

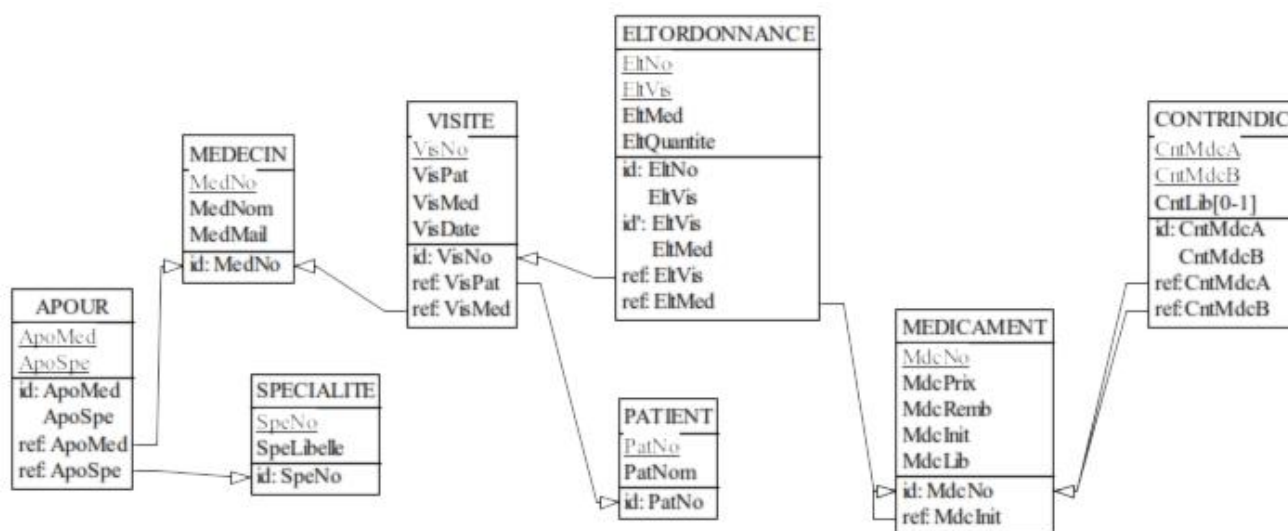
DON2 – Laboratoires DON2

TD5 et 6 – Jointure

Exercice 1

Compréhension de schéma

Voici un schéma conceptuel de base de données relatives à la gestion de prescriptions médicales.



Description

- ▷ **Medecin** : Un médecin susceptible de délivrer des ordonnances
 - ▷ MedNo : Num – Id. du médecin
 - ▷ MedNom : Varchar(100) – Nom du médecin
 - ▷ MedMail : Varchar(200) – Email du médecin
- ▷ **Patient** : Une personne ayant effectué au moins une visite chez un médecin
 - ▷ PatNo : Num – Id. du patient
 - ▷ PatNom : Varchar(100) – Nom du patient
- ▷ **Visite** : Une consultation d'un médecin par un patient
 - ▷ VisNo : Num – Id. de la visite
 - ▷ VisPat : Num – Id. du patient
 - ▷ VisMed : Num – Id. du médecin
 - ▷ VisDate : Date – Date de la visite
- ▷ **Specialite** : Une spécialité de médecine reconnue
 - ▷ SpeNo : Num – Id. de la spécialité

- ▷ **SpeLib** : Varchar(100) – Libellé de la spécialité
- ▷ **Apour** : Une spécialité d'un médecin
 - ▷ ApoMed : Num – Id. du médecin
 - ▷ ApoSpe : Num – Id de la spécialité
- ▷ **Médicament** : Un médicament prescriptible
 - ▷ MdcNo : Num – Id. du médicament
 - ▷ MdcLib : Varchar(100) – Libellé du médicament
 - ▷ MdcPrix : Num – Prix plein du médicament
 - ▷ MdcRemb : Num – Montant remboursé
 - ▷ MdcInit : Num – Id. du médicament original (null s'il s'agit d'un original, obligatoire s'il s'agit d'un générique)
- ▷ **Contrindic** : Une contre-indication entre deux médicaments originaux
 - ▷ CntMdcA : Num – Id d'un médicament
 - ▷ CntMdcB : Num – Id d'un médicament
 - ▷ CntLib : Varchar(500) – Libellé de la contre-indication
- ▷ **EltOrdonnance** : Prescription d'un médicament lors d'une visite (l'ensemble des EltOrdonnance d'une visite constitue une ordonnance)
 - ▷ EltNo : Num – Numéro d'ordre de la prescription pour la visite
 - ▷ EltVis : Num – Id de la visite
 - ▷ EltMed : Num – Id du médicament prescrit
 - ▷ EltQuantite : Num – Quantité prescrite

Remarques :

- ▷ Un médicament générique est la copie exacte d'un médicament original.
- ▷ Si deux médicaments originaux (non génériques) A et B sont contre-indiqués, la table Contrindic reprendra les tuples (A, B, lib) et le tuple (B, A, lib). Pour savoir si deux médicaments sont contre-indiqués, il faut voir si leurs originaux sont contre-indiqués.

