

Corrigés exercices SQL

ADMINISTRATEUR/DEVELOPPEUR DE BASES DE DONNEES



Sommaire

Administrateur/Développeur de Bases de Données.....	1
Sommaire	2
I. CREATION DES TABLES DE LA BASE COMMANDE	5
I.1 CREATION DES TABLES CLIENT, COMMANDE, ARTICLE,.....	5
I.1.1 Clés primaires	5
I.1.2 Clés étrangères	5
I.1.3 Table Représentant	5
I.1.4 Table Client.....	5
I.1.5 Table Article.....	6
I.1.6 Table Commande	6
I.1.7 Table Ligne de commande.....	6
I.2 Insertion des données	6
I.3 Augmentation de la taille d'une colonne	6
I.4 Diminution de la taille d'une colonne	6
I.5 Ajout d'une colonne à la table Client	6
II. LANGAGE DE MANIPULATION DE DONNEES.....	7
II.1 Visualiser la structure d'une table.....	7
II.2 Contenu d'une table.....	7
II.3 Afficher la raison sociale de tous les clients.....	7
II.4 Noms et identifiants des clients	7
II.5 Clients dont le code postal est compris dans un intervalle.....	7
II.6 Clients dont le code est égal à 1 ou à 5	7
II.7 Clients dont le nom commence par A.....	7
II.8 Clients dont le code postal commence par 3	7
II.9 Clients dont le nom ne commence pas par A.....	7
II.10 Clients sans adresse.....	7
II.11 Tri sur le nom des clients.....	7
II.12 Clients qui ne sont pas de Paris.....	8
II.13 Clients dont le nom contient une chaîne de caractères.....	8
II.14 Commandes passées il y a plus de trois mois	8
II.15 Représentants dont l'identifiant est supérieur à 2	8
II.16 Représentants dont le nom ne commence pas par G.....	8

II.17	Représentants dont l'initiale du prénom est comprise entre D et M	8
II.18	Représentants dont le nom commence par P ou Y.....	8
II.19	Représentant dont l'adresse contient 'rue' majuscule ou minuscule.....	8
II.20	Représentant dont le prénom est entre D et Z et nom commence par L ou A.....	8
II.21	Affichage de la date de commande au format 'MM-YYYY'	9
II.22	Affichage des dates de commandes sous format mois et année.....	9
III.	MODIFICATIONS DES TABLES	10
III.1	Suppression de lignes d'une table.....	10
III.2	Modification des lignes d'une table	10
III.3	Mise à jour de la date de commande.....	10
IV.	FONCTIONS PREDEFINIES	11
IV.1	Date courante du système	11
IV.2	L'heure courante du système.....	11
V.	JOINTURES.....	12
V.1	Clients ayant passé une commande avec les articles.....	12
V.2	Clients qui ont commandé et qui n'ont pas commandé	12
V.3	Articles non commandés	12
V.4	Noms des clients et de leurs représentants.....	13
V.5	Désignation et prix unitaire des articles de la commande n°1.....	13
V.6	Articles commandés au mois de février de l'année dernière	13
V.7	Clients parisiens qui ont commandé pendant les deux dernières années.....	13
V.8	Clients qui ont commandé cette année	14
V.9	Liste des clients qui ont passé une commande deux ans auparavant	14
V.10	Clients qui ont commandé pendant une période donnée	14
V.11	Les clients et leurs représentants même ceux qui n'en n'ont pas	14
VI.	FONCTIONS D'AGREGATS.....	15
VI.1	Nombre de commande	15
VI.2	Nombre de ligne par commande	15
VI.3	Nombre de commandes par client.....	15
VI.4	Nombre de clients par représentant.....	15
VI.5	Nombre de clients sans représentant	16
VI.6	Représentants qui n'ont pas de clients	16
VI.7	Montant total des commandes.....	16
VI.8	Montant de chaque commande.....	16
VI.9	Montant commandé pour chaque article	16

