



BASES DE DONNEES AVANCEES PROJET 01 SQL3-ORACLE

Realisé Par :

SEMMAR Hichem

AMEZIANE Abdelghani

Responsable du module :

Azzouz

Assistante TP :

Challal

Partie I : Modélisation orientée objet :

Dans la résolution de ce projet, Deux modèles objets ont été fait :

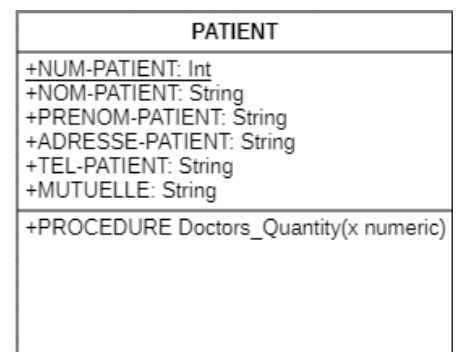
-Old project : la solution de ce model objet est orienté vers le model relationnel, peu d'associations existes et les requêtes sont presque identique à celle du model relationnel (Solution établi à travers les connaissances précédant seulement).

-Reworked project : ce model objet représente tous les connaissances qu'on a acquise dans ce module en plus d'expérience précédente (Beaucoup d'associations établi, Schéma complet et détaillé).

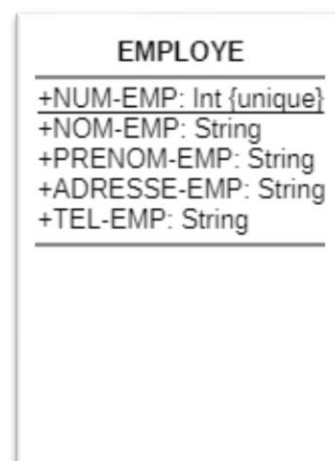
Il existe un **but** derrière la création des deux solutions mentionné dans la conclusion de ce rapport.

Definitions des Objects nécessaires (Old project):

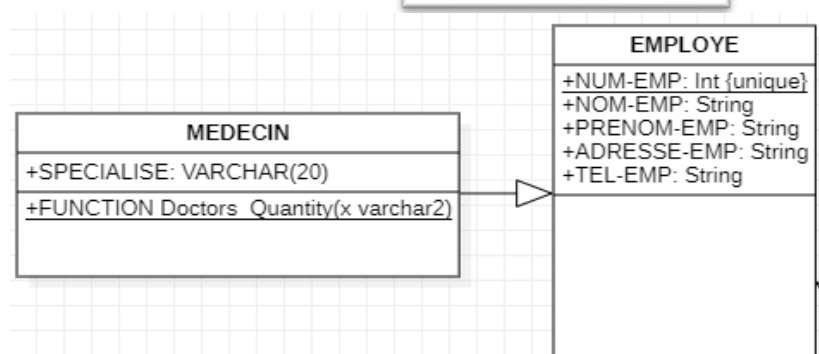
TPATIENT (NUM_PATIENT Number(7), NOM_PATIENT
varchar2(30), PRENOM_PATIENT varchar2(30),
ADRESSE_PATIENT varchar2(100), TEL_PATIENT varchar2(10),
MUTUELLE varchar2(10), STATIC PROCEDURE
Doctors_Quantity(x numeric))



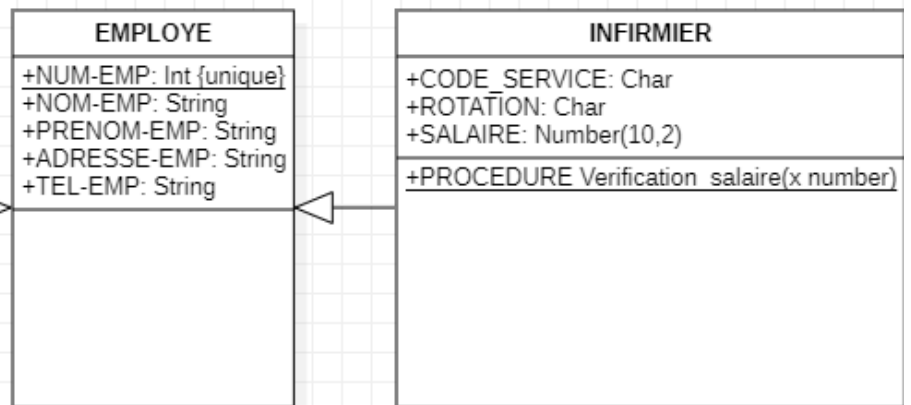
TEMPLOYE (NUM_EMP Number(7),NOM_EMP
varchar2(30),PRENOM_EMP varchar2(30),ADRESSE_EMP
varchar2(100),TEL_EMP varchar2(10)) Not final



