

Projet Bases de Données Juste Prescription des
Médicaments SQL3

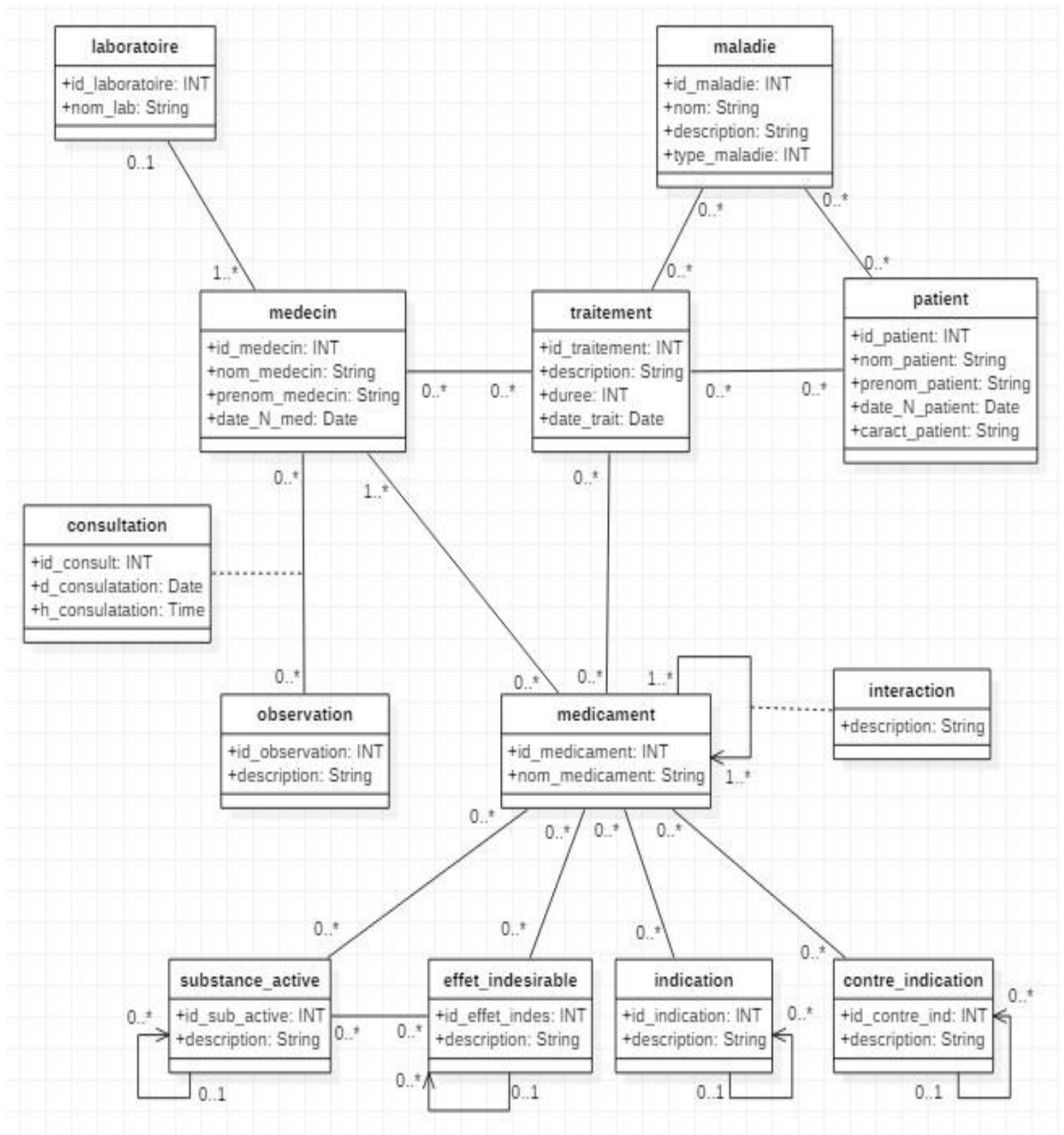
Le plan :

- Description du projet**
- Présentation du modèle relationnel**
- Description des règles de gestion**
- Les requêtes et les résultats**

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.

