

2023

# Corrigés langage SQL

## **BASE DE DONNÉES**

DIDIER TEXIER

DTX | VERSION 20180823-13664



# SOMMAIRE

---

1	CREATION DES TABLES .....	4
2	LANGAGE DE MANIPULATION DE DONNEES .....	6
3	MODIFICATIONS DES TABLES .....	21
4	FONCTIONS PREDEFINIES .....	9
5	JOINTURES .....	10
6	FONCTIONS D'AGREGATS .....	14
7	SOUS-REQUETES .....	18
8	AUTRES COMMANDES LDD .....	21
9	VUES SUR LES TABLES .....	24
10	SYNONYMES, SEQUENCES, INDEX, CLUSTERS .....	26



# 1 CREATION DES TABLES

## 1.1 CREATION DES TABLES CLIENT, COMMANDE, ARTICLE, ...

### 1.1.1 Clés primaires :

- Représentant : idrep
- Client : idclient
- Article : idarticle
- Commande : numcom
- Ligne\_com : (numcom,idarticle)

### 1.1.2 Clés étrangères :

- Client : idrep
- Commande : idclient
- Ligne\_com : (numcom, idarticle)

### 1.1.3 Table Représentant

```
CREATE TABLE Représentant
(idrep      NUMBER(5) CONSTRAINT pk_idrep PRIMARY KEY,
prenomrep  VARCHAR2(20),
nomrep     VARCHAR2(30),
adresse    VARCHAR2(20),
codepostal CHAR(5),
ville      VARCHAR2(30),
telephone  VARCHAR2(10)
);
```

### 1.1.4 Table Client

```
CREATE TABLE Client
(idclient   NUMBER(5) CONSTRAINT pk_idcli PRIMARY KEY,
raisonsociale VARCHAR2(20),
adresse     VARCHAR2(60),
codepostal  CHAR(5),
ville       VARCHAR2(20),
telephone   VARCHAR2(10)
idrep       NUMBER(5) CONSTRAINT fk_idrep REFERENCES
representant(idrep)
);
```

### 1.1.5 Table Article

```
CREATE TABLE Article
(idarticle  NUMBER(8) CONSTRAINT pk_idart PRIMARY KEY,
designation VARCHAR2(20),
prixunit   NUMBER(7,2),
qtestock   NUMBER(6)
);
```

### 1.1.6 Table Commande

```
CREATE TABLE Commande
(numcom      NUMBER(12) CONSTRAINT pk_num PRIMARY KEY,
idclient     NUMBER(5) CONSTRAINT fk_idcl REFERENCES client(idclient),
datecom      DATE
);
```

### 1.1.7 Table Ligne de commande

```
CREATE TABLE Ligne_com
(numcom      NUMBER(12) CONSTRAINT fk_num REFERENCES
    commande (numcom),
idarticle    NUMBER(8) CONSTRAINT fk_idar REFERENCES
    article(idarticle),
numlig       NUMBER(3),
qtecom       NUMBER(8),
CONSTRAINT pk_ligne_com PRIMARY KEY (numcom,idarticle)
);
```

Insérer les données suivantes dans les tables (@script.sql)

Utilisez le script fourni : script.sql.

## 2 LANGAGE DE MANIPULATION DE DONNEES

### I.1 Structure d'une table

Afficher la structure de la table client.

```
DESCRIBE Client;
```

### 2.1 Réponse 2

```
SELECT * FROM Client;
```

### 2.2 Réponse 3

```
SELECT raisonsociale FROM Client;
```

### 2.3 Réponse 5

```
SELECT raisonsociale, ville, codepostal  
FROM Client  
WHERE codepostal BETWEEN '60000' AND '95000';
```

Attention aux types de données. Ici, le code postal est une chaîne de caractères.

