

Exercices sur le Langage de Manipulation des Données

Pour réaliser les exercices de ce laboratoire, relire la théorie concernant le LMD, ainsi que le résumé donné dans le cahier de laboratoire de la page 65 à la page 72. Créer également la base de donnée Hopital complète avec les scripts fournis.

Créer également la base de donnée `Hopital` complète avec les scripts fournis dans le centre de ressources ([CreationHopital.sql](#) et [0-InsertAll.sql](#)).

LMD 1

Afficher, triés par âge, le numéro de SIS, le libellé de la mutuelle et l'âge de tous les patients du médecin `Collignon` habitant à `Tenneville`. Attention, le code `mutuelle` n'est pas toujours précisé dans la table Patients. Pour ces patients-là, on affichera, au lieu du libellé de la mutuelle, `Pas de mutuelle`. Résoudre au moyen de requêtes ensemblistes ainsi qu'au moyen de jointures.

```
-- Requête ensembliste :

select *
from (SELECT NrSis,
            SUBSTR(P.Nom, 1, 25)                                As NOM,
            SUBSTR(Mu.Libelle, 1, 60)                            As
Mutuelle,
            TO_CHAR(CURRENT_DATE, 'YYYY') - TO_CHAR(P.DateNaissance, 'YYYY') AS Age
FROM Mutuelles Mu
    INNER JOIN Patients P USING (CodeMutuelle)
    INNER JOIN PatientsMedecins PM USING (NrSis)
    INNER JOIN Medecins Me USING (NrMedecin)
    INNER JOIN Adresses Ad ON (Me.codeadresse = Ad.CodeAdresse)
WHERE UPPER(Me.Nom) = UPPER('Collignon')
    AND UPPER(Ad.Localite) = UPPER('Tenneville')
    AND CodeMutuelle IS NOT NULL
UNION
SELECT NrSis,
       SUBSTR(P.Nom, 1, 25),
       'Pas de Mutuelle',
       TO_CHAR(CURRENT_DATE, 'YYYY') - TO_CHAR(P.DateNaissance, 'YYYY') AS Age
FROM Patients P
    INNER JOIN PatientsMedecins PM USING (NrSis)
    INNER JOIN Medecins Me USING (NrMedecin)
    INNER JOIN Adresses Ad ON (Me.codeadresse = Ad.CodeAdresse)
WHERE UPPER(Me.Nom) = UPPER('Collignon')
```

```

        AND UPPER(Ad.Localite) = UPPER('Tenneville')
        AND CodeMutuelle IS NULL)
ORDER BY Age;

-- Requête avec jointure :
SELECT P.NrSis,
       SUBSTR(P.Nom, 1, 25)                                As NOM,
       COALESCE(SUBSTR(Mu.Libelle, 1, 60), 'Pas de Mutuelle') As Mutuelle,
       TO_CHAR(CURRENT_DATE, 'YYYY') - TO_CHAR(P.DateNaissance, 'YYYY') AS Age
FROM Patients P
     LEFT JOIN Mutuelles Mu
           ON P.CodeMutuelle = Mu.CodeMutuelle
     INNER JOIN PatientsMedecins PM ON P.NrSis = PM.NrSis
     INNER JOIN Medecins Me ON PM.NrMedecin = Me.NrMedecin
     INNER JOIN Adresses Ad ON Me.CodeAdresse = Ad.CodeAdresse
WHERE UPPER(Me.Nom) = UPPER('Collignon')
      AND UPPER(Ad.Localite) = UPPER('Tenneville')
ORDER BY Age;

```

Calcul de l'âge :

```

select (CURRENT_DATE - P.DateNaissance) / 365.25
from PATIENTS p;

select MONTHS_BETWEEN(CURRENT_DATE, P.DateNaissance) / 12
from PATIENTS p;

select EXTRACT(YEAR FROM CURRENT_DATE) - EXTRACT(YEAR FROM P.DateNaissance)
from PATIENTS p;

select TO_CHAR(CURRENT_DATE, 'YYYY') - TO_CHAR(P.DateNaissance, 'YYYY')
from PATIENTS p;

