

ATELIERS

&

EXAMENS

Travaux Pratiques et Examens
GESTION DES BASES DE DONNEES
Sous
ACCESS
Avec solutions

ATELIERS

Ateliers n°1 :
Travaux Pratiques sous Access: TP1
(Création d'une base de données et ses tables, établissement des relations entre les tables et saisie des données à travers des formulaires)

1°/ Créer la base de données nommée: **BD_gestion_des_commandes**

2°/ Créer les quatre tables de la base : **clients, commandes, produits et lignes**

- **Attribuer les clés nécessaires : clients** (cin), **commandes** (num_cmd), **produits** (ref) et **lignes** (num_cmd et ref)

Table Clients:

Nom du champ	Type de données	Description
Cin	Texte(20)	N° de la carte d'identité nationale
Nom	Texte(20)	Nom du client
Prenom	Texte(20)	Prénom du client
Ville	Texte(20)	Ville du client
Catg	Texte(10)	Catégorie du client C1, C2 ou C3

Table Commandes:

Nom du champ	Type de données	Description
Num_cmd	Numérique	Numéro de la commande
Cin	Texte(20)	Cin : Code client
Date	Date/Heure	Date de la commande

Table produits:

Nom du champ	Type de données	Description
Ref	Texte(20)	Référence du produit
Desg	Memo	Désignation du produit
Pu	Monétaire	Prix unitaire du produit

Table lignes:

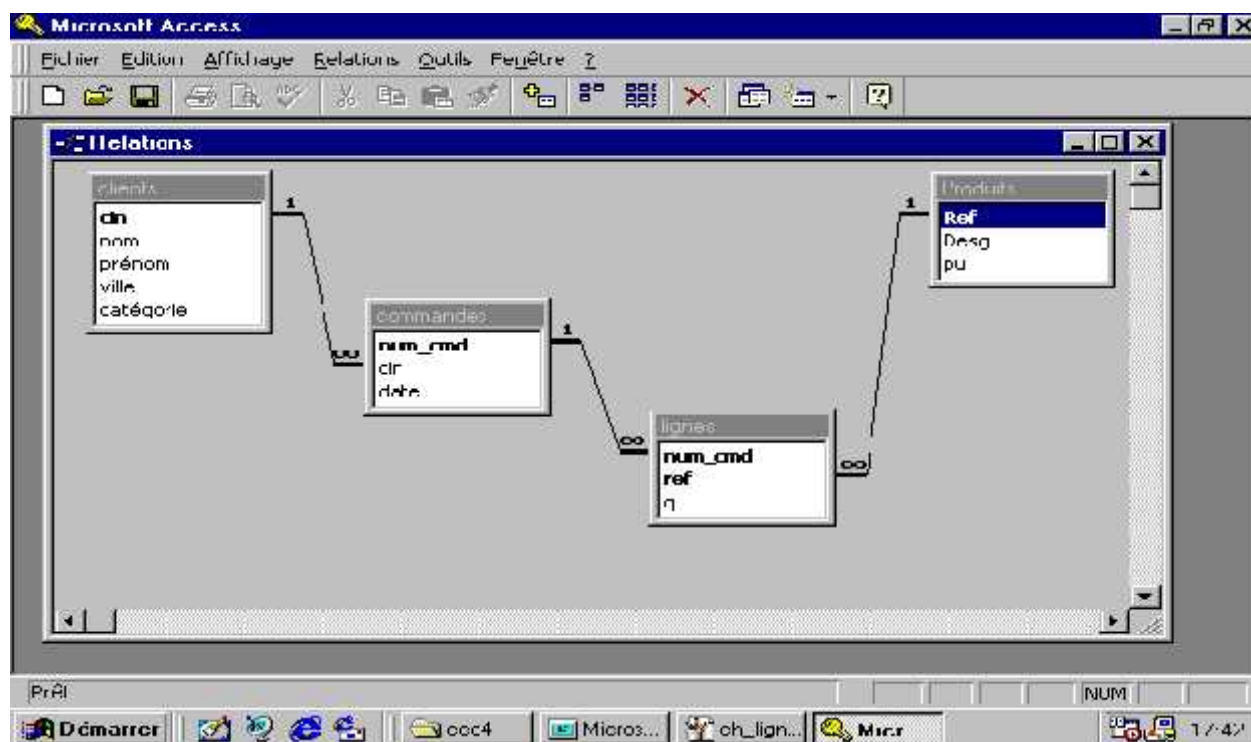
Nom du champ	Type de données	Description
Num_cmd	Numérique	Numéro de la commande
Ref	Texte(20)	Référence du produit
q	Numérique	La quantité vendue au client

3°/Etablir les relations entre les tables: relation de type 1 à n

Clients (1)----->(n) Commandes

Commandes (1)----->(n) lignes

Produits (1)----->(n) lignes



4°/Saisir les données :

Créez pour chaque table un formulaire de saisie des données. Vous devez créer quatre formulaires. Utilisez pour cela, la création d'un formulaire à l'aide de l'assistant :

frmclients, frmcommandes, frmproduits et frm lignes_commandes

Table clients :

cin	nom	prénom	ville	catégorie
A100	amani	mohamed	casablanca	C1
B200	sellami	amine	rabat	C1
C300	asmour	jalil	agadir	C3
D400	tounsi	mostafa	casablanca	C2
E500	kalami	samira	fes	C3
F600	alami	sanaa	casablanca	C2
G700	Mounir	hamid	casablnc	C1
H800	amine	soufiane	rabat	C2

Table commandes :

Numéro de	cin	date
1	A100	01/01/04
2	A100	01/02/04
3	A100	01/03/04
4	B200	15/04/04
5	B200	15/05/04
6	B200	15/06/04
7	C300	10/07/04
8	C300	25/09/04
9	D400	10/10/04
10	E500	25/10/04
11	F600	11/11/04
12	F600	12/12/04
13	G700	15/12/04
14	H800	25/12/04

Table produits :

Référence du	Désignation du	Prix unitaire
PRD01	Claviers	250
PRD02	Ecrans	700
PRD03	Souris	100
PRD04	haut-parleurs	100
PRD05	imprimantes	1200

Table lignes :

Numéro de commande	Référence du	Quantité
1	PRD01	10
1	PRD02	2
2	PRD02	5
2	PRD03	20
3	PRD04	30
4	PRD04	15
5	PRD04	20
6	PRD05	10
7	PRD05	20
8	PRD01	5
9	PRD01	5
10	PRD01	1
11	PRD02	3
11	PRD04	2
12	PRD03	25
12	PRD05	10
13	PRD05	5
14	PRD03	6

Travaux Pratiques sous Access: TP2
(Manipulation des requêtes de sélection)

La Base de données: Gestion_des_commandes

Donner les requêtes: (Utiliser les méthodes QBE et SQL)

1- Afficher les Noms et prénoms des clients.

QBE :

Champ :	nom	prenom
Table :	clients	clients
Tri :		
Afficher :	☒	☒
Critères :		
Ou :		

Résultat :

Requête1_qbe	
nom	prenom
amani	mohamed
sellami	amine
asmour	jalil
tounsi	mostafa
kalami	samira
alami	sanaa
Mounir	hamid
amine	soufiane
*	

SQL :

```
SELECT nom, prenom
FROM clients;
```

2- Afficher le numéro, le nom, le prénom et la ville de chaque client. Avec le nom et le prénom dans une seule colonne et le résultat trié par ordre croissant sur le nom et prénom.

QBE :

Champ :	cin	Nom et prénom: [nom] & " " & [prenom]	ville
Table :	clients		clients
Tri :		Croissant	
Afficher :	☒	☒	☒
Critères :			
Ou :			

SQL :

```
SELECT cin, [nom] & " " & [prenom] as [Nom et prénom], ville  
FROM clients  
ORDER BY [nom] & " " & [prenom] ASC;
```

3- Afficher les clients de la ville Casablanca.

QBE :

Champ :	cin		nom	ville
Table :	clients		clients	clients
Tri :				
Afficher :	☒		☒	☒
Critères :				"casablanca"
Ou :				

SQL :

```
SELECT cin, nom, ville  
FROM clients  
WHERE ville="casablanca";
```

4- Afficher toutes les commandes dont la quantité est supérieure à 25.

QBE :

Champ :	num_cmd		ref	q
Table :	lignes		lignes	lignes
Tri :				
Afficher :	☒		☒	☒
Critères :				>=25
Ou :				

SQL :

```
SELECT num_cmd, ref, q  
FROM lignes  
WHERE q>=25;
```

5- Afficher toutes les lignes de commandes dont la référence du produit est PRD03 **et** dont la quantité est supérieure ou égale à 20.

QBE

Champ :	num_cmd	ref	q
Table :	lignes	lignes	lignes
Tri :			
Afficher :	☒	☒	☒
Critères :		"PRD03"	> = 20
Ou :			

SQL :

```
SELECT num_cmd, ref, q
FROM lignes
WHERE (ref="PRD03") AND (q>=20);
```

6- Afficher toutes les lignes de commandes dont la référence du produit est PRD03 ou dont la quantité est supérieure à 20.

QBE :

Champ :	num_cmd	ref	q
Table :	lignes	lignes	lignes
Tri :			
Afficher :	☒	☒	☒
Critères :		"PRD03"	
Ou :			> = 20

SQL :

```
SELECT num_cmd, ref, q
FROM lignes
WHERE (ref="PRD03") OR (q>=20);
```

7- Afficher toutes les lignes de commandes dont la référence du produit est PRD03 **et** dont la quantité est supérieure à 20, **ou** alors le numéro de commande >5

QBE :

Champ :	num_cmd	ref	q
Table :	lignes	lignes	lignes
Tri :			
Afficher :	☒	☒	☒
Critères :		"PRD03"	>=20
Ou :	>5		

SQL :

```
SELECT num_cmd, ref, q
```

```
FROM lignes
```

```
WHERE ((ref="PRD03") AND (q>=20))or (num_cmd>5);
```

8- Afficher les commandes dont le numéro est compris entre 3 et 10.

QBE : façon1

Champ :	num_cmd	num_cmd	date
Table :	commandes	commandes	commandes
Tri :			
Afficher :	☒	☐	☒
Critères :	>=3	<=10	
Ou :			

QBE : façon2

Champ :	num_cmd	date
Table :	commandes	commandes
Tri :		
Afficher :	☒	☒
Critères :	>=3 Et <=10	
Ou :		

QBE : façon3

Champ :	num_cmd	Date
Table :	commandes	commandes
Tri :		
Afficher :	☒	☒
Critères :	Entre 3 Et 10	
Ou :		

SQL:

Manière1:

```
SELECT num_cmd, date
FROM commandes
where (num_cmd>=3) AND (num_cmd<=10);
```

Manière2:

```
SELECT num_cmd, date
FROM commandes
where num_cmd between 3 AND 10;
```

9- Afficher les clients de la ville Casablanca et rabat.

QBE : Manière1

Champ :	cin	nom	ville
Table :	clients	clients	clients
Tri :			
Afficher :	☒	☒	☒
Critères :			= "casablanca"
Ou :			= "rabat"

QBE : Manière2

Champ :	cin	nom	ville
Table :	clients	clients	clients
Tri :			
Afficher :	☒	☒	☒
Critères :			"casablanca" Ou "rabat"
Ou :			

QBE : Manière3

Champ :	cin	nom	ville
Table :	clients	clients	clients
Tri :			
Afficher :	☒	☒	☒
Critères :			In ("casablanca";"rabat")
Ou :			

SQL :

Manière1:

SELECT cin, nom, ville

FROM clients

WHERE (ville="casablanca") OR (ville="rabat");

Manière2:

SELECT cin, nom, ville

FROM clients

WHERE ville in ("casablanca","rabat");

10- Afficher les clients dont les noms commencent par A.

QBE :

Champ :	cin	nom	ville
Table :	clients	clients	clients
Tri :			
Afficher :	☒	☒	☒
Critères :		Comme "A*"	
Ou :			

SQL :

SELECT cin, nom, ville

FROM clients

WHERE nom like "A*";

11- Afficher les clients dont les noms ne commencent pas par A.

QBE :

Champ :	cin	nom	ville
Table :	clients	clients	clients
Tri :			
Afficher :	☒	☒	☒
Critères :		Pas Comme "A*"	
Ou :			

SQL :

```
SELECT cin, nom, ville
FROM clients
WHERE nom not like "A*";
```

12- Afficher la liste triée par le nom (décroissant) des clients de la ville Casa ou Casablanca.
QBE :

Champ :	cin	nom	ville
Table :	clients	clients	clients
Tri :		Décroissant	
Afficher :	☒	☒	☒
Critères :			Comme "casa*"
Ou :			

SQL :

```
SELECT cin, nom, ville
FROM clients
WHERE ville like "casa*"
ORDER BY nom DESC;
```

13- Afficher le numéro, le nom, le prénom et la ville des clients dont le nom commence par « S », le prénom contient la lettre « M » et la ville est casa ou rabat :
QBE :

Champ :	cin	nom	prenom	ville
Table :	clients	clients	clients	clients
Tri :				
Afficher :	☒	☒	☒	☒
Critères :		Comme "s*"	Comme "*m*"	Comme "casa*" Ou "rabat"
Ou :				

SQL :

```
SELECT cin, nom, ville
```

```

FROM clients
WHERE ((nom like "s*") and (prenom like "*m*")) and ((ville like "casa*") or
(ville="rabat"))

```

14- Afficher le numéro, le nom et la ville des clients dont le nom ne commence pas par « S » et la ville est Casa, Rabat, Mohammedia ou Fès :

QBE :

Champ :	cin	nom	ville
Table :	clients	clients	clients
Tri :			
Afficher :	☒	☒	☒
Critères :		Pas Comme "s*"	Comme "casa*" Ou "rabat" Ou "Mohammedia" Ou "Fès"
Ou :			

SQL :

```

SELECT cin, nom, ville
FROM clients
WHERE (nom not like "s*") and (ville comme "casa*" or ville="rabat" or
ville="mohammedia" or ville="Fes"))

```

15- Afficher le numéro, le nom et la ville des clients de casa, rabat et Fès:

Dont le nom :

- Ne commence pas par « S » pour les clients de casa.
- Commence par A pour les clients de rabat.
- Qui commence par A et se termine par e pour les clients de Fes.

QBE :

Champ :	cin	nom	ville
Table :	clients	clients	clients
Tri :			
Afficher :	☒	☒	☒
Critères :		Pas Comme "s*"	Comme "casa*"
Ou :		Comme "a*"	"rabat"
		Comme "A*e"	"fes"

SQL :

```

SELECT cin, nom, ville

```

```

FROM clients
WHERE ((nom not like "s*") and (ville like "casa*"))
or
((nom like "A*" and (ville="rabat")))
or
((nom like "A*e") and (ville="Fes"))

```

Manipulation de champ de type DATE :

16- Afficher le numéro et la date des commandes qui sont réalisées après le 15 janvier 2008.

