Projet de Base de données

Juste prescription de médicaments

Projet réalisé par

- BELKHOUS Redha Nabil
- ZEBOUCHI Mohammed

Sommaire

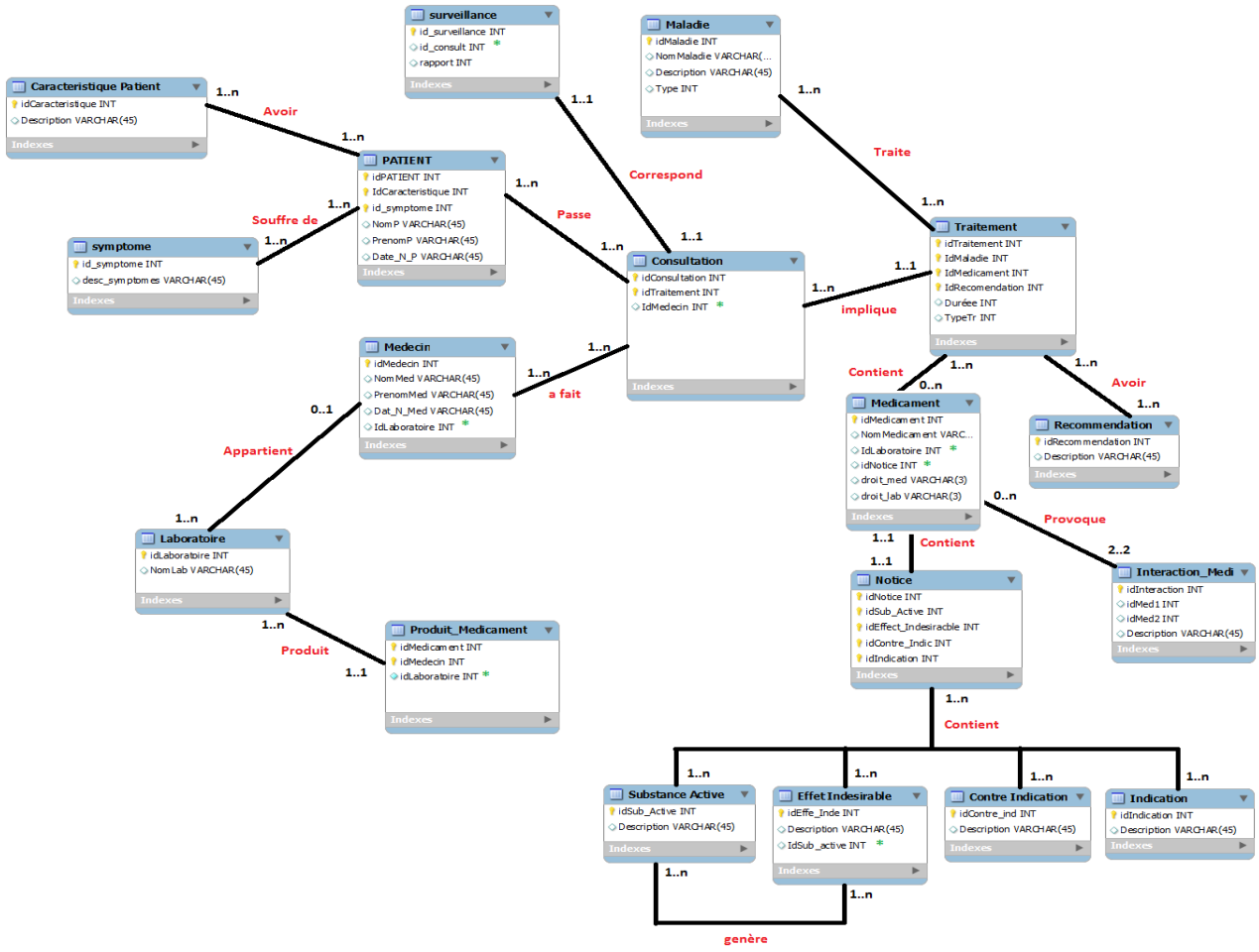
1 Description du projet

Description du projet

Le but de l'application est de gérer des patients qui suivent des traitements prescrits par des médecins suite à des observations faites lors de consultations.
Un traitement, un médecin, des observations et d'autres caractéristiques ont une structure bien spécifique, des règles de gestions et des contraintes.