```

Résultats :

NRSIS	NOM	MUTUELLE	AGE
86012200206	MATITA	Pas de Mutuelle	35
86030800127	BELANGE	Pas de Mutuelle	35
84092300166	CLOSE	Caisse des Soins de sante de la SNCB Centre Medical Regional	37
84011100621	COLLURA	Pas de Mutuelle	37
83042600572	COSENTINO	Pas de Mutuelle	38
75092100025	JIANG	Office regional de Namur	46
74080100511	VOROBIEVA	Office regional de Namur	47
73081800412	SANGWA MAYANI	Office regional de Namur	48
70120200144	OUAFIQ	Office regional de Namur	51

LMD 2

Afficher, triés par `nom`, le `nom` et le `prénom` des patients `célibataires` de `Saint-Nicolas`, de moins de `39` ans et qui n'ont pas de médecins. À résoudre par requêtes ensemblistes, imbriquées, corrélées et au moyen de jointures.

```
-- Requête ensembliste :
SELECT Nrsis, SUBSTR(Nom, 1, 10) AS Nom, Prenom
FROM Patients,
    Adresses
WHERE Patients.CodeAdresse = Adresses.CodeAdresse
    AND EtatCivil = 'C'
    AND UPPER(Localite) = UPPER('Saint-Nicolas')
    AND TO_CHAR(CURRENT_DATE, 'YYYY') - TO_CHAR(DateNaissance, 'YYYY') < 39
MINUS
SELECT Patients.Nrsis, Nom, Prenom
FROM Patients,
    PatientsMedecins
WHERE Patients.Nrsis = PatientsMedecins.Nrsis
ORDER BY Nom;

-- Si on ne veut afficher que nom et prénom, mais continuer à traiter les homonymes, on
aura la solution suivante :
SELECT Nom, Prenom
FROM Patients,
    Adresses
```

```

WHERE Patients.CodeAdresse = Adresses.CodeAdresse
AND EtatCivil = 'C'
AND UPPER(Localite) = UPPER('Saint-Nicolas')
AND TO_CHAR(CURRENT_DATE, 'YYYY') - TO_CHAR(DateNaissance, 'YYYY') < 39
AND NrSis IN
    (SELECT NrSis
     FROM Patients
     MINUS
     SELECT NrSis
     FROM PatientsMedecins)
ORDER BY Nom;

```

-- Requêtes imbriquées :

```

SELECT Nom, Prenom
FROM Patients,
     Adresses
WHERE Patients.CodeAdresse = Adresses.CodeAdresse
AND EtatCivil = 'C'
AND UPPER(Localite) = UPPER('Saint-Nicolas')
AND TO_CHAR(CURRENT_DATE, 'YYYY') -
    TO_CHAR(DateNaissance, 'YYYY') < 39
AND NrSis NOT IN (SELECT DISTINCT NrSis
                  FROM PatientsMedecins)
ORDER BY Nom;

```

-- Requêtes corrélées :

```

SELECT Nom, Prenom
FROM Patients,
     Adresses
WHERE Patients.CodeAdresse = Adresses.CodeAdresse
AND EtatCivil = 'C'
AND UPPER(Localite) = UPPER('Saint-Nicolas')
AND TO_CHAR(CURRENT_DATE, 'YYYY') -
    TO_CHAR(DateNaissance, 'YYYY') < 39
AND NOT EXISTS(SELECT *
               FROM PatientsMedecins
               WHERE Patients.NrSis = PatientsMedecins.Nrsis)
ORDER BY Nom;

```

-- Une autre solution corrélée est d'avoir une sous-requête dans laquelle on compte le nombre d'enregistrements pour le nrsis dans la table patientsmedecins et dont on compare le résultat avec 0. (Méthode non préconisée par Oracle)

```

SELECT Nom, Prenom
FROM Patients,
     Adresses
WHERE Patients.CodeAdresse = Adresses.CodeAdresse
AND EtatCivil = 'C'