QBE :

Champ :	num_cmd	Date
Table :	commandes	commandes
Tri :		
Afficher :	☒	☒
Critères :		>=#15/01/2008#
Ou :		

SQL :

```

SELECT num_cmd, date
FROM commandes
WHERE date>=#01/15/2008#;

```

17- Afficher le numéro et la date des commandes qui sont réalisées entre les dates : 10/01/2008 et 15/06/2008.

QBE :

Champ :	num_cmd	Date
Table :	commandes	commandes
Tri :		
Afficher :	☒	☒
Critères :		>=#10/01/2008# Et <=#15/06/2008#
Ou :		

Entre #10/01/2008# Et #15/06/2008#

SQL :

```

SELECT num_cmd, date
FROM commandes
WHERE (date>=#01/10/2008#) AND (date<=#06/15/2008#);

```

SQL :

```

SELECT num_cmd, date

```

```
FROM commandes
WHERE date between #01/10/2008#)AND #/06/15/2008#;
```

18- Afficher le numéro et la date des commandes faites au cours de l'année 2008(utiliser la fonction année (date))

QBE :

Champ :	num_cmd	Date	Année: Année([date])
Table :	commandes	commandes	
Tri :			
Afficher :	☒	☒	☒
Critères :			2008
Ou :			

SQL :

```
SELECT num_cmd, Date, Year([date]) AS Année
FROM commandes
WHERE Year([date])=2008;
```

19- Afficher le numéro et la date des commandes qui sont réalisées avant le 15 mai au cours de l'année 2008.

QBE :

Champ :	num_cmd	Date	année: Année([date])
Table :	commandes	commandes	
Tri :			
Afficher :	☒	☒	☒
Critères :		<=#15/05/2008#	2008
Ou :			

SQL:

```
SELECT num_cmd, Date, Year([date]) AS Année
FROM commandes
WHERE (date<=#05/15/2008#) AND
(Year([date])=2008);
```

20- Afficher le numéro et la date des commandes faites au cours de l'année 2008 dans les mois de Mars, mai ou juin (utiliser la fonction mois (date))

QBE :

Champ :	num_cmd	Date	année: Année([date])	mois: Mois([date])
Table :	commandes	commandes		
Tri :				
Afficher :	☒	☒	☒	☒
Critères :			2008	3 Ou 5 Ou 6
Ou :				

In(3 ;5 ;6)

Champ :	num_cmd	Date	année: Année([date])	mois: Format([date];"mm")
Table :	commandes	commandes		
Tri :				
Afficher :	☒	☒	☒	☒
Critères :			2008	In ("mars";"mai";"juin")
Ou :				

"mars" ou "mai" ou "juin")

SQL :

```
SELECT num_cmd, Date, Year([date]) AS année, Month([date]) AS mois
FROM commandes
WHERE (Year([date])=2008) AND (Month([date])=3 Or
Month([date])=5 Or Month([date])=6);
```

SQL :

```
SELECT num_cmd, Date, Year([date]) AS année,
Format([date],"mmmm") AS mois
FROM commandes
WHERE (Year([date])=2008) AND
(Format([date],"mmmm") In ("mars","mai","juin"));
```

21- Afficher le numéro et la date des commandes faites au premier de chaque mois au cours de l'année 2008(utiliser la fonction jour (date))

QBE :

Champ :	num_cmd	Date	année: Année([date])	jour: Jour([date])
Table :	commandes	commandes		
Tri :				
Afficher :	☒	☒	☒	☒
Critères :			2008	1
Ou :				

SQL :

```
SELECT num_cmd, Date, Year([date]) AS année, Day([date]) AS jour
FROM commandes
WHERE (Year([date])=2008) AND (Day([date])=1);
```

21b- Afficher le numéro et la date des commandes faites au dernier jour de chaque mois au cours de l'année 2008(utiliser la fonction jour (date))

QBE :

Champ :	num_cmd	Date	jour: Jour([date])	mois: Mois([date])	année: Année([date])
Table :	commandes	commandes			
Tri :					
Afficher :	☒	☒	☒	☒	☒
Critères :			29	2	2008
Ou :			30	In (4;6;9;11)	2008
			31	In (1;3;5;7;8;10;12)	2008

SQL :

```
SELECT num_cmd, Date, Day([date]) AS jour, Month([date]) AS mois, Year([date]) AS
année
FROM commandes
WHERE ((Day([date])=29) AND ((Month([date]))=2) AND
(Year([date])=2008))
OR
((Day([date])=30) AND (Month([date]) In (4,6,9,11)) AND
(Year([date])=2008))
OR
((Day([date])=31) AND (Month([date]) In (1,3,5,7,8,10,12))
AND (Year([date])=2008));
```

22- Afficher le numéro et la date des commandes faites dans le mois courant (utiliser la fonction mois (date ())) (N.B. Date () renvoie la date système)

Champ :	num_cmd	Date	mois: Mois([date])
Table :	commandes	commandes	
Tri :			
Afficher :	☒	☒	☒
Critères :			Mois(Date())
Ou :			

SQL :

23- Afficher le numéro, la date et la durée des commandes en nombre d'année (aaaa), trimestre (t), mois (m), semaine(e), jour (j), heure (h), minutes (n) et secondes(s)

Utiliser la fonction DiffDate("format";[Date];Date())

Exemple : Si le format est « aaaa » alors la fonction renvoie le nombre d'années

Champ :	num_cmd	Date	durée: DiffDate("aaaa";[date];Date())
Table :	commandes	commandes	
Tri :			
Afficher :			
Critères :			
Où :			

SQL :

```
SELECT num_cmd, Date, DateDiff("yyyy",[date],Date()) AS durée
FROM commandes;
```

Travaux Pratiques sous Access: TP3

(Manipulation des requêtes paramétrées, avec jointures et Requêtes avec opérations) :

1- Requête paramétrée :

Un paramètre est une variable dont le nom diffère des noms des différents champs de la BD et dont la valeur sera saisie au moment de l'exécution de la requête. Un paramètre peut être défini soit dans la zone « Champ » ou dans la zone « Critère » d'une requête.

Une requête paramétrée est une requête où on a définie un ou plusieurs paramètres.

Exemples :

1- Afficher les clients d'une ville dont le nom est un paramètre :

Champ :	Num_client	Nom	Prénom	ville
Table :	Clients	Clients	Clients	Clients
Tri :				
Afficher :	☒	☒	☒	☒
Critères :				[Nom de la ville]
Ou :				

Champ :	cin	nom	ville
Table :	clients	clients	clients
Tri :			
Afficher :	☒	☒	☒
Critères :			[Villex]
Ou :			

Champ :	cin	nom	ville
Table :	clients	clients	clients
Tri :			
Afficher :	☒	☒	☒
Critères :			[Entrez votre Ville]
Ou :			

A l'exécution de cette requête, Access affiche la boîte de dialogue suivante :



SQL :

```
SELECT cin, nom, ville
FROM clients
WHERE ville=[villex];
```

2-Afficher les commandes dont la quantité est inférieure ou égale à une valeur à spécifier par l'utilisateur.

QBE :

Champ :	num_cmd	▼	ref	q
Table :	lignes		lignes	lignes
Tri :				
Afficher :	☒		☒	☒
Critères :				<=[Quantex]
Ou :				

SQL :

```
SELECT num_cmd, ref,q
FROM lignes
WHERE q<=[qunatitex];
```

3- Afficher les commandes qui sont réalisées dans une période à déterminer par l'utilisateur.

Champ :	num_cmd	date
Table :	commandes	commandes
Tri :		
Afficher :	☒	☒
Critères :		Entre [date1] Et [date2]
Ou :		

SQL :
SELECT num_cmd, date
FROM commandes
WHERE date between [date1] and [date2];

2-Requête avec jointure:

Une requête avec jointure est une requête qui exploite plusieurs tables à la fois.

Exemples :

1-Afficher le cin et le nom des clients de Casablanca qui ont réalisé des commandes avant 2009.

QBE :

Champ :	cin	nom	ville	num_cmd	Date
Table :	clients	clients	clients	commandes	commandes
Tri :					
Afficher :	☒	☒	☒	☒	☒
Critères :			"casablanca"		<#01/01/2009#
Où :					

SQL :

Manière1:

SELECT clients.cin, clients.nom, clients.ville, commandes.num_cmd, commandes.Date
FROM clients INNER JOIN commandes ON clients.cin = commandes.cin
WHERE (clients.ville="casablanca") AND (commandes.Date<#1/1/2009#);

Manière2:

SELECT x.cin, x.nom, x.ville, y.num_cmd, y.Date
FROM clients as x, commandes as y
WHERE (x.cin = y.cin) and (x.ville="casablanca") AND (y.Date<#1/1/2009#);

2-Afficher le cin et le nom et la ville des clients de catégorie C1 qui ont réalisé des commandes dont la quantité dépasse 20.

QBE :

Champ :	cin	nom	ville	catg	num_cmd	q
Table :	clients	clients	clients	clients	commandes	lignes
Tri :						
Afficher :	☒	☒	☒	☒	☒	☒
Critères :				"C1"		>20
Où :						

SQL :

Manière1:

SELECT clients.cin, clients.nom, clients.ville, clients.catg, commandes.num_cmd,
lignes.q

```
FROM (clients INNER JOIN commandes ON clients.cin = commandes.cin) INNER
JOIN lignes ON commandes.num_cmd = lignes.num_cmd
WHERE (clients.catg="C1") AND (lignes.q>20);
```

Manière2:

```
SELECT x.cin, x.nom, x.ville, x.catg, y.num_cmd, z.q
FROM clients as x, commandes as y, lignes as z
WHERE (x.cin=y.cin) and (y.num_cmd = y.num_cmd) and(x.catg="C1") AND
(z.q>20);
```

3-Afficher le cin et le nom des clients de Casablanca qui ont commandé des produits dont le prix unitaire dépasse 240 DHS.

QBE :

Champ	cin	nom	ville	num_cmd	ref	pu
Table	clients	clients	clients	commandes	lignes	produits
Tri						
Afficher	☒	☒	☒	☒	☒	☒
Critères			"casablanca"			>240
ou						

SQL :

Manière1:

```
SELECT clients.cin, clients.nom, clients.ville, commandes.num_cmd, lignes.ref,
produits.pu
FROM produits INNER JOIN ((clients INNER JOIN commandes ON clients.cin =
commandes.cin) INNER JOIN lignes ON commandes.num_cmd = lignes.num_cmd) ON
produits.ref = lignes.ref
WHERE (clients.ville="casablanca") AND (produits.pu>240);
```

Manière2:

```
SELECT x.cin, x.nom, x.ville, y.num_cmd, z.ref, w.pu
FROM clients as x,commandes as y, lignes as z, produits as w
WHERE (x.cin=y.cin) and (y.num_cmd=z.num_cmd) and (w.ref=z.ref) and
(x.ville="casablanca") AND (w.pu>240);
```

3- Requêtes avec opérations

Pour formuler une opération dans une requête, il suffit de cliquer sur  de la barre d'outils une fois la requête est en mode création. Une ligne *opération* s'ajoute alors parmi les autres lignes de l'interface de la requête.

Les opérations possibles :

- Regroupement des données par un ou plusieurs champs
- Compte : nombre des occurrences des données dans la base

- Somme
- Moyenne
- Max
- Min
- Expression : formule générale
- Où : aucune opération
- var , ecartType, dernier et premier

Exemples :

1- Afficher le nombre total des clients :

QBE :

Champ :	Effectif_Global_Clients: cin
Table :	clients
Opération :	Compte
Tri :	
Afficher :	☒
Critères :	
Où :	

SQL :

```
SELECT count(cin) as effectif_Global_clients
FROM Clients;
```

2- Afficher le nombre de clients par ville :

QBE :

Champ :	ville		Nombre_clients: cin
Table :	clients		clients
Opération :	Regroupement		Compte
Tri :			
Afficher :	☒		☒
Critères :		>=2	
Où :			

SQL :

```
SELECT clients.ville, Count(clients.cin) AS Nombre_clients
FROM clients
GROUP BY clients.ville;
```

3- Afficher la quantité totale, la quantité moyenne, la quantité maximale et la quantité minimale de chaque commande.

QBE :

Champ :	num_cmd	Tota_Qte: q	Moyenne_Qte: q	Qte_maximale: q	Qte_minimale: q
Table :	lignes	lignes	lignes	lignes	lignes
Opération :	Regroupement:	Somme	Moyenne	Min	Max
Tri :					
Afficher :	☒	☒	☒	☒	☒
Critères :					
Où :					

SQL :

```
SELECT lignes.num_cmd,
Sum(lignes.q) AS Total_Qte,
Avg(lignes.q) AS Moyenne_Qte,
Min(lignes.q) AS Qte_maximale,
Max(lignes.q) AS Qte_minimale
FROM lignes
GROUP BY lignes.num_cmd;
```

- 4- Afficher la quantité moyenne, la quantité maximale et la quantité minimale de chaque commande dans les 25 jours avant la date d'aujourd'hui.

QBE :

Champ :	num_cmd	Date	Total_Qte: c	Moyenne_Qte: c	DiffDate('j',[date],Date())
Table :	commandes	commandes	lignes	lignes	
Opération :	Regroupement	Regroupement	Somme	Moyenne	Où
Tri :					
Afficher :	☒	☒	☒	☒	☐
Critères :					<=25
Où :					

SQL :

Manière1:

```
SELECT commandes.num_cmd, commandes.Date, Sum(lignes.q) AS Total_Qte,
Avg(lignes.q) AS Moyenne_Qte
FROM commandes INNER JOIN lignes ON commandes.num_cmd =
lignes.num_cmd
GROUP BY commandes.num_cmd, commandes.Date
HAVING (((commandes.Date)>=Date()-25));
```

Manière2:

```
SELECT commandes.num_cmd, commandes.Date, Sum(lignes.q) AS Total_Qte,
Avg(lignes.q) AS Moyenne_Qte
FROM commandes INNER JOIN lignes ON commandes.num_cmd =
lignes.num_cmd
WHERE DateDiff("d",[date],Date())<=25
GROUP BY commandes.num_cmd, commandes.Date;
```

- 5- Pour chaque commande, afficher le numéro, la date, le montant hors taxe HT, la TVA et le montant TTC avec taux de TVA fixé à 20%.

QBE :

Champ :	num_cmd	Date	HT: Somme([q]*[pu])	TVA: Somme([q]*[pu]*0.2)	TTC: Somme([q]*[pu]*1.2)
Table :	commandes	commandes			
Opération :	Regroupement	Regroupement	Expression	Expression	Expression
Tri :					
Afficher :	☒	☒	☒	☒	☒
Critères :					
Cu :					

SQL :

**SELECT commandes.num_cmd, commandes.Date, Sum([q]*[pu]) AS HT,
Sum([q]*[pu]*0.2) AS TVA, Sum([q]*[pu]*1.2) AS TTC**

**FROM produits INNER JOIN (commandes INNER JOIN lignes ON
commandes.num_cmd = lignes.num_cmd) ON produits.ref = lignes.ref**

GROUP BY commandes.num_cmd, commandes.Date;

Travaux Pratiques sous Access: TP4

(Manipulation des Requêtes analyse croisée et requêtes de mise à jour Modification et suppression)

Requêtes analyse croisée

Une requête Analyse croisée permet de calculer et de restructurer des données afin d'en faciliter l'analyse. Elle permet d'effectuer des calculs (somme, moyenne, compte etc.) pour des données regroupées en lignes ou en colonnes.