VI.10	Montant des commandes par jour.....	17
VI.11	Montant moyen d'une commande	17
VI.12	Articles dont la quantité commandée est supérieure à 5	17
VI.13	Commandes dépassant 1197.00€	17
VI.14	Commandes avec au moins deux lignes.....	17
VI.15	Articles en stock	18
VI.16	Total pour chaque commande dont le nombre de ligne est supérieur à 2.....	18
VII.	SOUS-REQUETES.....	19
VII.1	Articles dont le prix est inférieur a la moyenne	19
VII.2	Articles dont la quantité en stock est supérieur à la quantité commandée	19
VII.3	Article(s) ayant le stock le plus élevé	19
VII.4	Articles non-commandés.....	20
VII.5	Pour chaque commande les articles les plus commandés.....	20
VII.6	Article dont la quantité en stock est 30 fois supérieure à toute quantité commandée	20
VII.7	Article dont la quantité en stock est 30 fois supérieur à au moins une quantité commandée.....	21
VII.8	Clients des articles les plus commandés	21
VIII.	AUTRES COMMANDES LDD	22
VIII.1	Création de la table Client_paris à partir de client.....	22
VIII.2	Renommer la table client_paris	22
VIII.3	Création d'une table mail a partir de clients.....	22
VIII.4	Même exercice avec les clients qui n'ont jamais commandé	22
VIII.5	Création d'un savepoint et annulation d'une transaction	22
IX.	VUES ET INDEX SUR LES TABLES.....	23
IX.1	Nom des clients avec un update	23
IX.2	Désignation et quantité commandée d'un article avec update.....	23
IX.3	Création d'une vue sur la table ligne_com.....	23
IX.4	Index.....	23
IX.5	Index.....	23

I. CREATION DES TABLES DE LA BASE COMMANDE

I.1 CREATION DES TABLES CLIENT, COMMANDE, ARTICLE,...

I.1.1 Clés primaires

- représentant: idrep
- client : idclient
- article: idarticle
- commande: numcom
- ligne_com : (numcom,idarticle)

I.1.2 Clés étrangères

- Client: idrep
- Commande: idclient
- Ligne_com: (numcom, idarticle)

I.1.3 Table Représentant

```
CREATE TABLE representant(  
    idrep          int (5) Auto_increment NOT NULL ,  
    Prenomrep     Varchar (20) ,  
    Nomrep        Varchar (20) ,  
    adresse       Varchar (20) ,  
    codepostal    Int(5) ,  
    ville         Varchar (30) ,  
    telephone     Varchar (10) ,  
    PRIMARY KEY (idrep )  
)ENGINE=InnoDB;
```

I.1.4 Table Client

```
CREATE TABLE client(  
    idclient       int (5) Auto_increment NOT NULL ,  
    raisonsociale Varchar (20) ,  
    adresse        Varchar (60) ,  
    codepostal     Int(5) ,  
    ville          Varchar (30) ,  
    telephone      Varchar (10) ,  
    idrep          Int(5) ,  
    PRIMARY KEY (idclient )  
)ENGINE=InnoDB;  
ALTER TABLE client ADD CONSTRAINT FK_client_idrep FOREIGN KEY (idrep)  
REFERENCES representant(idrep);
```

I.1.5 Table Article

```
CREATE TABLE article(  
    idarticle    int (5) Auto_increment    NOT NULL ,  
    designation  Varchar (30) ,  
    prixunit    Decimal (8,2) ,  
    qtestock    Int(10) ,  
    PRIMARY KEY (idarticle )  
)ENGINE=InnoDB;
```

I.1.6 Table Commande

```
CREATE TABLE commande(  
    numcom    int (5) Auto_increment    NOT NULL ,  
    datecom   Date ,  
    idclient  Int(5) ,  
    PRIMARY KEY (numcom )  
)ENGINE=InnoDB;  
ALTER TABLE commande ADD CONSTRAINT FK_commande_idclient FOREIGN KEY  
(idclient) REFERENCES client(idclient);
```

