

TD : Requêtes SQL (BDR.TD2-1)

INSA 3IF

2008-2009

Loïc Maisonnasse

1 Objectif

Ce TD a pour objectif de vous familiariser avec le langage de requêtes SQL. Et de vous apprendre à écrire des requêtes SQL pour rechercher des informations précises dans une base de données en utilisant les différents opérateurs de SQL.

2 Exercice 1

2.1 Schéma

Soit la base de données suivante permettant de gérer les consultations entre des médecins identifiés par leurs numéros RPPS (répertoire partagé des professionnels de santé) et des patients identifiés par leur numéro de sécurité social et associés à un médecin traitant.

PATIENT (numSS, nomP, prenomP, sexe, dateNaiss, numRPPS#)

Le patient est identifié par son numSS numéro de sécurité social et son médecin traitant est numRPPS

MEDECIN (numRPPS, nomM, prenomM, specialite, ville, adresse)

Le médecin est identifié par numRPPS et il peut avoir une spécialité

CONSULTE (numRPPS#, numSS#, date, diagnostic, numOrd#)

Un patient consulte un médecin à une certaine date. Le médecin effectue un diagnostic et lui prescrit une ordonnance dont l'identifiant est numOrd.

2.2 Requêtes

Écrire en SQL les requêtes qui permettent d'effectuer les actions suivantes :

- 1) Afficher la liste de tous les médecins

```
SELECT *  
FROM MEDECIN
```

- 2) Afficher le nom et le prénom de tous les patients, trier par ordre inverse sur les noms;

```
SELECT nomP, prenomP  
FROM PATIENT  
ORDER BY nom DESC ;
```

- 3) Afficher le nom, le prénom et la spécialité de tous les médecins de Lyon, si le médecin n'a

pas de spécialité, afficher 'généraliste' ;

```
SELECT nomM, prénomM, nvl(specialite, 'généraliste') AS Spé
FROM MEDECIN
WHERE ville='Lyon'
```

4) Afficher tous les patients nés après le 01/05/1985 ;

```
SELECT nomP, prenomP
FROM PATIENT
WHERE dateNaiss>'1985-05-01'
```

5) Afficher le nom et le prénom des médecins dont la spécialité contient la chaîne "logue" ;

```
SELECT nomM, prénomM
FROM MEDECIN
WHERE specialite LIKE '%logue%' ;
```

6) Afficher les prénoms des patients dont le nom est Dupont, Durant ou Martin ;

```
SELECT prenom
FROM PATIENT
WHERE nom IN ('Dupont', 'Durant', 'Martin') ;
```

7) Afficher le nombre de médecins à Lyon ;

```
SELECT COUNT(*)
FROM MEDECIN
WHERE ville='Lyon' ;
```

8) Afficher la liste des consultations qui n'ont pas donné lieu à une ordonnance;

```
SELECT *
FROM CONSULT
WHERE numOrd is null;
```

9) Afficher tous les médecins qui ont la même spécialité et qui sont dans la même ville que le médecin Paul DESMON ;

```
SELECT nomM, prenomM
FROM MEDECIN
WHERE (specialite, ville) =
(SELECT specialite, ville FROM MEDECIN
WHERE nomM='DESMON' AND prenomM='Paul' ;
```

10) Afficher la liste des patients (nom, prénom) avec le nom de leur médecin traitant, par ordre alphabétique inverse sur les noms puis sur les prénoms ;

```
SELECT nomP, prenomP, nomM
FROM PATIENT, MEDECIN
WHERE PATIENT.numRPPS=MEDECIN.numRPPS
ORDER BY nomP desc, prenomP desc;
```

11) Afficher pour chaque patient (nom, prénom) le nom et le prénom des médecins qu'il a consulté ;

```
SELECT nomP, prénomP, nomM
FROM PATIENT, MEDECIN, CONSULT
WHERE PATIENT.numSS=CONSULTE.numSS AND
CONSULTE.numRPPS=MEDECIN.numRPPS;
```

12) Afficher le nombre de consultations de chaque médecin (numRPPS) pour la journée du 14-10-2008, étiqueter cette colonne 'nbConsult' ;

```
SELECT numRPPS, COUNT(*) as ('nbConsult')
FROM CONSULTE
WHERE date='14-10-2008'
GROUP BY numRPPS;
```

13) Numéro de sécurité sociale du patient qui a le plus de consultations ;

```
SELECT numSS
FROM CONSULTE
GROUP BY numSS
HAVING COUNT(*)= (SELECT MAX(COUNT(*)) FROM CONSULT GROUP BY numSS)
```

