

**Base de Données avancées**

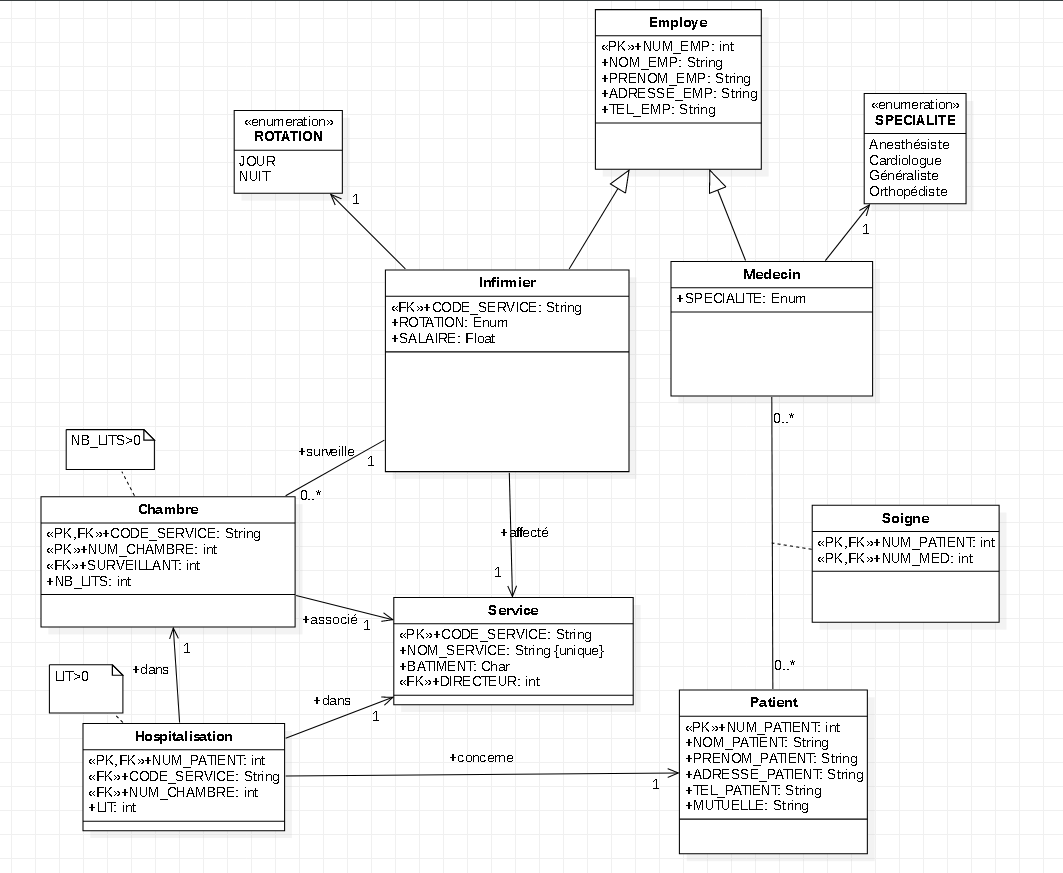
**RAPPORT DU Projet 01 SQL3-Oracle**

**Binôme:**

|  |  |  |
| --- | --- | --- |
| **MATRICULE** | **NOM** | **PRENOM** |
| **181831072145** | ZAIT | FOUAD |
| **181832028311** | HANANI | FOUAD |

**Partie I : Modélisation orientée objet**

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

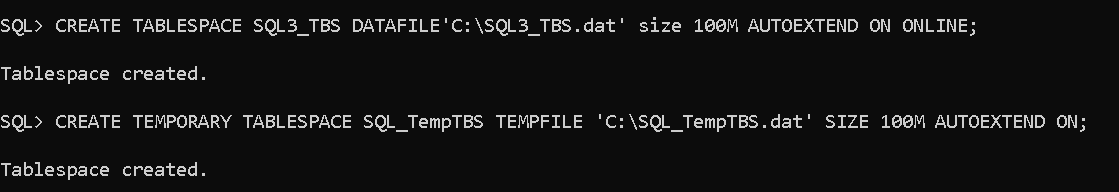


**Partie II : Création des TablesSpaces et utilisateur :**

**2- Créer deux TableSpaces SQL3\_TBS et SQL3\_TempTBS :**

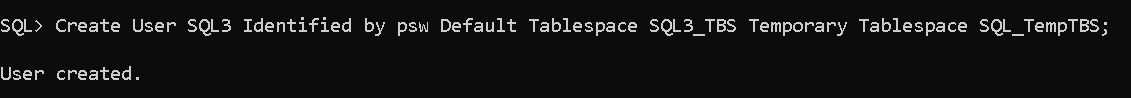
CREATE TABLESPACE SQL3\_TBS DATAFILE'C:\SQL3\_TBS.dat' size 100M AUTOEXTEND ON ONLINE;

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



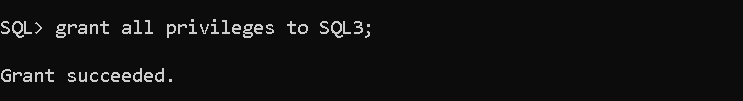
**3- Créer un utilisateur SQL3 en lui attribuant les deux tablespaces créés précédemment :**

