

ISPED-SITIS

Master Santé publique 2e année

PROJET INF201

2022/2023

## **PROJET : CONCEPTION ET DÉVELOPPEMENT D'UN OUTIL POUR INTERROGER DES BASES DE DONNÉES**

AMOUSSOU-GUENOU Gwladys

Projet à rendre le 16 /10 /2022



## 1. Contexte

---

Le programme de médicalisation des systèmes d'information (PMSI) est un programme qui permet de décrire de façon synthétique et standardisée l'activité médicale des établissements de santé. Ce programme repose sur l'enregistrement de données médico-administratives normalisées dans un recueil standard d'information. Actuellement ce programme est un outil de description et de mesure médico-économique de l'information. Les étudiants du Master 2 de du Système d'information et technologie informatique pour la Santé (SITIS) de l'institut de Santé publique, d'Epidémiologie et de Développement (ISPED) de bordeaux sont formés sur les systèmes d'information et l'une de leurs compétences est de créer des logiciels et applications adapter pour les problèmes de santé. Etant l'une de ces étudiants, je viens vous présenter mon application (**PMSI-INFO\_PATIENT**) qui est un outil permettant de connaître les informations sur les patients enregistrés dans la base de données de ce programme PMSI



## 2. DESCRIPTION DE L'APPLICATION

### 2.1. Base de données

#### 2.1.1. DBeaver

DBeaver est un logiciel permettant l'admission et le requêtage de base de données pour les données relationnelles et intègre plusieurs services de base de données tels qu'Oracle MySQL et MariaDB. C'est un driver qui permet de manipuler les données d'une base et de créer des applications à travers le logiciel Java. Java est en effet le premier langage de programmation et la première plate-forme de développement. Pour réaliser notre projet, on a utilisé le service de base de données MariaDB est le logiciel JAVA.

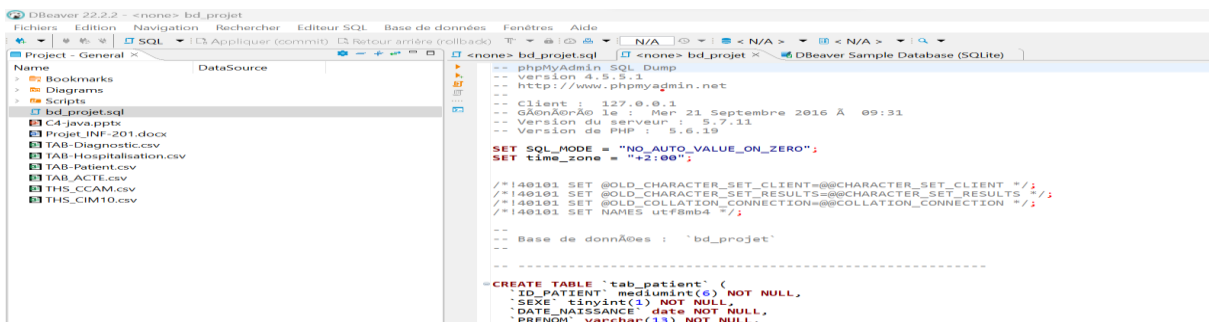
#### 2.1.2. Base PMSI

La base de données PMSI contenant 6 tables ont été importées dans le Driver DBeaver. Il s'agissait des tables suivantes : Patients, hospitalisation, diagnostique, Actes, Thésaurus CIM 10 et Thésaurus CCAM.

##### 2.1.2.1. Importation des données PMSI

Pour importer la base de données PMSI, les étapes suivantes ont été effectuées :

##### ➤ Installation de DBeaver



##### ➤ Création d'une nouvelle connexion my SQL



➤ Importation de la base PMSI

Requêtes réalisées : Les requêtes utilisées pour les scénarios ont toutes été testées dans DBEAVER avant importation dans JAVA.

### 2.1.3 Application dans JAVA

La connexion de Java avec Maria DB ont été réalisé avec le driver MySQL Connector.jar

Pour réaliser l'application Java, les étapes suivantes ont été réalisés

➤ Création de Packages et des Classes suivantes :

- com.pmsi
- pmsi

Le premier package (com.pmsi), contient quatre (04) classes :

- **DatabaseManager** : c'est une classe qui permet de lier les identifiants de connexion à une base de données et d'exécuter et de renvoyer les résultats des requêtes faites par l'utilisateur à travers l'application
- **DateManager** : toutes les classes qui ont besoin de formater une date, ont hérité de cette classe qui fournit une instance de datetime.
- **Thésaurus** : cette classe a été implémentée afin de s'assurer que chaque thésaurus, définit et réceptionne « un code » et « un label »



- **Utils** : cette classe permet de formater les résultats présentés sous forme de tableau et d'avoir un visuel propre.

