
7. Les fonctions de groupe

- ORACLE dispose de fonctions appelées fonctions de groupe ou fonctions agrégats qui s'appliquent à un ensemble de données :
 - ***AVG*** (*[distinct] col*) : Moyenne arithmétique des valeurs non nulles d'une colonne.
 - ***COUNT***(*) : Nombre de lignes.
 - ***COUNT***(*[distinct] col*) : Nombre de valeurs non nulles de la colonne.
 - ***MAX*** (*[distinct] col*) : Valeur maximale non nulle de la colonne.
 - ***MIN*** (*[distinct] col*) : Valeur minimale non nulle de la colonne.
 - ***SUM*** (*[distinct] col*) : Somme des valeurs non nulle de la colonne.

7. Les fonctions de groupe

Remarques : Ces fonctions

- peuvent être appliquées sur une expression arithmétique.
- ignorent les valeurs nulles
- prennent par défaut les valeurs multiples. Pour ne prendre que les valeurs distinctes, il faut ajouter l'opérateur ***DISTINCT***.
- ***ne peuvent pas être utilisées dans la clause WHERE.***

Exemples

- Donner le nombre total d'ordinateurs

```
SELECT Count(*) "Nombre"  
FROM Ordinateur;
```

- Afficher le nombre total d'ordinateurs I3

```
SELECT Count(*) "Nombre"  
FROM Ordinateur  
WHERE pro = 'I3';
```

- Afficher la quantité moyenne en stock des ordinateurs de couleur Rouge

```
SELECT avg(qtestk) "quantité"  
FROM ordinateur  
Where Couleur = 'R';
```

- Afficher les prix minimums et maximums des ordinateurs I3

```
SELECT max(prix) "maximum", min(prix) "minimum"  
FROM ordinateur  
WHERE pro = 'I3';
```

Exemples

- Donner le nombre d'ordinateurs avec offre

```
SELECT Count(*) "Nombre"  
FROM Ordinateur  
WHERE offre is not null;
```

- Ou bien

```
SELECT Count(offre) "Nombre"  
FROM Ordinateur;
```

7. Les fonctions de groupe : classification de résultat

- Il est possible de regrouper (classifier, diviser) les données d'une table en groupes.
- Objectif : calculer un résultat pour chaque groupe.
- **Syntaxe**

```
SELECT  
FROM  
WHERE  
GROUP BY expression  
HAVING condition
```

Permet de définir le critère de classement

Permet de définir une condition sur les groupes

Exemples

- Afficher la quantité en stock des ordinateurs selon le type du processeur

```
SELECT    pro, sum(Qtestk) "Qte totale"  
FROM      Ordinateur  
GROUP BY pro;
```

DC	5
I3	7
I5	10

- Afficher les prix min et max des ordinateurs selon le type du processeur

```
SELECT pro, min(prix) "P MIN", max(prix) "P MAX"  
FROM    Ordinateur  
GROUP BY pro;
```

Exemples

- Afficher la quantité en stock des ordinateurs par couleur et type du processeur

```
SELECT    pro, couleur, sum(Qtestk) "Qte totale"  
FROM      Ordinateur  
GROUP BY  pro, couleur;
```

Remarque : avec la **clause GROUP BY**, on ne peut mettre dans la **clause SELECT** qu'une caractéristique du groupe : ***critère de groupement et/ou fonction de groupe***

7. Les fonctions de groupe : classification de résultat

- Pour définir une condition portant sur un groupe et ayant une fonction de groupe comme prédicat, il faut utiliser la clause HAVING.
- Afficher la quantité en stock, ≥ 50 , des ordinateurs selon le type du processeur

```
SELECT    pro, sum(Qtestk) "Qte totale"  
FROM      Ordinateur  
GROUP BY  pro  
HAVING    sum(Qtestk) >= 50 ;
```

- Afficher les marques ayant un prix moyen < 1000

```
SELECT    marque, avg(prix) "Prix Moy"  
FROM      Ordinateur  
GROUP BY  marque  
HAVING    avg(prix) < 1000 ;
```

7. Les fonctions de groupe : classification de résultat

- Afficher adresses des magasins ayant au moins deux types d'ordinateurs I3

```
SELECT    adresse
FROM      Magasin, Ordinateur
WHERE     mag = nummag
and       pro = 'I3'
GROUP BY  adresse
HAVING    count(*) >= 2 ;
```