I.1.7 Table Ligne de commande

```
CREATE TABLE ligne_com(  
    numcom    Int NOT NULL ,  
    idarticle Int NOT NULL ,  
    qtecom    Int ,  
    numlig    Int ,  
    PRIMARY KEY (numcom ,idarticle )  
)ENGINE=InnoDB;  
  
ALTER TABLE ligne_com ADD CONSTRAINT FK_concerner_numcom FOREIGN KEY  
(numcom) REFERENCES commande(numcom);  
ALTER TABLE ligne_com ADD CONSTRAINT FK_concerner_idarticle FOREIGN KEY  
(idarticle) REFERENCES article(idarticle);
```

I.2 Insertion des données

Insérer les données suivantes dans les tables (source GestCom.sql)

Utilisez le script fourni : GestCom.sql .

I.3 Augmentation de la taille d'une colonne

```
ALTER TABLE client MODIFY column adresse VARCHAR(100);
```

I.4 Diminution de la taille d'une colonne

On ne peut diminuer la taille d'une colonne d'une table que si cette taille est supérieure ou égale à toute donnée dans cette colonne.

I.5 Ajout d'une colonne à la table Client

```
ALTER TABLE client ADD column col int(3);
```


II. LANGAGE DE MANIPULATION DE DONNEES

II.1 Visualiser la structure d'une table

```
DESC client;
```

II.2 Contenu d'une table

```
SELECT * FROM client;
```

II.3 Afficher la raison sociale de tous les clients

```
SELECT raisonsociale FROM client;
```

II.4 Noms et identifiants des clients

```
SELECT idclient,raisonsociale FROM client;
```

II.5 Clients dont le code postal est compris dans un intervalle

```
SELECT *  
FROM client  
WHERE codepostal BETWEEN 60000 AND 95000;
```

II.6 Clients dont le code est égal à 1 ou à 5

```
SELECT *  
FROM client  
WHERE idclient=1 OR idclient= 5;
```

II.7 Clients dont le nom commence par A

```
SELECT * FROM client  
WHERE raisonsociale LIKE 'A%';
```

II.8 Clients dont le code postal commence par 3

```
SELECT * from client  
WHERE codepostal LIKE '3%';
```

On remarque que Oracle fait une conversion automatique de nombre vers chaîne de caractères.

II.9 Clients dont le nom ne commence pas par A

```
SELECT * FROM client  
WHERE raisonsociale NOT LIKE 'A%';
```

II.10 Clients sans adresse

```
SELECT * FROM client  
WHERE adresse IS null;
```

II.11 Tri sur le nom des clients

```
SELECT idclient, raisonsociale  
FROM client  
ORDER BY raisonsociale desc;
```

II.12 Clients qui ne sont pas de Paris

```
SELECT * FROM client
WHERE lower(ville) NOT LIKE 'paris';
```

II.13 Clients dont le nom contient une chaîne de caractères

```
SELECT * FROM client
WHERE raisonsociale LIKE '%er%';
```

II.14 Commandes passées il y a plus de trois mois

```
SELECT numcom,datecom FROM commande WHERE
datecom>=SUBDATE(current_date(), INTERVAL 3 month);
```

numcom	datecom
4	2016-03-03
5	2016-02-28
6	2016-04-02

II.15 Représentants dont l'identifiant est supérieur à 2

```
SELECT * FROM representant
WHERE idrep >2
ORDER BY idrep DESC;
```

II.16 Représentants dont le nom ne commence pas par G

```
SELECT * FROM representant
WHERE nomrep NOT LIKE 'G%';
```

II.17 Représentants dont l'initiale du prénom est comprise entre D et M

```
SELECT * FROM representant
WHERE substr(prenomrep,1,1) BETWEEN 'D' AND 'M';
```

II.18 Représentants dont le nom commence par P ou Y

```
SELECT * FROM representant
WHERE substr(nomrep,1,1) IN ('P','Y');
```