### 2.4 Réponse 6

```
SELECT idclient, raisonsociale  
FROM Client  
WHERE idclient=1 OR idclient= 5;
```

Ou

```
SELECT idclient, raisonsociale  
FROM Client  
WHERE idclient IN (1,5);
```

### 2.5 Réponse 7

```
SELECT raisonsociale FROM Client  
WHERE upper(raisonsociale) LIKE '_A%';
```

### 2.6 Réponse 8

```
SELECT raisonsociale, codepostal FROM Client  
WHERE codepostal LIKE '3%';
```

## 2.7 Réponse 9

```
SELECT raisonsociale FROM Client  
WHERE upper(raisonsociale) NOT LIKE 'C%';
```

## 2.8 Réponse 10

```
SELECT raisonsociale, adresse FROM Client  
WHERE adresse IS null;
```

## 2.9 Réponse 11

```
SELECT idclient, raisonsociale  
FROM Client  
ORDER BY raisonsociale DESC;
```

## 2.10 Réponse 12

```
SELECT Raisonsociale, ville FROM Client  
WHERE lower(ville) NOT LIKE 'paris';
```

## 2.11 Réponse 13

```
SELECT raisonsociale FROM Client  
WHERE upper(raisonsociale) LIKE '%RE%';
```

## 2.12 Réponse 14

```
SELECT numcom, datecom FROM Commande  
WHERE months_between(sysdate, datecom)>3;
```

## 2.13 Réponse 15

```
SELECT idrep, nomrep, prenomrep FROM Representant  
WHERE idrep >2  
ORDER BY idrep DESC;
```

## 2.14 Réponse 16

```
SELECT nomrep, prenomrep FROM Representant  
WHERE nomrep NOT LIKE 'G%';
```

## 2.15 Réponse 17

```
SELECT nomrep, prenomrep FROM Representant  
WHERE upper(substr(prenomrep,1,1)) BETWEEN 'D' AND 'M';
```



Attention !

La requête suivante est incorrecte

```
SELECT * FROM Representant
WHERE prenomrep BETWEEN 'D%' AND 'M%';
```

Les jokers % et \_ (souligné) n'ont de sens qu'avec l'opérateur LIKE !

## 2.16 Réponse 18

```
SELECT * FROM Representant
WHERE upper(substr(nomrep,1,1)) IN ('L','S');
```

## 2.17 Réponse 19

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

## 2.18 Réponse 20

```
SELECT nomrep, prenomrep
FROM Representant
WHERE upper(Substr(PrenomRep,1,1)) BETWEEN 'D' AND 'R'
INTERSECT
SELECT nomrep, prenomrep
FROM Representant
WHERE upper(Substr(NomRep,1,1)) IN ('L', 'S')
```

```
SELECT nomrep, prenomrep
FROM Representant
WHERE upper(substr(prenomrep,1,1)) BETWEEN 'D' AND 'R'
AND (upper(substr(nomrep,1,1)) IN ('L', 'S'))
```

## 2.19 Réponse 21

```
SELECT numcom, TO_CHAR(datecom, 'MM-YYYY')
FROM Commande;
```

## 2.20 Réponse 22

```
SELECT numcom, TO_CHAR(datecom, 'mon/yy')
FROM Commande
```

## 3 FONCTIONS PREDEFINIES

---

### 3.1 Réponse 1

```
SELECT TO_CHAR(sysdate, 'DD-MM-YYYY') FROM Dual;
```

### 3.2 Réponse 2

```
SELECT TO_CHAR(sysdate, 'HH24-MI-SS') FROM Dual;
```

### 3.3 Réponse 3

```
ALTER SESSION SET nls_date_format='dd-mm-yy';
```

## 4 JOINTURES

### 4.1 Réponse 1

```
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	1
CARREFOUR	3	2
CARREFOUR	3	5
CD PRO	2	4
DARTY	1	2
DARTY	1	6
DARTY	4	1
DARTY	4	6
DARTY	4	2
INFO SYSTEM	5	1
Multimedia	6	4
Multimedia	6	3
Multimedia	6	6

13 ligne(s) sélectionnée(s).

### 4.2 Réponse 2