Create User SQL3 Identified by psw Default Tablespace SQL3\_TBS Temporary Tablespace SQL\_TempTBS;



**4- Donner tous les privilèges à cet utilisateur:**

grant all privileges to SQL3;



**Partie III : Langage de définition de données**

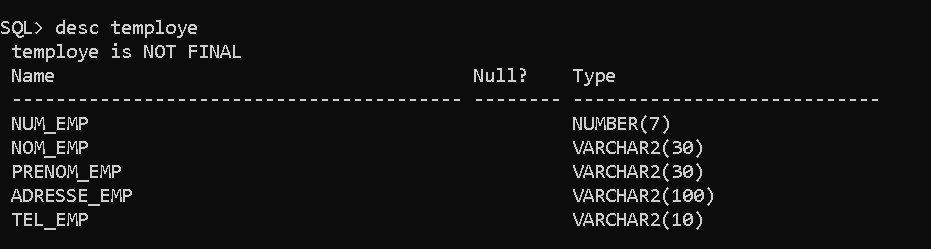
**5. En se basant sur le diagramme de classes fait, définir tous les types nécessaires. Prendre en compte toutes associations qui existent.**

**- Type temploye :**

**Requette:**

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;

/

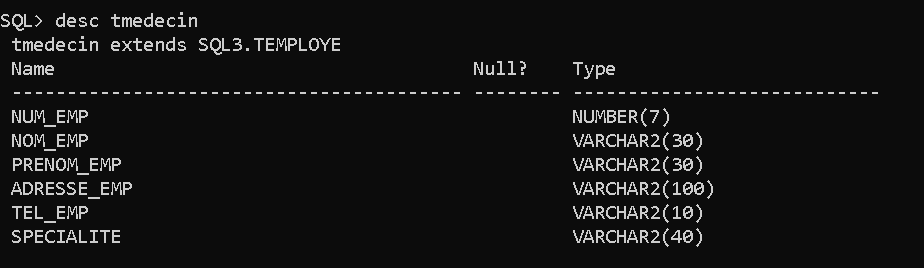


**-Type tmedecin :**

**Requette:**

CREATE TYPE tmedecin under temploye(SPECIALITE varchar2(40));

/

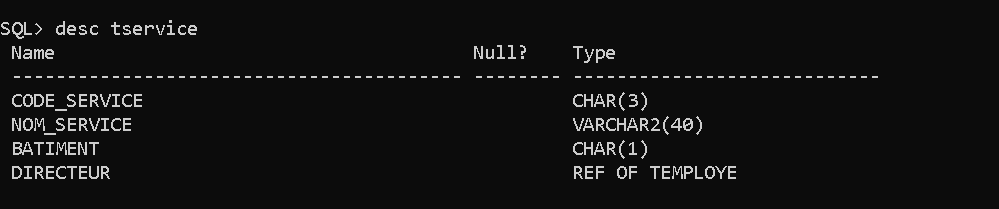


**-Type tservice** :

**Requette:**

CREATE TYPE tservice as object(CODE\_SERVICE char(3),NOM\_SERVICE varchar2(40),BATIMENT char,directeur ref temploye);

/

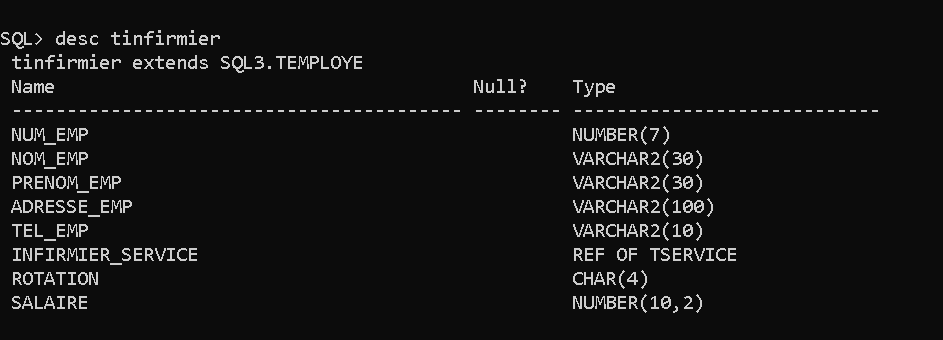


**-Type tinfirmier :**

**Requette :**

CREATE TYPE tinfirmier under temploye (NUM\_INF ref temploye,infirmier\_service ref tservice,ROTATION char(4),SALAIRE number(10,2)) ;

