

Corrigé du premier TP

Bases de Données

Partie I : Schéma relationnel et Méta-base

- Vérifiez la structures des relations suivantes cat, obj, cols, syn de la méta-base.

```
SQL> desc cat
Name Null? Type
-----
TABLE_NAME NOT NULL VARCHAR2(30)
TABLE_TYPE VARCHAR2(11)

SQL> desc obj
SQL> desc cols
SQL> desc syn
```

- Déterminez la structures des différentes relations dont vous êtes propriétaire.

```
SQL> set pages 24
SQL> select * from cat;
SQL> select * from cat
2 where table_type='TABLE'
3 /

TABLE_NAME TABLE_TYPE
-----
ADMETS TABLE
BUVEURS TABLE
PRODUCTEURS TABLE
RECOLTES TABLE
VINS TABLE

SQL> desc achats
Name Null? Type
-----
NB NOT NULL NUMBER(3)
NV NOT NULL NUMBER(3)
DAT NOT NULL DATE
LIEU VARCHAR2(25)
QTE NUMBER(4)
```

```
SQL> desc buveurs
SQL> desc producteurs
SQL> desc recoltes
SQL> desc vins
```

- Obtenir les mêmes informations que celles obtenues par la commande desc à l'aide d'une requête SQL sur la relation cols de la méta-base.

```
SQL> col Name format A31
SQL> col Null? format AB
SQL> col Type format A12
SQL> select column_name "Name",
2 decode(nullable, 'N', 'NOT NULL', ' ') "Null?",
3 data_type || '(' || data_length || ')' "Type"
4 from cols
5 where table_name='BUVEURS'
6 /
```

```
Name Null? Type
-----
NB NOT NULL NUMBER(22)
NV NOT NULL VARCHAR2(20)
PRESONS VARCHAR2(20)
TYPE VARCHAR2(31)
```

```
SQL> clear columns
columns cleared
```

Partie II : Interrogations SQL

- Quels sont les différents types de buveurs ?

```
SQL> select distinct type from B;

TYPE
-----
gros
moyen
petit
```

- Lister les vins triés par crus croissants et millésimes décroissants ?

```
SQL> set pag on
SQL> select cru, mill, nv, degre
2 from v
3 order by cru, mill desc ;
```

- Quelles sont les régions de production de Pommard ou de Brouilly ?

```
SQL> select distinct region, cru
2 from P,R,V
3 where p.region=p and r.region=r
4 and cru in ('Pommard','Brouilly');

REGION CRU
-----
Bourgogne Brouilly
Bourgogne Pommard
Rhône Pommard
```

- Quelles sont les régions de production de Pommard et de Brouilly ?

```
SQL> select distinct P1.region
2 from P P1, P P2, R R1, R R2, V V1, V V2
3 where P1.region=R1.region and R1.region=V1.region and V1.region='Pommard'
4 and P2.region=R2.region and R2.region=V2.region and V2.region='Brouilly'
5 and P1.region=P2.region ;
```

```
REGION
-----
Bourgogne
```

- Quantités achetées par cru et par millésime ?

```
SQL> select cru, mill, sum(qte)
2 from v, a
3 where v.region=a.region
4 group by cru, mill ;
```

- Retrouver les numéros des vins produits par plus de trois producteurs.

```
SQL> select nv
2 from r
3 group by nv
4 having count(np)>=3 ;
```

```
NV
--
12
43
45
54
67
69
74
78
79
87
88
88
```

- Quels sont les producteurs qui ne produisent pas ?

```
SQL> select np
2 from p
3 where np not in (select np from R) ;
```

- Quels sont les buveurs qui ont acheté au moins un vin de millésime 1980 ?

```
SQL> select nomb, cru, mill, dat
2 from b, a, v
3 where v.mill=1980
4 and b.nomb = a.nomb and v.region = a.region
5
```

```
NOMB CRU MILL DAT
-----
Anouilh Mercury 1980 19-SEP-87
Aragon Mercury 1980 15-SEP-82
Cotteau Seyssel 1980 24-DEC-87
Corneille Corneille 1980 20-MAR-88
Daute Chateau Corton Grancey 1980 31-DEC-84
Daute Volnay 1980 25-DEC-78
```

- Quels sont les buveurs qui n'ont acheté aucun vin de millésime 1980 ?

```
SQL> select nomb, prenomnb
2 from b
3 where nb not in ( select a.nb
4 from a, v
5 where v.mill=1980 and v.region = a.region ) ;
```