```
SELECT c.raisonsociale,
      NVL(TO_CHAR(co.numcom), 'Pas de commande...')
      AS "Numéro de commande"
FROM Client c LEFT JOIN Commande co USING(idclient)
ORDER BY 1 ;
```

RAISONSOCIALE	Numéro de commande
CARREFOUR	3
CD PRO	2
DARTY	1
DARTY	4
INFO SYSTEM	5
INFO SYSTEM	7
Multimedia	6
THOMSON	Pas de commande...

Ou (et préférable)

```
SELECT c.raisonsociale,
      COALESCE(TO_CHAR(co.numcom), 'Pas de commande...')
      AS "Numéro de commande"
FROM Client c LEFT JOIN Commande co USING(idclient)
ORDER BY 1 ;
```

Ou encore

```
SELECT c.raisonsociale,
       LISTAGG (NVL (TO_CHAR(co.numcom), 'Pas de commande'), ' - ')
       WITHIN GROUP (ORDER BY co.numcom) "numéros de commande"
FROM Client c LEFT JOIN Commande co
ON (c.idclient=co.idclient)
GROUP BY c.raisonsociale;
```

et encore

```
SELECT raisonsociale,
       CASE
         WHEN numcom IS NOT NULL THEN 'N°de commande ' ||
         TO_CHAR(numcom)
         WHEN numcom IS NULL THEN 'Pas de commande '
         END "Statut de la commande"
FROM Client c LEFT JOIN Commande co USING (idclient)
```

### 4.3 Réponse 3

```
SELECT a.designation, l.numcom
FROM Article a LEFT JOIN Ligne_com l
ON a.idarticle=l.idarticle
WHERE l.idarticle IS 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
```

### 4.4 Réponse 4

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

RAISONSOCIALE	NOMREP
INFO SYSTEM	Sanchez
DARTY	Sanchez
Multimedia	Garcia
CD PRO	Garcia
CARREFOUR	Legrand

### 4.5 Réponse 5

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

## 4.6 Réponse 6

```
SELECT designation, datecom
FROM Commande JOIN Ligne_com
ON ligne_com.numcom=commande.numcom
JOIN Article
ON article.idarticle=ligne_com.idarticle
WHERE TO_CHAR(datecom,'MMYYYY') LIKE '022020' ;
```

## 4.7 Réponse 7

```
/* Darty passe la cde n° 7 */
INSERT INTO commande VALUES(7,1,'12/12/2020');
COMMIT ;

(SELECT raisonsociale, ville
FROM Client c, Commande co
WHERE c.idclient=co.idclient
AND to_char(datecom,'yy') = '20'
AND upper(ville)='PARIS')
INTERSECT
(SELECT raisonsociale, ville
FROM Client c, Commande co
WHERE c.idclient=co.idclient
AND to_char(datecom,'yy') = '21'
AND upper(ville)='PARIS') ;
```

## 4.8 Réponse 8

```
SELECT cl.idclient,raisonsociale
FROM Client cl
JOIN Commande co ON (cl.idclient=co.idclient)
WHERE to_char(datecom,'YYYY') = to_char(sysdate,'YYYY') ;
```

Avec la fonction EXTRACT :

```
SELECT cli.raisonsociale, cmd.datecom
FROM Client cli
JOIN Commande cmd ON cmd.idclient = cli.idclient
WHERE EXTRACT(year FROM cmd.datecom) = EXTRACT(year FROM sysdate);
```

## 4.9 Réponse 9

```
/*DARTY passe la cde n° 8*/
INSERT INTO Commande VALUES(8,1,'15/04/20');
COMMIT;

SELECT DISTINCT raisonsociale
FROM Client cl JOIN Commande co
ON cl.idclient=co.idclient
```

```
AND to_char(datecom,'yy')=20';
```

#### 4.10 Réponse 10

Liste des clients qui ont commandés les trois premiers mois de 2022.

