



Université des Sciences et de la Technologie Houari Boumediene

Faculté Informatique

Master Systèmes Informatiques intelligents

Module : Bases de Données Avancées

RAPPORT DU PREMIER PROJET BASES DE DONNEES AVANCEES : SQL3

Réalisé par :

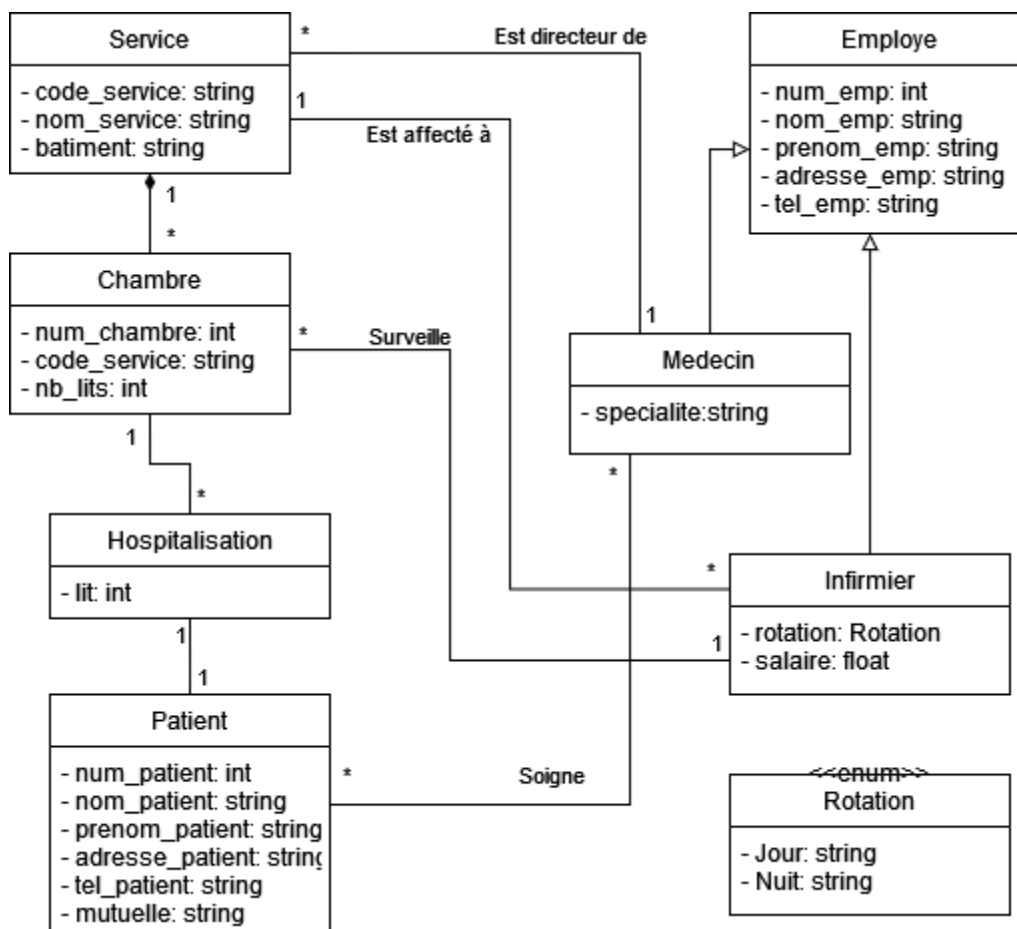
AIT AMARA Mohamed, 181831072170

HAMMAL Ayoub, 181831048403

Année universitaire : 2021 / 2022

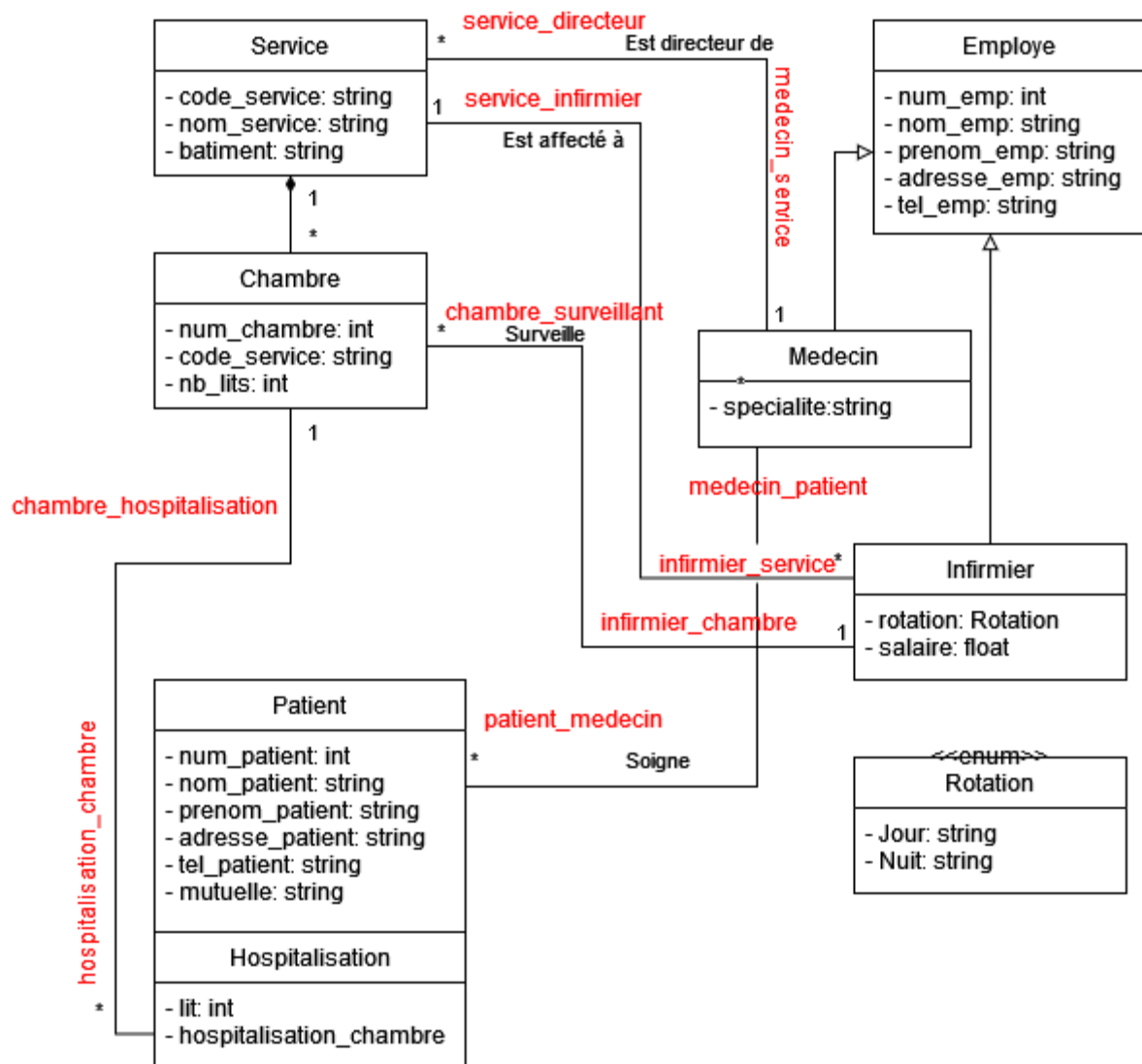
Partie I : Modélisation orientée objet

1 – Diagramme de classe :



Les associations deviennent des tableaux de références comme suit :

L'hospitalisation devient un attribut interne du type patient.



Partie II : Création des TableSpaces et utilisateur

2 – Création des TableSpaces :

```

SQL> CREATE TABLESPACE sql3_tbs
  2 DATAFILE 'c:\sql3_tbs.dbf'
  3 SIZE 50M
  4 AUTOEXTEND ON;

Tablespace created.

SQL>
SQL> CREATE TEMPORARY TABLESPACE sql3_temptbs
  2 TEMPFILE 'c:\sql3_temptbs.dbf'
  3 SIZE 50M
  4 AUTOEXTEND ON;

Tablespace created.