Présentation du modèle relationnel



On trouve dans le modèle relationnel les tables suivantes :

- **Maladie** : une table de type mal qui comporte un identifiant maladie, le nom de la maladie, une description qui sont définies dans le type mal. Id_maladie c'est la clé primaire de la table maladie.

- **Patient** : qui est caractérisé par un identifiant id_patient (clé primaire), un nom_patient, un prénom_patient et une date de naissance, caractéristiques d'un patient qui sont « **jeune, vieux, enceinte et sportif** » et une list_maladie de type table_mal comme une table imbriquée représente la liste des maladies du patient.

- **Laboratoire** : se compose d'un identifiant (clé primaire) et d'un nom de laboratoire.

- **Médicament** : est une table de type medic qui contient un identifiant de médicament, un nom. Id_medicament c'est la clé primaire de la table médicament.

- **Interaction** : qui est définie par un identifiant propre (clé primaire), deux identifiants de médicament comme des clés étrangères et une description de l'interaction.

- **Médecin** : contient un identifiant (clé primaire), un nom, un prénom, une date de naissance et un identifiant de laboratoire comme clé étrangère qui représente le laboratoire auquel le médecin appartient (il peut être nul) et une table imbriquée list_dev_medic de type medic qui représente la liste des médicaments qu'il a développés.

- **Traitement** : qui se traduit par un identifiant de traitement (clé primaire), un identifiant de maladie, un identifiant de médicament, un identifiant de médecin, une durée et une list_medic de type table_medic comme une table imbriquée représente la liste des médicaments. Elle contient aussi, id_medicine, id_patient et id_maladie sont des clés étrangères dans *traitement*.

- **Substance active** : contient un identifiant (clé primaire), une description, et une table imbriquée list_dev_medic de type medic qui représente la liste des médicaments.

- **Effet indésirable** : contient un identifiant (clé primaire), une description, et une table imbriquée list_dev_medic de type medic qui représente la liste des médicaments.

- **substance_effet** : qui est définie par un identifiant propre (clé primaire), deux identifiants de la table substance active et effet comme des clés étrangères.

- **Indication** : contient un identifiant (clé primaire), une description, et une table imbriquée list_dev_medic de type medic qui représente la liste des médicaments.

- **Contre-indication** : contient un identifiant (clé primaire), une description, et une table imbriquée list_dev_medic de type medic qui représente la liste des médicaments.

- **observation** : contient un identifiant (clé primaire), une description.

- **Consultation** : qui est définie par un identifiant propre (clé primaire), deux identifiants de la table médecin et observation comme des clés étrangères.

Description des règles de gestion

- 1) Des patients qui suivent des traitements
- 2) Des traitements prescrits par des médecins suite à des observations faites lors de consultations.
- 3) Un patient souffre d'une ou plusieurs maladies.
- 4) Un médecin n'appartient à aucun ou un seul laboratoire.
- 5) Un laboratoire contient un ou plusieurs médecins.
- 6) Un médicament est développé par un seul laboratoire et par un ou plusieurs médecins.
- 7) Un traitement traite un ou plusieurs maladies.
- 8) Un médicament possède plusieurs caractéristiques. Notamment : les indications, contre-indications, des substances actives, des effets indésirables.
- 9) Une substance active peut générer des effets indésirables.
- 10) Deux médicaments pris simultanément peuvent provoquer des interactions médicamenteuses.
- 11) Une ou plusieurs substances actives génèrent un ou plusieurs effets indésirables.
- 12) Une maladie peut être traitée par un ou plusieurs traitements.
- 13) Un médicament peut être dans plusieurs traitements.
- 14) Un effet indésirable, une substance active, une contre-indication et une indication peuvent être indiqués dans un ou plusieurs médicaments.