```

```

AND UPPER(Localite) = UPPER('Saint-Nicolas')
AND TO_CHAR(CURRENT_DATE, 'YYYY') -
    TO_CHAR(DateNaissance, 'YYYY') < 39
AND 0 = (SELECT COUNT(*)
        FROM PatientsMedecins
        WHERE Patients.NrSis = PatientsMedecins.NrSis)
ORDER BY Nom;

-- Requête avec jointure :
SELECT nrsis, Nom, Prenom
FROM Patients
    INNER JOIN Adresses
        USING (CodeAdresse)
    LEFT OUTER JOIN PatientsMedecins
        USING (nrsis)
WHERE EtatCivil = 'C'
AND UPPER(Localite) = UPPER('Saint-Nicolas')
AND TO_CHAR(CURRENT_DATE, 'YYYY') -
    TO_CHAR(DateNaissance, 'YYYY') < 39
AND NrMEdecin IS NULL
ORDER BY Nom;

-- En créant une vue :

CREATE OR REPLACE VIEW CSaintNicolasN33 AS
(
SELECT nrSis, nom, prenom
FROM adresses a,
    patients p
WHERE a.codeadresse = p.codeadresse
AND EtatCivil = 'C'
AND Upper(Localite) = 'SAINT-NICOLAS'
AND TO_CHAR(CURRENT_DATE, 'YYYY') - TO_CHAR(DateNaissance, 'YYYY') < 39);

SELECT CSaintNicolasN33.nom, CSaintNicolasN33.prenom
FROM CSaintNicolasN33
    LEFT JOIN PatientsMedecins
        ON (CSaintNicolasN33.NrSis = PATIENTSMEDECINS.NrSis)
WHERE PATIENTSMEDECINS.NrSis IS NULL;

```

Résultat en 2021 :

NOM	PRENOM
AIT HMAD	Youcef
BELLEN	Antzo
BELLOMO	Nazaire
ZUCCHETTO	Eozenn

LMD3

Afficher, triés par nom, le nom et le prénom des patientes nées dans les années 1970 qui se font soigner par les médecins Geoffroy Munnix et Lala Kaison. À résoudre par requêtes ensemblistes, imbriquées, corrélées et au moyen de jointures.

```
-- Requête ensembliste :
SELECT NrSis, P.Nom, P.Prenom
FROM Patients P
      INNER JOIN PatientsMedecins PM Using (NrSis)
      INNER JOIN Medecins M USING (NrMedecin)
WHERE P.Sexe = 'F'
      AND EXTRACT(YEAR FROM P.DateNaissance) BETWEEN 1970 AND 1979
      AND UPPER(M.Nom) = 'MUNNIX'
      AND UPPER(M.Prenom) = 'GEOFFROY'
INTERSECT
SELECT NrSis, P.Nom, P.Prenom
FROM Patients P
      INNER JOIN PatientsMedecins PM Using (NrSis)
      INNER JOIN Medecins M USING (NrMedecin)
WHERE P.Sexe = 'F'
      AND EXTRACT(YEAR FROM P.DateNaissance) BETWEEN 1970 AND 1979
      AND UPPER(M.Nom) = 'KAISON'
      AND UPPER(M.Prenom) = 'LALA'
ORDER BY Nom;

-- Attention, il faut ajouter le nrSis dans les select pour éviter des problèmes avec
les homonymes.
-- ATTENTION : un seul ORDER BY, tout à la fin !!

-- Requêtes imbriquées : (SDD : a faire avec des INNER JOIN plutôt que des jointures
manuelles)

SELECT P.NrSis, P.Nom, P.Prenom
FROM Patients P,
      PatientsMedecins PM,
```

```

        Medecins M
WHERE P.NrSis = PM.NrSis
      AND M.NrMedecin = PM.NrMedecin
      AND P.Sexe = 'F'
      AND EXTRACT(YEAR FROM P.DateNaissance) BETWEEN 1970 AND 1979
      AND UPPER(M.Nom) = 'MUNNIX'
      AND UPPER(M.Prenom) = 'GEOFFROY'
      AND P.NrSis IN (SELECT NrSis
                      FROM PatientsMedecins PM,
                           Medecins M
                      WHERE M.NrMedecin = PM.NrMedecin
                           AND UPPER(M.Nom) = 'KAISON'
                           AND UPPER(M.Prenom) = 'LALA')

ORDER BY Nom;