14) Afficher tous les patients (numSS, nom, prénom) qui ont consulté un autre médecin que leur médecin traitant ;

```
SELECT numSS, nomP, prénomP
FROM PATIENT p
WHERE EXISTS
(SELECT * FROM CONSULTE WHERE numSS=p.numSS AND numRPPS!=p.numRPPS);
```

```
SELECT numSS, nomP, prénomP
FROM CONSULTE, PATIENT
WHERE PATIENT.numRPPS <> CONSULT.numRPPS
AND PATIENT.numSS=CONSUL.numSS ;
```

15) Afficher le nom et le prénom du ou des médecins qui sont médecins traitants du plus grand nombre de patients ;

```
SELECT DISTINCT nomM, prénomM
FROM MEDECIN
WHERE MEDECIN.numRPPS =
(SELECT numRPPS
FROM PATIENT
GROUP BY numRPPS
HAVING COUNT(numSS) =
(SELECT MAX(COUNT(numSS))
FROM PATIENT
GROUP BY numRPPS));
```

16) Afficher tous les patients (nom, prénom) qui ont consulté le médecin ayant le numRPPS '12345' entre le 18-06-2008 et le 17-07-2008 ;

```

SELECT nomP, prenomP
FROM PATIENT
WHERE numSS IN
  (SELECT numSS FROM CONSULTE
   WHERE numRPPS='12345' AND date BETWEEN '18-06-2008' AND '17-07-2008') ;

```

17) Afficher tous les patients (nom, prénom) qui ont consulté le médecin ayant le numRPPS '12345' le 18-06-2008 et le 17-07-2008 ;

```

SELECT nomP, prenomP
FROM PATIENT, CONSULT AS c1, CONSULT AS c2
WHERE PATIENT.numSS = c1.numSS
AND PATIENT.numSS = c2.numSS
AND c1.numRPPS = c2.numRPPS
AND c2.numRPPS = '12345'
AND c1.date = '18-06-2008'
AND c2.date = '17-07-2008'

```

```

SELECT nomP, prenomP
FROM PATIENT, CONSULT
WHERE PATIENT.numSS = CONSULT.numSS AND date = '18-06-2008'
INTERSECT
SELECT nomP, prenomP
FROM PATIENT, CONSULT
WHERE PATIENT.numSS = CONSULT.numSS AND date = '17-07-2008'

```

18) Liste des patients qui ont consulté au moins deux médecins de la même spécialité le même jour ;

```

SELECT nomP, prenomP
FROM PATIENT, CONSULTE, MEDECIN
WHERE PATIENT.numSS = CONSULTE.numSS AND CONSULTE.numRPPS = MEDECIN.numRPPS
GROUP BY CONSULTE.numSS, CONSULTE.date, MEDECIN.spécialité
HAVING COUNT(DISTINCT numRPPS) > 2 ;

```

```

SELECT nomP, prenomP
FROM PATIENT P
WHERE
  (SELECT COUNT(DISTINCT numRPPS)
   FROM CONSULTE, MEDECIN
   WHERE CONSULTE.numRPPS = MEDECIN.numRPPS AND CONSULTE.numSS = P.numRPPS
   GROUP BY CONSULTE.numSS, CONSULTE.date, MEDECIN.spécialité) >2

```

Exercice 2

2.3 Schéma

On souhaite maintenant ajouter les deux tables suivantes pour pouvoir gérer les prescriptions des médecins.

PRESCRIPTION (numOrd, idMed#, nbBoites)

Une ligne de l'ordonnance du médecin 'numOrd' prévoit de donner nbBoites du médicament idMed

MEDICAMENT (idMed, nomMed, prix, categorie)

Le médicament est identifié, il a un certain prix et correspond à une catégorie de médicament

2.4 Création des tables

19) Créer la table PRESCRIPTION ;

```
CREATE TABLE PRESCRIPTION (numOrd NUMBER, idMed NUMBER, nbBoites NUMBER, PRIMARY KEY (numOrd, idMed), FOREIGN KEY (idMed) REFERENCES MEDICAMENT(idMed))
```

20) Créer la table MEDICAMENT ;

```
CREATE TABLE MEDICAMENT (idMed NUMBER PRIMARY KEY, nomMed CHAR(40), prix FLOAT)
```

