

Franck Caron, Pierre Dechamps
M1 GIL

Projet de Base de Données

Informations générales

Ce document a pour but de décrire le contenu du projet ainsi que d'expliquer les différents choix effectués sur la structure de notre schéma de base de données.

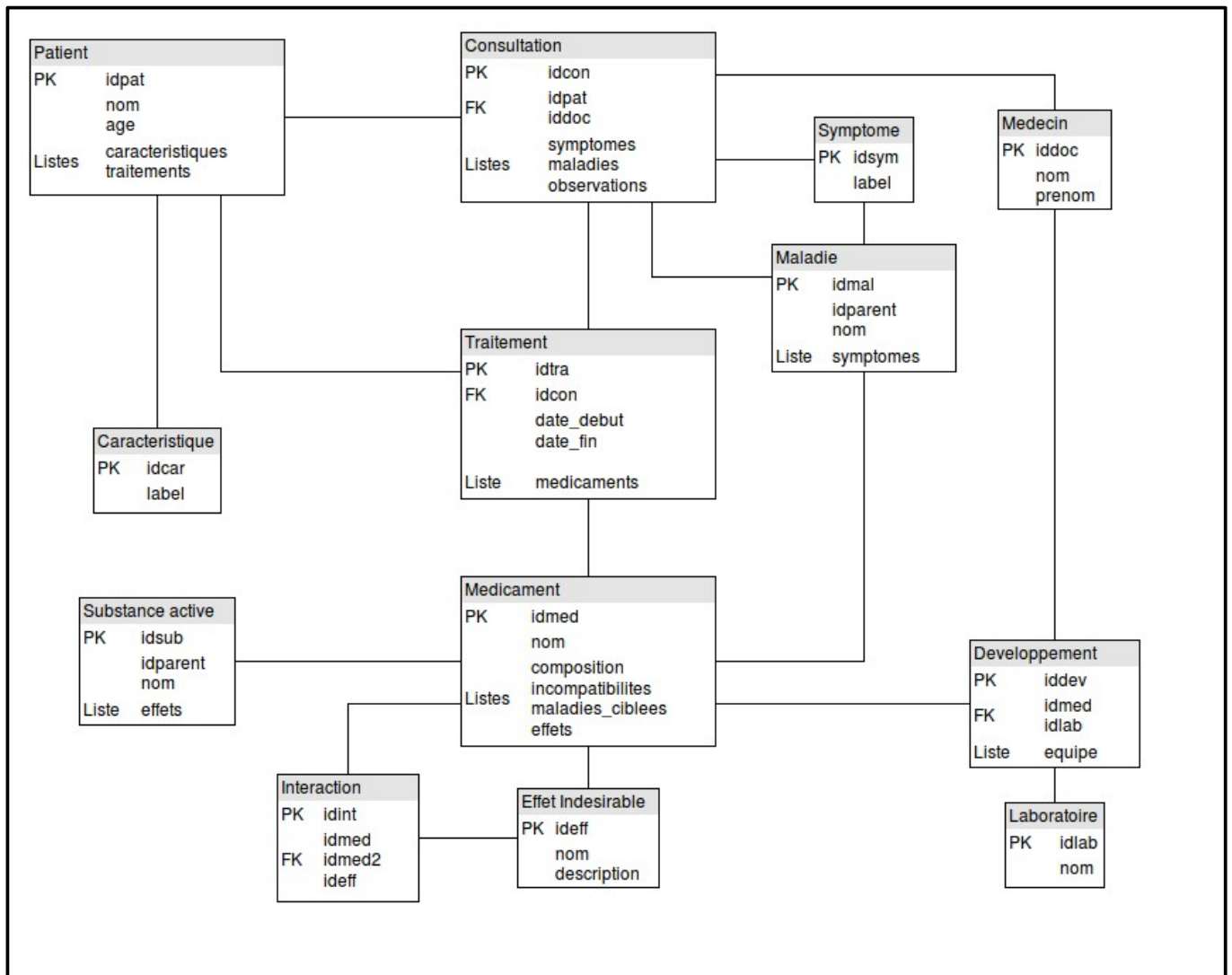
Le code source du projet se décompose en plusieurs fichiers présents dans le sous-dossier src :

1. **clean.sql** : Contient les requêtes permettant de désinstaller le contenu défini par le projet (tables, types, triggers...).
2. **base.sql** : Définit les différents types et tables définissant la structure de notre modèle de données.
3. **triggers.sql** : décrit l'ensemble des procédures automatiques (triggers) définis pour ce modèle de données.
4. **insert.sql** : Offre un jeu de données initial pour les tables du modèle
5. **procedures.sql** : qui décrit l'ensemble des procédures et fonctions définies pour le projet

Afin d'installer le projet sur une base de données vierge, il vous suffit simplement d'évaluer le contenu des fichiers 2 à 5 dans le terminal d'évaluation des requêtes SQL. Le contenu du fichier n°1 ne vous sera utile que pour la désinstallation.

Structure

Nous avons représenté notre modèle de données selon le schéma suivant :



Nous avons défini chacun des types conformément à la modélisation initiale définie par le sujet. Chaque type d'entité représenté ci-dessus correspond à une table et à un type défini (où la table, est une table d'entité du type donné).

Plusieurs des types d'entités définis représentent des types hiérarchisés (Substance active, Maladie, ...), nous avons alors défini cette hiérarchie à l'aide d'un identifiant supplémentaire idparent. Cela permettra lors des procédures et des fonctions, de pouvoir faire appel aux fonctionnalités de SQL Oracle concernant les requêtes hiérarchiques, au moyen de la commande CONNECT BY.

Afin de modéliser les différentes interactions entre les types d'entité, nous avons défini un type modélisant les listes de clés. Ce dernier représente une table d'entiers qui servira à simuler des listes (de longueur variable) d'identifiants. L'utilité de ces listes intervient lors des relations de contenance entre entités : au lieu de créer des tables de données pour modéliser des associations « contient », il suffit ici de définir les identifiants des contenus au sein du contenant, évitant ainsi une redondance des données.