```

3 – Création de l'utilisateur SQL3 :

```
SQL> CREATE USER sql3
  2 IDENTIFIED BY zerty
  3 DEFAULT TABLESPACE sql3_tbs
  4 TEMPORARY TABLESPACE sql3_temptbs
  5 QUOTA UNLIMITED ON sql3_tbs;

User created.
```

4 – Attribution des privilèges pour l'utilisateur SQL3 :

```
SQL> GRANT ALL PRIVILEGES
  2 TO sql3;

Grant succeeded.
```

Partie III : Langage de définition de données

5 – Définition des types nécessaires :

```
SQL> CREATE OR REPLACE TYPE temploye;
  2 /

Type created.

SQL> CREATE OR REPLACE TYPE tmedecin;
  2 /

Type created.

SQL> CREATE OR REPLACE TYPE tinfirmier;
  2 /

Type created.

SQL> CREATE OR REPLACE TYPE thospitalisation;
  2 /

Type created.

SQL> CREATE OR REPLACE TYPE tpatient;
  2 /

Type created.

SQL> CREATE OR REPLACE TYPE tchambre;
  2 /

Type created.

SQL> CREATE OR REPLACE TYPE tservice;
  2 /

Type created.
```

Création des tables de références :

```
SQL> CREATE OR REPLACE TYPE t_set_ref_service AS TABLE OF REF tservice;  
2 /  
  
Type created.  
  
SQL> CREATE OR REPLACE TYPE t_set_ref_infirmier AS TABLE OF REF tinfirmier;  
2 /  
  
Type created.  
  
SQL> CREATE OR REPLACE TYPE t_set_ref_chambre AS TABLE OF REF tchambre;  
2 /  
  
Type created.  
  
SQL> CREATE OR REPLACE TYPE t_set_ref_patient AS TABLE OF REF tpatient;  
2 /  
  
Type created.  
  
SQL> CREATE OR REPLACE TYPE t_set_ref_medecin AS TABLE OF REF tmedecin;  
2 /  
  
Type created.
```

Création des types finaux :

```
SQL> CREATE OR REPLACE TYPE temploye AS OBJECT (  
  2     num_emp INTEGER,  
  3     nom_emp VARCHAR2(100),  
  4     prenom_emp VARCHAR2(100),  
  5     adresse_emp VARCHAR2(150),  
  6     tel_emp VARCHAR2(15)  
  7 ) NOT FINAL;  
  8 /
```

Type created.

```
SQL>  
SQL> CREATE OR REPLACE TYPE tmedecin UNDER temploye (  
  2     specialite VARCHAR2(30),  
  3     directeur_service t_set_ref_service,  
  4     medecin_patient t_set_ref_patient  
  5 );  
  6 /
```

Type created.

```
SQL>  
SQL> CREATE OR REPLACE TYPE tinfirmier UNDER temploye (  
  2     rotation VARCHAR2(4),  
  3     salaire FLOAT(2),  
  4     infirmier_service REF tservice,  
  5     infirmier_chambre t_set_ref_chambre  
  6 );  
  7 /
```

Type created.

```
SQL>  
SQL> CREATE OR REPLACE TYPE thospitalisation AS OBJECT (  
  2     lit INTEGER,  
  3     hospitalisation_chambre REF tchambre  
  4 );  
  5 /
```

Type created.

```

SQL> CREATE OR REPLACE TYPE tpatient AS OBJECT (
  2     num_patient INTEGER,
  3     nom_patient VARCHAR2(100),
  4     prenom_patient VARCHAR2(100),
  5     adresse_patient VARCHAR2(150),
  6     tel_patient VARCHAR2(15),
  7     mutuelle VARCHAR2(10),
  8     hospitalisation thospitalisation,
  9     patient_medecin t_set_ref_medecin
10 );
11 /

```

Type created.

```

SQL>
SQL> CREATE OR REPLACE TYPE tchambre AS OBJECT (
  2     code_service VARCHAR2(3),
  3     num_chambre INTEGER,
  4     nb_lits INTEGER,
  5     chambre_service REF tservice,
  6     chambre_surveillant REF tinfirmier,
  7     chambre_patient t_set_ref_patient
  8 );
  9 /

```

Type created.

```

SQL>
SQL> CREATE OR REPLACE TYPE tservice AS OBJECT (
  2     code_service VARCHAR2(3),
  3     nom_service VARCHAR2(50),
  4     batiment VARCHAR2(1),
  5     service_directeur REF tmedecin,
  6     service_infirmier t_set_ref_infirmier,
  7     service_chambre t_set_ref_chambre
  8 );
  9 /

```

Type created.