Questions

1. Donnez tous les identifiants primaires et toutes les clés étrangères de ce schéma.
 - ▷ Les identifiants primaires sont les attributs soulignés dans le schéma :
 - ▷ **Medecin** MedNo
 - ▷ **Patient** PatNo
 - ▷ **Visite** VisNo
 - ▷ **Specialite** SpeNo
 - ▷ **Apour** {ApoMed, ApoSpe}
 - ▷ **Médicament** MdcNo
 - ▷ **Contrindic** {CntMdcA, CntMdcB}
 - ▷ **EltOrdonnance** {EltNo, EltVis}, l'identifiant {EltVis, EltMed} n'est pas l'identifiant primaire
 - ▷ Les clés étrangères sont précédées du mot clé **ref** dans le schéma et font référence à un identifiant primaire d'une autre table, indiqué par une flèche :
 - ▷ **Medecin** *aucune*

- ▷ **Patient** *aucune*
- ▷ **Visite** VisPat et VisMed
- ▷ **Specialite** *aucune*
- ▷ **Apour** ApoMed et ApoSpe
- ▷ **Médicament** MdcInit
- ▷ **Contrindic** CntMdcA et CntMdcB
- ▷ **EltOrdonnance** EltVis et EltMed

2. Le schéma fourni permet-il de représenter :

- (a) une ordonnance sans visite ?
non, une ordonnance à une clé étrangère obligatoire vers une visite
- (b) une visite ayant plusieurs médicaments prescrits ?
oui, si plusieurs ordonnances ont toutes la même visite comme référence
- (c) un médecin sans visite ?
oui, si on a aucune visite avec ce médecin comme référence VisMed
- (d) un médecin sans spécialité ?
oui, si on a aucune *apour* avec ce médecin comme référence ApoMed
- (e) une visite qui ne donne lieu à aucune prescription ?
oui, si aucune ordonnance référence cette visite
- (f) plusieurs fois la prescription du même médicament pour la même visite ?
non, car {EltVis, EltMed} est un identifiant de *EltOrdonnance*

3. Les requêtes SQL suivantes sont-elles correctes syntaxiquement ? Si non, pourquoi ?

- (a) **SELECT** medNo, speNo
FROM medecin
JOIN specialite **ON** medNo = apoMed and speNo = apoSpe
WHERE *medNom* **LIKE** 'S%' ;
incorrecte, un *join* entre medecin et apour doit être effectué avant celui de specialite
- (b) **SELECT DISTINCT** visDate, medNom
FROM visite, medecin
WHERE *visMed* = *medNo* ;
correcte, mais un *join* serait plus efficace, liste les dates avec le médecin pour lesquels une visite a eu lieu
- (c) **SELECT** patNom, medNom
FROM visite
JOIN medecin **ON** vismed = medNo
FROM visite
JOIN patient **ON** patNo = visPat
WHERE *visDate* = '01/02/2019' ;
incorrecte, pas de **FROM** après un **JOIN**
- (d) **SELECT** patNom, medNom
FROM visite
JOIN medecin **ON** vismed = MedNo
JOIN patient **ON** patNo = visPat
WHERE *visDate* = '01/02/2019' ;
correcte, on sélectionne les couples patient-médecin qui se sont vu le 01/02/2019
- (e) **SELECT** patNom, medNom, eltMed
FROM visite

JOIN medecin **ON** vismed = MedNo

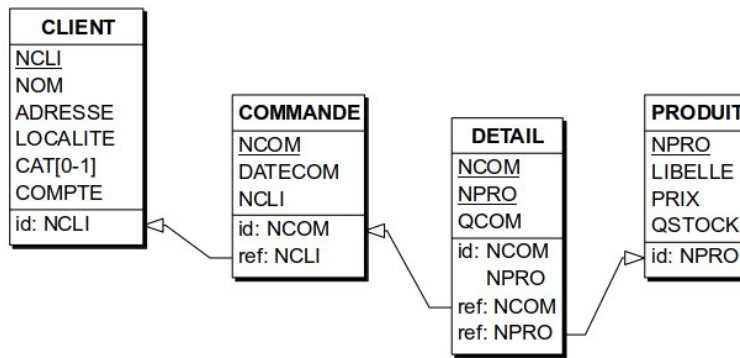
JOIN patient **ON** patNo = visPat

JOIN eltOrdonnance **ON** eltNo = visNo

WHERE *visDate* = '01/02/2019' ;

incorrecte, la condition de jointure **eltNo = visNo** n'a pas de sens et devrait être **eltVis = visNo** pour pouvoir lister les médicaments prescrits le 01/02/2019 ainsi que le médecin qui a fait la prescription et à quel patient