Exemples :

- 1- Pour chaque ville, calculer la quantité total commandée, par les clients de cette ville et ceci pour chaque produit.

Champ :	ref	ville	q
Table :	produits	clients	lignes
Opération :	Regroupement	Regroupement	Somme
Analyse :	En-tête de ligne	En-tête de colonne	Valeur
Tri :			
Critères :			
Ou :			

SQL :

```
TRANSFORM Sum(lignes.q) AS SommeDeq--->opération (Valeur)
```

```
SELECT produits.ref--->(entête de ligne)
```

```
FROM produits INNER JOIN ((clients INNER JOIN commandes ON clients.cin =  
commandes.cin) INNER JOIN lignes ON commandes.num_cmd = lignes.num_cmd)  
ON produits.ref = lignes.ref
```

```
GROUP BY produits.ref
```

```
PIVOT clients.ville;---->(entête de colonne)
```

Chaque désignation d'un produit doit être affichée dans une seule colonne et chaque ville doit être placée dans une seule ligne :

Etapes :

1. Créer une nouvelle requête en introduisant les tables : «Clients », « Commandes », « Lignes commandes » et produits.
2. En mode «Création» de requête, cliquez sur le bouton «Analyse Croisée»
 ou sélectionnez la commande *Analyse croisée* du menu *Requête*. Une nouvelle ligne nommée « Analyse » apparaît dans la grille.
3. Dans la ligne « Analyse » de la grille de la requête, choisir « *en tête de ligne* » pour le champ « Ville » et « *en tête de colonne* » pour le champ « désignation ». Tout champ défini en ligne ou en colonne doit avoir «Regroupement» dans la zone «Opération». Choisir « Valeur » pour le champ « Quantité » et définir l'opération somme pour ce champ dans la ligne « Opération »

2- Afficher les quantités totales mensuelles (janvier, février....décembre) commandées pour chaque produit : (utiliser la fonction format (date ; « mmmm »))

QBE :Manière1

Champ :	mois: Mois([date])	desg	q
Table :		produits	lignes
Opération :	Regroupement	Regroupement	Somme
Analyse :	En-tête de ligne	En-tête de colonne	Valeur
Tri :			
Critères :			
Ou :			

QBE :Manière2

Champ :	mois: Format([date];"mmmm")	desg	q	Mois([date])
Table :		produits	lignes	
Opération :	Regroupement	Regroupement	Somme	Regroupement
Analyse :	En-tête de ligne	En-tête de colonne	Valeur	
Tri :				Croissant
Critères :				
Ou :				

SQL :

```

TRANSFORM Sum(lignes.q) AS SommeDeq --->opération (Valeur)

SELECT Format([date],"mmmm") AS mois --->(entête de ligne)

FROM produits INNER JOIN (commandes INNER JOIN lignes ON

commandes.num_cmd = lignes.num_cmd) ON produits.ref =

lignes.ref

GROUP BY Format([date],"mmmm"), Month([date])

ORDER BY Month([date])

PIVOT produits.desg; ---->(entête de colonne)

```

3- Afficher les quantités totales mensuelles commandées pour chaque produit au cours du premier semestre de l'année 2008.

Cnamp :	mois: Format([date],"mmmm")	Mois:[date]	desg	Année([date])	q	Mois([date])
Table :			produits		lignes	
Opération :	Regroupement	Regroupement	Regroupement	Regroupement	Somme	Ou
Analyse :	En-tête de ligne		En-tête de colonne		Valeur	
Tri :		Croissant				
Critères :						Entre 1 Et 6
Ou :						

SQL :

```

TRANSFORM Sum(lignes.q) AS SommeDeq

SELECT Format([date],"mmmm") AS mois

```

```

FROM produits INNER JOIN (commandes INNER JOIN lignes ON
commandes.num_cmd = lignes.num_cmd) ON produits.ref =
lignes.ref
WHERE (Month([date]) Between 1 And 6) AND (Year([date])=2008)

GROUP BY Format([date],"mmmm"), Month([date]), Year([date])
ORDER BY Month([date])

PIVOT produits.desg;

```

Requêtes de mise à jour et suppression

Une requête de mise à jour permet de mettre à jour des données dans une table. Soit en modifiant les données de cette table, soit en les supprimant.

Exemples :

- 2- Mettre à jour les ttva à 19% pour les produits dont le prix unitaire est >=400 DHS.

QBE :

Champ :	ttva	▼	pu
Table :	produits		produits
Mise à jour :	19		
Critères :			>=400
Ou :			

SQL :

```

UPDATE produits
SET ttva = 19
WHERE pu>=400;

```

```

UPDATE produits
SET produits.ttva = 19
WHERE produits.pu>=400));

```

2- Augmentez de 1 dh les prix unitaires des produits dont le taux de TVA est 19%.

QBE :

Champ :	pu	ttva
Table :	produits	produits
Mise à jour :	[pu] +1	
Critères :		19
Où :		

SQL :

```

UPDATE produits
SET produits.pu = [pu]+1
WHERE produits.ttva=19;

```

3- Supprimer les clients de la ville safi.

QBE :

Champ :	ville
Table :	clients
Supprimer :	Où
Critères :	= "safi"
Où :	

SQL :

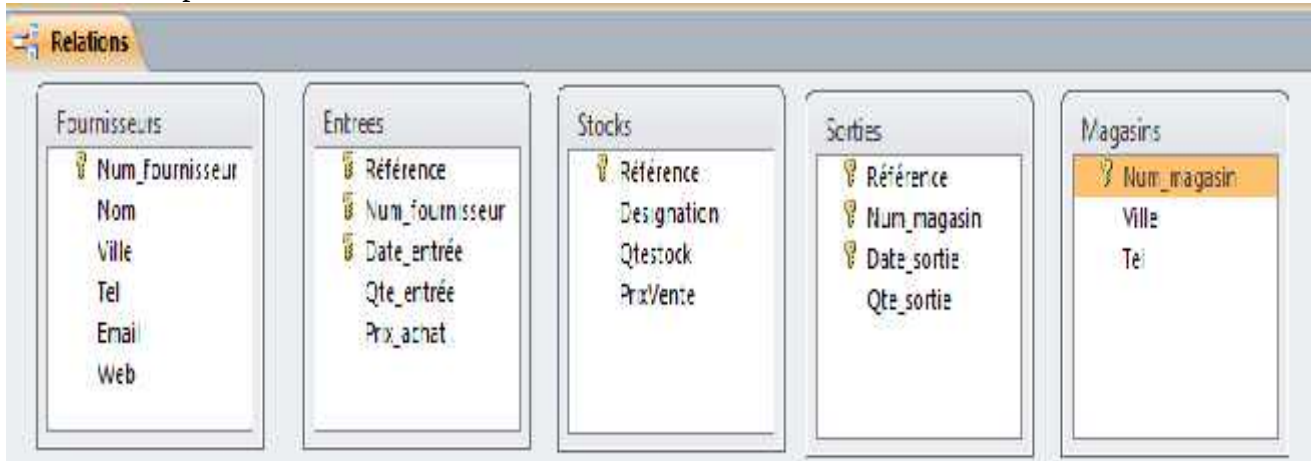
```

DELETE clients.ville
FROM clients
WHERE clients.ville="safi";

```

Ateliers n°2 :
Travaux Pratiques sous Access: TP1
(Création d'une base de données et ses tables, établissement des relations entre les
tables et saisie des données à travers des formulaires)

La base de données d'un système de gestion des produits en stocks d'une entreprise est constituée en partie des tables suivantes :



Soient les règles de gestion suivantes :

- Un fournisseur peut fournir un ou plusieurs produits enregistrés dans la table « Entrées » et peut fournir le même produit mais à des dates différentes.
- A une date donnée, un produit enregistré dans la table « Entrées » peut être fourni par un ou plusieurs fournisseurs.
- Un produit enregistré dans la table « Sorties » peut concerner un ou plusieurs magasins à la fois.

1°/ Créer la base de données nommée: BD_gestion_stock

2°/ Créer les quatre tables de la base : Fournisseurs, Stocks, Magasins, Entrées, et Sorties.

➤ **Attribuer les clés nécessaires :**

Fournisseurs		
Nom du champ	Type de données	
Num_fournisseur	Texte	Numéro de fournisseur (Clé primaire)
Nom	Texte	Nom de fournisseur
Ville	Texte	Ville de fournisseur
Tel	Texte	Téléphone de fournisseur
Email	Texte	Adresse mail de fournisseur
Web	Texte	Adresse web de fournisseur

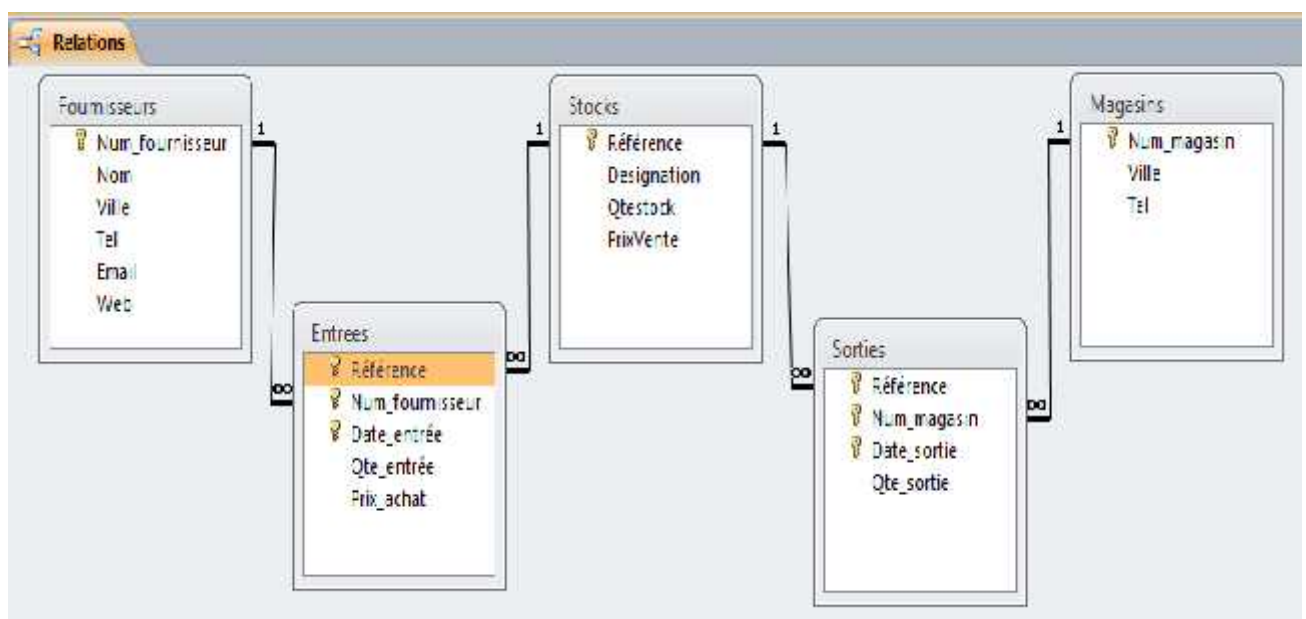
Stocks			
Nom du champ	Type de données		
⚡ Référence	Texte	Référence de produit(Clé primaire)	
Designation	Mémo	Désignation de produit	
Qtestock	Numérique	Quantité en stock	
PrixVente	Monétaire	Prix de vente	

Magasins			
Nom du champ	Type de données		
⚡ Num_magasin	Numérique	Numéro de Magasin	
Ville	Texte	Ville où se trouve le magasin	
Tel	Texte	Téléphone concernant le magasin	

Entrees			
Nom du champ	Type de données		
⚡ Référence	Texte	Référence de produit	
⚡ Num_fournisseur	Texte	Numéro de fournisseur	
⚡ Date_entrée	Date/Heure	Date d'entrée	
Qte_entrée	Numérique	Quantité d'entrée	
Prix_achat	Monétaire	Prix d'achat	

Sorties			
Nom du champ	Type de données		
⚡ Référence	Texte	Référence de produit	
⚡ Num_magasin	Texte	Numéro de magasin	
⚡ Date_sortie	Date/Heure	Date de sortie	
Qte_sortie	Numérique	Quantité de sortie	

3°/Etablir les relations entre les tables: relation de type 1 à n



4°/Saisir les données :

Créez pour chaque table un formulaire de saisie des données. Vous devez créer les formulaires pour chaque table. Utilisez pour cela, la création d'un formulaire à l'aide de l'assistant :

Formulaire1:Gestion de Fournisseurs

Société MAT INFO
Vente de Matériels Informatiques(Ordinateurs, imprimantes, Accessoires,...)
Gestion de Fournisseurs

Num_fournisseur: F100

Nom: AMINE

Ville: CASABLANCA

Tel: 066654563

Email: F100@GMAIL.COM

Web: WWW.F100.COM

Saisissez les données pour chaque table.

