 nb:=0;
 5 SELECT count(treat(value(e) as tmedecin)) into nb from employe e where value(e) is of (tmedecin ) AND treat(value(e) as tmedecin).SPECIALITE=spec;
 6 return nb;
 7 end nbmed;
 8 end;
 9 /
Type body created.
```

```
SQL> desc tmedecin
tmedecin extends SQL3.EMPLOYEE
Name                                     Null?   Type
-----
NUM_EMP                                NUMBER(7)
NOM_EMP                                VARCHAR2(30)
PRENOM_EMP                             VARCHAR2(30)
ADRESSE_EMP                             VARCHAR2(100)
TEL_EMP                                 VARCHAR2(10)
SPECIALITE                             VARCHAR2(40)
MEDECIN_SOIGNE                          T_SET_REF_SOIGNE

METHOD
-----
MEMBER FUNCTION NBMED RETURNS NUMBER
Argument Name      Type                               In/Out Default?
-----
SPEC                VARCHAR2                               IN
```

2- calculer pour chaque service donné, le nombre d'infirmier(ères) affecté(es) et le nombre de patients hospitalisés.

Cette fonction nous permet selon le char donné en entré avec le code service si le caractère donnée est 'i' cette methode nous calcule le nombre d'infirmiers sinon si le caractère donnée est h alors cette methode nous calcule le nombre d'hospitalisés pour le service caractérisé par son code service donnée.

On ajoute d'abord la fonction au type tservice:

```
alter type tservice add member function nbinfirmierhospitalise (code varchar2,b char ) return
numeric cascade;
```

Fonction :

```
create or replace type body tservice
```

```
as member function nbinfirmierhospitalise(code varchar2,b char ) return numeric is
```

```
nb number ;
```

```
begin
```

```
nb:=0;
```

```
if b='i' then
```

```
SELECT count(*) into nb from employe e where value(e) is of (tinfirmer) AND deref(treat(value(e)
as tinfirmer).infirmer_service).CODE_SERVICE=code;
```

```
return nb;
```

```
elsif b='h' then
```

```
SELECT COUNT (*) into nb from hospitalisation h where h.hospitalisation_service.CODE_SERVICE=
code;
```

```
return nb;
```

```
end if;
```

```
end nbinfirmierhospitalise;
```

end;

/

```
SQL> alter type tservice add member function nbinfirmerhospitalise (code varchar2,b char ) return numeric cascade;
Type altered.

SQL> create or replace type body tservice
 2 as member function nbinfirmerhospitalise(code varchar2,b char ) return numeric is
 3 nb number ;
 4 begin
 5 nb:=0;
 6 if b='i' then
 7 SELECT count(*) into nb from employe e where value(e) is of (tinfirmer) AND deref(treat(value(e) as tinfirmer).infirmer_service).CODE_SERVICE=code;
 8 return nb;
 9 elsif b='h' then
10 SELECT COUNT (*) into nb from hospitalisation h where h.hospitalisation_service.CODE_SERVICE= code;
11 return nb;
12 end if;
13 end nbinfirmerhospitalise;
14 end;
15 /
Type body created.
```

3- calculer pour chaque patient le nombre total de ses médecins soignants:

On ajoute d'abord à tpatient la fonction :

```
alter type tpatient add member function nbmedsoignants return numeric cascade;
```

Fonction :

```
create or replace type body tpatient
```

```
as member function nbmedsoignants return numeric is
```

```
nb number;
```

```
begin
```

```
nb:=0;
```

```
    select count(s.soigne_medecin.NUM_EMP) INTO nb from soigne s where
    self.NUM_PATIENT=s.soigne_patient.NUM_PATIENT ;
```

```
return nb;
```

```
end nbmedsoignants;
```

```
end;
```

```
/
```

```

SQL> alter type tpatient add member function nbmedsoignants return numeric cascade;

Type altered.

SQL>
SQL> create or replace type body tpatient
  2  as member function nbmedsoignants return numeric is
  3  nb number;
  4  begin
  5  nb:=0;
  6  select count(s.soigne_medecin.NUM_EMP) INTO nb from soigne s where self.NUM_PATIENT=s.soigne_patient.NUM_PATIENT ;
  7  return nb;
  8  end nbmedsoignants;
  9  end;
 10  /

Type body created.

```

4- afficher « vérification positive » si le salaire de l’infirmier est entre 10000 DA et 30000 DA et affiche « Vérification négative » sinon.

On ajoute d’abord à tinfirmer la procédure:

```
alter type temploie add member procedure verifsalaire cascade;
```

La procédure :

```
create or replace type body temploie
```

```
as member procedure verifsalaire is
```

```
begin
```

```
if(self is of (tinfirmer) AND treat(self as tinfirmer).SALAIRE between 10000 AND
30000)then dbms_output.put_line('vérification positive');
```

```
else dbms_output.put_line('vérification negative');
```

```
end if;
```

```
end verifsalaire;
```

```
end;
```

```
/
```

```

SQL> alter type temploie add member procedure verifsalaire cascade;

Type altered.

