

# TP 6 – Langage SQL– Requêtes, sous-requêtes synchronisées, division, expressions et fonctions

Bases de Données – Licence 1 – Semestre 2

Comme dans le TP1, on considère le modèle relationnel d'un centre médical dont un schéma est donné ci-dessous.

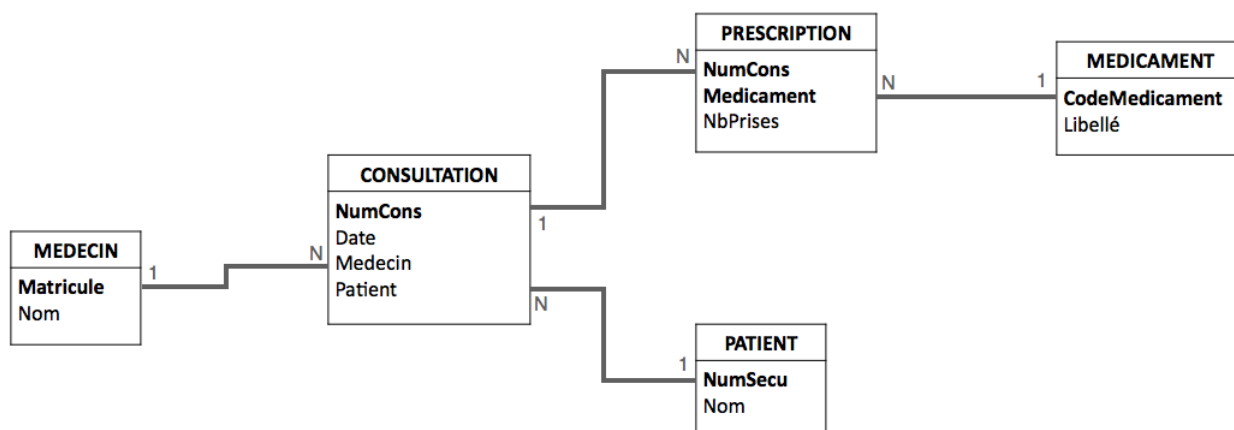


FIGURE 1 – Schéma relationnel du centre médical

## Exercice – Requêtes synchronisées

**Question 1.1.** NumSecu, Code et libellé des médicaments qui ont été prescrits à plusieurs reprises au ou à la même patient-e identifié-e par NumSecu. On affichera dans ce cas la date de la dernière prescription.

**Question 1.2.** Consultations (numéro, docteur-e, date) où le-la docteur-e a prescrit davantage de médicaments qu'il-elle n'en prescrit en moyenne.

## Exercice – Divisions

**Question 2.1.** Docteur-es qui ont effectué des consultations avec tous les patients.

**Question 2.2.** Patient-es qui ont goûté à tous les médicaments.

## Exercice – Expressions et fonctions

**Question 3.1.** Jour de la semaine où il y a le plus de consultations.

**Question 3.2.** Chiffre d'affaire de chaque docteur-e par année sachant que le tarif d'une consultation

— est de 25 euros si c'est un tarif 1 ;

— est de 30 euros si c'est un tarif 2.

L'attribut tarif a été ajouté à la table Medecin.

**Question 3.3.** Pour chaque patient-e, afficher son sexe à la naissance (en toutes lettres) et l'année de sa naissance. On pourra s'inspirer du code suivant pour créer une fonction puis l'utiliser dans une requête.

```
DELIMITER #
CREATE FUNCTION SEXE_JFC (NumSecu bigint(13))
RETURNS INT
BEGIN
DECLARE S INT DEFAULT 2;
SET S = LEFT(NumSecu, 1);
RETURN S;
END #

SELECT SEXE_JFC(NumSecu) FROM PATIENT WHERE 1
```

Attention, comme vous partagez votre base de données, vous risquez de définir la même fonction que votre voisin-e. Il est nécessaire d'avoir des noms différents. C'est pour cela que `_JFC` a été ajouté au nom de la fonction (à changer).

Pour supprimer une fonction déjà définie, il suffit de faire un `DROP` :

```
DROP FUNCTION IF EXISTS SEXE_JFC
```