

Bases de Données Relationnelles

TP1

MAM4 - SI3

Le schéma de la base de données du TP1 est donné en Annexe.

Dans les réponses attendues, les marques devront être identifiées par leur nom, classe et pays, et les sociétés devront être identifiées par leur nom et pays.

1. Afficher le contenu complet de la table pays

```
SELECT * FROM PAYS
```

2. Afficher le contenu complet de la table pays mais par ordre alphabétique sur le nom des pays

```
SELECT * from pays order by nom
```

3. Afficher toutes les marques, c'est à dire, d'après la consigne leur nom, classe et pays, en respectant cet ordre sur les colonnes. Les réponses (lignes) devront être triées par pays, classe puis nom.

```
SELECT NOM, CLASSE, PAYS FROM MARQUE ORDER BY PAYS, CLASSE, NOM;
```

4. Afficher les propriétaires associés aux marques qu'ils possèdent.

- Continuez à respecter la consigne et afficher Nom et Pays pour les sociétés, et Nom, Classe, Pays pour les marques.
- Les résultats devront être ordonnés par Id de Société puis par Identifiant de Marque
- Les intitulés des colonnes de la sortie devront être dans l'ordre NomSociete, PaysSociete, NomMarque, PaysMarque et ClasseMarque

```
SELECT S.Nom AS NomSociete, S.pays AS PaysSociete,
M.nom AS NomMarque, M.pays AS PaysMarque, M.Classe AS ClasseMarque
FROM marque M JOIN societe S
ON M.prop=S.id
ORDER BY S.Id, M.Id;
```

5. On cherche à détecter une information incohérente en affichant les marques vendues avant leur enregistrement. Les réponses seront ordonnées par classe, puis par nom puis par pays. Les colonnes restent dans l'ordre nom, pays, classe.

```
SELECT distinct M.nom, M.pays, M.classe
FROM (marque M JOIN enr E ON M.id=E.marque)
JOIN vente V ON V.marque=M.id
WHERE V.date_vente<E.date_enr
ORDER BY M.classe, M.nom,M.pays;
```

6. Afficher le nom, le pays et la classe des marques non enregistrées, classées par pays, nom et classe.

```
SELECT M.nom, M.pays, M.classe from marque M
EXCEPT
SELECT M.nom, M.pays, M.classe
FROM marque M JOIN enr E ON M.id=E.marque
ORDER BY M.pays, M.nom, M.classe;
```

Autre solution:

```
SELECT M.nom, M.pays, M.classe from marque M
WHERE NOT EXISTS (
SELECT * FROM enr E WHERE M.id=E.marque )
ORDER BY M.pays, M.nom, M.classe;
```

Autre solution:

```
SELECT M.nom, M.pays, M.classe from marque M
WHERE M.id NOT IN (
SELECT E.marque FROM enr E )
ORDER BY M.pays, M.nom, M.classe;
```

7. Afficher les couples de marques de même nom et de même classe dans des pays différents et avec des propriétaires différents. Les entêtes de colonnes seront:

- *nom* (commun aux deux marques)
- *classe* (commune aux deux marques)
- *pays₁* (code du pays de la première marque)
- *prop₁* (identifiant du propriétaire de la première marque)
- *pays₂* (code du pays de la deuxième marque)
- *prop₂* (identifiant du propriétaire de la deuxième marque)

Ordonner les résultats par *nom*, *classe*, *pays₁*, *prop₁*, *pays₂*, *prop₂*.

```
SELECT nom, classe, M1.pays AS pays_1,
M1.prop AS prop_1, M2.pays AS pays_2, M2.prop AS prop_2
FROM marque M1 JOIN marque M2
USING (nom,classe)
WHERE M1.prop<>M2.prop AND M1.pays<M2.pays
ORDER BY nom, classe, pays_1, prop_1, pays_2, prop_2;
```

Pour éviter d'avoir deux fois les mêmes couples de marques, on doit tester que le pays de l'un est inférieur à celui de l'autre (selon l'ordre lexicographique) – et non simplement différent.