21) Ajouter le médicament n° 5432, avec le nom 'aspirine' et le prix '11,05' et de catégorie 'Antalgique';

```
INSERT INTO MEDICAMENT VALUES (5432, 'aspirine', '11,05', 'Antalgique')
```

2.5 Requêtes

22) Pour chaque médicament écrire une requête qui donne le résultat suivant :

< nom du médicament > coute < prix > euros toutes taxes et < prix / 1.055 > hors taxes

```
SELECT nom||' coute '||prix||' euros toutes taxes et '|| prix/1.055 " hors taxes"
FROM MEDICAMENT
```

23) Donner le prix et le nom des médicaments dont le prix est supérieur au prix moyen de leur catégorie.

```
SELECT nomMed, prix
FROM MEDICAMENT m
WHERE m.prix > (SELECT AVG(prix) FROM MEDICAMENT WHERE m.categorie=categorie)
```

24) Donner le nom de la catégorie et le prix du médicament le moins cher de la catégorie ayant le plus grand prix moyen.

```
SELECT categorie, MIN(prix)
FROM MEDICAMENT
GROUP BY categorie
HAVING avg (prix) = (SELECT MAX(AVG(prix)) FROM MEDICAMENT GROUP BY categorie)
```

25) Afficher pour tous les médecins (numRPPS) en activité le '11-08-2008' la liste des patients (numSS) qu'ils ont traité ainsi que le montant total de leurs ordonnances (somme des médicaments prescrits avec le prix multiplié par le nombre de boîtes) ;

```
SELECT numRPPS, numSS, SUM(prix*nbBoite)
FROM CONSULT c,PRESCRIPTION p,MEDICAMENT m
WHERE c.numOrd=p.numOrd AND p.numMed=m.numMed AND c.date='11-08-2008'
group by (numRPPS, numSS)
```

26) Afficher le prix de la plus petite ordonnance ;

```
SELECT SUM(prix*nbBoites)
FROM PRESCRIPTION,MEDICAMENT
WHERE PRESCRIPTION.id_med = MEDICAMENT.id_med
GROUP BY (numOrd)
HAVING SUM(prix*nbBoites)=
  (SELECT MIN(SUM(prix*nbBoites)) FROM PRESCRIPTION,MEDICAMENT
  WHERE PRESCRIPTION.id_med = MEDICAMENT.id_med
  GROUP BY (numOrd)
  )
```

2.6 Requêtes facultatives

27) Afficher la liste des médecins accompagnée du prix maximum de leur ordonnance ;

```
SELECT nomM,prenomM, max(px)
FROM MEDECIN,
  (SELECT numOrd, numRPPS, SUM(prix*nbBoites) AS px
  FROM CONSULTE,PRESCRIPTION,MEDICAMENT
  WHERE CONSULTE.numOrd=prescription.numOrd
  PRESCRIPTION.id_med = MEDICAMENT.id_med
  GROUP BY numOrd, numRPPS)
WHERE DEP.numRPPS=MEDECIN.numRPPS
```

28) Donner le nom et le prénom des patients qui ont eu la plus grosse prescription ;

```
SELECT nomP,prenomP
FROM PATIENT,CONSULT
WHERE PATIENT.numSS=CONSULT.numSS AND CONSULT.numOrd IN
  (SELECT numOrd
  FROM PRESCRIPTION,MEDICAMENT
  WHERE PRESCRIPTION.id_med = MEDICAMENT.id_med
  GROUP BY (numOrd)
  HAVING SUM(prix*nbBoites)=
    (SELECT MAX(SUM(prix*nbBoites)) FROM PRESCRIPTION,MEDICAMENT
    WHERE PRESCRIPTION.id_med = MEDICAMENT.id_med
    GROUP BY (numOrd)
    )
  )
```

29) Donner le nom du médicament le plus prescrit par les 'pneumologues' ;

```
SELECT idMed
FROM MEDECIN, CONSULT, PRESCRIPTION
WHERE MEDECIN.specialite = 'pneumologue'
AND MEDECIN.numRPPS=CONSULT.numRPPS
```

```
AND CONSULT.numOrd=PRESCRIPTION.numOrd
GROUP BY idMed
HAVING count(*)=(SELECT MAX(COUNT(*))
FROM MEDECIN, CONSULT, PRESCRIPTION
WHERE MEDECIN.specialite = 'pneumologue'
AND MEDECIN.numRPPS=CONSULT.numRPPS
AND CONSULT.numOrd=PRESCRIPTION.numOrd
GROUP BY idMed)
```