6 – Définition des tables nécessaires :

Pour ce qui est des types *tmedecin* et *tinfirmier*, nous avons défini une table pour chaque type au lieu d'utiliser un seul table de type *temploie* car ces deux sous types contiennent des tables imbriquées, et nous ne pouvons pas définir le stockage des tables imbriquées des sous types dans définition de la table du super type.

```
SQL> CREATE TABLE medecin OF tmedecin (  
2     PRIMARY KEY (num_emp)  
3 )  
4 NESTED TABLE directeur_service STORE AS table_directeur_service,  
5 NESTED TABLE medecin_patient STORE AS table_medecin_patient;
```

Table created.

```
SQL>  
SQL> CREATE TABLE service OF tservice (  
2     PRIMARY KEY (code_service),  
3     FOREIGN KEY (service_directeur) REFERENCES medecin  
4 )  
5 NESTED TABLE service_infirmier STORE AS table_service_infirmier,  
6 NESTED TABLE service_chambre STORE AS table_service_chmabre;
```

Table created.

```
SQL>  
SQL> CREATE TABLE infirmier OF tinfirmier (  
2     PRIMARY KEY (num_emp),  
3     FOREIGN KEY (infirmier_service) REFERENCES service,  
4     CHECK (rotation in ('JOUR', 'NUIT'))  
5 )  
6 NESTED TABLE infirmier_chambre STORE AS table_infirmier_chambre;
```

Table created.

```
SQL> CREATE TABLE chambre OF tchambre (  
2     PRIMARY KEY (num_chambre, code_service),  
3     FOREIGN KEY (chambre_service) REFERENCES service,  
4     FOREIGN KEY (chambre_surveillant) REFERENCES infirmier  
5 )  
6 NESTED TABLE chambre_patient STORE AS table_chambre_patient;
```

Table created.

```
SQL>  
SQL> CREATE TABLE patient OF tpatient (  
2     PRIMARY KEY (num_patient),  
3     FOREIGN KEY (hospitalisation.hospitalisation_chambre) REFERENCES chambre  
4 )  
5 NESTED TABLE patient_medecin STORE AS table_patient_medecin;
```

Table created.

7 – Définition des méthodes

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


```
SQL> ALTER TYPE tmedecin ADD STATIC FUNCTION nb_med_spe(spe VARCHAR2) RETURN INTEGER CASCADE;
Type altered.

SQL>
SQL> CREATE OR REPLACE TYPE BODY tmedecin AS
  2     STATIC FUNCTION nb_med_spe(spe VARCHAR2) RETURN INTEGER IS
  3         nb INTEGER;
  4     BEGIN
  5         SELECT count(m.num_emp) INTO nb
  6         FROM medecin m
  7         WHERE m.specialite = spe
  8         GROUP BY m.specialite;
  9         RETURN nb;
10     END;
11 END;
12 /

Type body created.
```

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

```
SQL> ALTER TYPE tservice ADD MEMBER FUNCTION nb_inf RETURN INTEGER CASCADE;
Type altered.

SQL> ALTER TYPE tservice ADD MEMBER FUNCTION nb_pat RETURN INTEGER CASCADE;
Type altered.

SQL>
SQL> CREATE OR REPLACE TYPE BODY tservice AS
  2     MEMBER FUNCTION nb_inf RETURN INTEGER IS
  3         nb INTEGER;
  4     BEGIN
  5         SELECT COUNT(DISTINCT t.column_value) INTO nb
  6         FROM TABLE(self.service_infirmier) t;
  7         RETURN nb;
  8     END nb_inf;
  9     MEMBER FUNCTION nb_pat RETURN INTEGER IS
10         nb INTEGER;
11     BEGIN
12         SELECT COUNT(DISTINCT b.column_value) INTO nb
13         FROM chambre c, TABLE(self.service_chambre) l, TABLE(c.chambre_patient) b
14         WHERE REF(c) = l.column_value;
15         RETURN nb;
16     END nb_pat;
17 END;
18 /

Type body created.
```

- Pour calculer pour chaque patient le nombre total de ses médecins soignants.

```

SQL> ALTER TYPE tpatient ADD MEMBER FUNCTION nb_med RETURN INTEGER CASCADE;

Type altered.

SQL>
SQL> CREATE OR REPLACE TYPE BODY tpatient AS
  2     MEMBER FUNCTION nb_med RETURN INTEGER IS
  3         nb INTEGER;
  4     BEGIN
  5         SELECT COUNT(DISTINCT m.column_value) INTO nb
  6         FROM TABLE(self.patient_medecin) m;
  7         RETURN nb;
  8     END nb_med;
  9 END;
10 /

Type body created.

