

Administration des SGBD (L3 2018-2019)

Corrigé TD N°1 : SQL Avancé

Soit le schéma relationnel de la base de données d'une entreprise commerciale :

Fournisseur (FNum, Nom, Adresse, Ville)

Produit (PNum, Design, Prix, Qte, Couleur, Poids)

Commande (CNum, FNum#, PNum#, Qte, Date)

L'ensemble des instances des différentes tables sont représentées dans les tableaux suivants :

Table Fournisseur

FNum	Nom	Adresse	Ville
F451	Ets KLC		Alger
F028	SARL ABM	N° 22 Bel Horizon	Tlemcen
F290	Cellulose	N°74 Gambetta	

Table Produit

PNum	Design	Prix	Qte	Couleur	Poids
P4521	Roulement 3.5	1 100.00	155	Noir	
P1482	Ecrou 15x21	45.00			
P2999	Courroie 14		25	Bleu	

Table Commande

CNum	FNum	PNum	Qte	Date
C22	F290	P2999	50	14/10/2018
C33	F028	P2999	20	
C44	F290	P4521	90	06/12/2018

Donner les requêtes SQL suivantes :

Langage de Définition de Données (Tables)

1. Créer les relations de la base de données.

```
CREATE TABLE Fournisseur (  
    FNum VARCHAR (4) PRIMARY KEY,  
    Nom VARCHAR (30) NOT NULL,  
    Adresse VARCHAR (40),  
    Ville VARCHAR (20) );
```

```
CREATE TABLE Produit (  
    PNum VARCHAR (5) PRIMARY KEY,  
    Design VARCHAR (25) NOT NULL,  
    Prix NUMBER (9,2),  
    Qte NUMBER (6,2),  
    Poids NUMBER (6,2),  
    Couleur VARCHAR (10) );
```

```
CREATE TABLE Commande (  
    CNum VARCHAR (4) NOT NULL ,  
    FNum VARCHAR (4) REFERENCES Fournisseur (FNum),  
    PNum VARCHAR (5) REFERENCES Produit (PNum),  
    Qte NUMBER (6,2) NOT NULL,  
    Date DATE,  
    PRIMARY KEY (CNum, FNum, PNum) );  
OU FOREIGN KEY (PNum) REFERENCES Produit (PNum);  
    FOREIGN KEY (FNum) REFERENCES Fournisseur (FNum) );
```

2. Créer la table « Commandes_Importantes » contenant le numéro et quantité commandée ainsi que la table « Produits_Soldés » contenant le numéro, désignation et prix.

```
CREATE TABLE Commandes_Importantes (  
    CNum VARCHAR (4) PRIMARY KEY,  
    Qte    NUMBER (6,2) );
```

```
CREATE TABLE Produits_Soldés (  
    PNum VARCHAR (5) PRIMARY KEY ,  
    Design VARCHAR (25),  
    Prix NUMBER (6,2) );
```

3. Créer la table « Courroie » avec insertion des numéros, désignations et quantités des produits dont « Désignation » commence par la lettre « C ».

```
CREATE TABLE Courroie (  
    PNumC    VARCHAR (5) PRIMARY KEY ,  
    DesignC  VARCHAR (25),  
    QteC     NUMBER (6,2),  
    AS SELECT PNum, Design, Qte FROM Produit WHERE Design LIKE 'C%') ;
```

4. On a oublié lors de la définition des données le numéro de téléphone et l'email du fournisseur ainsi qu'on a jugé la colonne « Poids » de la table Produit, inutile. Ajouter l'attribut « NumTel » à la relation Fournisseur et Supprimer la colonne « Poids » de la table Produit. Vérifier les modifications.

```
-ALTER TABLE Fournisseur ADD NumTel VARCHAR (10), Email VARCHAR (20), ;  
-ALTER TABLE Produit DROP COLUMN Poids;  
-DESCRIBE Fournisseur ;  
-DESCRIBE Produit ;
```

5. Renommer la colonne « Date » dans la table Commande par « Date_Cmd », de même pour éviter les confusions renommer la colonne « Qte » dans la table Produit par « Qte_Stock » et la colonne « Qte » dans la table Commande par « Qte_Cmd ». Vérifier les modifications.

```
-ALTER TABLE Commande RENAME COLUMN Date TO Date-Cmd;  
-ALTER TABLE Produit RENAME COLUMN Qte TO Qte-Stock;  
-ALTER TABLE Commande RENAME COLUMN Qte TO Qte-Cmd;  
-DESCRIBE Commande;  
-DESCRIBE Produit;
```