-
8. Trouver si elles existent les marques qui ne respectent pas la contrainte suivante: le pays d'une marque doit être le même que celui de son propriétaire. Colonnes à afficher :

- nom de la marque
- classe de la marque
- paysM de la marque
- paysS de la société

Ordonner par nom, classe, PaysM.

```
SELECT M.nom, M.classe, M.pays AS paysM, S.pays AS paysS
FROM marque M JOIN societe S ON M.prop=S.id
WHERE S.pays<>M.pays
ORDER BY M.nom, M.classe, M.pays;
```

9. Trouver si elles existent les marques qui violent la contrainte suivante : Le pays d'enregistrement d'une marque doit être le pays de la marque. Entêtes de colonnes attendues :

- nom
- classe
- paysMarque
- paysEnr

Ordonner par nom, classe, paysMarque.

```
SELECT  M.nom, M.classe, M.pays AS paysMarque, E.pays AS paysEnr
FROM    marque M JOIN enr E ON  M.id=E.marque
WHERE   E.pays<>M.pays
ORDER BY M.nom, M.classe, M.pays;
```

10. Afficher les sociétés qui possèdent des marques qui ne sont pas toutes enregistrées, classées par nom puis par pays.

C'est-à-dire les sociétés qui possèdent au moins une marque non enregistrée.

```
SELECT DISTINCT S.nom, S.pays
FROM    societe S JOIN marque M ON S.id=M.prop
WHERE   NOT EXISTS (SELECT * FROM enr E WHERE E.marque=M.id)
ORDER BY S.nom, S.pays;
```

Autre solution :

```
SELECT DISTINCT S.nom, S.pays
FROM    societe S JOIN marque M ON S.id=M.prop
WHERE   M.id NOT IN (SELECT E.marque FROM enr E)
ORDER BY S.nom, S.pays;
```

Attention on ne peut pas trouver de solution avec une différence (EXCEPT) entre deux tables: on calculerait alors l'ensemble des sociétés dont *aucune* marque n'est enregistrée car EXCEPT est une opération ensembliste.

Par contre on peut faire un EXCEPT entre deux tables où les lignes sont complètes (pas de projection sur le nom et le pays des sociétés) et considérer le résultat comme point de départ (dans le FROM) d'une requête qui fait simplement la projection sur les attributs voulus:

```
SELECT DISTINCT nom, pays
FROM (
    SELECT M.id, S.nom, S.pays
    FROM marque M JOIN societe S ON M.prop=S.id
    EXCEPT
    SELECT M.id, S.nom, S.pays
    FROM marque M JOIN enr E JOIN societe S ON M.id=E.marque AND M.prop=S.id)
ORDER BY nom, pays;
```

11. Afficher les propriétaires qui ne possèdent que des marques enregistrées, ordonnés par nom puis par pays. Un propriétaire est une société qui possède au moins une marque. Les entêtes de colonnes doivent être nom et pays.

On cherche donc le complémentaire des réponses précédentes parmi les propriétaires.

```
SELECT DISTINCT S.nom, S.pays FROM societe S
WHERE S.id IN (SELECT M.prop FROM marque M)
  AND NOT EXISTS (SELECT * FROM marque MM
                  WHERE S.id=MM.prop
                  AND MM.id NOT IN (SELECT E.marque FROM enr E))
ORDER BY S.nom, S.Pays;
```

Autre solution :

```
SELECT S.nom, S.pays FROM societe S JOIN marque M ON S.id=M.prop
EXCEPT
SELECT S.nom, S.pays FROM societe S JOIN marque M ON S.id=M.prop
WHERE NOT EXISTS (SELECT * FROM enr E WHERE E.marque=M.id);
```

Autre solution:

```
SELECT S.nom , S.pays
FROM societe S JOIN marque M ON M.prop = S.id
EXCEPT
SELECT S.nom, S.pays
FROM societe S JOIN (
    SELECT M.id, M.prop FROM marque M
    EXCEPT
    SELECT M.id, M.prop
    FROM marque M JOIN enr E ON M.id = E.marque) T
ON S.id = T.prop
ORDER BY S.nom, S.pays
```

