

TP3 : Jointures + fonctions d'agrégation

■ Création de Base de données Livraison :

²Création des tables :

```
SQL> CREATE TABLE PRODUIT (  
  2      Numprod NUMBER,  
  3      Nomprod CHAR(20)  
  4  );
```

Table créée.

```
SQL> CREATE TABLE FOURNISSEUR (  
  2      Numfou NUMBER,  
  3      Nomfou CHAR(3)  
  4  );
```

Table créée.

```
SQL> CREATE TABLE PROPOSER (  
  2      Numfou NUMBER,  
  3      Numprod NUMBER,  
  4      prix INT  
  5  );
```

Table créée.

```
SQL> CREATE TABLE LIVRAISON (  
  2     Numfou NUMBER,  
  3     Numli NUMBER,  
  4     Dateli DATE  
  5 );
```

Table créée.

```
SQL> CREATE TABLE DETAILLIVRAISON (  
  2     Numfou NUMBER,  
  3     Numli NUMBER,  
  4     Numprod NUMBER,  
  5     Qte INT  
  6 );
```

Table créée.

²Création des contraintes :

```
SQL> ALTER TABLE PRODUIT  
  2 ADD CONSTRAINT pk_produit PRIMARY KEY (Numprod);
```

Table modifiée.

```
SQL> ALTER TABLE FOURNISSEUR  
  2 ADD CONSTRAINT pk_four PRIMARY KEY (Numfou);
```

Table modifiée.

```
SQL> ALTER TABLE PROPOSER  
  2 ADD CONSTRAINT pk_proposer PRIMARY KEY (Numfou, Numprod);
```

Table modifiée.

```
SQL> ALTER TABLE PROPOSER
  2  ADD CONSTRAINT fk1_proposer FOREIGN KEY (Numfou)
  3  REFERENCES FOURNISSEUR(Numfou);
```

Table modifiée.

```
SQL> ALTER TABLE PROPOSER
  2  ADD CONSTRAINT fk2_proposer FOREIGN KEY (Numprod)
  3  REFERENCES PRODUIT(Numprod);
```

Table modifiée.

```
SQL> ALTER TABLE LIVRAISON
  2  ADD CONSTRAINT pk_liv PRIMARY KEY (Numfou, Numli);
```

Table modifiée.

```
SQL> ALTER TABLE LIVRAISON
  2  ADD CONSTRAINT fk_liv FOREIGN KEY (Numfou)
  3  REFERENCES FOURNISSEUR(Numfou);
```

Table modifiée.

```
SQL> ALTER TABLE DETAILLIVRAISON
  2  ADD CONSTRAINT pk_detailliv PRIMARY KEY (Numfou, Numli, Numprod);
```

Table modifiée.

```
SQL> ALTER TABLE DETAILLIVRAISON
  2  ADD CONSTRAINT fk1_detailliv FOREIGN KEY (Numfou,Numli)
  3  REFERENCES LIVRAISON(Numfou,Numli);
```

Table modifiée.

```
SQL> ALTER TABLE DETAILLIVRAISON
  2  ADD CONSTRAINT fk2_detailliv FOREIGN KEY (Numfou,Numprod)
  3  REFERENCES PROPOSER(Numfou,Numprod);
```

Table modifiée.

² Insertion des données :

```
SQL> INSERT INTO PRODUIT VALUES (1, 'Roue de secours');
```