```

-- Requêtes corrélées : (SDD : a faire avec des INNER JOIN plutôt que des jointures manuelles)

```

SELECT P.NrSis, P.Nom, P.Prenom
FROM Patients P,
     PatientsMedecins PM,
     Medecins M
WHERE P.NrSis = PM.NrSis
      AND M.NrMedecin = PM.NrMedecin
      AND P.Sexe = 'F'
      AND EXTRACT(YEAR FROM P.DateNaissance) BETWEEN 1970 AND 1979
      AND UPPER(M.Nom) = 'MUNNIX'
      AND UPPER(M.Prenom) = 'GEOFFROY'
      AND EXISTS(SELECT *
                 FROM PatientsMedecins PMI,
                      Medecins MI
                 WHERE MI.NrMedecin = PMI.NrMedecin
                      AND PMI.NrSis = P.NrSis
                      AND UPPER(MI.Nom) = 'KAISON'
                      AND UPPER(MI.Prenom) = 'LALA')

ORDER BY Nom;

```

-- Requête avec jointure : (SDD : a faire avec des INNER JOIN plutôt que des jointures manuelles)

```

SELECT P.NrSis, P.Nom, P.Prenom
FROM Patients P,
     PatientsMedecins PM1,
     Medecins M1,
     PatientsMedecins PM2,
     Medecins M2
WHERE P.NrSis = PM1.NrSis
      AND PM1.NrMedecin = M1.NrMedecin
      AND UPPER(M1.Nom) = 'MUNNIX'
      AND UPPER(M1.Prenom) = 'GEOFFROY'

```

```

AND P.NrSis = PM2.NrSis
AND PM2.NrMedecin = M2.NrMedecin
AND UPPER(M2.Nom) = 'KAISON'
AND UPPER(M2.Prenom) = 'LALA'
AND P.Sexe = 'F'
AND EXTRACT(YEAR FROM P.DateNaissance) BETWEEN 1970 AND 1979
ORDER BY Nom;

```

LMD 4

Rechercher, triés par ordre alphabétique, le `nom` et le `prénom` de tous les médecins homme, `célibataires` qui ne sont chef d'aucun service. À résoudre par requêtes ensemblistes, imbriquées, corrélées et au moyen de jointures.

```

--Requête ensembliste :
SELECT nom, prenom
FROM Medecins
WHERE Sexe = 'M'
      AND EtatCivil = 'C'
MINUS
SELECT nom, prenom
FROM Medecins M,
      HopitauxServices HP
WHERE M.NrMedecin = HP.ChefService
      AND M.Sexe = 'M'
      AND M.EtatCivil = 'C'
ORDER BY 1, 2;

-- Requêtes imbriquées :
SELECT nom, prenom
FROM Medecins
WHERE Sexe = 'M'
      AND EtatCivil = 'C'
      AND NrMedecin NOT IN (SELECT ChefService
                             FROM HopitauxServices)
ORDER BY 1, 2;

-- Requêtes corrélées :
SELECT M.nom, M.prenom
FROM Medecins M
WHERE Sexe = 'M'
      AND EtatCivil = 'C'
      AND NOT EXISTS(SELECT *
                      FROM HopitauxServices HP
                      WHERE HP.ChefService = M.NrMedecin)
ORDER BY 1, 2;

```



```
-- Requête avec jointure :
SELECT nom, prenom
FROM Medecins M
      LEFT JOIN HopitauxServices HP
            ON M.NrMedecin = HP.ChefService
WHERE M.Sexe = 'M'
      AND M.EtatCivil = 'C'
      AND HP.ChefService IS NULL
ORDER BY 1, 2;
```

