



Département Informatique
ADIA-IISE/2023-2024

Module Bases de données et modélisation
TP 4

Travaux pratiques 4
Requêtes avancées

Réaliser par :
TARZOUKT Abderrahim
EL RHYATI Aymane

Encadré par :
M.OUKDACH Yassine

LDD

1. Construire un utilisateur nommé tp4 et lui donner des droits pour créer des tables.

```
SQL> alter session set "_ORACLE_SCRIPT"=TRUE;
Session altered.

SQL> create user tp4 identified by aymane;
User created.

SQL> grant connect;
grant connect
*
ERROR at line 1:
ORA-00990: missing or invalid privilege

SQL> grant sysdba to tp4;
grant sysdba to tp4
*
ERROR at line 1:
ORA-01031: insufficient privileges

SQL> grant all privileges to tp4;
Grant succeeded.
```

2. Après avoir indiqué les clés primaires et étrangères, construire les tables ci-dessus dans le schéma de tp4.

Clés primaires : Usine : NU. Produit : NP. Fournisseur : NF.

Livraison : (NP, NU, NF).

Clés étrangères : Livraison : NU, NP, NF

- Table USINE/PRODUIT :

```
SQL> create table usine(nu number(20) constraint nu_usine primary key, nomu varchar2(20), ville varchar2(20));
Table created.

SQL> create table produit(np number(20) constraint np_produit primary key, nomp varchar2(20), couleur varchar2(20), poids number(10));
Table created.
```

- **Table livraison :**

```
SQL> create table livraison(np number constraint np_lv references produit(np),
2      nu number constraint nu_lv references usine(nu),
3      nf number constraint nf_lv references fournisseur(nf));
Table created.
```

- **Table FOURNISSEUR :**

```
SQL> create table fournisseur(nf number constraint nf_four primary key, nomf varchar2(20),statut varchar2(20), ville varchar2(20));
Table created.
```

LMD

3. Remplir les tables par des données vraisemblables.

- **Table usine :**

```
SQL> insert into usine values (1, 'usine1', 'agadir');
1 row created.

SQL> insert into usine values(2,'usine2','tanger');
1 row created.

SQL> insert into usine values (3, 'usine3' , 'tata');
1 row created.
```

- **Table LIVRAISON :**

```
SQL> insert into livraison values(1,1,1,10);
1 row created.

SQL> insert into livraison values(3,1,2,16);
1 row created.

SQL> insert into livraison values(2,3,1,30);
1 row created.
```

- **Table produit :**

```
SQL> insert into produit values (1,'samira', 'red',2);
1 row created.

SQL> insert into produit values (2, 'dior' , 'blue', 10);
1 row created.

SQL> insert into produit values (3, 'adidas' , 'yellow', 30);
1 row created.
```

- **Table livraison**

```
SQL> insert into livraison values(1,1,1,10);
1 row created.

SQL> insert into livraison values(3,1,2,16);
1 row created.

SQL> insert into livraison values(2,3,1,30);
1 row created.
```

LID

4. Donner en capitale le nom et la ville de toutes les usines.

```
SQL> select upper(nomu),upper(ville) from usine;

UPPER(NOMU)      UPPER(VILLE)
-----
USINE1           AGADIR
USINE2           TANGER
USINE3           TATA
```

5. Donner le numéro, le nom et la ville de toutes les usines dont le nom contient plast.

```
SQL> select * from usine
  2  where nomu like '%plast%' ;

no rows selected
```

6. Donner les numéros des fournisseurs qui approvisionnent l'usine n°1.

```
SQL> select NF from livraison where NU = 1 ;
```

NF
1

7. Donner le nom des fournisseurs qui approvisionnent l'usine n°1 en produit n°1.

```
SQL> select nomf from livraison natural join fournisseur where nu = 1 and np =1;
```

NOMF
fou1

8. Donner les numéros des fournisseurs qui approvisionnent les usines n°1 ou n°2.

```
SQL> select nf from livraison where nu = 1 or nu =2;
```

NF
1
2

9. Donner les numéros des usines qui ne reçoivent aucun produit du fournisseur n°1.

```
SQL> select nu from livraison where nf = 1 and quantite = 0;
```

no rows selected

10. Donner les numéros des usines qui s'approvisionnent chez le fournisseur n°3.

```
SQL> select NU from Livraison where NF = 3;
```

no rows selected

11. Donner les numéros des fournisseurs qui fournissent au moins un produit fourni par le fournisseur n°1.

```

SQL> SELECT DISTINCT L.NF
  2 FROM Livraison L
  3 WHERE EXISTS (
  4   SELECT nf
  5   FROM Livraison
  6   WHERE NF = 1 AND NP = L.NP AND NU = L.NU
  7 );

      NF
-----
      1

```

12. Donner les numéros des usines qui s'approvisionnent uniquement chez le fournisseur n°3.

```

SQL> SELECT NU
  2 FROM Livraison L1
  3 WHERE NF = 3
  4   AND NOT EXISTS (
  5    SELECT NF
  6    FROM Livraison L2
  7    WHERE L1.NU = L2.NU AND L2.NF <> 3
  8 );

no rows selected

```

13. Chercher pour chaque fournisseur le nom et le nombre d'usines qu'il approvisionne. L'affichage sera ordonné par nombre d'usines.

```

SQL> SELECT F.NomF, COUNT(L.NU) AS nbU
  2 FROM Fournisseur F
  3 JOIN Livraison L ON F.NF = L.NF
  4 GROUP BY F.NF, F.NomF
  5 ORDER BY COUNT(L.NU);

NOMF                                NBU
-----
fou2                                1
fou1                                2

```

14. Chercher le nom des fournisseurs qui approvisionne plus de 4 usines.

```

SQL> SELECT F.NomF
  2 FROM Fournisseur F
  3 JOIN Livraison L ON F.NF = L.NF
  4 GROUP BY F.NomF
  5 HAVING COUNT(DISTINCT L.NU) > 4;

no rows selected

```

15. Trouver le nom des produits fournis à plus de 4 usines d'Agadir.

```

SQL> SELECT P.NomP
  2 FROM Produit P
  3 JOIN Livraison L ON P.NP = L.NP
  4 JOIN Usine U ON L.NU = U.NU
  5 WHERE U.Ville = 'Agadir'
  6 GROUP BY P.NomP
  7 HAVING COUNT(DISTINCT L.NU) > 4;

no rows selected

```

16. Afficher pour chaque produit le nom (seule l'initiale en majuscule) et la quantité totale fournie aux usines.

```
SQL> SELECT UPPER(SUBSTR(P.NomP, 1, 1)) AS InitialeNomProduit, SUM(L.Quantite) AS QuantiteTotale
  2 FROM Produit P
  3 JOIN Livraison L ON P.NP = L.NP
  4 GROUP BY UPPER(SUBSTR(P.NomP, 1, 1));
no rows selected
```

17. Chercher, ordonnée par nom, les fournisseurs qui approvisionnent toutes les usines.

```
SQL> SELECT F.NomF
  2 FROM Fournisseur F
  3 WHERE NOT EXISTS (
  4     SELECT U.NU
  5     FROM Usine U
  6     WHERE U.NU NOT IN (
  7         SELECT L.NU
  8         FROM Livraison L
  9         WHERE L.NF = F.NF
 10     )
 11 )
 12 ORDER BY F.NomF;
no rows selected
```