

Fiche de TP no 5

Il existe plusieurs outils de modélisation de base de données, (MysqlWorkbench1, PowerAMC, WinDesign2, AnalyseSI, DBMain3, etc). Dans le cadre du TP (et du contrôle continu qui suivra), vous allez découvrir l'outil **JMerise** qui existe en version étudiant gratuite pour la modélisation des modèles conceptuels des données.

Pour installer **JMerise**, vous allez vous rendre à l'adresse suivante :

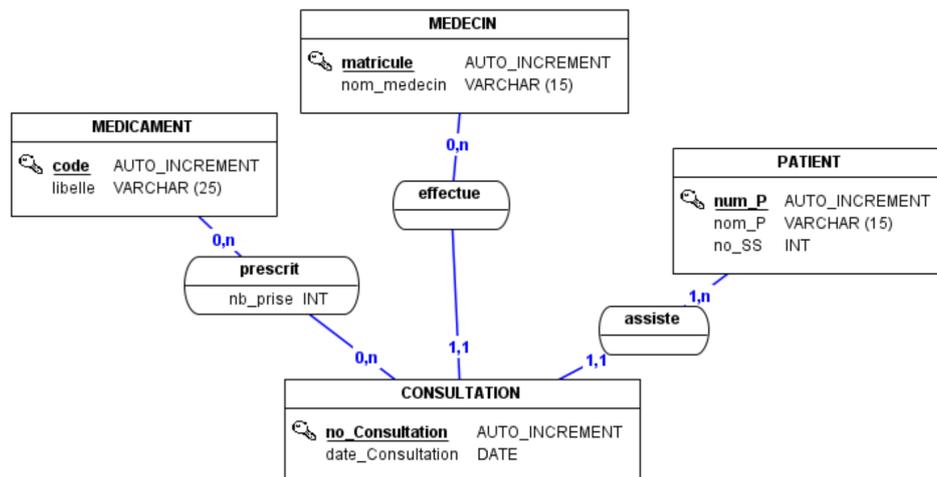
<http://www.jfreesoft.com/JMerise/>

et télécharger **JMerise.zip**. N'oubliez pas de suivre les conditions d'activation décrites sur le site. Pour lancer l'outil **JMerise** :

- décompressez l'archive
- lancer le terminal, rendez vous dans le bon répertoire (en tapant `cd nomduRépertoire` dans lequel se trouve **JMerise**)
- tapez la commande suivante :

```
java -jar JMerise.jar
```

Exercice 1. On vous donne un schéma E/A représentant des visites dans un centre médical. Vous allez saisir le dictionnaire des données et ce diagramme sous **JMerise** (Cliquer un nouveau diagramme MCD que vous nommerez `.mcd`).



1. Vérifiez votre MCD en cliquant sur 
2. Proposez une modification du schéma précédent en prenant en considération les besoins suivants :

Les patients de l'hôpital sont répartis dans des services (caractérisés chacun par un nom identifiant, sa localisation, sa spécialisation). Chaque médecin appartient à un service. Lors d'une consultation, une ou plusieurs prescriptions peuvent être rédigées. Chaque prescription mentionne le nom du médicament et la fréquence de prise d'un médicament. Il est évident que le médecin ne prescrit pas deux fois le même médicament lors d'une même consultation.

3. Sauvegardez votre modèle, vérifiez-le, et exportez-le en format graphique (fichier `exo1.png`).
4. Générez le modèle relationnel.
5. Indiquez les différences entre le modèle conceptuel et le modèle relationnel.
6. Expliquez comment les clés étrangères ont été générées.
7. Vous remarquerez que le script SQL de création des tables a été également généré. Le système de gestion de base de données par défaut est MySQL. Changez de système cible en générant un pour SQLite.

8. Enregistrez le script généré pour le SGBD SQLite sous le nom de «SQLiteScript.sql ». Consultez le script généré.
9. Exécutez le script SQL généré.
10. Après la création de vos tables, vous aurez besoin de les remplir. Rédigez le script SQL d'insertion d'un jeu d'essai. Vous pouvez générer un script d'insertion en vous rendant sur le site generatedata :

<http://www.generatedata.com/?lang=fr>.

Exécutez le script d'insertion.

Exercice 2. Pour rappel, voici le texte depuis le dernier exercice du TP de la semaine dernière :

Une personne peut ouvrir un ou plusieurs comptes dans une agence bancaire. Elle doit donner son nom, son adresse, et éventuellement son numéro de téléphone. Chaque compte a une seule date d'ouverture, un numéro d'identification de compte, et un type (compte chèques, compte d'épargne, etc). Chaque compte appartient à un seul titulaire. Chaque agence bancaire a un nom, un numéro d'agence, une adresse et un numéro de téléphone. Une opération est effectuée par une seule personne (le titulaire du compte ou un tiers) et concerne au plus deux comptes : le compte émettant et le compte recevant l'opération. Une opération a un numéro d'identification, une date, un lieu, un type qui indique la nature de l'opération : retrait par carte bancaire, versement de salaire, encaissement ou paiement de chèque, virement de compte à compte, etc. Le solde d'un compte est établi à une date précise, au début de chaque mois. Si le solde est négatif, alors une lettre sera adressée au titulaire du compte.

Nous avons développé, en TD, un modèle E/A du domaine décrit par le texte.

1. Saisissez le modèle dans JMerise (nouveau diagramme MCD nommée `exo2.mcd`) et vérifiez-le.
2. Exportez le modèle en format graphique (nom du fichier `exo2.png`)
3. Créez le modèle relationnel et sauvegardez le script pour engendrer les tables (nom du fichier : `exo2_schema.sql`)
4. Générez un script pour insérer un jeu d'essai (nom du fichier : `exo2_insertion.sql`)