➤ Création des classes suivants

- **Acte** : entité du projet représentant une ligne de la tab\_acte en base de données
- **Application** : contient toutes les requêtes des différents scénarios mis en place dans le cadre du projet
- **Diagnostic** : entité du projet représentant une ligne de la tab\_diagnostic en base de données  
Hospitalisation : entité du projet représentant une ligne de la tab\_hospitalisation en base de données
- **Patient** : entité du projet représentant une ligne de la tab\_patient en base de données
- **ThesaurusCCAM** : entité du projet représentant une ligne de la tab\_ccam en base de données
- **ThesaurusCIM10** : entité du projet représentant une ligne de la tab\_cim10 en base de données



### 3. Fonctionnement de l'application

---

#### ***A- Fichier configuration :***

Pour se connecter à la base de données, il faut configurer les informations de connexion du fichier « conf.properties ».

#### ***B- Ouverture de l'application***

##### ***- Fonctionnement après ouverture***

Ici l'utilisateur est celui qui lance et qui arrête l'application.

Pour passer d'une requête à une autre il faut répondre à la question « Voulez-vous continuer » par « oui : o »

Pour quitter l'application taper « non : n »



#### 4. Liste des scénarios

---

Neuf scénarios ont été testés :

1. Liste des patients et leur date d'entrée à l'hôpital
2. Liste des hospitalisations et des actes liés à un patient
3. Nombre d'hospitalisation selon le type de diagnostic
4. Liste des patients qui avaient moins de 60 ans le jour de leur hospitalisation
5. Liste des hospitalisations et des diagnostics liés à un patient
6. Âge moyen des patients selon leur sexe

Pour avoir les résultats liés à chaque scénario, il faut lancer l'application.

Ensuite « Faire un choix »

**1** : Il faut fournir l'ID du patient recherché afin d'avoir les informations liées à son (ses) hospitalisation(s). Exemple utilisé : 100130

**2** : Il faut fournir l'ID du patient afin d'obtenir les informations désirées à savoir : l'id d'hospitalisation, son nom, prénom et les informations liées à l'acte. Exemple utilisé :527230

**3** : Il faut fournir le type de diagnostic recherché (S, R ou D) afin d'obtenir le nombre d'hospitalisation enregistré selon le type de diagnostic.

**4** : Il faut fournir un numéro lié au sexe des patients (1 : Homme, 2 : Femme)

**5** : Il faut fournir l'ID du patient afin d'obtenir les informations désirées à savoir : l'id d'hospitalisation, son nom, prénom et les informations liées à son diagnostic. Exemple utilisé : 100130 (Code ne fonctionne pas)

**6** : Il faut fournir un numéro lié au sexe des patients (1 : Homme, 2 : Femme)



## 5. Création d'une interface graphique

---

Après la réalisation du projet, j'ai commencé la création d'une interface graphique, mais je n'ai pas terminé. Je me suis arrêté à la première interface. Veuillez trouver ci-joint les codes et les deux classes que j'ai créé.

Pour créer l'interface, j'ai dû utiliser une autre méthode carrément différente du processus utiliser pour réaliser le projet. l'interface s'ouvre avec la console de JAVA. C'est un module JFRAME avec pour l'instant la création de deux classes. La classe CONNECTION et la classe UTILISATEUR.

- PLAN DE L'interface
- INTERFACE UTILISATEUR
- INTERFACE MENUE :
  - INFO\_PATIENT
  - INFO\_HOSPITALISATION
  - INFO\_DIAGNOSTIC
  - INFO\_ACTES

PREMIERE INTERFACE DE L'APPLICATION





```
JPasswordField Password;  
Login = txtlogin.getText();  
Password = passwordpassword;
```

PMSI

## UTILISATEUR

Login

Password

[Forget your password?](#)

[ENREGISTRER](#) [Connection](#)



## 6. Difficulté rencontrée

---

Interface : L'ai eu beaucoup de difficulté à :

établir la création entre la le JFRAME (Utilisateur) et la base de données

Gérer les erreurs de codes pour que l'application s'affiche et se connecter à la base.

J'ai réalisé au début 12 requêtes mais je n'ai pas pu les intégrer dans l'application du projet comme au départ prévu

Parmi toutes les requêtes proposées, la requête 5 refuse de s'afficher.



## 7 Point positif et conclusion

---

### ➤ Point Fort

Je me suis beaucoup Appliquée et je comprends mieux le langage JAVA avec les difficultés que j'ai rencontrés

J'ai aussi appris à créer une interface graphique avec comme Back-end le logiciel Java

### ➤ Conclusion

Ce projet à été un gros exercice avec lequel j'ai mieux compris le langage Java et appliquer les explications du cours.



