

## TD-TP SEANCE 4 : MANIPULATION DES DONNEES

**Préparation.** Charger le schéma et la base de données en lançant le script CLEAN.sql.

**Répondez aux questions suivantes en SQL :**

1. Quelle est la liste des clients ?
2. Quelle est la liste des clients français triés par ordre alphabétique ?
3. Quels sont les noms des clients français ou anglais ?
4. Quelle est la liste des produits vendus à "Antonio" ?
5. Quels sont les noms des clients ayant commandé en 2002 ou 2003 ?
6. Quelle est la liste des fournisseurs de produits coutant de moins de 50 euros?
7. Quels fournisseurs offrent à la fois des produits à moins de 50 et à plus de 400 euros ?
8. Quels sont les fournisseurs qui fournissent le produit ‘Chef’ le moins cher ?
9. Quels sont les noms des fournisseurs qui fournissent le produit ‘Chef’ le moins cher ?
10. Pour chaque fournisseur, quelle est la liste des produits fournis par ordre alphabétique ?
11. Quels sont les fournisseurs qui sont aussi clients (même nom) ?
12. Quels sont les fournisseurs qui ne fournissent que des produits à moins de 50 euros?
13. Quels sont les fournisseurs qui ne fournissent aucun produit à moins de 50 euros?
14. Quels sont les fournisseurs qui fournissent tous les produits de moins de 50 euros?
15. Quels sont les noms des clients qui n'ont jamais passé de commandes ?
16. Quels sont les couples de fournisseurs (sémantiquement) différents ?
17. Quel est le nombre de produits dans la base ?
18. Quels sont les prix moyen et maximum des produits pour chaque type ?
19. Même question mais seulement pour les types où il y a plus de 10 produits.
20. Quelle est pour chaque client, la liste des produits commandés et leur nombre pour l'ensemble des commandes passées ?
21. Quel est, pour chaque ligne de commande, le prix total à payer (penser à la remise et à la quantité) ?
22. Quel est le total à payer par commande ?
23. Quel est le total à payer par client ?
24. Quels sont les clients qui ont commandé pour plus de 10000 euros?

```
CREATE TABLE CLI( -- Client
  NumCli number(5) Constraint PK_Cli PRIMARY KEY,
  NomCli char(20),
  Pays char(30),
  Tel char(15));
```

```
CREATE TABLE FOU( -- Fournisseur
  NumFou number(2) Constraint PK_FOU PRIMARY KEY,
  NomFou char(20),
  Pays char(30),
  Tel char(15));
```

```
CREATE TABLE COM( -- Commande
  NumCom number(5) Constraint PK_COM PRIMARY KEY,
  NumCli Number(5) Constraint COM_REF_CLI REFERENCES Cli, --ON DELETE CASCADE,
  FraisPort number(4),
  AnCom number(4));
```

```
CREATE TABLE PRO( -- Produit
  NumPro number(5) Constraint PK_PRO PRIMARY KEY,
  NumFou number(2) Constraint PRO_REF_FOU REFERENCES Fou, --ON DELETE CASCADE,
  NomPro char(20),
  TypePro char(10),
  PrixUnit number(3));
```

```
CREATE TABLE DET( -- Detail
  NumCom number(5) Constraint DET_REF_COM REFERENCES Com, --ON DELETE CASCADE,
  NumPro number(5) Constraint DET_REF_PRO REFERENCES Pro, --ON DELETE CASCADE,
  Qte number(5),
  Remise number(5),
  Constraint PK_DET PRIMARY KEY (NumCom, NumPro));
```

## TD SEANCE 4 : MANIPULATION DES DONNEES

1. Liste des clients ?

`SELECT * FROM cli ;`

2. Liste des clients français triés par ordre alphabétique ?

`SELECT * FROM cli WHERE Pays = 'France' ORDER BY nomcli;`

CLI: NumCli

->

COM: NumCom

->

DET: NumPro

->

PRO: NumPro

3. Noms des clients français ou anglais ?

`SELECT nomcli FROM cli WHERE Pays='France' OR Pays='Royaume-Uni';`

4. Quels sont les produits vendus à "Antonio"

`SELECT DISTINCT pro.numpro FROM cli, com, det, pro WHERE cli.numcli=com.numcli AND com.numcom = det.numcom AND det.numpro = pro.numpro AND nomcli = 'Antonio' ;`

5. Noms des clients ayant commandé en 2002 ou 2003

`SELECT DISTINCT nomcli FROM cli, com WHERE cli.numcli = com.numcli AND (ancom = 2002 OR ancom = 2003);`

6. Quels sont les fournisseurs de produits de prix < 50 Euros

`SELECT DISTINCT numfou FROM pro WHERE prixunit < 50;`

7. Y a-t-il des fournisseurs de produits de prix < 50 Euros et > 400 Euros ?

`SELECT DISTINCT P1.numfou FROM pro P1, pro P2 WHERE P1.numfou=P2.numfou AND P1.prixunit < 50 AND P2.prixunit > 400;`