1 ligne créée.

```
SQL> INSERT INTO PRODUIT VALUES (2, 'Poupée batman');
```

1 ligne créée.

```
SQL> INSERT INTO PRODUIT VALUES (3, 'Cotons tiges');
```

1 ligne créée.

```
SQL> INSERT INTO PRODUIT VALUES (4, 'cornichons');
```

1 ligne créée.

```
SQL> INSERT INTO FOURNISSEUR VALUES (1, 'F1');
```

1 ligne créée.

```
SQL> INSERT INTO FOURNISSEUR VALUES (2, 'F2');
```

1 ligne créée.

```
SQL> INSERT INTO FOURNISSEUR VALUES (3, 'F3');
```

1 ligne créée.

```
SQL> INSERT INTO FOURNISSEUR VALUES (4, 'F4');
```

1 ligne créée.

```
SQL> INSERT INTO PROPOSER VALUES (1, 1, 200);
1 ligne créée.

SQL> INSERT INTO PROPOSER VALUES (1, 2, 15);
1 ligne créée.

SQL> INSERT INTO PROPOSER VALUES (2, 2, 1);
1 ligne créée.

SQL> INSERT INTO PROPOSER VALUES (3, 3, 2);
1 ligne créée.

SQL> INSERT INTO LIVRAISON VALUES (1, 1, NULL);
1 ligne créée.

SQL> INSERT INTO LIVRAISON VALUES (1, 2, NULL);
1 ligne créée.

SQL> INSERT INTO LIVRAISON VALUES (3, 1, NULL);
1 ligne créée.

SQL> INSERT INTO DETAILLIVRAISON VALUES (3, 1, 3, 10);
1 ligne créée.

SQL> INSERT INTO DETAILLIVRAISON VALUES (1, 1, 1, 25);
1 ligne créée.

SQL> INSERT INTO DETAILLIVRAISON VALUES (1, 1, 2, 20);
1 ligne créée.

SQL> INSERT INTO DETAILLIVRAISON VALUES (1, 2, 1, 15);
1 ligne créée.

SQL> INSERT INTO DETAILLIVRAISON VALUES (1, 2, 2, 17);
1 ligne créée.
```

Jointures :

- 1- Afficher tous les noms des produits dont le numéro a une occurrence dans la table PROPOSER

- Écriture relationnelle

```
SQL> SELECT DISTINCT(pd.Nomprod) FROM PROPOSER ps, PRODUIT pd WHERE (pd.Numprod = ps.Numprod);

NOMPROD
-----
Roue de secours
Poupée batman
Cotons tiges
```

- Écriture SQL2

```
SQL> SELECT DISTINCT(pd.Nomprod)
  2  FROM PRODUIT pd
  3  JOIN PROPOSER ps ON (pd.Numprod=ps.Numprod);

NOMPROD
-----
Roue de secours
Poupée batman
Cotons tiges
```

- 2- Afficher tous les noms des fournisseurs dont le numéro a une occurrence dans la table PROPOSER

- Écriture relationnelle

```
SQL> SELECT DISTINCT(fr.Nomfou) FROM PROPOSER ps, FOURNISSEUR fr WHERE (fr.Numfou = ps.Numfou) ORDER BY Nomfou;

NOM
---
F1
F2
F3
```

- Écriture SQL2

```
SQL> SELECT DISTINCT(fr.Nomfou)
2 FROM FOURNISSEUR fr
3 JOIN PROPOSER ps ON (fr.Numfou=ps.Numfou)
4 ORDER BY fr.Nomfou;

NOM
---
F1
F2
F3
```

3- Afficher les noms des fournisseurs avec pour chaque fournisseur la liste des produits proposés.

- Écriture relationnelle

```
SQL> SELECT fr.Nomfou, pd.Nomprod FROM FOURNISSEUR fr, PROPOSER ps, PRODUIT pd
2 WHERE fr.Numfou=ps.Numfou AND pd.Numprod=ps.Numprod;

NOM NOMPROD
---
F1 Roue de secours
F1 Poupée batman
F2 Poupée batman
F3 Cotons tiges
```

- Écriture SQL2

```
SQL> SELECT fr.Nomfou, pd.Nomprod
2 FROM FOURNISSEUR fr
3 JOIN PROPOSER ps ON fr.Numfou=ps.Numfou
4 JOIN PRODUIT pd ON pd.Numprod=ps.Numprod;

NOM NOMPROD
---
F1 Roue de secours
F1 Poupée batman
F2 Poupée batman
F3 Cotons tiges
```

4- Afficher les noms des fournisseurs proposant des 'Poupées Batman' par ordre de prix croissant.

- Écriture relationnelle

```
SQL> SELECT fr.Nomfou FROM FOURNISSEUR fr, PRODUIT pd, PROPOSER ps
2  WHERE fr.Numfou=ps.Numfou AND pd.Numprod=ps.Numprod AND pd.Nomprod='Poupée batman' ORDER BY prix;

NOM
---
F2
F1
```

- Écriture SQL2

```
SQL> SELECT fr.Nomfou
2  FROM FOURNISSEUR fr
3  JOIN PROPOSER ps ON (fr.Numfou=ps.Numfou)
4  JOIN PRODUIT pd ON (pd.Numprod=ps.Numprod)
5  WHERE pd.Nomprod='Poupée batman'
6  ORDER BY ps.prix;

NOM
---
F2
F1
```

5- Afficher les noms de tous les produits déjà livrés par le fournisseur 'f3'.