6. Réduire la taille de l'adresse fournisseur à 20.

```
ALTER TABLE Fournisseur MODIFY COLUMN Adresse VACHAR (20);
```

Langage de Manipulation des Données (Insertion, Modification, Suppression)

7. Remplir toutes les tables par les instances représentées ci-dessus.

```
INSERT INTO Fournisseur ( FNum ,Nom,Adresse,Ville)  
VALUES('F451','ETS KLC','N°104Hydra','Alger') ;  
Refaire la même chose pour toutes les instances
```

8. Ajouter dans la table « Commandes_Importantes », les numéros et quantités commandées > 40.

```
INSERT INTO Commandes_Importantes (CNum,Qte)  
SELECT CNum, Qte-Cmd FROM commande  
WHERE Qte-Cmd > 40;
```

9. Ajouter dans la table « Produits_Soldés », les numéros de produits et leurs désignations dont le prix est compris entre 100 DA et 500 DA et les afficher à un prix de 100 DA.

```
INSERT INTO Produits_Soldés (PNum, Design,Prix)  
SELECT PNum, Design,100 FROM Produit  
WHERE Produit.Prix <=500 AND Produit.Prix >=100 ;    (ou BETWEEN 100 AND 500)
```

10. Augmenter le prix de tous les produits bleus de 20%.

```
UPDATE Produit
SET Prix=Prix*1.2
WHERE Couleur ='bleu' ;
```

11. Initialiser les Qte_Stock de tous les produits à 0.

```
UPDATE Produit
SET Qte-stock =0 ;
```

12. Actualiser les prix des produits dans la table « Produits_Soldés », en appliquant une réduction de 50% sur leurs prix initiaux dans la table Produit.

```
UPDATE Produits_Soldés
SET Produits_Soldés.Prix = Produit.Prix*0.5
WHERE PNum.Produits_Soldés = PNum.Produit;
```

13. Vider la table Produit.

```
DELETE FROM Produit;
```

14. Supprimer les commandes des années précédentes.

```
DELETE FROM Commande
WHERE Date <'01/01/2019';
```

15. Supprimer les fournisseurs qui n'ont rien fourni.

```
DELETE FROM Fournisseur
WHERE FNum NOT IN (SELECT FNum FROM Commande);
```

Attr IN requête : Compare l'appartenance d'un attribut au résultat d'une requête. L'attribut, quel que soit son type peut être comparé à l'attribut de projection dans le Select à condition qu'ils sont de même type.

Langage de Manipulation des Données (Interrogation)

16. Lister les différents noms des fournisseurs.

```
SELECT DISTINCT Nom FROM Fournisseur;
```

17. Afficher les désignations de tous les produits avec leurs TTC sachant que $TTC = Prix + Prix * TVA$ ($TVA = 19\%$).

```
SELECT Design, Prix*1.19 FROM Produit;
```

18. Lister les produits dont le prix n'est pas compris entre 200 et 800 DA.

```
SELECT * FROM Produit
WHERE Prix NOT BETWEEN 200 AND 800;
```

Dans la majorité des SGBD, les bornes sont incluses dans le between

19. Lister les produits dont la couleur est renseignée, **et** les produits dont la quantité en stock est > 50 .

```
SELECT * FROM Produit
WHERE Couleur IS NOT NULL OR Qte-stock >50;
```

20. Présenter toutes les informations de commandes avec des noms de colonnes compréhensibles, triées sur la date croissante et la quantité décroissante en priorisant celles qui n'ont pas une date.

```
SELECT CNum AS Numéro de commande, FNum AS Numéro de Fournisseur, PNum AS Numéro de produit, Qte-Cmd AS Quantité Commandée, Date-Cde AS Date de la Commande FROM Commande
ORDER BY Date-Cmd Asc NULLS FIRST, Qte-Cmd DESC;
```

21. Compter le nombre de commandes du produit P2999 de dates différentes et renseignées.

```
SELECT COUNT(DISTINCT Date-Cmd) FROM Commande
WHERE PNum='P2999' ;
```

COUNT(Attribut) : ne comptabilise que les enregistrements dont l'attribut est renseigné

COUNT(DISTINCT Attribut) : ne comptabilise que les valeurs distinctes renseignées de l'attribut

22. Récupérer toutes les statistiques sur les quantités commandées : Somme, Max, Min, Moyenne.

```
SELECT Sum(Qte-cmd), Max(Qte-cmd), Min (Qte-cmd), Avg (Qte-cmd) FROM Commande ;
```

23. Calculer la somme des quantités commandées à chaque fournisseur.

```
SELECT FNum, SUM(Qte-cmd) FROM Commande
GROUP BY FNum;
```

Dans la liste des colonnes d'un Select, ne peuvent figurer que les fonctions de groupe ou les colonnes dans Group By

