



Agréger des données à l'aide de fonctions de groupe

Objectifs

A la fin de ce chapitre, vous pourrez :

- **identifier les fonctions de groupe disponibles**
- **expliquer l'utilisation des fonctions de groupe**
- **regrouper des données à l'aide de la clause GROUP BY**
- **inclure ou exclure des groupes de lignes à l'aide de la clause HAVING**

Définition des fonctions de groupe

Les fonctions de groupe agissent sur des groupes de lignes et donnent un résultat par groupe.

EMPLOYEES

DEPARTMENT_ID	SALARY
90	24000
90	17000
90	17000
60	9000
60	6000
60	4200
50	5800
50	3500
50	3100
50	2600
50	2500
80	10500
80	11000
80	8600
	7000
10	4400

...

20 rows selected.

**Salaire
maximum dans
la table
EMPLOYEES.**

MAX(SALARY)
24000

Types de fonction de groupe

- **AVG**
- **COUNT**
- **MAX**
- **MIN**
- **STDDEV**
- **SUM**
- **VARIANCE**

Syntaxe des fonctions de groupe

```
SELECT      [column,] group_function(column), ...  
FROM        table  
[WHERE      condition]  
[GROUP BY   column]  
[ORDER BY   column];
```

Utiliser les fonctions AVG et SUM

Les fonctions AVG et SUM s'utilisent avec des données numériques.

```
SELECT AVG(salary), MAX(salary),  
       MIN(salary), SUM(salary)  
FROM   employees  
WHERE  job_id LIKE '%REP%';
```

AVG(SALARY)	MAX(SALARY)	MIN(SALARY)	SUM(SALARY)
8150	11000	6000	32600

Utiliser les fonctions MIN et MAX

Les fonctions MIN et MAX s'utilisent avec tous les types de données.

```
SELECT MIN(hire_date), MAX(hire_date)
FROM   employees;
```

MIN(HIRE_	MAX(HIRE_
17-JUN-87	29-JAN-00

Utiliser la fonction COUNT

La fonction COUNT (*) renvoie le nombre de lignes d'une table

```
SELECT COUNT ( * )  
FROM   employees  
WHERE  department_id = 50;
```

COUNT(*)
5

Utiliser la fonction COUNT

- La fonction **COUNT(*expr*)** renvoie le nombre de lignes contenant des valeurs non NULL dans la colonne *expr*.
- Affichez le nombre de valeurs contenues dans la colonne **DEPARTMENT_ID** de la table **EMPLOYEES**, à l'exception des valeurs NULL.

```
SELECT COUNT(commission_pct)
FROM   employees
WHERE  department_id = 80;
```

COUNT(COMMISSION_PCT)

3

Utiliser le mot-clé DISTINCT

- La fonction **COUNT(DISTINCT expr)** renvoie le nombre de valeurs non NULL distinctes de la colonne *expr*.
- Affichez le nombre de services distincts contenus dans la table EMPLOYEES.

```
SELECT COUNT(DISTINCT department_id)  
FROM employees;
```

COUNT(DISTINCTDEPARTMENT_ID)
7

Fonctions de groupe et valeurs NULL

Les fonctions de groupe ignorent les valeurs NULL des colonnes.

```
SELECT AVG(commission_pct)  
FROM employees;
```

AVG(COMMISSION_PCT)
.2125

Utiliser la fonction NVL avec les fonctions de groupe

La fonction NVL contraint les fonctions de groupe à intégrer des valeurs NULL.

```
SELECT AVG(NVL(commission_pct, 0))  
FROM   employees;
```

AVG(NVL(COMMISSION_PCT,0))

.0425

Créer des groupes de données

EMPLOYEES

DEPARTMENT_ID	SALARY
10	4400
20	13000
20	6000
50	5800
50	3500
50	3100
50	2500
50	2600
60	9000
60	6000
60	4200
80	10500
80	8600
80	11000
90	24000
90	17000

...

20 rows selected.

4400

9500

3500

6400

10033

Salaire
moyen
par
service
dans
la
table
EMPLOYEES.

DEPARTMENT_ID	AVG(SALARY)
10	4400
20	9500
50	3500
60	6400
80	10033.3333
90	19333.3333
110	10150
	7000

Créer des groupes de données : syntaxe de la clause GROUP BY

```
SELECT      column, group_function(column)
FROM        table
[WHERE      condition]
[GROUP BY   group_by_expression]
[ORDER BY   column];
```

La clause GROUP BY permet d'organiser les lignes d'une table en groupes restreints.

Utiliser la clause GROUP BY

La clause **GROUP BY** doit inclure toutes les colonnes de la liste **SELECT** qui ne figurent pas dans des fonctions de groupe.

```
SELECT    department_id, AVG(salary)
FROM      employees
GROUP BY  department_id ;
```

DEPARTMENT_ID	AVG(SALARY)
10	4400
20	9500
50	3500
60	6400
80	10033.3333
90	19333.3333
110	10150
	7000

8 rows selected.