/



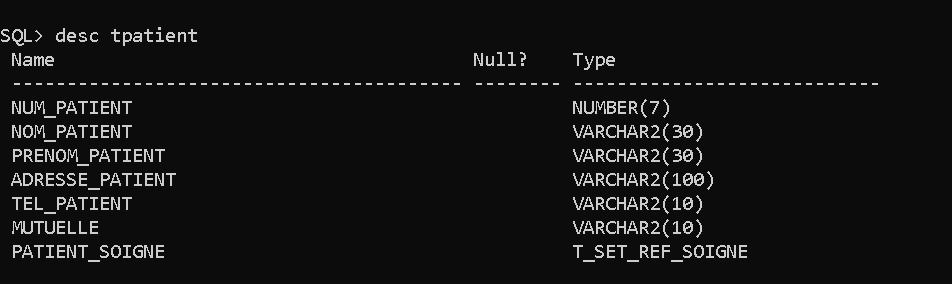
**-Type tpatient:**

**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));

/

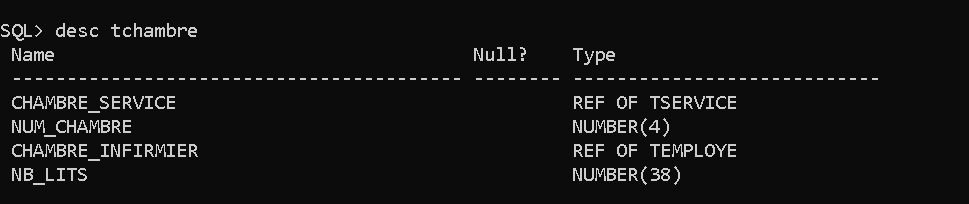


**-Type tchambre :**

**Requette:**

CREATE TYPE tchambre as object (chambre\_service ref tservice,NUM\_CHAMBRE Number(4), chambre\_infirmier ref temploye,NB\_LITS int);

/

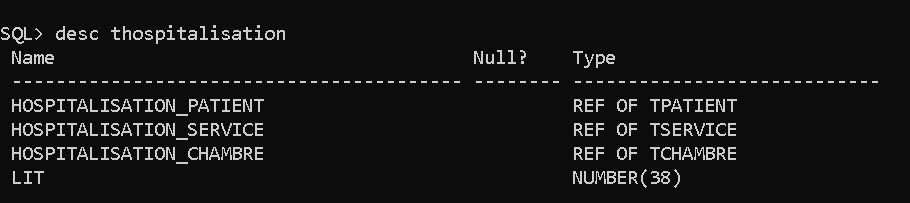


**-Type thospitalisation:**

**Requette:**

CREATE TYPE thospitalisation as object(hospitalisation\_patient ref tpatient,hospitalisation\_service ref tservice,hospitalisation\_chambre ref tchambre,LIT int);

