

Filière d'excellence ADIA

**TRAVAUX PRATIQUE DE
Basse de Données 3**

Réaliser par :

MOUHIB Zakaria
GASSEM YOUNESSE

Encadré par :

M.OUKDACH Yassine

TP4

Q1* : Construire un utilisateur nommé tp4 et lui donner des droits pour créer des tables :

Create user tp4 identified by passtp4 quota 10 M on system;

```
SQL> CREATE USER tp4 IDENTIFIED BY masterkln QUOTA 10 M ON SYSTEM;
User created.
```

Grant create table to tp4 ;

```
SQL> grant create table to tp4;
Grant succeeded.
```

Q2* : Construire les tables ci-dessus dans le schéma de tp4 :

```
SQL> CREATE TABLE Usine(
  2  NU INT CONSTRAINT PKU PRIMARY KEY,
  3  NomU VARCHAR(20),
  4  Ville VARCHAR(20));
Table created.
```

```
SQL> CREATE TABLE Produit(
  2  NP INT CONSTRAINT PKP PRIMARY KEY,
  3  NomP VARCHAR(30),
  4  Couleur VARCHAR(10),
  5  Poids NUMBER(5,2));
Table created.
```

```
SQL> CREATE TABLE Fournisseur(
  2  NF INT CONSTRAINT PKF PRIMARY KEY,
  3  NomF VARCHAR(20),
  4  Statut VARCHAR(20),
  5  Ville VARCHAR(20));
Table created.
```

```
SQL> CREATE TABLE Livraison(
  2  NP INT,
  3  NU INT,
  4  NF INT,
  5  Quantite INT,
  6  CONSTRAINT FKP FOREIGN KEY(NP) REFERENCES produit ON DELETE CASCADE,
  7  CONSTRAINT FKU FOREIGN KEY(NU) REFERENCES usine ON DELETE CASCADE,
  8  CONSTRAINT FKF FOREIGN KEY(NF) REFERENCES fournisseur ON DELETE CASCADE);
Table created.
```

Q3* : Remplir les tables par des données vraisemblables :

```
SQL> INSERT INTO usine VALUES(1,'KLN','NEW YORK');
1 row created.

SQL> INSERT INTO usine VALUES(2,'FMB','TOKYO');
1 row created.

SQL> INSERT INTO produit VALUES(1,'Hoodies','Black',1);
1 row created.

SQL> INSERT INTO produit VALUES(2,'Sneakers','White',1);
1 row created.

SQL> INSERT INTO fournisseur VALUES(1,'Master','READY','Agadir');
1 row created.

SQL> INSERT INTO fournisseur VALUES(2,'Mirembe','NOT READY','Agadir');
1 row created.

SQL> INSERT INTO Livraison VALUES(2,1,1,1000);
1 row created.

SQL> INSERT INTO Livraison VALUES(1,2,2,700);
1 row created.

SQL> commit;
Commit complete.
```

Q4* : Donner en capitale le nom et la ville de toutes les usines :

Select upper(nomU) Nom_Usine, upper(ville) Ville from usine;

```
SQL> SELECT upper(nomU) NOM_USINE, upper(ville) VILLE FROM usine;

NOM_USINE          VILLE
-----
KLN                NEW YORK
FMB                TOKYO
US PLASTICS        LOS ANGELES
WEPLAST            SEOUL
LOLLIESB1          AGADIR
LOLLIESB2          AGADIR
LOLLIESB3          AGADIR
LOLLIESB4          AGADIR

8 rows selected.
```

Q5* : Donner le numéro, le nom et la ville de toutes les usines dont le nom contient «Plast» :

Select NU as Num_Usine, nomU as Nom_Usine, ville from Usine where nomU like '%Plast%';

```
SQL> SELECT NU AS NUM_USINE,nomU AS NOM_USINE,ville FROM usine WHERE nomU LIKE '%Plast%';
```

NUM_USINE	NOM_USINE	VILLE
3	US Plastics	LOS ANGELES
4	wePlast	SEOUL

Q6* : Donner les numéros des fournisseurs qui approvisionnent l'usine n°1 :

Select NF as Num_Fournisseur from Livraison where NU=1;

```
SQL> SELECT NF AS NUM_FOURNISSEUR FROM livraison WHERE NU=1;
```

NUM_FOURNISSEUR
1
3

Q7* : Donner le nom des fournisseurs qui approvisionnent l'usine n°1 en produit n°1 :

Select f.nomF as Nom_Fournisseur from Fournisseur f INNER JOIN Livraison l ON f.NF=l.NF where l.NU=1 and l.NP=1 group by f.nomF;

```
SQL> SELECT f.nomF AS NOM_FOURNISSEUR FROM fournisseur f
  2  INNER JOIN livraison l ON f.NF = l.NF WHERE l.NU=1 AND l.NP=1
  3  GROUP BY f.nomF;
```

NOM_FOURNISSEUR
Jenner

Q8* : Donner les numéros des fournisseurs qui approvisionnent les usines n°1 ou n°2 :

