

```

CREATE TABLE si4(groupe int,nom char(10),ddn date);
INSERT INTO si4 VALUES(1, 'Paul','1989-12-09');
INSERT INTO si4 VALUES(1, 'Pierre','1991-12-09');
INSERT INTO si4 VALUES(2, 'Paul','1988-12-09');
INSERT INTO si4 VALUES(2, 'Pierre','1989-12-09');
INSERT INTO si4 VALUES(2, 'Jean','1987-12-09');
INSERT INTO si4 VALUES(3, 'Paul','1988-12-09');

-- Calculer pour chaque groupe le nombre d'étudiants
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;

/*
  groupe | count |      min      |      max
-----+-----+-----+-----
        1 |      2 | 1989-12-09 | 1991-12-09
        3 |      1 | 1988-12-09 | 1988-12-09
        2 |      3 | 1987-12-09 | 1989-12-09
(3 rows)

*/

-- 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';

/*
  departure | totaltime | arrival
-----+-----+-----
SF          | 10:30:00 | NY
SF          | 11:00:00 | NY
(2 rows)*/

-- Recherche des vols avec le nombre d'escales

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;

/*
  departure | escales | arrival
-----+-----+-----
SF          |      2 | NY
DEN         |      2 | NY
SF          |      2 | CHI
SF          |      3 | NY
(4 rows)

```

```

*/
-- Calcul des nombres pairs inferieurs ou égal a 100
WITH RECURSIVE t(n) AS (
    SELECT 2 -- VALUES (2)
    UNION
    SELECT n+2 FROM t WHERE n < 100
)
-- SELECT n FROM t;

/*
n
-----
2
4
6
8
..
100
*/
SELECT count(*), sum(n), max(n), min(n) FROM t;
/*
sum | max | min
-----+-----+-----
2550 | 100 | 2
(1 ligne)
*/
-- SELECT COUNT(*) FROM t GROUP by n;
/*
1
...
1
1
1
(50 rows)
*/

--- Avec une vue
CREATE VIEW view_t(n) AS (
    WITH RECURSIVE t(n) AS
    ( SELECT 2 -- VALUES (2)
    UNION
    SELECT n+2 FROM t WHERE n < 100)
    SELECT n FROM t
);
SELECT count(*), sum(n), max(n), min(n) FROM view_t;

```

```

-- Recherche des ascendants

```

```

DROP TABLE IF EXISTS Parents;
CREATE TABLE Parents
( Pere CHAR(10),
  Mere CHAR(10),
  Enfant CHAR(10));

```

```

-- population de la table
INSERT INTO Parents VALUES ('Clara', 'Jean', 'Paul');
INSERT INTO Parents VALUES ('Jean', 'Nadia', 'Claire');
INSERT INTO Parents VALUES ('Blandine', 'Paul', 'Zoe');
INSERT INTO Parents VALUES ('Claire', 'Zoile', 'Louis');
INSERT INTO Parents VALUES ('Zoe', 'Yvan', 'Julia');
INSERT INTO Parents VALUES ('Louis', 'Julia', 'Adrien');

```

```

/*
    Blandine

```

```
CREATE TABLE telephone( CLI_ID    int,TEL char(18));
INSERT INTO telephone VALUES (1,'05-59-45-72-42');
```

```

/*
On veut 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.
*/

```

```

SELECT *
FROM Client
FULL OUTER JOIN telephone
    USING (CLI_ID)
FULL OUTER JOIN email
    USING (CLI_ID)
FULL OUTER JOIN adresse
    USING (CLI_ID) ;

```

```

/*

```

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 cli_id, cli_nom, tel, eml_adresse, adr_ville
FROM Client
LEFT OUTER JOIN telephone
    USING (CLI_ID)
LEFT OUTER JOIN email
    USING (CLI_ID)
LEFT OUTER JOIN adresse
    USING (CLI_ID) ;

```

```

/*

```

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		

```

(6 rows)
*/

```

```

/* DS AUTRE

```

5. Remarquant que la date de naissance de certains auteurs est aussi la date de la mort d'un ou de plusieurs autres auteurs, lesquels à leur tour peuvent être nés le jour de la mort d'un ou plusieurs auteurs on désire afficher pour chaque auteur combien de fois on peut remonter dans le temps sur ces coïncidences de dates. Par exemple si a est né le jour de la mort de b et de c (et de personne d'autre) et que b est né le jour de la mort de d, que ni c ni d ne sont né le jour de la mort d'un auteur pour l'auteur a on peut remonter deux fois dans le temps.*/

```

WITH RECURSIVE PL(Idauteur, Idancetre, Generation) AS (
    SELECT A1.IdAuteur, A1.IdAuteur, 0 FROM auteur A1
    union
    Select PL.Idauteur, A2.Idauteur, Generation+1 from PL, auteur A2 , auteur A3
    WHERE A3.IdAuteur=PL.IdAncetre and A3.Datenaissance=A2.DateDeces)

SELECT Idauteur, Max(Generation) FROM PL group by Idauteur;

```