SQL>
SQL> create or replace type body temploie
  2  as member procedure verifsalaire is
  3  begin
  4  if(self is of (tinfirmer) AND treat(self as tinfirmer).SALAIRE between 10000 AND 30000)then dbms_output.put_line('vérification positive');
  5  else dbms_output.put_line('vérification negative');
  6  end if;
  7  end verifsalaire;
  8  end;
  9  /

Type body created.

```

Partie VI : Langage de manipulation de données

8. Remplir toutes les tables par les instances représentées ci-dessus :

Dans le script on a rempli toutes les tables par les instances données , dans ce rapport nous allons donner un exemple de l'insertion dans chaque table:

Exemple d'insertion dans chaque table :

Table employe :

-**medecin**: INSERT INTO EMPLOYE VALUES (tmedecin(4,'BOUROUBI','Taous','Lotissement Dauphin n°30 DRARIA/ALGER','021356085','Orthopédiste',t_set_ref_soigne()));

-**infirmier**: INSERT INTO EMPLOYE VALUES(tinfirmier(12,'HADJ','Zouhir','Cité de la Mosquée Bt 14-Boufarik-Blida','025474882',(SELECT REF(s) FROM service s WHERE s.CODE_SERVICE='REA'),'JOUR',12560.78,t_set_ref_chambre()));

Table Patient : INSERT INTO PATIENT VALUES (1,'GRIGAHCINE','Nacer','95,Bd Bougara-El biar-Alger','021920313','MNAM',t_set_ref_soigne());

Table service : INSERT INTO SERVICE VALUES ('CAR','Cardiologie','B',(SELECT REF(e) FROM employe e WHERE e.NUM_EMP=80));

Table Chambre: INSERT INTO CHAMBRE VALUES ((SELECT REF(s) FROM SERVICE s WHERE s.CODE_SERVICE='CAR'),101,(SELECT REF(e) FROM employe e WHERE e.NUM_EMP=95),3);

Table Soigne : INSERT INTO SOIGNE VALUES ((SELECT ref(p) from PATIENT p where p.NUM_PATIENT=13),(SELECT ref(e) from EMPLOYE e where e.NUM_EMP=4));

Table hospitalisation : INSERT INTO HOSPITALISATION VALUES ((SELECT REF(p) FROM PATIENT p WHERE p.NUM_PATIENT=1),(SELECT REF(s) FROM SERVICE s WHERE s.CODE_SERVICE='REA'),(SELECT REF(c) FROM CHAMBRE c WHERE c.NUM_CHAMBRE=101 AND Deref(value(c).chambre_service).CODE_SERVICE='REA'),1);

Partie V : Langage d'interrogation de données

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

```
SELECT p.NOM_PATIENT,p.PRENOM_PATIENT from patient p WHERE p.mutuelle='MAAF';
```

```
SQL> SELECT  p.NOM_PATIENT,p.PRENOM_PATIENT  from patient p WHERE p.mutuelle='MAAF';
```

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

8 rows selected.

10. 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.

```
SELECT h.lit,DEREF(value(h).hospitalisation_chambre).NUM_CHAMBRE,  
DEREF(value(h).hospitalisation_service).NOM_SERVICE,  
DEREF(value(h).hospitalisation_patient).NOM_PATIENT,  
DEREF(value(h).hospitalisation_patient).PRENOM_PATIENT ,  
DEREF(value(h).hospitalisation_patient).MUTUELLE FROM hospitalisation h  
where UPPER(DEREF(value(h).hospitalisation_patient).MUTUELLE)like 'MN%' AND  
DEREF(value(h).hospitalisation_service).CODE_SERVICE in  
(select s.CODE_SERVICE from service s where batiment='B');
```



```
SQL> SELECT h.lit,DEREF(value(h).hospitalisation_chambre).NUM_CHAMBRE,
2 Deref(value(h).hospitalisation_service).NOM_SERVICE,
3 Deref(value(h).hospitalisation_patient).NOM_PATIENT,
4 Deref(value(h).hospitalisation_patient).PRENOM_PATIENT ,
5 Deref(value(h).hospitalisation_patient).MUTUELLE FROM hospitalisation h
6 where UPPER(Deref(value(h).hospitalisation_patient).MUTUELLE)like 'MN%' AND
7 Deref(value(h).hospitalisation_service).CODE_SERVICE in
8 (select s.CODE_SERVICE from service s where batiment='B');
```

```

LIT Deref(VALUE(H).HOSPITALISATION_CHAMBRE).NUM_CHAMBRE
-----
Deref(VALUE(H).HOSPITALISATION_SERVICE). Deref(VALUE(H).HOSPITALISATION
-----
Deref(VALUE(H).HOSPITALISATION Deref(VALU
-----
```

```

1
Cardiologie
Mustapha
MNAM
SERIR
101
```

```

3
Cardiologie
Lamia
MNH
TAHMI
101
```

```

LIT Deref(VALUE(H).HOSPITALISATION_CHAMBRE).NUM_CHAMBRE
-----
Deref(VALUE(H).HOSPITALISATION_SERVICE). Deref(VALUE(H).HOSPITALISATION
-----
Deref(VALUE(H).HOSPITALISATION Deref(VALU
-----
```