Table1 : Fournisseurs

	Num_fourni	Nom	Ville	Tel	Email	Web
+	F100	AMINE	CASABLANCA	066654563	F100@GMAIL.COM	WWW.F100.COM
+	F200	MCUNIR	RABAT	0564654563	F200@HOTMAIL.COM	WWW.F400.COM
+	F300	HAMID	FES	0686654582	F300@GMAIL.COM	WWW.F300.COM
+	F400	SALIM	AGADIR	0634654551	F400@HOTMAIL.COM	WWW.F400.COM
+	F500	MCSTAF	SAFI	0522654573	F500@GMAIL.COM	WWW.F500.COM
+	F600	LAILA	CASABLANCA	0612654592	F600@GMAIL.COM	WWW.F600.COM
+	F700	ASMOUR	RABAT	0633654501	F700@HOTMAIL.FR	WWW.F700.COM
+	F800	ALAMI	FES	0522654594	F800@GMAIL.COM	WWW.F800.COM

Table1 : Magasins

Magasins			
	Num_magasin	Ville	Tel
+	1	CASABLANCA	0522654543
+	2	RABAT	0663234176
+	3	FES	0654647489
+	4	CASABLANCA	0522423657
+	5	RABAT	0662764598

Table1 : Stocks

Stocks				
	Référence	Designation	Qtestock	PrixVente
+	P100	ORDINATEUR	40	5 000,00
+	P200	IMPRIMANTE	70	2 000,00
+	P300	PHOTOCOPIEUR	20	3 500,00
+	P400	ROUTEUR	100	750,00
+	P500	MODEM	80	200,00

Table1 : Entrées

Entrees					
	Référence	Num_fourni	Date_entré	Qte_entrée	Prix_achat
	P100	F100	10/12/2014	10	4 500,00
	P100	F200	07/01/2015	30	4 600,00
	P100	F300	22/04/2015	20	4 700,00
	P200	F400	10/05/2015	10	1 800,00
	P200	F500	15/07/2015	25	1 850,00
	P300	F600	18/06/2015	20	3 200,00
	P300	F700	21/07/2015	15	3 300,00
	P400	F800	16/09/2015	12	90,00
	P500	F100	15/10/2015	20	70,00

Table1 : Sorties

Sorties				
	Référence	Num_maga:	Date_sortie	Qte_sortie
	P100	1	25/12/2014	10
	P100	4	12/08/2015	25
	P200	2	11/02/2015	30
	P200	3	08/09/2015	10
	P300	3	15/03/2015	15
	P400	4	17/06/2015	20
	P500	5	18/08/2015	40

Travaux Pratiques sous Access: TP2
(Manipulation des requêtes de sélection)

La Base de données: BD_Gestion_de_Stock

Donner les requêtes: (Utiliser les méthodes QBE et SQL)

- 1- Afficher le numéro et Nom des Fournisseurs.
- 2- Afficher le numéro, le nom et la ville de chaque Fournisseur. Le résultat trié par ordre croissant par nom.
- 3- Afficher les Fournisseurs de la ville Casablanca.
- 4- Afficher la référence, le numéro de fournisseur et la date dont la quantité d'entrée est supérieure à 20.
- 5- Afficher la référence, le numéro de fournisseur et la date des Entrées dont la référence du produit est P100 et dont la quantité est supérieure ou égale à 20.
- 6- Afficher la référence, le numéro de fournisseur et la date des Entrées dont la référence du produit est P100 ou dont la quantité est supérieure ou égale à 20.
- 7- Afficher le numéro, la ville des magasins dont le numéro est compris entre 2 et 4.
- 8- Afficher le numéro, le nom et la ville des Fournisseurs de Casablanca et rabat.
- 9- Afficher le numéro, le nom et la ville des Fournisseurs dont le nom commence par A.
- 10- Afficher le numéro, le nom et la ville des Fournisseurs dont le nom ne commence pas par « S » et la ville est Casa, Rabat ou Fès

Manipulation de champ de type DATE :

- 1- Afficher la référence et la date des Sorties qui sont réalisées après le 15 janvier 2015.
- 2- Afficher la référence et la date des Sorties qui sont réalisées entre le 15 janvier 2015 et 15 juin 2015.
- 3- Afficher la référence et la date des Sorties qui ne sont pas réalisées entre le 15 janvier 2015 et 15 juin 2015.
- 4- Afficher la référence et la date des Entrées qui sont réalisées au cours de 2014.
- 5- Afficher la référence et la date des Entrées qui sont réalisées avant le 15 avril 2015 au cours de la même année.
- 6- Afficher la référence et la date des Entrées qui sont réalisées au cours de l'année 2015 et ceci pour les mois de février, avril et Mai (utiliser la fonction mois (date))

7- Afficher la référence et la date des Entrées qui sont réalisées le premier de chaque mois au cours de l'année 2015(utiliser la fonction jour (date))

8- Afficher la référence et la date des sorties qui sont réalisées le dernier jour de chaque mois au cours de l'année 2015(utiliser la fonction jour (date))

9- Afficher la référence et la date des Entrées qui sont réalisées dans le mois courant (utiliser la fonction mois (date ())) (N.B. Date () renvoie la date système)

10- Afficher la référence, la date et la durée des Entrées en nombre d'année (aaaa), trimestre (t), mois (m), semaine(e), jour (j), heure (h), minutes (n) et secondes(s) à ce jour.

Travaux Pratiques sous Access: TP3

(Manipulation des requêtes paramétrées, avec jointures et avec opérations) :

1-Requête paramétrée :

- Un paramètre est une variable dont le nom diffère des noms des différents champs de la BD et dont la valeur sera saisie au moment de l'exécution d'une requête.
- Un paramètre peut être défini soit dans la zone « Champ » ou dans la zone « Critère » d'une requête.
- Une requête paramétrée est une requête où on a définie un ou plusieurs paramètres.

Exemples :

- 4- Afficher le numéro, le nom et la ville des Fournisseurs dont la ville a spécifié comme paramètre.
- 5- Afficher la référence, le numéro de fournisseur et la date dont la quantité est inférieure à une valeur d'entrée qui est un paramètre.
- 6- Afficher la référence et la date des Sorties qui sont réalisées dans une période à spécifier.

2-Requête avec jointure :(exploitation de plusieurs tables à la fois)

Exemples :

- 1-Afficher le numéro, le nom et la ville des fournisseurs dont la ville est Casablanca et qui ont réalisé des entrées avant 2015.
- 2-Afficher le numéro, le nom et la ville des fournisseurs qui ont réalisé des entrées au cours de l'année 2015 et ceci pour les produits dont le prix de vente est compris entre 1000 DH et 4000DH.
- 3-Afficher le numéro, le nom et la ville des fournisseurs qui ont réalisé des sorties vers les magasins de la ville de Casablanca ou Rabat.

3- Requêtes avec opérations

Pour formuler une opération dans une requête, il suffit de cliquer sur  de la barre d'outils une fois la requête est en mode création. Une ligne *opération* s'ajoute alors parmi les autres lignes d

e l'interface de la requête.

Les opérations possibles :

- Regroupement des données par un ou plusieurs champs
- Compte : nombre des occurrences des données dans la base
- Somme
- Moyenne
- Max
- Min
- Expression : formule générale
- Où : aucune opération
- var , ecartType, dernier et premier

Exemples :

- 6- Afficher le nombre de Fournisseurs par ville
- 7- Afficher la quantité totale, la quantité moyenne, la quantité maximale et la quantité minimale en entrée pour chaque produit.
- 8- Pour chaque produit, afficher la référence, la désignation la date des entrées, le montant hors taxe HT, la TVA et le montant TTC avec le taux de TVA fixé à 20% ou spécifié comme paramètre.

Travaux Pratiques sous Access: TP4

(Manipulation des Requêtes analyse croisée et requêtes de mise à jour Modification et suppression)

Requêtes analyse croisée

Une requête Analyse croisée permet de calculer et de restructurer des données afin d'en faciliter l'analyse. Elle permet d'effectuer des calculs (somme, moyenne, compte etc.) pour des données regroupées en lignes ou en colonnes.

Exemples :

- 3- Pour chaque ville, calculer la quantité total livrée par les Fournisseurs de cette ville et ceci pour chaque produit.

Chaque désignation d'un produit doit être affichée dans une seule colonne et chaque ville doit être placée dans une seule ligne :

- 2- Afficher les quantités totales mensuelles (janvier, février....décembre) 1 pour chaque produit : (utiliser la fonction format (date ; « mmmm »)

Requêtes de mise à jour

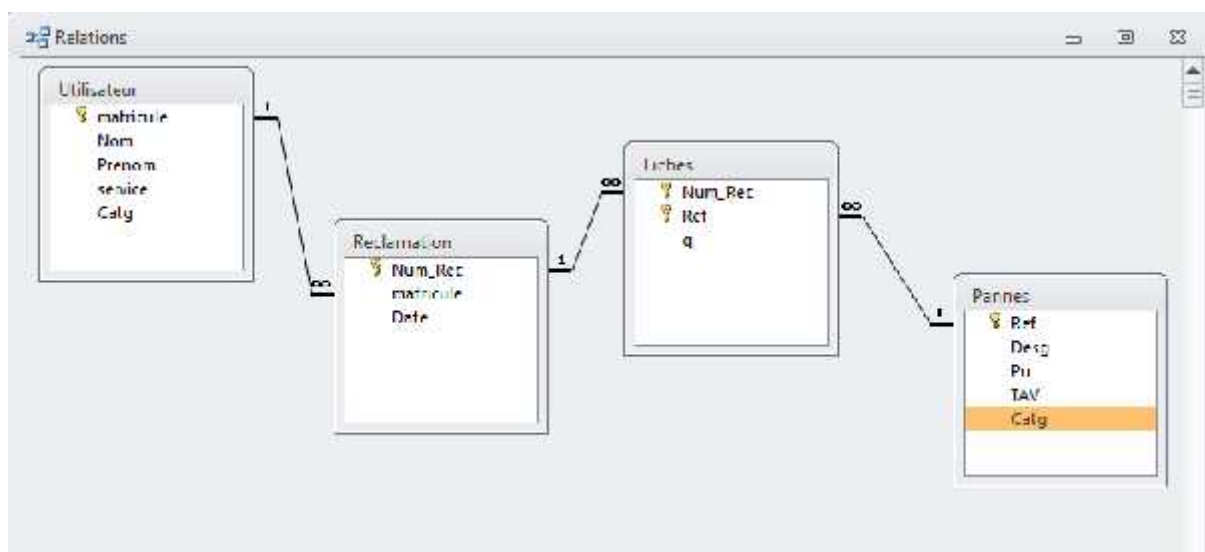
Une requête de mise à jour permet de mettre à jour des données dans une table. Soit en modifiant les données de cette table, soit en les supprimant.

Exemples :

- 1- Mettre à jour les quantités en stock à 500 pour les produits dont la référence est P100 et P200.
- 2- Augmentez de 20 DH les prix de vente des produits de référence P100 et P200.
- 4- Supprimer les Fournisseur de la ville Tanger.

Ateliers n°3 :

Soit la base de données gestion des pannes :



1- afficher les Noms et prénoms des utilisateurs.

Champ :	Nom	Prenom
Table :	Utilisateur	Utilisateur
Tri :		
Afficher :	☒	☒
Critères :		
Où :		

SELECT Utilisateur. Nom, Utilisateur.
Prenom

FROM Utilisateur;

Nom	Prenom
amani	mohamed
sellami	amine
asmour	jalil
tounsi	mostafa
kalami	samira
alami	sanaa
Mounir	hamid
alami	soufiane

2- Afficher le numéro, le nom, le prénom de chaque utilisateur. Avec le nom et le prénom dans une seule colonne et le résultat trié par ordre croissant sur le nom et prénom

Champ :	matricule	Nom et Prenom: [Nom] & " " & [Prenom]
Table :	Utilisateur	Utilisateur
Tri :		
Afficher :	☒	☒
Critères :		
Où :		

```
SELECT Utilisateur. Matricule, Utilisateur.[Nom] & " " &
[Prénom] AS [Nom et Prénom]
FROM Utilisateur;
```

QBE_Req2

matricule	Nom et Prénom
A100	amani mohamed
B200	sellami amine
C300	asmour jalil
D400	tounsi mostafa
E500	kalami samira
F600	alami sanaa
G700	Mounir hamid
H800	alami soufiane

3- Afficher les utilisateurs du service achat

Champ :	matricule	Nom	Prenom	service
Table :	Utilisateur	Utilisateur	Utilisateur	Utilisateur
Tri :				
Afficher :	☒	☒	☒	☒
Critères :				"achat"
Ou :				

SELECT

```
Utilisateur. Matricule, Utilisateur. Nom, Utilisateur. Prénom, Utilisateur. Service
FROM Utilisateur
WHERE (((Utilisateur. Service)="achat"));
```

QBE_Req3

matricule	Nom	Prenom	service
D400	tounsi	mostafa	achat
H800	alami	soufiane	achat

4 -Afficher les utilisateurs de la ville rabat : Si on regarde le contenu du champ "catégorie", on s'aperçoit que toutes les clients ont un catégorie, sauf les ceux de la ville rabat. Alors, il faut indiquer à Access de rechercher les enregistrements dont le champ catégorie est vide. Pour ce faire, il faut utiliser le critère "Null".

Champ :	matricule	Nom	Prenom	service	Catg
Table :	Utilisateur	Utilisateur	Utilisateur	Utilisateur	Utilisateur
Tri :					
Afficher :	☒	☒	☒	☒	☒
Critères :					Est Null
Ou :					

```
SELECT Utilisateur. Matricule, Utilisateur. Nom, Utilisateur. Prénom, Utilisateur. Service,
Utilisateur.Catg
FROM Utilisateur
WHERE (((Utilisateur.Catg) Is Null));
```

QBE_Req4

matricule	Nom	Prenom	service
C300	asmour	jalil	production
E500	kalami	samira	production
S400	alami	ahalm	production

5- Afficher les clients du service achat ou comptabilité

Champ :	matricule	Nom	Prenom	service
Table :	Utilisateur	Utilisateur	Utilisateur	Utilisateur
Tri :				
Afficher :	☒	☒	☒	☒
Critères :				"achat" Ou "comptabilité"
Ou :				

```
SELECT Utilisateur. Matricule, Utilisateur. Nom, Utilisateur. Prénom, Utilisateur. Service
FROM Utilisateur
```

```
WHERE (((Utilisateur. Service)="achat" Or (Utilisateur. Service)="comptabilité"));
```

QBE_Req5

matricule	Nom	Prenom	service
B200	sellami	amine	comptabilité
D400	tounsi	mostafa	achat
G700	Mounir	hamid	comptabilité
H800	alami	soufiane	achat
K900	nour	lamia	achat
M300	souani	mustapha	comptabilité

6- Afficher les clients qui ne sont pas du service achat ou comptabilité

Champ :	matricule	Nom	Prenom	service
Table :	Utilisateur	Utilisateur	Utilisateur	Utilisateur
Tri :				
Afficher :	☒	☒	☒	☒
Critères :				<>"achat" Et <>"comptabilité"
Ou :				

```
SELECT Utilisateur. Matricule, Utilisateur. Nom, Utilisateur. Prénom, Utilisateur. Service
FROM Utilisateur
```

```
WHERE (((Utilisateur. Service)<>"achat" And (Utilisateur. Service)<>"comptabilité"));
```

QBE_Req6

matricule	Nom	Prenom	service
A100	amani	mohamed	finance
C300	asmour	jalil	production
E500	kalami	samira	production
F600	alami	sanaa	finance
N500	lakhdar	rachid	finance
R100	fariss	mounir	finance
S400	alami	ahalm	production

7- Afficher les utilisateurs dont les noms commencent par A.

Champ :	matricule	Nom	Prenom
Table :	Utilisateur	Utilisateur	Utilisateur
Tri :			
Afficher :	☒	☒	☒
Critères :		Comme "a*"	
Ou :			

```
SELECT Utilisateur. Matricule, Utilisateur. Nom, Utilisateur. Prénom
FROM Utilisateur
WHERE (((Utilisateur. Nom) Like "a*"));
```

matricule	Nom	Prenom
A100	amani	mohamed
C300	asmour	jalil
F600	alami	sanaa
H800	alami	soufiane
S400	alami	ahalm

8 -Afficher les utilisateurs dont les noms ne commencent pas par A.

