

## SGBD2 TP

### Base de données : Gestion de Commandes

-- Créer et utiliser la base de données

```
create database GCommandeDb;
use GCommandeDb;
```

-- Créer les tables

```
create table Article (
    NumArt int primary key,
    DesArt varchar(30),
    PUArt money,
    QteEnStock int,
    SeuilMinimum int,
    SeuilMaximum int
);
create table Commande (
    NumCom int primary key,
    DatCom date,
);
create table LigneCommande (
    NumCom int foreign key references Commande (NumCom),
    NumArt int foreign key references Article (NumArt),
    QteCommandee int,
    primary key (NumCom, NumArt)
);
```

-- Insérer des données de test dans les tables

```
insert into Article values
(1, 'Ordinateur Poste', 10000, 30, 10, 50),
(2, 'Ordinateur Portable', 5000, 15, 20, 70),
(3, 'Téléphone Fixe', 200, 100, 25, 100),
(4, 'Smartphone', 2000, 50, 50, 200),
(5, 'Tablette Graphique', 3000, 70, 10, 80);
insert into Commande values
(1, '2016-01-01'), (2, '2016-01-01'),
(3, '2016-01-02'), (4, '2016-01-03'),
(5, '2016-01-04'), (6, '2016-01-04');
insert into LigneCommande values
(1, 3, 1), (1, 4, 1), ----- Ligne Commande 1
(2, 2, 1), ----- Ligne Commande 2
(3, 1, 3), (3, 2, 3), (3, 4, 3), (3, 5, 3), ----- Ligne Commande 3
(4, 3, 5), ----- Ligne Commande 4
(5, 2, 2), (5, 5, 1), ----- Ligne Commande 5
(6, 4, 2), (6, 5, 1), (6, 1, 3); ----- Ligne Commande 6
```

-- 1) Ecrire un programme qui calcule le montant de la commande N°10 et affiche un message 'Commande Normale' ou 'Commande Spéciale' selon que le montant est inférieur ou supérieur à 100000 DH.

```
declare @num_com int = 10;
declare @montant_com money, @type_com varchar(8);
set @montant_com = (
    select sum(PUArt*QteCommandee)
    from LigneCommande inner join Article
    on LigneCommande.NumArt = Article.NumArt
    where NumCom = @num_com
);
if @montant_com is null
    print 'Commande inexistante ou vide!';
else
    begin
        if @montant_com > 100000
            set @type_com = 'Spéciale';
        else
            set @type_com = 'Normale';
    end
```

```

-- Affichage des informations
print 'Commande numéro: ' + convert(varchar, @num_com);
print 'Montant: ' + convert(varchar, @montant_com) + ' DH';
print 'Type: ' + @type_com;

end

```

-- 2) Ecrire un programme qui supprime l'article numéro 8 de la commande numéro 5 et met à jour le stock. Si après la suppression de cet article, la commande numéro 5 n'a plus d'articles associés, la supprimer.

```

declare @num_art int = 8;
declare @num_com int = 5;
declare @qte_com int = (
    select QteCommandee from LigneCommande
    where NumCom = @num_com and NumArt = @num_art
);

delete from LigneCommande where NumArt = @num_art and NumCom = @num_com;
update Article set QteEnStock += @qte_com where NumArt = @num_art;
if not exists (select NumCom from LigneCommande where NumCom = @num_com)
delete from Commande where NumCom = @num_com;

```

-- 3) Ecrire un programme qui affiche la liste des commandes et indique pour chaque commande dans une colonne 'Type' s'il s'agit d'une 'commande normale' (montant <=100000 DH) ou d'une 'commande spéciale' (montant > 100000 DH).

```

select Commande.*, sum(PUArt*QteCommandee) as 'Montant', 'Type' = Case
    when sum(PUArt*QteCommandee) <= 100000
        then 'Commande Normale'
    else
        'Commande Spéciale'
    end
from LigneCommande
inner join Commande on Commande.NumCom = LigneCommande.NumCom
inner join Article on Article.NumArt = LigneCommande.NumArt
group by Commande.NumCom, DatCom;

```

-- 4) A supposer que toutes les commandes ont des montants différents, écrire un programme qui stocke dans une nouvelle table temporaire les 5 meilleures commandes (ayant le montant le plus élevé) classées par montant décroissant (la table à créer aura la structure suivante : NumCom, DatCom, MontantCom).

```

declare @top_commandes table (
    NumCom int,
    DatCom date,
    MontantCom money
);
insert into @top_commandes
select top 5 Commande.*, sum(PUArt*QteCommandee) as 'Montant'
from LigneCommande
inner join Commande on Commande.NumCom = LigneCommande.NumCom
inner join Article on Article.NumArt = LigneCommande.NumArt
Group by Commande.NumCom, DatCom
Order by [Montant] Desc
select * from @top_commandes;

```

-- 5) Ecrire un programme qui :

```

-- • Recherche le numéro de commande le plus élevé dans la table commande et l'incrémente de 1.
declare @num_com int = (select max(NumCom) from commande) + 1;
-- • Enregistre une commande avec ce numéro.
insert into Commande values (@num_com, getdate());
-- • Pour chaque article dont la quantité en stock est inférieure ou égale au seuil minimum
--   enregistre une ligne de commande avec le numéro calculé et une quantité commandée
--   égale au triple du seuil minimum.
insert into LigneCommande
select @num_com, NumArt, 3 * SeuilMinimum from Article
Where QteEnStock <= SeuilMinimum;

```