/

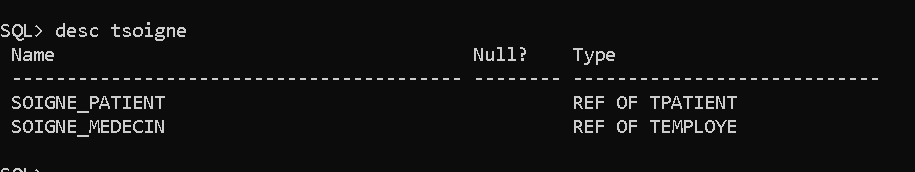


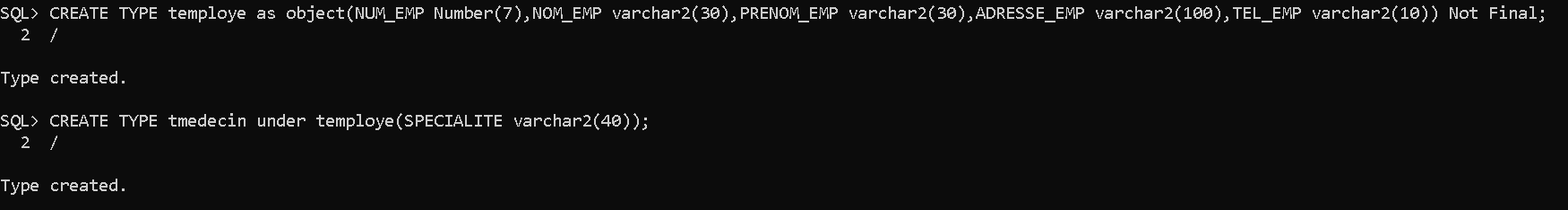
**-Type tsoigne :**

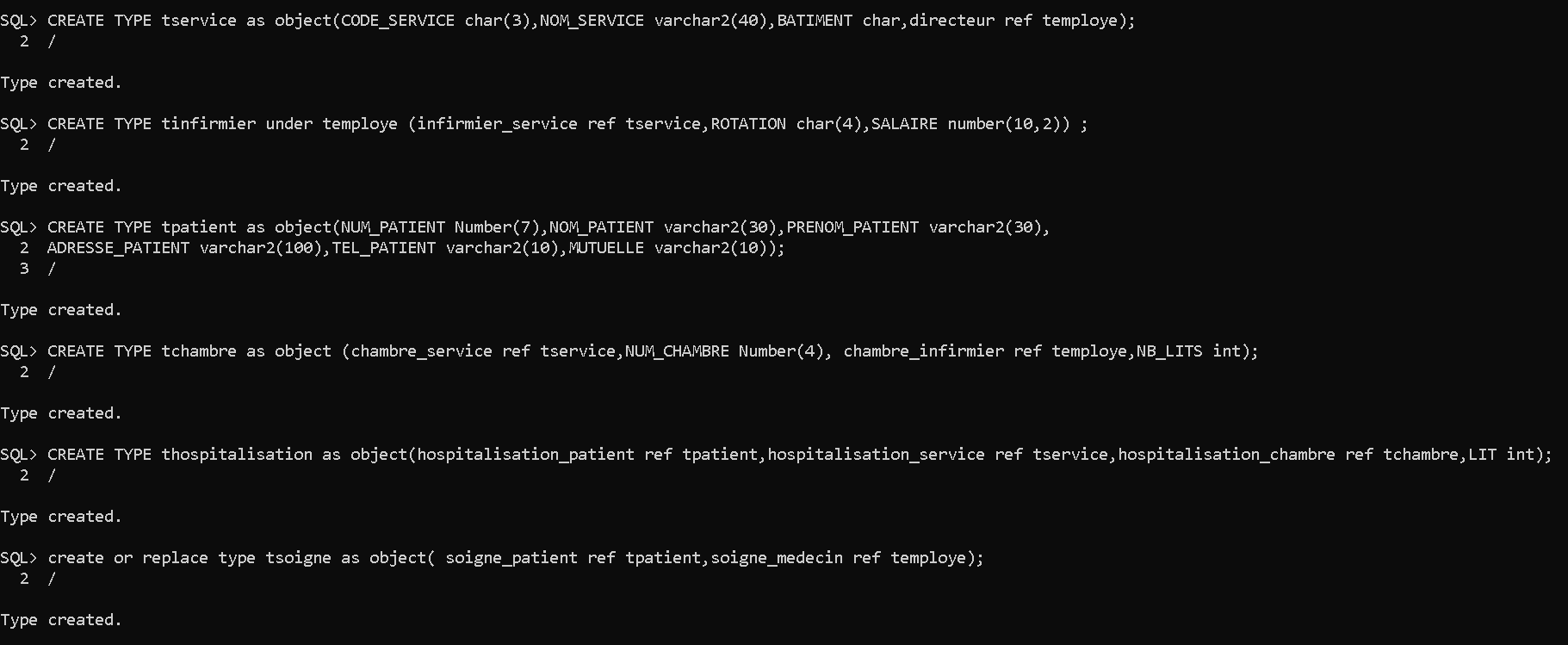
**Requette:**

create or replace type tsoigne as object( soigne\_patient ref tpatient,soigne\_medecin ref temploye);

/







**-Prendre en compte les associations qui existent :**

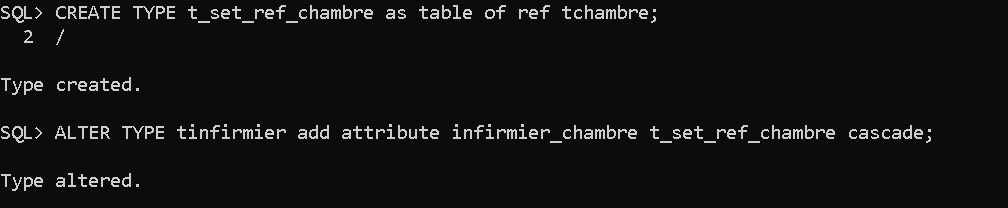
**-association infirmier chambre :**

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;



**-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 temploye)

**-association médecin soigne patient :**

-soigne medecin : on ajoute à soigne l’attribut soigne\_medecin à soigne (soigne\_medecin ref temploye) 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 tpaient une table de ref de tsoigne (patient\_soigne t\_set\_ref\_soigne)

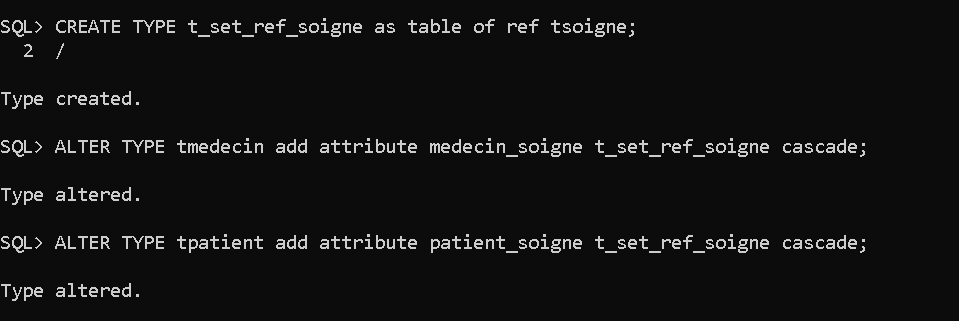
**Requette :**

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;