-
12. Trouver si elles existent les marques qui ne respectent pas la contrainte suivante : Le vendeur d'une marque doit être le déposant s'il s'agit d'une première vente de la marque. Les colonnes à afficher sont :

- marque : identifiant de la marque concernée
- vendeur : identifiant du vendeur de la première vente de la marque
- déposant : déposant de la marque
- date-vente : date de la première vente

Les résultats doivent être ordonnés selon les identifiants de marque.

```
SELECT V.marque, V.vendeur, E1.deposant, V.date_vente
FROM vente V JOIN enr E1 ON V.marque=E1.marque
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 E2
    WHERE E2.deposant=V.vendeur and E2.marque=V.marque )
ORDER BY V.marque
```

-
13. On suppose qu'il y a au moins une marque dans chaque classe. Pour chaque classe (identifiée par son numéro), afficher le nombre de marques de la classe. Classer les résultats par ordre décroissant des numéros de classe. La colonne qui contient le nombre de marques devra s'appeler "Nombre de Marques".

```
SELECT classe, count (*) AS "Nombre_de_Marques"
FROM marque M
GROUP BY classe
ORDER BY classe DESC;
```

-
14. Pour chaque classe (identifiée par son numéro) afficher le nombre de pays dans lesquels il y a au moins une marque de la classe. Classer les résultats par ordre croissant des numéros de classe. La colonne qui affiche le nombre de pays devra s'appeler "Nombre de Pays"
-

```
SELECT classe , count(distinct pays) as "Nombre_de_Pays"
FROM marque
GROUP BY classe
ORDER BY classe;
```

15. Afficher les sociétés qui sont propriétaires d'au moins une marque, avec le nombre de marques dont elles sont propriétaires. Les sociétés seront classées par ordre alphabétique sur leur nom, puis sur leur pays. Les entêtes des colonnes attendues sont : nom, pays, "Nombre de Marques".
-

```
SELECT S.nom , S.pays , count(M.id) AS "Nombre_de_Marques"
FROM marque M JOIN societe S ON M.prop=S.id
GROUP BY S.Id, S.nom, S.pays
ORDER BY S.nom, S.Pays;
```

Les attributs S.nom et S.pays doivent figurer dans le GROUP By pour pouvoir figurer dans le SELECT.

16. Afficher les sociétés qui ne possèdent aucune marque. Classer les réponses par pays puis par nom de société. Les entêtes des colonnes attendues sont nom et pays.
-

```
SELECT nom, pays FROM societe S
WHERE S.id NOT IN (SELECT M.prop FROM marque M)
ORDER BY pays, nom;
```

Autre solution:

```
SELECT S.nom, S.pays FROM societe S
EXCEPT
SELECT S.nom, S.pays FROM societe S JOIN marque M ON M.prop=S.id
ORDER BY pays, nom;
```

Autre solution:

```
SELECT S.nom, S.pays FROM societe S
WHERE NOT EXISTS (
    SELECT * FROM marque M WHERE M.prop=S.id )
ORDER BY pays, nom;
```

17. Trouver si elles existent les ventes qui ne respectent pas la contrainte suivante : Le vendeur doit être l'acquéreur de la vente précédente. Les résultats seront classés selon les identifiants de marque puis les dates. Pour chaque résultat, afficher :

- l'identifiant de l'acquéreur de la vente précédente
 - l'identifiant du vendeur
 - l'identifiant de la marque
 - la date de la vente
-

```
SELECT V1.acquereur, V2.vendeur, V2.marque, V2.date_vente
FROM vente V1 JOIN vente V2 ON V1.marque=V2.marque
WHERE V1.date_vente<V2.date_vente
AND V1.acquereur <> V2.vendeur
AND NOT EXISTS (
    SELECT * FROM vente V3 -- la marque n'a pas ete vendue entre V1 et V2
    WHERE V3.marque=V2.marque AND
    V1.date_vente< V3.date_vente AND
    V3.date_vente< V2.date_vente )
ORDER by V2.marque, V2.date_vente;
```

Il ne suffit pas de tester que le vendeur d'une vente n'est pas l'acquéreur d'une vente précédente, il faut aussi vérifier qu'il n'y a pas eu de vente ces deux ventes. C'est ce que permet de vérifier la sous-requête.

