

# Rapport - Base de données : projet Système d'information d'un hôpital

XIAN YANG, M1 Bio-Statistique

04 décembre 2016

## Résumé

Ce présent projet, dont le sujet consiste à concevoir et à réaliser un système d'information d'un hôpital, a pour but, d'une part, d'analyser le problème donné en modèle de base de données relationnel et de l'autre, de mettre en place une base de données sous MySQL à partir de son schéma conceptuel obtenu par l'étape précédente. Dans l'ensemble de la réalisation de ce projet et en plus de faire ce qui est demandé, une grande partie de mes efforts a été dédiée à optimiser la fiabilité du modèle, c'est à dire à tester la base de données avec des jeux de données de taille réelle. Dans cet objectif, la génération de données de test volumineuses a été étudiée et de différentes fonctionnalités d'importation et d'exportation de données de MySQL ont été utilisées.

Vous avez devant vous un compte-rendu intégré qui est composé d'un rapport `rapportxyang.pdf`, un compte-rendu `compteRenduxyang.pdf`, trois fichiers sql `createDBxyang.sql`, `dropTablesxyang.sql`, `insertTablesxyang.sql` ainsi que 2 fichiers de données `myrawmalade.csv` et `CIS_bdpm.txt`. Le premier contient les 600 malades virtuels séjournant à notre hôpital lorsque le deuxième est un relevé des 14.297 médicaments circulant actuellement en France téléchargé de `gouv.fr`. Vous êtes priée de mettre le chemin de ces fichiers dans votre ordinateur aux endroits où il est indiqué dans le fichier sql `insertTablesxyang.sql`, afin que ce dernier puisse être bien exécuté. Aussi je vous conseille fortement de jeter un coup d'œil dans le fichier `createDBxyang.sql` où vous trouverez des commentaires ici et là expliquant mes idées de programmation.

L'environnement de programmation de ce projet est principalement MySQL Workbench 3.8, lorsque la génération de données étrangères a été effectuée à travers le logiciel statistique R (paquet "randomNames") et de divers sites d'internet à cet effet (`generatedata.com` etc.).

## 1 Hypothèses supplémentaires

Il convient de mettre en place les hypothèses suivantes afin de donner plus de clarté au problème.

### *Bâtiment*

L'hôpital possède en total 6 bâtiments numérotés de 1 à 6. Ici seul le numéro est d'importance et toute autre information telle que la localisation ne sera pas prise en compte. Toutefois, j'ai établi une table pour cette unité de crainte d'une extension future.

### *Service*

Chaque service de l'hôpital porte un numéro d'identité unique de 2 chiffres, dont le premier correspond à celui du bâtiment dans lequel il se situe. Les services sont répartis comme suit :

11 - Direction	12 - Service Comptabilité	13 - Administration
14 - Logistique	15 - Salle de Garde	21 - Médecine générale
22 - Médecine interne	23 - Cardiologie	24 - Pneumologie
25 - Gastro-entérologie	31 - Immunologie	32 - Endocrinologie
33 - Urologie	34 - Gynécologie	35 - Maternité
41 - Traumatologie	42 - Dermatologie	43 - Service des grands brûlés
44 - Chirurgie	51 - Transfusion sanguine	52 - Laboratoires d'analyse
53 - Radiologie	61 - Pharmacie	62 - Service d'accueil de traitement des urgences

### *Médecin*

Tout le personnel de l'hôpital est appelé médecin, qu'il travaille dans un secteur médical ou non. Il y a 214 médecins inscrits.

### *Chambre*

Chaque service médical (21-25, 31-35, 41-44, 62) dispose de 14 chambres à 3 lits à l'exception des services d'urgence (62), qui dispose de 12 chambres de lit simple. L'ensemble de ces 210 chambres assure le séjour de 600 malades. Il y a de plus 2 chambres non-attribuées pour des cas particuliers.

### *Malade*

Six cents malades, âgés de 16 à 80 ans, dont 350 hommes et 250 femmes, séjournent à cet hôpital. Environ 40% d'entre eux portent une contre-indication parmi « grossesse » (uniquement femme), « diabète », « allergie » et « claustrophobie ». Je suppose qu'au départ ou décès d'un malade on ne remet que son numéro de chambre à NULL, grâce à quoi on peut garder trace de son historique. Cela étant dit, j'entends par garde de trace la conservation des renseignements du patient pendant son séjour à *notre* hôpital. On négligera toutes ses visites médicales d'ailleurs et ne fera ici pas de « Carte Vitale ».

### *Ordonnance*

Dans la base de données on conserve tous les ordonnances de chaque patient mais on suppose cependant qu'une ordonnance par jour ne soit nécessaire. Il peut bien arriver qu'après le contrôle du malade par son médecin, ce dernier décide qu'il suffit de suivre l'ordonnance du jours d'avant et qu'aucune nouvelle prescription de médicament n'est prévue.

### *Instruction*

La relation pour chaque prescription de l'ordonnance. La posologie peut être « voie cutanée », « voie orale » etc. lorsque le dosage peut être « 1 comprimé par jour », « 1 fois le matin 1 fois le soir avant repas » etc. On négligera les instructions « dynamiques » genre « prenez 3 fois par jour pendant une semaine, ensuite si vous vous sentez mieux continuez 1 fois par jour pendant 3 semaines », ce qui est de toute façon très rare dans un hôpital me semble-t-il.



