

BASES DE DATOS

Consultas con agrupaciones

©Jesús García, 2021 | <http://jgarcia.dev>



BASES DE DATOS

Consultas con agrupaciones

Introducción

Funciones de grupo

Agrupaciones

Restricciones de fila y grupo

Ejemplos

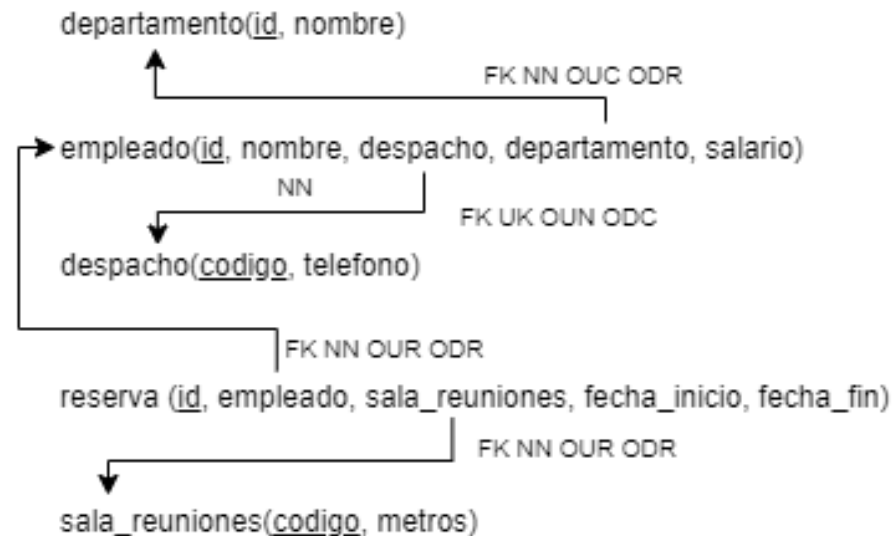
Introducción

Para la consulta de los datos de la base de datos se utiliza el lenguaje de consulta de datos (*Data Query Language* o *DQL*).

En esta unidad veremos cómo realizar operaciones sobre grupos de filas dentro de una misma tabla.

Introducción

Todos los ejemplos de este tema hacen uso de la base de datos "oficina" cuyo grafo relacional es el siguiente:



- La fecha de fin de reserva debe ser posterior a la fecha de inicio.
- El salario por defecto es 1200€.

Puedes crear la base de datos ejecutando el *script oficina-con-datos.sql*

BASES DE DATOS

Consultas con agrupaciones

Introducción

Funciones de grupo

Agrupaciones

Restricciones de fila y grupo

Ejemplos

Funciones de grupo

Son funciones del lenguaje SQL que permiten realizar operaciones sobre un grupo de filas de una tabla.

Función	Descripción
COUNT(columna)	Cuenta el número de valores de la columna que no son nulos.
COUNT(*)	Cuenta el número de filas.
SUM(columna)	Suma los valores.
AVG(columna)	Calcula la media aritmética.
MIN(columna)	Obtiene el valor mínimo de la columna.
MAX(columna)	Obtiene el valor máximo de la columna.



Las funciones de grupo no se pueden utilizar en el WHERE.

Funciones de grupo

Obtén el número de empleados:

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

Obtén el número de empleados con despacho propio:

```
SELECT COUNT(despacho) FROM empleado;
```

O también:

```
SELECT COUNT(*) FROM empleado WHERE despacho IS NOT NULL;
```

Funciones de grupo

Obtén la suma de todos los salarios:

```
SELECT SUM(salario) FROM empleado;
```

Obtén el salario medio de los empleados:

```
SELECT AVG(salario) FROM empleado;
```


Funciones de grupo

Obtén el tamaño de la sala de reuniones más pequeña:

```
SELECT MIN(metros) FROM sala_reuniones;
```

Obtén el tamaño de la sala de reuniones más grande:

```
SELECT MAX(metros) FROM sala_reuniones;
```

Obtén el tamaño de la sala de reuniones más pequeña y más grande:

```
SELECT MIN(metros), MAX(metros) FROM sala_reuniones;
```

BASES DE DATOS

Consultas con agrupaciones

Introducción

Funciones de grupo

Agrupaciones

Restricciones de fila y grupo

Ejemplos

Agrupaciones

Mediante la cláusula GROUP BY podemos realizar agrupaciones de filas para poder hacer cálculos sobre ellas.

- Tras la cláusula GROUP BY indicaremos la columna o columnas de la tabla por las que queremos realizar la agrupación.
- La cláusula GROUP BY se escribe después de la condición del WHERE.

Agrupaciones

```
SELECT nombre, departamento FROM empleado  
GROUP BY departamento;
```

Al agrupar por la columna departamento se crean 5 grupos.

De cada grupo obtenemos su nombre y departamento.

nombre	departamento
Sergio	1
Juan	2
Sara	3
Elena	4
Leopoldo	5

id	nombre	despacho	departamento	salario
1	Juan	NULL	2	1200.00
2	Sara	DS01	3	2700.00
3	Sergio	NULL	1	1200.00
4	Marta	NULL	1	1400.00
5	Elena	DS02	4	1700.00
6	Sebastián	NULL	4	1600.00
7	Sara	NULL	4	1300.00
8	Martín	DS03	1	1300.00
9	Laura	NULL	1	1200.00
10	Pedro	NULL	2	1200.00
11	Leopoldo	DS04	5	1700.00

Agrupaciones

```
SELECT nombre, departamento FROM empleado  
GROUP BY departamento;
```

Si en una de las columnas seleccionadas del grupo existen diferentes valores, el valor obtenido será el primero de ellos.