II.19 Représentant dont l'adresse contient 'rue' majuscule ou minuscule

```
SELECT * FROM representant
WHERE upper(adresse) LIKE '%RUE%';
```

II.20 Représentant dont le prénom est entre D et Z et nom commence par L ou A

```
SELECT nomrep, prenomrep
FROM representant
WHERE upper(substr(prenomrep, 1, 1)) BETWEEN 'D' AND 'Z'
AND (upper(substr(nomrep, 1, 1)) IN ('A', 'L'))
SQL> /
```

NOMREP

PRENOMREP

lebeaux
Legrand

Patrick
David

II.21 Affichage de la date de commande au format 'MM-YYYY'

```
SELECT DATE_FORMAT (datecom, '%m-%y') FROM commande;
```

II.22 Affichage des dates de commandes sous format mois et année

```
SELECT numcom, DATE_FORMAT(datecom, '%M-%Y') FROM commande;
```

III. MODIFICATIONS DES TABLES

III.1 *Suppression de lignes d'une table*

```
DELETE FROM ligne_com WHERE idarticle=1;
```

Retour en arrière : OUI en faisant un ROLLBACK

III.2 *Modification des lignes d'une table*

```
UPDATE client SET adresse='21, Rue Martelet' WHERE idclient=4;
```

III.3 *Mise à jour de la date de commande*

```
UPDATE commande SET datecom=curdate() WHERE idclient=3;  
UPDATE commande SET datecom=subdate(curdate(), interval 4 day) WHERE  
idclient=5;
```

Ou encore

```
UPDATE commande  
SET datecom = case idclient  
WHEN 3 THEN curdate()  
WHEN 5 THEN subdate(curdate(), interval 4 day)  
END  
WHERE idclient in (3,5);
```

IV. FONCTIONS PREDEFINIES

IV.1 *Date courante du système*

```
SELECT DATE_FORMAT(curdate(), '%d-%m-%Y') Date_jour;
```

IV.2 *L'heure courante du système*

```
SELECT DATE_FORMAT(NOW(), '%T') Heure;  
SELECT curtime() Heure;
```

V. JOINTURES

V.1 Clients ayant passé une commande avec les articles

```
SELECT c.raisonsociale, co.numcom, l.idarticle
FROM client c JOIN commande co
      ON c.idclient=co.idclient
      JOIN ligne_com l
      ON co.numcom=l.numcom
ORDER BY 1,2 ;
```

raisonsociale	numcom	idarticle
CARREFOUR	3	5
CARREFOUR	3	1
CARREFOUR	3	2
CD PRO	2	4
DARTY	1	2
DARTY	1	6
DARTY	4	6
DARTY	4	1
DARTY	4	2
INFO SYSTEM	5	1
Multimedia	6	3
Multimedia	6	4
Multimedia	6	6

13 rows in set (0.00 sec)

V.2 Clients qui ont commandé et qui n'ont pas commandé

```
select raisonsociale,ifnull(numcom,' Pas de commande ') 'Statut de la
commande' from client c left join commande co using(idclient);
```

raisonsociale	Statut de la commande
DARTY	1
DARTY	4
INFO SYSTEM	5
CD PRO	2
Multimedia	6
CARREFOUR	3
THOMSON	Pas de commande

7 rows in set (0.00 sec)

V.3 Articles non commandés

```
SELECT a.designation, l.numcom
FROM article a LEFT JOIN ligne_com l
ON a.idarticle=l.idarticle
WHERE l.idarticle IS NULL ;
```

DESIGNATION	NUMCOM
TV LED-3D	NULL

Pour y voir plus clair :

```
SELECT designation, a.idarticle "Dans article", l.idarticle "Dans  
ligne_com"  
FROM article a LEFT JOIN ligne_com l  
ON (a.idarticle=l.idarticle)  
-- WHERE l.idarticle IS NULL
```

V.4 Noms des clients et de leurs représentants

```
SELECT c.raisonsociale, r.nomrep  
FROM client c JOIN representant r  
USING(idrep);
```