**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 assossié.

**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:**

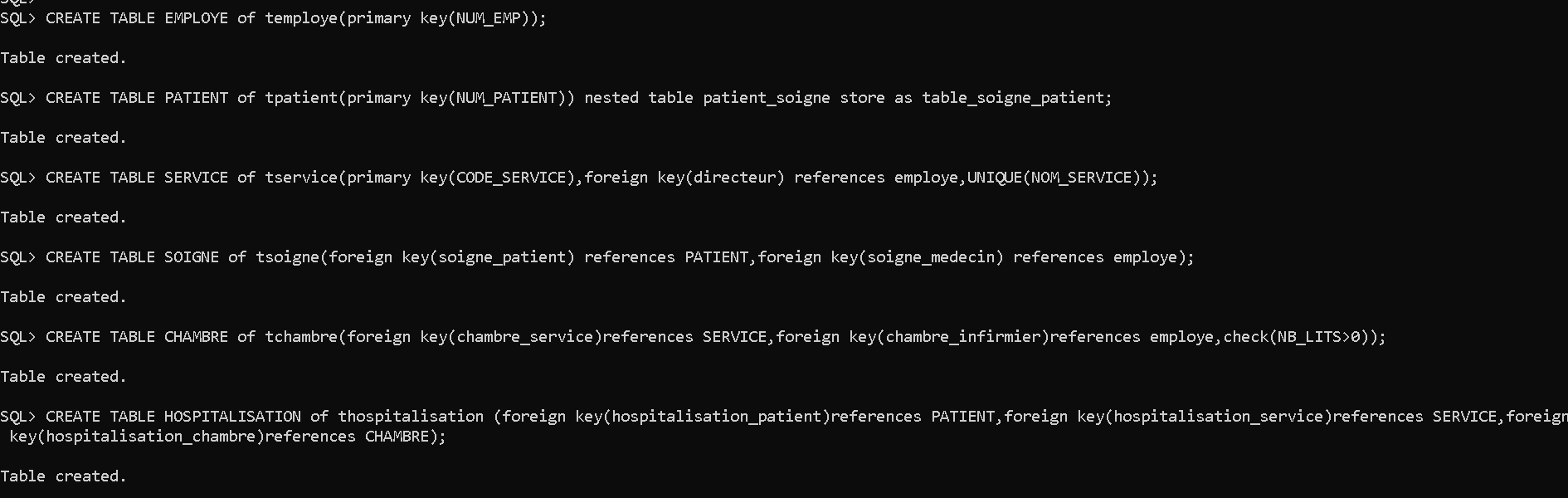
CREATE TABLE SOIGNE of tsoigne(foreign key(soigne\_patient) references PATIENT,foreign key(soigne\_medecin) references employe);

**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);



**7-Définitions des méthodes :**

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

**On ajoute d’abord la fonction au type temploye:**

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;

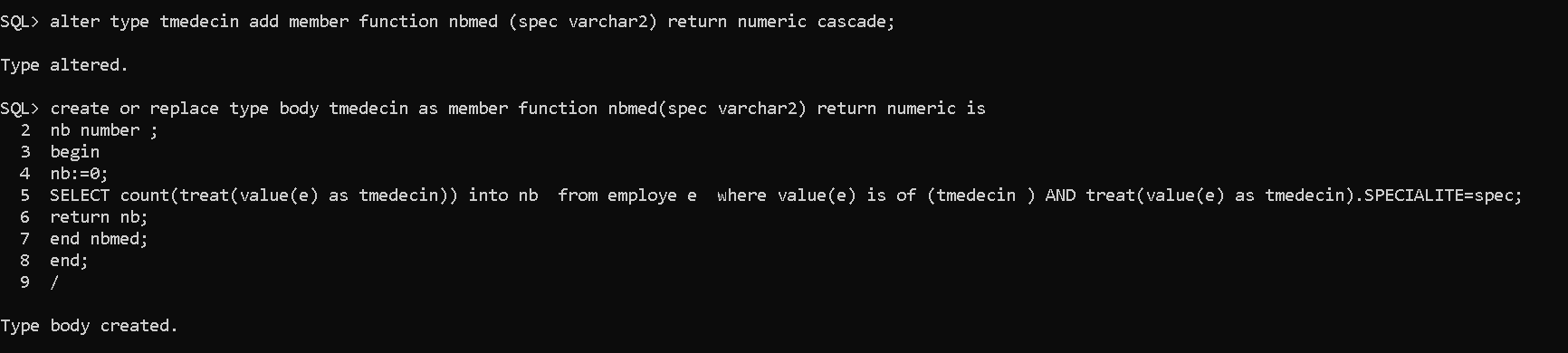
SELECT count(treat(value(e) as tmedecin)) into nb from employe e where value(e) is of (tmedecin ) AND treat(value(e) as tmedecin).SPECIALITE=spec;