```
SELECT cl.idclient,cl.raisonsociale , co.datecom  
FROM Client cl  
JOIN Commande co ON (cl.idclient=co.idclient)  
WHERE to_char(datecom,'YYYYMM') BETWEEN '202201' AND '202203';
```

#### 4.11 Réponse 11

```
SELECT raisonsociale,nomrep,  
       decode(cl.idrep,null,'pas de représentant','Le  
representant est '||nomrep) representation  
FROM Client cl LEFT JOIN Representant rep  
ON cl.idrep=rep.idrep;
```

## 5 FONCTIONS D'AGREGATS

### 5.1 Réponse 1

Afficher le Nombre de commande.

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

### 5.2 Réponse 2

Afficher le nombre de ligne par commande.

```
Col commande format A15
SELECT 'commande n°' || numcom AS "commande", COUNT(numcom)
FROM ligne_com
GROUP BY numcom ;
```

### 5.3 Réponse 3

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	RAISONSOCIALE	Nombre de commandes
2	INFO SYSTEM	1
1	DARTY	2
3	CD PRO	1
5	CARREFOUR	1
4	Multimedia	1

### 5.4 Réponse 4

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
5	Garcia	2
7	Legrand	1

4 Sanchez

2

ou

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

## 5.5 Réponse 5

Idem avec les clients qui n'ont pas de représentant

```
SELECT count (idclient)
FROM client
WHERE idrep is null
```

## 5.6 Réponse 6

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
Fontaine	0
lebeaux	0
Hoareau	0
Dupond	0

## 5.7 Réponse 7

Calcul du montant total de toutes les commandes.

```
SELECT SUM(prixunit*qtecom) AS "Chiffre d'affaire"
FROM article JOIN ligne_com
USING(idarticle) ;
```

## 5.8 Réponse 8

Calcul du montant de chaque commande.

```
SELECT l.numcom, sum(l.qtecom*a.prixunit) AS Montant
FROM article a NATURAL JOIN ligne_com l
GROUP BY l.numcom
ORDER BY Montant ;
```

## 5.9 Réponse 9

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 1;
```

## 5.10 Réponse 10

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;
```

## 5.11 Réponse 11

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 ;
OU
```

```
SELECT AVG(SUM(Prixunit*qtecom)) AS "Moyenne des montants"
FROM ligne_com NATURAL JOIN article
GROUP BY numcom ;
```

## 5.12 Réponse 12

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;
```

## 5.13 Réponse 13

Liste et total des commandes dont la somme dépasse 2000 €.

```
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)>2000;
```

### 5.14 Réponse 14

Liste des commandes qui ont au moins deux lignes (2 références d'articles).

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

### 5.15 Réponse 15

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

### 5.16 Réponse 17

```
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(numlig)>2;
```

### 5.17 Réponse 18

```
SELECT a.idarticle,a.designation,a.prixunit,l.numcom,co.datecom,
SUM(l.qtecom*a.prixunit) montant
FROM article a JOIN ligne_com l
ON a.idarticle=l.idarticle
JOIN commande co
ON l.numcom=co.numcom
WHERE a.prixunit>200
AND months_between(sysdate,co.datecom) >2
GROUP BY a.idarticle,a.designation,a.prixunit,l.numcom,co.datecom;
```

## 6 SOUS-REQUETES

### 6.1 Réponse 1

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

```
SELECT designation, prixunit
FROM Article
WHERE prixunit < (SELECT AVG(prixunit)
                  FROM Article)
```

### 6.2 Réponse 2

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

```
SELECT a.idarticle, a.designation, a.qtestock
FROM Article a
WHERE a.qtestock > (SELECT SUM(li.qtecom)
                   FROM Ligne_com li WHERE li.idarticle=
a.idarticle)
ORDER BY a.idarticle ;
```

Avec la quantité commandée :

