

UNIVERSITÉ DU QUÉBEC À MONTRÉAL

VENTES DE LA PÉPINIÈRE PLEIN DE FOIN

RAPPORT DE TP
PRÉSENTÉ À
ROBERT GODIN
DANS LE CADRE DU COURS
INF3180
GROUPE 31

PAR
MATHIEU TRUDEL-LAPIERRE
TRUM18068508
ALEX LEBLANC
LEBA00000000

OCTOBRE 2013

```

SQL>
SQL> alter session set nls_date_format = 'dd/mm/yyyy'
2 /

Session altered.

SQL>
SQL> — 1. Les numéros des clients (sans répétition) qui ont
      placé au moins une commande
SQL>
SQL> select distinct client.noclient
2      from client, commande
3      where client.noclient = commande.noclient
4 /

NOCLIENT
-----
        30
        40
        20
        10

SQL>
SQL> — 2. Le numéro et la description des articles dont le
      numéro est entre 20 et 80 (inclusivement) et le prix est 10.99
      ou 25.99
SQL>
SQL> select noArticle, description
2      from article
3      where noArticle BETWEEN 20 AND 80 AND (prixUnitaire = 10.99 OR
      prixUnitaire = 25.99)
4 /

NOARTICLE DESCRIPTION
-----
        40 Epinette bleue
        70 Herbe puce

SQL>
SQL> — 3. Le numéro et la description des articles dont la
      description débute par la lettre C ou contient la chaîne 'bl'
SQL>
SQL> select noArticle, description
2      from article where description like 'C%' or description like

```

```
'%b1%'
3 /
```

NOARTICLE DESCRIPTION

```
10 Cedre en boule
40 Epinette bleue
50 Chene
60 Erable argente
81 Catalpa
```

SQL>

SQL> — 4. *Le numéro et le nom des clients qui ont placé une commande le 9 juillet 2000*

SQL>

```
SQL> select noClient, nomClient from client natural join commande
       where dateCommande = '09-07-2000'
```

```
2 /
```

NOCIENT NOMCLIENT

```
30 Lin Bo
20 Dollard Tremblay
```

SQL>

SQL> — 5. *Les noms des clients, numéros de commande, date de commande et noArticle pour les articles livrés le 4 juin 2000 dont la quantité livrée est supérieure à 1*

SQL>

```
SQL> select nomclient, nocommande, datecommande, noarticle
       2 from client natural join commande natural join lignecommande
       natural join detaillivraison natural join livraison
       3 where quantitelivree >1 and datelivraison = '04-06-2000'
```

```
4 /
```

NOMCLIENT	NOCOMMANDE	DATECOMMAN	NOARTICLE
-----------	------------	------------	-----------

Luc Sansom	1	01/06/2000	10
Dollard Tremblay	2	02/06/2000	40

SQL>

SQL> — 6. *La liste des dates du mois de juin 2000 pour lesquelles il y a au moins une livraison ou une commande. Les résultats sont produits en une colonne nommée DateÉvénement.*

SQL>

```

SQL> select dateevenem as DateEvenement
2  from (
3  select datecommande dateevenem from commande where
   datecommande between '01-06-2000' and '30-06-2000'
4  union
5  select datelivraison dateevenem from livraison where
   datelivraison between '01-06-2000' and '30-06-2000'
6  )
7  /

```

DATEEVENEM

```

01/06/2000
02/06/2000
03/06/2000
04/06/2000
05/06/2000

```

SQL>

SQL> — 7. Les noArticle et la quantité totale livrée de l'article
incluant les articles dont la quantité totale livrée est égale
à 0.

SQL>

```

SQL> select noarticle, nvl(sum(quantitelivree),0) quantitelivree
2  from article natural left outer join detaillivraison
3  group by noarticle
4  order by noarticle
5  /

```

NOARTICLE QUANTITELIVREE

10	10
20	1
40	3
50	0
60	0
70	7
80	0
81	0
90	1
95	1

10 rows selected.

SQL>

SQL> — 8. Les noArticle et la quantité totale livrée de l'article

pour les articles dont le prix est inférieur à \$20 et dont la quantité totale livrée est inférieure à 5

SQL>

```
SQL> select noarticle, quantitelivree
2  from (
3  select noarticle, nvl(sum(quantitelivree),0) quantitelivree
4  from article natural left join detaillivraison
5  where prixunitaire <20 group by noarticle
6  )
7  where quantitelivree <5 or quantitelivree is null
8  /
```

NOARTICLE QUANTITELIVREE

20	1
60	0
95	1

SQL>

SQL> — 9. *Le noLivraison, noCommande, noArticle, la date de la commande, la quantité commandée, la date de la livraison, la quantité livrée et le nombre de jours écoulés entre la commande et la livraison dans le cas où ce nombre a dépassé 2 jours et le nombre de jours écoulés depuis la commande jusqu'à aujourd'hui est supérieur à 100*

