PROJET BDA

Juste Prescription des Médicaments en SQL2 et SQL3

Léo BIREBENT
Maxime DOURLEN

Etudiant en M1 Informatique Promotion 2022 - 2023

TABLE DES MATIERES

PRESENTATION

L'objectif du projet était la création d'une base de données en sql2 et sql3 qui permet de gérer les prescriptions de médicaments, données à des patients. Les patients suivent des traitements qui sont prescrits par des médecins suite à des observations faites lors de consultations. Les observations permettent de déduire une ou plusieurs maladies. Pour chaque maladie, un ou plusieurs médicaments peuvent être prescrits. Un traitement a une durée et est constitué de médicaments et peut inclure des recommandations. Un médicament possède des caractéristiques, comme les maladies qu'il soigne, les contre-indications, les différentes substances actives qui le compose, les effets indésirables connus disponibles dans des notices retrouvables par exemple à cette adresse : http://base-donnees-publique.medicaments.gouv.fr/achageDoc.php?specid=60044492&typedoc=R.
Chaque substance active possède une ou plusieurs classes chimiques et/ou une ou plusieurs classes pharmacologiques. Les substances actives ainsi que les 2 classes citées précédemment peuvent générer des effets indésirables. Quant au médecin, il peut travailler pour un ou plusieurs laboratoires pharmaceutiques et peut également développer un ou plusieurs médicaments.

