

Filière :

Analytique des données et intelligence artificielle

COMPTE RENDU TP 4:

Base de données



Réalisé par :

OUJAID Soumia

Professeur :

Mr OUKBACH Yassine

Année universitaire : 2023/2024

Objectifs :

Le travail pratique 4 du module de Bases de données et modélisation se concentrent sur l'utilisation des requêtes avancées en SQL. Les objectifs principaux sont :

- ⇒ Opérateurs ensemblistes
- ⇒ Requêtes imbriquées
- ⇒ Division

Exécution des tâches :

Soit les tables suivantes :

- ⇒ Usine (NU, NomU, Ville)
- ⇒ Produit (NP, NomP, Couleur, Poids)
- ⇒ Fournisseur (NF, NomF, Statut, Ville)
- ⇒ Livraison (NP, NU, NF, quantité)

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

```
SQL> create user c##tp4 identified by root123;
Utilisateur cr    .
SQL> grant create table to c##tp4;
Autorisation de privil  ges (GRANT) accept    .
```

- 2. Apr  s avoir indiqu   les cl  s primaires et   trang  res, construire les tables ci-dessus dans le sch  ma de tp4.**

Cr  ation des 4 tables :

```
SQL> create table Usine(
2     NU number primary key,
3     NomU varchar2(10),
4     Ville varchar2(10)
5 );
Table cr      .
SQL> create table Produit(
2     NP number Primary key,
3     NomP varchar2(10),
4     Couleur varchar2(10),
5     Poids decimal(10,2)
6 );
Table cr      .
SQL> create table Fournisseur(
2     NF number primary key,
3     NomF varchar2(10),
4     Statut varchar2(20),
5     Ville varchar2(10)
6 );
Table cr      .
```

```
SQL> create table Livraison(
  2     NP number,
  3     NU number,
  4     NF number,
  5     quantite number,
  6     constraint pk primary key(NU,NP,NF),
  7     constraint fkU foreign key(NU) references Usine(NU),
  8     constraint fkP foreign key(NP) references Produit(NP),
  9     constraint fkF foreign key(NF) references Fournisseur(NF)
10 );

Table cr   e.
```

3. Remplir les tables par des donn  es vraisemblables

Insertion    la table Usine :

```
SQL> insert into Usine(NU,NomU,Ville) values(2,'UsineB','VilleB');
1 ligne cr   e.

SQL> insert into Usine(NU,NomU,Ville) values(3,'UsineC','VilleC');
1 ligne cr   e.

SQL> insert into Usine(NU,NomU,Ville) values(4,'UsineD','VilleD');
1 ligne cr   e.

SQL> insert into Usine(NU,NomU,Ville) values(5,'UsineE','VilleE');
1 ligne cr   e.
```

Insertion    la table Produit :

```
SQL> insert into Produit (NP,NomP,Couleur,Poids) values(10,'ProduitA','Rouge',5.40);
1 ligne cr   e.

SQL> insert into Produit (NP,NomP,Couleur,Poids) values(11,'ProduitB','Jaune',2.35);
1 ligne cr   e.

SQL> insert into Produit (NP,NomP,Couleur,Poids) values(12,'ProduitC','Vert',7.50);
1 ligne cr   e.

SQL> insert into Produit (NP,NomP,Couleur,Poids) values(13,'ProduitD','Blanc',10.20);
1 ligne cr   e.

SQL> insert into Produit (NP,NomP,Couleur,Poids) values(14,'ProduitE','Noir',9);
1 ligne cr   e.
```

Insertion à la table Fournisseur :

```
SQL> insert into Fournisseur (NF,NomF,Statut,Ville) values(100,'FourA','Actif','VillePA');
1 ligne cr    e.

SQL> insert into Fournisseur (NF,NomF,Statut,Ville) values(101,'FourB','inactif','VillePB');
1 ligne cr    e.

SQL> insert into Fournisseur (NF,NomF,Statut,Ville) values(102,'FourC','inactif','VillePC');
1 ligne cr    e.

SQL> insert into Fournisseur (NF,NomF,Statut,Ville) values(103,'FourD','Actif','VillePD');
1 ligne cr    e.

SQL> insert into Fournisseur (NF,NomF,Statut,Ville) values(104,'FourE','Actif','VillePE');
1 ligne cr    e.
```

