Rapport de Projet

Electif : Base de Données 2

Eva Cheng – Antonin Boulnois

EFREI paris  Groupe 2

# Rapport de Projet de Bases de Données 2

## Introduction

Selon vous, comment pourrait-on optimiser le temps passé par les médecins sur la partie administrative et logistique afin de leur permettre de se concentrer sur le corps de métier ? Engager un secrétaire ? Cette option pourrait s’avérer coûteuse sur le long terme. Prendre un carnet de note pour noter ? Quant à celle-ci, nous pouvons dire que sur le long terme, en plus de ne pas être très écologique, pourrait facilement nous perdre. Numériser la chose ? Ce serait la meilleure des solutions surtout si nous stockons tout cela dans une base de données, qui pour rappel est une implémentation organisée, concise, structurée, cohérente et indexée des données.

Dans le cadre de nos électifs d’informatique du 6e semestre à l’EFREI Paris, et plus particulièrement le module de base de données, nous avons été confrontés à un problème auquel nous devrons répondre par un projet réalisé avec nos connaissances en bases de données.

Le problème qui nous a été posé est celui d’une psychologue qui n’est pas en mesure de pouvoir retenir les données de ses patients et ses propres rendez-vous, ce qui peut s’avérer être problématique dans son quotidien.

Les fonctionnalités principales et primordiales qui sont l’intérêt même de notre base de données sont la visibilité des rendez-vous par la psychologue et par le patient.

En partant de cette base, nous avons pu également implémenter d’autres fonctionnalités aussi pertinentes que nous vous détaillerons dans une prochaine partie.

## Dictionnaire de données

|  |  |  |
| --- | --- | --- |
| **Intitulé du champ** | **Nom du champ** | **Type de données** |
| Identifiant du moyen de paiement | ID\_Moyen | Entier généré par le système, jamais NULL |
| Intitulé du moyen de paiement | Titre | Texte, jamais NULL |
| Identifiant d’un patient  Idem  Cas de 2 patients  Cas de 3 patients | ID\_Patient  ID\_Patient1  ID\_Patient2  ID\_Patient3 | Entier, jamais NULL  Entier, jamais NULL  Entier, peut être NULL  Entier, peut être NULL |
| Nom du patient | Nom | Texte, jamais NULL |
| Prénom du patient | Prenom | Texte, jamais NULL |
| Numéro de téléphone du patient | Numero\_De\_Telephone | Entier, peut être NULL |
| Email du patient | Email | Texte, jamais NULL |
| Mot de passe du patient | Mot\_De\_Passe | Texte, jamais NULL |
| Adresse du patient | Adresse | Texte, jamais NULL |
| Ville de résidence | Ville | Texte, jamais NULL |
| Code Postal de la ville | CodePostal | Entier, jamais NULL |
| Identifiant de la classication | ID\_Genre | Entier, jamais NULL, généré |
| Intitulé de la classification | Titre | Texte, jamais NULL |
| Date de début de quelque chose | Date\_Debut | Date, jamais NULL |
| Date de fin de quelque chose | Date\_Fin | Date, jamais NULL |
| Identifiant de la profession | ID\_Profession | Entier, jamais NULL, généré |
| Intitulé de la profession | Titre | Texte, jamais NULL |
| Identifiant de la consultation | ID\_Consultation | Entier, jamais NULL, généré |
| Date de début de la consultation | Date\_Debut\_RDV | Date, jamais NULL |
| Date de fin de la consultation | Fin\_RDV | Date, jamais NULL, date de début + 30 min |
| Comportements durant la consultation | Comportements | Texte, peut être NULL |
| Gestes durant la consultation | Gestes | Texte, peut être NULL |
| Postures durant la consultation | Postures | Texte, peut être NULL |
| Note d’Anxiété du patient | Note\_Anxiete | Décimal, peut être NULL |
| Identifiant de la facture | Code\_Facture  ID\_Facture | Entier, jamais NULL, généré |
| Montant de la facture | Prix\_En\_Euros | Décimal, jamais NULL |
| Moyen de paiement | MoyenPaiement | Texte, peut être NULL |
| Etat du paiement | StatutPaye | Entier, jamais NULL, 0 si non payé et 1 si payé |