LMD 5

Enregistrer la patiente **Lea Dupont** de nationalité **française**, domiciliée **Rue du centre** à **Yvoir**. Cette patiente est **célibataire**, elle est née le 14 juillet 1975 et est affiliée à la mutuelle **Office régional de Namur**. Cette patiente est allergique à la substance **Codéine**. Elle se fait soigner par le médecin chef de service de la **Neuropsychiatrie** de l'hôpital **Clinique Saint-Vincent**. Le **NrSis** est constitué de 11 caractères sous la forme **YYMMDDXXXXX** **YYMMDD** sont l'année, le moi et le jour de la date de naissance du patient. **xxxxx** est le numéro de SIS le plus élevé mémorisé dans la base plus UN.

```
INSERT INTO Adresses
      (CodeAdresse, Adresse, Localite, CodePostal)
VALUES (1 + (COALESCE((SELECT MAX(CodeAdresse) FROM Adresses), 0)),
      'Rue du centre',
      'YVOIR',
      (SELECT CodePostal FROM Adresses WHERE Localite = 'YVOIR'));

INSERT INTO Patients
      (NrSis, Nom, Prenom, Sexe, EtatCivil, Nationalite, Datenaissance, CodeMutuelle,
      CodeAdresse)
VALUES ('750714' || COALESCE(LPAD(TO_CHAR(1 + (SELECT MAX(Substr(NrSis, 7)) FROM
Patients)), 5, '0'), '00001'),
      'DUPONT', 'Lea', 'F', 'C',
      (SELECT CodePays FROM Pays WHERE libelle = 'FRANCE'),
      TO_DATE('14/07/75', 'DD/MM/RR'),
      (SELECT Codemutuelle FROM Mutuelles WHERE libelle = 'Office régional de
Namur'),
      (SELECT CodeAdresse
      FROM Adresses
      WHERE Adresse = 'Rue du centre'
            AND Localite = 'YVOIR')); -- mieux : (SELECT MAX(CodeAdresse) FROM adresses)

INSERT INTO allergies (NrSis, NrSubstance)
```

```

SELECT NrSis, NrSubstance
FROM (SELECT NrSis
      FROM Patients
      WHERE UPPER(Nom) = 'DUPONT'
        AND UPPER(Prenom) = 'LEA') p,
(SELECT NrSubstance
 FROM Substances
 WHERE UPPER(Nom) = 'CODEINE') s
;

INSERT INTO PatientsMedecins (NrSis, NrMedecin)
VALUES ((SELECT NrSis
        FROM Patients
        WHERE UPPER(Nom) = 'DUPONT'
          AND UPPER(Prenom) = 'LEA'),
(SELECT M.NrMedecin
 FROM Medecins M,
        HopitauxServices HS,
        Hopitaux H,
        Services S
 WHERE M.NrMedecin = HS.ChefService
       AND HS.NrService = S.NrService
       AND HS.NrHopital = H.NrHopital
       AND UPPER(S.Libelle) = 'NEUROPSYCHIATRIE'
       AND UPPER(H.Nom) = 'CLINIQUE SAINT-VINCENT')));

Rollback;

```

Rem : On devrait faire un `commit`, mais on fait un rollback dans le cadre du labo afin de pouvoir refaire le test et de ne pas modifier les données de la base.

LMD 6

Le patient qui a le plus d'allergies vient de se marier. Il faut donc mettre son état civil à `M`. Il change de médecin(s). Il se fait soigner par tous les médecins de plus de 55 ans domiciliés dans sa localité. Faire les modifications nécessaires dans la base de données.

```

-- Pas besoin de patients dans requête imbriquée !!!

UPDATE Patients
SET EtatCivil = 'M'
WHERE NrSis =
  (SELECT NrSis
   FROM Allergies