2 Présentation du modèle relationnel

Présentation du modèle relationnel



On trouve dans le modèle relationnel les tables suivantes :

- *Caractéristique Patient* : qui contient un identifiant et une description.
  Comme son nom l'indique, cette table représente les caractéristiques d'un patient qui sont « **jeune**, **vieux**, **fragile** et **sportif** ».
- *Patient :* qui est caractérisé par un identifiant patient, un identifiant caractéristique, un identifiant symptôme, un nom, un prénom et une date de naissance.

La clef primaire est constituée de l'identifiant patient, l'identifiant caractéristique, de l'identifiant symptôme car **un patient peut avoir plusieurs symptômes et plusieurs caractéristiques**.

- *Symptôme : qui* est définie par un identifiant et une description.
  Cette table reflète les éventuels symptômes.
- *Maladie :* qui comporte un identifiant maladie, le nom de la maladie, une description, et un **type de maladie** pour caractériser la **hiérarchie** si elle existe.
- *Médecin :* qui contient un identifiant, un nom, un prénom, une date de naissance et un identifiant de laboratoire qui représente le laboratoire auquel le médecin appartient (ou non).
- *Consultation :* qui regroupe un identifiant de consultation, un identifiant de traitement, et un identifiant de médecin.
  La clef primaire est « **identifiant consultation** et **identifiant traitement** ».
- *Traitement :* qui se traduit par un identifiant de traitement, un identifiant de maladie, un identifiant de médicament, un identifiant de recommandation, une durée et un type de traitement (traitement normal, ou chronique).
  La clef primaire est « **identifiant traitement**, **identifiant maladie**, **identifiant médicament** et **identifiant recommandation**».
- *Laboratoire :* se compose d'un identifiant et d'un nom de laboratoire.
- *Médicament :* qui contient un identifiant de médicament, un nom, un identifiant de laboratoire, un identifiant de notice, un droit de prescription par un médecin et un droit de prescription par un laboratoire.
- *Recommandation :* qui se résume par un identifiant et une description.
- *Produit :* qui contient un identifiant de médicament, un identifiant de médecin et un identifiant de laboratoire.
- La clef primaire est « **identifiant médicament**, **identifiant médecin** et **identifiant laboratoire** ».
- *Interaction Médicament :* qui est définie par un identifiant propre, deux identifiants de médicament et une description.
- *Notice :* qui se résume par un identifiant de notice, un identifiant de substance active, d'effet(s) indésirable(s), d'indication(s) et de contre indication(s).
- La clef primaire est « **identifiant notice**, **identifiant substance active**, **identifiant effet(s) indésirable(s)**, **identifiant indication(s)**, et **identifiant contre indication(s)**».
- *Substance active :* qui contient un identifiant, une description, et un **type** pour caractériser la **hiérarchie**.
- *Effet indésirable :* qui regroupe un identifiant, une description et un identifiant de substance active.
- *Contre indication :* qui est décrite par un identifiant et une description.
- *Indication :* qui contient un identifiant et une description.
- *Surveillance :* qui comporte un identifiant surveillance, un identifiant de consultation et un rapport qui est un calcule spécifique.

3 Description des règles de gestion

Description des règles de gestion

1) Un patient **a une** ou **plusieurs** caractéristiques.
2) Un patient **passe une** ou **plusieurs** consultations
3) Un patient **souffre** d'**un** ou **plusieurs** symptômes.
4) Un médecin **réalise une** ou **plusieurs** consultations.
5) Un médecin **appartient à aucun** ou **un seul** laboratoire.
6) Un laboratoire **contient un** ou **plusieurs** médecins.
7) Un laboratoire **développe un** ou **plusieurs** produits.
8) Un produit est **développé** par **un seul laboratoire** et par **un** ou **plusieurs** médecins.
9) Une consultation **donne lieu** à **un** ou **plusieurs** traitements.
10) Un traitement **correspond à une et une seule** consultation.
11) Un traitement **traite un** ou **plusieurs** maladies.
12) Une maladie **peut être traitée** par **un** ou **plusieurs** traitements.
13) Un traitement **contient un** ou **plusieurs** médicaments.
14) Un médicament **peut être dans plusieurs** traitements **ou** dans **aucuns.**
15) Un traitement **contient une** ou **plusieurs** recommandations.
16) Une recommandation **correspond** à **un** ou **plusieurs** traitements.
17) Un médicament **contient une et une seul** notice.
18) Une notice **concerne à un et un** seul médicament.
19) Une notice **contient un** ou **plusieurs** effets indésirables, substances actives, contre indications et indications.
20) **Un** effet indésirable, substance active, contre indication, indication **peuvent être indiqués dans une** ou **plusieurs** notices.
21) Une interaction médicamenteuse **arrive** si on a **deux médicaments** qui sont donnés dans le **même traitement**, qui pose problème.
22) Une maladie a un type et des sous types.
23) Un traitement aura un type, soit un traitement court ou un traitement long.
24) Une substance active aura un type et des sous types.
25) Une substance active **génère un** ou **plusieurs** effets indésirables.
26) Une consultation **donne lieu à une** et **une seule** surveillance.

