

# 5

## Agrupando datos

# Objetivos

**Tras completar esta lección seremos capaces de hacer lo siguiente:**

- **Identificar las funciones de agrupamiento disponibles.**
- **Describir el uso de las funciones de agrupamiento.**
- **Agupar datos usando la cláusula GROUP BY.**
- **Incluir o excluir grupos de filas mediante el uso de la cláusula HAVING .**

# Funciones de agrupamiento

Operan en conjuntos de filas para dar un resultado por grupo.

EMP

DEPTNO	SAL
10	2450
10	5000
10	1300
20	800
20	1100
20	3000
20	3000
20	2975
30	1600
30	2850
30	1250
30	950
30	1500
30	1250

“Máximo salario de  
la tabla EMP”

MAX (SAL)

5000

# Tipos de funciones de grupo

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

# Uso de funciones de grupo

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

# Funciones AVG y SUM

Podemos usar AVG y SUM para datos numéricos.

```
SQL> SELECT  AVG(sal), MAX(sal),  
2           MIN(sal), SUM(sal)  
3 FROM      emp  
4 WHERE     job LIKE 'SALES%';
```

AVG (SAL)	MAX (SAL)	MIN (SAL)	SUM (SAL)
-----	-----	-----	-----
1400	1600	1250	5600

# Funciones MIN y MAX

Podemos usar MIN y MAX con cualquier tipo de datos.

```
SQL> SELECT MIN(hiredate), MAX(hiredate)
         2 FROM emp;
```

MIN (HIRED	MAX (HIRED
-----	-----
17-DEC-80	12-JAN-83

# La función COUNT

**COUNT(\*) devuelve el número de filas de una tabla.**

```
SQL> SELECT COUNT (*)  
2 FROM emp  
3 WHERE deptno = 30;
```

COUNT (\*)

-----

6



# Ejemplo de la función COUNT

**COUNT(*expr*)** devuelve el número de filas no nulas.

```
SQL> SELECT COUNT(comm)
      2 FROM emp
      3 WHERE deptno = 30;
```

COUNT (COMM)
-----
4

# Funciones de grupo y valores nulos

Ignoran valores nulos en las columnas.

```
SQL> SELECT AVG(comm)
      2 FROM emp;
```

AVG (COMM)

-----

550

# Uso de la función NVL con funciones de grupo

La función NVL fuerza a las funciones de grupo a incluir los valores nulos.

```
SQL> SELECT AVG (NVL (comm, 0) )  
2 FROM emp ;
```

```
AVG (NVL (COMM, 0) )  
-----  
157.14286
```

# Creación de grupos de datos

EMP

DEPTNO	SAL
10	2450
10	5000
10	1300
20	800
20	1100
20	3000
20	3000
20	2975
30	1600
30	2850
30	1250
30	950
30	1500
30	1250

2916.6667

“salario medio  
en la tabla  
EMP para cada  
departamento”

2175

1566.6667

DEPTNO	AVG (SAL)
10	2916.6667
20	2175
30	1566.6667

# Creación de grupos de datos: Cláusula GROUP BY

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

Se dividen las filas de una tabla en grupos más pequeños mediante el uso de esta cláusula.

# Uso de la cláusula GROUP BY

Todas las columnas de la lista de **SELECT** que no estén en funciones de grupo deben aparecer en la cláusula **GROUP BY** .

```
SQL> SELECT deptno, AVG(sal)
2 FROM emp
3 GROUP BY deptno;
```

DEPTNO	AVG (SAL)
10	2916.6667
20	2175
30	1566.6667

# Uso de la cláusula GROUP BY

La columna GROUP BY puede no aparecer en la lista de SELECT.

```
SQL> SELECT      AVG(sal)
  2  FROM          emp
  3  GROUP BY deptno;
```

AVG (SAL)
2916.6667
2175
1566.6667

# Agrupando por más de una columna

EMP

DEPTNO	JOB	SAL
10	MANAGER	2450
10	PRESIDENT	5000
10	CLERK	1300
20	CLERK	800
20	CLERK	1100
20	ANALYST	3000
20	ANALYST	3000
20	MANAGER	2975
30	SALESMAN	1600
30	MANAGER	2850
30	SALESMAN	1250
30	CLERK	950
30	SALESMAN	1500
30	SALESMAN	1250

“suma de los  
salarios de la tabla  
EMP para cada JOB  
agrupados por  
departamento”

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



# Uso de la cláusula GROUP BY con varias columnas

```
SQL> SELECT deptno, job, sum(sal)
2 FROM emp
3 GROUP BY deptno, job;
```

DEPTNO	JOB	SUM(SAL)
10	CLERK	1300
10	MANAGER	2450
10	PRESIDENT	5000
20	ANALYST	6000
20	CLERK	1900
...		

9 rows selected.

# Consultas inválidas

## Usando Funciones de grupo

Cualquier columna o expresión en la lista de la SELECT que no sea una función de agrupamiento debe estar en la cláusula GROUP BY .

```
SQL> SELECT deptno, COUNT(ename)
2 FROM emp;
```

**Column missing in the GROUP BY clause**

```
SELECT deptno, COUNT(ename)
*
```

```
ERROR at line 1:
```

```
ORA-00937: not a single-group group function
```

# Consultas inválidas

## Usando Funciones de grupo

- No podemos usar la cláusula WHERE para restringir grupos.
- Para ello se emplea la cláusula HAVING.

```
SQL> SELECT      deptno, AVG(sal)
  2  FROM          emp
  3  WHERE         AVG(sal) > 2000
  4  GROUP BY     deptno;
```

```
WHERE AVG(sal) > 2000
      *
```

```
ERROR at line 3:
```

```
ORA-00934: group function is not allowed here
```

**Cannot use the WHERE clause  
to restrict groups**

# Filtrando resultados de grupo

EMP

DEPTNO	SAL
-----	-----
10	2450
10	5000
10	1300
20	800
20	1100
20	3000
20	3000
20	2975
30	1600
30	2850
30	1250
30	950
30	1500
30	1250

5000

3000

2850

“máximo  
salario por  
departamento  
mayor que  
2900”

DEPTNO	MAX (SAL)
-----	-----
10	5000
20	3000

# Filtrando resultados de grupo : cláusula HAVING

Para restringir grupos se emplea la cláusula HAVING

- Las filas se agrupan.
- Se aplica la función de grupo.
- Los grupos que cumplan la cláusula HAVING son visualizados.

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

# Uso de la cláusula HAVING

```
SQL> SELECT      deptno, max(sal)
  2  FROM        emp
  3  GROUP BY    deptno
  4  HAVING      max(sal)>2900;
```

DEPTNO	MAX (SAL)
10	5000
20	3000

# Uso de la cláusula HAVING

```
SQL> SELECT      job, SUM(sal) PAYROLL
  2  FROM          emp
  3  WHERE         job NOT LIKE 'SALES%'
  4  GROUP BY     job
  5  HAVING        SUM(sal)>5000
  6  ORDER BY     SUM(sal) ;
```

JOB	PAYROLL
ANALYST	6000
MANAGER	8275

# Anidando funciones de grupo

Visualizar el máximo salario medio.

```
SQL> SELECT      max (avg (sal) )  
      2  FROM      emp  
      3  GROUP BY deptno;
```

```
MAX (AVG (SAL) )  
-----  
      2916.6667
```



# Resumen

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

Orden de evaluación de las cláusulas:

1.WHERE

2.GROUP BY

3.HAVING