

# Bases de Données Relationnelles

## TP1 : Exercices de sélection et de vérifications de contraintes en SQL

### SI2 et MAM2

La base de données TP1 est constituée de 6 tables que vous devez découvrir. Ces tables ne vous appartiennent pas, mais vous sont accessibles en lecture (select). Vous pouvez accéder directement à cette base de donnée par la connection :

```
% psql -h dbms.essi.fr -U nomutilisateur tp1
```

Pour découvrir la structure de ces tables :

```
tp1=> \d
List of relations
Schema | Name      | Type  | Owner
-----+-----+-----+-----
public | classe    | table | claudine
public | enr       | table | claudine
public | marque    | table | claudine
public | pays      | table | claudine
public | societe   | table | claudine
public | vente     | table | claudine
(6 rows)

tp1=> \d classe
Table "public.classe"
Column      |          Type          | Modifiers
-----+-----+-----+-----
num         | integer                | not null
libelle     | character varying(30) | not null
Indexes:
    "classe_pkey" PRIMARY KEY, btree (num)

.....
```

## 1 interrogations

Poser les questions suivantes (dans les réponses, les marques devront être identifiées par leur nom, classe et pays; et les sociétés devront être identifiées par leur nom et pays) :

1. Les marques (avec nom/classe/pays), classées par pays puis par classe.

---

```
select M.nom AS NomMarque, M.pays, M.classe
from marque M
order by 3, 2;
-- ou order by M.pays, M.classe;
```

```

/*
      nommarque      | pays | classe
-----+-----+-----
    Shalimar        | DE   |      1
    Passion          | DE   |      1
..
(92 rows) */

```

- 
2. Les marques de chaque propriétaire (ne pas sortir les sociétés sans marques).
- 

```

select M.nom AS NomMarque , S.nom AS NomPays ,    S.pays
from marque M, societe S
where M.prop=S.id
order by S.id;
/*
      nommarque      |      nompays      | pays
-----+-----+-----
    Coca Cola        | Coca Cola, Inc.    | US
..
(92 rows) */

/*

```

- 
3. Les sociétés qui ne possèdent aucune marque.
- 

```

select S.nom, S.pays  from societe S
where not exists (select * from marque  M where M.prop=S.id);

% ou, encore

select S.nom , S.pays  from societe S
except
select S.nom , S.pays  from societe S, marque  M where M.prop=S.id;
/*
              nom              | pays
-----+-----
    SUN MicroSystems, Inc.      | US
    Ste. SUN                    | FR
REPONSES:
..
(32 rows) */

```

- 
4. Les marques vendues avant leur enregistrement(ce qui est incorrect!).
-

```

select distinct M.nom , M.pays, M.classe
from marque M, enr E, vente V
where M.id=E.marque and V.marque=M.id and V.date_vente<E.date_enr;

-- ou
select M.nom , M.pays, M.classe
from marque M
where exists (select * from enr E, vente V
              where M.id=E.marque and V.marque=M.id and V.date_vente<E.date_enr);
/*
      nom      | pays | classe
-----+-----+-----
Lanvin         | FR   |      11
Lanvin         | UK   |      11
Lanvin         | US   |      11
Orangina       | US   |       4
PEPSI          | FR   |       4
Passion        | US   |       1
Pepsi Cola     | FR   |       4
(7 rows) */

```

---

5. Les marques non enregistrées, classées par pays.

```

select M.nom, M.pays, M.classe from marque M
except
select M.nom, M.pays, M.classe from marque M, enr E
where M.id=E.marque order by 2;

-- ou encore

select M.nom, M.pays, M.classe from marque M
where not exists (select * from enr E where M.id=E.marque )
order by 2;

/*
      nom      | pays | classe
-----+-----+-----
Cortisol       | DE   |      10
ORACLE7        | DE   |       6
Cola           | FR   |       4
Cortisol       | FR   |      10
Dune           | FR   |       4
ORACLE7        | FR   |       6
Cortisol       | MX   |      10
Coup de Coeur  | MX   |      11
Drobsol        | MX   |      10
Dune           | MX   |       1
Exorcyl        | MX   |      10
Miss Dior      | MX   |       1
Cortislo       | UK   |      10
(13 rows)
*/

```

---

6. Les propriétaires avec leur nombre de marques.

```

select S.nom AS NomSoc, S.pays, count(*) AS Nb
from marque M, societe S
where M.prop=S.id
group by S.Id, S.nom, S.pays ;
/*

```

nomsoc	pays	nb
Coca Cola, Inc.	US	10
Ste. Coca Cola	FR	3
Coca Cola Gmbh.	DE	5
ORACLE Corp.	US	5
Ste. ORACLE	FR	1
ORACLE Ltd.	UK	1
ORACLE Gmbh.	DE	1
Ste.Christian DIOR	FR	10
Ste. L.V.H.M.	FR	8
Ste. GUERLAIN	FR	9
Lab. Roussel	FR	8
Helena Rubinstein Ltd.	US	1
Hoescht Gmbh.	DE	7
PEPSI Inc.	US	7
Ste. Danone	FR	5
Ste. Orangina	FR	6
Pepsi Gmbh.	DE	2
Soc. Perfumes y Trajes	MX	3