SQL>

```
SQL> select nolivraison, nocommande, noarticle, datecommande,
quantite, datelivraison, quantitelivree, (datelivraison -
datecommande) nombrejoursecoule
2  from commande natural join lignecommande natural join
detaillivraison natural join livraison natural join livraison
3  where ( datelivraison - datecommande > 2 )
4  order by nolivraison
5  /
```

NOLIVRAISON NOCOMMANDE NOARTICLE DATECOMMAN QUANTITE DATELIVRAI

QUANTITELIVREE NOMBREJOURSECOULE

101	1	10	01/06/2000	10	04/06/2000
3		3			
103	1	90	01/06/2000	1	05/06/2000
1		4			

```
SQL>
SQL> — 10. La liste des Articles triée en ordre décroissant de
prix et pour chacun des prix en ordre croissant de numéro
```

```
SQL>
SQL> select *
2 from article
3 order by prixunitaire desc, noarticle asc
4 /
```

NOARTICLE	DESCRIPTION	PRIXUNITAIRE	QUANTITEENSTOCK
80	Poirier	26,99	10
40	Epinette bleue	25,99	10
81	Catalpa	25,99	10
90	Pommier	25,99	10
50	Chene	22,99	10
60	Erable argente	15,99	10
95	Genevrier	15,99	10
20	Sapin	12,99	10
10	Cedre en boule	10,99	10
70	Herbe puce	10,99	10

10 rows selected.

```
SQL>
SQL> — 11. Le nombre d'articles dont le prix est supérieur à 25 et
le nombre d'articles dont le prix est inférieur à 15 (en deux
colonnes)
```

```
SQL>
SQL> select nombrepluscherque25,nombremoinscherque15
2 from (select count(noarticle)nombrepluscherque25 from article
where prixunitaire >25), (select count( noarticle)
nombremoinscherque15 from article where prixunitaire <15)
3 /
```

NOMBREPLUSCHERQUE25	NOMBREMOISCHERQUE15
---------------------	---------------------

4

3

```
SQL>
SQL> — 12. Les noCommande des commandes qui n'ont aucune
livraison correspondante
```

```
SQL>
SQL> select nocommande
2 from commande natural left outer join detaillivraison
3 where nolivraison is null
```

4 /

NOCOMMANDE

8
6
7

SQL>

SQL> — 13. En deux colonnes, les paires de numéros de commandes (différentes) qui sont faites à la même date ainsi que la date de commande. Il faut éviter de répéter deux fois la même paire.

SQL>

SQL> **select distinct** least(c1.nocommande, c2.nocommande) **as**
nocommande, greatest(c1.nocommande, c2.nocommande) **as** nocommande
, c1.datecommande
2 **from** commande c1, commande c2
3 **where** c1.nocommande <> c2.nocommande
4 **and** c1.datecommande = c2.datecommande
5 /

NOCOMMANDE NOCOMMANDE DATECOMMAN

5 6 09/07/2000
2 3 02/06/2000
7 8 15/07/2000

SQL>

SQL> — 14. Le montant total commandé pour chaque paire (dateCommande, noArticle) dans les cas où le montant total dépasse 50\$.

SQL>

SQL> **select** commande.datecommande, lignecommande.noarticle, **sum**(
article.prixunitaire * lignecommande.quantite) **as** "Montant total
commande"
2 **from** commande, lignecommande, article
3 **where** commande.nocommande = lignecommande.nocommande **and**
4 lignecommande.noarticle = article.noarticle
5 **group by** commande.datecommande, lignecommande.noarticle
6 **having** **sum**(article.prixunitaire * lignecommande.quantite) >
50
7 /

DATECOMMAN NOARTICLE Montant total commande

01/06/2000 70 54,95

09/07/2000	20	64,95
01/06/2000	10	109,9
09/07/2000	10	109,9
02/06/2000	40	51,98

SQL>

SQL> — 15. Les noArticle des articles commandés dans toutes et
chacune des commandes du client 20

SQL>

```
SQL> select t.article as NOARTICLE
2      from
3      (select lignecommande.noarticle as article, count(*) as
total
4      from commande, lignecommande
5      where commande.nocommande = lignecommande.nocommande
6      and commande.noclient = 20
7      group by lignecommande.noarticle) t
8  where t.total = (
9      select count(distinct lignecommande.nocommande)
10     from commande, lignecommande
11     where commande.nocommande = lignecommande.nocommande
12     and commande.noclient = 20)
13  /
```

NOARTICLE

40

SQL>

SQL>

SQL> spool off