```

2
Cardiologie
Mohamed
MNAM
TITOUCHE
105
```

11. 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.

```
select t.NUM_PATIENT,t.nbmedsoignants(),count(distinct(treat(value(e)as
tmedecin).SPECIALITE))
```

```
from employe e ,soigne s , patient t
```

```
where t.nbmedsoignants(>)3 AND
```

```
Deref(value(s).soigne_medecin).NUM_EMP=treat(value(e)as tmedecin).NUM_EMP AND
```

```
Deref(value(s).soigne_patient).NUM_PATIENT=t.NUM_PATIENT group
```

```
by(t.NUM_PATIENT,t.nbmedsoignants());
```

```

NUM_PATIENT T.NBMEDSOIGNANTS()
-----
COUNT(DISTINCT(TREAT(VALUE(E)ASTMEDECIN).SPECIALITE))
-----
      6              4              3
      76             4              3
     108             4              4

NUM_PATIENT T.NBMEDSOIGNANTS()
-----
COUNT(DISTINCT(TREAT(VALUE(E)ASTMEDECIN).SPECIALITE))
-----
     117             5              3
     145             4              4
     147             4              4

NUM_PATIENT T.NBMEDSOIGNANTS()
-----
COUNT(DISTINCT(TREAT(VALUE(E)ASTMEDECIN).SPECIALITE))
-----
     159             5              4
     172             4              3

8 rows selected.

```

12. Quelle est la moyenne des salaires des infirmiers(ères) par service ?

```

select distinct AVG( treat(value(e)as tinfirmier).salaire),DEREF(treat(value(e)as
tinfirmier).infirmier_service).CODE_SERVICE

from employe e where value (e)is of(tinfirmier)

group by(DEREF(treat(value(e)as tinfirmier).infirmier_service).CODE_SERVICE) ;

```

```

SQL> select distinct AVG( treat(value(e)as tinfirmier).salaire),DEREF(treat(value(e)as tinfirmier).infirmier_service).CODE_SERVICE
  2  from employe e where value (e)is of(tinfirmier)
  3  group by(DEREF(treat(value(e)as tinfirmier).infirmier_service).CODE_SERVICE) ;

AVG(TREAT(VALUE(E)ASTINFIRMIER).SALAIRE) DER
-----
      13788,1529 REA
      14771,4523 CHG
      15068,5413 CAR

```

13. Pour chaque service quel est le rapport entre le nombre d’infirmier(ères) affecté(es) au service et le nombre de patients hospitalisés dans le service ?

Requette:

```
select
s.code_service,s.nbinfirmierhospitalise(s.code_service,'i')/s.nbinfirmierhospitalise(s.code_service,'h')
as rapport from service s;
```

```
SQL> select s.code_service,s.nbinfirmierhospitalise(s.code_service,'i')/s.nbinfirmierhospitalise(s.code_service,'h') as rapport from service s;

COD      RAPPORT
---  -----
CAR      ,888888889
CHG              ,65
REA              ,7
```

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

```
select distinct treat(value(e)as tmedecin).NOM_EMP,treat(value(e)as tmedecin).PRENOM_EMP
from employe e,soigne s,hospitalisation h where
DEREF(value(s).soigne_medecin).NUM_EMP=treat(value(e)as tmedecin).NUM_EMP AND
DEREF(value(h).hospitalisation_patient).NUM_PATIENT=DEREF(value(s).soigne_patient).NUM_PATIENT
NT
group by(treat(value(e)as tmedecin).NOM_EMP,treat(value(e)as tmedecin).PRENOM_EMP)
having count (distinct Deref(value(h).hospitalisation_service).CODE_SERVICE)=(select count
(s.CODE_SERVICE)
from service s);
```

```
SQL> select distinct treat(value(e)as tmedecin).NOM_EMP,treat(value(e)as tmedecin).PRENOM_EMP
2  from employe e,soigne s,hospitalisation h where Deref(value(s).soigne_medecin).NUM_EMP=treat(value(e)as tmedecin).NUM_EMP AND
3  Deref(value(h).hospitalisation_patient).NUM_PATIENT=DEREF(value(s).soigne_patient).NUM_PATIENT
4  group by(treat(value(e)as tmedecin).NOM_EMP,treat(value(e)as tmedecin).PRENOM_EMP)
5  having count (distinct Deref(value(h).hospitalisation_service).CODE_SERVICE)=(select count (s.CODE_SERVICE)
6  from service s);

TREAT(VALUE(E)ASTMEDECIN).NOM_  TREAT(VALUE(E)ASTMEDECIN).PREN
-----
BAALI                          Souad
BASSI                          Fatima
RAHALI                          Ahcene
BAHBOUH                        Naïma
BENDALI                        Hacine
TEBIBEL                        Nabila
BELGHALI                       Mohammed
ABDELAZIZ                      Ahmed
MOHAMMEDI                      Mustapha

9 rows selected.
```