UNIVERSITÉ POLYTECHNIQUE DE BINGERVILLE

Cours base de données

CHAPITRE 5 : SQL

Dr Coulibaly Tiekoura

PLAN DU CHAPITRE

1. Extraire des données à l'aide de l'instruction SQL SELECT
2. Restreindre et trier les données
3. Utiliser des fonctions monolignes afin de personnaliser la sortie
4. Afficher des données agrégées à l'aide des fonctions de groupe
5. Afficher des données de plusieurs tables
6. Utiliser des sous –interrogations
7. Utiliser des opérateurs ensemblistes

PLAN DU CHAPITRE (SUITE)

8. Manipuler des données
9. Utiliser des instructions LDD pour créer et gérer des tables

Cours base de données

CHAPITRE 5.4:

AFFICHER DES DONNÉES AGRÉGÉES À L'AIDE DES FONCTIONS DE GROUPE

OBJECTIFS

- Identifier les fonctions de groupe disponible.
- Décrire l'utilisation des fonctions de groupe.
- Regrouper des données à l'aide de la clause GROUP BY.
- Inclure ou exclure des lignes regroupées à l'aide de la clause HAVING.

QUE SONT LES FONCTIONS DE GROUPE?

- Les fonctions de groupe opèrent sur des ensembles de lignes afin de renvoyer un seul 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

...

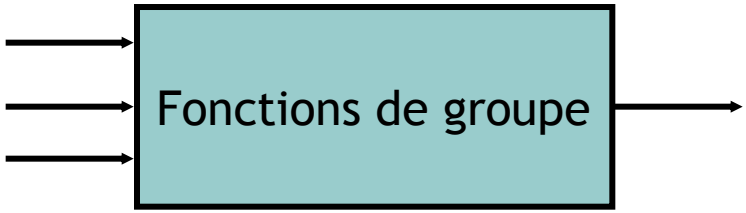
20 rows selected.

Salaire maximum de la table EMPLOYEES

MAX(SALARY)
24000

TYPES DE FONCTION DE GROUPE

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



FONCTIONS DE GROUPE : SYNTAXE

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

UTILISER LES FONCTIONS AVG ET SUM

- Vous pouvez utiliser les fonctions AVG et SUM pour les 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

- Vous pouvez utiliser MIN et MAX pour les valeurs numériques, les valeurs de type caractère et les valeurs de type date.

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

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

UTILISER LA FONCTION COUNT

- `COUNT(*)` renvoie le nombre de lignes d'une table:

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

COUNT(*)
5

- `COUNT(expr)` renvoie le nombre de lignes avec des valeurs non NULL pour expr:`r`

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

COUNT(COMMISSION_PCT)
3

UTILISER LE MOT CLÉ DISTINCT

- COUNT(DISTINCT expr) renvoie le nombre de valeurs non NULL distinctes de expr.
- Pour afficher le nombre de départements distincts de 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 de la colonne :

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

AVG(COMMISSION_PCT)
.2125

- La fonction NVL force les fonctions de groupe à inclure les 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
De la table
EMPLOYEES
Pour chaque
département

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];
```

- Vous pouvez diviser les lignes d'une table en groupes plus petits à l'aide de la clause GROUP BY.

UTILISER LA CLAUSE GROUP BY

- Toutes les colonnes de la liste SELECT qui ne sont pas incluses dans des fonctions de groupe doivent figurer dans la clause GROUP BY:

```
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.

REGROUPER EN FONCTION DE PLUSIEURS COLONNES

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.

Addionner les Salaires de la Table EMPLOYEES pour chaque Poste, regroupés Par département

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.

INTERROGATIONS ILLÉGALES AVEC DES FONCTIONS DE GROUPE

- Toute colonne ou expression de la liste SELECT qui ne constitue pas une fonction d'agrégation doit figurer 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`

INTERROGATIONS ILLÉGALES AVEC DES FONCTIONS DE GROUPE

- Vous ne pouvez pas utiliser des 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
```

Impossible d'utiliser la clause WHERE pour restreindre des groupes

RESTREINDRE LES RÉSULTATS DES GROUPES

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.

Le salaire Maximum par Département Lorsqu'il est Supérieur à 10 000$

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

RESTREINDRE LES RÉSULTATS DES GROUPES À L'AIDE DE LA CLAUSE HAVING

- Lorsque vous utilisez la clause HAVING, le serveur Oracle restreint les groupes de la façon suivante:
  - Les lignes sont regroupées
  - La fonction de groupe est appliquée
  - Les groupes qui correspondent à la clause HAVING s'affiche.

```
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

- Afficher le salaire moyen maximal:

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

MAX(AVG(SALARY))
19333.3333

SYNTHÈSE

- Ce chapitre vous a permis d'apprendre :
  - Utiliser les fonctions de groupe COUNT, MAX, MIN et AVG.
  - Ecrire des interrogations qui utilisent la clause GROUP BY.
  - Ecrire des interrogations qui utilisent la clause HAVING.

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