Et dans le détail nous avons cela :

### Moyen

|  |  |  |
| --- | --- | --- |
| **Nom de variable** | **Type (requis si not NULL)** | **Clé ?** |
| ID\_Moyen | INT NOT NULL | Oui, clé primaire |
| Titre | VARCHAR(15) NOT NULL | Non |

### Patient

|  |  |  |
| --- | --- | --- |
| **Nom de variable** | **Type (requis si not NULL)** | **Clé ?** |
| ID\_Patient | INT NOT NULL | Oui, clé primaire |
| Nom | VARCHAR(25) NOT NULL | Non |
| Prenom | VARCHAR(25) NOT NULL | Non |
| Email | VARCHAR(50) NOT NULL | Non |
| Numero\_De\_Telephone | INT | Non |
| Mot\_De\_Passe | VARCHAR(50) NOT NULL | Non |
| ID\_Moyen | INT NOT NULL | Oui, clé étrangère de Moyen |
| Adresse | VARCHAR(50) NOT NULL | Non |
| Ville | VARCHAR(25) NOT NULL | Non |
| CodePostal | INT NOT NULL | Non |

### Genre

|  |  |  |
| --- | --- | --- |
| **Nom de variable** | **Type (requis si not NULL)** | **Clé ?** |
| ID\_Genre | INT NOT NULL | Oui, clé primaire |
| Titre | VARCHAR(10) | Non |

### Classification\_Patient

|  |  |  |
| --- | --- | --- |
| **Nom de variable** | **Type (requis si not NULL)** | **Clé ?** |
| ID\_Genre | INT NOT NULL | Oui, clé composite + étrangère |
| ID\_Patient1 | INT NOT NULL | Oui, clé composite + étrangère |
| ID\_Patient2 | INT | Oui, clé étrangère |
| Date\_Debut | DATE NOT NULL | Non |
| Dat\_Fin | DATE NOT NULL | Non |

### Profession

|  |  |  |
| --- | --- | --- |
| **Nom de variable** | **Type (requis si not NULL)** | **Clé ?** |
| ID\_Profession | INT NOT NULL | Oui, clé primaire |
| Titre | VARCHAR(30) NOT NULL | Non |

### Profession\_Patient

|  |  |  |
| --- | --- | --- |
| **Nom de variable** | **Type (requis si not NULL)** | **Clé ?** |
| ID\_Patient | INT NOT NULL | Oui, clé composite + étrangère |
| ID\_Profession | INT NOT NULL | Oui, clé composite + étrangère |
| Date\_Debut | DATE NOT NULL | Non |
| Date\_Fin | DATE | Non |

### Consultation

|  |  |  |
| --- | --- | --- |
| **Nom de variable** | **Type (requis si not NULL)** | **Clé ?** |
| ID\_Consultation | INT NOT NULL | Oui, clé primaire |
| Date\_Debut\_RDV | DATE NOT NULL | Non |
| Fin\_RDV | DATE NOT NULL | Non |

### Type\_Consutation

|  |  |  |
| --- | --- | --- |
| **Nom de variable** | **Type (requis si not NULL)** | **Clé ?** |
| ID\_Type | INT NOT NULL | Oui, clé primaire |
| Titre | VARCHAR(30) NOT NULL | Non |

### Consultation\_Patient

|  |  |  |
| --- | --- | --- |
| **Nom de variable** | **Type (requis si not NULL)** | **Clé ?** |
| ID\_Patient1 | INT NOT NULL | Oui, clé composite + étrangère |
| ID\_Patient2 | INT | Oui, clé étrangère |
| ID\_Patient3 | INT | Oui, clé étrangère |
| ID\_Type | INT NOT NULL | Oui, clé composite + étrangère |
| ID\_Consultation | INT NOT NULL | Oui, clé composite + étrangère |
| Comportements | VARCHAR(200) | Non |
| Gestes | VARCHAR(255) | Non |
| Postures | VARCHAR(255) | Non |
| Note\_Anxiete | DECIMAL | Non |