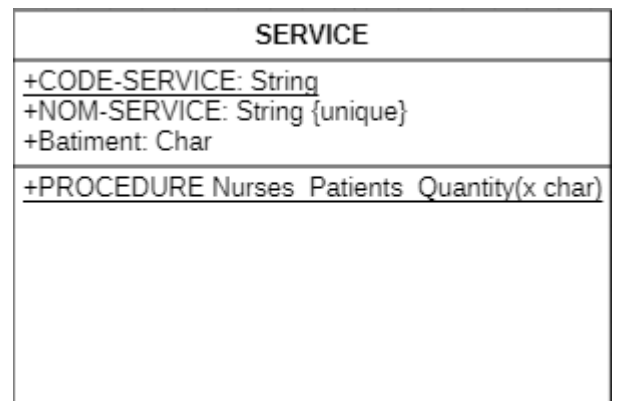
TMEDECIN under
temployme(SPECIALITE
varchar2(20),
Doctors_Quantity(x varchar2))



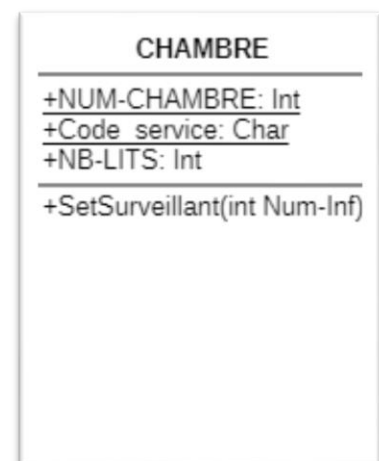
TINFIRMIER under
 temploze(CODE_SERVICE
 char(3), rotation char(4), salaire
 number(10,2), STATIC
 PROCEDURE
 Verification_salaire(x number))



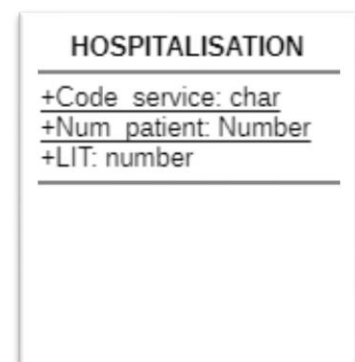
TSERVICE (CODE_SERVICE char(3), NOM_SERVICE
 varchar2(40), BATIMENT char, DIRECTEUR ref temploze,
 STATIC PROCEDURE Nurses_Patients_Quantity(x char))



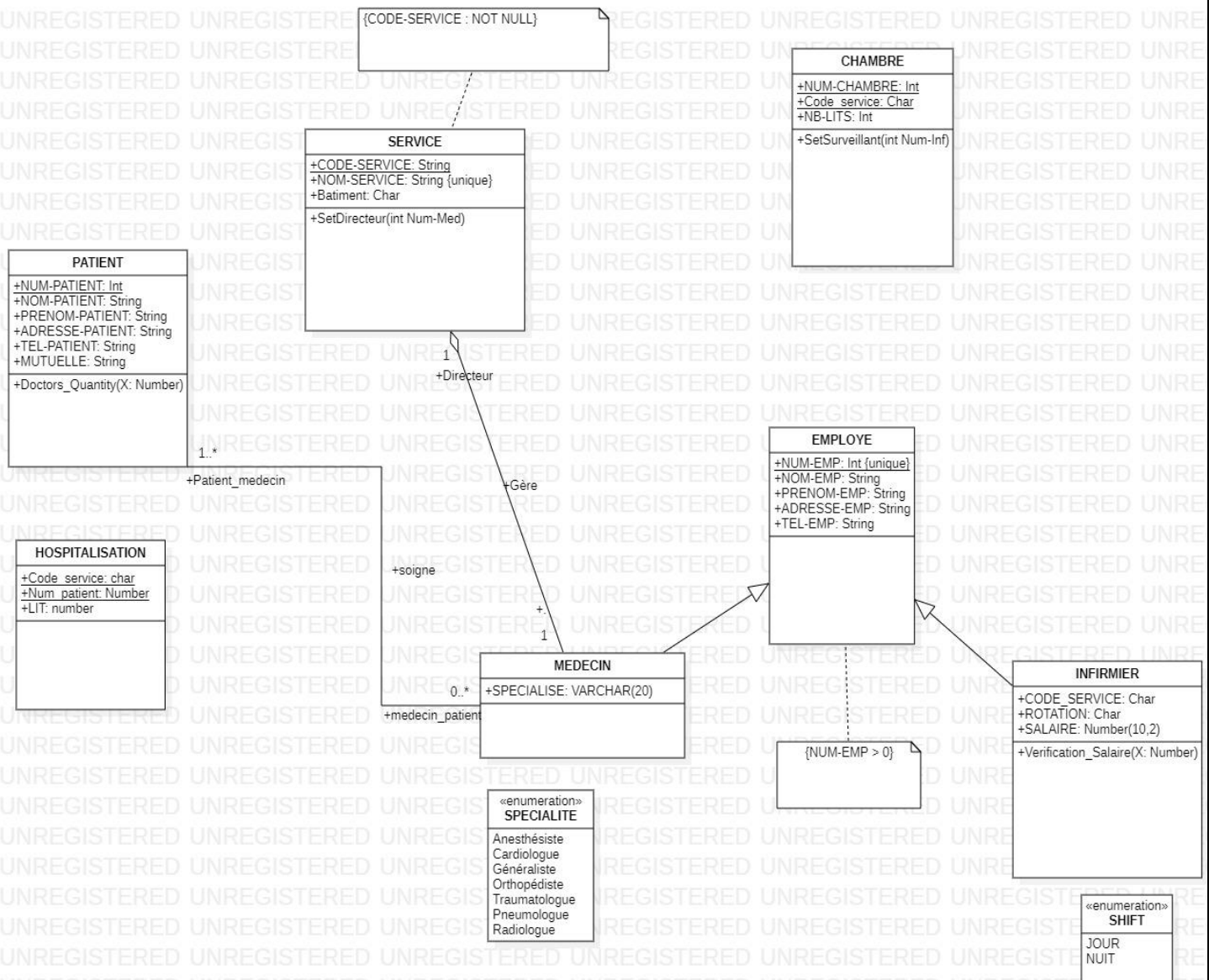
TCHAMBRE (CODE_SERVICE char(3), NUM_CHAMBRE Number(4),
 SURVEILLANT ref temploze, NB_LITS integer)