```

```

        GROUP BY NrSis
        HAVING COUNT(*) = (SELECT MAX(COUNT(NrSis))
                           FROM Allergies
                           GROUP BY NrSis));

DELETE
FROM PatientsMedecins
WHERE NrSis =
    (SELECT NrSis
     FROM Allergies
     GROUP BY NrSis
     HAVING COUNT(*) >= ALL (SELECT MAX(COUNT(NrSis))
                             FROM Allergies
                             GROUP BY NrSis));

INSERT INTO PatientsMedecins
    (SELECT NrSis, NrMedecin
     FROM Patients P,
          Adresses AP,
          Medecins M,
          Adresses AM
     WHERE P.CodeAdresse = AP.CodeAdresse
          AND M.CodeAdresse = AM.CodeAdresse
          AND AP.codepostal = AM.codepostal
          AND EXTRACT(YEAR FROM CURRENT_DATE) - EXTRACT(YEAR FROM M.DateNaissance) > 55
          AND NrSis =
              (SELECT NrSis
               FROM Allergies
               GROUP BY NrSis
               HAVING COUNT(*) = (SELECT MAX(COUNT(NrSis))
                                   FROM Allergies
                                   GROUP BY NrSis))));

Rollback;

-- Cette requête peut être simplifiée en créant 2 vues : une qui contient toutes les
données de patients, y compris leur adresse (donc une vue qui utilise la table patients
et la table adresse) et une vue qui contient toutes les données des médecins et leur
adresse. => la requête de l'exercice ne demande plus que ces 2 vues à la place des 4
tables !

```

Rem : On devrait faire un `commit`, mais on fait un rollback dans le cadre du labo afin de pouvoir refaire le test et de ne pas modifier les données de la base.

LMD 7

Tous les patients de Welkenraedt nés en 1982 qui n'avaient pas d'allergies sont allergiques à toutes les substances du médicament Emcoretic. Faire les modifications nécessaires dans la base de données.

```

INSERT INTO Allergies
SELECT NrSis, NrSubstance
FROM (SELECT NrSis
      FROM Patients P,
           Adresses A
      WHERE P.CodeAdresse = A.CodeAdresse
           AND EXTRACT(YEAR FROM DateNaissance) = 1982
           AND UPPER(Localite) = 'WELKENRAEDT'
           AND NrSis NOT IN (SELECT NrSis FROM Allergies)) P,
      (SELECT NrSubstance
      FROM Composition,
           Medicaments
      WHERE Composition.NrMedicament = Medicaments.NrMedicament
           AND UPPER(Denomination) = 'EMCORETIC') S;

rollback;

```

LMD 8

Le médecin **Miquel Brasseur** prend sa retraite. Il cède tous ses patients au médecin de sa spécialité habitant sa commune. On désire ne plus garder de trace de ce médecin.

```

UPDATE PatientsMedecins
SET NrMedecin =
    (SELECT NrMedecin
     FROM Medecins M,
          Adresses A
     WHERE M.CodeAdresse = A.CodeAdresse
          AND Localite = (SELECT Localite
                          FROM Medecins M,
                               Adresses A
                          WHERE M.CodeAdresse = A.CodeAdresse
                               AND UPPER(M.Nom) = 'BRASSEUR'
                               AND UPPER(M.Prenom) = 'MIQUEL')
     AND Specialite = (SELECT Specialite
                       FROM Medecins
                       WHERE UPPER(Nom) = 'BRASSEUR'
                              AND UPPER(Prenom) = 'MIQUEL')
     AND NrMedecin <> (SELECT NrMedecin
                       FROM Medecins
                       WHERE UPPER(Nom) = 'BRASSEUR'
                              AND UPPER(Prenom) = 'MIQUEL'))
WHERE NrMedecin =
    (SELECT NrMedecin
     FROM Medecins
     WHERE UPPER(Nom) = 'BRASSEUR'
          AND UPPER(Prenom) = 'MIQUEL');

```

```
DELETE
FROM MedecinsServices
WHERE NrMedecin = (SELECT NrMedecin
                    FROM Medecins
                    WHERE UPPER(Nom) = 'BRASSEUR'
                    AND UPPER(Prenom) = 'MIQUEL');

DELETE
FROM HopitauxServices
WHERE ChefService = (SELECT NrMedecin
                     FROM Medecins
                     WHERE UPPER(Nom) = 'BRASSEUR'
                     AND UPPER(Prenom) = 'MIQUEL');

DELETE
FROM Medecins
WHERE UPPER(Nom) = 'BRASSEUR'
      AND UPPER(Prenom) = 'MIQUEL';

Rollback;
```