Les requêtes et les résultats :

1. Une méthode prescription qui permettra de sauvegarder le choix de traitement (liste des médicaments) et les maladies diagnostiquées par le médecin pour un patient.

```
create OR replace procedure prescription (id_med in integer, id_pat in integer) IS
    x integer;
    list table_medic;
    nom_maladie varchar(50);
BEGIN
    select list_medic , id_maladie
    into list, x
    from traitement
    where id_medecin = id_med and id_patient=id_pat;

    select nom
    into nom_maladie
    from maladie
    where id_maladie=x;

    DBMS_OUTPUT.PUT_LINE('maladie prescrit = '||nom_maladie);
    for i in (select l.* from table(list) l)
        loop
            DBMS_OUTPUT.PUT_LINE('medic prescrit = '||i.nom_medicament);
        end loop;
EXCEPTION
    WHEN no_data_found THEN
        dbms_output.put_line('No such customer!');
    WHEN others THEN
        dbms_output.put_line('Error!');
END;
```

/

```
execute      prescription(10,4);
```

/*resultat

maladie prescrit = Ultaire

medic prescrit = CELESTENE 4 mg/1 ml, solution injectable

medic prescrit = CLARITHROMYCINE ABBOTT 250 mg, comprimé pelliculé

2. Une méthode donnant les traitements en cours d'un patient. id_patient=4

```
create or replace procedure tre (id_pat in integer, date_trait1 in date) is

    idt traitement.id_traitement%type;
    descript traitement.description%type;
    x integer;
    y varchar(50);
    CURSOR c1
    IS
        SELECT t.id_traitement,t.description, id_patient
        FROM traitement t
        where id_patient=id_pat and
        (select ABS(to_date(date_trait1, 'DD-MM-YYYY') - to_date(t.date_trait, 'DD-MM-YYYY'))
diff_hours from dual)<t.duree;

    BEGIN
        DBMS_OUTPUT.PUT_LINE('-----');
        open c1;
        loop
            FETCH c1 into idt, descript, x;
            EXIT WHEN c1%notfound;
            --DBMS_OUTPUT.PUT_LINE(idt);
            select nom_patient
            into y
            from patient
            where id_patient = x;
            DBMS_OUTPUT.PUT_LINE('nom patient = '||y);
            DBMS_OUTPUT.PUT_LINE('traitement en cours = '||descript);
        end loop;
        close c1;
    END;

/

execute tre(4,'05-09-2016');
```

/*résultat pour id_patient=4

nom patient = cady

traitement en cours = BARACLUDE 3 fois par jour....

3. une méthode donnant la liste et le nombre d'effets indésirables connus d'un médicament.

```
create or replace procedure listNbrEffect (id_medic in integer) IS
    nbr_eff integer;
    c_id effect_indesirable.description%type;
    CURSOR c1
    IS
        SELECT e.description
        FROM effect_indesirable e, table(e.list_effect_medic) m
        WHERE m.id_medicament =id_medic;
    BEGIN
        select count(*)
        into nbr_eff
        from effect_indesirable e, table(e.list_effect_medic) m
        where m.id_medicament =id_medic;
        DBMS_OUTPUT.PUT_LINE('nbr effets indesirables = '||nbr_eff);
        DBMS_OUTPUT.PUT_LINE('-----');
        DBMS_OUTPUT.PUT_LINE('les decription des effets sont = ');
        IF (nbr_eff <> 0) then
            open c1;
            loop
                FETCH c1 into c_id;
                EXIT WHEN c1%notfound;
                DBMS_OUTPUT.PUT_LINE(c_id);
            end loop;
            close c1;
        end IF;
    END;
/
execute      listNbrEffect(3);
```


