

Université de Rouen, UFR Sciences et
Techniques
Master 1 IGIS – Bases de données
Gestion des patients et des traitements

Boubekri, Abdelmalek `abdelmalek.boubekri@etu.univ-rouen.fr`
Mezheri, Bilal `bilal.mezheri@etu.univ-rouen.fr`

7 janvier 2017

Table des matières

1	Contexte	2
1.1	Règles métier du système d'information	3
2	Modélisation	5
2.1	Modèle entité association	5
2.2	Modèle relationnel	5
2.2.1	Règles de passage au modèle relationnel REF!!!	5
2.2.2	Modèle logique des données	5
2.3	Implémentation de la base de données	6
2.4	Fonctionnalités demandées	6
2.4.1	Sauvegarde des diagnostics et des choix de traitement .	6
2.4.2	Traitements en cours	6
2.4.3	Effets indésirables	6
2.4.4	Médicaments générant des interactions	6
2.4.5	Recommandations à partir d'observations	6
2.4.6	Effets indésirables d'un médicament	6
2.4.7	Contrôle des prescriptions	7
2.4.8	7
2.4.9	Identification des maladies probables	7
2.4.10	Risques d'interaction lors des prescriptions	7
2.5	Outils Oracle utilisés	7
2.5.1	Tables en ligne ou en colonne et types correspondants .	7
2.5.2	Tables imbriquées	8
2.5.3	Vues	8
2.5.4	Séquences	8
2.5.5	Requêtes hiérarchiques	8
2.5.6	Index	8
2.5.7	Triggers	8
2.5.8	Procédures/fonctions	8
2.5.9	PL	8
2.5.10	Curseurs	8
2.5.11	Références	8

1 Contexte

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 a une durée et est constitué de médicaments.

Un médicament possède plusieurs caractéristiques. Notamment : les indications, contre-indications, des substances actives, des effets indésirables connus, disponibles dans des notices.

Une substance active peut générer des effets indésirables.

Deux médicaments pris simultanément peuvent provoquer des interactions médicamenteuses. Les maladies, substances actives, les effets indésirables sont organisés de manière hiérarchique avec héritage des propriétés.

Un patient a également des caractéristiques et peut souffrir d'une maladie chronique qui nécessite un traitement de longue durée.

Un médecin peut travailler pour un laboratoire pharmaceutique et peut également développer un médicament.

1.1 Règles métier du système d'information

les règles de gestion sont décrites ci-dessous:

1. Un patient peut consulter un médecin;
2. Une maladie a plusieurs symptômes;
3. Durant une consultation, plusieurs symptômes peuvent être observés;
4. On peut traiter une maladie avec un traitement;
5. Un médicament peut être développé par un laboratoire et plusieurs médecins;
6. Un médicament a une notice;
7. Une notice contient des indication, contre-indications et des substances actives;
8. Une substance active peut avoir plusieurs effets indésirables;
9. Une substance active peut interagir avec une autre substance active;
- 10.

Après avoir pris connaissance des exigences, on doit suivre les étapes suivantes afin de concrétiser la base de données.

- Modélisation E/A avec l'outil open-source Dia;
- Transformation au modèle relationnel ;
- Implémentation de la Base de données en PL/SQL via Oracle-xe-11g.