`SELECT DISTINCT numfou FROM pro WHERE prixunit < 50`

`INTERSECT`

`SELECT DISTINCT numfou FROM pro WHERE prixunit > 400 ;`

1. Find the min prixunit  
2. Search for the Fou that has (1) product 'Chef' and prixunit is min value of prixunit

8. Numéro des fournisseurs qui fournissent le produit 'Chef' le moins cher ?

`SELECT numfou FROM pro WHERE nompro = 'Chef' AND prixunit = (SELECT MIN(prixunit) FROM pro WHERE nompro='Chef');`

9. Nom des fournisseurs qui fournissent le produit 'Chef' le moins cher ?

`SELECT DISTINCT nomfou FROM pro, fou WHERE nompro = 'Chef' AND prixunit = (SELECT MIN(prixunit) FROM pro WHERE nompro='Chef') AND pro.numfou = fou.numfou;`

10. Donner pour chaque fournisseur la liste des noms des produits par ordre alphabétique

`SELECT DISTINCT numfou, nompro FROM pro ORDER BY numfou, nompro;`

1. First order by NumFou  
2. Then order by NomPro

11. Y a-t-il des fournisseurs qui sont clients (même nom)

`SELECT nomfou, nomcli FROM cli, fou WHERE cli.nomcli = fou.numfou;`

12. Quels sont les fournisseurs qui ne fournissent que des produits de prix < 50 Euros

`SELECT DISTINCT numfou FROM pro`

`MINUS`

`SELECT DISTINCT numfou FROM pro WHERE prixunit > 50;`

13. Quels sont les fournisseurs qui ne fournissent aucun produits de prix < 50 Euros

`SELECT DISTINCT numfou FROM fou`

`MINUS`

`SELECT DISTINCT numfou FROM pro WHERE prixunit < 50;`

14. Quels sont les fournisseurs qui fournissent tous les produits de moins de 50 Euros

My response is at the end of the file

`SELECT numfou FROM fou WHERE NOT EXISTS ( f s.t. (FORALL p s.t. p.prix < 50, (f, p) IN PRO) = f s.t. NOEXIST p s.t. prix < 50, (f,p) NOTIN PRO`

`SELECT * FROM pro WHERE prixunit < 50 AND NOT EXISTS (`

`SELECT * from det WHERE det.numpro = pro.numpro and det.numfou = fou.numfou) ) );`

15. Noms des clients qui n'ont pas passé de commandes ?

```
SELECT DISTINCT nomcli FROM cli WHERE numcli NOT IN (SELECT numcli FROM com);
```

16. Couples des numéros de clients différents ?

```
SELECT C1.numcli, C2.numcli FROM cli C1, cli C2 WHERE C1.numcli > C2.numcli;
```

17. Nombre de produits dans la base

```
SELECT COUNT(*) nb_produits FROM pro;
```

18. Pour chaque type de produit, le prix moyen et le prix max

```
SELECT typepro, AVG(prixunit) prix_moyen, MAX(prixunit) prix_max FROM pro GROUP BY typepro ;
```

19. Même question mais uniquement pour les types où il y a plus de 10 produits.

```
SELECT typepro, AVG(prixunit) prix_moyen, MAX(prixunit) prix_max FROM pro GROUP BY  
typepro HAVING COUNT(*) > 10;
```

Qte have two keys

20. Donner pour chaque client, la liste des produits commandés et leur nombre pour l'ensemble des commandes passées.

```
SELECT cli.numcli, numpro, SUM(qte) nb_produits FROM cli, com, det WHERE cli.numcli = com.numcli AND  
com.numcom = det.numcom GROUP BY cli.numcli, numpro ORDER BY cli.numcli, numpro ;
```

21. Calculez, pour chaque ligne de commande, le prix total à payer. (Penser à la remise et à la quantité)

```
SELECT numcom, det.numpro, qte*prixunit*(1-remise/100) prix_total FROM det, pro WHERE  
det.numpro=pro.numpro;
```

22. Calculez le total à payer par commande

```
SELECT numcom, SUM(qte*prixunit*(1-remise/100)) total_commande FROM det, pro WHERE  
det.numpro=pro.numpro GROUP BY numcom;
```

23. Calculez le total à payer par client

```
SELECT numcli, SUM(qte*prixunit*(1-remise/100)) total_client FROM det, pro, com WHERE  
det.numpro=pro.numpro AND com.numcom=det.numcom GROUP BY numcli;
```

-- Attention : les clients qui n'ont pas commandé n'apparaissent pas !

24. Quels sont les clients qui ont commandé plus de 10000 Euros

```
SELECT numcli, SUM(qte*prixunit*(1-remise/100)) total_client FROM det, pro, com WHERE  
det.numpro=pro.numpro AND com.numcom=det.numcom GROUP BY numcli HAVING SUM(qte*prixunit*(1-  
remise/100)) > 10000;
```

My response for 14.

```
SELECT DISTINCT NumFou  
FROM PRO P1  
WHERE NOT EXISTS (  
  SELECT 1  
  FROM PRO P2  
  WHERE PrixUnit < 50  
  AND NOT EXISTS (  
    SELECT 1  
    FROM PRO P3  
    WHERE P1.NumFou == P3.NumFou  
    AND P2.NumPro == P3.NumPro  
  ));
```