Utiliser la clause GROUP BY

La colonne GROUP BY ne doit pas nécessairement figurer dans la liste SELECT.

```
SELECT    AVG(salary)
FROM      employees
GROUP BY  department_id ;
```

AVG(SALARY)	
	4400
	9500
	3500
	6400
	10033.3333
	19333.3333
	10150
	7000

Créer des sous-groupes

EMPLOYEES

DEPARTMENT_ID	JOB_ID	SALARY
90	AD_PRES	24000
90	AD_VP	17000
90	AD_VP	17000
60	IT_PROG	9000
60	IT_PROG	6000
60	IT_PROG	4200
50	ST_MAN	5800
50	ST_CLERK	3500
50	ST_CLERK	3100
50	ST_CLERK	2600
50	ST_CLERK	2500
80	SA_MAN	10500
80	SA_REP	11000
80	SA_REP	8600
...		
20	MK_REP	6000
110	AC_MGR	12000
110	AC_ACCOUNT	8300

20 rows selected.

Dans la table
EMPLOYEES,
calcul du total
des salaires
pour chaque
poste,
au sein de
chaque service.

DEPARTMENT_ID	JOB_ID	SUM(SALARY)
10	AD_ASST	4400
20	MK_MAN	13000
20	MK_REP	6000
50	ST_CLERK	11700
50	ST_MAN	5800
60	IT_PROG	19200
80	SA_MAN	10500
80	SA_REP	19600
90	AD_PRES	24000
90	AD_VP	34000
110	AC_ACCOUNT	8300
110	AC_MGR	12000
	SA_REP	7000

13 rows selected.

Utiliser la clause GROUP BY sur plusieurs colonnes

```
SELECT    department_id dept_id, job_id, SUM(salary)
FROM      employees
GROUP BY  department_id, job_id ;
```

DEPT_ID	JOB_ID	SUM(SALARY)
10	AD_ASST	4400
20	MK_MAN	13000
20	MK_REP	6000
50	ST_CLERK	11700
50	ST_MAN	5800
60	IT_PROG	19200
80	SA_MAN	10500
80	SA_REP	19600
90	AD_PRES	24000
90	AD_VP	34000
110	AC_ACCOUNT	8300
110	AC_MGR	12000
	SA_REP	7000

13 rows selected.

Erreurs d'utilisation des fonctions de groupe dans une interrogation

Toute colonne ou expression de la liste SELECT autre qu'une fonction d'agrégation doit être incluse dans la clause GROUP BY.

```
SELECT department_id, COUNT(last_name)
FROM employees;
```

```
SELECT department_id, COUNT(last_name)
      *
ERROR at line 1:
ORA-00937: not a single-group group function
```

Colonne manquante dans la clause GROUP BY

Erreurs d'utilisation des fonctions de groupe dans une interrogation

- Vous ne pouvez pas utiliser la clause WHERE pour limiter les groupes.
- Utilisez la clause HAVING.
- Vous ne pouvez pas utiliser de fonctions de groupe dans la clause WHERE.

```
SELECT    department_id, AVG(salary)
FROM      employees
WHERE     AVG(salary) > 8000
GROUP BY department_id;
```

```
WHERE     AVG(salary) > 8000
          *
```

ERROR at line 3:

ORA-00934: group function is not allowed here

N'utilisez pas la clause WHERE pour limiter les groupes

ORACLE

Exclure des groupes de résultats

EMPLOYEES

DEPARTMENT_ID	SALARY
90	24000
90	17000
90	17000
60	9000
60	6000
60	4200
50	5800
50	3500
50	3100
50	2600
50	2500
80	10500
80	11000
80	8600
...	...
20	6000
110	12000
110	8300

20 rows selected.

**Salaire maximum
par service,
à condition
qu'il soit supérieur
à 10 000 \$**

DEPARTMENT_ID	MAX(SALARY)
20	13000
80	11000
90	24000
110	12000

Exclure des groupes de résultats : clause HAVING

Utilisez la clause HAVING pour restreindre les groupes.

1. Les lignes sont regroupées.
2. La fonction de groupe est appliquée.
3. Les groupes qui correspondent à la clause HAVING s'affichent.

```
SELECT      column, group_function
FROM        table
[WHERE      condition]
[GROUP BY   group_by_expression]
[HAVING     group_condition]
[ORDER BY   column];
```

Utiliser la clause HAVING

```
SELECT    department_id, MAX(salary)
FROM      employees
GROUP BY  department_id
HAVING    MAX(salary)>10000 ;
```

DEPARTMENT_ID	MAX(SALARY)
20	13000
80	11000
90	24000
110	12000

Utiliser la clause HAVING

```
SELECT    job_id, SUM(salary) PAYROLL
FROM      employees
WHERE     job_id NOT LIKE '%REP%'
GROUP BY  job_id
HAVING    SUM(salary) > 13000
ORDER BY  SUM(salary);
```

JOB_ID	PAYROLL
IT_PROG	19200
AD_PRES	24000
AD_VP	34000

Imbriquer des fonctions de groupe

Affichez le salaire moyen maximum.

```
SELECT MAX(AVG(salary))  
FROM employees  
GROUP BY department_id;
```

MAX(AVG(SALARY))
19333.3333

Synthèse

Ce chapitre vous a permis d'apprendre à :

- utiliser les fonctions de groupe COUNT, MAX, MIN, AVG
- écrire des instructions contenant la clause GROUP BY
- écrire des intructions contenant la clause HAVING

```
SELECT      column, group_function(column)
FROM        table
[WHERE      condition]
[GROUP BY   group_by_expression]
[HAVING     group_condition]
[ORDER BY   column];
```

Présentation de l'exercice 5

Dans cet exercice, vous allez :

- **écrire des instructions contenant des fonctions de groupe**
- **grouper des lignes pour obtenir plusieurs résultats**
- **exclure des groupes en utilisant la clause HAVING**