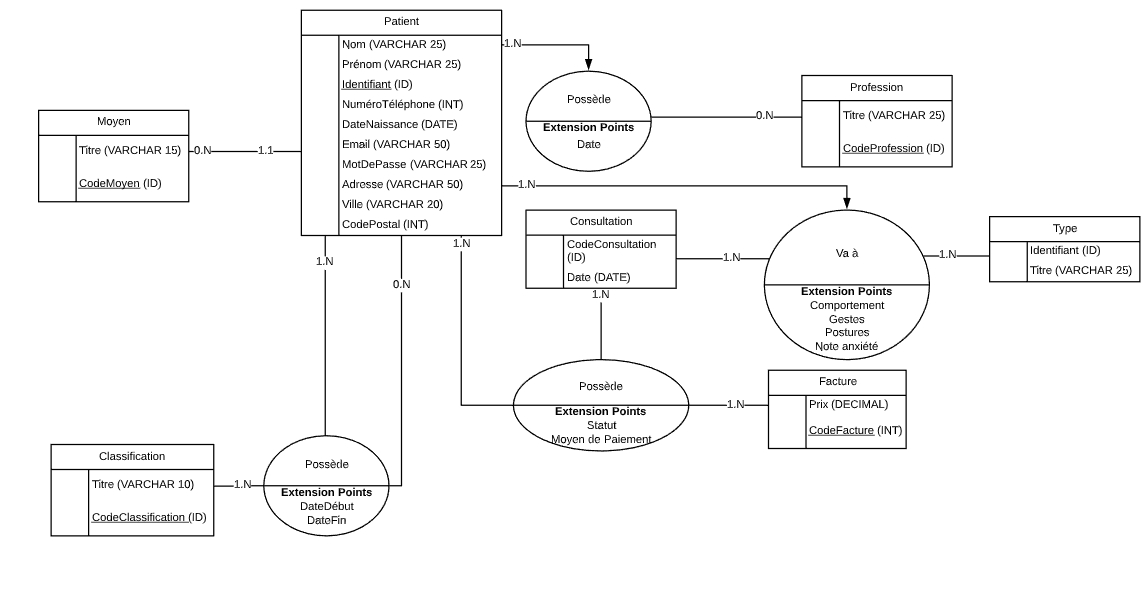
### Facture

|  |  |  |
| --- | --- | --- |
| **Nom de variable** | **Type (requis si not NULL)** | **Clé ?** |
| CodeFacture | INT NOT NULL | Oui, clé primaire |
| Prix\_En\_Euros | DECIMAL NOT NULL | Non |

### Paiement

|  |  |  |
| --- | --- | --- |
| **Nom de variable** | **Type (requis si not NULL)** | **Clé ?** |
| ID\_Patient | INT NOT NULL | Oui, clé étrangère |
| ID\_Consultation | INT NOT NULL | Oui, clé étrangère |
| ID\_Facture | INT NOT NULL | Oui, clé étrangère |
| StatutPaye | INT NOT NULL | Non |
| MoyenPaiment | VARCHAR(25) NOT NULL | Non |

## Réalisation du Modèle Conceptuel

Avant de procéder à la création de notre base de données sur machine, nous avons commencé à établir un modèle conceptuel pour pouvoir simplifier notre création de base ultérieure.

Dans le détail, nous avons l’entité principale autour de laquelle va tourner toutes les autres fonctionnalités. Le patient possède un nom, un prénom, un identifiant qui lui est unique et qui lui servira pour faire marcher les autres fonctionnalités, un numéro de téléphone pour que la psychologue puisse appeler les patients, sa date de naissance, son email pour lui permettre d’accéder au site, idem pour le mot de passe attribué par la psychologue, une adresse, une ville et un code postal (pour respecter la 1re forme normale).

