Thibault ROSA

TP1 BI

Exercice 1 .

a)

SQL> select DEPTNO,ENAME,SAL,RANK() OVER (PARTITION BY deptno ORDER BY SAL DESC)"RANK"from EMP where DEPTNO = 30 or DEPTNO = 10 ORDER BY DEPTNO;

```
    DEPTNO ENAME      SAL       RANK
---------- ---------- ---------- ----------
        10 KING             5000          1
        10 CLARK            2450          2
        10 MILLER           1300          3
        30 BLAKE            2850          1
        30 ALLEN            1600          2
        30 TURNER           1500          3
        30 MARTIN           1250          4
        30 WARD             1250          4
        30 JAMES             950          6

9 rows selected.
```

b)

SQL> select DEPTNO,ENAME,SAL,Dense_RANK() OVER (PARTITION BY deptno ORDER BY SAL DESC)"RANK"from EMP where DEPTNO = 30 or DEPTNO = 10 ORDER BY DEPTNO;

```
    DEPTNO ENAME      SAL       RANK
---------- ---------- ---------- ----------
        10 KING             5000          1
        10 CLARK            2450          2
        10 MILLER           1300          3
        30 BLAKE            2850          1
        30 ALLEN            1600          2
        30 TURNER           1500          3
        30 MARTIN           1250          4
        30 WARD             1250          4
        30 JAMES             950          5

9 rows selected.
```

c)

SQL> select distinct DEPTNO,SAL,Dense_RANK() OVER (PARTITION BY deptno ORDER BY SAL DESC)"RANK"from EMP where DEPTNO = 20 or DEPTNO = 10 ORDER BY DEPTNO;

```
   DEPTNO      SAL      RANK
---------- ---------- ----------
         10       5000          1
         10       2450          2
         10       1300          3
         20       3000          1
         20       2975          2
         20       1100          3
         20        800          4

7 rows selected.
```

d)

SQL> select JOB,sum(SAL)"TOT_SAL_JOB" from EMP GROUP BY JOB;

```
JOB       TOT_SAL_JOB
--------- -----------
CLERK                4150
SALESMAN   5600
PRESIDENT   5000
MANAGER    8275
ANALYST      6000
```

SQL> select distinct JOB,sum(SAL)OVER(PARTITION BY JOB)"TOT_SAL_JOB" from EMP ;

```
JOB       TOT_SAL_JOB
--------- -----------
MANAGER    8275
PRESIDENT   5000
SALESMAN   5600
CLERK                4150
ANALYST      6000
```

e)

PARTITION BY : permet d'organiser le jeu de résultats en groupes logiques, d'après les différentes valeurs de l'expression spécifiée. Cela correspond à peu de chose près à la clause GROUP BY lorsque vous utilisez des agrégats directement dans une requête.

En cas d'omission de cette clause, l'intégralité du jeu de résultats est considérée comme une partition.

Les éléments de la liste PARTITION BY sont interprétés à peu près de la même façon que les éléments de GROUP BY , sauf qu'ils sont toujours des expressions simples et jamais le nom ou le numéro d'une colonne en sortie. Une autre différence est que ces expressions peuvent contenir des appels à des fonctions d'aggrégat, ce qui n'est pas autorisé dans une clause GROUP BY classique. Ceci est autorisé ici parce que le windowing se produit après le regroupement et l'aggrégation.

f)

SQL> select distinct DEPTNO,JOB,sum(SAL) from EMP GROUP BY ROLLUP(DEPTNO,JOB) ORDER BY DEPTNO;

```
   DEPTNO JOB              SUM(SAL)
---------- --------- ----------
        10 CLERK         1300
        10 MANAGER             2450
        10 PRESIDENT    5000
        10                      8750
        20 ANALYST     6000
        20 CLERK         1900
        20 MANAGER             2975
        20                      10875
        30 CLERK          950
        30 MANAGER             2850
        30 SALESMAN            5600

   DEPTNO JOB              SUM(SAL)
---------- --------- ----------
        30                     9400
                              29025
```

g)

SQL>select distinct DEPTNO,decode(grouping(JOB),1,'TousEmployes',to_char(JOB)) JOB#,sum(SAL) from EMP GROUP BY ROLLUP(DEPTNO,JOB) ORDER BY DEPTNO;

```
   DEPTNO JOB#        SUM(SAL)
---------- ------------ ----------
        10 CLERK           1300
        10 MANAGER               2450
        10 PRESIDENT             5000
        10 TousEmployes       8750
        20 ANALYST       6000
        20 CLERK           1900
        20 MANAGER               2975
        20 TousEmployes      10875
        30 CLERK            950
        30 MANAGER               2850
```

```
        30 SALESMAN              5600

   DEPTNO JOB#       SUM(SAL)
---------- ----------- ----------
        30 TousEmployes       9400
           TousEmployes      29025
```

SQL> select distinct DEPTNO,case when JOB is null then 'TousEmployes' else JOB end JOB,sum(SAL) from EMP GROUP BY ROLLUP(DEPTNO,JOB) ORDER BY DEPTNO;

```
   DEPTNO JOB                 SUM(SAL)
---------- ----------- ----------
        10 CLERK           1300
        10 MANAGER                2450
        10 PRESIDENT              5000
        10 TousEmployes       8750
        20 ANALYST       6000
        20 CLERK           1900
        20 MANAGER                2975
        20 TousEmployes      10875
        30 CLERK            950
        30 MANAGER                2850
        30 SALESMAN               5600

   DEPTNO JOB                 SUM(SAL)
---------- ----------- ----------
        30 TousEmployes       9400
           TousEmployes      29025
```