V.5 Désignation et prix unitaire des articles de la commande n°1

```
SELECT l.numcom, a.designation, a.prixunit  
FROM article a JOIN ligne_com l  
ON a.idarticle=l.idarticle  
WHERE l.numcom=1;
```

V.6 Articles commandés au mois de février de l'année dernière

```
SELECT designation, datecom  
FROM commande JOIN ligne_com  
ON ligne_com.numcom=commande.numcom  
JOIN article  
ON article.idarticle=ligne_com.idarticle  
WHERE  
DATE_FORMAT(datecom,'%Y') = DATE_FORMAT(now()-Interval 1  
Year,'%Y')  
and  
DATE_FORMAT(datecom,'%m') = '02';
```

V.7 Clients parisiens qui ont commandé pendant les deux dernières années

```
/* Darty passe la cde n° 7 */  
INSERT INTO commande(idclient,datecom) VALUES(1,'2017/12/12');  
COMMIT ;
```

```
SELECT raisonsociale, ville,datecom  
FROM client c, commande co  
WHERE c.idclient=co.idclient  
AND DATE_FORMAT(datecom,'%Y') in  
(  
    (DATE_FORMAT(now()-Interval 1  
    Year,'%Y'),  
    (DATE_FORMAT(now()-Interval 2  
    Year,'%Y'))  
)  
AND upper(ville)='PARIS';
```

V.8 Clients qui ont commandé cette année

```
SELECT cl.idclient,raisonsociale
FROM client cl
JOIN commande co ON (cl.idclient=co.idclient)
WHERE DATE_FORMAT(datecom,'%Y') = DATE_FORMAT(curdate(),'%Y') ;
```

V.9 Liste des clients qui ont passé une commande deux ans auparavant

```
/*DARTY passe la cde n° 8*/
INSERT INTO commande(datecom,idclient) Values('2016-04-15',1);
COMMIT;
```

```
SELECT DISTINCT raisonsociale FROM client cl, commande co
WHERE cl.idclient=co.idclient
AND DATE_FORMAT(datecom,'%Y')= DATE_FORMAT(curdate(),'%Y') - 2;
```

V.10 Clients qui ont commandé pendant une période donnée

Liste des clients qui ont commandés les trois premiers mois de l'année dernière.

```
SELECT cl.idclient,cl.raisonsociale , co.datecom
FROM client cl
JOIN commande co ON (cl.idclient=co.idclient)
WHERE DATE_FORMAT(datecom,'%Y%m') BETWEEN '201701' AND '201703';
```

V.11 Les clients et leurs représentants même ceux qui n'en n'ont pas

```
select raisonsociale, nomrep from client right join representant
using(idrep);
```

raisonsociale	nomrep
NULL	Dupond
NULL	Fontaine
NULL	Hoareau
DARTY	Sanchez
INFO SYSTEM	Sanchez
CD PRO	Garcia
Multimedia	Garcia
NULL	lebeaux
CARREFOUR	Legrand

VI. FONCTIONS D'AGREGATS

VI.1 Nombre de commande

Afficher le Nombre de commande.

```
SELECT count(numcom)
FROM commande;
```

VI.2 Nombre de ligne par commande

Afficher le nombre de ligne par commande.

```
SELECT concat('commande n°',numcom) AS ``COMMANDE``,COUNT(numcom)
FROM ligne_com
GROUP BY numcom ;
```

VI.3 Nombre de commandes par client

Afficher le nombre de commandes par client.

```
SELECT c.idclient,c.raisonsociale,count(numcom) "Nombre de commandes"
FROM client c
JOIN commande co ON c.idclient=co.idclient
GROUP BY c.idclient,c.raisonsociale ;
```

idclient	raison sociale	Nombre de commandes
1	DARTY	3
2	INFO SYSTEM	1
3	CD PRO	1
4	Multimedia	1
5	CARREFOUR	1

5 rows in set (0.00 sec)