- Quels sont les buveurs qui n'ont acheté que des vins de millésime 1980 ?

```
SQL> col nb format 90
SQL> select nb, nomb
2 from b
```

```

3 where nb in ( select nb
4               from a, v
5               where v.nv = a.nv and v.mill = '1980' )
6 and nb not in ( select nb
7                 from a, v
8                 where v.nv=a.nv and v.mill <> '1980' ) ;

NB NOMB
-----
23 Corneille
31 Dante

• Quels sont les buveurs qui ont acheté tous les vins de millésime 1980 ?

SQL> rem Ajout du buveur numéro 0 qui a acheté tous les vins de mill 1980
SQL> insert into b
2 values (0, 'Dark', 'Vador', 'CyberEponge');
SQL> rem Ajout des achats du buveur 0 pour chaque vin de mill 1980
SQL> insert into a
2 select 0, nv, sysdate, 'Tataouine', 1
3 from v
4 where mill = '1980' ;

SQL> rem Recherche des buveurs qui ont acheté tous les vins de mill 1980
SQL> select nb, nomb || presomb "nom"
2 from b
3 where not exists ( select nv
4                   from v
5                   where mill='1980'
6                   and not exists ( select * from a
7                                   where a.nv=v.nv and a.nb=b.nb ) )
8 /

```

```

NB non
-----
0 Buveur0

SQL> rem Une autre façon de faire
SQL> select nb, nomb
2 from b
3 where nb not in ( select nb
4                   from b, v v1
5                   where v1.mill = '1980'
6                   and nb not in ( select nb
7                                   from a
8                                   where a.nv = v1.nv ) )
9 /

SQL> rem Encore une autre façon de faire
SQL> select nb, nomb
2 from b
3 where nb not in ( select nb
4                   from b, v
5                   where mill = '1980'
6                   and (nb, nv) not in ( select nb, nv from a ) ) ;

```

Partie III : Création et interrogations de vues

- Créez la vue `bons_buveurs` des buveurs de type 'gros' ou 'moyen'.

```

SQL> create view bons_buveurs as
2 select * from buveurs
3     where type = 'gros' or type = 'moyen' ;

SQL> Rem pour vérifier que le code SQL associé à la vue Bons_Buveurs :
SQL> set long 1024
SQL> select text from user_views
2 where view_name = 'BONS_BUVEURS' ;

```

- Créez la vue `buveurs_asec` des buveurs qui n'ont jamais rien acheté.

```

SQL> create view buveurs_asec as
2 select * from buveurs
3     where nb not in (select nb from Achats) ;

```

- Créez la vue `buveurs_achats` complémentaire de la précédente.

```

SQL> create view buveurs_Achats as
2 select * from buveurs
3     where nb in (select nb from Achats);

SQL> create view buveurs_Achats2 as
2 select * from buveurs
3     where nb not in (select nb from buveurs_asec) ;

SQL> create view buveurs_Achats3 as
2 select b.* from b, a
3     where b.nb=a.nb ;

```

- Créez une vue `q83pl` (LIEU, CRU, QTE_BUE) qui donne par LIEU et CRU les quantités totales achetées en 1983 par tous les buveurs. Peut on définir la même vue avec un ordre croissant sur l'attribut QTE ? Proposez une explication ?

```

SQL> create view q83pl (lieu, cru, qte_bue) as
2 select Achats.lieu, vins.cru, sum(Achats.qte)
3     from Achats, vins
4     where Achats.vins=vins.nv
5     and Achats.dat > '31-DEC-82'
6     and Achats.dat < '01-JAN-84'
7     group by Achats.lieu, vins.cru
8 /

```

SQL> Rem Ce n'est pas logique de définir une vue avec un ordre sur un attribut. Une vue est une table virtuelle et il n'y a pas d'ordre sur les tuples dans le modèle relationnel. Cependant, Oracle ne rejette pas la déclaration de vues avec un clause order by.

- Ecrivez une requête qui donne le vin de degré le plus élevé ? Avez vous besoin d'une vue pour réaliser cette requête ?

```

SQL> select * from v
2 where degre = (select max(degre) from v) ;

NV CRU      MILL      DEGRE
-----
68 Riesling 1962      14

SQL> Rem Pas besoin de vue !

```