Voici un schéma conceptuel de base de données en extension d'une entreprise de construction.



Exercice 2

Algèbre relationnelle

Donner les expressions algébriques fournissant une relation reprenant :

1. les noms des clients qui ont au moins passé une commande

$$\pi_{\{nom\}}(client\ c \bowtie_{c.ncli=m.ncli} commande\ m)$$

2. les noms des clients qui ont au moins passé une commande avec le numéro de produit 'XXX'

$$\pi_{\{nom\}}(((client\ c \bowtie_{c.ncli=m.ncli} commande\ m) \bowtie_{d.ncom=m.ncom} (\sigma_{\{npro='XXX'\}}(detail\ d))))$$

3. les noms des clients qui ont au moins passé une commande avec un produit dont le prix unitaire est supérieur à 100 euros

$$\pi_{\{nom\}}(((client\ c \bowtie_{c.ncli=m.ncli} commande\ m) \bowtie_{d.ncom=m.ncom} detail\ d) \bowtie_{p.npro=d.npro} (\sigma_{\{prix>=100\}}(produit\ p)))$$

4. les localités où il y a au moins un compte à zéro ou un compte négatif

$$\pi_{\{localite\}}\sigma_{\{compte<=0\}}(client)$$

5. les localités où il y a des clients en catégorie B1

$$\pi_{\{localite\}}\sigma_{\{categorie='B1'\}}(client)$$

Exercice 3

SQL

Répondez aux questions suivantes, par une requête SQL.

1. Afficher pour chaque commande, le numéro de la commande et le nom du client qui l'a commandée.

```
SELECT commande.ncom, client.nom
FROM commande
JOIN client ON client.ncli = commande.ncli ;
```

2. Afficher pour chaque produit commandé, le numéro de la commande, le libellé et le prix du produit et enfin la quantité commandée.

```
SELECT detail.ncom, produit.libelle, produit.prix, detail.qcom
FROM detail
JOIN produit ON produit.npro = detail.npro ;
```

3. Ajouter la date de la commande à la question précédente.
SELECT detail.ncom, produit.libelle, produit.prix, detail.qcom, commande.datecom
FROM detail
JOIN produit **ON** produit.npro = detail.npro
JOIN commande **ON** commande.ncom = detail.ncom ;
4. Afficher la commande 30188 en détail, c'est à dire avec le numéro de commande, le nom, l'adresse et la localité du client, la date de commande et pour chaque produit commandé le numéro, le libellé, le prix et la quantité commandée.
SELECT commande.ncom, client.nom, client.adresse, client.localite, commande.datecom, produit.npro, produit.libelle, produit.prix, detail.qcom
FROM client
JOIN commande **ON** commande.ncli = client.ncli
JOIN detail **ON** detail.ncom = commande.ncom
JOIN produit **ON** produit.npro = detail.npro
WHERE *commande.ncom* = 30188 ;
5. Ajouter à la question précédente la somme à payer pour chaque produit commandé.
SELECT commande.ncom, client.nom, client.adresse, client.localite, commande.datecom, produit.npro, produit.libelle, produit.prix, detail.qcom, produit.prix * detail.qcom
AS total
FROM client
JOIN commande **ON** commande.ncli = client.ncli
JOIN detail **ON** detail.ncom = commande.ncom
JOIN produit **ON** produit.npro = detail.npro
WHERE *commande.ncom* = 30188 ;
6. Afficher le montant total de la commande 30188.
SELECT SUM(produit.prix * detail.qcom) **AS** total
FROM detail
JOIN produit **ON** produit.npro = detail.npro
WHERE *detail.ncom* = 30188 ;
7. Afficher les localités des clients qui commandent le produit de numéro 'CS464', avec doublons.
SELECT client.localite
FROM client
JOIN commande **ON** commande.ncli = client.ncli
JOIN detail **ON** detail.ncom = commande.ncom
WHERE *detail.npro* = 'CS464' ;
8. Quels sont les produits dont le libellé contient 'SAPIN' en majuscules et qui font l'objet d'une commande ?
SELECT DISTINCT produit.npro, produit.libelle
FROM detail
JOIN produit **ON** produit.npro = detail.npro
WHERE *produit.libelle* **LIKE** '%SAPIN%' ;
9. Afficher le numéro et le libellé des produits dont le libellé contient 'SAPIN' en majuscule et qui ont été commandés à Toulouse.
SELECT DISTINCT produit.npro, produit.libelle
FROM client
JOIN commande **ON** commande.ncli = client.ncli
JOIN detail **ON** detail.ncom = commande.ncom
JOIN produit **ON** produit.npro = detail.npro