VI.4 Nombre de clients par représentant

Afficher le nombre de clients par représentant

```
SELECT c.idrep,r.nomrep,count(c.idclient) AS "Nb de clients"
FROM client c JOIN representant r
ON c.idrep=r.idrep
GROUP BY c.idrep,nomrep;
```

idrep	nomrep	Nb de clients
4	Sanchez	2
5	Garcia	2
7	Legrand	1

3 rows in set (0.00 sec)

ou

```
SELECT idrep, count(idclient)
FROM client
WHERE idrep is not null
GROUP BY idrep;
```

VI.5 Nombre de clients sans représentant

Idem avec les clients qui n'ont pas de représentant

```
SELECT count(idclient) FROM client
WHERE idrep IS null;
```

VI.6 Représentants qui n'ont pas de clients

Recherche des représentants qui n'ont pas de clients

```
SELECT rep.nomrep, COUNT(c.idclient) nbre_clients
FROM client c RIGHT JOIN representant rep
ON c.idrep=rep.idrep
GROUP BY rep.nomrep
HAVING COUNT(c.idclient)=0;
```

nomrep	nbre_clients
Dupond	0
Fontaine	0
Hoareau	0
lebeaux	0

4 rows in set (0.01 sec)

VI.7 Montant total des commandes

```
/* Calcul du montant total de toutes les commandes.
*/
SELECT SUM(prixunit*qtecom) AS "Chiffre d'affaire"
FROM article JOIN ligne_com
USING(idarticle) ;
```

VI.8 Montant de chaque commande

Calcul du montant de chaque commande.

```
SELECT SUM(a.prixunit*l.qtecom) AS "Montant commandé", numcom
FROM article a
JOIN ligne_com l ON(a.idarticle=l.idarticle)
GROUP BY numcom
ORDER BY 1;
```

VI.9 Montant commandé pour chaque article

Calcul du montant commandé pour chaque article

```
SELECT SUM(a.prixunit*l.qtecom) AS "Montant commandé", a.designation
FROM article a
```

```
JOIN ligne_com l ON(a.idarticle=l.idarticle)
GROUP BY l.idarticle, a.designation
ORDER BY l;
```

VI.10 Montant des commandes par jour

Liste du montant des commandes par date de commande.

```
/* Montant des commandes par date de commande de toute les commandes */
SELECT datecom Date_de_commande ,sum(qtecom* prixunit) montant_par_date
FROM article a
JOIN ligne_com l USING(idarticle)
JOIN commande co USING(numcom)
GROUP BY datecom;
```

VI.11 Montant moyen d'une commande

Montant moyen d'une commande.

```
SELECT (SUM(Prixunit*qtecom))/COUNT(DISTINCT numcom) AS "Moyenne des montants"
FROM ligne_com l JOIN article a
ON l.idarticle=a.idarticle ;
```

VI.12 Articles dont la quantité commandée est supérieure à 5

Liste des articles dont la quantité commandée est supérieure à 5.

```
SELECT a.idarticle,designation,SUM(qtecom) somme
FROM article a
JOIN ligne_com l ON(a.idarticle=l.idarticle)
GROUP by a.idarticle,a.designation
HAVING SUM(qtecom)>5;
```

VI.13 Commandes dépassant 1197.00€

Liste et total des commandes dont la somme dépasse 1197.00 €.

```
SELECT l.numcom, SUM(a.prixunit*l.qtecom) somme
FROM article a
JOIN ligne_com l ON (a.idarticle=l.idarticle)
GROUP BY l.numcom
HAVING sum(a.prixunit*l.qtecom)>1197.00;
```

VI.14 Commandes avec au moins deux lignes

Liste des commandes qui ont au moins deux lignes.

```
SELECT numcom, count(numlig) FROM ligne_com
GROUP BY numcom
HAVING count(numlig)>=2;
```

VI.15 Articles en stock

```
SELECT a.idarticle, a.designation, qtestock-sum(qtecom) Reste
FROM article a
      LEFT JOIN ligne_com l ON(a.idarticle=l.idarticle)
GROUP BY a.idarticle,a.designation, qtestock;
```