4 Présentation des requêtes, jeux d'exemples et résultats

1. Sauvegarder le choix de traitement (médicament et/ou recommandation) et les maladies diagnostiquées par le médecin).

```
CREATE OR REPLACE PROCEDURE prescription ( id_pat  NUMBER,  id_trait
NUMBER , id_med NUMBER )  is
            id_cons NUMBER(10);
            nbPatient number(10);
            nbMedecin number(10);
BEGIN
    select MAX ( id_consult )  into id_cons
    from consultation;

    id_cons:= id_cons+1;

    select count(*) into nbMedecin
    from medecin m
    where m.id_medecin = id_med ;

            if ( nbMedecin= 0 )  then
                    raise_application_error ( -20900,'Ce medecin n existe pas !' ) ;
                    else
                            select count(*) into nbPatient
                            from patient p
                            where  p.id_patient  =  id_pat ;

                            if ( nbPatient = 0 ) then
                                    raise_application_error( -20901,'Ce patient n existe pas
                                    veulliez le créer' ) ;
                                    else
                                            insert into consultation values( id_cons , id_trait ,
                                            id_pat , id_med );
                            end if;
            end if;
    END ;
/

execute prescription( 1 , 1 , 1 );
```

Explications

- On récupère le dernier id de consultation et on l'incrémente.
- On vérifie que le médecin existe.
- On insère la nouvelle consultation (patient, traitement et médecin).

```
SQL> CREATE OR REPLACE PROCEDURE prescription (id_pat NUMBER, id_trait NUMBER, id_med NUMBER) is
  2    id_cons NUMBER(10);
  3    nbPatient number(10);
  4    nbMedecin number(10);
  5   BEGIN
  6    select MAX(id_consult) into id_cons
  7    from consultation;
  8
  9    id_cons:= id_cons+1;
 10
 11    select count(*) into nbMedecin
 12    from medecin m
 13    where m.id_medecin = id_med;
 14
 15    if(nbMedecin=0) then
 16     raise_application_error(-20900,'Ce medecin n existe pas !');
 17     else
 18      select count(*) into nbPatient
 19      from patient p
 20      where p.id_patient = id_pat;
 21
 22      if(nbPatient=0) then
 23       raise_application_error(-20901,'Ce patient n existe pas veulliez le créer');
 24       else
 25        insert into consultation values(id_cons,id_trait,id_pat,id_med );
 26      end if;
 27    end if;
 28   END ;
 29  /

Procédure créée.
```

À partir de cet exemple, on peut observer qu'avant d'exécuter la procédure, on n'a que 8 consultations. Après exécution de la procédure, on constate l'enregistrement d'une nouvelle consultation.

```
SQL> select id_consult
  2  from consultation;

ID_CONSULT
----------
         1
         2
         3
         4
         5
         6
         7
         8

8 ligne(s) sélectionnée(s).

SQL> execute prescription(1,4,4);

Procédure PL/SQL terminée avec succès.

SQL> select id_consult
  2  from consultation;

ID_CONSULT
----------
         1
         2
         3
         4
         5
         6
         7
         8
         9

9 ligne(s) sélectionnée(s).
```

2. Proposer une liste de médicaments à partir de la maladie diagnostiquée, même si un lien direct maladie-médicament n'existe pas.

```
create or replace type arraymedicamanet as table of varchar(300) ;
/

set serveroutput on ;

CREATE OR REPLACE function liste_Medicament (id_malad NUMBER) return
arraymedicamanet as
        arraymed arraymedicamanet := arraymedicamanet();
         i integer :=0;
         nom_med varchar(300);
        BEGIN

                for nom_medic1 in (select m.nom_medic
                from traitement t , medicament m
                where t.id_medicament = m.id_medicament and t.id_maladie = id_malad) loop
                        i := i+1;
                        arraymed.extend;
                        arraymed(i) := nom_medic1.nom_medic ;
                end loop;

                return arraymed;
        END ;
/
```

```
SQL> create or replace type arraymedicamanet as table of varchar(300) ;
  2  /

Type créé.

SQL>
SQL> set serveroutput on ;
SQL>
SQL> CREATE OR REPLACE function liste_Medicament (id_malad NUMBER) return arraymedicamanet as
  2   arraymed arraymedicamanet := arraymedicamanet();
  3    i integer :=0;
  4    nom_med varchar(300);
  5   BEGIN
  6
  7    for nom_medic1 in (select m.nom_medic
  8    from traitement t , medicament m
  9    where t.id_medicament = m.id_medicament and t.id_maladie = id_malad) loop
 10     i := i+1;
 11     arraymed.extend;
 12     arraymed(i) := nom_medic1.nom_medic ;
 13    end loop;
 14
 15    return arraymed;
 16   END ;
 17  /

Fonction créée.
```