BASE DE DONNEES SQL2

1. Schéma relationnel et Schéma E/A

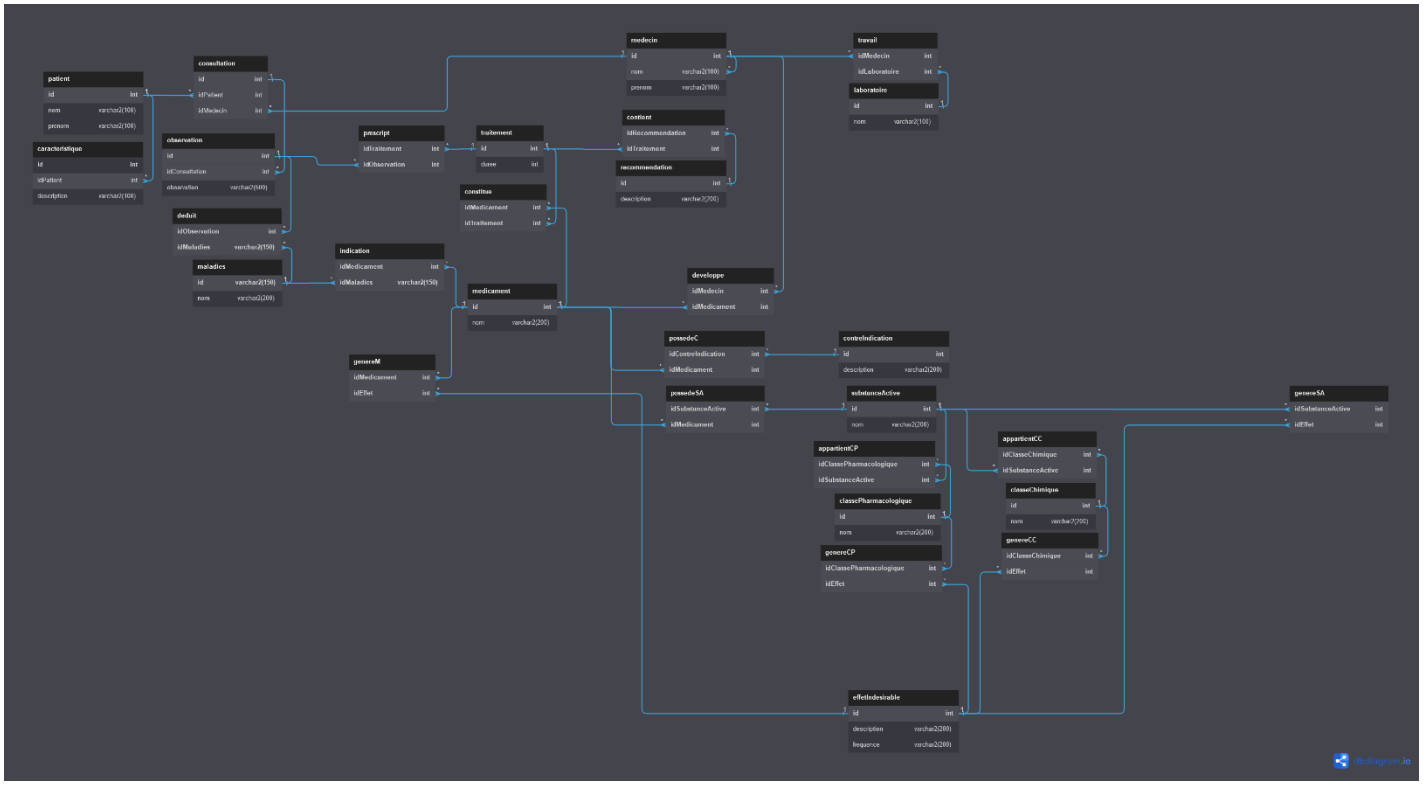


Schéma Relationnel

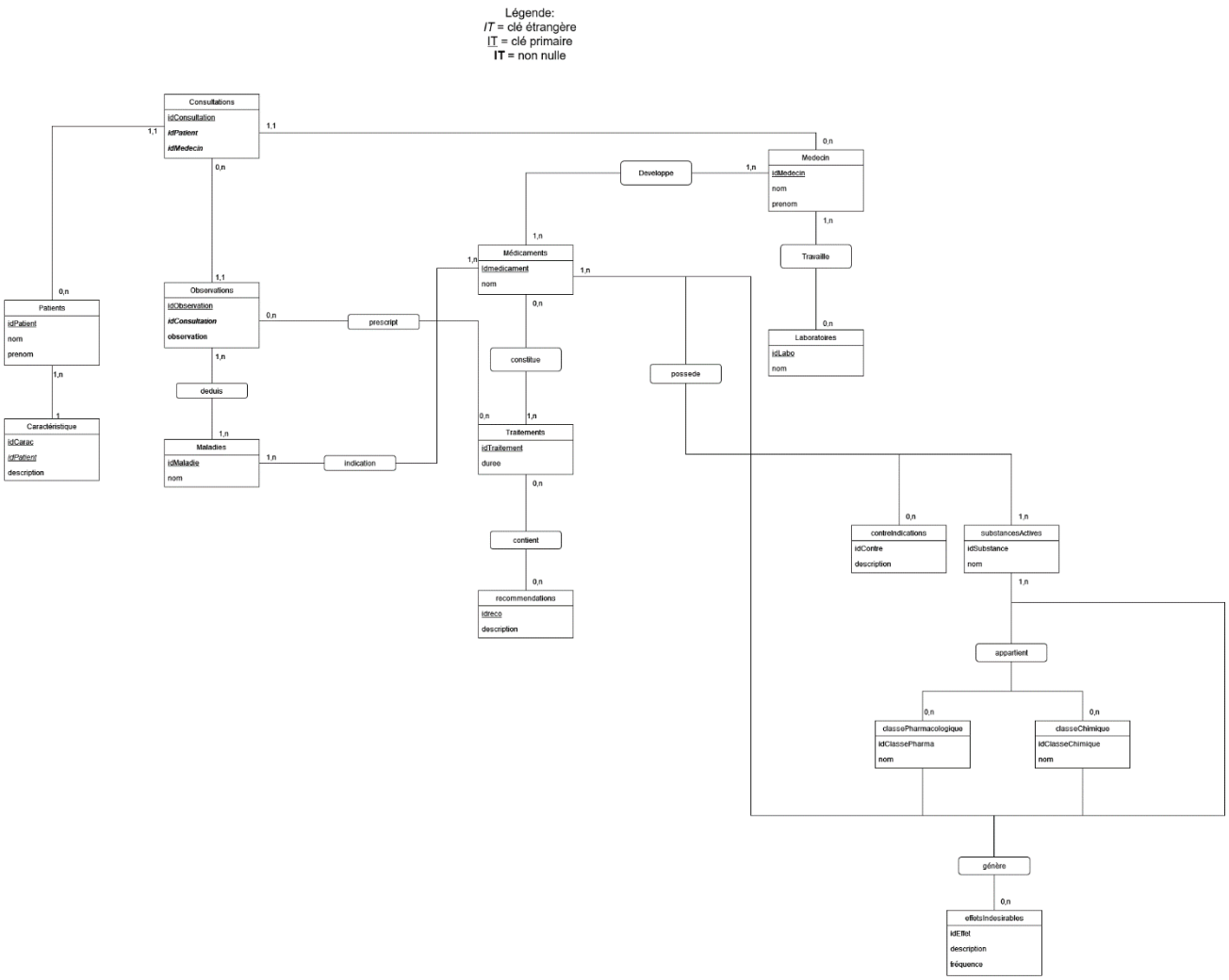


Schéma E/A

Nous avons décidé de construire et lier les différentes tables en SQL2 de la manière suivante : Un patient possède une ou plusieurs caractéristiques qui lui sont propres. Un patient peut avoir plusieurs consultations avec le même médecin ou non. Suite à cette consultation, le médecin fait des observations qui lui permettront d'en déduire une ou plusieurs maladies et lui prescrira un ou plusieurs médicaments à partir des observations faites. Un traitement contient une durée (ici en jours), des recommandations, ainsi qu'un certain nombre de médicaments. Un médicament contient une indication qui permet de savoir la ou les maladies qu'il soigne, les médecins qui ont fabriqué ce médicament et ses potentiels effets indésirables. Il possède aussi une ou des contre-indications, ainsi que ou une plusieurs substances actives qui elle-même contient une ou plusieurs classes pharmacologique et/ou classes chimiques (création de 2 tables pour faire le lien entre substance active et les 2 classes). Les substances actives, ainsi que les différentes classes génèrent elles aussi, comme les médicaments, des effets indésirables (création pour chacune des 3 tables, une table pour faire le lien avec les effets indésirables). Chaque médecin travaille pour un ou des laboratoires (création de la table "travail").

2. Fonctions et Procédures

Pendant la réalisation de ce projet, plusieurs requêtes/procédures/fonctions nous ont été demandé :