```

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

```

SQL> ALTER TYPE tinfirmier ADD MEMBER PROCEDURE verif_salaire CASCADE;

Type altered.

SQL>
SQL> CREATE OR REPLACE TYPE BODY tinfirmier AS
  2     MEMBER PROCEDURE verif_salaire IS
  3     BEGIN
  4         IF self.salaire BETWEEN 10000 AND 30000 THEN
  5             DBMS_OUTPUT.PUT_LINE('verification positive');
  6         ELSE
  7             DBMS_OUTPUT.PUT_LINE('verification negative');
  8         END IF;
  9     END verif_salaire;
10 END;
11 /

Type body created.

```

Partie IV : Langage de manipulation de données

8 – Remplissage des tables :

Nous remplissons les tables sans les associations d’abord, et puis nous créons les références entre les tables.

Pour la table medecin :

```

SQL> INSERT INTO medecin VALUES (
2      tmedecin (
3          4,
4          'BOUROUBI',
5          'Taous',
6          'Lotissement Dauphin n° 30 DRARIA / ALGER',
7          '021356085',
8          'Orthopédiste',
9          t_set_ref_service(),
10         t_set_ref_patient()
11     )
12 );

1 row created.

SQL> INSERT INTO medecin VALUES (
2      tmedecin (
3          7,

```

Pour la table infirmier :

```

SQL> INSERT INTO infirmier VALUES (
2      tinfirmier (
3          12,
4          'HADJ',
5          'Zouhir',
6          'Cité de la Mosquée Bt 14-Boufarik-Blida',
7          '025474882',
8          'JOUR',
9          12560.78,
10         NULL,
11         t_set_ref_chambre()
12     )
13 );

1 row created.

SQL> INSERT INTO infirmier VALUES (
2      tinfirmier (

```

Pour la table patient :

```

SQL> INSERT INTO patient VALUES (
2      tpatient (
3          1,
4          'GRIGAHCINE',
5          'Nacer',
6          '95, Bd Bougara-El biar-Alger',
7          '021920313',
8          'MNAM',
9          NULL,
10         t_set_ref_medecin()
11     )
12 );

1 row created.

SQL> INSERT INTO patient VALUES (
2      tpatient (
3          2,

```

Pour la table service :

```
SQL> INSERT INTO service VALUES (  
  2      tservice (  
  3          'CAR',  
  4          'Cardiologie',  
  5          'B',  
  6          NULL,  
  7          t_set_ref_infirmier(),  
  8          t_set_ref_chambre()  
  9      )  
10 );  
  
1 row created.  
  
SQL> INSERT INTO service VALUES (  
  2      tservice (  
  3          'CHG',  
  4          'Chirurgie générale',  
  5          'A',  
  6          NULL,  
  7          t_set_ref_infirmier(),  
  8          t_set_ref_chambre()  
  9      )  
10 );  
  
1 row created.  
  
SQL> INSERT INTO service VALUES (  
  2      tservice (  
  3          'REA',  
  4          'Réanimation et Traumatologie',  
  5          'A',  
  6          NULL,  
  7          t_set_ref_infirmier(),  
  8          t_set_ref_chambre()  
  9      )  
10 );  
  
1 row created.
```

Pour la table chambre :

```
SQL> INSERT INTO chambre VALUES (  
  2      tchambre (  
  3          'CAR',  
  4          101,  
  5          3,  
  6          NULL,  
  7          NULL,  
  8          t_set_ref_patient()  
  9      )  
10 );  
  
1 row created.  
  
SQL> INSERT INTO chambre VALUES (  
  2      tchambre (  
  3          'CAR'
```