Example:

```
select liste_Medicament(111)
from dual;
```

On peut voir qu'on a récupéré les trios médicaments du traitement contre la maladie 111.

```
select liste_Medicament(12232)
from dual;
```

On peut voir qu'on a récupéré les trios médicaments du traitement contre la maladie 12232.

```
SQL> select liste_Medicament(111)
  2  from dual;

LISTE_MEDICAMENT(111)
--------------------------------------------------------------------------------
ARRAYMEDICAMANET('AMOXICILLINE / ACIDE CLAVULANIQUE MYLAN 500 mg', 'OMEPRAZOLE A
BBOTT 10 mg, gélule gastro-résistante', 'CLARITHROMYCINE ABBOTT 250 mg, comprimé
 pelliculé')


SQL>
SQL> select liste_Medicament(12232)
  2  from dual;

LISTE_MEDICAMENT(12232)
--------------------------------------------------------------------------------
ARRAYMEDICAMANET('BARACLUDE 0,05 mg/ml')
```

Explication

- On récupère la liste des médicaments d'une maladie dans un tableau qu'on retourne à la fin de la fonction grâce a une jointure entre la table traitement et maladie.

3. Déterminer pour un médicament la liste des effets indésirables connus et probables

```
create or replace type arrayeffet as table of varchar(300) ;
/

CREATE OR REPLACE function liste_Effet (effet NUMBER) return arrayeffet as
        arrayeffet_sec arrayeffet  := arrayeffet();
         i integer :=0;
         nom_med varchar(300);
        BEGIN

                for nom_eff in (select distinct e.description
                                from effect_indesirable e , notice n , medicament m
                                where m.id_not = n.id_notice and e.id_effet_indes =
                                        n.id_effect_indesir and m.id_medicament = effet)
                 loop
                        i := i+1;
                        arrayeffet_sec.extend;
                        arrayeffet_sec(i) := nom_eff.description ;
                end loop;
                return arrayeffet_sec;
        END ;
/
```

```
SQL> create or replace type arrayeffet as table of varchar(300) ;
  2  /

Type créé.

SQL>
SQL> CREATE OR REPLACE function liste_Effet (effet NUMBER) return arrayeffet as
  2   arrayeffet_sec arrayeffet  := arrayeffet();
  3    i integer :=0;
  4    nom_med varchar(300);
  5   BEGIN
  6
  7    for nom_eff in (select distinct e.description
  8         from effect_indesirable e , notice n , medicament m
  9         where m.id_not = n.id_notice and e.id_effet_indes = n.id_effect_indesir and m.id_medicam
ent = effet) loop
 10     i := i+1;
 11     arrayeffet_sec.extend;
 12     arrayeffet_sec(i) := nom_eff.description ;
 13    end loop;
 14
 15    return arrayeffet_sec;
 16   END ;
 17  /

Fonction créée.
```

Exemple:

```
select liste_Effet(1)
from dual;
```

```
SQL> select liste_Effet(1)
  2  from dual;

LISTE_EFFET(1)
--------------------------------------------------------------------------------
ARRAYEFFET('nausées, vomissements, Allergies')
```

Explication

- Récupérer à travers la requête les effets indésirables d'un médicament, les mettre dans un tableau et le retourner.