VI.16 Total pour chaque commande dont le nombre de ligne est supérieur à 2

```
SELECT l.numcom, SUM(a.prixunit*qtecom)
FROM article a JOIN ligne_com l
ON l.idarticle=a.idarticle
GROUP BY l.numcom
HAVING COUNT(a.idarticle)>2;
```

VII. SOUS-REQUETES

VII.1 Articles dont le prix est inférieur à la moyenne

Liste des articles dont le prix est inférieur à la moyenne.

```
SELECT designation, prixunit
FROM article
WHERE prixunit < (SELECT AVG(prixunit)
                  FROM article);
```

designation	prixunit
Moniteur 28"	300.00
Moniteur 19"	169.00
Disque dur 5To	240.00
Disque dur 2To	104.00
Clavier SF	79.00
UC HP500-310	514.00

6 rows in set (0.00 sec)

VII.2 Articles dont la quantité en stock est supérieur à la quantité commandée

Lister tous les articles dont la quantité en stock est supérieure à toute quantité commandée du même article

```
SELECT DISTINCT a.idarticle,a.designation,a.qtestock
FROM article a
WHERE a.qtestock >ALL(SELECT (li.qtecom)
                     FROM ligne_com li WHERE li.idarticle= a.idarticle)
ORDER BY a.idarticle
```

idarticle	designation	qtestock
1	Moniteur 28"	30
2	Moniteur 19"	20
3	Disque dur 5To	250
4	Disque dur 2To	350
5	Clavier SF	200
6	UC HP500-310	50
7	TV LED-3D	10

7 rows in set (0.00 sec)

VII.3 Article(s) ayant le stock le plus élevé

Liste du ou des articles dont la quantité en stock est la plus élevée.

```
SELECT idarticle,designation,qtestock
FROM article WHERE qtestock= (SELECT MAX(qtestock)
                             FROM article);
```

idarticle	designation	qtestock
-----------	-------------	----------

```
|          4 | Disque dur 2To |          350 |
+-----+-----+-----+
```

ou

```
SELECT idarticle,designation,qtestock
FROM article
WHERE qtestock>= ALL(SELECT qtestock
                     FROM article);
```

VII.4 Articles non-commandés

Liste des articles qui n'ont pas été commandé

```
SELECT a.idarticle, designation FROM article a
WHERE a.idarticle NOT IN (SELECT idarticle FROM ligne_com);
```

```
+-----+-----+
| idarticle | designation |
+-----+-----+
|          7 | TV LED-3D   |
+-----+-----+
1 row in set (0.00 sec)
```

VII.5 Pour chaque commande les articles les plus commandés

Pour chaque commande rechercher le ou les articles qui ont été le plus commandé.

```
SELECT l.numcom, l.qtecom, l.idarticle,a.designation
FROM ligne_com l JOIN article a
ON l.idarticle = a.idarticle
WHERE l.qtecom = (SELECT MAX(l2.qtecom)
FROM ligne_com l2
WHERE l.numcom = l2.numcom
GROUP BY l2.numcom)
ORDER BY l.numcom,l.idarticle
```

```
+-----+-----+-----+-----+
| numcom | qtecom | idarticle | designation |
+-----+-----+-----+-----+
|        1 |        2 |          6 | UC HP500-310 |
|        2 |        1 |          4 | Disque dur 2To |
|        3 |        3 |          2 | Moniteur 19" |
|        4 |        3 |          1 | Moniteur 28" |
|        5 |        1 |          1 | Moniteur 28" |
|        6 |        3 |          6 | UC HP500-310 |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

VII.6 Article dont la quantité en stock est 30 fois supérieure à toute quantité commandée

Sans calculer le max, rechercher les articles dont la quantité en stock est plus de 30 fois supérieure à toute quantité commandée de l'article.