Insertion à la table Livraison :

```
SQL> insert into Livraison (NP,NU,NF,quantite) values(10,1,100,50);
1 ligne cr    e.

SQL> insert into Livraison (NP,NU,NF,quantite) values(11,2,101,40);
1 ligne cr    e.

SQL> insert into Livraison (NP,NU,NF,quantite) values(12,3,102,24);
1 ligne cr    e.

SQL> insert into Livraison (NP,NU,NF,quantite) values(13,4,103,69);
1 ligne cr    e.

SQL> insert into Livraison (NP,NU,NF,quantite) values(14,5,104,4);
1 ligne cr    e.
```

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

```
SQL> select Upper(NomU) as Nom_Usine ,Upper(Ville) as Ville_Usine
2 From Usine;
```

| NOM_USINE | VILLE_USIN |
|-----------|------------|
| USINEA | VILLEA |
| USINEB | VILLEB |
| USINEC | VILLEC |
| USINED | VILLED |
| USINEE | VILLEE |

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

```
SQL> Select NU ,NomU,Ville
2 From Usine
3 Where NomU like '%A%';
```

| NU | NOMU | VILLE |
|----|--------|--------|
| 1 | UsineA | VilleA |

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

```
SQL> Select F.NF
2 From Fournisseur F
3 join Livraison L on L.NF = F.NF
4 join Usine U on L.NU = U.NU
5 Where U.NU = 1;
```

| NF |
|-----|
| 100 |

SQL>

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

```
SQL> Select F.NomF
2 From Fournisseur F
3 join Livraison L on L.NF = F.NF
4 join Usine U on L.NU = U.NU
5 join Produit P on L.NP = P.NP
6 Where U.NU = 1 and P.NP = 10;
```

| NOMF |
|-------|
| FourA |

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

```
SQL> Select F.NF
2 From Fournisseur F
3 join Livraison L on L.NF = F.NF
4 join Usine U on L.NU = U.NU
5 Where U.NU in(1,2);
```

| NF |
|-----|
| 100 |
| 101 |

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

```
SQL> Select U.NU
  2 From Usine U
  3 join Livraison L on L.NU = U.NU
  4 join Fournisseur F on F.NF = L.NF
  5 Where F.NF <> 100;
```

```
      NU
-----
      2
      3
      4
      5
```

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

```
SQL> Select U.NU
  2 From Usine U
  3 join Livraison L on L.NU = U.NU
  4 join Fournisseur F on F.NF = L.NF
  5 Where F.NF = 102;
```

```
      NU
-----
      3
```

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

```
SQL> Select F.NF
  2 From Fournisseur F
  3 join Livraison L on L.NF = F.NF
  4 Where L.NP in (
  5   Select NP
  6   From Livraison
  7   Where NF=100
  8 );
```

```
      NF
-----
     100
```

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

```
SQL> Select U.NU
  2 From Usine U
  3 join Livraison L on L.NU = U.NU
  4 Where L.NF = 102;
```

```
      NU
-----
      3
```

```
SQL> _
```

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(unique U.NU) as Nb_Usine
2 From Usine U
3 join Livraison L on L.NU = U.NU
4 join Fournisseur F on F.NF = L.NF
5 Group by F.NomF
6 Order by Nb_Usine;
```

| NOMF | NB_USINE |
|-------|----------|
| FourA | 1 |
| FourB | 1 |
| FourE | 1 |
| FourD | 1 |
| FourC | 1 |

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 L.NF = F.NF
4 join Usine U on U.NU = L.NU
5 Group by F.NomF
6 Having count(unique U.NU) >4;
```

aucune ligne sélectionnée

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 L.NP = P.NP
4 join Usine U on U.NU = L.NU
5 Where Ville = 'VilleA'
6 Group by P.NomP
7 Having count(unique U.NU) >4;
```

aucune ligne sélectionnée

SQL>

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

```
SQL> Select InitCap(P.NomP) as NomProduit, sum(L.quantite) as quantite_totale
2 From Produit P
3 join Livraison L on L.NP = P.NP
4 Group by P.NomP;
```

| NOMPRODUIT | QUANTITE_TOTALE |
|------------|-----------------|
| Produita | 50 |
| Produitb | 40 |
| Produitc | 24 |
| Produitd | 69 |
| Produite | 4 |

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

```
SQL> Select  F.NF, F.NomF, F.Statut, F.Ville
  2 From Fournisseur F
  3 Join Livraison L on L.NF = F.NF
  4 Group by  F.NF, F.NomF, F.Statut, F.Ville
  5 having count(distinct L.NU) = (Select count(distinct NU) FROM Usine);

aucune ligne sélectionnée
```