Ensuite, chaque patient possède une profession, nous avons donc une entité profession à laquelle est rattachée l’entité patient par une relation qui donne la date étant donné qu’un patient peut avoir plusieurs professions et que différents patients peuvent avoir une même profession, donc il faut dater.

Ensuite nous avons une entité consultation qui donne la date de consultation ainsi qu’un identifiant pour que celle-ci soit unique. Pour éviter les redondances, nous avons établi une table type qui définit le type de consultation (Anxiété, Stress, Dépression) avec encore une fois un identifiant.

Ces deux entités sont reliées à l’entité patient par la relation ConsultationPatient qui prendra en compte les comportements que la psychologue pourra noter, les gestes, les postures et éventuellement la note d’anxiété si nécessaire. Cette relation mettra en lien 1 à 3 patients car 3 peuvent être présents, une consultation et un type de consultation. Etant donné qu’un patient peut avoir plusieurs consultations, une consultation plusieurs patients et types.

Pour suivre, nous savons qu’à la fin d’une consultation ou même dans le cas où un patient prend un forfait pour plusieurs consultations, nous avons une facture avec un prix. Cette facture peut être rattachée à plusieurs consultation et plusieurs patient (dans le cas où le patient choisi de payer pour les autres), la consultation peut avoir plusieurs factures (dans le cas où les patients choisissent de payer séparément) et plusieurs patients. Chaque patient peut avoir plusieurs factures et bien sûr plusieurs consultations. Donc nous en arrivons à une relation reliant les trois entités avec bien sûr le statut de paiement (payé, impayé) et le moyen de paiement.

De plus, chaque patient possède un moyen de découverte de la psychologue, pour respecter la 2e forme normale, on a choisi de créer une entité « moyen de découverte » qui sera reliée au patient par une relation 1.N :1.1.

Pour finir, nous avons l’entité classification pour respecter les 3 formes normales, femme, homme, enfant, marié. Supposant qu’une femme peut être à la fois femme et mariée et que nous voulons tenir compte du fait que nous voulons savoir qui est son mari, on aura une relation de l’entité patient avec l’entité patient et l’entité classification prenant en compte la date de début / fin dans le cas où un enfant n’est plus un enfant ou dans le cas où un couple n’est plus marié ou qu’il se marie.

On peut donc à partir de cela établir un modèle logique.

## Modèle Logique

En tenant compte du fait que les relations ...1 : …N deviennent des éléments clefs, c’est-à-dire des clés étrangères et que les relation …N : …N deviennent des entités, on réussit à établir un modèle relationnel logique à partir du modèle conceptuel précédent fait sur luccidchart.

On a donc :

MoyenDeDécouverte(IDMoyen, Titre)

Patient(IDPatient, Nom, Prénom, Email, NuméroDeTéléphone, MotDePasse, DateDeNaissance, Adresse, Ville, CodePostal, **#IDMoyen**)

Genre(IDGenre, Titre)

ClassificationPatient(**#IDGenre**, **#IDPatient1**, **#IDPatient2**, DateDéut, DateFin)

Profession(IDProfession, Titre)

ProfessionPatient(**#IDPatient**, **#IDProfession**, DateDébut, DateFin)

Consultation(IDConsultation, DateDébutRDV, DateFinRDV)

Type(IDType, Titre)

ConsultationPatient(**#IDConsultation**, **#IDType**, **#IDPatient1**, **#IDPatient2**, **#IDPatient3**, Comportements, Gestes, Postures, NoteAnxiété)

Facture(IDFacture, Prix)

Paiement(**#IDPatient**, **#IDConsultation**, **#IDFacture**, Statut, MoyenDePaiement)

Nous pourrons remarquer que les IDPatient2 et IDPatient3 ne figurent pas dans les éléments de la clé primaire étant donné qu’ils peuvent le pas avoir de valeur, si le patient décide de venir tout seul ou si celui-ci est célibataire.

## Implémentation de base de données

Pour créer la base de données, nous avons commencé par créer un utilisateur pour la psychologue à qui on a octroyé tous les privilèges, avec la commande « CREATE USER Psycho IDENTIFIED BY psycho123; GRANT ALL PRIVILEGES TO Psycho; », on se connecte au compte de la psychologue sur lequel nous créons la base de données.

