



Série N 4/ Module 19  
Système de Gestion de Base de Données (I)  
LDD

Pour tout exercice, exécuter les requêtes correspondant dans l'analyseur SQL après création !

**Exercice I :**

**Soit le schéma relationnel suivant:**

- Fournisseur (**Num\_F**, nom, ville)
- Produit (**Code\_P**, libellé, origine, couleur)
- Fourniture (Num\_F, Code\_P, quantité)

**Donnez le script de création de ce schéma relationnel avec les règles de gestion suivant :**

- Les attributs en gras clé primaire, Les attributs souligné clé étrangère
- Quantité a une valeur strictement positive

**Solution :**

```
Use Master
```

```
Go
```

```
Create DataBase S4Ex1
```

```
Go
```

```
use S4Ex1
```

```
Go
```

```
Create Table Fournisseur(Num_F int identity primary key , Nom varchar(30), Ville  
Varchar(50))
```

```
Create Table Produit(Code_P int identity primary key, Libelle VarChar(100), Origine  
VarChar(50), Couleur varchar(50))
```

```
Create Table Fourniture(Num_F int foreign key references Fournisseur(Num_F) , Code_P  
int foreign key references Produit(Code_P), Quantite int Check (Quantite>0))
```

```
Go
```

```
Insert into Fournisseur (nom, ville) values ('Ali', 'Casa')
```

```
Insert into Fournisseur (nom, ville) values ('Ahmed', 'Casa')
```

```
Insert into Fournisseur (nom, ville) values ('Imane', 'Rabat')
```

```
Select * From Fournisseur
```

```
Insert Into Produit(Libelle,Origine,Couleur) values ('Table', 'Dijon', 'Verts')
```

```
Insert Into Produit(Libelle,Origine,Couleur) values ('Chaise', 'Dijon', 'Verts')
```

```
Insert Into Produit(Libelle,Origine,Couleur) values ('Table Typel', 'Rabat', 'Rouge')
```

```
Insert Into Produit(Libelle,Origine,Couleur) values ('Tableau', 'Casa', 'Jaune')
```

```
Insert Into Produit(Libelle,Origine,Couleur) values ('Tableau T1', 'Dijon', 'Verts')
```

```
Select * From Produit
```

```
Insert into Fourniture(Num_F, Code_P,Quantite) values (1,1,5)
```

```
Insert into Fourniture(Num_F, Code_P,Quantite) values (1,2,6)
```

```
Insert into Fourniture(Num_F, Code_P,Quantite) values (2,1,10)
```

```
Insert into Fourniture(Num_F, Code_P,Quantite) values (3,4,14)
```

```
Insert into Fourniture(Num_F, Code_P,Quantite) values (2,1,44)
```

```
Insert into Fourniture(Num_F, Code_P,Quantite) values (1,3,15)
```

```
Insert into Fourniture(Num_F, Code_P,Quantite) values (3,1,8)
```

```
Insert into Fourniture(Num_F, Code_P,Quantite) values (2,5,25)
```

```
Insert into Fourniture(Num_F, Code_P,Quantite) values (2,4,25)
```

```
Insert into Fourniture(Num_F, Code_P,Quantite) values (1,5,50)
```

```
Select * From Fourniture
```

```
-- a)
```

```
Select DISTINCT Num_F from Fourniture
```

```
-- b)
```

```
Select DISTINCT Num_F from Fourniture Where Code_P=4
```

```
-- c)
```

```
Select DISTINCT Num_F from Fourniture Where Code_P<>4
```

```
-- d)
```



```
Select Num_F, Nom From Fournisseur Where Num_F IN (Select DISTINCT Num_F From
Fourniture)
-- e)
Select Num_F, Nom From Fournisseur Where Num_F IN (Select DISTINCT Num_F From
Fourniture where Code_P<>4)
-- f)
Select Num_F From Fourniture
where Code_p in (Select code_p from produit Where Origine like 'Dijon')
Group by Num_F
Having Count(DISTINCT Code_P)=(Select Count(DISTINCT Code_P) From Produit Where
Origine like 'Dijon')
-- g)
Select Nom, ville From Fournisseur INNER JOIN Fourniture ON
Fournisseur.Num_F=Fourniture.Num_F INNER JOIN Produit ON
Fourniture.Code_P=Produit.Code_P Where ville=origine
-- h)
Select DISTINCT Num_F From Fourniture INNER JOIN Produit ON
Fourniture.Code_P=Produit.Code_P Where Couleur like 'Verts'
Select DISTINCT Num_F From Fourniture Where Code_P IN (Select Code_P From Produit
Where Couleur like 'Rouge')
```