Une première requête qui, à partir d'une observation, nous affiche toutes les maladies déduites. La seconde consistait à créer une procédure qui, en fonction de la maladie sélectionnée, nous affiche les différents médicaments disponibles qui peuvent traiter cette maladie. La troisième consiste à créer un trigger qui vérifie qu'un médecin ne prescrit pas un traitement qui contient un médicament qu'il a développé. A noter que le trigger créé ne vérifie que lorsqu'un médecin prescrit un traitement à un patient, et non lorsque celui modifie un traitement. Un second trigger aurait pu être créé mais on aurait dû vérifier toutes les prescriptions faites avec ce traitement, ce qui rend la requête lourde lors d'une insertion pour une vérification pas très utile. La quatrième se compose d'une procédure qui, à partir d'un médicament choisi, nous affiche les différents effets indésirables des substances actives, classes pharmacologiques et classes chimiques qui le compose. Quant à la dernière demande, elle consiste à créer une requête qui liste les différents médicaments que chaque médecin a prescrit.

Première requête :

```
select m.nom
from maladies m
inner join deduit d on d.idmaladies = m.id
where d.idobservation = 1 /*<-MODIFIER ICI L'ID Du SYMPTONE*/
order by m.nom desc;
```

Exemple de résultat attendu :

```
select m.nom
  2  from maladies m
  3  inner join deduit d on d.idmaladies = m.id
  4  where d.idobservation = 3 /*<-MODIFIER ICI L'ID Du SYMPTONE*/
  5  order by m.id desc;

NOM
--------------------------------------------------------------------------------
-------
infections ?? arbovirus
```