```
SELECT a.designation, a.qtestock , COALESCE(SUM(lig1.qtecom), 0)
      "Quantité commandée",
      COALESCE(a.qtestock-SUM(lig1.qtecom), a.qtestock) "Reste"
FROM Article a LEFT JOIN Ligne_com lig1
      ON (a.idarticle=lig1.idarticle)
WHERE a.qtestock > (SELECT COALESCE(SUM(lig2.qtecom), 0)
                  FROM Ligne_com lig2
                  WHERE lig2.idarticle=a.idarticle)
GROUP BY a.designation, a.qtestock
ORDER BY a.designation ;
```

### 6.3 Réponse 3

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);
```

ou

```
SELECT idarticle, designation, qtestock
```

```
FROM Article
WHERE qtestock >= ALL (SELECT qtestock
                        FROM Article);
```

## 6.4 Réponse 5

Liste des articles qui n'ont pas été commandés

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

Ou

```
SELECT designation, idarticle
FROM Article art
WHERE NOT EXISTS
    (SELECT idarticle
     FROM Ligne_com lgn
     WHERE art.idarticle=lgn.idarticle)
;
```

## 6.5 Réponse 6

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)
ORDER BY l.numcom, l.idarticle ;
```

## 6.6 Réponse 7

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, NVL(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);
```

## 6.7 Réponse 8

```
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 ) ;
```

## 6.8 Réponse 9

```
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) ;
```

Et merci à Gauthier qui nous propose son interprétation de la question (attention, ce n'est pas la même interprétation...) :

```
SELECT c.numcom, listagg(a1.designation,' - ') WITHIN GROUP (ORDER
BY a1.designation)
AS "Articles",
LISTAGG(l1.qtecom, ' - ') WITHIN GROUP (ORDER BY a1.designation)
AS "Quantité commandée",
LISTAGG(c1.raisonsociale, ' - ') WITHIN GROUP (ORDER BY
a1.designation)
AS "Nom client",
LISTAGG(a1.prixunit, ' - ') WITHIN GROUP (ORDER BY a1.designation)
AS "Prixunit"
FROM Client c1 JOIN Commande c ON (c1.IDCLIENT=c.idclient)
JOIN Ligne_com l1 ON (l1.numcom=c.numcom)
JOIN Article a1 ON (a1.idarticle=l1.idarticle)
WHERE a1.prixunit <= ((SELECT avg(a2.prixunit)
                      FROM Article a2)
                     )
AND l1.qtecom = (SELECT MAX(l2.qtecom)
                FROM ligne_com l2
                WHERE l2.numcom=l1.numcom)
GROUP BY c.numcom
;
```

## 7 MODIFICATIONS DES TABLES

### 7.1 Réponse 1

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

Retour en arrière : OUI en faisant un ROLLBACK

### 7.2 Réponse 2

Majoration de 2% des articles dont le prix est inférieur à la moyenne.

```
UPDATE Article SET prixunit=prixunit+(prixunit*0.02)
WHERE prixunit<(SELECT AVG(prixunit)
                FROM Article);
3 ligne(s) mise(s) à jour.
```

### 7.3 Réponse 3

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

### 7.4 Réponse 4

```
UPDATE commande SET datecom=sysdate WHERE idclient=3;
UPDATE commande SET datecom=sysdate-4 WHERE idclient=5;
```

ou

```
UPDATE commande
SET datecom = sysdate + 2* (3 - idclient)
WHERE idclient IN (3,5);
```

ou

```
UPDATE commande
SET datecom = CASE idclient
                WHEN 3 then sysdate
                WHEN 5 then sysdate -4
                END
WHERE idclient in (3,5);
```

## 8 AUTRES COMMANDES LDD

### 8.1 Réponse 1

```
COMMENT ON TABLE Article IS 'Table des articles';  
SELECT comments FROM User_tab_comments WHERE Table_name = 'ARTICLE'
```

```
COMMENT ON COLUMN Article.idarticle IS 'identifiant d'un article';  
SELECT column_name,comments  
FROM User_col_comments  
WHERE table_name = 'ARTICLE';
```