18. Afficher pour chaque classe et chaque propriétaire le nombre de marques possédées par le propriétaire dans la classe. Ordonner les résultats selon la classe et le propriétaire. Les entêtes des colonnes attendues sont : classe, prop et NB.
-

```
SELECT M.classe, M.prop, count(*) AS NB
FROM marque M
GROUP BY M.classe, M.prop
ORDER BY classe, prop
```

19. Afficher pour chaque classe, le ou les propriétaires possédant le plus grand nombre de marques. Pour cela, créer d'abord une vue qui "nomme" le résultat de la requête de la question précédente:

```
DROP VIEW IF EXISTS NBM_proprio;
CREATE view NBM_proprio AS (
    SELECT M.classe, M.prop, COUNT(*) AS NB
    FROM marque M
    GROUP BY M.classe, M.prop);
```

Puis écrire une requête dont les entêtes des colonnes du resultat sont : classe, NB, nom et pays. Trier les résultats selon la classe.

```
DROP VIEW IF EXISTS NBM_proprio;
CREATE view NBM_proprio AS (
    SELECT M.classe, M.prop, COUNT(*) AS NB
    FROM marque M
    GROUP BY M.classe, M.prop);

SELECT distinct NBM1.classe, NBM1.NB, S.nom, S.pays
FROM NBM_proprio AS NBM1 JOIN societe S
ON S.Id = NBM1.prop
WHERE NBM1.NB = (
    SELECT MAX(NBM2.NB)
    FROM NBM_proprio AS NBM2 WHERE NBM2.classe=NBM1.classe)
ORDER BY classe
```

La sous-requête permet de calculer le nombre max de marques possédées par un même propriétaire dans la classe de la ligne (donc du propriétaire) testée dans la requête principale.

Autre solution:

```
DROP VIEW IF EXISTS NBM_proprio ;
CREATE view NBM_proprio AS
SELECT M.classe, M.prop, count (*) AS NB
FROM marque M
GROUP BY M. classe, M. prop ;

SELECT NBM1.classe, NBM1.NB, S.nom, S. pays
FROM NBM_proprio AS NBM1 JOIN societe S
ON S.Id = NBM1.prop
GROUP BY NBM1.classe
HAVING NBM1.NB = MAX(NBM1.NB) ;
```

On peut avoir plusieurs propriétaires par classes s'ils possèdent le même nombre (max) de marques.

1 Annexe : schéma de la base

```
CREATE TABLE marque (  
    id integer PRIMARY KEY,  
    nom character varying(30) NOT NULL,  
    classe integer NOT NULL,  
    pays character(2) NOT NULL,  
    prop integer NOT NULL  
);  
  
CREATE TABLE societe (  
    id integer NOT NULL,  
    nom character varying(40) NOT NULL,  
    ville character varying(20),  
    pays character(2) NOT NULL  
);  
  
CREATE TABLE enr (  
    marque integer PRIMARY KEY,  
    num integer NOT NULL,  
    pays character(2) NOT NULL,  
    deposant integer NOT NULL,  
    date_enr date NOT NULL  
);  
  
CREATE TABLE vente (  
    marque integer NOT NULL,  
    vendeur integer NOT NULL,  
    acquereur integer NOT NULL,  
    date_vente date NOT NULL  
);  
  
CREATE TABLE classe (  
    num integer PRIMARY KEY,  
    libelle character varying(30) NOT NULL  
);  
  
CREATE TABLE pays (  
    code character(2) PRIMARY KEY,  
    nom character varying(50) NOT NULL  
);
```