THOSPITALISATION (num_patient Number(7), CODE_SERVICE
 char(3), hos_chambre ref tchambre, lit integer)

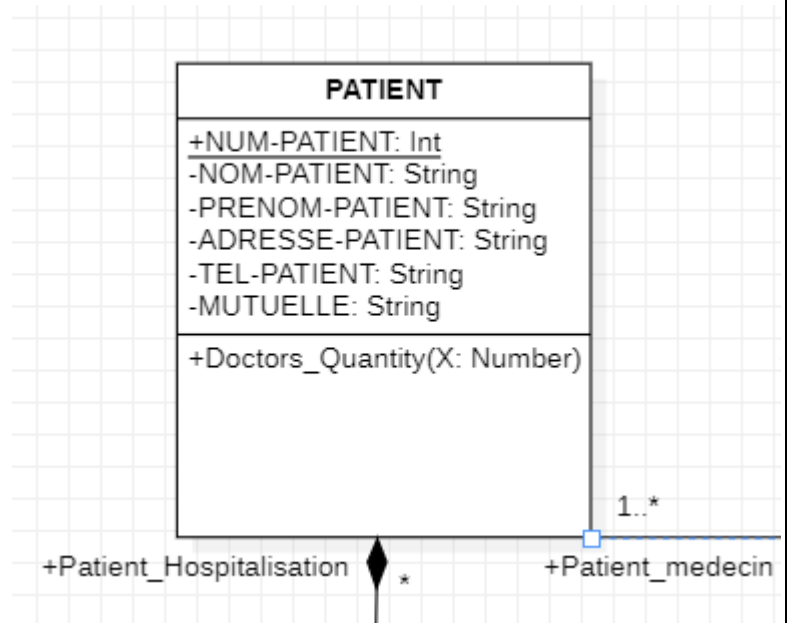


Le schema objet complet (Old project) :

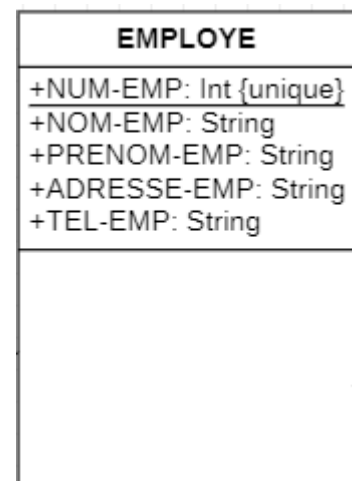


Definitions des Objects nécessaires (Reworked project):

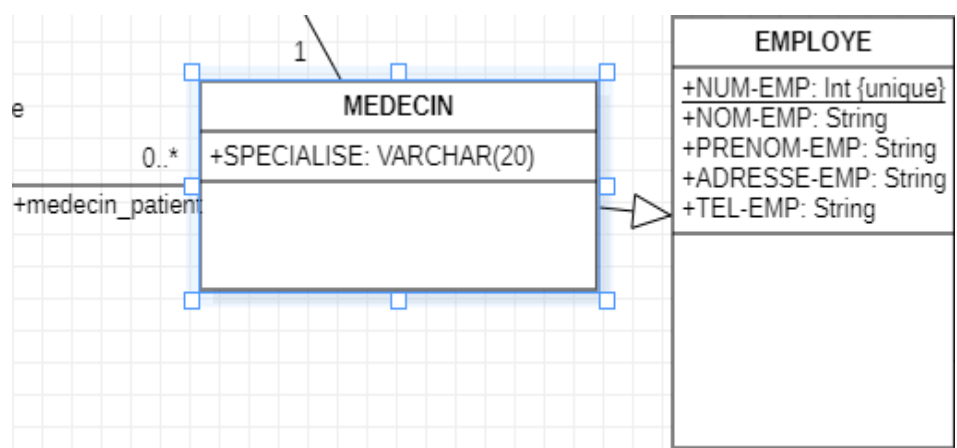
Create type tpatient as
 object(NUM_PATIENT Number(7),
 NOM_PATIENT varchar2(30),
 PRENOM_PATIENT varchar2(30),
 ADRESSE_PATIENT varchar2(100),
 TEL_PATIENT varchar2(10), MUTUELLE
 varchar2(10), patient_hospitalisation
 ref thospitalisation, STATIC PROCEDURE
 Doctors_Quantity(x numeric))