Pour l'association entre infirmier et chambre :

```
SQL> UPDATE chambre SET chambre_surveillant = (SELECT REF(i) FROM infirmier i WHERE i.num_emp = 95) WHERE code_service = 'CAR' AND num_chambre = 101;
1 row updated.

SQL> INSERT INTO TABLE(SELECT i.infirmier_chambre FROM infirmier i WHERE i.num_emp = 95) VALUES (
  2   (SELECT REF(c) FROM chambre c WHERE c.code_service = 'CAR' AND c.num_chambre = 101)
  3 );
1 row created.

SQL>
SQL> UPDATE chambre SET chambre_surveillant = (SELECT REF(i) FROM infirmier i WHERE i.num_emp = 95) WHERE code_service = 'CAR' AND num_chambre = 102;
```

Pour l'association entre infirmier et service :

```
SQL> UPDATE infirmier SET infirmier_service = (SELECT REF(s) FROM service s WHERE s.code_service = 'CHG') WHERE num_emp = 195;
1 row updated.

SQL> INSERT INTO TABLE(SELECT s.service_infirmier FROM service s WHERE s.code_service = 'CHG') VALUES (
  2   (SELECT REF(i) FROM infirmier i WHERE i.num_emp = 195)
  3 );
1 row created.
```

Pour l'association entre medecin et service :

```
SQL> UPDATE service SET service_directeur = (SELECT REF(m) FROM medecin m WHERE m.num_emp = 19) WHERE code_service = 'REA';
1 row updated.

SQL> INSERT INTO TABLE(SELECT m.directeur_service FROM medecin m WHERE m.num_emp = 19) VALUES (
  2   (SELECT REF(s) FROM service s WHERE s.code_service = 'REA')
  3 );
1 row created.
```

Pour l'association entre service et chambre :

```
SQL> UPDATE chambre SET chambre_service = (SELECT REF(s) FROM service s WHERE s.code_service = 'REA') WHERE code_service = 'REA' AND num_chambre = 108;
1 row updated.

SQL> INSERT INTO TABLE(SELECT s.service_chambre FROM service s WHERE s.code_service = 'REA') VALUES (
  2   (SELECT REF(c) FROM chambre c WHERE c.code_service = 'REA' AND c.num_chambre = 108)
  3 );
1 row created.
```

Pour l'association d'hospitalisation en chambre et patient :

```
SQL> UPDATE patient p
  2 SET hospitalisation = thospitalisation(1,(SELECT REF(c) FROM chambre c WHERE c.num_chambre = 108 AND c.code_service = 'REA'))
  3 WHERE p.num_patient = 117;
1 row updated.

SQL> INSERT INTO TABLE(SELECT c.chambre_patient FROM chambre c WHERE c.num_chambre = 108 AND c.code_service = 'REA') VALUES (
  2   (SELECT REF(p) FROM patient p WHERE p.num_patient = 117)
  3 );
1 row created.
```

Pour l'association entre medecin et patient :

```
SQL> INSERT INTO TABLE(SELECT p.patient_medecin FROM patient p WHERE p.num_patient = 192) VALUES (
  2      (SELECT REF(m) FROM medecin m WHERE m.num_emp = 179)
  3 );

1 row created.

SQL> INSERT INTO TABLE(SELECT m.medecin_patient FROM medecin m WHERE m.num_emp = 179) VALUES (
  2      (SELECT REF(p) FROM patient p WHERE p.num_patient = 192)
  3 );

1 row created.
```

Test des méthodes définies précédemment :

```
SQL> SELECT DISTINCT specialite, tmedecin.nb_med_spe(specialite)
  2      FROM medecin;

SPECIALITE                                TMEDECIN.NB_MED_SPE(SPECIALITE)
-----
Pneumologue                                5
Orthopédiste                               5
Traumatologue                              5
Cardiologie                                8
Radiologue                                  4
Anesthésiste                               5

6 rows selected.
```

```
SQL> SELECT s.code_service, s.nb_inf()
  2      FROM service s;

COD S.NB_INF()
---
CAR      8
CHG     13
REA      7

SQL> SELECT s.code_service, s.nb_pat()
  2      FROM service s;

COD S.NB_PAT()
---
CAR      9
CHG     20
REA     10
```

```
SQL> SELECT p.num_patient, p.nb_med()
2      FROM patient p;
```