---

## Exercice II :

Soit le **MODELE LOGIQUE DE DONNEE** suivant :

ARTICLES (NoArt, Libelle, Stock, PrixInvent)

FOURNISSEURS (NoFour, NomFour, AdrFour, VilleFour)

FOURNIR (NoFour#, NoArt#, PrixArticle, Delai)

Donnez le script de création de ce schéma relationnel avec les règles de gestion suivant :

- Les attributs en gras clé primaire, Les attributs souligné clé étrangère
- Stock, PrixInvent, PrixArticle et Delai ont une valeur strictement positive

### Solution :

```
USE Master
Go
Create DataBase S4Ex2
Go
USE S4Ex2
Go
Create Table ARTICLES (NoArt int primary key, Libelle varchar(50), Stock int,
PrixInvent money)
Create Table FOURNISSEURS (NoFour int primary key, NomFour varchar(50), AdrFour
varchar(100), VilleFour varchar(50))
Create Table FOURNIR (NoFour int foreign key references FOURNISSEURS (NoFour), NoArt
int foreign key references ARTICLES(NoArt), PrixArticle money, Delai int)
Go

insert into ARTICLES(NoArt , Libelle , Stock , PrixInvent) values (1,'PC
Dell',0,5500.50)
insert into ARTICLES(NoArt , Libelle , Stock , PrixInvent) values (2,'PC
HP',0,6500.75)
insert into ARTICLES(NoArt , Libelle , Stock , PrixInvent) values (3,'PC
Acer',0,3500.50)
Select * From ARTICLES
Go

insert into FOURNISSEURS (NoFour , NomFour , AdrFour , VilleFour) values
(1,'Ali', 'rue 5, N 6', 'Casa')
insert into FOURNISSEURS (NoFour , NomFour , AdrFour , VilleFour) values
(2,'Ahmed', 'rue 7, N 34', 'Rabat')
insert into FOURNISSEURS (NoFour , NomFour , AdrFour , VilleFour) values
(3,'Imane', 'rue 2, N 20', 'Casa')
Select * From FOURNISSEURS
```



Go

```
insert into FOURNIR values (1,1,100.5,15)
insert into FOURNIR values (1,2,500.5,5)
insert into FOURNIR values (3,1,10.5,10)
insert into FOURNIR values (2,3,400.75,25)
insert into FOURNIR values (1,3,160.25,21)
insert into FOURNIR values (2,1,250.5,13)
insert into FOURNIR values (1,3,100.5,14)
select * from FOURNIR
Go
```

--Execution requête

--1)

```
SELECT NoArt, Libelle FROM Articles Where Stock<10
```

--2)

```
SELECT * FROM Articles WHERE PrixInvent between 100 AND 300
```

--3)

```
SELECT * FROM Fournisseurs WHERE NomFour LIKE 'STE%'
```

--4)

```
SELECT Fournisseurs.NomFour, AdrFour
```

```
FROM Fournisseurs INNER JOIN Fournir ON Fournisseurs.NoFour=Fournir.NoFour
```

```
WHERE Delai>20
```

--5)

```
SELECT NoArt, Libelle FROM Articles ORDER BY Stock DESC
```

--6)

```
SELECT Articles.NoArt , Libelle , Max(PrixArticle) AS 'Prix Max', Min(PrixArticle) AS  
'Prix Min', AVG(PrixArticle) AS 'Prix Moyenne'
```

```
FROM Articles INNER JOIN Fournir ON Articles.NoArt=Fournir.NoArt
```

```
GROUP BY Articles.NoArt , Libelle
```

---

### Exercice III :

#### Soit le modèle relationnel suivant :

EMP (**Matr**, NomE, Poste, DateEmb, ID\_SUP, Salaire, Commission, NumDept)

DEPT (**NumDept**, NomDept, Lieu)

PROJET (**CodeP**, NomP)

PARTICIPATION (Matr, CodeP, Fonction)