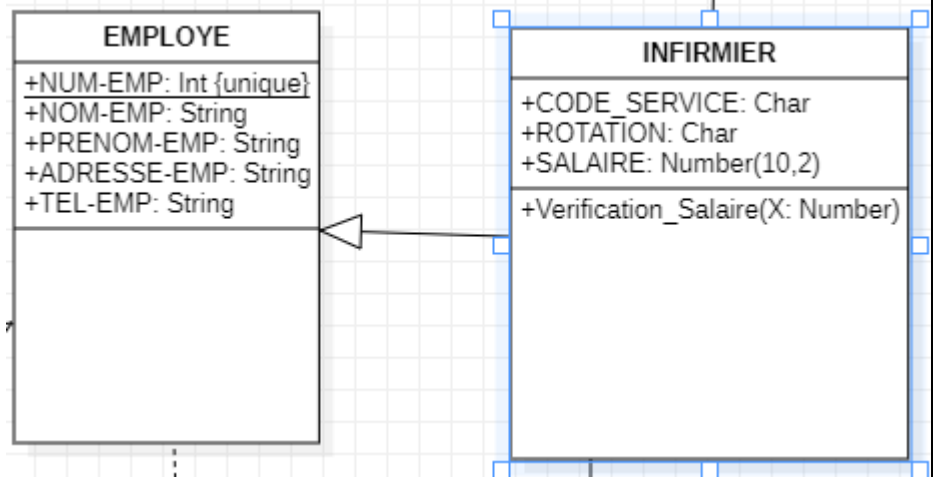
Create type temploye as object(NUM_EMP Number(7),
 NOM_EMP varchar2(30), PRENOM_EMP varchar2(30),
 ADRESSE_EMP varchar2(100), TEL_EMP varchar2(10)) Not final



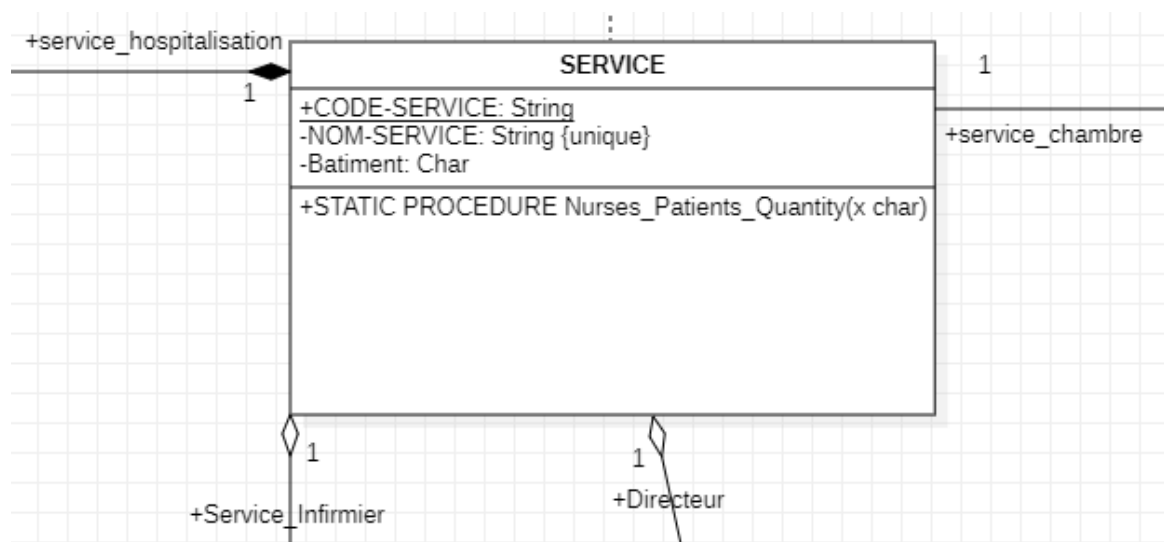
Create type tmedecin
 under
 temploye(SPECIALITE
 varchar2(20), STATIC
 PROCEDURE
 Doctors_Quantity(x
 numeric))



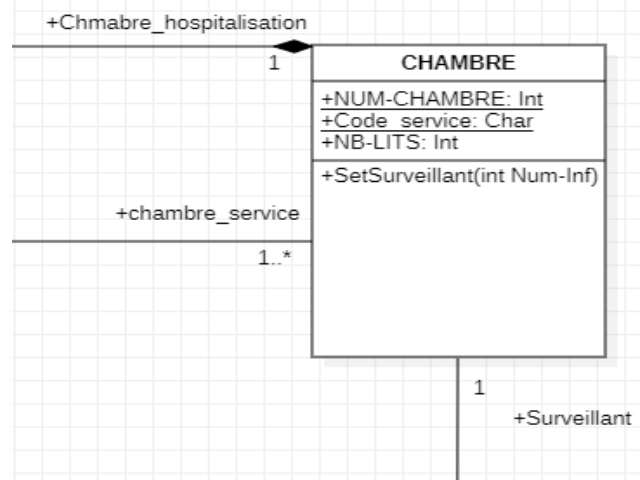
Create type tinfirmer
under
temploye(CODE_SERVICE
char(3), rotation
char(4), salaire
number(10,2),
infirmer_service ref
t_service,
infirmer_chambre ref
t_chambre)