Deuxième requête :

```
SET SERVEROUTPUT ON;
DECLARE
  v_id  varchar2(150) := 'C01.252.354.900.675'/*<-MODIFIER ICI L'ID DE LA MALADIE*/;
  v_count number := 0;
begin
  loop
```

```
        select count(*) into v_count from indication i where i.idMaladies = v_id;
        if v_count > 0 then
        For med_rec
        in (select distinct me.nom
        from medicament me
        where me.id in (select i.idMedicament from indication i where i.idMaladies = v_id))
        loop
        dbms_output.put_line(med_rec.nom);
        end loop;
        EXIT;
        end if;
        if LENGTH(v_id)<2 then
        EXIT;
        end if;
        select replace(v_id, regexp_substr(v_id,'\.?[0-9]*$',1,1)) into v_id from dual;
  end loop;
end;
```

Exemple de résultat attendu :

```
DECLARE
  v_id  varchar2(150) := 'C01.252.200.100.122'/*<-MODIFIER ICI L'ID DE LA
 MALADIE*/;
  3    v_count number := 0;
  4  begin
  5    loop
    select count(*) into v_count from indication i where i.idMaladies = v
_id;
  7      if v_count > 0 then
  8          For med_rec
  9          in (select distinct me.nom
 10              from medicament me
           where me.id in (select i.idMedicament from indication i where
 i.idMaladies = v_id))
 12          loop
         dbms_output.put_line(med_rec.nom);
 14          end loop;
 15          EXIT;
 16      end if;
    if LENGTH(v_id)<2 then
 18        EXIT;
 19      end if;
    select replace(v_id, regexp_substr(v_id,'\.?[0-9]*$',1,1)) into v_id
from dual;
 21    end loop;
 22  end;
 23
 24  /
calment
soin
```

Troisième requête :

```
create or replace trigger verifPrescription
before insert on prescript
for each row
declare
        id_med int;
```

```
        nb_medicament number := 0;
begin
        select idMedecin into id_med
        from consultation
        where id = (select idConsultation
                from observation
                where id = :new.idObservation);
        select count(*) into nb_medicament
        from developpe
        where idMedecin = id_med
        and idMedicament in (select idMedicament
                from constitue
                where idTraitement = :new.idTraitement);
        if (nb_medicament > 0) then
        RAISE_APPLICATION_ERROR(-20001, 'Le medecin numero ' || id_med || ' ne peut
pas prescrire le medicament pour lequel il a participe a l laboration!');
        end if;
exception
        WHEN NO_DATA_FOUND THEN NULL;
end;
/
```

Exemple de résultat attendu :

```
Trigger created.

SQL> insert into prescript values(1,3);
insert into prescript values(1,3)
            *
ERROR at line 1:
ORA-20001: Le medecin numero 1 ne peut pas prescrire le medicament pour l
equel
il a participe a l laboration!
```

