



Autor: **Nicola Strappazon**
Web: <http://nicola.strappazon.me>
Revisión: 17/11/11

Lección 4

Usando funciones para manipular grupos de datos

Este material se encuentra basado en el curso de Fundamentos a SQL de ORACLE, el cual es adaptado para el producto PostgreSQL, todos los ejemplos, códigos fuentes y la Base de Datos HR es propiedad de ORACLE.

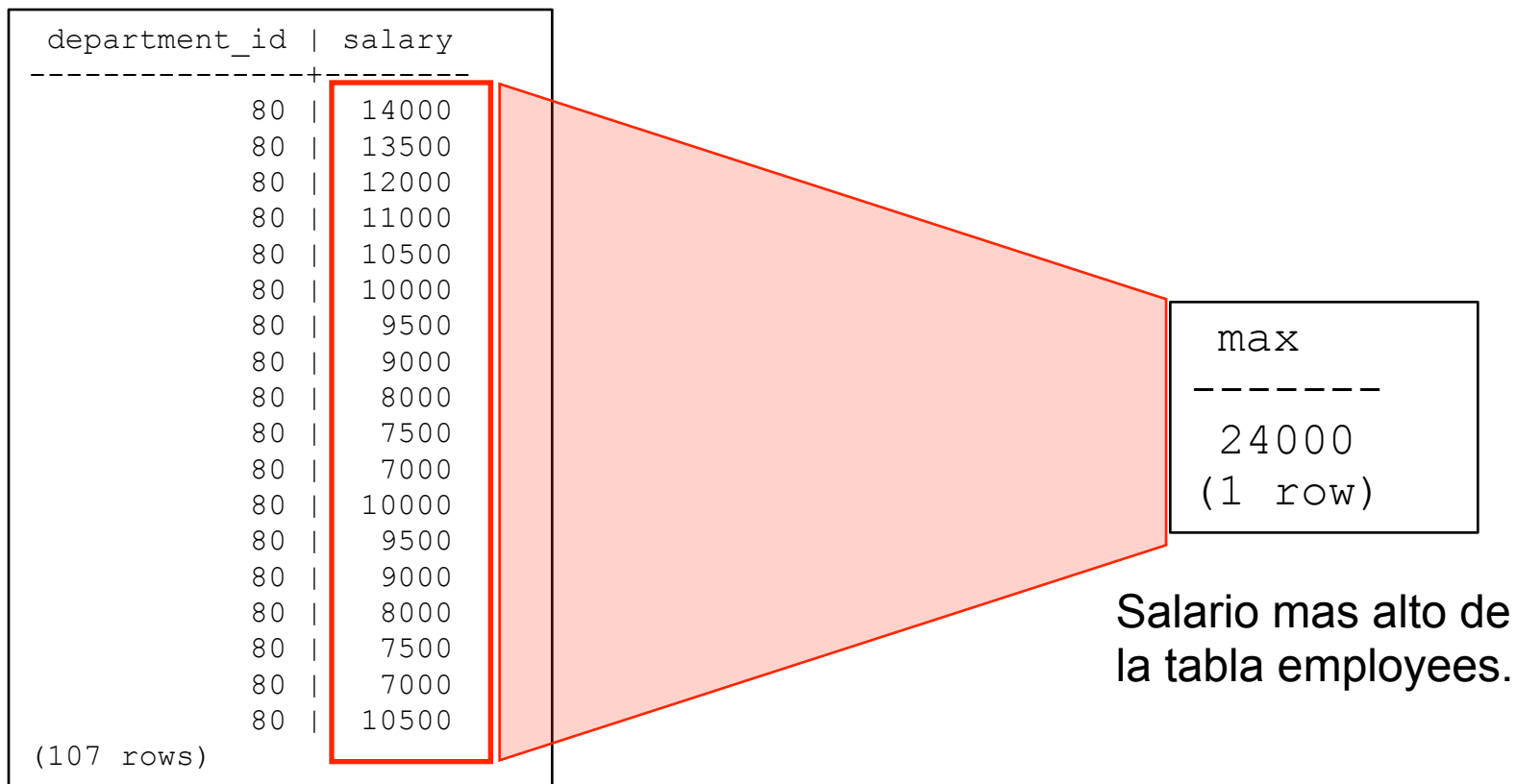
Objetivos

Al completar esta lección usted podrá entender los siguientes puntos:

- Identificar las funciones disponibles.
- Describir como usar una función de grupo.
- Agrupar datos mediante la cláusula **GROUP BY**.
- Incluir o excluir datos agrupados mediante el uso de la cláusula **HAVING**.