4. Déterminer s'il y a un ensemble de médicaments qui ne sont prescrits que par des médecins qui ont travaillé à leur développement.

```
select  m.nom_medic as Prescription_Medecin_Createur
from  medicament m
where  m.droit_med  =  'oui' ;
```

```
SQL> select m.nom_medic as Prescription_Medecin_Createur
  2  from medicament m
  3  where m.droit_med = 'oui';

PRESCRIPTION_MEDECIN_CREATEUR
--------------------------------------------------
AMOXICILLINE / ACIDE CLAVULANIQUE MYLAN 500 mg
OMEPRAZOLE ABBOTT 10 mg, gélule gastro-résistante
ALDACTONE 50 mg, comprimé sécable
SOVALDI 400 mg, comprimé pelliculé

     .
```

Explication

- Sélectionner les médicaments qui ne peuvent être prescrits que par leurs médecins fabricants, et cela grâce a l'attribut qui permet de savoir si un médicament est prescriptible que par les médecins qui l'on fabriqué (« droit_med = 'oui' » ou « droit_med = 'non' »).

5. Déterminer s'il y a des médicaments qui ne sont prescrits que par des médecins ayant travaillé dans les laboratoires les fabricants.

```
Select m.nom_medic as Prescription_Medecin_Lab
from medicament m
where m.droit_lab = 'oui' ;
```

```
SQL> select m.nom_medic as Prescription_Medecin_Lab
  2  from medicament m
  3  where m.droit_lab = 'oui';

PRESCRIPTION_MEDECIN_LAB
--------------------------------------------------
AMOXICILLINE / ACIDE CLAVULANIQUE MYLAN 500 mg
ALDACTONE 50 mg, comprimé sécable
CORTANCYL 20 mg, comprimé sécable
HAVRIX 1440
SOVALDI 400 mg, comprimé pelliculé
```

Explication

- Sélectionner les médicaments qui ne peuvent être prescrits que par les médecins qui travailles dans le laboratoire ou ils ont été prescrits, et cela grâce à l'attribut qui permet de savoir si un médicament est prescriptible que par les médecins qui appartiennent au laboratoire qui l'a fabriqué (« droit_lab = 'oui' » ou « droit_lab = 'non' »)

6. Identifier en fonction des symptômes et des caractéristiques les maladies possibles.

En fonction des symptômes

```
Select distincte m.nom_maladie as Maladie
from maladie m , consultation c , traitement t , patient p
where m.id_maladie = t.id_maladie and t.id_trait = c.id_trait and
      p.id_patient = c.id_patient and p.id_symptome = &symptome ;
```

Exemple avec le symptôme N°15

```
SQL> select distinct m.nom_maladie as Maladie
  2  from maladie m , consultation c , traitement t , patient p
  3  where m.id_maladie = t.id_maladie and t.id_trait = c.id_trait and p.id_patient = c.id_patient a
nd p.id_symptome = &symptome;
Entrez une valeur pour symptome : 15
ancien   3 : where m.id_maladie = t.id_maladie and t.id_trait = c.id_trait and p.id_patient = c.id_pa
nouveau   3 : where m.id_maladie = t.id_maladie and t.id_trait = c.id_trait and p.id_patient = c.id_

MALADIE
--------------------------------------------------
Hepatite A
Hepatite alcoolique
Hepatite B
```

En fonction des caractéristiques

```
select distincte m.nom_maladie as Maladie
from maladie m , consultation c , traitement t , patient p
where m.id_maladie = t.id_maladie and t.id_trait = c.id_trait and p.id_patient = c.id_patient
and p.id_caracteristique = &Caracteristique ;
```

Exemple avec la caractéristique N°3

```
SQL> select distinct m.nom_maladie as Maladie
  2  from maladie m , consultation c , traitement t , patient p
  3  where m.id_maladie = t.id_maladie and t.id_trait = c.id_trait and p.id_patient = c.id_patient a
nd p.id_caracteristique = &Caracteristique;
Entrez une valeur pour caracteristique : 3
ancien   3 : where m.id_maladie = t.id_maladie and t.id_trait = c.id_trait and p.id_patient = c.id_pa
nouveau   3 : where m.id_maladie = t.id_maladie and t.id_trait = c.id_trait and p.id_patient = c.id_

MALADIE
--------------------------------------------------
Hepatite alcoolique
Hepatite B
Gastrite
```

Explication

- Afficher à partir de caractéristiques ou de symptômes les maladies possibles grâce à une jointure entre les tables : maladie, consultation, traitement et patient, puis on récupère ce qui a été donné par l'utilisateur (« &Caracteristique » ou « &symptome »).