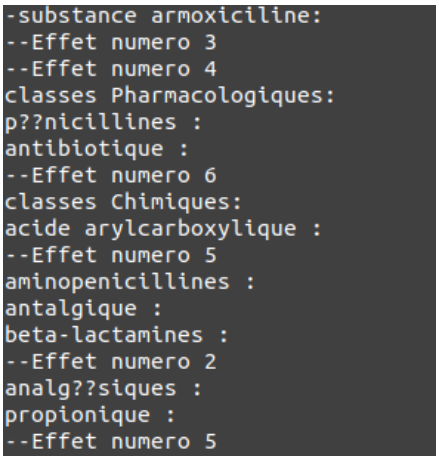
Quatrième requête :

```
DECLARE
  id_medicament int := 3/*<-MODIFIER ICI L'ID DU MEDICAMENT*/;
begin
  For subs
  in (select distinct sa.nom, sa.id
      from possedeSA psa, SubstanceActive sa
      where psa.idMedicament = id_medicament and sa.id = psa.idSubstanceActive)
  loop
    dbms_output.put_line('-substance ' || subs.nom || ':');
    For effet0
    in (select ei.id, ei.description, ei.frequence
        from effetIndesirable ei, genereSA gSA
        where gSA.idEffet = ei.id and gSA.idSubstanceActive = subs.id)
```

```
loop
        dbms_output.put_line('--Effet numero '||effet0.id || ' ');
end loop;
dbms_output.put_line('classes Pharmacologiques:');
For classesP
in (select distinct cp.id, cp.nom
    from appartientCP acp, classePharmacologique cp
    where acp.idSubstanceActive = subs.id and acp.idclassePharmacologique = cp.id)
loop
  dbms_output.put_line(classesP.nom || ' :');
  For effet1
  in (select ei.id
        from effetIndesirable ei
        where ei.id in
        (select gCP.idEffet
        from genereCP gCP
        where gCP.idClassePharmacologique in
                (select CP.id
                from classePharmacologique CP
                START WITH CP.id = classesP.id /* idclassepharma */
                connect by prior CP.idparent = CP.id)))
  loop
    dbms_output.put_line('--Effet numero '||effet1.id || ' ');
  end loop;
end loop;
dbms_output.put_line(' classes Chimiques:');
For classesC
in (select distinct cc.id, cc.nom
    from appartientCC acc
    NATURAL JOIN classeChimique cc
    where acc.idSubstanceActive = subs.id)
loop
  dbms_output.put_line(classesC.nom || ' :');
  For effet2
  in (select ei.id
        from effetIndesirable ei
        where ei.id in
        (select gCC.idEffet
        from genereCC gCC
        where gCC.idClasseChimique in
                (select CC.id
                from classeChimique CC
                START WITH CC.id = classesC.id /* idclassepharma */
                connect by prior CC.idparent = CC.id)))
  loop
    dbms_output.put_line('--Effet numero '||effet2.id || ' ');
  end loop;
```

```
    end loop;
  end loop;
end;
```

Exemple de résultat attendu :



Cinquième requête :

```
select ide "id medecin", mednom "nom medecin", LISTAGG(names, ', ') WITHIN GROUP
(ORDER BY names) "Medicament"
from (select distinct me.id ide, m.nom names, me.nom mednom
from medicament m
inner join constitue c on c.idMedicament = m.id
    inner join traitement t on t.id = c.idTraitement
    inner join prescript p on p.idTraitement = t.id
    inner join observation o on o.id = p.idObservation
    inner join consultation c2 on c2.id = o.idConsultation
    inner join medecin me on me.id = c2.idMedecin)
GROUP BY ide, mednom
ORDER BY ide;
```

Exemple de résultat attendu :

```
id medecin
----------
nom medecin
--------------------------------------------------------------------------
-------
Medicament
--------------------------------------------------------------------------
-------
         1
Usclat
antibiotique, calment, soin

         2
Filleux
ventoline

id medecin
----------
nom medecin
--------------------------------------------------------------------------
-------
Medicament
--------------------------------------------------------------------------
-------
```