Champ :	matricule	Nom	Prenom
Table :	Utilisateur	Utilisateur	Utilisateur
Tri :			
Afficher :	☒	☒	☒
Critères :		Pas Comme "a*"	
Ou :			

```
SELECT Utilisateur. Matricule, Utilisateur. Nom, Utilisateur. Prénom
FROM Utilisateur
WHERE (((Utilisateur. Nom) Not Like "a*"));
```

matricule	Nom	Prenom
B200	sellami	amine
D400	tounsi	mostafa
E500	kalami	samira
G700	Mounir	hamid
K900	nour	lamia
M300	souani	mustapha
N500	lakhdar	rachid
R100	fariss	mounir

9- Afficher le numéro, le nom et le service des utilisateurs dont le nom commence par « S », le prénom contient la lettre « M » et le service est achat ou finance

Champ :	matricule	Nom	Prenom	service
Table :	Utilisateur	Utilisateur	Utilisateur	Utilisateur
Tri :				
Afficher :	☒	☒	☒	☒
Critères :		Comme "s*"	Comme "*m*"	"achat"
Ou :		Comme "s*"	Comme "*m*"	"finance"

SELECT Utilisateur. Matricule, Utilisateur. Nom, Utilisateur. Prénom, Utilisateur. Service
FROM Utilisateur

WHERE (((Utilisateur. Nom) Like "s*") AND ((Utilisateur. Prénom) Like "*m*") AND
((Utilisateur. Service)="achat")) OR (((Utilisateur. Nom) Like "s*") AND ((Utilisateur.
Prénom) Like "*m*") AND ((Utilisateur. Service)="finance"));

10- Afficher le numéro, le nom et le service des utilisateurs dont le nom ne commence pas par « S » et le service est achat, finance, production ou comptabilité

Champ :	matricule	Nom	Prenom	service
Table :	Utilisateur	Utilisateur	Utilisateur	Utilisateur
Tri :				
Afficher :	☒	☒	☒	☒
Critères :		Pas Comme "s*"		In ("achat, finance, production , comptabilité ")
Ou :				

SELECT Utilisateur. Matricule, Utilisateur. Nom, Utilisateur. Prénom, Utilisateur. Service
FROM Utilisateur

WHERE (((Utilisateur. Nom) Not Like "s*") AND ((Utilisateur. Service) In ("achat, finance,
production, comptabilité ")));

11- Afficher tout les pannes dont la quantité est supérieure ou à égale 20

Champ :	Ref	Desg	q
Table :	Fiches	Pannes	Fiches
Tri :			
Afficher :	☒	☒	☒
Critères :			>= 20
Ou :			

SELECT Fiches. Réf, Pannes.Desg, Fiches.q
FROM Pannes INNER JOIN Fiches ON Pannes.Ref = Fiches. Réf
WHERE (((Fiches.q)>=20));

12- Afficher tout les pannes dont la quantité totale est supérieure à 20

Champ :	Ref	Desg	q
Table :	Pannes	Pannes	Fiches
Opération :	Regroupement	Regroupement	Somme
Tri :			
Afficher :	☒	☒	☒
Critères :			> 20
Ou :			

```
SELECT Pannes.Ref, Pannes.Desg, Sum(Fiches.q) AS SommeDeq
FROM Pannes INNER JOIN Fiches ON Pannes.Ref = Fiches.Ref
GROUP BY Pannes.Ref, Pannes.Desg
HAVING (((Sum(Fiches.q))>20));
```

13- Afficher les réclamations dont le numéro est compris entre 3 et 10.

Champ :	Num_Rec
Table :	Reclamation
Tri :	
Afficher :	☒
Critères :	Entre 3 Et 10
Ou :	

```
SELECT Reclamation.Num_Rec, Réclamation. Date
FROM Réclamation
WHERE (((Reclamation.Num_Rec) Between 3 And 10));
```

14- Afficher la liste triée par le nom (décroissant) des utilisateurs de service achat.

Champ :	matricule	Nom	Prenom	service
Table :	Utilisateur	Utilisateur	Utilisateur	Utilisateur
Tri :		Décroissant		
Afficher :	☒	☒	☒	☒
Critères :				"achat"
Ou :				

```
SELECT Utilisateur. Matricule, Utilisateur. Nom, Utilisateur. Prénom, Utilisateur. Service
FROM Utilisateur
WHERE (((Utilisateur. Service)="achat"))
ORDER BY Utilisateur. Nom DESC;
```

15- Afficher le numéro et la date des réclamations qui sont réalisées après le 15 janvier 2008.

Champ :	Num_Rec	Date
Table :	Reclamation	Reclamation
Tri :		
Afficher :	☒	☒
Critères :		>=#15/01/2008#
Ou :		

```
SELECT Reclamation.Num_Rec, Reclamation.Date
FROM Reclamation
WHERE (((Reclamation.Date)>=#1/15/2008#));
```

16- Afficher le numéro et la date des réclamations qui sont réalisées entre les dates : 10/01/2008 et 15/06/2008.

Champ :	Num_Rec	Date
Table :	Reclamation	Reclamation
Tri :		
Afficher :	☒	☒
Critères :		Entre #01/01/2008# Et #15/06/2008#
Ou :		

```
SELECT Reclamation.Num_Rec, Reclamation.Date
FROM Reclamation
WHERE (((Reclamation.Date) Between #1/1/2008# And #6/15/2008#));
```

17- Afficher le numéro et la date des réclamations faites au cours de l'année 2008 (utiliser la fonction année (date))

Champ :	Num_Rec	Année: Année([date])
Table :	Reclamation	
Tri :		
Afficher :	☒	☒
Critères :		2008
Ou :		

```
SELECT Reclamation.Num_Rec, Year([date]) AS Année
FROM Reclamation
WHERE (((Year([date]))=2008));
```

18- Afficher le numéro et la date des réclamations qui sont réalisées avant le 15 mai au cours de l'année 2008.

Champ :	Num_Rec	Année: Année([date])	Date
Table :	Reclamation		Reclamation
Tri :			
Afficher :	☒	☒	☒
Critères :		2008	<=#15/05/2008#
Ou :			

```
SELECT Reclamation.Num_Rec, Year([date]) AS Année, Reclamation.Date
FROM Reclamation
WHERE (((Year([date]))=2008) AND ((Reclamation.Date)<=#5/15/2008#));
```

19- Afficher le numéro et la date des réclamations faites au cours de l'année 2008 dans les mois de Mars, mai ou juin (utiliser la fonction mois (date))

Champ :	Num_Rec	Année: Année([date])	Mois: Mois([date])
Table :	Reclamation		
Tri :			
Afficher :	☒	☒	☒
Critères :		2008	In (5,6,7)
Ou :			

```
SELECT Reclamation.Num_Rec, Year([date]) AS Année, Month([date]) AS Mois
FROM Reclamation
WHERE (((Year([date]))=2008) AND ((Month([date])) In (5,6,7)));
```

20-Afficher le numéro et la date des réclamations faites au cours de premier semestre de l'année 2008

Champ :	Num_Rec	Année: Année([date])	Mois: Mois([date])
Table :	Reclamation		
Tri :			
Afficher :	☒	☒	☒
Critères :		2008	<6
Ou :			

```
SELECT Reclamation.Num_Rec, Year([date]) AS Année, Month([date]) AS Mois
FROM Reclamation
WHERE (((Year([date]))=2008) AND ((Month([date]))<6));
```

21- Afficher le numéro et la date des réclamations faites au premier de chaque mois au cours de l'année 2008(utiliser la fonction jour (date))

Champ :	Num_Rec	Année: Année([date])	Mois: Mois([date])	Jour: Jour([date])
Table :	Reclamation			
Tri :				
Afficher :	☒	☒	☒	☒
Critères :		2008		1
Où :				

```
SELECT Reclamation.Num_Rec, Year([date]) AS Année, Month([date]) AS Mois,
Day([date]) AS Jour
FROM Reclamation
WHERE (((Year([date]))=2008) AND ((Day([date]))=1));
```

22- Afficher le numéro et la date des réclamations faites à la fin de chaque mois au cours de l'année 2008

Champ :	Num_Rec	Année: Année([date])	Mois: Mois([date])	Jour: Jour([date])
Table :	Reclamation			
Tri :				
Afficher :	☒	☒	☒	☒
Critères :		2008	In (1;3;5;7;8;10;12)	31
Où :		2008	In (4;6;9;11)	30
		2008	2	28

```
SELECT Reclamation.Num_Rec, Year([date]) AS Année, Month([date]) AS Mois,
Day([date]) AS Jour
FROM Reclamation
WHERE (((Year([date]))=2008) AND ((Month([date])) In (1,3,5,7,8,10,12)) AND
((Day([date]))=31)) OR (((Year([date]))=2008) AND ((Month([date])) In (4,6,9,11)) AND
((Day([date]))=30)) OR (((Year([date]))=2008) AND ((Month([date]))=2) AND
((Day([date]))=28));
```

23- Afficher le numéro et la date des réclamations faites dans le mois courant (utiliser la fonction mois (date ())) (N.B. Date () renvoie la date système)

Champ :	Num_Rec	Date	Mois: Mois([date])
Table :	Reclamation	Reclamation	
Tri :			
Afficher :	☒	☒	☒
Critères :			Mois(Date())
Où :			

```
SELECT Reclamation.Num_Rec, Reclamation.Date, Month([date]) AS Mois
FROM Reclamation
WHERE (((Month([date]))=Month(Date())));
```

24- Afficher le numéro, la date et la durée des réclamations en nombre d'année (aaaa), trimestre (t), mois (m), semaine(e), jour (j), heure (h), minutes (n) et secondes(s)

Champ :	Num_Rec	Date	Nbr du jour: DiffDate("j",[date],Date())	Nbr du mois: DiffDate("m",[date],Date())	Nbr du tri: DiffDate("t",[date],Date())
Table :	Reclamation	Reclamation			
Tri :					
Afficher :	☒	☒	☒	☒	☒
Critères :					
Où :					

```
SELECT Reclamation.Num_Rec, Reclamation.Date, DateDiff("d",[date],Date()) AS [Nbr
du jour], DateDiff("m",[date],Date()) AS [Nbr du mois], DateDiff("q",[date],Date()) AS
[Nbr du tri]
FROM Reclamation;
```

Requête paramétrée :

25- Afficher les utilisateurs d'un service dont le nom est un paramètre

Champ :	matricule	Nom	Prenom	service
Table :	Utilisateur	Utilisateur	Utilisateur	Utilisateur
Tri :				
Afficher :	☒	☒	☒	☒
Critères :				[nom de service]
Ou :				

A l'exécution de cette requête, Access affiche la boîte de dialogue suivante :

Entrer une valeur de paramètre ? X

nom de service :

OK Annuler

26- Afficher les réclamations qui sont réalisées dans une période à déterminer par l'utilisateur

Champ :	Num_Rec	Date
Table :	Reclamation	Reclamation
Tri :		
Afficher :	☒	☒
Critères :		[date de réclamation]
Ou :		

Entrer une valeur de paramètre ? X

date de réclamation

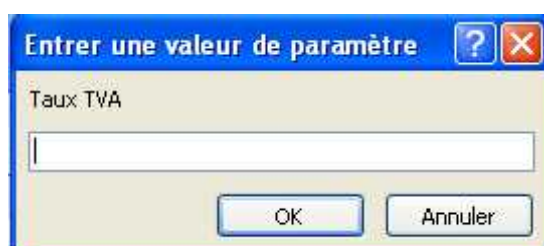
OK Annuler

27- Afficher pour chaque commande le montant HT et TTC à payer

Champ :	Num_Rec	MontantHT: Somme([Pu]*[q])	MontantTTC: Somme([Pu]*[q]*1,2)
Table :	Fiches		
Opération :	Regroupement	Expression	Expression
Tri :			
Afficher :	☒	☒	☒
Critères :			
Ou :			

28-Afficher pour chaque commande le montant HT et TTC à payer avec un taux TVA variable

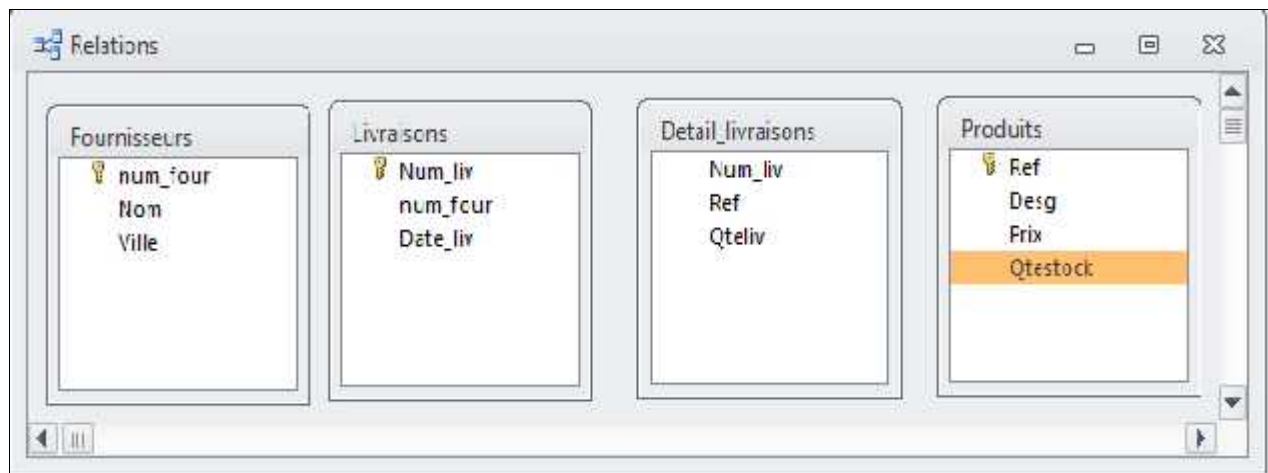
Champ :	Num_Rec	Montant HT: ((([Pu]*[q]))	Montant TTC: ([Pu]*[q]*(1+[Taux TVA]))
Table :	Fiches		
Opération :	Regroupement	Somme	Somme
Tri :			
Afficher :	☒	☒	☒
Critères :			
Ou :			



EXAMENS

Exercice 1 :

Le schéma simplifié d'une base de données pour la gestion des livraisons d'un système d'information, comprend les tables suivantes :



Soient les règles de gestion suivantes :

- **Règle1** : Un fournisseur peut réaliser une ou plusieurs livraisons et une livraison est réalisée par un seul fournisseur
- **Règle2** : Un produit peut être livré plusieurs fois et une livraison peut comprendre plusieurs produits.