Exercice 2

1.)

```
select
TEMPS.annee,CLIENTS.CL_R,PRODUITS.CATEGORY,AVG(VENTES.QTE*VENTES.PU) as
CA_MOY
from TEMPS, CLIENTS, PRODUITS, VENTES
WHERE VENTES.CID=CLIENTS.CL_ID
and VENTES.TID=TEMPS.TID
and PRODUITS.PID=VENTES.PID
and TEMPS.annee < 2011
and TEMPS.annee > 2008
GROUP BY ROLLUP(TEMPS.annee,CLIENTS.CL_R,PRODUITS.CATEGORY);
```

671 rows selected.

2.)

```
select
TEMPS.annee,CLIENTS.CL_R,PRODUITS.CATEGORY,AVG(VENTES.QTE*VENTES.PU) as
CA_MOY
from TEMPS, CLIENTS, PRODUITS, VENTES
WHERE VENTES.CID=CLIENTS.CL_ID
and VENTES.TID=TEMPS.TID
and PRODUITS.PID=VENTES.PID
and TEMPS.annee < 2011
and TEMPS.annee > 2008
GROUP By cube(TEMPS.annee,CLIENTS.CL_R,PRODUITS.CATEGORY);
```

1155 rows selected.

3.)

```
select ANNEE, CATEGORY, PNAME FROM ( select TEMPS.annee, PRODUITS.category,
PRODUITS.pname,
  rank () over (partition by TEMPS.annee, PRODUITS.category order by sum(VENTES.pu *
VENTES.qte) desc) as rang
  from TEMPS, VENTES, PRODUITS
  where PRODUITS.pid = VENTES.pid and TEMPS.tid = VENTES.tid
  group by grouping sets ((TEMPS.annee, PRODUITS.category, PRODUITS.pname)))
          where rang = 1;
```

4.)

```
select TEMPS.annee, PRODUITS.category, sum(VENTES.qte * VENTES.pu)
from TEMPS, PRODUITS, VENTES
where PRODUITS.pid = VENTES.pid and TEMPS.tid = VENTES.tid
having GROUPING_ID(TEMPS.annee, PRODUITS.category) != 3
group by rollup (TEMPS.annee, PRODUITS.category);
```

5.)

```
select ANNEE, MOIS, CA_TOTAL FROM (select TEMPS.annee,TEMPS.mois, sum(VENTES.pu
* VENTES.qte) as CA_TOTAL, rank () over (partition by TEMPS.annee order by
sum(VENTES.pu * VENTES.qte) desc) as rang from TEMPS, VENTES
where TEMPS.tid = VENTES.tid
and VENTES.pid= 61
group by grouping sets ((TEMPS.annee, TEMPS.mois)))
where rang = 1;
```

6.)

```
select TEMPS.annee, CLIENTS.cl_name, PRODUITS.category, sum(VENTES.pu * VENTES.qte)
as CA_TOTAL
from TEMPS, CLIENTS, PRODUITS, VENTES
where TEMPS.tid = VENTES.tid and PRODUITS.pid = VENTES.pid and CLIENTS.cl_id =
VENTES.cid
group by grouping sets ((TEMPS.annee,CLIENTS.cl_name),
(TEMPS.annee,PRODUITS.category));
```

7.)

```
select PRODUITS.category, sum(VENTES.qte) as QTE_VENDUE_2010, ntile(3) over (order by
sum(VENTES.qte) desc) as TIERS
from PRODUITS, TEMPS, VENTES
where PRODUITS.pid = VENTES.pid and TEMPS.tid = VENTES.tid and TEMPS.annee = 2010
group by PRODUITS.category;
```

8.)

```
select category,mois,sum(qte) as QTE_5_JOURS
from (select PRODUITS.category,TEMPS.annee,TEMPS.mois, VENTES.qte,Dense_Rank() OVER
(PARTITION BY TEMPS.annee,TEMPS.mois  order by TEMPS.mois, TEMPS.jour ) as jour
                from PRODUITS, TEMPS, VENTES
                where PRODUITS.pid = VENTES.pid
                and TEMPS.tid = VENTES.tid
                and TEMPS.annee = 2010
                group by
                PRODUITS.category,TEMPS.annee,TEMPS.mois,TEMPS.jour,VENTES.qte
                order by TEMPS.mois,TEMPS.jour
                )
        where jour < 6
        group by category,mois
        order by mois;
```

Dans la requête interne je prends tout les jours de chaques mois de l'année 2010 ou il y a eu une vente en les numérotant par jour de ventes dans le mois par catégorie.

Soit si en janvier le premier jours où il y a une vente est le 20 du mois, il sera numéroté 1
Si il y a deux ventes ce premier jours le deux première serons numéroté 1.
Si le prochain jours avec vente et le 26, il sera noté 2. D'ou le DENSE_RANK() ;

De la je fait un autre requette par dessus où je fais la somme de la quantité des produit vendu le cinq premier jour avec ventes, ceci groupé par catégorie et par mois.

Mais … j'ai pas le résultat du tp.