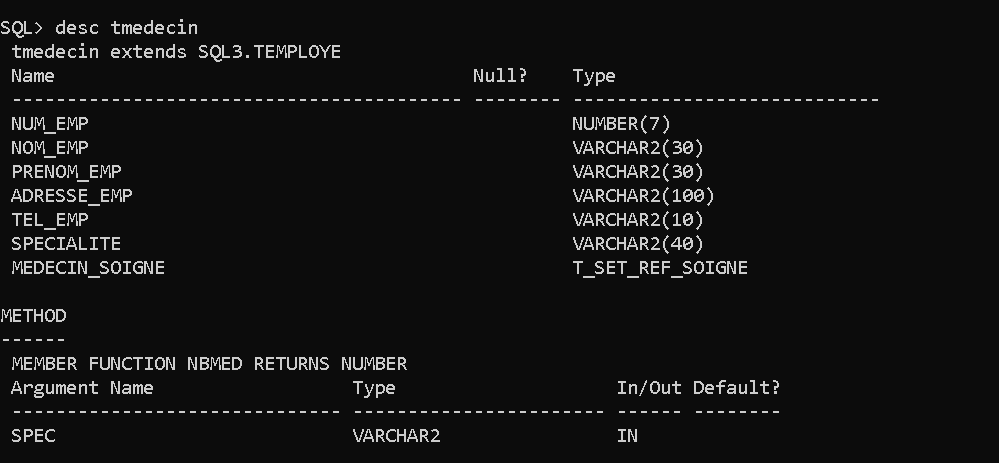
return nb;

end nbmed;

end;

/





**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 carractére donnée est ‘i’ cette methode nous calcule le nombre d’infirmiers sinon si le carractére donnée est h alors cette methode nous calcule le nombre d’hospitalisées pour le service carractérisé 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 (tinfirmier) AND deref(treat(value(e) as tinfirmier).infirmier\_service).CODE\_SERVICE=code;

return nb;

elsif b='h' then

SELECT COUNT (\*) into nb from hospitalisation h where h.hospitalisation\_service.CODE\_SERVICE= code;

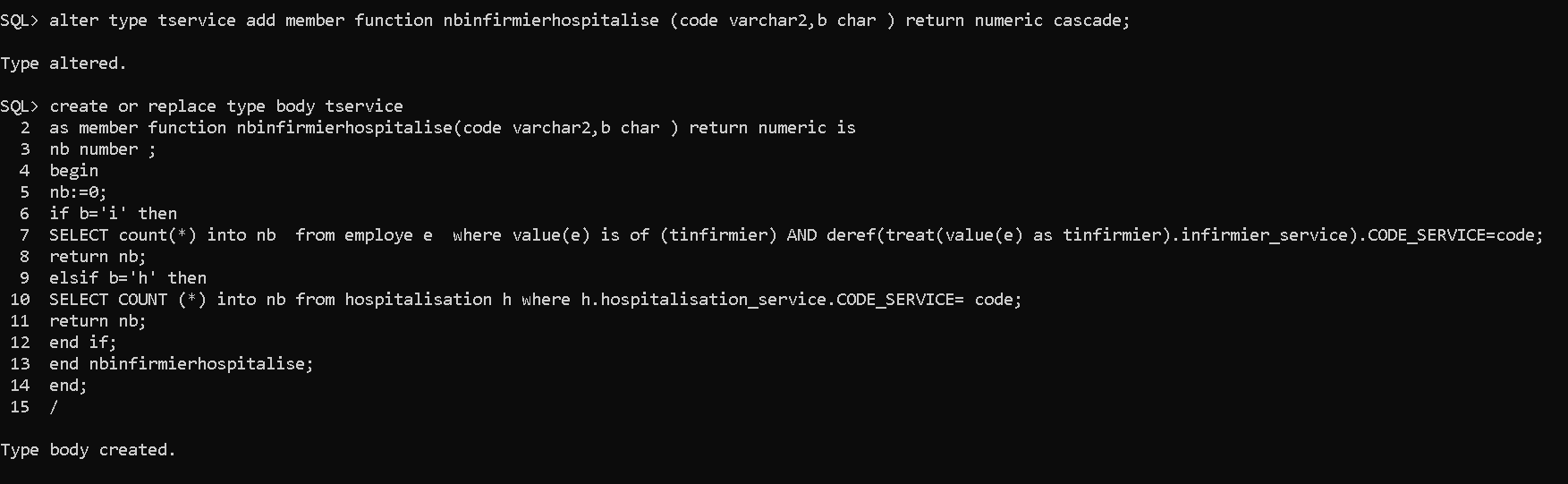
return nb;

end if;

end nbinfirmierhospitalise;

end;