7. Pouvoir analyser l'ensemble des prescriptions des médecins et donner un rapport.

```
CREATE OR REPLACE PROCEDURE rapport (  consut NUMBER  ) is
        nbm NUMBER(10);  nbl number(10);  nbt number(10);
        id number(10);  res number(10,2);
    BEGIN
        select count( t.id_medicament ) into nbm
        from consultation c , traitement t , produit p
        where  c.id_consult  =  consut and  c.id_trait = t.id_trait and
              t.id_medicament = p.id_medicament and  p.id_medecin = c.id_medecin ;

        select count( t.id_medicament ) into nbl
        from consultation c , traitement t , produit p, medecin m
        where c.id_consult = consut and c.id_trait = t.id_trait and
              t.id_medicament =  p.id_medicament and m.id_medecin !=
              c.id_medecin and p.id_lab = m.id_laboratoire  ;

        select count(*) into nbt
        from consultation ;

        select count(*) into id
        from surveillance ;
        id:=id+1 ;
        if nbm+nbl = 0 then
              insert into surveillance values(id ,consut ,0 ) ;
              else
                    res := nbt/(nbm+nbl);
                    insert into surveillance values( id , consut , res );
        end if;
    END ; /
```

```
SQL> CREATE OR REPLACE PROCEDURE rapport (consut NUMBER) is
  2    nbm NUMBER(10);
  3    nbl number(10);
  4    nbt number(10);
  5    id number(10);
  6    res number(10,2);
  7   BEGIN
  8    select count(t.id_medicament) into nbm
  9    from consultation c , traitement t , produit p
 10    where c.id_consult = consut and  c.id_trait = t.id_trait and t.id_medicament = p.id_medicamen
t and
 11      p.id_medecin = c.id_medecin ;
 12
 13    select count(t.id_medicament) into nbl
 14    from consultation c , traitement t , produit p, medecin m
 15    where c.id_consult = consut and c.id_trait = t.id_trait and t.id_medicament = p.id_medicament
 and
 16      m.id_medecin != c.id_medecin and
 17      p.id_lab = m.id_laboratoire ;
 18
 19    select count(*) into nbt
 20    from consultation;
 21
 22    select count(*) into id
 23    from surveillance;
 24
 25    id:=id+1;
 26
 27    if nbm+nbl = 0 then
 28     insert into surveillance values(id ,consut ,0 );
 29     else
 30      res:= nbt/(nbm+nbl);
 31      insert into surveillance values(id ,consut ,res );
 32    end if;
 33   END ;
 34  /

Procédure créée.
```

Exemple :

Select * from surveillance **;**

```
SQL> select * from surveillance ;

ID_SURVEILLANCE ID_CONSULT    RAPPORT
--------------- ---------- ----------
              1          7          4
```

execute rapport**(**7**);**

```
SQL> execute rapport(7);

Procédure PL/SQL terminée avec succès.

SQL> select * from surveillance ;

ID_SURVEILLANCE ID_CONSULT    RAPPORT
--------------- ---------- ----------
              1          7          4
              2          7          3
```

On peut voir le résultat avant et après exécution de la fonction.

Explications

- Compter le nombre de médicaments fabriqués par le médecin (N1).
- Compter le nombre de médicaments fabriqués par le laboratoire (N2).
- Compter le nombre total de Consultation (N).
- Compter le rapport N / N1+N2

8. Indiquer à un médecin prescrivant si le traitement envisagé risque d'interagir avec un traitement 'en cours' et proposer le cas échéant un autre traitement.

```
CREATE OR REPLACE PROCEDURE Verfier_traitement (id_pat number , traitement2
number) is
        BEGIN
                for tr in (select distinct m.id_medicament ,m.nom_medic , t.id_trait
                from traitement t , patient p , consultation c , medicament m
                where p.id_patient = c.id_patient and t.id_trait = c.id_trait and
                        m.id_medicament = t.id_medicament and p.id_patient = id_pat)
                loop
                        for t in (select  m.id_medicament , i.id_medic1 , i.id_medic2
                                from traitement t , medicament m , intercation_medicament i
                                        where  t.id_trait = traitement2 and
                                        m.id_medicament = t.id_medicament and
                                (m.id_medicament = i.id_medic1 or m.id_medicament =
                                i.id_medic2))
                                loop
                                        if (t.id_medic2 = tr.id_medicament) or (t.id_medic1 =
                                        tr.id_medicament) then

                                        DBMS_OUTPUT.PUT_LINE('Problème avec le
                                        traitement '|| tr.id_trait ||' et le traitement '||traitement2);

                                        DBMS_OUTPUT.PUT_LINE('Interaction entre le
                                        medicament '|| tr.id_medicament ||' et le medicament
                                        '||t.id_medic2);
                                        END IF;
                                end loop;
                end loop;
        END;
/
```

```
SQL> CREATE OR REPLACE PROCEDURE Verfier_traitement (id_pat number , traitement2 number) is
  2
  3   begin
  4
  5    for tr in (select distinct m.id_medicament ,m.nom_medic , t.id_trait
  6    from traitement t , patient p , consultation c , medicament m
  7    where p.id_patient = c.id_patient and t.id_trait = c.id_trait and m.id_medicament = t.id_medi
cament and p.id_patient = id_pat)
  8    loop
  9      for t in (select  m.id_medicament , i.id_medic1 , i.id_medic2
 10       from traitement t , medicament m , intercation_medicament i
 11       where  t.id_trait = traitement2 and m.id_medicament = t.id_medicament and (m.id_medicament
 = i.id_medic1 or m.id_medicament = i.id_medic2))
 12       loop
 13        if (t.id_medic2 = tr.id_medicament) or (t.id_medic1 = tr.id_medicament) then
 14
 15         DBMS_OUTPUT.PUT_LINE('Problème avec le traitement '|| tr.id_trait ||' et le traitement '
||traitement2);
 16         DBMS_OUTPUT.PUT_LINE('Interaction entre le medicament '|| tr.id_medicament ||' et le med
icament '||t.id_medic2);
 17
 18        END IF;
 19
 20       end loop;
 21    end loop;
 22
 23   end;
 24  /

Procédure créée.
```