Select NF as Num_Fournisseur from Livraison where NU in(1,2);

```
SQL> SELECT NF AS NUM_FOURNISSEUR FROM livraison WHERE NU IN(1,2);
```

NUM_FOURNISSEUR
1
2
3
1

Q9* : Donner les numéros des usines qui ne reçoivent aucun produit du fournisseur n°1 :

Select NU as Num_Usine from Livraison MINUS select NU from Livraison where NF=1;

```
SQL> SELECT NU as NUM_USINE FROM livraison MINUS SELECT NU FROM livraison WHERE NF=1;
no rows selected
```

Q10* : Donner les numéros des usines qui s'approvisionnent chez le fournisseur n°3 :

Select NU as Num_Usine from Livraison where NF=3;

```
SQL> SELECT NU AS NUM_USINE FROM livraison WHERE NF=3;

NUM_USINE
-----
          1
          8
```

Q11* : Donner les numéros des fournisseurs qui fournissent au moins un produit fourni par le fournisseur n°1 :

Select NF as Num_Fournisseur from Livraison where NP in(select NP from Livraison where NF=1) group by NF;

```
SQL> SELECT NF AS NUM_FOURNISSEUR FROM livraison
  2  WHERE NP IN(select NP from LIVRAISON where NF=1) GROUP BY NF;

NUM_FOURNISSEUR
-----
                1
                2
                4
                3
```

Q12* : Donner les numéros des usines qui s'approvisionnent uniquement chez le fournisseur n°3 :

Select NU as Num_Usine from Livraison Minus select NU from Livraison where NF<>3;

```
SQL> SELECT NU AS NUM_USINE FROM livraison MINUS SELECT NU FROM livraison WHERE NF<>3;

no rows selected
```

Q13* : Chercher pour chaque fournisseur le nom et le nombre d'usines qu'il approvisionne. L'affichage sera ordonné par nombre d'usines :

Select nomF, count(unique NU) as Nbr_Usine from Livraison NATURAL JOIN Fournisseur group by nomF order by 2;

```
SQL> SELECT nomF, count(unique NU) AS NBR_USINE FROM livraison
  2  NATURAL JOIN fournisseur GROUP BY nomF ORDER BY 2;

NOMF          NBR_USINE
-----
Ryan              1
Mirembe          2
Jenner           2
Master           8
```

Q14* : Chercher le nom des fournisseurs qui approvisionne plus de 4 usines :

Select nomF, count(unique NU) as Nbr_Usine from Livraison NATURAL JOIN Fournisseur having count(unique NU)>4 group by nomF order by 2;

```
SQL> SELECT nomF, count(unique NU) AS NBR_USINE FROM livraison
  2  NATURAL JOIN fournisseur HAVING count(unique NU)>4 GROUP BY nomF ORDER BY 2;
```

NOMF	NBR_USINE
Master	8

Q15* : Trouver le nom des produits fournis à plus de 4 usines d'Agadir :

Select NP as num_Produit, nomP as Nom_Produit, count(unique NU) as Nbr_Usine_Agadir from produit JOIN Livraison using (NP) JOIN(select NU from Usine where ville='Agadir') using(NU) group by NP, nomP having count(NU)>=4 order by 1;

```
SQL> SELECT NP AS NUM_PRODUIT, nomP AS NOM_PRODUIT, count(unique NU) AS NBR_USINE_AGADIR
  2  FROM produit JOIN livraison USING (NP)
  3  JOIN (select NU from USINE where VILLE='Agadir') USING (NU)
  4  GROUP BY NP, nomP HAVING count(NU)>=4 ORDER BY 1;
```

NUM_PRODUIT	NOM_PRODUIT	NBR_USINE_AGADIR
4	Chirone	4

Q16* : Afficher pour chaque produit le nom (seule l'initiale en majuscule) et la quantité totale fournie aux usines :

Select initcap(nomP) as Nom_Produit, sum(quantite) as Quantite_total from Produit JOIN Livraison using (NP) group by nomP order by 1 ;

```
SQL> SELECT initcap(nomP) AS NOM_PRODUIT, sum(quantite) AS QUANTITE_TOTAL
  2  FROM produit JOIN livraison USING (NP)
  3  GROUP BY nomP ORDER BY 1;
```

NOM_PRODUIT	QUANTITE_TOTAL
Bikes	225
Chirone	140
Hoodies	1510
Sneakers	1090

Q17* : Chercher, ordonnée par nom, les fournisseurs qui approvisionnent toutes les usines :

Select nomF as Nom_Fournisseur from Fournisseur where exists(select NP from Produit where not exists(select NU from Usine where not exists(select * from Livraison where Fournisseur.NF=Livraison.NF and Usine.NU=Livraison.NU)));

```
SQL> SELECT nomF AS NOM_FOURNISSEUR FROM fournisseur
  2  WHERE EXISTS(select NP from PRODUIT where not exists(select NU from USINE
  3  where not exists(select * from LIVRAISON where FOURNISSEUR.NF=LIVRAISON.NF and USINE.NU=LIVRAISON.NU)));
```

NOM_FOURNISSEUR
Master