?Como trabajan las funciones de grupo?

Las funciones de grupo opera sobre un conjunto de registros y se retorna un solo resultado por grupo.



Funciones de grupo

Funciones básicas que realizan operaciones a un grupo de registros en específico:

- AVG
- COUNT
- MAX
- MIN
- SUM

Sintaxis Básica

```
SELECT [ column ], group function(column) [, ...]  
FROM   from_item  
[WHERE condition]  
[GROUP BY column]  
[ORDER BY column];
```

Usando las funciones AVG, MAX, MIN y SUM

```
SELECT AVG(salary), MAX(salary), MIN(salary), SUM(salary)
FROM   employees
WHERE  department_id = 80;
```

avg		max		min		sum
8955.8823529411764706		14000		6100		304500

(1 row)

Usando las funciones MAX y MIN

Puede utilizar las funciones MAX y MIN para los tipos de datos numéricos, caracteres y fechas.

```
SELECT MAX(hire_date) , MIN(hire_date)
FROM   employees
WHERE  department_id = 80;
```

max	min
2000-04-21	1996-01-30

(1 row)

Usando la funciones COUNT

- 1 La función COUNT (*) retorna la cantidad total de registros que se encuentran en una tabla.

```
SELECT COUNT (*)  
FROM employees;
```

```
count  
-----  
107
```

- 2 La función COUNT (expr) retorna la cantidad total de registros NO NULL que se encuentran en una tabla.

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

```
count  
-----  
35
```


Usando la palabra **DISTINCT**

- Usando la palabra **DISTINCT** se eliminan todos los valores duplicados.
- Puede emplear la palabra **DISTINCT** dentro de cualquier función.
- Retorna todo los elementos que **NO** son del tipo **NULL**.

Usando la palabra DISTINCT

La siguiente consulta cuenta cuantos departamentos se encuentran asignados en la tabla employees.

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

```
count
-----
      11
(1 row)
```

Funciones de grupo y valores NULL

- 1 Las funciones de grupo ignoran los valores NULL.

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

```
avg
```

```
-----
```

```
0.22285714285714285714
```

- 2 La función COALESCE obliga a las funciones de grupo que incluya los valores de tipo NULL.

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

```
avg
```

```
-----
```

```
0.07289719626168224299
```

Creando grupo de datos

department_id	salary
10	4400
20	13000
20	6000
30	3100
30	11000
30	2500
30	2900
40	6500
50	2500
50	8000
50	8200
60	6000
60	9000
70	10000
80	9500
80	9000
80	8000
90	24000
90	17000
90	17000
100	6900
100	9000
100	12000
110	8300
110	12000

(107 rows)

department_id	avg
10	4400.000000000000000000
20	9500.000000000000000000
30	4150.000000000000000000
40	6500.000000000000000000
50	3475.555555555555555556
60	5760.000000000000000000
70	10000.000000000000000000
80	8955.8823529411764706
90	19333.333333333333333333
100	8600.000000000000000000
110	10150.000000000000000000
	7000.000000000000000000

(12 rows)

Promedio del salario por cada departamento y por cada empleado del mismo departamento

Sintaxis Básica

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

Permite dividir los registros de una tabla en pequeños grupos usando la cláusula GROUP BY.

- No puedes utilizar Alias dentro de la cláusula **GROUP BY**.
- Solo puedes incluir el nombre de la columna dentro de la cláusula **GROUP BY**.
- Si no usas funciones de grupo no podrás apreciar resultados individuales.

Usando la cláusula GROUP BY

Todas las columnas dentro de la sentencia **SELECT** no deben estar en la función de grupo si no en la cláusula **GROUP BY**.

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

department_id	avg
10	4400.000000000000000000
20	9500.000000000000000000
30	4150.000000000000000000
40	6500.000000000000000000
50	3475.5555555555555556
60	5760.000000000000000000

(12 rows)

Usando la cláusula GROUP BY

Las columnas dentro la cláusula GROUP BY no tienen que estar dentro de la sentencia SELECT .