I-Questions générales :

1. Définir les notions suivantes : Bases de données et SGBD. (2 points)

- Structure de données permettant de stocker, et de fournir à la demande, des données à de multiples utilisateurs.
- Une base de données est un ensemble d'informations structurées sous forme de Tables. Une table est constituée des colonnes représentant les champs de la table et les lignes représentent les enregistrements de la table.
- **Systèmes de Gestion de Bases de Données (DataBase Management Systems - DBMS) :**

Ensemble de logiciels systèmes permettant aux utilisateurs d'insérer, de modifier, et de rechercher efficacement des données spécifiques dans une grande masse d'informations (pouvant atteindre plusieurs milliards d'octets) partagée par de multiples utilisateurs

2. A quoi sert la clé primaire d'une table ? Montre que le couple (num_liv, ref) constitue la clé primaire de la table Détail_livraisons. (1,5 points)

Une clé est une combinaison minimale de propriétés qui permet d'identifier d'une manière unique un enregistrement

Une clé peut être simple (un seul champ) ou composée (plusieurs champs)

Il y a deux types de clés :

Clé primaire (table principale)

Clé étrangère (table secondaire)

Montre que le couple (num_liv, ref) constitue la clé primaire de la table Détail livraisons

Montrer que le couple (num_liv, ref) constitue la clé primaire de table

Détail livraisons

On cherche la combinaison minimale

Cas1 : on suppose que le num_liv est la clé primaire de la table **Détail livraisons**

Num_liv	ref	Qteliv
1	P1	10
1	P2	25

Le deuxième enregistrement est rejeté car le num_liv est clé primaire

Dans ce cas dans la livraison n°1 on va livrer un seul produit P1. Impossible d'après la règle de gestion R2

Donc, le num_liv n'est pas la clé primaire.

Cas2 : on suppose que la ref est la clé primaire

ref	Num_liv	Qteliv
P1	1	10
P1	2	20

Le deuxième enregistrement est rejeté car la ref est clé primaire

Le produit de ref P1 va être livré une seule fois

Impossible d'après la règle de gestion R2.

Cas3 : on suppose que le couple (num_liv, ref) constitue la clé primaire.

Num_liv	ref	Qteliv
1	P1	10
1	P2	20
2	P1	25

Donc , le couple (num_liv, ref) constitue la clé primaire.

3. Citer les types de relation que vous connaissez. Expliquer Comment peut-on interpréter la relation de type plusieurs à plusieurs. **(2 points)**

Généralement, on distingue trois types de relations

1/ relation de type un à plusieurs

1 à n

1 à l'infini

2/ relation de type plusieurs à plusieurs

n à n

L'infini à l'infini

3/ relation un à un 1 à 1

Expliquer Comment peut-on interpréter la relation de type plusieurs à plusieurs.

- Ce type de relation n'est possible qu'après définition d'une troisième table (appelée table de jonction)
- Une relation plusieurs-à-plusieurs n'est en fait rien d'autre que deux relations un-à-plusieurs avec une troisième table.

4. Donner les relations entre les différentes tables dans le schéma ci-dessus. **(2 points)**

II-Donner les requêtes suivantes en mode QBE ou SQL:

- a) Afficher le numéro, le nom et la ville des Fournisseurs de Casablanca, Agadir et Rabat dont le nom commence par « A » pour les Fournisseurs de Casablanca et qui commence par B pour les fournisseurs d'Agadir. **(1,5 points)**

```
SELECT num_four, nom, ville
FROM fournisseurs
WHERE (ville="casablanca" and nom like "a*" ) or
(ville="Agadir" and nom like "B*" ) or ville="rabat"
```

- b) Afficher le numéro et la date des livraisons qui sont réalisées dans le dernier jour de chaque mois au cours du premier semestre des années 2012 et 2013. **(2 points)**

```
SELECT num_liv, Date, Day([date_liv]) AS jour, Month([date_liv]) AS mois,
Year([date_liv]) AS année
FROM livraisons
WHERE
((Day([date])=30) AND (Month([date]) In (4,6)) AND
(Year([date])in(2012, 2013))
OR
((Day([date])=31) AND (Month([date]) In (1,3,5))
AND (Year([date])in(2012, 2013)));

((Day([date_liv])=29) AND ((Month([date]))=2) AND
(Year([date])=2012))
OR
((Day([date_liv])=28) AND ((Month([date]))=2) AND
(Year([date])=2013))
```

- c) Afficher le numéro, le nom et la ville des fournisseurs de Casablanca qui ont livré des produits dont la quantité en stock est nulle. **(1,5 points)**

```
SELECT fournisseurs.num_four, fournisseurs.nom, fournisseurs.ville
FROM produits INNER JOIN ((fournisseurs INNER JOIN livraisons ON
fournisseurs.num_four = livraisons.num_four) INNER JOIN détail_livraisons ON
livraisons.num_liv = détail_livraisons.num_liv) ON produits.ref = détail_livraisons.ref
WHERE (fournisseurs.ville="casablanca") AND (produits.Qtestock=0);
```

- d) Afficher les villes dont le nombre de Fournisseurs n'est pas compris entre 5 et 10. **(1,5 points)**

```
SELECT fournisseurs.ville, Count(fournisseurs.num_four) AS Nombre_fournisseurs
FROM fournisseurs
GROUP BY fournisseurs.ville
having Count(fournisseurs.num_four) not between 5 and 10 ;
```

- e) Pour chaque produit, afficher la référence, la désignation et le nombre de fournisseurs ayant fournis ce produit au cours du deuxième semestre d'une année donnée comme paramètre. **(2 points)**

```
SELECT produits.ref, produits.desg, Count(fournisseurs.num_four) AS
Nombre_fournisseurs
FROM produits INNER JOIN ((fournisseurs INNER JOIN livraisons ON
fournisseurs.num_four = livraisons.num_four) INNER JOIN détail_livraisons ON
livraisons.num_liv = détail_livraisons.num_liv) ON produits.ref = détail_livraisons.ref
WHERE (Month(livraisons.date_liv) <=6) and (Year([date_liv])=[saissez votre
année])
GROUP BY produits.ref, produits.desg
```

- f) Afficher le numéro et la date des livraisons qui sont réalisées dans le mois courant au cours des années 2012, 2013 et 2014. **(2 points)**

```
SELECT num_liv, Date_liv
FROM livraisons
WHERE (Month([date_liv])=Month(Date()))
and (Year([date_liv] in(2012,2013,2014));
```

- g) Pour chaque livraison, afficher le numéro, la date, le montant hors taxe HT, la REMISE et le montant TTC (le taux de TVA et le taux de REMISE sont donnés comme paramètres). **(2 points)**

```
SELECT livraisons.num_liv, livraisons.Date,
Sum([Qteliv]*[prix]) AS HT,
Sum([Qteliv]*[prix]*[taux_tva]) AS TVA,
Sum([Qteliv]*[prix]*[taux_remise]) AS REMISE,
HT+TVA-REMISE AS TTC

FROM produits INNER JOIN(livraisons INNER JOIN détail_livraisons ON
```

livraisons.num_liv = détail_livraisons.num_liv) ON produits.ref = détail_livraisons.ref
GROUP BY livraisons.num_liv, livraisons.Date_liv;

Exercice 2 :

La base de données d'un système de gestion des produits en stocks d'une entreprise est constituée en partie des tables suivantes :



Soient les règles de gestion suivantes :

- Un fournisseur peut fournir un ou plusieurs produits enregistrés dans la table « Entrées » et peut fournir le même produit mais à des dates différentes.
- A une date donnée, un produit enregistré dans la table « Entrées » peut être fourni par un ou plusieurs fournisseurs.
- Un produit enregistré dans la table « Sorties » peut concerner un ou plusieurs magasins à la fois.

1. Déterminer les clés primaires des tables « Entrées » et « Sorties ».
2. Donner le schéma relationnel de cette base de données.
3. Exprimer en SQL les requêtes suivantes :
 - a- Afficher les numéros et les noms des fournisseurs ayant fournis, au cours du mois de janvier 2007, un nombre de produits supérieurs à 10.
 - b- Pour chaque produit en stock, afficher la référence, la désignation et la quantité restante sachant que : **Quantité_ restante= total des quantités entrées – total des quantités sorties**
 - c- Afficher les numéros et les noms des fournisseurs qui ont fourni des produits dont les quantités en stock dépassent 200, pour les magasins dont les numéros sont compris entre 1 et 10.

Exercice 3 :

La base de données pour la gestion des comptes bancaires est constituée en partie des tables suivantes :

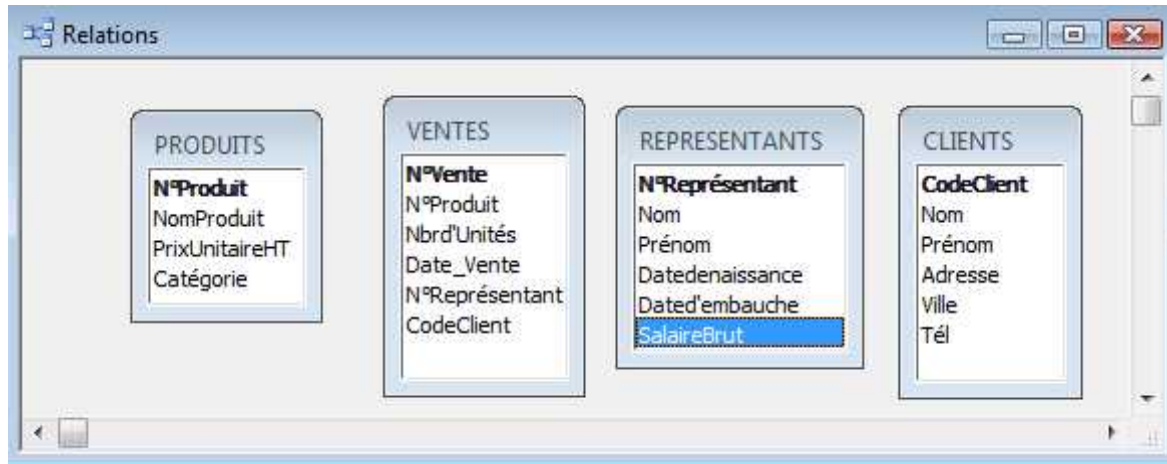
compte_clients	opérations	crédits	pénalité
num_compte nom prénom sexe cin date_création ville tel	num_compte date_opération heure_opération mode libellé montant	num_crédit num_compte date_crédit type duree taux_intérêts montant	num_pénalité num_compte type date_pénalité montant

La table « Opérations » permet de décrire les différentes opérations faites par le client. Le champ « mode » permet de définir le type de paiement (chèque, espèce, etc.) et le champ « Libellé » permet de spécifier si l'opération est créditeur ou débiteur.

1. Selon la structure de la table « opérations », un client peut-il faire plusieurs opérations à la même date ? expliquer pourquoi ?
2. Donner le schéma relationnel de cette base de données.
3. Donner en mode QBE la requête suivante :
Afficher les comptes clients qui ont effectué des opérations en compte débiteurs le 15/12/2007 dont le montant est non compris entre 1000 DH et 2000 DH.
4. Donner en mode SQL les requêtes suivantes :
 - a. Afficher pour chaque compte client créé en 2007, le nombre de pénalités dont le montant dépasse 10000 DH.
 - b. Afficher pour chaque compte client de sexe « Masculin » le montant total d'opérations de type débiteur.
 - c. Afficher les comptes clients de Casablanca et rabat qui ont effectué un crédit de type logement pour une durée 20 ans et qui n'ont jamais eu de pénalité à la fin du trimestre 2007.
 - d. Afficher les numéros de comptes clients dont le montant total d'opérations de type créditeur est inférieur ou égal au montant total d'opérations de type débiteur.

Exercice 4 :

Les ventes des produits d'une entreprise sont gérées par une Base de Données dont le schéma simplifié est le suivant :



- Chaque vente est assurée par un et un seul représentant et porte sur un seul produit et concerne un seul client.
- Un représentant peut faire plusieurs ventes sur plusieurs produits pour plusieurs clients.
- Un client peut être concerné par plusieurs ventes.

I- Questions générales :

1. A quoi sert la clé primaire d'une table ? Citer deux exemples de votre choix. **(1,5points)**
2. Que se passe t-il si on prend le couple (N° **Produit**, N°**Représentant**) comme clé primaire de la table « **ventes** » ? **(1,5points)**
3. Donner les relations entre les différentes tables en précisant l'intégrité référentielle. **(1,5points)**

II- Créer les requêtes suivantes en mode Création (QBE) :

- a. Pour chaque représentant, afficher numéro, nom, le chiffre d'affaire réalisé par ce représentant et sa commission au cours du premier trimestre et dernier trimestre d'une année donnée comme paramètre. Notez bien que chaque représentant perçoit une commission de 10% sur le chiffre d'affaire réalisé. **(2,5points)**
- b. Affichez les Chiffres d'Affaires mensuels réalisés dans les six derniers mois de l'année en cours, par catégorie de produit, les catégories doivent être affichées en colonne et les noms des mois en ligne. **(2,5points)**
- c. Effectuer une augmentation de salaire de 10% pour les représentants ayant réalisés au cours de l'année 2007 et 2008 des ventes dont le montant est compris entre 150000 et 200000. **(2,5points)**

III- Créer les requêtes suivantes en mode SQL :

- a. Afficher le code, le nom et la ville des clients ayant réalisés des ventes dans le premier trimestre 2008 qui sont faites par les représentants dont le nom ne commence pas par la lettre « S » et dont le salaire n'est pas compris entre 2000 et 5000 DHS. **(2,5points)**
- b. Afficher le numéro, nom et la catégorie A ou B des produits dont le prix est non compris entre 200 et 300 DHS qui sont vendus aux clients résidant à la ville de casa ou rabat. **(2,5points)**
- c. Afficher le numéro, le nom et le prénom des représentants dont le salaire dépasse 10000 DHS et qui n'ont vendu aucun produit aux clients. **(3 points)**

Exercice 5 :

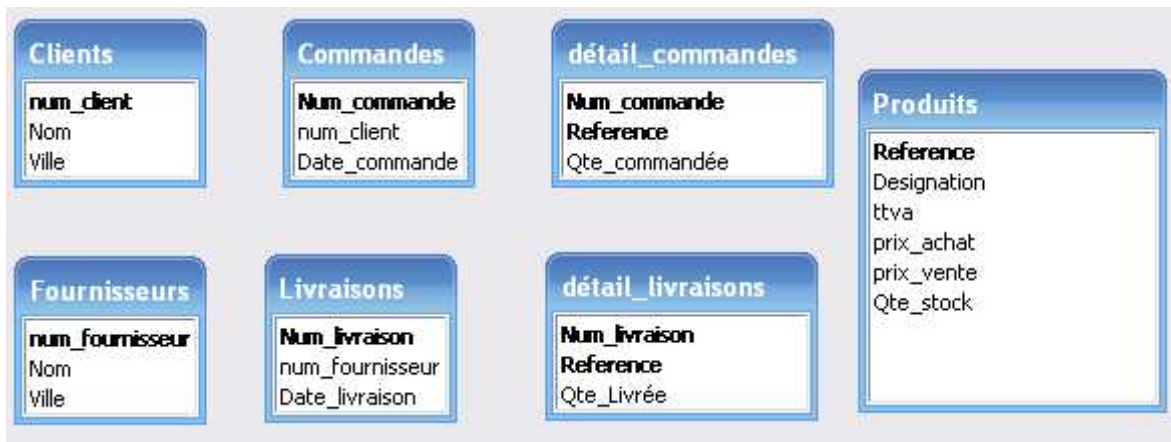
La base de données simplifiée pour la gestion d'une bibliothèque est constituée des tables suivantes:



1. Expliquer l'utilité d'appliquer l'intégrité référentielle entre les tables Livres et Emprunts ?
2. Donner en mode QBE la requête suivante :
Afficher les livres édités avant 2006 par la maison d'édition « Dunod » pour l'auteur ayant le numéro 100 et dont le prix dépasse 500 DHS.
3. Donner en mode SQL les requêtes suivantes :
 - e. Afficher les adhérents de la ville de Casablanca et de sexe Féminin « F » ayant emprunté le livre numéro 100 au cours du dernier semestre 2006.
 - f. Afficher les noms de maisons qui ont édités des livres dont le nombre dépasse 4000.
 - g. Afficher les noms et les numéros des auteurs dont les livres n'ont jamais été empruntés par les adhérents de Casablanca.