### 3 Schéma conceptuel

#### Les relations

batiment(idBatiment)  
 service(idService, libelle, tel, batiment, chef)  
 chambre(idChambre, nombreLits, service)  
 medecin(matricule, nom prenom, specialite)  
 malade(idMalade, nom, prenom, sexe, age, contreIndication, chambre)  
 ordonnance(idOrdonnance, datePrescription, malade, medecin)  
 medicament(idMedicament, libelle)  
 instruction(ordonnance, medicament, posologie, dosage, duree, destockage, motifRefus)

#### Les contraintes d'intégrités

Selon de différentes taxonomies il peut y avoir 4 à 7 types de contraintes d'intégrités en base de données relationnelle. Ici on distingue les CI d'entité (dites entity integrity en anglais), les CI référentielles, les CI de champ (dites domain integrity en anglais) et d'autres CI. Comme les CI d'entité de notre base, ce qui n'est qu'un terme théorique pour dire qui est la clé primaire, ont été toutes soulignées dans leur relation plus haut, on commence donc ici par les CI référentielles qui sont les suivantes :

$\text{proj}_{\text{batiment}}(\text{service}) \subseteq \text{proj}_{\text{idBatiment}}(\text{batiment})$   
 $\text{proj}_{\text{service}}(\text{chambre}) \subseteq \text{proj}_{\text{idService}}(\text{service})$   
 $\text{proj}_{\text{specialite}}(\text{medecin}) \subseteq \text{proj}_{\text{idService}}(\text{service})$   
 $\text{proj}_{\text{chef}}(\text{service}) \subseteq \text{proj}_{\text{matricule}}(\text{medecin} \mid \text{specialite} = \text{proj}_{\text{idService}}(\text{service}))$   
 $\text{proj}_{\text{chambre}}(\text{malade}) \subseteq \text{proj}_{\text{idChambre}}(\text{chambre})$   
 $\text{proj}_{\text{malade}}(\text{ordonnance}) \subseteq \text{proj}_{\text{idMalade}}(\text{malade})$   
 $\text{proj}_{\text{medecin}}(\text{ordonnance}) \subseteq \text{proj}_{\text{matricule}}(\text{medecin})$   
 $\text{proj}_{\text{ordonnance}}(\text{instruction}) \subseteq \text{proj}_{\text{idOrdonnance}}(\text{ordonnance})$   
 $\text{proj}_{\text{medicament}}(\text{instruction}) \subseteq \text{proj}_{\text{idMedicament}}(\text{medicament})$

et on trouvera surtout les 3 CI de champ suivantes :

$\text{proj}_{\text{nombreLits}}(\text{chambre}) \in \{1, 2, 3\}$   
 $\text{proj}_{\text{sexe}}(\text{malade}) \in \{\text{'male'}, \text{'female'}\}$   
 $\text{proj}_{\text{destockage}}(\text{instruction}) \in \{\text{NULL}, 0, 1\}.$

Toutes celles-ci peuvent être réalisées avec un **trigger**, voir la fin de mon fichier sql de la création de tables. Je ne vois d'ailleurs pas d'attributs dans notre problème, qui nécessitent d'interdire strictement de prendre la valeur NULL, à part les clés primaires, d'où ici il n'y pas de CI de ce genre. Cela étant dit, d'après l'énoncé il pourra y avoir d'autres CI qui n'appartiennent à aucune de ces trois catégories précédentes et sont légèrement plus sophistiquées, comme le nombre total de malades ne doit pas excéder 600, il ne doit pas y avoir plus de malades dans une chambre que son nombre de lits, ou encore un service médical n'a pas la capacité d'accueillir plus que 60 malades.

$\text{sum}(\text{malade}) \leq 600$   
 $\text{card}(\text{proj}_{\text{chambre}}(\text{malade} \mid \text{chambre} = i)) \leq \text{proj}_{\text{nombreLits}}(\text{chambre} \mid \text{idChambre} = i)$   
 pour tout  $i \in \text{proj}_{\text{idChambre}}(\text{chambre})$   
 $\text{sum}(\text{proj}_{\text{nombreLits}}(\text{chambre} \mid \text{service} = i)) \in [50, 60]$  pour tout  $i \in \text{proj}_{\text{idService}}(\text{service})$   
 ...

#### Les relations sont-elles toutes en 3<sup>e</sup> forme normale ?

Oui toutes les tables sauf **instruction** sont en 3<sup>e</sup> forme normale. Dans la relation **instruction** il y a une dépendance indirecte : **motifRefus** dépend d'abord de **destockage** qui dépend à son tour de la clé de la relation, car quand le médicament est déstocké il n'y a pas de motif de refus. Mais dans la pratique, il me paraît inimaginable d'établir une autre relation juste pour conserver le motif de refus (l'autre façon de la transformer en 3FN c'est d'enlever le status de déstockage et qu'on met toute de suite un motif de refus lorsque le médicament ne peut pas être déstocké. Mais ceci a aussi un désavantage : si le pharmacien a refusé le déstockage et qu'il ne sait quoi mettre comme motif...).