BASE DE DONNEES SQL3

1. Schéma relationnel et Schéma E/A

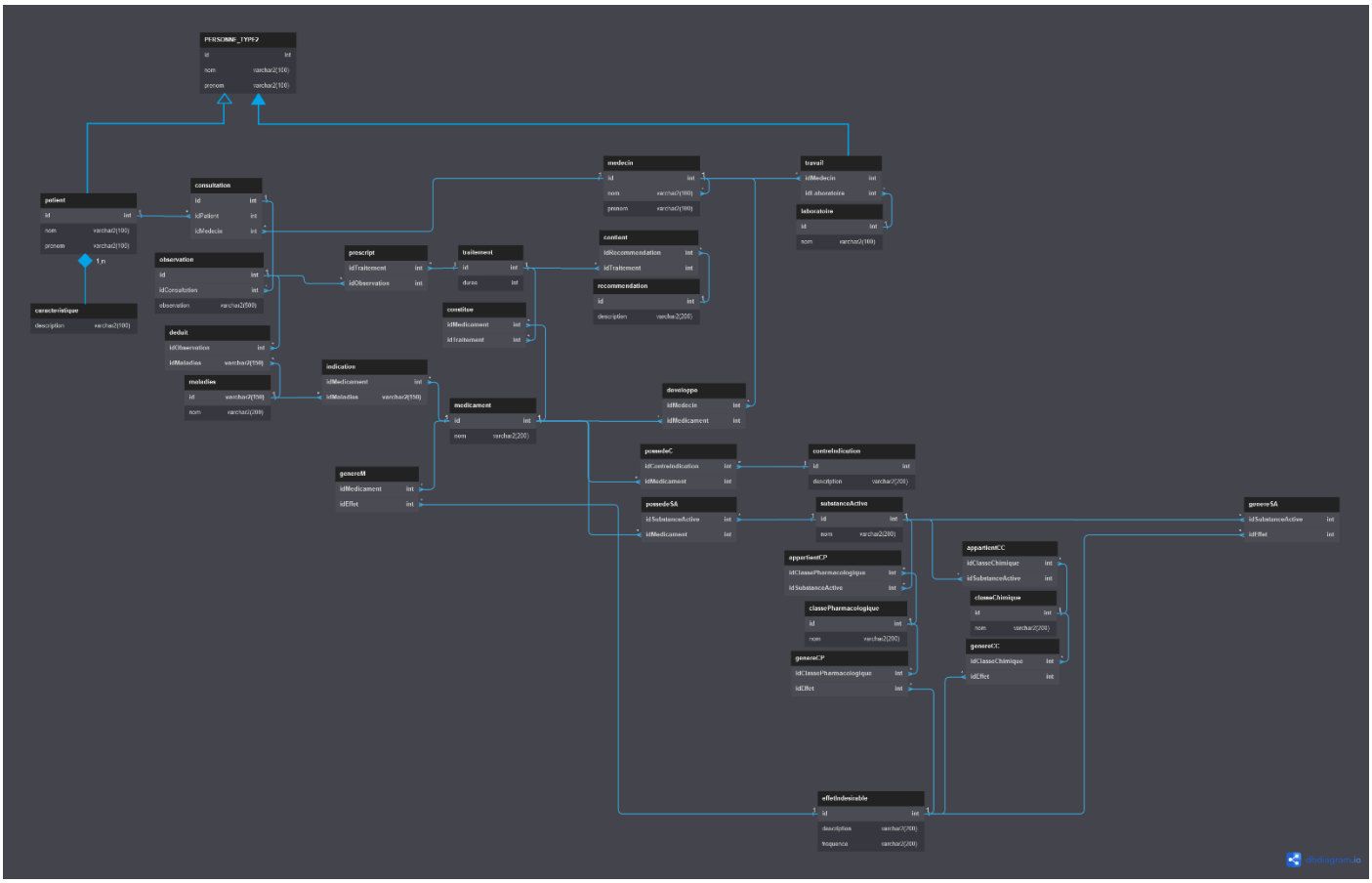


Schéma Relationnel

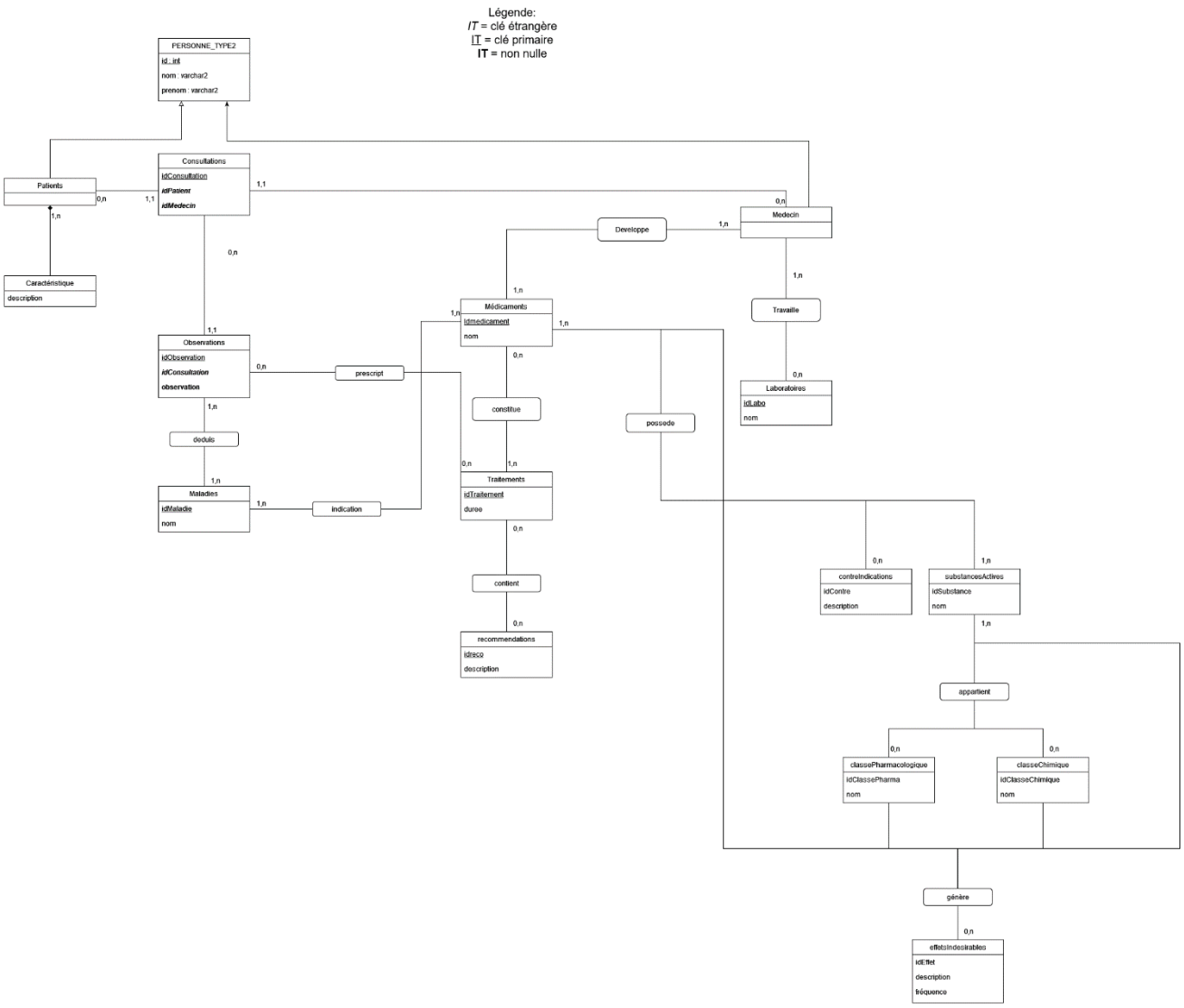


Schéma E/A

Pour l'implémentation de la base de données en SQL3, quelques modifications ont été apportées par rapport au model SQL2 qui sont les suivants : Création d'un type objet PERSONNE_TYPE2 et CARACTERISTIQUE_TYPE. Création d'un type CARACTERISTIQUES_TYPE (table en colonne) qui est un tableau de CARACTERISTIQUE_TYPE et qui va nous permettre de créer une table imbriquée. Création d'un type PATIENT_TYPE qui hérite de PERSONNE_TYPE2 et dans lequel on ajoute un attribut de type CARACTERISTIQUES_TYPE. Création de la table « patient » de type PATIENT_TYPE et de la table medecin de type PERSONNE_TYPE2 avec l'ajout de la contrainte de la clé primaire pour l'id.

2. Fonctions et Procédures

Etant donnée que peu de changement ont été apporté entre l'architecture de la base de données SQL2 et SQL3, aucune modification n'a été apportée pour les procédures, car les modifications n'affectent pas celle-ci.

CONCLUSION

Dans le cadre de nos études, le projet nous a permis de mettre en application ce que nous avons appris tout le long de se semestre. Cependant plusieurs difficultés sont apparues : La compréhension du sujet était parfois ambigüe. Par exemple, la procédure 1 où on devait déduire les maladies à partir des symptômes, nous en avons déduit qu'à partir d'une observation, nous devons lister toutes les maladies déduites. Le jeu de données n'étant pas fournie, nous avons dû notamment chercher sur internet les médicaments, classes Chimiques, Pharmacologiques et leurs effets indésirables. La version d'Oracle (11G) nous a parfois posé problèmes, particulièrement sur l'affichage des sorites de procédures.