

5

Datos Agregados por medio de Funciones de Grupo

Objetivos

Al completar esta lección, debería ser capaz de hacer lo siguiente:

- **Identificar las funciones de grupo disponibles.**
- **Explicar el uso de las funciones de grupo.**
- **Mostrar estadísticas para diferentes grupos usando la cláusula GROUP BY.**
- **Incluir o excluir filas agrupadas usando la cláusula HAVING.**

¿Qué son las Funciones de Grupo?

Las Funciones de Grupo operan sobre conjuntos de registros para dar un resultado a nivel 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 |

“máximo salario en la tabla EMP”

MAX (SAL)

5000

Tipos de Funciones de Grupo

- **AVG ([DISTINCT|ALL]n)**
- **COUNT ({ *|[DISTINCT|ALL]expr})**
- **MAX ([DISTINCT|ALL]expr)**
- **MIN ([DISTINCT|ALL]expr)**
- **STDDEV ([DISTINCT|ALL]x)**
- **SUM ([DISTINCT|ALL]n)**
- **VARIANCE ([DISTINCT|ALL]x)**

Tipos de Funciones de Grupo

- **AVG ([DISTINCT|ALL]n)**
- **COUNT ({ *|[DISTINCT|ALL]expr})**
- **MAX ([DISTINCT|ALL]expr)**
- **MIN ([DISTINCT|ALL]expr)**
- **STDDEV ([DISTINCT|ALL]x)**
- **SUM ([DISTINCT|ALL]n)**
- **VARIANCE ([DISTINCT|ALL]x)**

Uso de las Funciones AVG y SUM

Puede 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 |

Uso de las Funciones MIN y MAX

Puede usar MIN y MAX para cualquier tipo de dato.

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

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

Uso de la Función COUNT

COUNT(*) devuelve el nº de registros en una tabla.

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

| COUNT (*) |
|-----------|
| ----- |
| 6 |

Uso de la Función COUNT

COUNT(expr) devuelve el nº de registros “no nulos” en una tabla.

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

| COUNT (COMM) |
|--------------|
| ----- |
| 4 |

Funciones de Grupo y Val. Nulos

Las Funciones de Grupo ignoran los valores nulos de 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 valores nulos (null).

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

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

Creando Grupos de Datos

EMP

| DEPTNO | SAL |
|--------|-----------|
| 10 | 2450 |
| | 5000 |
| | 1300 |
| 20 | 800 |
| | 1100 |
| | 3000 |
| | 3000 |
| 20 | 2975 |
| | 1600 |
| | 2850 |
| | 1250 |
| | 950 |
| 30 | 1500 |
| | 1250 |
| | 1566.6667 |

“media de
salarios
en EMP
para cada
departamento”

2916.6667

2175

1566.6667

| DEPTNO | AVG (SAL) |
|--------|-----------|
| 10 | 2916.6667 |
| 20 | 2175 |
| 30 | 1566.6667 |

Creando Grupos de Datos: La Cláusula GROUP BY

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

Divide los registros de una tabla en grupos más pequeños, por medio de GROUP BY.

Uso de la Cláusula GROUP BY

Todas las columnas mencionadas en la SELECT que no son funciones de grupo, tienen que estar 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 referencia por GROUP BY no es necesario seleccionarla.

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

| AVG (SAL) |
|-----------|
| ----- |
| 2916.6667 |
| 2175 |
| 1566.6667 |

Agrupación por Más de Una Columna

EMP

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

“suma de salarios
de EMP para
cada oficio (job),
agrupado por
departamento”

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

Uso de la Cláusula GROUP BY sobre Múltiples 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 No válidas

Usando Funciones de Grupo

Cualquier columna o expresión en la SELECT que no sea una función agregada tiene que ser especificada en la cláusula GROUP BY.

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

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

Consultas No válidas

Usando Funciones de Grupo

- No puede usar una cláusula WHERE para restringir grupos.
- Utilice la cláusula HAVING para restringir grupos.

```
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
```

No puede usar la cláusula WHERE
para restringir grupos

Exclusión de Resultados de un 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 de 2900\$”

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

Exclusión de Resultados de un Grupo: Cláusula HAVING

Use la cláusula HAVING para restringir grupos:

- Los registros son agrupados.
- Se aplica la función de grupo.
- Los grupos que se corresponden con la cláusula HAVING se visualizan.

```
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%'  
  3  GROUP BY    job  
  4  HAVING      SUM(sal)>5000  
  5  ORDER BY    SUM(sal);
```

| JOB | PAYROLL |
|---------|---------|
| ANALYST | 6000 |
| MANAGER | 8275 |

Anidamiento de Funciones de Grupo

Visualizar la máxima media de salarios por departamento:

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

| MAX (AVG (SAL)) |
|------------------|
| ----- |
| 2916.6667 |

Resumen

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

Visión General de la Práctica

- Mostrar diferentes consultas que puedan usar todas las funciones de grupo.
- Agrupar las filas para obtener más de un resultado.
- Excluir grupos usando la cláusula HAVING.