Exemple:

```
execute Verfier_traitement(1 , 7);
```

```
SQL> execute Verfier_traitement(1 , 7);
Problème avec le traitement 1 et le traitement 7
Interaction entre le medicament 3 et le medicament 9

Procédure PL/SQL terminée avec succès.
```

Explications

- Sélectionner les médicaments de chaque traitement et vérifier dans la table interaction s'il n'existe pas d'interaction médicamenteuse.
- Afficher un message selon le résultat.

9. Vérifiera si ces effets indésirables sont connus ou pas, sinon l'ajouter.

```
CREATE OR REPLACE PROCEDURE Ajout_Effect_Indesi ( effet varchar ) is
        id_eff number(10) ;
        id_f number(10) ;

    BEGIN
        select MAX( id_effet_indes ) into id_eff
        from effect_indesirable;

        id_eff := id_eff+1 ;

        select id_effet_indes into id_f
        from effect_indesirable e
        where e.description like CONCAT ('%',CONCAT(effet,'%')) ;

        IF SQL%ROWCOUNT !=0 then
            raise_application_error( -20910,'Effet Indesirable existant!' ) ;
        END IF ;

        EXCEPTION
            WHEN NO_DATA_FOUND THEN insert into effect_indesirable
            values ( id_eff,effet ) ;
    END ;
/

execute Ajout_Effect_Indesi('Lourdeure');
```

Explications

- Récupérer le dernier identifiant de la table effet indésirable.
- Vérifier que l'effet indésirable donné n'existe pas dans l'un des effets déjà enregistrés qu'il soit seul ou avec d'autres.
- Si aucun résultat n'est retourné, l'effet indésirable sera ajouté.

Exemple :

On peut voir qu'il n'existe pas d'effet indésirable « Perturbation », et qu'après création et éxécution de la procédure, on peut voir que le nouvel effet indésirable est ajouté.

```
SQL> select *
  2  from effect_indesirable
  3  where description = 'Perturbation';

aucune ligne sélectionnée

SQL>
SQL> CREATE OR REPLACE PROCEDURE Ajout_Effect_Indesi (effet varchar) is
  2    id_eff number(10);
  3    id_f number(10);
  4
  5   BEGIN
  6
  7    select MAX(id_effet_indes) into id_eff
  8    from effect_indesirable;
  9
 10    id_eff:= id_eff+1;
 11
 12    select id_effet_indes into id_f
 13    from effect_indesirable e
 14    where e.description like CONCAT('%',CONCAT(effet,'%'));
 15
 16    IF SQL%ROWCOUNT !=0 then
 17     raise_application_error(-20910,'Effet Indesirable existant!');
 18
 19    END IF;
 20
 21    EXCEPTION
 22     WHEN NO_DATA_FOUND THEN insert into effect_indesirable values(id_eff,effet);
 23
 24   END ;
 25  /

Procédure créée.

SQL>
SQL> execute Ajout_Effect_Indesi('Perturbation');

Procédure PL/SQL terminée avec succès.

SQL> select *
  2  from effect_indesirable
  3  where description = 'Perturbation';

ID_EFFET_INDES
--------------
DESCRIPTION
--------------------------------------------------------------------------------
            10
Perturbation
```