/



**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;

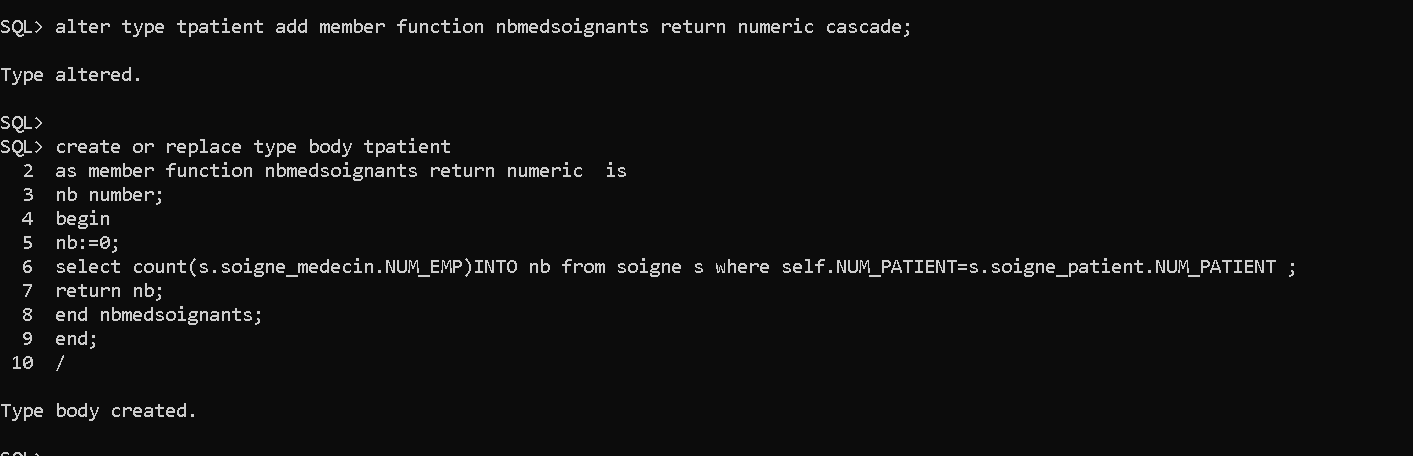
select count(s.soigne\_medecin.NUM\_EMP)INTO nb from soigne s where self.NUM\_PATIENT=s.soigne\_patient.NUM\_PATIENT ;

return nb;

end nbmedsoignants;

end;

/



**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 tinfirmier la procédure:

alter type temploye add member procedure verifsalaire cascade;

**La procédure :**

create or replace type body temploye

as member procedure verifsalaire is

begin

if(self is of (tinfirmier) AND treat(self as tinfirmier).SALAIRE between 10000 AND 30000)then dbms\_output.put\_line('vérification positive');

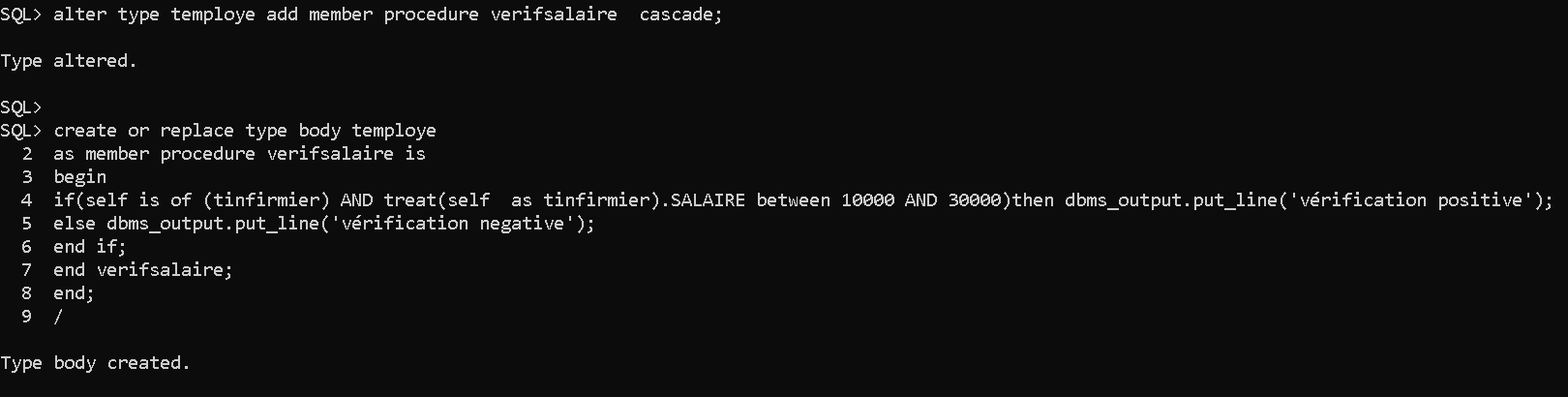
else dbms\_output.put\_line('vérification negative');

end if;

end verifsalaire;

end;

/



**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);

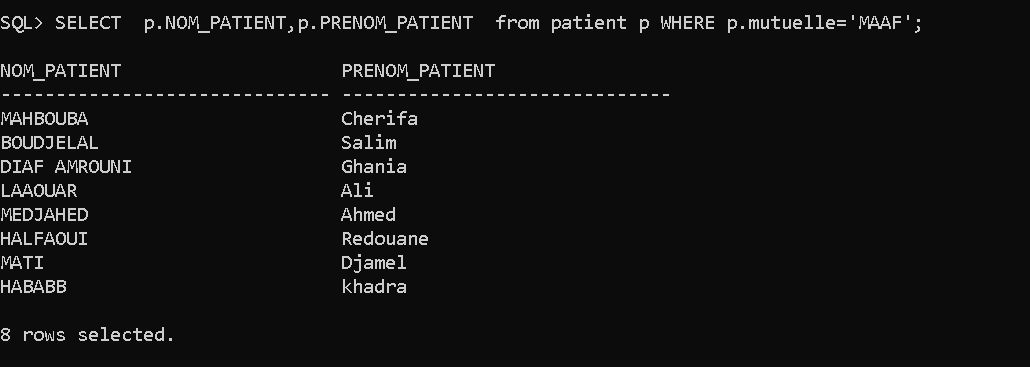
**Table Soigne** : INSERT INTO SOIGNE VALUES ((SELECT ref(p) from PATIENT p where p.NUM\_PATIENT=13),(SELECT ref(e) from EMPLOYE e where e.NUM\_EMP=4));