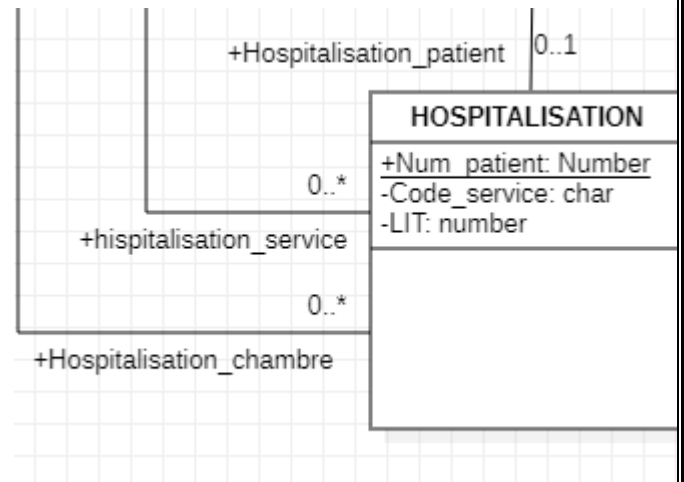
Create or replace type tservice as object(CODE_SERVICE char(3), NOM_SERVICE
varchar2(40), BATIMENT char, DIRECTEUR ref temploye, service_hospitalisation
t_set_hospitalisation, service_infirmer t_set_infirmer, service_chambre
t_set_chambre, STATIC PROCEDURE Nurses_Patients_Quantity(x char))



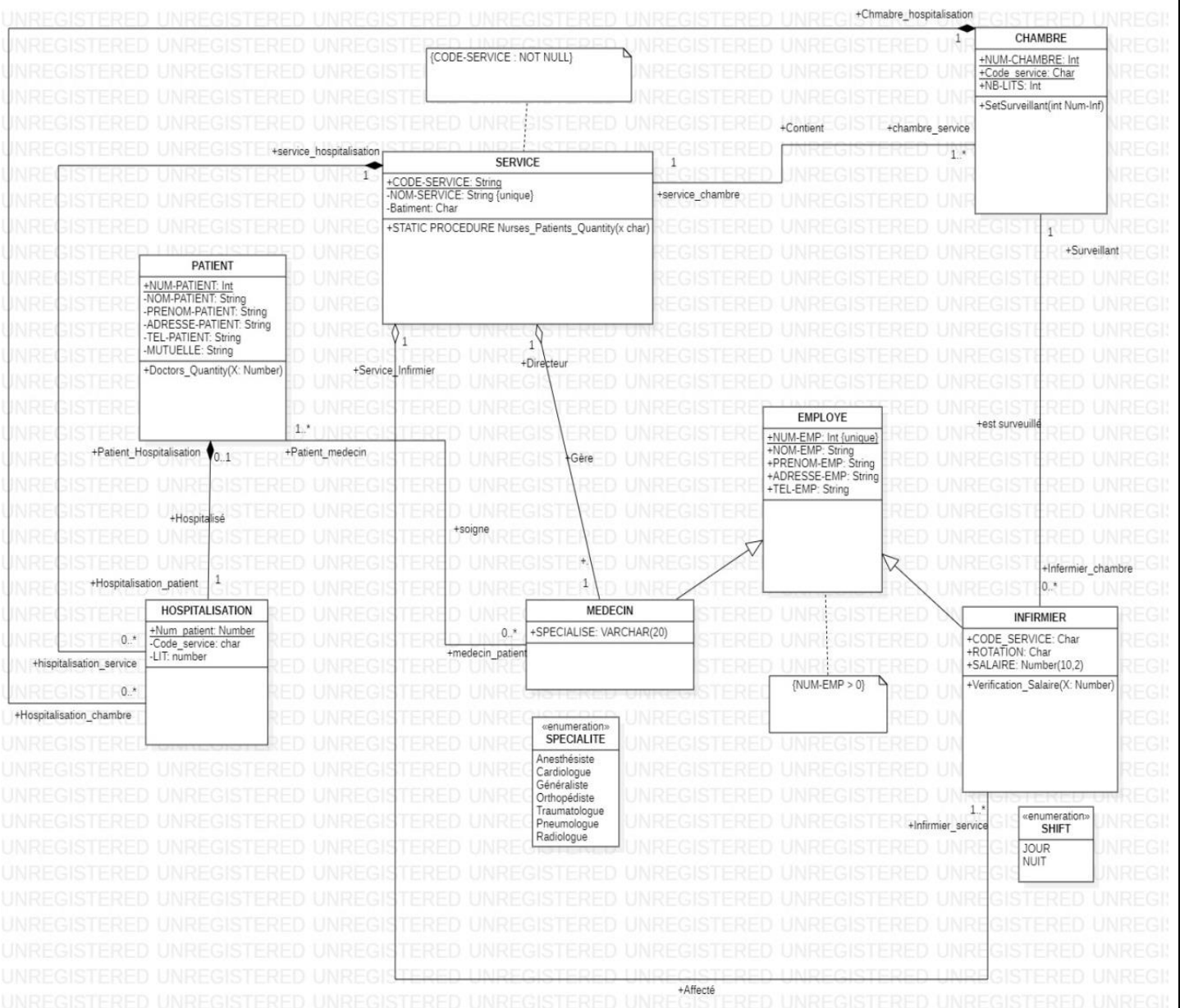
Create type `tchambre` as
 object(CODE_SERVICE char(3),
 NUM_CHAMBRE Number(4),
 SURVEILLANT ref temploye, NB_LITS
 integer, chambre_hospitalisation
 t_set_hospitalisation, chambre_service
 ref tservice)