(18 rows)

```

*/

```

- 
7. Les marques homonymes (même nom et même classe) dans des pays différents et avec des propriétaires différents.
- 

```

select M1.nom AS NOM_M, M1.classe AS CLASSE, M1.pays AS PAYS_M1,
M1.prop AS PROP1, M2.pays AS PAYS_M2, M2.prop AS PROP2
from marque M1, marque M2
where M1.nom =M2.nom and M1.classe=M2.classe and M1.prop<>M2.prop
and M1.pays<M2.pays ;

```

```

/*
On notera que pour eliminer les doublons symetriques on a utilise < plutot
que <> pour la derniere condition

```

nom_m	classe	pays_m1	prop1	pays_m2	prop2
Coca Cola	4	DE	3	FR	1
Coca Cola	4	DE	3	UK	1
Coca Cola	4	DE	3	US	1
...					

(54 rows)

```

*/

```

- 
8. Les propriétaires dont les marques déposées (i.e., qui sont dans la table marque) ne sont pas toutes enregistrées.
-

```
select distinct S.nom, S.pays from societe S, marque M
where S.id=M.prop and not exists (select * from enr E where E.marque=M.id );
```

-- ou encore

```
select distinct S.nom, S.pays
from societe S, marque M
where S.id=M.prop
and
M.id <> all (select E.marque from enr E );
```

```
/*
      nom          | pays
-----+-----
Hoescht Gmbh.      | DE
Lab. Roussel       | FR
ORACLE Corp.       | US
Soc. Perfumes y Trajes | MX
Ste. Coca Cola     | FR
Ste. Danone        | FR
(6 rows)
*/
```

---

9. Les propriétaires dont les marques (i.e., qui sont dans la table marque) sont toutes enregistrées.

---

```
select distinct S.nom, S.pays from societe S, marque M
where S.id=M.prop and not exists (select * from marque MM where
S.id=MM.prop and not exists (select * from enr E where E.marque=MM.id ));
```

-- ou

```
select S.nom, S.pays from societe S, marque M where S.id=M.prop
except
select S.nom, S.pays from societe S, marque M
where S.id=M.prop and not exists (select * from enr E where E.marque=M.id );
```

```
/*
      nom          | pays
-----+-----
Coca Cola Gmbh.    | DE
Coca Cola, Inc.    | US
Helena Rubinstein Ltd. | US
ORACLE Gmbh.       | DE
ORACLE Ltd.        | UK
PEPSI Inc.         | US
Pepsi Gmbh.        | DE
Ste. GUERLAIN      | FR
Ste. L.V.H.M.      | FR
Ste. ORACLE        | FR
Ste. Orangina      | FR
Ste.Christian DIOR | FR
(12 rows)
*/
```

- 
10. Dans chacune des classes, le/les propriétaires ayant le plus grand nombre de marque dans cette classe.
- 

```
DROP view countproprio;
create view countproprio AS
(select M.classe AS classe, M.prop AS proprio, count(*) AS compte
 from marque M
 group by M.classe, M.prop );

select CM1.classe AS CLASSE, CM1.compte AS "COMPTE MAXI",
S.nom AS "NOM SOC.", S.pays AS "PAYS SOC."
from societe S, countproprio AS CM1,
(select CPC.classe AS classe, max(CPC.compte) AS compte_max
 from countproprio AS CPC group by CPC.classe
) AS CM2
where CM1.classe = CM2.classe and CM1.compte=CM2.compte_max and S.id=CM1.proprio order by 1;

/*
classe | COMPTE MAXI | NOM SOC. | PAYS SOC.
-----+-----+-----+-----
      1 |           8 | Ste. GUERLAIN | FR
      4 |          10 | Coca Cola, Inc. | US
      6 |           5 | ORACLE Corp. | US
     10 |           8 | Lab. Roussel | FR
     11 |           8 | Ste. L.V.H.M. | FR
(5 rows)
*/
```

ou

```
select classe, S.nom, S.pays, count( * )
 from marque as M1, societe AS S
 where M1.prop = S.id
 group by classe, S.nom, S.pays
 having count( * ) >= all
(select count ( * ) from marque AS M2
 where M2.classe = M1.classe
 group by prop);
```

---

## 2 Contraintes

Vérifier les contraintes suivantes (donnez les marques qui violent éventuellement chacune des contraintes)

1. Le pays d'une marque est le même que le pays où elle est enregistrée
- 

```
select M.nom, M.classe, M.pays, E.pays
from marque M, enr E where M.id=E.marque and E.pays<>M.pays;

nom | classe | pays | pays
-----+-----+-----+-----
Cortisol | 10 | US | MX

-- (1 ligne)
```