**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';



**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');



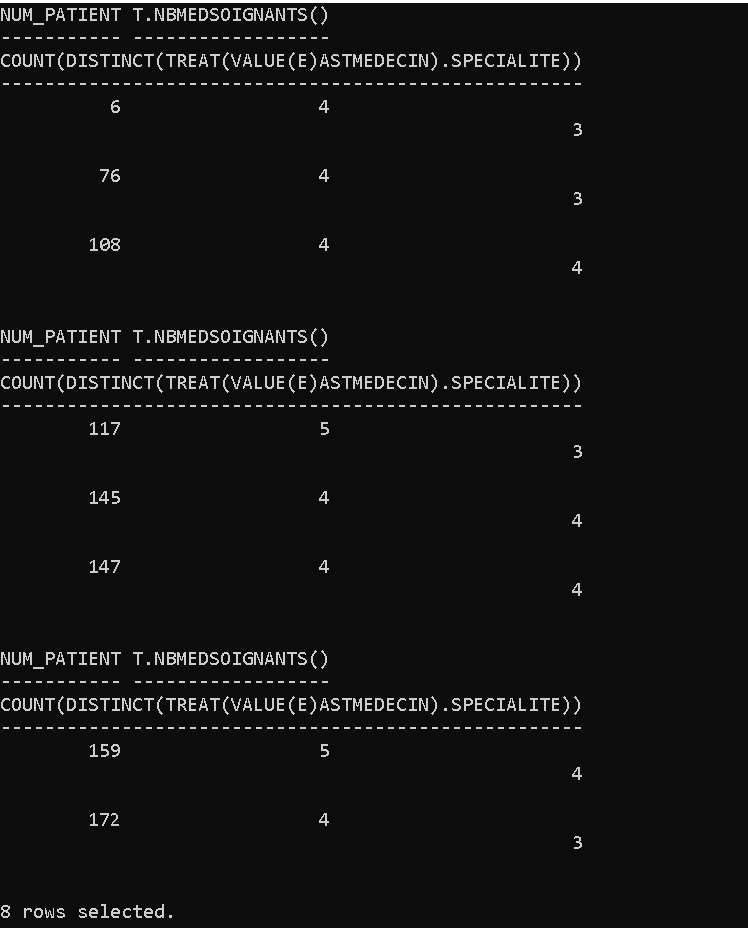
**11. Pour chaque patient soigné par plus de 3 médecins donner le nombre total de ses médecins ainsi que le nombre correspondant de spécialités médicales concernées.**

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());

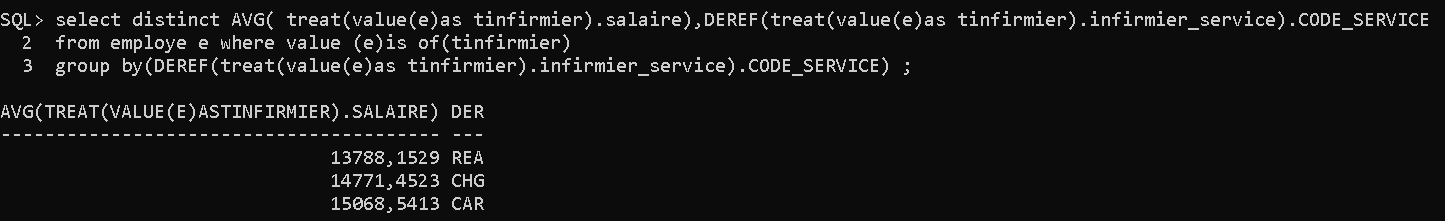


**12. Quelle est la moyenne des salaires des infirmiers(ères) par service ?**

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)

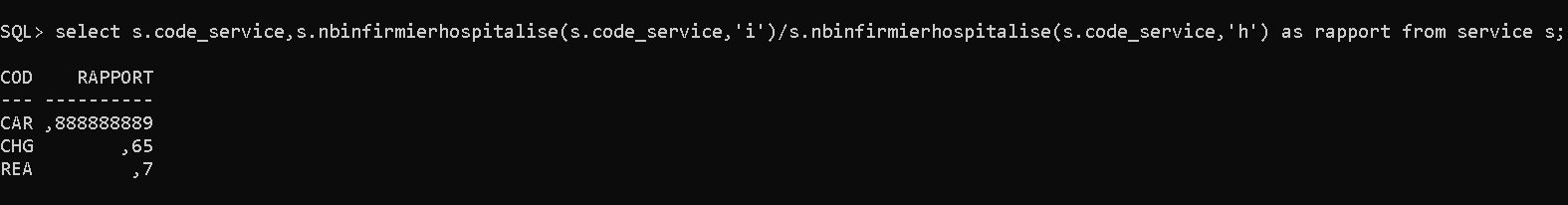
group by(DEREF(treat(value(e)as tinfirmier).infirmier\_service).CODE\_SERVICE) ;



**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;



**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

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);