Create or replace type `thospitalisation` as
 object(num_patient Number(7), CODE_SERVICE
 char(3), hos_chambre ref tchambre, lit integer,
 hospitalisation_patient ref tpatient,
 hospitalisation_service ref tservice)



Le schema objet complet (Reworked project) :



Partie II : Création des TableSpaces et Utilisateur :

Create tablespace SQL3_TBS datafile 'D:\SQL3_TBS.dat' size 100M AUTOEXTEND ON
ONLINE;

Create temporary tablespace SQL3_TempTBS TEMPFILE 'D:\SQL3_TempTBS.dat' size
100M AUTOEXTEND ON;

Create User SQL3 IDENTIFIED by Psw Default Tablespace SQL3_TBS Temporary
Tablespace SQL3_TempTBS;

Grant all privileges to SQL3;

Conn Sql3/Psw

Pour Insérer tous les données/Résoudre le projet, choisissez :

Pour lancer Old_Project :

@ "C:\Users\Tp_sql\Desktop\Project\Old_Project\Project.sql"

Pour lancer Reworked_Project :

@ "C:\Users\Tp_sql\Desktop\Project\Reworked_Project\Project_Reworked.sql"

Partie III : Langage de définition de données :

Les tables nécessaires à la base de données :

Create table patient of tpatient(primary key (NUM_PATIENT))

Create table medecin of tmedecin(primary key (NUM_EMP))

Create table infirmier of tinfirmier(primary key (NUM_EMP));

create table service of tservice(primary key (CODE_SERVICE), UNIQUE(nom_service))

create table chambre of tchambre(primary key(code_service, num_chambre), foreign
key(code_service) references service(code_service))

Create table HOSPITALISATION of thospitalisation(primary key(num_patient), foreign
key(num_patient) references patient(NUM_PATIENT), foreign key(code_service)
references service(code_service));

Définition des méthodes :

1-MEMBER FUNCTION Doctors_Quantity(x varchar2)

```
SQL> CREATE OR REPLACE TYPE BODY TMEDECIN AS
  2      MEMBER FUNCTION Doctors_Quantity(x varchar2) return numeric IS
  3      num number;
  4      BEGIN
  5          select count(*) into num
  6          from medecin
  7          where specialite = x ;
  8          return num;
  9      END;
10 END;
11 /
```

Type body created.

Elapsed: 00:00:00.00

SQL>

SQL> set SERVEROUTPUT ON

SQL> set timing on

SQL> select distinct m.doctors_quantity('Traumatologue') from medecin m;

M.DOCTORS_QUANTITY('TRAUMATOLOGUE')

5

2-STATIC PROCEDURE Nurses_Patients_Quantity(x char)

```
SQL> CREATE OR REPLACE TYPE BODY TSERVICE AS
  2     STATIC PROCEDURE Nurses_Patients_Quantity(x char) IS
  3         num number;
  4         num1 number;
  5     BEGIN
  6         select count(distinct Deref(surveillant).num_emp), count(distinct num_patient) into num, num1
  7         from chambre, hospitalisation
  8         where chambre.code_service = hospitalisation.code_service and chambre.code_service = x;
  9         dbms_output.put_line('service : ' || x || chr(10) || 'Nombre_infirmiers : ' || num || chr(10) || 'Nombre de Patients : ' || num1);
 10     END;
 11 END;
 12 /
```

Type body created.

Elapsed: 00:00:00.07

SQL>

SQL> set SERVEROUTPUT ON

SQL> exec tservice.Nurses_Patients_Quantity('CAR');

service : CAR

Nombre_infirmiers : 2

Nombre de Patients : 9

PL/SQL procedure successfully completed.

3-STATIC PROCEDURE Doctors_Quantity(x numeric)

```
SQL> CREATE OR REPLACE TYPE BODY TPATIENT AS
  2     STATIC PROCEDURE Doctors_Quantity(x numeric) IS
  3         num number;
  4     BEGIN
  5         select cardinality(patient_medecin) into num
  6         from patient
  7         where num_patient = x;
  8         dbms_output.put_line('Num_Patient : ' || x || chr(10) || 'Nombre_Docs : ' || num);
  9     END;
 10 END;
 11 /
```

Type body created.