Pour créer la base de données, il nous suffit de transformer notre MLD en langage SQL pour pouvoir générer des tables basées sur notre schéma conceptuel.

## Présentation des fonctionnalités implémentées

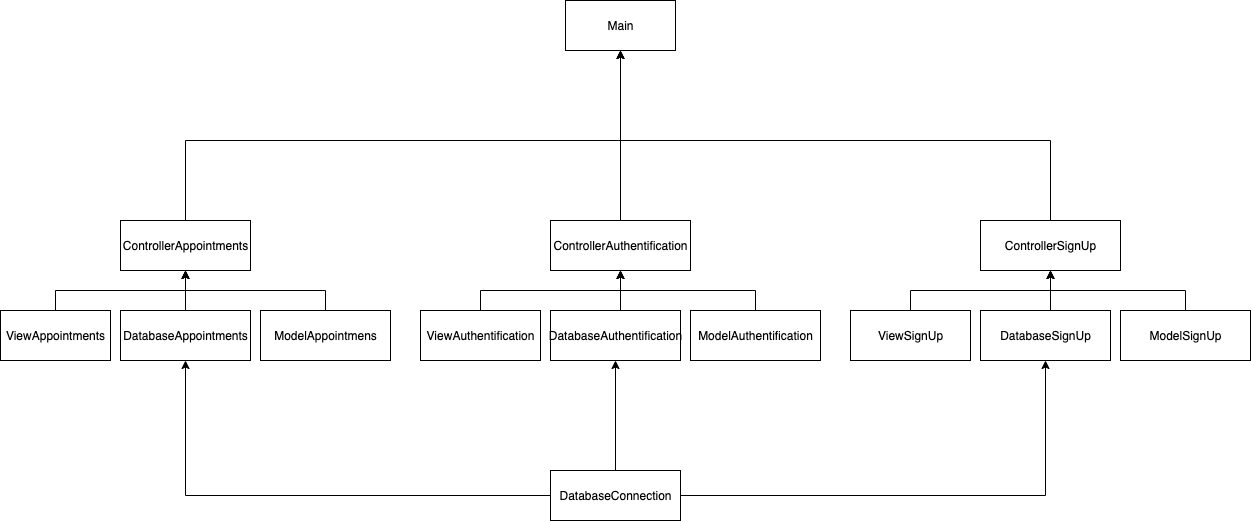
La base de données a été créée en SQL Oracle avec SQL Developer sur la machine virtuelle Oracle DB Developer et l’interface ainsi que certaines fonctionnalités en Java.

Grâce à notre code et interface en Java, nous gérons les erreurs des entrées utilisateurs et autres erreurs potentielles qui ne sont pas gérables en SQL. En d’autres termes, nous vérifions les entrées utilisateurs également appelées input.

Notre base de données implémente grâce à Java et SQL des requêtes préparées, des fonctions en SQL, des procédures ainsi que des vues.

De plus, pour faire ce projet, nous avons utilisé le concept « Modèle, Vue, Contrôleur » abordé au préalable dans ce module.

En voici un diagramme UML :



Passons maintenant aux fonctionnalités implémentées en Java et en SQL. Tout d’abord, nous pouvons ajouter un nouveau patient à la base de données, ce qui est primordial pour la psychologue.

Ensuite, un patient peut se connecter afin de pouvoir voir ses rendez-vous, son profil. Avec cette fonctionnalité, le patient rentre son email et son mot de passe, on vérifie en cherchant dans la base de données si nous avons un profil qui correspond à ce couple.

Pour suivre, nous pouvons prendre rendez-vous, ce rendez-vous prend en compte la date à laquelle ce rendez-vous veut être pris et pour quel patient (en utilisant le couple nom – prénom), bien sûr étant donné que les patients peuvent être jusqu’à 3, nous autorisons 3 couples nom prénom.