#### Donnez le script de création de ce schéma relationnel avec les règles de gestion suivant :

- Les attributs en gras clé primaire, Les attributs souligné clé étrangère

#### Solution :

```
USE Master
```

Go

```
Create DataBase S4Ex3
```

Go

```
USE S4Ex3
```

Go

```
Create table DEPT (NumDept int primary key, NomDept varchar(100), Lieu varchar(50))
```

```
Create table EMP (Matr int primary key, NomE varchar(30), Poste varchar(100), DateEmb  
smalldatetime, ID_SUP int, Salaire real, Commission decimal(4,2), NumDept int foreign  
key references DEPT (NumDept))
```

```
create table PROJET (CodeP int primary key, NomP varchar(100))
```

```
create table PARTICIPATION (Matr int foreign key references EMP (Matr), CodeP int  
foreign key references PROJET (CodeP), Fonction varchar(100))
```

Go

```
Insert into DEPT values (1, 'Comptabilité', 'Casa')
```

```
Insert into DEPT values (2, 'Production', 'Casa')
```

```
Insert into DEPT values (3, 'Adminisration', 'Casa')
```

```
Select * from DEPT
```

Go



```
truncate table EMP
Insert Into EMP values (1, 'Ali', 'Responsable', '1/7/2004', 0, 5000, 0.15, 1)
Insert Into EMP values (2, 'Ahmed', 'Comptable', '5/2/2005', 1, 3500, 0.10, 1)
Insert Into EMP values (3, 'Khadija', 'Directeur', '2/6/2003', 0, 10000, 0.15, 3)
Insert Into EMP values (4, 'Imane', 'secretaire', '2/9/2008', 3, 3000, 0.05, 3)
Insert Into EMP values (5, 'Amine', 'secretaire', '2/9/2008', 3, 3000, 0.05, 3)
select * from EMP
Go

Insert Into Projet values (1, 'Mise a niveau')
Insert Into Projet values (2, 'préparation appel d''offre')
Insert Into Projet values (3, 'restructuration')
Select * from Projet
Go

Insert into PARTICIPATION values (1,1, 'Consultant')
Insert into PARTICIPATION values (1,2, 'colaborateur')
Insert into PARTICIPATION values (4,3, 'responsable')
Insert into PARTICIPATION values (3,2, 'rapporteur')
Insert into PARTICIPATION values (4,1, 'Consultant')
select * From PARTICIPATION
Go

--1)
SELECT Emp.NomE, EmpS. nome
FROM Emp INNER JOIN Emp AS EmpS ON Emp.id_sup=EmpS.matr
--2)
SELECT Emp.NomE
FROM Emp INNER JOIN Emp AS EmpS on Emp.id_sup=EmpS.matr
WHERE Emp.salaire <= (EmpS.Salaire/2)
--3)
SELECT NomE , Salaire
FROM Emp Emp1
WHERE Salaire >= (SELECT Max(Salaire) from Emp Emp2 where Emp1.NumDept=Emp2.NumDept)
--4)
SELECT NomDept, Salaire
FROM Dept INNER JOIN Emp E1 ON Dept.NumDept=E1.NumDept
WHERE Salaire IN (SELECT TOP 3 Salaire From Emp E2 WHERE E1.NumDept=E2.NumDept ORDER
BY Salaire DESC)
--5)
SELECT NomDept, Poste, SUM(Salaire) AS 'Total'
FROM Dept INNER JOIN Emp ON Dept.NumDept=Emp.NumDept
GROUP BY NomDept, Poste
```

#### Exercice IV :

Soit le modèle relationnel suivant relatif à la gestion des notes annuelles d'une promotion d'étudiants :

ETUDIANT (**NEtudiant**, Nom, Prénom)

MATIERE (**CodeMat**, LibelléMat, CoeffMat)

EVALUER (**NEtudiant**, **CodeMat**, Date, Note)

Donnez le script de création de ce schéma relationnel avec les règles de gestion suivant :

- Les attributs en gras clé primaire, Les attributs souligné clé étrangère
- Note a une valeur entre 0 et 20