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

```
avg
-----
 7000.0000000000000000
10150.0000000000000000
 8600.0000000000000000
 19333.333333333333
 8955.8823529411764706
10000.0000000000000000
 5760.0000000000000000
 3475.5555555555555556
 6500.0000000000000000
 4150.0000000000000000
(12 rows)
```

Agrupando por mas de una columna

department_id	job_id	salary
10	AD_ASST	4400
30	PU_CLERK	3100
30	PU_CLERK	2800
30	PU_CLERK	2500
30	PU_CLERK	2600
30	PU_CLERK	2900
50	SH_CLERK	3800
50	SH_CLERK	3000
50	SH_CLERK	3400
50	SH_CLERK	4100
50	SH_CLERK	4200
50	SH_CLERK	2800
50	SH_CLERK	2500
60	IT_PROG	9000
60	IT_PROG	4200
60	IT_PROG	4800
60	IT_PROG	4800
60	IT_PROG	6000
100	FI_ACCOUNT	7700
100	FI_ACCOUNT	6900
100	FI_ACCOUNT	9000
100	FI_ACCOUNT	8200
100	FI_ACCOUNT	7800
100	FI_MGR	12000

(107 rows)

department_id	job_id	sum
10	AD_ASST	4400
20	MK_MAN	13000
20	MK_REP	6000
30	PU_CLERK	13900
30	PU_MAN	11000
40	HR_REP	6500
50	SH_CLERK	64300
50	ST_CLERK	55700
50	ST_MAN	36400
60	IT_PROG	28800
70	PR_REP	10000
80	SA_MAN	61000

(20 rows)

Promedio del salario por cada departamento y por cada empleado del mismo cargo y departamento

Usando la cláusula GROUP BY con múltiples columnas

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

department_id	job_id	sum
10	AD_ASST	4400
20	MK_MAN	13000
20	MK_REP	6000
30	PU_CLERK	13900
30	PU_MAN	11000
40	HR_REP	6500
50	SH_CLERK	64300
50	ST_CLERK	55700
50	ST_MAN	36400
60	IT_PROG	28800
70	PR_REP	10000
80	SA_MAN	61000

(20 rows)

Consultas mal escritas usando las funciones de grupo

Toda columna o expresión que no se encuentra dentro de una función de grupo, debe estar dentro de la cláusula GROUP BY.

```
SELECT department_id, SUM(salary)
FROM   employees;
```

```
ERROR: column "employees.department_id" must appear in the GROUP
BY clause or be used in an aggregate function
```

Consultas mal escritas usando las funciones de grupo

- No se puede usar condiciones en la cláusula **WHERE** para restringir grupos.
- Use la cláusula **HAVING** para restringir los grupos.
- No use las funciones de grupo dentro de la cláusula **WHERE**.

```
SELECT    department_id, job_id, AVG(salary)
FROM      employees
WHERE     AVG(salary) > 1000
GROUP BY  department_id, job_id
ORDER BY  department_id;
```

```
ERROR:  aggregates not allowed in WHERE clause
```

Restringiendo el resultado de los grupo con la cláusula HAVING

- Filtra por registros agrupados.
- Muestra la coincidencia definida en la cláusula **HAVING**.

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

Usando la cláusula HAVING

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

department_id	max
110	12000
100	12000
90	24000
80	14000
30	11000
20	13000

(6 rows)

Usando la cláusula HAVING

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

job_id	PAYROLL
PU_CLERK	13900
AD_PRES	24000
IT_PROG	28800
AD_VP	34000
ST_MAN	36400
FI_ACCOUNT	39600
ST_CLERK	55700
SA_MAN	61000
SH_CLERK	64300

(9 rows)

Funciones jerárquicas de grupo

No se puede usar las funciones de grupo en modo jerárquico.

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

```
ERROR:  aggregate function calls may not be nested
```

Resumen

En esta lección, usted debió entender como:

- Usar las funciones básicas de grupo en SQL.
 - **SUM**
 - **COUNT**
 - **MAX**
 - **MIN**
 - **AVG**
- Escribir consultas usando la cláusula **GROUP BY**.
- Escribir consultas usando la cláusula **HAVING**.