NUM_PATIENT	P.NB_MED()
1	2
3	3
6	4
13	3
14	1
21	3
23	3
33	2
35	2
36	3
37	1

```
SQL> DECLARE
2      inf tinfirmer;
3 BEGIN
4      SELECT value(i) INTO inf
5      FROM infirmer i
6      WHERE i.num_emp = 12;
7      inf.verif_salaire;
8 END;
9 /
verification positive
PL/SQL procedure successfully completed.
```

Partie V : Langage d'interrogation de données

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

```
SQL> SELECT p.prenom_patient AS "prenom",
2      p.nom_patient AS "nom"
3 FROM patient p
4 WHERE p.mutuelle = 'MAAF';
```

prenom

nom

Cherifa
MAHBOUBA

Salim
BOUDJELAL

AMROUNI
DIAF

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.

```
SQL> SELECT p.hospitalisation.lit AS "numero lit",
2      Deref(p.hospitalisation.hospitalisation_chambre).num_chambre AS "numero chambre",
3      Deref(Deref(p.hospitalisation.hospitalisation_chambre).chambre_service).nom_service AS "nom service",
4      p.prenom_patient AS "prenom patient",
5      p.nom_patient AS "nom patient",
6      p.mutuelle AS "mutuelle"
7  FROM patient p
8  WHERE p.hospitalisation IS NOT NULL AND
9        p.mutuelle LIKE 'MN%' AND
10       Deref(Deref(p.hospitalisation.hospitalisation_chambre).chambre_service).batiment = 'B';
```

numero lit	numero chambre	nom service	prenom patient	nom patient	mutuelle
3	101	Cardiologie	Lamia	TAHMI	MNH

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.

```
SQL> SELECT p.num_patient AS "numero patient",
2      p.prenom_patient AS "prenom patient",
3      p.nom_patient AS "nom patient",
4      COUNT(DISTINCT pm.column_value) AS "nombre de medecins",
5      COUNT(DISTINCT Deref(pm.column_value).specialite) AS "nombre de specialites"
6  FROM patient p,
7      TABLE(p.patient_medecin) pm
8  GROUP BY p.num_patient,
9          p.prenom_patient,
10         p.nom_patient
11  HAVING count(pm.column_value) > 3;
```

numero patient	prenom patient	nom patient	nombre de medecins	nombre de specialites
6	Aboukhalil	ABERKANE	4	3

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


```
SQL> SELECT s.code_service AS "code service",
2      s.nom_service AS "nom service",
3      AVG(DEREF(si.column_value).salaire) AS "salaire moyen"
4 FROM service s,
5      TABLE(s.service_infirmier) si
6 GROUP BY s.code_service,
7      s.nom_service;
```

cod nom service	salaire moyen
REA Réanimation et Traumatologie	11428,5714
CHG Chirurgie générale	13076,9231
CAR Cardiologie	12500

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 ?

```
SQL> SELECT s.code_service AS "code service",
2      s.nom_service AS "nom service",
3      COUNT(DISTINCT si.column_value) AS "nombre infirmiers",
4      COUNT(DISTINCT cp.column_value) AS "nombre patients",
5      (COUNT(DISTINCT si.column_value) / COUNT(DISTINCT cp.column_value)) AS "rapport inf/pat"
6 FROM service s,
7      TABLE(s.service_chambre) sc,
8      TABLE(s.service_infirmier) si,
9      TABLE(DEREF(sc.column_value).chambre_patient) cp
10 GROUP BY s.code_service,
11      s.nom_service;
```

cod nom service	nombre infirmiers
nombre patients rapport inf/pat	
CAR Cardiologie	8
9 ,888888889	
CHG Chirurgie générale	13
20 ,65	
REA Réanimation et Traumatologie	7
10 ,7	

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

```

SQL> SELECT m.num_emp AS "numero medecin",
2      m.prenom_emp AS "prenom medecin",
3      m.nom_emp AS "nom medecin"
4 FROM medecin m,
5      TABLE(m.medecin_patient) mp
6 WHERE Deref(mp.column_value).hospitalisation IS NOT NULL
7 GROUP BY m.num_emp,
8      m.prenom_emp,
9      m.nom_emp
10 HAVING COUNT(DISTINCT Deref(Deref(mp.column_value).hospitalisation.hospitalisation_chambre).code_service) = (
11      SELECT COUNT(*) FROM service
12 );

```

numero medecin

prenom medecin

nom medecin

196

Nabila

TEBIBEL

89

Naima

BAHBOUH