Exercice 6 :

Le schéma simplifié d'une base de données pour la gestion des commandes et des livraisons d'un système d'information, comprend les tables suivantes :



Les livraisons sont faites par des fournisseurs et la table « détail_livraisons » contient les références et les quantités livrées des produits de chaque livraison.

I- Questions générales :

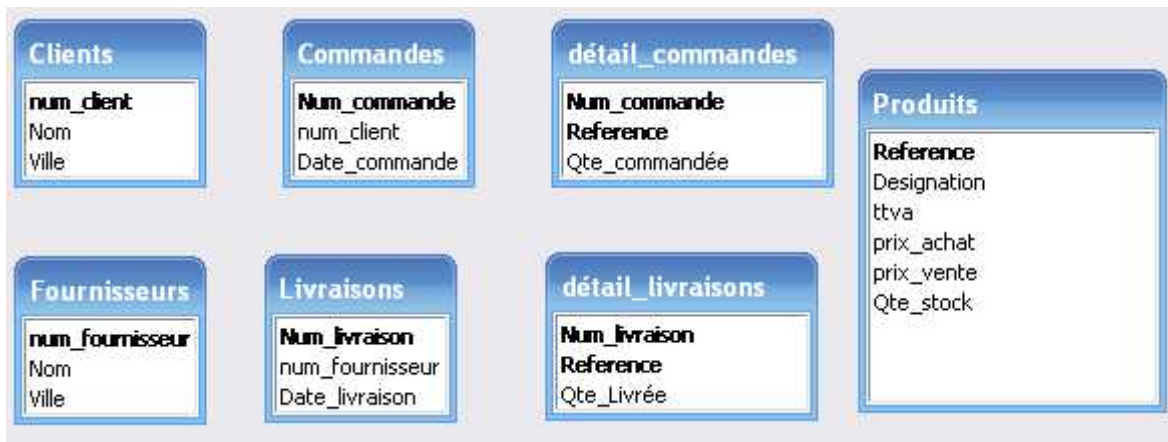
1. A quoi sert la clé primaire d'une table ? Citer deux exemples (clé simple et composée).
2. Montrer que le couple (**Num_commande**, **Référence**) constitue la clé primaire de la table « **détail_commandes** » ?
3. Donner les relations entre les différentes tables en précisant l'intégrité référentielle.

II-Donner les requêtes suivantes en mode création :

1. Pour chaque client, afficher numéro, nom, le montant TTC à payer par ce client au cours des deux dernières années. Notez bien que le taux de tva est variable, Il doit être spécifié comme paramètre.
2. Afficher les quantités totales mensuelles livrées pour chaque Fournisseur au cours du premier trimestre de l'année 2009.
3. Afficher le numéro, le nom et la ville des Fournisseurs dont le nom ne commence pas par « S » et la ville est Casa, Rabat, Mohammedia ou Fès qui ont livrés des produits de référence « prdo2 » et « prod3 » dont la quantité est non comprise entre 250 et 500.
4. Effectuer un changement de taux de tva pour les produits dont le prix de vente dépasse 500 DHS.
5. Pour chaque Fournisseur, afficher le nombre de livraisons faites au cours du dernier jour de chaque mois. (exemple le dernier jour du mois de janvier est le 31,.. etc)

Exercice 7 :

Le schéma simplifié d'une base de données pour la gestion des commandes et des livraisons d'un système d'information, comprend les tables suivantes :



Les livraisons sont faites par des fournisseurs et la table « détail_livraisons » contient les références et les quantités livrées des produits de chaque livraison.

Questions :

1. Qu'est ce qu'une Base de données et en quoi consiste sa gestion ? Citer les types de requêtes que vous connaissez. A quoi sert une requête de type analyse croisée ? (2,5 points)
2. Donner la relation avec l'intégrité référentielle entre la table fournisseurs et livraisons. Donner la signification et l'utilité de l'intégrité référentielle dans ce cas. (2 points)
3. Que se passe t-il si on prend uniquement la « Référence » comme clé primaire de la table « Détail_livraisons » ? Que se passe t-il si on prend uniquement « Num_livraison » comme clé de cette même table. (2 points)
4. On suppose qu'on a établi les relations avec l'intégrité référentielle entre les différentes tables. Parmi les enregistrements suivants lesquels seront rejetés par le SGBD et pourquoi ? (2,5 points)

Table : Fournisseurs		
<i>Num_four</i>	<i>Nom</i>	<i>Ville</i>
1	amine	casablanca
2	ahmed	rabat
3	ali	fes
3	samir	agadir

Table : livraisons		
<i>Num_liv</i>	<i>Num_four</i>	<i>Dat_liv</i>
1	1	22/01/2009

Table : Détails_livraisons		
<i>Num_liv</i>	<i>Ref</i>	<i>Qte_liv</i>
1	R1	100

2	2	22/02/2009
3	5	10/03/2009
4	3	25/05/2009

1	R2	A66
2		200
2	R1	300
56	R2	80

5. Donner les requêtes suivantes en mode création :

- Afficher le numéro et la date des livraisons faites au cours de l'année 2009 dans le dernier mois de chaque trimestre. **(1,5 points)**
- Pour chaque livraison, afficher le numéro, la date, le nombre de jours qui sont passés depuis la création de la livraison, le montant hors taxe et le montant TTC avec taux de TVA spécifié comme paramètre. Limiter la sélection uniquement pour les livraisons dont la durée est inférieur à 100 jours et ayant un montant TTC entre 5000 DH et 10000 DH. **(2,5 points)**
- Afficher les quantités totales mensuelles livrées pour chaque produit au cours du premier semestre de l'année 2009. Les références des produits doivent être affichées en ligne et les noms des mois en colonne. **(2 points)**
- Effectuer une augmentation de prix de vente de 10% pour les produits dont la quantité en stock est comprise entre 1 et 100 au cours du dernier semestre de 2008. **(2,5 points)**
- Pour chaque ville des clients, afficher le nombre de commandes faites au cours des N derniers jours de la date d'aujourd'hui. Avec N un paramètre. **(2,5 points)**

Exercice 8 :

Le schéma simplifié d'une base de données pour la gestion des factures, d'un système d'information, comprend les tables suivantes :

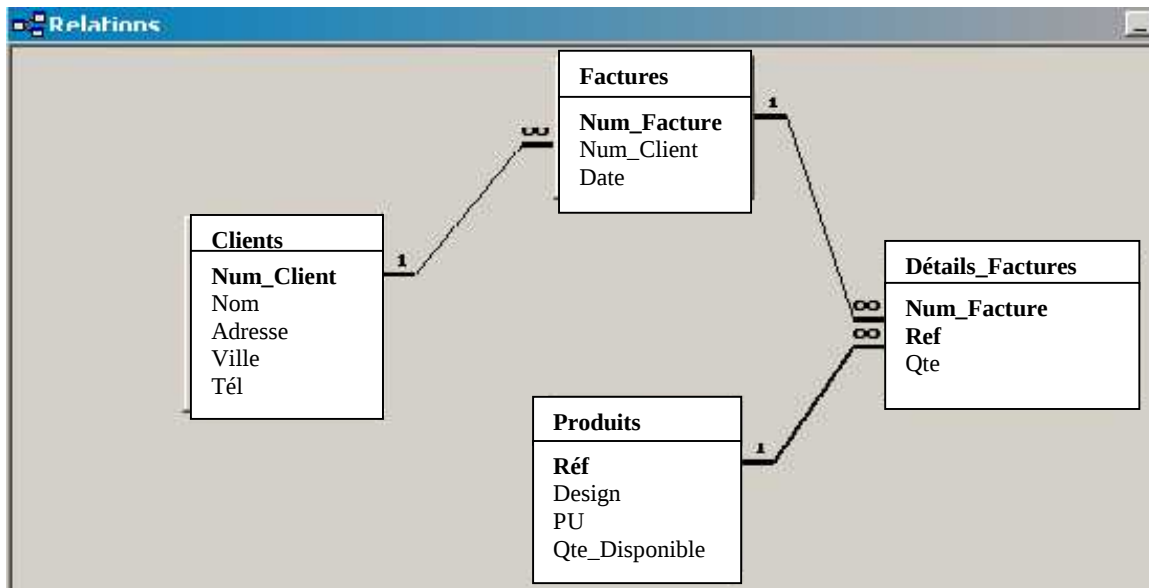
Clients : Table		
	Nom du champ	Type de données
	Num_Client	Numérique
	Nom	Texte
	Adresse	Texte
	Ville	Texte
	Tél	Texte

Produits : Table		
	Nom du champ	Type de données
	Réf	Texte
	design	Texte
	PU	Monétaire
	Qte_Disponible	Numérique

Factures : Table		
	Nom du champ	Type de données
	Num_Facture	NuméroAuto
	Num_Client	Numérique
	Date	Date/Heure

Détails_Factures : Table		
	Nom du champ	Type de données
	Num_Facture	Numérique
	Ref	Texte
	Qte	Numérique

Ces tables sont reliées entre elles par les relations suivantes :



6. Que se passe t-il si le champ « **Ref** » ne fait pas parti de la clé de la table « Détails_Factures » ?
Même question pour le champ « **Num_Facture** » ?
7. Que se passe t-il si on n'établit aucune relation entre la table « Produits » et « Détails_Factures » ?

8. Parmi les enregistrements suivants lesquels seront rejetés par le SGBD et pourquoi ?

Table : Factures		
Num_Facture	Num_Client	Date
22	12	22/10/04
23	12	22/10/04
23		10/10/04
25	13	

Table : Détails_Factures		
Num_Facture	Ref	Qte
22	R1	54
22	R2	A66
22		56
22	R1	66
56	R2	80

9. Soit la requête suivante :

```

SELECT Factures.Num_Facture, Factures.Date, Sum([Qte]*[PU]) AS Total
FROM Produits, Factures, Détails_Factures
WHERE Factures.num_Factures=Détails_Factures.num_Factures and
      Détails_Factures.Ref=Produits.Ref
GROUP BY Factures.Num_Facture, Factures.Date
HAVING (Factures.Date between D1 and D2) or Sum([Qte]*[PU])>10000;
  
```

- a- Quel résultat va afficher cette requête une fois exécuté ?
- b- Pourquoi on a introduit la jointure « Détails_Factures.Ref=Produits.Ref » ?
- c- Quel est le champ de la clause « Select » qui nous a obligé d'introduire la clause « Group by » ? et pourquoi ?

- d- Reformuler la requête ci-dessus pour qu'elle affiche le même résultat mais uniquement pour les clients de la ville de « casa ».

10. formuler en langage SQL les requêtes suivantes :

- a- Afficher le nombre de factures établis entre deux dates données comme paramètres.
- b- Afficher pour chaque client le montant HT et le montant TTC à payer par facture avec un taux de TVA variable.
- c- Afficher pour chaque produit la quantité moyenne facturée, la quantité totale facturée et la Quantité_Restante. Avec Quantité_Restante = quantité total – quantité disponible.
- d- Afficher les factures qui comprennent tous les produits et dont le montant HT est soit inférieur à 6000 soit supérieure à 14000.
- e- Afficher les clients ayant au moins deux factures dont chacune contient au moins deux produits.
- f- Afficher le nombre de factures qui contiennent tous les produits.

Corrigé et barèmes

1. (1pt)

- Une facture ne peut contenir alors qu'un seul produit. (0,5 pt)
- Un produit ne peut être facturer qu'une seule fois. (0,5 pt)

2. (1pt) On risque d'établir des factures pour des produits qui n'existent pas.

3. (3pts)

- Pour table factures :(1pt)
..3 éme enregistrement car la relation entre les tables clients et factures impose que le champ « Num_Client » de la table « factures » doit exister dans la table « Clients » et comme c'est la clé il ne peut pas être vide.
- Pour table « détails_factures » :(2pts)
..2éme enregistrement car le champ « Qte » doit être numérique (0,5 pt)
..3éme enregistrement car le champ « Ref » fait parti de la clé et ne peut donc pas être vide.(0,5 pt)
..4éme enregistrement car il s'agit d'une clé répétée(redondance). (0,5 pt)
..5éme enregistrement car aucune facture n'a le numéro « 56 ». (0,5 pt)

4. (4,5 pts)

- a- Afficher le montant total hors taxe pour les factures faites entre les dates D1 et D2 et ayant un montant total supérieure à 10000.(1 pt)
- b- Pour avoir pour chaque produit facturé le PU correspondant. Ainsi la quantité « Sum(Qte)*Pu » aura un sens.(1 pt)
- c- C'est le champ « Sum(Qte)*Pu » et le group by est nécessaire pour factoriser par num_facture. (1 pt)
- d- (1,5 pt)

```
SELECT Factures.Num_Facture, Factures.Date, Sum([Qte]*[PU]) AS Total
FROM Produits, Factures, Détails_Factures , Clients
WHERE Factures.num_Factures=Détails_Factures.num_Factures and
      Détails_Factures.Ref=Produits.Ref and Clients.Num_Client=Factures.Num_Client
And Ville= « Casa »
GROUP BY Factures.Num_Facture, Factures.Date
HAVING (Factures.Date between D1 and D2) or Sum([Qte]*[PU])>10000;
```

5. (10,5 pts)

a- (1,5 pt)

```
SELECT count(Factures.Num_Facture) AS Nbr_Factures
FROM Factures
WHERE Factures.Date between D1 and D2)
```

b- (2 pts)

```
SELECT Clients.Num_Client, Factures.Num_Facture, Factures.Date, Sum([Qte]*[PU]) AS MHT,
Sum([Qte]*[PU])*(1+TTVA) AS MTTC
FROM Clients, Factures, Produits, détails_factures
WHERE Factures.num_Factures=Détails_Factures.num_Factures and
      Détails_Factures.Ref=Produits.Ref and Clients.Num_Client=Factures.Num_Client
GROUP BY Clients.Num_Client, Factures.Num_Facture;
```

c- (1,5 pt)

```
SELECT Produits.Réf, avg(factures.Qte), sum(factures.Qte), sum(factures.Qte)-Qte_disponible as
qte_restante
FROM Produits,détails_factures
WHERE Détails_Factures.Ref=Produits.Ref
GROUP BY Produits.Réf ;
```

d- (2 pts)

```
SELECT détails_factures.Num_Fact, count(Réf), sum(Qte*Pu) AS HT
FROM Produits, détails_factures
WHERE Détails_Factures.Ref=Produits.Ref
GROUP BY détails_factures.Num_Facture
Having (sum(Qte*Pu)<6000 or sum(Qte*Pu)>14000) and count(Réf)=(select count(*) from produits);
```

e- (2 pts)

```
SELECT factures.cin, count(factures.num_fact)
FROM factures
GROUP BY factures.cin
Having count(factures.num_fact)>1 and factures.num_fact in (select num_fat, count(ref) from
détails_factures group by num_fact having count(ref)>1);
```

f- (1,5 pt)

```
SELECT count(factures.num_fact)
FROM factures
Where factures.num_fact in (select num_fact, count(ref) from détails_fact group by num_fact having
count(ref) = (select count(*) from produits));
```

Exercice 9 :

Les ventes des produits d'une entreprise sont gérées par une Base de Données dont le schéma simplifié est le suivant :



- Chaque vente est assurée par un et un seul représentant et porte sur un seul produit et concerne un seul client.
- Un représentant peut faire plusieurs ventes sur plusieurs produits pour plusieurs clients.
- Un client peut être concerné par plusieurs ventes.