Quelques requêtes en supplément :

10. Afficher l'arborescence d'une maladie

```
SELECT id_maladie , nom_maladie , level
FROM maladie
START WITH type_maladie = &Maladie
CONNECT BY PRIOR id_maladie = type_maladie
ORDER BY LEVEL ASC , id_maladie ASC;
```

```
SQL> SELECT id_maladie , nom_maladie , level
  2   FROM maladie
  3   START WITH type_maladie = &Maladie
  4   CONNECT BY PRIOR id_maladie = type_maladie
  5   ORDER BY LEVEL ASC , id_maladie ASC;
Entrez une valeur pour maladie : 0
ancien   3 :  START WITH type_maladie = &Maladie
nouveau   3 :  START WITH type_maladie = 0

ID_MALADIE NOM_MALADIE                                          LEVEL
---------- -------------------------------------------------- ----------
         1 Maladie de l appareil digestif                              1
        11 Maladie de l estomac                                        2
        12 Maladies du foie                                            2
        13 Maladies du colon                                           2
       111 Gastrite                                                    3
       112 Ulsaire                                                     3
       121 cirrhose du foie                                            3
       122 Hepatite                                                    3
       123 Tuberculose hepatique                                       3
       131 Crhon                                                       3
       132 Colon héritable                                             3

ID_MALADIE NOM_MALADIE                                          LEVEL
---------- -------------------------------------------------- ----------
      1221 Hepatite alcoolique                                         4
      1222 Hepatite chronique                                          4
      1223 Hepatite virales humaines                                   4
     12231 Hepatite A                                                  5
     12232 Hepatite B                                                  5
     12233 Hepatite C                                                  5

17 ligne(s) sélectionnée(s).
```

11. Donner tout les médecins d'un laboratoire.

```
select m.nom_medecin as Medecin , l.id_laboratoire as Laboratoire
from medecin m , laboratoire l
where m.id_laboratoire = l.id_laboratoire and l.id_laboratoire = &laboratoire;
```

```
SQL> select m.nom_medecin as Medecin , l.id_laboratoire as Laboratoire
  2  from medecin m , laboratoire l
  3  where m.id_laboratoire = l.id_laboratoire and l.id_laboratoire = &laboratoire;
Entrez une valeur pour laboratoire : 7
ancien   3 : where m.id_laboratoire = l.id_laboratoire and l.id_laboratoire = &laboratoire
nouveau   3 : where m.id_laboratoire = l.id_laboratoire and l.id_laboratoire = 7

MEDECIN                                            LABORATOIRE
-------------------------------------------------- -----------
GARAY-LOPEZ                                                  7
GILES                                                        7
ERIKSSON                                                     7
```

12. Liste des médecins et les médicaments qu'ils ont créés.

```
select m.nom_medecin as Medecin, me.nom_medic as Medicament
from medecin m, medicament me , produit p
where p.id_medecin = m.id_medecin and p.id_medicament = me.id_medicament
order by m.nom_medecin;
```

```
SQL> select m.nom_medecin as Medecin, me.nom_medic as Medicament
  2  from medecin m, medicament me , produit p
  3  where p.id_medecin = m.id_medecin and p.id_medicament = me.id_medicament
  4  order by m.nom_medecin;

MEDECIN
--------------------------------------------------
MEDICAMENT
--------------------------------------------------
ALLON
BARACLUDE 0,05 mg/ml

BACARD
HAVRIX 1440

BACARD
OMEPRAZOLE ABBOTT 10 mg, gélule gastro-résistante


MEDECIN
--------------------------------------------------
MEDICAMENT
--------------------------------------------------
BAKER
CELESTENE 4 mg/1 ml, solution injectable

CHRISTIE
AMOXICILLINE / ACIDE CLAVULANIQUE MYLAN 500 mg

ERIKSSON
CLARITHROMYCINE ABBOTT 250 mg, comprimé pelliculé
```

13. Interaction entre les médicaments.

```
select i.id_interaction as Interaction,i.id_medic1 as Medicament, i.id_medic2 as Medicament
from intercation_medicament i;
```

```
SQL> select i.id_interaction as Interaction,i.id_medic1 as Medicament, i.id_medic2 as Medicament
  2  from intercation_medicament i;

INTERACTION MEDICAMENT MEDICAMENT
----------- ---------- ----------
          1          1          5
          2          2          8
          3          3          9
```