- Afficher marques ayant un prix moyen < au prix moyen total

```
SELECT    marque, avg(prix)
FROM      Ordinateur
GROUP BY  marque
HAVING    avg(prix) < (SELECT avg(prix)
                        FROM      Ordinateur);
```

TD 2

1.1)

```
select * from dept;
```

1.2)

```
select ename,sal from emp;
```

1.3)

```
select distinct job from emp;
```

1.4)

```
select hiredate  
from emp  
where job='ANALYST';
```

TD 2

2.1)

```
select ename,dname      from emp e,dept d
where e.deptno=d.deptno;
```

2.2)

```
select ename from emp
where (job,hiredate)in(select job,hiredate from emp
                        where upper(ename) = 'FORD' )
and ename<>'FORD' ;
```

2.3)

```
select ename      from emp
where mgr=(select mgr from emp
            where upper(ename) = 'CLARK' )
and upper(ename)<>'CLARK' ;
```

TD 2

2.4)

```
select ename      from emp
where mgr=(select empno from emp
           Where job='PRESIDENT' );
```

2.5)

```
select e.ename,e.sal,c.ename,c.sal
from emp e,emp c
where e.sal>c.sal          and e.mgr=c.empno;
```

2.6)

```
select ename      from emp e,dept d
where e.deptno=d.deptno          and dname='RESEARCH'
and sal >=(select losal from salgrade where grade=4)
and sal <=(select hisal from salgrade where grade=4)
;
```

TD 2

2.7)

```
select distinct e.ename from emp e,emp s
where e.empno=s.mgr;
```

2.8)

```
select ename
from emp e,dept d
where e.deptno=d.deptno
and dname='RESEARCH'
and hiredate in (select hiredate
                  from emp a,dept b
                  where a.deptno=b.deptno
                  and dname='SALES' ) ;
```

TD 2

3.1)

```
select deptno from dept  
minus  
select deptno from emp;
```

3.2)

```
select distinct empno,ename,job  
from emp  
minus  
select distinct e1.empno,e1.ename,e1.job  
from emp e1,emp e2  
where e1.empno=e2.mgr;
```

TD 2

4.1)

```
select count(distinct deptno)          from emp;
```

4.2)

```
select deptno, count(*) nb_employe      from emp
group by deptno;
```

4.3)

```
select deptno, sum(sal) total_salaire    from emp
group by deptno;
```

4.4)

```
select deptno, avg(sal) salaire_moyen
from emp
group by deptno;
```

TD 2

4.5)

```
select ename,job,sal,grade
from emp,salgrade
Where sal >=(select losal from salgrade)
and sal <=(select hisal from salgrade)
order by grade,job;
```

4.6)

```
select e.ename,e.hiredate,c.ename,c.hiredate
from emp e,emp c
where e.hiredate<c.hiredate
and e.mgr=c.empno
order by e.hiredate;
```

TD 2

4.7)

```
select ename, sal+nvl(comm,0)          from emp
where sal+nvl(comm,0)>all(select sal+nvl(comm,0)
                        from emp
                        where deptno=30)
and deptno!=30;
```

4.8)

```
select dname,loc                      from dept d,emp e
where d.deptno=e.empno
group by dname,loc
having count(e.deptno)>3
and avg(sal)>=(select min(sal)*0.1    from emp);
```

TD 2

4.9)

```
select deptno      from emp
group by deptno
having count(distinct job) in (select count(distinct
                                job)      from emp)
```

4.10)

```
select avg(sal+nvl(comm,0)) "salaire moyen"
from emp
```

4.11)

```
select job,count(*)"nb_employe"      from emp
group by (job)
order by job,2 desc;
```

TD 2

4.12)

```
select job          from emp
group by job
having avg(sal)<=all(select avg(sal)
                    from emp group by job)
```

4.13)

```
select empno,deptno          from emp
where deptno=(select deptno
              from emp
              group by deptno
              having min(sal)=(select min(sal)
                              from emp )
              );
```

TD 2

4.14)

```
select empno,sal          from emp
where sal>(select avg(sal)
           from emp);
```

4.15)

```
select deptno,dname
from dept
where deptno in(select deptno
                from emp
                where upper(job)='ANALYST'
                group by deptno
                having count(*)>=2);
```