De plus, nous avons créé une vue qui permet l’affichage des rendez-vous passés et à suivre d’un patient (en comparant son id avec l’id présent dans la table consultation soit dans celui du premier patient, soit le deuxième soit le troisième) afin d’avoir entièrement les rendez-vous auxquels il a assisté ou alors qui l’attendent.

En toute logique, après avoir créé les rendez-vous, nous avons la création d’une facture et également la modification d’une facture dans le cas où le patient a payé sa facture après.

Pour suivre, dans le cahier des charges, un affichage des rendez-vous d’une semaine donnée nous était demandé, nous l’avons fait en créant une vue.

Ensuite, nous permettons aux personnes de modifier l’heure d’un rendez-vous en rentrant l’identifiant du rendez-vous et l’heure à laquelle nous souhaitons reporter les rendez-vous. Grâce à une procédure, nous pouvons gérer cela.

Etant donné que nous avons créé la table Facture et paiement de ce fait, nous voulons créer une vue sur les factures qui n’ont pas été payées. De fait nous avons créé une vue pour ce faire.

En voulant respecter le cahier des charges, nous voulions permettre de supprimer un rendez-vous (c’est-à-dire l’annuler), supprimer un patient d’un rendez-vous et également éventuellement ajouter un patient à une consultation. Pour cela, nous avons créé des procédures conditionnelles.

Et pour finir, une fois une consultation passée, nous voulions permettre à la psychologue de changer les données d’une consultation c’est-à-dire, rentrer les notes qui ont été prises, à savoir les gestes, comportements, postures et anxiété.

Donc, grâce à des requêtes SQL et des fonctions en Java, nous avons réussi à implémenter tout cela pour respecter notre cahier des charges.

La plupart des captures d’écran des fonctionnalités seront dans le scénario, sauf pour l’annulation d’un rendez-vous et la suppression d’un patient d’une consultation.

Annulation d’un rendez-vous :

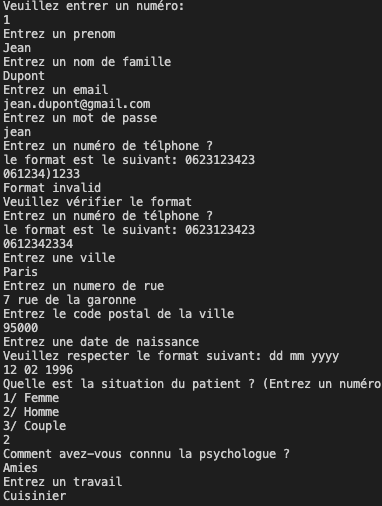


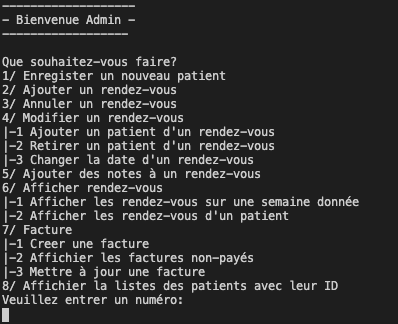
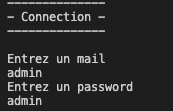
## Scénario

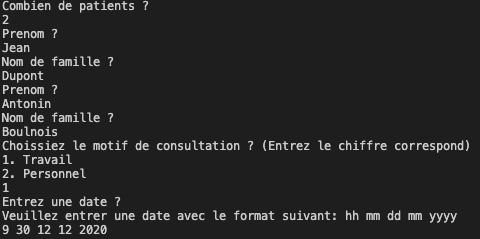
Pour essayer notre base de données, imaginons le scénario suivant :

Imaginons un homme du nom de Jean Dupont, il commence par faire son inscription auprès de la psychologue puis il décide de prendre rendez-vous.