/*résultat pour id_medicament=3

nbr effets indésirables = 2

les descriptions des effets sont =

Irritabilité, Maux de tête, Douleur et rougeur au point d'injection

Syndrome hépato-rénal

4. Une méthode donnant la liste des médicaments pouvant générer des interactions et l'interaction pour un médicament donné.

```
create or replace procedure genereInteractions (id_medic in integer) IS
```

```
    nom_medic varchar(50);
```

```
    nom_medic2 varchar(50);
```

```
    --descrip varchar(50);
```

```
    med medicament.nom_medicament%type;
```

```
    id_med1 intercation.id_medicament1%type;
```

```
    id_med2 intercation.id_medicament2%type;
```

```
    descrip intercation.description%type;
```

```
    des intercation.description%type;
```

```
    CURSOR c1
```

```
    IS
```

```
        SELECT id_medicament2 , description
```

```
        FROM intercation
```

```
        WHERE id_medicament1 =id_medic;
```

```
    BEGIN
```

```
        select nom_medicament
```

```
        into nom_medic
```

```
        from medicament
```

```
        where id_medicament=id_medic;
```

```
        DBMS_OUTPUT.PUT_LINE('le nom de medicament = '||nom_medic);
```

```
        DBMS_OUTPUT.PUT_LINE('-----');
```

```
        DBMS_OUTPUT.PUT_LINE('les medicament provoque interactions avec '||nom_medic||' sont =
```

```
    ');
```

```
        open c1;
```

```
        loop
```

```

        FETCH c1 into id_med2, descrip;
        EXIT WHEN c1%notfound;

        DBMS_OUTPUT.PUT_LINE('id_medic ='||id_med2);

        select nom_medicament
        into med
        from medicament
        where id_medicament=id_med2;

        DBMS_OUTPUT.PUT_LINE('nom de medicament' || med);
        DBMS_OUTPUT.PUT_LINE('intercation provoquée      ='||descrip);

    end loop;

    close c1;

END;

/
execute      genereInteractions(4);

```

/*résultat pour id_medicament=4

le nom de medicament = ALDACTONE 50 mg, comprimé sécable

les medicament provoque interactions avec ALDACTONE 50 mg, comprimé sécable sont

=

id_medic =2

nom de medicament OMEPRAZOLE ABBOTT 10 mg, gélule gastro-résistante

intercation provoquée =Ensemble ces deus medicament provoque des effet negatifs

des poumons

5. Une fonction permettant de proposer une liste de médicaments `a partir de la maladie diagnostiquée, même si un lien direct maladie-médicament n'existe pas.

```
create or replace function poposmedic (id_mal in integer)
```

```
RETURN table_medic IS
```

```
x integer;
```

```
list_medicament table_medic;
```

```
nom_maladie varchar(50);
```

```
BEGIN
```

```
    select nom into nom_maladie
```

```

        from maladie where id_maladie=id_mal;

        DBMS_OUTPUT.PUT_LINE('le nom de la maladie = '||nom_maladie);

        select list_medic into list_medicament

        from traitement where id_maladie = id_mal;

        return list_medicament;

    END;

/

```

/* l'exécution de la fonction

```

        declare

        liste table_medic;

        begin

        liste := poposmedic(12233);

        for r in (select l.* from table(liste) l)

                loop

                        DBMS_OUTPUT.PUT_LINE('medic prescrit = '||r.nom_medicament);

                end loop;

        END;

/

```

/*résultat pour id_maladie = 12233

le nom de la maladie = Hepatite C

medic prescrit = AMOXICILLINE / ACIDE CLAVULANIQUE MYLAN 500 mg

medic prescrit = SOVALDI 400 mg, comprimé pelliculé

medic prescrit = BARACLUDE 0,05 mg/ml

medic prescrit = HAVRIX 1440

6. Une méthode qui détermine pour un médicament la liste des effets indésirables probables (dédiés des hiérarchies de substances actives).

create or replace procedure determinEffet (id_medic in integer) IS

```

        id1 sub_effect.id_effet_indes%type;

        des_eff_ind varchar(250);

        id substance_active.id_sub_active%type;

        CURSOR c1

        IS

                SELECT s.id_sub_active