Elapsed: 00:00:00.00

SQL>

SQL> set SERVEROUTPUT ON

SQL> exec tpatient.Doctors_Quantity(3);

Num_Patient : 3

Nombre_Docs : 3

PL/SQL procedure successfully completed.

Elapsed: 00:00:00.01

SQL>

4-STATIC PROCEDURE Verification_salaire(x number)

```
SQL> SET SERVEROUTPUT ON
SQL>
SQL> CREATE OR REPLACE TYPE BODY tinfirmer AS
  2     STATIC PROCEDURE Verification_salaire(x number) IS
  3     num number;
  4     BEGIN
  5         SELECT salaire into num
  6         FROM infirmier
  7         where num_emp = x;
  8
  9         if(num > 10000 and num < 30000) then
10             DBMS_OUTPUT.PUT_LINE(chr(10) || 'Verification positive');
11         else
12             DBMS_OUTPUT.PUT_LINE(chr(10) || 'Verification Negative');
13         end if;
14     END;
15 End;
16 /

Type body created.

Elapsed: 00:00:00.00
SQL>
SQL> exec tinfirmer.Verification_salaire(195);

Verification positive

PL/SQL procedure successfully completed.

Elapsed: 00:00:00.00
SQL>
SQL> exec tinfirmer.Verification_salaire(200);

Verification Negative

PL/SQL procedure successfully completed.
```

Les méthodes du schema Old_Project sont écrites dans le fichier Project.sql

Les méthodes du schema New_Project sont écrites dans le fichier Project_Reworked.sql

Les deux méthodes des fichiers donnent le même résultat, la différence est que Old_project utilise la jointure des tableaux dans la majorité des requêtes, par contre Project_reworked utilise les references ou bien des pointeurs vers les objets demandé dans la requête.

Partie VI : Langage de manipulation de données :

Old_Project : Toutes les données de la base de données sont localisé dans le dossier Tables.

Table chambre.sql : Les données de la table Chambre

Tablea hospitalisation.sql : Les données de la table hospitalisation

Table Infirmier.sql : Les données de la table infirmier

Table medecin.sql : Les données de la table medecin

Table patient.sql : Les données de la table Patient

Table service.sql : Les données de la table service

Table soigne.sql : sert à remplir Les tables de refference de Patient_medecin et Medecin_Patient des deux tables medecin et patients (Recordages des pointeurs)

Reworked_Project : Toutes les données de la base de données sont localisé dans le dossier TablesRe.

Table chambre.sql : Les données de la table Chambre

Tablea hospitalisation.sql : Les données de la table hospitalisation

Table Infirmier.sql : Les données de la table infirmier

Table medecin.sql : Les données de la table medecin

Table patient.sql : Les données de la table Patient

Table service.sql : Les données de la table service

Table soigne.sql : sert à remplir Les tables de référence de Patient_medecin et Medecin_Patient des deux tables medecin et patients(Recordages des pointeurs)

Chambre_fill.sql : sert à remplir la table de référence de Chambre_hospitalisation de tous les chambre par tous les hospitalisations existante.

infirmier_fill.sql : sert à remplir la référence du service et chambre de tous les infirmiers

service_fill.sql : Sert à remplir les table de référence de service_hospitalisation, service_infirmier, service_chambre de tous les services par tous les références des hospitalisations, infirmiers et chambres respectivement

Table Patient_hosfill.sql : sert à remplir la de référence de Patient_hospitalition de tous les patients par leur hospitalisation.

Durant le remplissage de ces tables, des fonctions et procédures ont été utilisé pour la réalisation de ce dernier.

Toutes ces fonctions et procédures sont localisées dans le fichier Fonctions.

fonctions Ref_XXX() : sert à retourner la référence d'un objet

EXAMPLE : FUNCTION ref_patient(x number) return REF TPATIENT :
cette fonction retourne la référence d'un patient.

```
SQL> select ref_patient(1) from patient where num_patient = 1;

REF_PATIENT(1)
-----
0000280209A51999B3BD0A4C9A91ED5B56C49C2D77CC47629D206D43758D4E08A5C9E4BEEC014000
950000
```

FUNCTION check_medecin_spec(x varchar2) : vérifier l'existence de la spécialité d'un médecin.

```
INSERT INTO medecin VALUES(tmecin(99,'BASSI','Fatima','Cité du 5 juillet bloc 130-Mostaganem','045217227',check_medecin_spec('test'), t_set_patient()))
ERROR at line 1:
ORA-20001: Medecin specialite does not exist or Not found
ORA-06512: at "SQL3.CHECK_MEDECIN_SPEC", line 8
```

xxx_fill() : sert a remplir les tables de référence d'un objet.