- 
2. Le vendeur d'une marque est soit le déposant (et c'est la première vente de cette marque) , soit l'acquéreur de la vente précédente.
- 

Il faut essayer de trouver un vendeur d'une marque qui viole cette contrainte :  
- soit parce c'est la premiere vente de cette marque et qu'il n'en est pas le deposant  
- soit parce qu'il n'est pas l'acquireur de la vente precedente

La difficulte est de trouver la vente precedente !!!!!  
\*/

```
-- Le premier vendeur n'est pas le déposant
select V.vendeur, V.marque, V.date_vente from vente V
where not exists (select * from vente V1
where V1.marque=V.marque and V1.date_vente<V.date_vente)
and not exists (select * from enr E
where E.deposant=V.vendeur and E.marque=V.marque)
union
-- Le vendeur n'est pas l'acquéreur de la vente précédente
select V1.vendeur, V1.marque, V1.date_vente from vente V1, vente V2
where V1.marque=V2.marque and V1.date_vente<V2.date_vente
and V1.acquereur <> V2.vendeur
and not exists (select * from vente V3 -- la marque n'a pas été vendu entre V1 et V2
where V3.marque=V2.marque and
V1.date_vente< V3.date_vente and
V3.date_vente< V2.date_vente );
```

-- \*\* ou encore

```
-- Le premier vendeur n'est pas le déposant
select V.vendeur, V.marque, V.date_vente from vente V
where
V.marque != all (select V1.marque from vente V1 -- Il n'y a pas eu de vente avant V
where V1.date_vente<V.date_vente)
and
V.vendeur <> (select E.deposant from enr E -- Le vendeur n'est pas le déposant
where E.marque=V.marque)
```

union all

```
-- Le vendeur n'est pas l'acquéreur de la vente précédente
select V.vendeur, V.marque, V.date_vente from vente V, vente V1
where V1.marque=V.marque
and V1.date_vente<V.date_vente
and V1.acquereur <> V.vendeur
and V.marque != all (select V2.marque from vente V2
where V1.date_vente< V2.date_vente and
V2.date_vente< V.date_vente );
```

-- \*\* ou encore

```
-- Le premier vendeur n'est pas le déposant
select V.vendeur, V.marque, V.date_vente from vente V
where
V.date_vente = (select Min(V1.date_vente) from vente V1 -- Il n'y a pas eu de vente avant
group by V1.marque
having V1.marque=V.marque)
and
V.vendeur <> (select E.deposant from enr E -- Le vendeur n'est pas le déposant
where E.marque=V.marque)
union all
```

```
-- Le vendeur n'est pas l'acquéreur de la vente précédente
select V.vendeur, V.marque, V.date_vente from vente V, vente V1
where      V1.marque=V.marque
      and V1.date_vente<V.date_vente
      and V1.acquereur <> V.vendeur
      and V.marque != all (select V2.marque from vente V2
      where      V1.date_vente< V2.date_vente and
      V2.date_vente< V.date_vente );
```

```
/*
vendeur | marque | date_vente
-----+-----+-----
      5 |      3 | 1980-02-05
     12 |     30 | 2068-03-07
(2 lignes)

*/
```

---

3. Le pays d'une marque est le même que celui de son propriétaire.

---

```
select M.nom, M.classe, M.pays, S.pays from marque M, societe S
where M.prop=S.id and S.pays<>M.pays;
/*
      nom      | classe | pays | pays
-----+-----+-----+-----
Coca Cola      |      4 | FR   | US
Coca Cola Light |      4 | FR   | US
...
(48 rows)
```

---

Annexe : Description des tables

Table "public.classe"

<i>Column</i>	<i>Type</i>	<i>Modifiers</i>
num	integer	not null
libelle	character varying(30)	not null

Table "public.enr"

<i>Column</i>	<i>Type</i>	<i>Modifiers</i>
marque	integer	not null
num	integer	not null
pays	character(2)	not null
deposant	integer	not null
date_enr	date	not null

Table "public.marque"

<i>Column</i>	<i>Type</i>	<i>Modifiers</i>
id	integer	not null
nom	character varying(30)	not null
classe	integer	not null
pays	character(2)	not null
prop	integer	not null

Table "public.societe"



<i>Column</i>	<i>Type</i>	<i>Modifiers</i>
id	integer	not null
nom	character varying(40)	not null
ville	character varying(20)	
pays	character(2)	not null

Table "public.vente"

<i>Column</i>	<i>Type</i>	<i>Modifiers</i>
marque	integer	not null
vendeur	integer	not null
acquereur	integer	not null
date_vente	date	not null

Table "public.pays"

<i>Column</i>	<i>Type</i>	<i>Modifiers</i>
code	character(2)	not null
nom	character varying(50)	not null