Questions :

1. Qu'est ce qu'une requête et quelle est la différence entre une requête d'analyse croisée et une requête de mise à jour ? **(1,5points)**
2. Que se passe t-il si on prend le couple (N° **Produit**, N°**Représentant**) comme clé primaire de la table « **ventes** » ? **(1,5point)**
3. Que signifie l'intégrité référentielle entre les tables « **produits** » et « **ventes** » et qu'elle est son utilité dans ce cas ? **(1,5point)**
4. Donner les relations entre la table **ventes**, la table **représentants** et la table **clients** en précisant l'intégrité référentielle. **(1,5point)**
5. Soit la requête suivante :

Champ :	N° Représentant	Nom	N° Vente	Expr1: Mois([Date_Vente])	Expr2: Année([Date_Vente])
Table :	REPRESENTANTS	REPRESENTANTS	VENTES		
Opération :	Regroupement	Regroupement	Compte	Où	Où
Tri :					
Afficher :	☒	☒	☒	☐	☐
Critères :				<=6	2007

Dites qu'il sera le résultat de cette requête une fois exécutée. **(2points)**

6. Créer les requêtes suivantes :
 - a. Pour chaque représentant, afficher numéro, nom, le chiffre d'affaire réalisé par ce représentant et sa commission au cours du premier trimestre et dernier trimestre d'une année donnée comme paramètre. Notez bien que chaque représentant perçoit une commission de 10% sur le chiffre d'affaire réalisé. **(2,5points)**

Champ :	N.Représentant	Nom	Expr1: Somme([Produits+T]*[Nbrd'Unités])	Expr2: Somme([Produits+T]*[Nbrd'Unités])*1,1	Mois([date_vente])	Année([date_vente])
Table :	REPRESENTANTS	REPRESENTANTS				
Opération :	Regroupement	Regroupement	Expression	Expression	Où	Où
Tri :						
Afficher :	☒	☒	☒	☒	☒	☒
Critères :					<=	[*]
Où :					>=12 Et <=12	[*]

Ou

Champ :	N.Représentant	Nom	Expr1: Somme([Produits+T]*[Nbrd'Unités])	Expr2: Somme([Produits+T]*[Nbrd'Unités])*1,1	Mois([date_vente])	Année([date_vente])
Table :	REPRESENTANTS	REPRESENTANTS				
Opération :	Regroupement	Regroupement	Expression	Expression	Où	Où
Tri :						
Afficher :	☒	☒	☒	☒	☒	☒
Critères :					<=3 Ou >=10 Et <=12	[*]
Où :						

- b. Afficher pour chaque ville le nombre total d'unités vendues par les représentants dont le nom ne commence pas par la lettre « S » et dont le salaire n'est pas compris entre 2000 et 5000. (2,5points)

Champ :	Ville	Nbrd'Unités	Nom	SalaireBrut
Table :	CLIENTS	VENTES	REPRESENTANTS	REPRESENTANTS
Opération :	Regroupement	Somme	Où	Où
Tri :				
Afficher :	☒	☒	☒	☒
Critères :			Pas Comme "S**"	<2000 Ou >5000
Où :				

Ou

Champ :	Ville	Nbrd'Unités	Nom	SalaireBrut
Table :	CLIENTS	VENTES	REPRESENTANTS	REPRESENTANTS
Opération :	Regroupement	Somme	Où	Où
Tri :				
Afficher :	☒	☒	☒	☒
Critères :			Pas Comme "S**"	<2000
Où :			Pas Comme "S**"	>5000

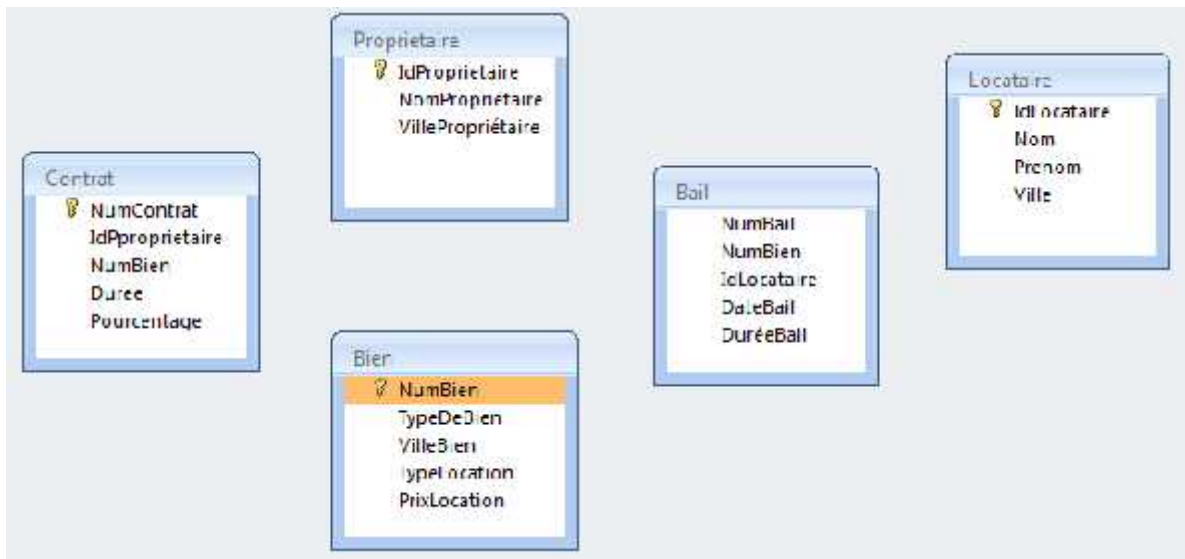
- c. Affichez les Chiffres d'Affaires mensuels réalisés dans les six derniers mois de l'année en cours, par catégorie de produit, les catégories doivent être affichées en colonne et les noms des mois en ligne. (3,5points)

Champ :	Expr1: Mois([date_vente];'mmmm')	Catégorie	Expr2: Somme([Produits+T]*[Nbrd'Unités])	Différence([*];[date_vente];Date())
Table :		PRODUITS		
Opération :	Regroupement	Regroupement	Expression	Où
Analyse :	Fin de ligne	Fin de ligne	Valeur	
Tri :				
Critères :				<=6
Où :				

- d. Effectuer une augmentation de salaire de 50% pour les représentants ayant réalisés au cours de l'année 2007 des ventes dont le montant est supérieur à 150000. **(3,5points)**

Exercice 10 :

Une agence immobilière gère la location (type meublée ou vide) des appartements et des maisons pour le compte des propriétaires à travers une base de données dont le schéma simplifié est ci-dessous:



- Chaque propriétaire remet en gestion à l'agence plusieurs appartements ou maisons par le biais d'un contrat, identifié par une durée et un pourcentage, propre à chaque bien.
- Chaque type de bien (appartement ou maison) est donné en location par un bail pour une durée déterminée à un locataire contre un prix mensuel, et le même locataire peut louer le même bien à des dates différentes.

I. Répondre aux questions suivantes :

1. Donner la définition de : Table et requête de sélection.
2. Expliquer l'utilité d'appliquer l'intégrité référentielle entre les tables « Proprietaire » et « Contrat ».
3. Que se passe-t-il si on prend le couple (NumBien, IdLocataire) comme clé primaire de la table « Bail » ? Déterminer la clé primaire de cette table.
4. Donner le schéma relationnel de la base de données ci-dessus.
5. Parmi les enregistrements suivants lesquels seront rejetés par le SGBD et pourquoi ?

Table : Locataire			
IdLocataire	Nom	Prenom	Ville
1	Amine		Casablanca
	Salim	Rajae	Tanger
3	Amine	Jamal	Agadir
4	Nasri	Reda	Tanger

Table : Bail				
NumBail	NumBien	IdLocataire	DateBail	DureeBail
123	11	2	1/3/2012	3
124		4	2/1/1013	5
125	1	3	10/4/013	2
124	1	3	2/1/2014	4

II. Soit la requête suivante :

Champ :	NumBien	TypeLocation	NumBail	DiffDate("aaaa",[DateBail]:Date())
Table :	Bien	Bien	Bail	
Opération :	Regroupement	Regroupement	Compte	Où
Tri :				
Afficher :	☒	☒	☒	☐
Critères :				<10
Où :				

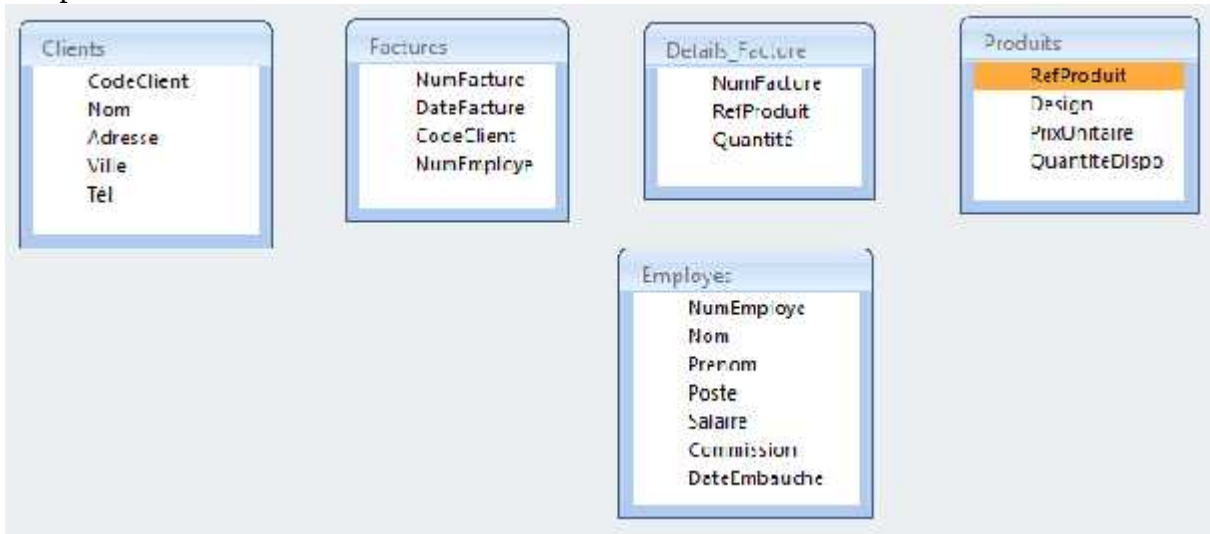
Reformuler cette requête en SQL.

III. Créer les requêtes suivantes en mode QBE ou SQL :

- Afficher le numéro, la ville et le prix de location des appartements meublés sur Casablanca ou Rabat avec un prix soit inférieur à 5000 ou supérieur à 8000.
- Afficher le numéro de bail, de bien et l'Id de locataire pour les bails effectués le premier ou le dernier jour des mois de Janvier, Avril ou Juin des années 2014 et 2015 (les mois sont exprimés en lettres).
- Afficher le nom, le prénom des locataires et la durée de leur bail effectué le mois courant des années 2014 et 2015. Limiter la sélection pour les bails d'une durée différente de 5 ans.
- Afficher le nombre de bails effectués entre deux dates données comme paramètres.
- Afficher pour chaque propriétaire le nom, la ville et le nombre total de ses biens mis en location avec un pourcentage supérieur à 9%.
- Afficher pour chaque bien le numéro, le type et le prix total annuel de location. Limiter la sélection pour les biens meublés mis en location à des durées inférieures à 6 ans.

Exercice 11 :

Le schéma simplifié d'une base de données pour la gestion de la facturation d'une entreprise, comprend les tables suivantes :



Soient les règles de gestion suivantes :

- Un client peut avoir plusieurs factures, et une facture appartient à un seul client.
- Un produit peut être contenu dans plusieurs factures, et une facture ne peut contenir pas qu'un seul produit.
- Une facture est établie par un seul employé, et ce dernier peut générer plusieurs factures.
- Poste désigne l'occupation que l'employé occupe dans l'entreprise.

I. Répondre aux questions suivantes :

- Donner la définition de : Base de données, Table et Relation.
- Citer les types de relation que vous connaissez. Expliquer Comment peut-on interpréter la relation de type plusieurs à plusieurs.
- A quoi sert la clé primaire ? Déterminer les clés primaires de chaque table en utilisant uniquement les champs cités dans les tables.
- Donner le schéma relationnel de la base de données ci-dessus.
- Ma requête doit afficher le numéro, le nom et le prénom de chaque employé ainsi que le nombre des factures qu'il a établi entre deux dates données comme paramètres. Exprimer cette requête en SQL.

II. Créer les requêtes suivantes en mode QBE ou SQL :

- a. Afficher le nom et l'adresse de tous les clients qui habitent Mohammedia ou Rabat et dont le nom commence par la lettre "E" et l'avant dernière lettre est également "E".
- b. Afficher le numéro et la date des factures effectuées le dernier jour des mois de janvier, avril ou juillet au cours des années 2015 et 2016.
- c. Afficher le prénom, le nom et l'ancienneté de tous les employés sauf les vendeurs. Limiter la sélection pour les employés avec une ancienneté supérieure à 10 ans.
- d. Affichez pour chaque poste la somme des revenus des employés embauchés en 2010 (Revenu = Salaire + Commission).
- e. Afficher pour chaque employé le nom, le prénom et le nombre de factures qu'il a établi pendant ses trois premières années de service.