```

```

FROM substance_active s, table(s.list_sub_medic) l
WHERE l.id_medicament =id_medic;

BEGIN
    open c1;
        loop
            FETCH c1 into id;
            EXIT WHEN c1%notfound;
            DBMS_OUTPUT.PUT_LINE('id sub active ='||id);
            DBMS_OUTPUT.PUT_LINE('-----');
            select id_effet_indes
            into id1
            from sub_effect
            where id_sub_active=id;
            DBMS_OUTPUT.PUT_LINE('id iffet ='||id1);
            DBMS_OUTPUT.PUT_LINE('-----');
            select description
            into des_eff_ind
            from effect_indesirable
            where id_effet_indes = id1;
            DBMS_OUTPUT.PUT_LINE('description effet indesirable = '||des_eff_ind);
        end loop;
    close c1;
END;
/
execute      determinEffet(8);

```

/*résultat pour id_medicament=8

id sub active =5

id iffet =6

description effet indesirable = gonflement et rougeur du visage, prise de poids

id sub active =7

id iffet =4

description effet indésirable = Troubles digestifs, Eruption cutanée,

Insuffisance rénale aiguë

id sub active =8

id effet =3

description effet indésirable = allongement de l'intervalle QT, dépression,

vertiges, trouble du goût, perte de l'odorat

7. Afin de contrôler les prescriptions, on doit pouvoir déterminer s'il y a un ensemble de médicaments qui ne sont prescrits que par des médecins qui ont travaillé à leur développement.

```
create or replace procedure medic_prescrit_medecin_dev is
```

```
    id_med traitement.id_medecin%type;
```

```
    id_med medicament.id_medicament%type;
```

```
    nom_med medicament.nom_medicament%type;
```

```
    nom_med medecin.nom_medecin%type;
```

```
    x integer;
```

```
    CURSOR c1
```

```
    IS
```

```
        SELECT m1.id_medicament, t.id_medecin
```

```
        FROM traitement t, table(t.list_med) m1
```

```
        WHERE m1.id_medicament NOT IN
```

```
            (select m2.id_medicament
```

```
            from medecin m0, table(m0.list_dev_med) m2
```

```
            where m0.id_medecin <> t.id_medecin);
```

```
    BEGIN
```

```
        open c1;
```

```
        loop
```

```
            FETCH c1 into id_med, id_med;
```

```
            EXIT WHEN c1%notfound;
```

```
            x:=0;
```

```
            select count(*) into x from traitement t1, table(t1.list_med) m3 where
```

```
m3.id_medicament = id_med;
```

```

                                if (x=1)then
                                    select m4.nom_medicament
                                    into nom_medic
                                    from medicament m4
                                    where m4.id_medicament=id_medic;
                                    DBMS_OUTPUT.PUT_LINE('medicament ='||nom_medic);
                                    select nom_medecin
                                    into nom_med
                                    from medecin
                                    where id_medecin=id_med;
                                    DBMS_OUTPUT.PUT_LINE('medecin qui a developpé le
medicament ='||nom_med);
                                    DBMS_OUTPUT.PUT_LINE('-----');
                                end if;
                            end loop;
                        close c1;
                    END;
/
execute medic_prescrit_medecin_dev();

```

/*résultat

medicament =ALDACTONE 50 mg, comprimé sécable

medecin qui a developpé le medicament =BELAIR

medicament =OMEPRAZOLE ABBOTT 10 mg, gélule gastro-résistante

medecin qui a developpé le medicament =FANTINI

8. On doit pouvoir déterminer s'il y a des médicaments qui ne sont prescrits que par des médecins ayant travaillé dans les laboratoires les fabriquant.

```

create or replace procedure medic_prescrit_lab_dev is
    id_med traitement.id_medecin%type;
    id_medic medicament.id_medicament%type;
    nom_medic medicament.nom_medicament%type;
    nom_med medecin.nom_medecin%type;
    x integer;

