Bases de Données Relationnelles

TP2 : Exercices sur les jointures, le group by et la récursivité en SQL

SI3 – SI4 – MAM4

Les données sont disponibles à l'adresse:
`http://users.polytech.unice.fr/~rueher/Cours/BD/TP2/`

1 Group By

Soit la table *si4* (cf. data_group_by.sql), écrire une requête SQL qui affiche pour chaque groupe : le numéro, le nombre d'étudiants, la date de naissance du plus vieil étudiant du groupe, et la date de naissance du plus jeune étudiant du groupe.

```
\i data_group_by.sql
SELECT   groupe,
        COUNT(*)     -- compte les étudiants par groupe
,        MIN(ddn)    -- date de naissance du plus vieil étudiant par groupe
,        MAX(ddn)    -- date de naissance du plus jeune étudiant par groupe
FROM     si4
GROUP BY groupe;
```

---

2 Récursivité

1. Vols: Reprendre l'exemple du cours sur les vols et rechercher les liaisons avec au moine deux escales.

```
\i data_vols.sql

-- Recherche des vols dont la duree de vol est superieure a 10h

WITH RECURSIVE reaches(departure,totaltime, arrival) AS
(SELECT departure, atime-dtime, arrival FROM vols
UNION  --ALL
SELECT R1.departure,  R1.atime-R1.dtime + R2.totaltime, R2.arrival
FROM  vols AS R1,
reaches AS R2
WHERE R1.arrival =R2.departure)
SELECT * FROM reaches where totaltime > '10:00';

-- Recherche des vols avec au moins une escale

WITH RECURSIVE reaches1(departure,escales, arrival) AS
(SELECT departure, 0, arrival FROM vols
UNION  --ALL
SELECT R1.departure,  1 + R2.escales, R2.arrival
FROM  vols AS R1,
reaches1 AS R2
```

```
WHERE R1.arrival =R2.departure)
SELECT * FROM reaches1 where escales > 1;
```

---

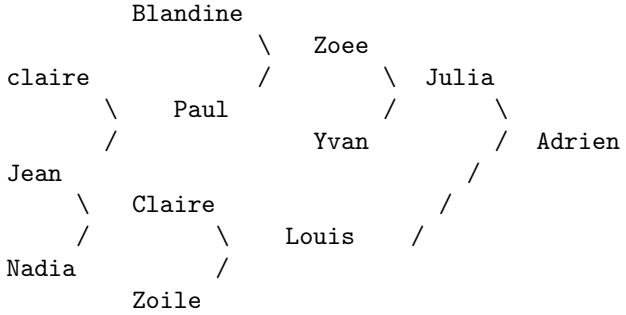
2. Nombres pairs
   Utiliser la récursivité en SQL pour calculer l'ensemble des nombres pairs inférieurs ou égal à 100 (et la somme de ces nombres).

```
WITH RECURSIVE t(n) AS (
    VALUES (2)
  UNION
    SELECT n+2 FROM t WHERE n < 100
)
SELECT n FROM t;

WITH RECURSIVE t(n) AS (
    VALUES (2)
  UNION
    SELECT n+2 FROM t WHERE n < 100
)
SELECT sum(n) FROM t;
```

---

3. Ascendants
   Soit le graphe de relations:



   - Stocker cette relation dans une table qui a pour attributs : le père, la mère et l'enfant;
   - Calculer l'ensemble des ascendants de 'Julia' en utilisant la récursivité en SQL.

```
\i data_ascendants.sql

WITH RECURSIVE ancetre(Aieul, Enfant) AS
(SELECT Pere, Enfant FROM Parents
UNION
SELECT Mere, Enfant FROM Parents
UNION
SELECT R2.Aieul, R1.Enfant
FROM  Parents AS R1,
ancetre AS R2
WHERE R1.Pere =R2.Enfant
                                OR  R1.Mere =R2.Enfant )
```

```
--SELECT * FROM ancetre ;
SELECT * FROM ancetre where Enfant='Julia';
```

---

3 Jointure externe

Soit les tables:

```
DROP table client;
CREATE TABLE client
( CLI_ID   int,
 CLI_NOM   char(12));
INSERT INTO Client VALUES (1 , 'Dupont');
INSERT INTO Client VALUES ( 2);
INSERT INTO Client VALUES (3 , 'Durand');

DROP table telephone;
CREATE TABLE telephone
( CLI_ID   int,
TEL char(18));
INSERT INTO telephone VALUES (1,'05-59-45-72-42' );
INSERT INTO telephone VALUES (3,'01-44-28-52-50' );
INSERT INTO telephone VALUES (3,'06-54-18-51-90' );

DROP table email;
CREATE TABLE  email
( CLI_ID   int,
 EML_ADRESSE  char(14));
INSERT INTO email VALUES (1,'dupe@free.fr' );
INSERT INTO email VALUES (1,'dd@hotmail.com' );
INSERT INTO email VALUES (2,'mm@free.fr' );

DROP table adresse;
CREATE TABLE   adresse
( CLI_ID   int,
 ADR_VILLE CHAR(20));
INSERT INTO adresse VALUES (2,'Nice' );
INSERT INTO adresse VALUES (2,'Paris' );
INSERT INTO adresse VALUES (4,'Pau' );
```

On veux contacter tous les clients, quelque soit le mode de contact, dans le cadre d'une campagne publicitaire. Une reponse contenant tous les clients, meme ceux qui n'ont pas de telephone, d'e-mail ou d'adresse est donc souhaitee.

Ecrire les requêtes qui génèrent la tables suivante:

*/

```
 cli_id |   cli_nom    |        tel         |  eml_adresse   |      adr_ville
--------+--------------+--------------------+----------------+----------------------
      1 | Dupont       | 05-59-45-72-42     | dupe@free.fr   |
      1 | Dupont       | 05-59-45-72-42     | dd@hotmail.com |
      2 |              |                    | mm@free.fr     | Nice
      2 |              |                    | mm@free.fr     | Paris
      3 | Durand       | 01-44-28-52-50     |                |
      3 | Durand       | 06-54-18-51-90     |                |
      4 |              |                    |                | Pau
(7 rows)
```

```
SELECT *
FROM Client C
   FULL  JOIN telephone T
        USING(CLI_ID)
   FULL  JOIN email E
        USING(CLI_ID)
   FULL  JOIN adresse A
        USING(CLI_ID) ;
```