**Solution :**

```
USE Master
Go
Create DataBase S4Ex4
Go
USE S4Ex4
Go
```



```
Create table ETUDIANT (NEtudiant int identity primary key, Nom varchar(30), Prénom
varchar(30))
Create table MATIERE (CodeMat int identity primary key, LibelleMat varchar(30),
CoeffMat int)
Create table EVALUER (NEtudiant int foreign key references ETUDIANT (NEtudiant),
CodeMat int foreign key references MATIERE (CodeMat), Date smalldatetime, Note
decimal(4,2) check (Note between 0 and 20))
Go

insert into ETUDIANT values ('Ali', 'P-Ali')
insert into ETUDIANT values ('Ahmed', 'P-Ahmed')
insert into ETUDIANT values ('Imane', 'P-Imane')
insert into ETUDIANT values ('Amine', 'P-Amine')
Select * from ETUDIANT
Go

insert into MATIERE values ('SGBD1', 2)
insert into MATIERE values ('SGBD2', 2)
insert into MATIERE values ('ADO.Net', 3)
insert into MATIERE values ('ASP.Net', 3)
Select * From MATIERE
Go

truncate table EVALUER
insert into EVALUER values (1, 1, '24/10/2010', 12)
insert into EVALUER values (1, 2, '24/10/2010', 20)
insert into EVALUER values (1, 3, '24/10/2010', 12)
insert into EVALUER values (2, 3, '24/10/2010', 13)
insert into EVALUER values (1, 4, '24/10/2010', 5)
insert into EVALUER values (3, 1, '24/10/2010', 9.5)
Select * from EVALUER
Go

--1)
SELECT Count(NEtudiant) FROM Etudiant
--2)
SELECT Max(Note) as 'Note Max', Min(Note) as 'Note Min' FROM Evaluator
--3)
SELECT Nom, LibelleMat, AVG(Note) as 'Moyenne' FROM Etudiant, Evaluator, Matiere
WHERE Etudiant.NEtudiant=Evaluator.NEtudiant AND Evaluator.CodeMat=Matiere.CodeMat
GROUP By Etudiant.NEtudiant, Nom, Matiere.CodeMat, LibelleMat
Order by nom
--4)
SELECT LibelleMat, AVG(Note) as 'Moyenne' FROM Evaluator, Matiere
WHERE Evaluator.CodeMat=Matiere.CodeMat
GROUP By Matiere.CodeMat, LibelleMat

--5)
SELECT Nom, SUM(Note* CoeffMat)/SUM(CoeffMat) as 'MgEtud'
FROM Etudiant, Evaluator, Matiere
WHERE Etudiant.NEtudiant=Evaluator.NEtudiant AND Evaluator.CodeMat=Matiere.CodeMat
GROUP By Etudiant.NEtudiant, Nom
--6)
Create View MgEtud
as SELECT SUM(Note* CoeffMat)/SUM(CoeffMat) as 'MgEtud'
FROM Etudiant, Evaluator, Matiere
WHERE Etudiant.NEtudiant=Evaluator.NEtudiant AND Evaluator.CodeMat=Matiere.CodeMat
GROUP By Etudiant.NEtudiant
Go
SELECT AVG(MgEtud) as 'MoyPromotion'
FROM MgEtud
Go
--7)
```



```
Create View MoyPromotion
as SELECT AVG(MgEtud) as 'MoyPromotion'
FROM MgEtud
Go
```

```
SELECT Nom, SUM(Note* CoeffMat)/SUM(CoeffMat) as 'MgEtud'
FROM Etudiant, Evaluer, Matiere
WHERE Etudiant.NETudiant=Evaluer.NETudiant AND Evaluer.CodeMat=Matiere.CodeMat
GROUP By Etudiant.NETudiant, Nom
Having SUM(Note* CoeffMat)/SUM(CoeffMat)>(select MoyPromotion from MoyPromotion)
Go
```

---

### Exercice V :

#### Soit le schéma relationnel suivant:

EQUIPE (**CodeEquipe**, NomEquipe, DirecteurSportif)

COUREUR (**NuméroCoureur**, NomCoureur, CodeEquipe, CodePays)

PAYS (**CodePays**, NomPays)

TYPE\_ETAPE (**CodeType**, LibelléType)

ETAPE (**NuméroEtape**, DateEtape, VilleDép, VilleArr, NbKm, CodeType)

PARTICIPER (NuméroCoureur, NuméroEtape, TempsRéalisé)

ATTRIBUER\_BONIFICATION (NuméroEtape, km, Rang, NbSecondes, NuméroCoureur)

#### Donnez le script de création de ce schéma relationnel avec les règles de gestion suivant :

- Les attributs en gras clé primaire, Les attributs souligné clé étrangère