Inscription :

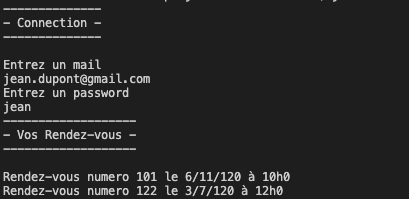


Interface administrateur de la psychologue :

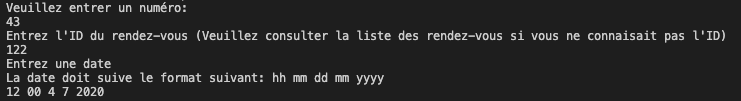
Prise de rendez-vous :

Monsieur Dupont plus tard consulte ses rendez-vous en ligne et s’aperçoit qu’il n’est pas disponible pour le prochain rendez-vous donc il décide d’appeler la psychologue pour changer l’heure du rendez-vous.

Affichage des rendez-vous (donc connexion) :



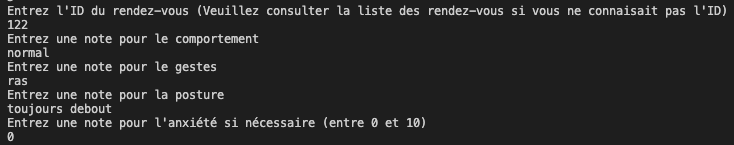
Modification de rendez-vous (retour sur l’interface de la psychologue, pour rappel, opition 4 et 3 puis) :



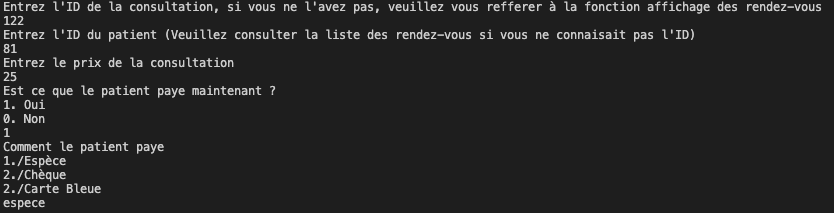
Ayant un ami qui voudrait également consulter avec Jean, il décide de lui propose de se rajouter au rendez-vous et donc la psychologue modifie.



Jean Dupont se rend au rendez-vous, la psychologue rempli les informations à savoir gestes comportements, postures etc.



Puis au final, Jean Dupont paie la facture.



Afficher la facture :



## Les difficultés rencontrées

Les difficultés rencontrées lors de la réalisation de ce projet étaient en grande partie dû à la charge de travail que nous avions en dehors du module de base de données. En vue de notre manque de temps, nous avons été contraints à abandonner certaines fonctionnalités (en dehors de celles du cahier des charges) .

De plus, ayant abordé le sujet différemment durant notre semestre à l’étranger et ayant étudié la matière sur des logiciel de gestion de base de données différents, nous avons dû nous réadapter à de nouvelles conventions.

Ensuite, en raison du confinement, nous n’étions pas en présentiel donc il était d’autant plus difficile pour notre professeur de nous venir en aide quand nous étions dans le besoin.

## Axes d’amélioration

Certaines fonctionnalités, comme mentionné précédemment, n’ont pas pu être implémentées. Ce sont les suivantes :

* Implémentation d’une interface graphique pour rendre l’expérience utilisateur plus ergonomique
* La question de pouvoir voir qui est marié avec qui même si on peut l’inscrire dans la base de données, nous n’avons pas d’affichage dessus
* Savoir dans le cas où le patient a découvert la psychologue par parrainage, qui est le patient qui l’a parrainé
* Nous n’avons pas implémenté tous les messages de confirmation comme quoi l’opération a bien été effectuée
* Dans le cas où l’opération n’a pas été effectuée, certes, il n’y a pas de problème mais l’utilisateur ne reçoit pas d’information comme quoi une erreur a été commise.
* Nous aurions pu créer des utilisateurs de type patient dans notre base de données avec uniquement comme privilège un select sur les choses qui le concerne

## Conclusion

Pour conclure, ce projet nous aura permis d’utiliser nos compétences acquises dans le module de bases de données pour mener à bien un projet qui réponds à une problématique qui aurait pu nous être posée dans la vie quotidienne.

Ce projet nous aura également permis de mettre en application les connaissances apprises en cours et ainsi de progresser grâce à cela.