- Écriture relationnelle

```
SQL> SELECT pd.Nomprod
2  FROM PRODUIT pd, FOURNISSEUR fr, PROPOSER ps, LIVRAISON li, DETAILLIVRAISON dli
3  WHERE dli.Numli=li.Numli AND dli.Numprod=ps.Numprod AND dli.Numfou=li.Numfou AND li.Numfou=ps.Numfou
4  AND ps.Numfou=fr.Numfou AND pd.Numprod=ps.Numprod AND fr.Nomfou='F3';

NOMPROD
-----
Cotons tiges
```

- Écriture SQL2

```
SQL> SELECT pd.Nomprod
2  FROM PRODUIT pd
3  JOIN PROPOSER ps ON pd.Numprod=ps.Numprod
4  JOIN FOURNISSEUR fr ON fr.Numfou=ps.Numfou
5  JOIN LIVRAISON li ON ps.Numfou=li.Numfou
6  JOIN DETAILLIVRAISON dli ON (dli.Numfou=li.Numfou AND dli.Numprod=ps.Numprod AND dli.Numli=li.Numli)
7  WHERE fr.Nomfou='F3';

NOMPROD
-----
Cotons tiges
```


Fonctions d'agrégation:

1- Donner le nombre de fournisseurs.

```
SQL> SELECT COUNT(*) FROM FOURNISSEUR;

COUNT(*)
-----
         4
```

2- Donner le nombre de fournisseurs ayant déjà effectuée une livraison.

```
SQL> SELECT COUNT(DISTINCT(Numfou)) FROM LIVRAISON;

COUNT(DISTINCT(NUMFOU))
-----
                        2
```

3- Quel est le prix du produit proposé au prix le plus élevé par 'f1' ?

```
SQL> SELECT MAX(ps.prix) FROM PROPOSER ps, FOURNISSEUR fr WHERE ps.Numfou=fr.Numfou AND Nomfou='F1';

MAX(PS.PRIX)
-----
        200
```

4- Combien de produits sont proposés pour chaque fournisseur proposant au moins un produit ?

```
SQL> SELECT fr.Nomfou, COUNT(DISTINCT ps.Numprod) AS Nbprod
  2  FROM FOURNISSEUR fr, PROPOSER ps
  3  WHERE fr.Numfou = ps.Numfou
  4  GROUP BY Nomfou;

NOM      NBPROD
-----
F1         2
F3         1
F2         1
```

5- Afficher le nombre de produits qui ne sont proposés par aucun fournisseur.

```
SQL> SELECT COUNT(DISTINCT pd.numprod) - COUNT(DISTINCT ps.numprod)
2 FROM PRODUIT pd, PROPOSER ps;

COUNT(DISTINCTPD.NUMPROD)-COUNT(DISTINCTPS.NUMPROD)
-----
1
```

```
SQL> SELECT COUNT(*) FROM (SELECT Numprod FROM PRODUIT MINUS SELECT Numprod FROM PROPOSER);

COUNT(*)
-----
1
```

```
SQL> SELECT COUNT(pd.Numprod) FROM PRODUIT pd, PROPOSER ps WHERE pd.Numprod=ps.Numprod(+) AND ps.Numprod is NULL;

COUNT(PD.NUMPROD)
-----
1
```

6- Afficher, pour chaque produit (dont on affichera le nom), le nombre de fournisseurs l'ayant déjà livré.

```
SQL> SELECT pd.Nomprod, COUNT(DISTINCT dliv.numfou)
2 FROM PRODUIT pd, DETAILLIVRAISON dliv
3 WHERE pd.numprod = dliv.numprod
4 GROUP BY nomprod;

NOMPROD          COUNT(DISTINCTDLIV.NUMFOU)
-----
Roue de secours          1
Poupée batman           1
Cotons tiges            1
```

- 7- Donner pour chaque livraison le nom du fournisseur, le numéro de livraison et le nombre de produits livrés.

```
SQL> SELECT fr.Nomfou, liv.Numli, COUNT(dliv.Numprod) AS Nbprod
  2  FROM FOURNISSEUR fr, LIVRAISON liv, DETAILLIVRAISON dliv
  3  WHERE fr.Numfou = liv.Numfou
  4  AND dliv.Numfou = liv.Numfou
  5  AND dliv.numli = liv.Numli
  6  GROUP BY fr.Nomfou, liv.Numli;
```

NOM	NUMLI	NBPROD
F3	1	1
F1	1	2
F1	2	2