- WHERE** *produit.libelle* **LIKE** '%SAPIN%' **AND** *client.localite* = 'Toulouse'
;
10. Dans quelles localités a-t-on commandé en décembre 2000 ?
SELECT DISTINCT *client.localite*
FROM *client*
JOIN *commande* **ON** *commande.ncli* = *client.ncli*
WHERE *commande.datecom* **BETWEEN** '2000 - 12 - 01' **AND** '2000 - 12 - 31' ;
11. Trouver le(s) client(s) qui a(ont) commandé des pointes en acier (ayant les mots 'POINTE' et 'ACIER' dans leur libellé, en majuscules). Afficher leurs coordonnées pour leur envoyer une lettre.
SELECT DISTINCT *client.nom*, *client.adresse*, *client.localite*
FROM *client*
JOIN *commande* **ON** *commande.ncli* = *client.ncli*
JOIN *detail* **ON** *detail.ncom* = *commande.ncom*
JOIN *produit* **ON** *produit.npro* = *detail.npro*
WHERE *produit.libelle* **LIKE** '%POINTE%' **AND** *produit.libelle* **LIKE** '%ACIER%'
;
12. Donner les clients dont le compte est en négatif et qui ont commandé en 2001.
SELECT DISTINCT *client.ncli*, *client.nom*, *client.compte*
FROM *client*
JOIN *commande* **ON** *commande.ncli* = *client.ncli*
WHERE *client.compte* < 0 **AND** *commande.datecom* **BETWEEN** '2001 - 01 - 01' **AND** '2001 - 12 - 31' ;
13. Quels produits ont été commandés au mois de janvier 2001 ?
SELECT DISTINCT *produit.npro*, *produit.libelle*
FROM *commande*
JOIN *detail* **ON** *detail.ncom* = *commande.ncom*
JOIN *produit* **ON** *produit.npro* = *detail.npro*
WHERE *commande.datecom* **BETWEEN** '2001 - 01 - 01' **AND** '2001 - 01 - 31' ;
14. Donnez la liste des numéros de commandes faites par des clients habitant Namur ou Bruxelles.
SELECT DISTINCT *ncom*
FROM *client*
JOIN *commande* **ON** *commande.ncli* = *client.ncli*
WHERE *localite* **IN** ('Namur', 'Bruxelles') ;
15. On recherche le(s) client(s) qui habite(nt) sur un boulevard (ayant le mot 'boulevard' ou une des abréviations suivantes 'bvd', 'bld', 'bd' dans leur adresse)
SELECT *ncli*, *nom*
FROM *client*
WHERE *adresse* **LIKE** '%boulevard%' **OR**
adresse **LIKE** '%bvd%' **OR**
adresse **LIKE** '%bld%' **OR**
adresse **LIKE** '%bd%' ;
16. Donner les numéros et libellés des produits qui ont déjà fait l'objet d'une commande pour plus de 50 exemplaires (dans la même commande), sans doublons.
SELECT DISTINCT *produit.npro*, *produit.libelle*
FROM *detail*
JOIN *produit* **ON** *produit.npro* = *detail.npro*
WHERE *detail.qcom* > 50 ;

17. Donner les numéros et libellés des produits qui ont déjà été inclus dans une commande pour plus de 5000 euros (dans la même commande), sans doublons.
- ```
SELECT DISTINCT produit.npro, produit.libelle, produit.prix * detail.qcom
AS somme
FROM detail
JOIN produit ON produit.npro = detail.npro
WHERE somme > 5000 ;
```
18. Donner la plus grande quantité en stock parmi tous les produits.
- ```
SELECT MAX(produit.qstock)  
FROM produit ;
```
19. Donner la plus grande quantité en stock parmi tous les produits *qui ont déjà fait l'objet d'une commande*.
- ```
SELECT MAX(produit.qstock)
FROM produit
JOIN detail ON detail.npro = produit.npro ;
```