

**INF3710 –Fichiers et Bases de données**

**Hiver 2019**

**TP No. 2**

**Groupe 4**

1340155 – Alassane Maiga

1862313 – Mohamed Esseddik BENYAHIA

**Soumis à : Manel Grinchi**

**2019-02-13**

**Réponse 1 : Le modèle relationnel (en syntaxe abstraite)**

**Patient(Patientno, Name, Adress, PhoneNo, Dob)**

**PRIMARY KEY (Patientno)**

Explications : Patient est une entitée forte dans le modèle nous pouvons lui assigner une clée primaire sans clef étrangère

**Payment(Paymentno, Details, Method, Patientno)**

**PRIMARY KEY (Paymentno, PatientNo)**

**FOREIGN KEY (PatientNo) REFERENCES Patient(PatientNo)**

Explications : Un patient peut faire un ou plusieurs payements mais un payement ne peut appartenir qu’à un patient. C’est une association binaire 1:\* Donc on doit inclure la clef primaire de patient comme clef étrangère dans la table payement.

**Appointment(Apptno, Date, Time, PatientNo, Doctorid)**

**PRIMARY KEY (Apptno, PatientNo, Doctorid)**

**FOREIGN KEY (PatientNo) REFERENCES Patient(Patientno)**

**FOREIGN KEY (Doctorid) REFERENCES Doctor(Doctorid)**

Explications : Même raison que précédemment, mais ici nous avons deux associations binaires 1:\* . Nous devons aussi inclure la clé primaire de docteur dans appointement

**Doctor(Doctorid, Name, Dob, Address, Phoneno, Salary)**

**PRIMARY KEY(Doctorid)**

Explications : Pas de clef étrangère ici. Entité forte.

**Medical(Doctorid, Overtime)**

**PRIMARY KEY (Doctorid)**

**FOREIGN KEY (Doctorid) REFERENCES Doctor(Doctorid)**

Explications : Ici medical est nécessairement un docteur. Nous devons donc inclure une clef étrangère de Docteur comme clef primaire de Medical

**Specialist(Doctorid, Fieldarea)**

**PRIMARY KEY (Doctorid)**

**FOREIGN KEY (Doctorid) REFERENCES Doctor(Doctorid)**

Explications : Même chose que précédemment

**Bill(Billno,Total, Doctorid)**

**PRIMARY KEY (Billno, Doctorid)**

**FOREIGN KEY (Doctorid) REFERENCES Doctor(Doctorid)**

Explications : Même situation que la relation appointement et payement. On inclue une clef primaire de doctor ici.

**BillPayment(Paymentno, PatientNo, Billno, Doctorid)**

**PRIMARY KEY (PaymentNo, PatientNo, BillNo, Doctorid)**

**FOREIGN KEY (PaymentNo, PatientNo) REFERENCES Payment(Paymentno, PatientNo)**

**FOREIGN KEY (BillNo, Doctorid) REFERENCES Bill(BillNo, Doctorid)**

Explications : Nous créons cette relation pour avoir une relation entre les payements et les factures. Nous avons une situation 1:\* --- 1:\*  entre payment et bill. Nous devons donc établir une paire de relation 1:\* --- 1:1.

**Question 2 : La base de données** medi

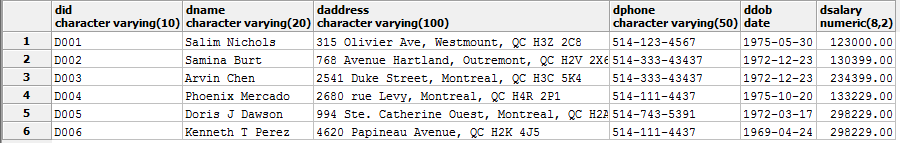
Voir le fichier *m*ed*i\_schema.sql*

**Question 3 : Les donnés**

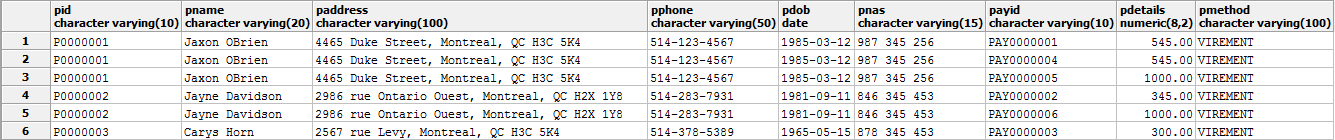
Voir le fichier *medi\_data.sql*

**Question 4 : Les requêtes**

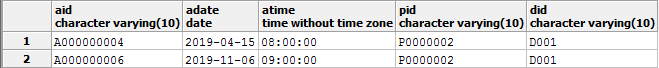
**4.a) Affichez tous les docteurs et leurs détails.**



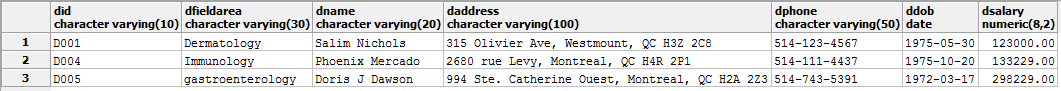
**4.b) Affichez les patients et leurs paiements.**



**4.c) Les rendez-vous du docteur dont le matricule est D001**



**4.d) Liste des spécialistes**



**4.e) Les nom des patients et leur dates de naissance**