```

```

y integer;
z varchar(250);
CURSOR c1
IS
    SELECT m1.id_medicament, t.id_medecin
    FROM traitement t, table(t.list_medic) m1
    WHERE m1.id_medicament NOT IN
        (select m2.id_medicament
        from medecin m0, table(m0.list_dev_medic) m2
        where m0.id_medecin <> t.id_medecin);

BEGIN

    open c1;
    loop
        FETCH c1 into id_medic,id_med;
        EXIT WHEN c1%notfound;

        x:=0;
        select count(*) into x from traitement t1, table(t1.list_medic) m3 where
m3.id_medicament = id_medic;

        if (x=1)then
            select m4.nom_medicament
            into nom_medic
            from medicament m4
            where m4.id_medicament=id_med;
            DBMS_OUTPUT.PUT_LINE('medicament ='||nom_medic);
            select id_laboratoire
            into y
            from medecin
            where id_medecin=id_med;
            select nom_lab
            into z
            from laboratoire
            where id_laboratoire=y;
            DBMS_OUTPUT.PUT_LINE('nom laboratoire ='||z);
            DBMS_OUTPUT.PUT_LINE('-----');
        end if;
    end loop;
END;

```

```

                                end if;

                                end loop;

                                close c1;

                                END;

/
execute medic_prescrit_lab_dev();

```

/*résultat de l'exécution

medicament =ALDACTONE 50 mg, comprimé sécable

nom laboratoire =GROUPE AGUETTANT SANTÉ, FRANCE

medicament =OMEPRAZOLE ABBOTT 10 mg, gélule gastro-résistante

nom laboratoire =ABBOTT France

9. On doit pouvoir identifier la/les maladie(s) probable(s) et aider à la prescription en fonction d'observations (symptômes) et des caractéristiques du patient (vous pourrez trier les traitements proposés par nombre d'effets indésirables par exemple).

```

create or replace procedure maladie_propable(id_obs in integer, cars_pat in varchar) is

```

```

    x integer;

```

```

    y integer;

```

```

    z integer;

```

```

    nom_mal varchar(50);

```

```

    id integer;

```

```

    lis table_mal;

```

```

    CURSOR c1

```

```

    IS

```

```

        SELECT id_patient

```

```

        FROM patient

```

```

        WHERE caract_patient = cars_pat;

```

```

    BEGIN

```

```

        select id_medecin

```

```

        into x

```

```

        FROM consultation

```



```

WHERE id_observation = id_obs;

select id_maladie, id_patient

into y,z

FROM traitement

WHERE id_medecin = x;

select m.nom

into nom_mal

FROM maladie m

WHERE m.id_maladie = y;

DBMS_OUTPUT.PUT_LINE('maladie = '||nom_mal);

open c1;

    loop

        FETCH c1 into id;

        EXIT WHEN c1%notfound;

        SELECT p.list_maladie

        into lis

        FROM patient p

        WHERE id_patient = id;

        for i in (select l.* from table(lis) l)

        loop

            DBMS_OUTPUT.PUT_LINE('maladie = '||i.nom);

        end loop;

    end loop;

close c1;

END;

/

execute maladie_propable(3,'vieux');

```

/*résultat pour id_observation=3

maladie = Hepatite alcoolique

maladie = Gastrite

maladie = cirrhose du foie

maladie = Hepatite alcoolique

- 10.** Une fonction permettant d'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. Il prend en paramètre une liste de médicament, et id_patient et date de prescription de traitement, pour trouver si y a des traitements en cours qui peuvent interagir avec le traitement prescrit

[illegible]

```

interactions entre = '||i.nom_medicament||' et : '||j.nom_medicament||' en cours de traitement');

                                end if;

                                end loop;

                                end loop;

                                end loop;

                                close c1;

END;

/

execute terer(4,'05-09-2016',table_medic(medic(2,'OMEPRAZOLE ABBOTT 10 mg, gélule gastro-
résistante'),medic(9,'SOVALDI 400 mg, comprimé pelliculé')));

```

/*resultat

```

medicament =BARACLUDE 0,05 mg/ml

on a 1 interactions entre = BARACLUDE 0,05 mg/ml et : OMEPRAZOLE ABBOTT 10 mg,
gélule gastro-résistante en cours de traitement

medicament =HAVRIX 1440

medicament =AMOXICILLINE / ACIDE CLAVULANIQUE MYLAN 500 mg

medicament =SOVALDI 400 mg, comprimé pelliculé

```