```
/* Le résultat comporte même les articles non commandés */
SELECT a.idarticle,a.designation,a.qtestock, IFNULL(l.qtecom,0)
FROM article a
```



```
LEFT JOIN ligne_com l ON (l.idarticle = a.idarticle)
WHERE a.qtestock/30 > ALL (SELECT l2.qtecom
FROM ligne_com l2 WHERE l2.idarticle = a.idarticle);
```

VII.7 Article dont la quantité en stock est 30 fois supérieur à au moins une quantité commandée

```
SELECT DISTINCT a.idarticle,a.designation,a.qtestock
FROM article a JOIN ligne_com l
ON l.idarticle = a.idarticle
WHERE a.qtestock/30 > ANY (SELECT l2.qtecom
FROM ligne_com l2
WHERE l2.idarticle = a.idarticle )
```

idarticle	designation	qtestock
3	Disque dur 5To	250
4	Disque dur 2To	350
5	Clavier SF	200

3 rows in set (0.00 sec)

VII.8 Clients des articles les plus commandés

```
SELECT c.raisonsociale,a.designation,a.prixunit,l.qtecom,l.numcom
FROM Client c JOIN Commande co
ON co.idclient = c.idclient
JOIN ligne_com l ON l.numcom=co.numcom
JOIN article a ON a.idarticle = l.idarticle
WHERE l.qtecom = ( SELECT max(l2.qtecom)
FROM ligne_com l2 JOIN article a2
ON a2.idarticle=l2.idarticle
WHERE l2.numcom=l.numcom
AND a2.prixunit >= (SELECT AVG(a3.prixunit)
FROM article a3)
GROUP BY l2.numcom);
```

VIII. AUTRES COMMANDES LDD

VIII.1 Création de la table *Client_paris* à partir de *client*

```
Create table Client_Paris
As select idclient , raisonsociale , adresse , idrep , telephone
From client
Where upper(ville)='PARIS' ;
```

VIII.2 Renommer la table *client_paris*

```
RENAME client_Paris TO Client_Par;
```

VIII.3 Création d'une table *mail* à partir de *clients*

```
CREATE TABLE Mail as SELECT raisonsociale,adresse, concat(codepostale,
'-' ,ville) FROM client;
```

VIII.4 Même exercice avec les clients qui n'ont jamais commandé

```
CREATE TABLE Commande_zero
AS SELECT raisonsociale,adresse,codepostal,ville
FROM client c
WHERE NOT EXISTS (select null from commande co
                  where co.idclient = c.idclient);
```

VIII.5 Création d'un *savepoint* et annulation d'une transaction

```
savepoint S1;
delete from representant where idrep=1;
rollback to S1;
```

IX. VUES ET INDEX SUR LES TABLES

IX.1 Nom des clients avec un update

```
CREATE OR REPLACE VIEW nom_cli(idcl,nom)
AS SELECT idclient,raisonsociale FROM client;
```

```
UPDATE nom_cli SET nom='NOM' WHERE idcl=1;
```

IX.2 Désignation et quantité commandée d'un article avec update

```
CREATE OR REPLACE VIEW vue_art_com (code,libelle,qte)
AS SELECT a.idarticle,a.designation,sum(l.qtecom)
   FROM article a JOIN ligne_com l
   ON (a.idarticle=l.idarticle)
GROUP BY a.idarticle,a.designation;
```

```
UPDATE vue_art_com
SET code=1 ;
```

On ne peut pas faire de UPDATE, de DELETE et de INSERT sur une vue avec jointure (colonnes virtuelles).

IX.3 Création d'une vue sur la table ligne_com

```
CREATE OR REPLACE view vue_lig
AS SELECT a.designation,a.prixunit,l.numcom,l.numlig,l.qtecom
   FROM article a JOIN ligne_com l
   ON (a.idarticle=l.idarticle);INDEX
```

IX.4 Index

```
CREATE INDEX ind_sur_rep ON Representant (idrep,nomrep,prenomrep);
```

IX.5 Index

```
CREATE INDEX ind_sur_com ON Commande(numcom,idclient);
```