EXAMPLE : Procedure Chambre_fill(x number, y char, z number) :

sert a remplir la table des références chambre_hospitalisation de chaque table chambre par tous ses références de hospitalisation.

```
SQL> select code_service, service_hospitalisation from service;

COD
---
SERVICE_HOSPITALISATION
-----
CAR
T_SET_HOSPITALISATION(00002202086EC8DF4AB1764CE98BC7782C9E3EE1AAF7CE120B409846CE
BDD2C3919423F1B8, 00002202084574895CA8964BCF8097720C99A2A317F7CE120B409846CEBDD2
C3919423F1B8, 00002202080211175CF AFC461C998272DC5A472125F7CE120B409846CEBDD2C391
9423F1B8, 000022020894904E23E50244478B6ADEAAE1997B84F7CE120B409846CEBDD2C3919423
F1B8, 00002202082C290DA5E9854A0585AFB0AC6811ABFAF7CE120B409846CEBDD2C3919423F1B8
, 0000220208D0B6FF026AC74B95B7AA30838939957DF7CE120B409846CEBDD2C3919423F1B8, 00
002202084200B7E6DBEE4AD791AB93377A4EC940F7CE120B409846CEBDD2C3919423F1B8, 000022
020850FAF32619F04837A9D83E9273D5B92DF7CE120B409846CEBDD2C3919423F1B8, 0000220208
```

Partie V : Langage d'interrogation de données :

(Tous les requêtes sont disponible dans le fichier Project_reworked.sql | Project.sql)

Le but de la création des deux schémas s'implique dans cette partie, on exécute les requêtes et on remarque les résultats :

1 :

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

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

2 :

```
SQL> select Deref(h.hospitalisation_service).nom_service as Service, Deref(h.hos_chambre).num_chambre as chambre, lit, Deref(h.hospitalisation_patient).nom_patient as N
OM, Deref(h.hospitalisation_patient).prenom_patient as PRENOM, Deref(h.hospitalisation_patient).mutuelle as mutuelle
  2 from hospitalisation h
  3 where Deref(h.hospitalisation_service).batiment = 'B' and Deref(h.hospitalisation_patient).mutuelle like 'M%';

SERVICE          CHAMBRE          LIT
-----
NOM              PRENOM              MUTUELLE
-----
Cardiologie       101                1
SERIR             Mustapha            MNAM
Cardiologie       101                3
TAHMI             Lamia               MNH
Cardiologie       105                2
TITOUCHE          Mohamed             MNAM
```

3 :

```
SQL> with rws as(
2  select p.num_patient, t.*
3  from patient p, table (p.patient_medecin) t
4  )
5  select num_patient, count(column_value) as Nombre_medecins, count(distinct Deref(column_value).SPECIALITE) as Specialities_total
6  from rws
7  having count(column_value) > 2
8  group by num_patient;
```

NUM_PATIENT	NOMBRE_MEDECINS	SPECIALITIES_TOTAL
123	3	3
76	4	3
6	4	3
13	3	3
147	4	4
108	4	4
21	3	3
68	3	3
184	3	3
119	3	3
117	4	3

NUM_PATIENT	NOMBRE_MEDECINS	SPECIALITIES_TOTAL
61	3	3
23	3	2
41	3	3
52	3	3
3	3	3
172	4	3
105	3	3
145	4	4
192	3	3
182	3	3
36	3	3

NUM_PATIENT	NOMBRE_MEDECINS	SPECIALITIES_TOTAL
159	4	4
103	3	3

24 rows selected.

4 :

```
SQL> select s.code_service, avg(deref(inf.column_value).salaire)
2  from service s, table(s.service_infirmier) inf
3  group by s.code_service;
```

COD	AVG(DEREF(INF.COLUMN_VALUE).SALAIRE)
CAR	15068.5413
CHG	14771.4523
REA	13788.1529

5 :

```
SQL> select distinct Deref(med.column_value).NOM_EMP as nom, Deref(med.column_value).PRENOM_EMP as prenom
2  from patient, table(patient_medecin) med
3  where patient_hospitalisation is not NULL;
```

NOM	PRENOM
RAHALI	Ahcene
MOHAMMEDI	Mustapha
AMARA	Dahbia
BASSI	Fatima
FEKAR	Abdelaziz
AAKOUB	Fatiha
ABBACI	Abdelmadjid
BAHBOUH	Naima
BEKKAT	Hadia
TEBIBEL	Nabila
ABADA	Mohamed

NOM	PRENOM
ABDELAZIZ	Ahmed
ABDELMOUMEN	Nassima
BENOUADAH	Mohammed
BAALI	Souad
BELGHALI	Mohammed
BENDALI	Hacine
AZOUG	Dalila
BACHA	Nadia
BEHADI	Youcef
CHAOUI	Farid
BOUCHEMLA	Elias

NOM	PRENOM
ADDAD	Fadila
AROUEL	Leila
TERKI	Amina
BOULARAS	Fatima

26 rows selected.

Elapsed: 00:00:00.04

Partie VI : conclusion

On remarque que les requêtes de Reworked_Project sont 3X plus rapide que celles du Old_Project, la raison et que old_project génère les requêtes par la jointure des tables ce qui est très couteau, par contre Reworked_Project génère tous les requêtes en utilisant une seul table à chaque fois Grace au accède des objets par référencement.