### 8.2 Réponse 2

```
CREATE TABLE Client_Paris (idclient,  
raison sociale, adresse, idrep, telephone)  
AS SELECT idclient, raison sociale, adresse, idrep, telephone  
FROM Client  
WHERE upper(ville)='PARIS';
```

### 8.3 Réponse 3

```
RENAME Client_Paris TO Client_Par;
```

### 8.4 Réponse 4

```
CREATE TABLE Mail(nom, adresse, codeville)  
AS SELECT raison sociale, adresse, code postale || '-' || ville  
FROM Client;
```

### 8.5 Réponse 5

```
CREATE TABLE Commande_zero(nom, adresse, codeville)  
AS SELECT raison sociale, adresse, ' ' || code postale || ' ' || ville  
FROM client c  
WHERE NOT EXISTS (select null from commande co  
where co.idclient = c.idclient);
```

### 8.6 Réponse 6

```
ALTER SESSION SET nls_date_format='DD/MM/YY';  
UPDATE Commande SET datecom='30/12/19'  
where idclient=4;
```

## 8.7 Réponse 7

```
SAVEPOINT S1;  
DELETE FROM representant WHERE idrep=1;  
ROLLBACK TO S1;
```



## 9 VUES SUR LES TABLES

### 9.1 Réponse 1

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

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

### 9.2 Réponse 2

```
CREATE OR REPLACE VIEW Vue_art_com (code,libelle,qte)
AS SELECT
  a.idarticle,a.designation,sum(l.qtecom)
FROM article a,ligne_com l
WHERE 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).

### 9.3 Réponse 3

```
Create or replace view vue_lig
as select a.designation,a.prixunit,l.numcom,l.numlig,l.qtecom
from article a, ligne_com l
where a.idarticle=l.idarticle;
```

Un exemple de vue avec fonction analytique

```
CREATE OR REPLACE VIEW rapport
AS SELECT
  nvl(c.raisonsociale, 'GRAND TOTAL') Nom,
  nvl(to_char(co.numcom), 'Total') numcom,
  SUM(a.prixunit * l.qtecom) montant
FROM
  client c
  JOIN commande co ON ( c.idclient = co.idclient )
  JOIN ligne_com l ON ( co.numcom = l.numcom )
  JOIN article a ON ( l.idarticle = a.idarticle )
GROUP BY
```

```
ROLLUP (c.raisonsociale,  
        co.numcom) ;
```

# 10 SYNONYMES, SEQUENCES, INDEX, CLUSTERS

## 10.1 Réponse 1

```
CREATE SYNONYM com  
FOR commande;
```

## 10.2 Réponse 2

```
DROP SEQUENCE seq_cli;  
CREATE SEQUENCE seq_cli  
START WITH n;  
n étant l'identifiant du dernier client + 1  
INSERT INTO Client VALUES(seq_cli.nextval, 'ELESTA', 'Rue  
Palestro', 93100, 'Pantin', 0141710022, 1);
```

## 10.3 Réponse 3

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

## 10.4 Réponse 4

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

## 10.5 Réponse 5

Les étapes de la création d'un cluster à index doivent suivre l'ordre suivant :

Création du cluster :

```
Create Cluster Clus_com (numcom number);
```

Création d'un index sur le Cluster :

```
Create Index ind_clus On Cluster Clus_com
```

Création des tables du cluster :

```
Create Table commande (numcom number,  
                        idclient number,  
                        datecom date)  
Cluster Clus_com(numcom);  
Create Table ligne_com(numcom number,  
                       numlig number,  
                       idarticle number,  
                       qtecom number)  
Cluster Clus_com(numcom);
```

## 10.6 Réponse 6

Les étapes de la création d'un cluster à hachage sont les suivantes:

-Création du cluster

```
Create Cluster Clus_ligne (numcom number)
Tablespace nom_du_tablespace
Hashkeys 10;
```

-Création des tables du cluster