2 Modélisation

2.1 Modèle entité association

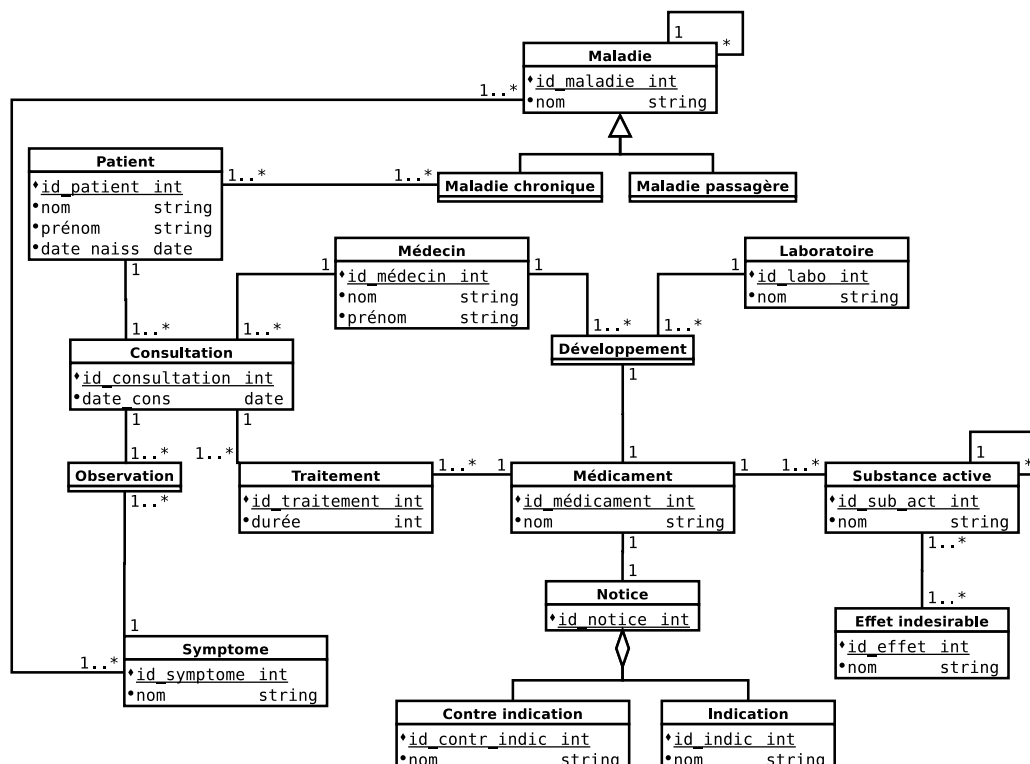


FIG. 1 – *Diagramme de classes*

2.2 Modèle relationnel

2.2.1 Règles de passage au modèle relationnel REF!!!

1. Toute classe d'entités du diagramme E/A est représentée par une relation dans le schéma relationnel équivalent. La clé de cette relation est l'identifiant de la classe d'entités correspondante.
2. Toute classe d'association est transformée en relation. La clé de cette relation est composée de tous les identifiants des entités participantes.
3. Toute classe d'associations reliée à une classe d'entités avec une cardinalité de type 0,1 ou 1,1 peut être fusionnée avec la classe d'entités. Dans ce cas on déplace les attributs de la classe d'associations vers ceux de la relation traduisant la classe d'entités.

2.2.2 Modèle logique des données

indication (id_indication, nom_indication)
contre_indication (id_contre_indic, nom_contre_indic)
substance_active (id_sub_act, nom_sub_act)

effet_indésirable(id_effet, nom_effet)
avoir_effet(id_sub_act#, id_effet#)
notice(id_notice, indication, contre_indication, substance_active)
laboratoire(id_labo, nom, adresse)
médecin(id_médecin, nom, prénom, adresse)
médicament(id_médicament, nom, id_labo#, id_médecin#, id_notice#)
symptôme(id_symptôme, nom)
maladie(id_maladie, nom)
avoir_symptôme(id_symptôme#, id_maladie#)
consultation(id_patient#, id_médecin#, date)
patient(id_patient, nom, date_naissance)
observation(id_consultation#, id_symptôme#)
traitement(durée, id_médicament#, id_maladie#)
recommandation()

2.3 Implémentation de la base de données

Implémentation effectuée sous Oracle-xe-11g shipé dans un conteneur Docker, accessible sur le Docker Hub ainsi que sur Github.

2.4 Fonctionnalités demandées

2.4.1 Sauvegarde des diagnostics et des choix de traitement

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

2.4.2 Traitements en cours

Une méthode donnant les traitements en cours d'un patient.

2.4.3 Effets indésirables

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

2.4.4 Médicaments générant des interactions

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

2.4.5 Recommandations à partir d'observations

Une fonction permettant de proposer une liste de médicaments à partir de la maladie diagnostiquée, même si un lien direct maladie-médicament n'existe

pas.

2.4.6 Effets indésirables d'un médicament

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

2.4.7 Contrôle des prescriptions

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.

2.4.8

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.

2.4.9 Identification des maladies probables

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).

2.4.10 Risques d'interaction lors des prescriptions

Une fonction permettant d'indiquer à un médecin prescrivait si le traitement envisagé, risque d'interagir avec un traitement 'en cours' et proposer le cas échéant un autre traitement.

2.5 Outils Oracle utilisés

2.5.1 Tables en ligne ou en colonne et types correspondants

indication_tpe et **contre_indic_tpe** sont des tables en ligne.

```
1
2 CREATE TYPE indication_tpe AS TABLE OF indic_type;
3 /
4
5 CREATE TYPE contre_indic_tpe AS TABLE OF contre_indic_type;
6 /
```

2.5.2 Tables imbriquées

L'imbrication des tables **indication** et **contre_indication** dans **notice**.

```
1
2 CREATE TABLE notice(id_notice NUMBER(38) CONSTRAINT pk_notice
3     PRIMARY KEY,
4     indications indication_tpe,
5     contre_indications contre_indic_tpe)
6 NESTED TABLE indications STORE AS indications_table,
7 NESTED TABLE contre_indications STORE AS contre_indic_table;
```

- 2.5.3 Vues**
- 2.5.4 Séquences**
- 2.5.5 Requêtes hiérarchiques**
- 2.5.6 Index**
- 2.5.7 Triggers**
- 2.5.8 Procédures/fonctions**
- 2.5.9 PL**
- 2.5.10 Curseurs**
- 2.5.11 Références**