24. Donner les numéros de produits dont la somme des quantités commandées est > 10000.

```
SELECT PNum FROM Commande
GROUP BY PNum HAVING SUM(Qte-cmd) >10000;
```

25. Lister les noms des fournisseurs avec les désignations des produits commandés ainsi que les quantités commandées.

```
SELECT Nom, Design, Qte-cmd
FROM Fournisseur AS F, Produit AS P, Commande AS C
WHERE F.FNum =C.FNum AND P.PNum= C.PNum;
```

26. Donner les numéros des produits qui sont fournis uniquement par le fournisseur N° F028.

- a.

```
SELECT DISTINCT PNum FROM Commande WHERE FNum = F028
EXCEPT
SELECT PNum FROM Commande WHERE FNum <> F028
```
- b.

```
SELECT DISTINCT PNum FROM Commande
EXCEPT
SELECT PNum FROM Commande WHERE FNum <> F028
```
- c.

```
SELECT DISTINCT PNum FROM Commande
WHERE PNum NOT IN (SELECT PNum FROM Commande WHERE FNum <> F028)
```

27. Donner les numéros des produits qui ont été fournis par tous les fournisseurs.

On peut décomposer la question en 2 sous questions :

a- Le nombre de fournisseurs qui ont fourni chaque produit, on aura :

```
SELECT PNum, COUNT(DISTINCT FNum)
FROM Commande
GROUP BY PNum
```

b- Mnt pour avoir les produits qui ont été approvisionné par tous les fournisseurs, on doit ajouter une condition sur le groupe qui va vérifier que ce nombre de fournisseurs est égal au nombre total des fournisseurs existant dans la table fournisseur qui est:

```
SELECT COUNT(*) FROM Fournisseur
```

A- Division avec Group By

```
SELECT PNum
FROM Commande
GROUP BY PNum
HAVING COUNT(DISTINCT (FNum)) = (SELECT COUNT(*) FROM Fournisseur)
```

B- Division avec NOT EXISTS : Double négation

- Il n'existe pas un produit non fourni par tous les fournisseurs.
- Pour tout NP sélectionné, il n'existe pas un fournisseur qui n'a pas approvisionné ce produit.
- Quels sont les produits qui vérifient : il est faux qu'il n'existe pas de fournisseurs qui n'ont pas approvisionnée ces produits.

```
SELECT DISTINCT PNum FROM Commande C1 (ou FROM Produit c'est la même chose)
WHERE NOT EXISTS (SELECT * FROM Fournisseur F
WHERE NOT EXISTS (SELECT * FROM Commande C2
WHERE C2.FNum = F.FNum and C1.PNum=C2.PNum))
```

28. Lister les couples de noms de fournisseurs situés dans la même ville.

a. **SELECT F1.Nom , F2.Nom FROM Fournisseur AS F1 , Fournisseur AS F2
WHERE F1.Ville=F2.Ville;**

b. **SELECT F1.Nom , F2.Nom FROM Fournisseur AS F1 , Fournisseur AS F2
WHERE F1.Ville=F2.Ville AND F1.FNum <> F2.FNum;**

c. **SELECT F1.Nom , F2.Nom FROM Fournisseur AS F1 , Fournisseur AS F2
WHERE F1.Ville=F2.Ville AND F1.FNum > F2.FNum;**

La 2ème solution améliore la 1ère

La 3ème solution améliore la 2ème

29. Afficher la liste des noms de tous les fournisseurs avec leurs numéros de commandes éventuelles.

**SELECT Nom, CNum
FROM Fournisseur LEFT JOIN Commande
ON Fournisseur.FNum = Commande.FNum;**

**LEFT OUTER JOIN, RIGHT OUTER JOIN, FULL OUTER JOIN : Par défaut c'est INNER. OUTER est facultatif.
En complétant les colonnes manquantes par des valeurs NULL.**

30. Donner les fournisseurs qui habitent la même ville que le fournisseur F290.

**SELECT *
FROM Fournisseur
WHERE Ville =(SELECT Ville
FROM Fournisseur
WHERE FNum=F290);**

31. Afficher les différentes références de fournisseurs livrant au moins un produit dont la quantité est supérieure à chacun des produits livrés par le fournisseur F451.

a. **SELECT DISTINCT FNum
FROM Commande
WHERE Qte-Cmd > ALL (SELECT Qte
FROM Commande
WHERE FNum=F451);**

b. **SELECT DISTINCT FNum
FROM Commande
WHERE Qte-Cmd > (SELECT Max(Qte)
FROM Commande
WHERE FNum=F451);**

Mr MATALLAH H