nombre	departamento
Sergio	1
Juan	2
Sara	3
Elena	4
Leopoldo	5

id	nombre	despacho	departamento	salario
1	Juan	NULL	2	1200.00
2	Sara	DS01	3	2700.00
3	Sergio	NULL	1	1200.00
4	Marta	NULL	1	1400.00
5	Elena	DS02	4	1700.00
6	Sebastián	NULL	4	1600.00
7	Sara	NULL	4	1300.00
8	Martín	DS03	1	1300.00
9	Laura	NULL	1	1200.00
10	Pedro	NULL	2	1200.00
11	Leopoldo	DS04	5	1700.00

Agrupaciones

```
SELECT sala_reuniones, empleado FROM reserva  
GROUP BY sala_reuniones;
```

¿Cuántos grupos se formarán?

id	empleado	sala_reuniones	fecha_inicio	fecha_fin
1	1	SR01	2021-03-25 18:30:00	2021-03-25 19:30:00
2	2	SR01	2021-07-23 09:00:00	2021-07-23 10:30:00
3	2	SR02	2021-11-10 09:00:00	2021-11-10 10:00:00
4	3	SR03	2021-05-07 12:00:00	2021-05-07 13:00:00

Agrupaciones

```
SELECT sala_reuniones, empleado FROM reserva  
GROUP BY sala_reuniones;
```

¿Cuántos grupos se formarán?

id	empleado	sala_reuniones	fecha_inicio	fecha_fin
1	1	SR01	2021-03-25 18:30:00	2021-03-25 19:30:00
2	2	SR01	2021-07-23 09:00:00	2021-07-23 10:30:00
3	2	SR02	2021-11-10 09:00:00	2021-11-10 10:00:00
4	3	SR03	2021-05-07 12:00:00	2021-05-07 13:00:00

En total 3 grupos.

Agrupaciones

```
SELECT sala_reuniones, empleado FROM reserva  
GROUP BY sala_reuniones;
```

¿Qué obtendrá la sentencia SELECT?

id	empleado	sala_reuniones	fecha_inicio	fecha_fin
1	1	SR01	2021-03-25 18:30:00	2021-03-25 19:30:00
2	2	SR01	2021-07-23 09:00:00	2021-07-23 10:30:00
3	2	SR02	2021-11-10 09:00:00	2021-11-10 10:00:00
4	3	SR03	2021-05-07 12:00:00	2021-05-07 13:00:00

sala_reuniones	empleado
SR01	1
SR02	2
SR03	3

Agrupaciones

Lo interesante de realizar agrupaciones es poder utilizar funciones de grupo.

Obtén el número de empleados de cada departamento.

```
SELECT departamento,  
COUNT (*) FROM empleado  
GROUP BY departamento;
```

departamento	COUNT(*)
1	4
2	2
3	1
4	3
5	1

id	nombre	despacho	departamento	salario
1	Juan	NULL	2	1200.00
2	Sara	DS01	3	2700.00
3	Sergio	NULL	1	1200.00
4	Marta	NULL	1	1400.00
5	Elena	DS02	4	1700.00
6	Sebastián	NULL	4	1600.00
7	Sara	NULL	4	1300.00
8	Martín	DS03	1	1300.00
9	Laura	NULL	1	1200.00
10	Pedro	NULL	2	1200.00
11	Leopoldo	DS04	5	1700.00

Agrupaciones

Obtén el salario mínimo y máximo de cada departamento.

```
SELECT departamento, MIN(salario), MAX(salario)
FROM empleado
GROUP BY departamento;
```

departamento	MIN(salario)	MAX(salario)
1	1200.00	1400.00
2	1200.00	1200.00
3	2700.00	2700.00
4	1300.00	1700.00
5	1700.00	1700.00

Agrupaciones

Podemos aplicar condiciones a un grupo mediante la cláusula HAVING.

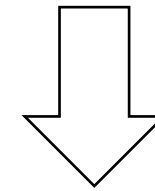
- La cláusula HAVING funciona de forma similar que la cláusula WHERE pero en vez de aplicar la condición sobre cada una de las filas, la aplica sobre cada uno de los grupos.
- La cláusula HAVING se escribe después de la cláusula GROUP BY.
- Podemos utilizar funciones de grupo en la condición HAVING.

Agrupaciones

Obtén el salario medio de cada departamento.

```
SELECT departamento, AVG(salario) FROM empleado  
GROUP BY departamento;
```

id	nombre	despacho	departamento	salario
1	Juan	NULL	2	1200.00
2	Sara	DS01	3	2700.00
3	Sergio	NULL	1	1200.00
4	Marta	NULL	1	1400.00
5	Elena	DS02	4	1700.00
6	Sebastián	NULL	4	1600.00
7	Sara	NULL	4	1300.00
8	Martín	DS03	1	1300.00
9	Laura	NULL	1	1200.00
10	Pedro	NULL	2	1200.00
11	Leopoldo	DS04	5	1700.00



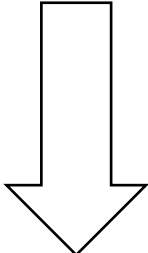
departamento	AVG(salario)
1	1275.000000
2	1200.000000
3	2700.000000
4	1533.333333
5	1700.000000

Agrupaciones

Obtén los departamentos donde el salario medio es superior a 1500€.

```
SELECT departamento, AVG(salario) FROM empleado  
GROUP BY departamento HAVING AVG(salario) > 1500;
```

id	nombre	despacho	departamento	salario
1	Juan	NULL	2	1200.00
2	Sara	DS01	3	2700.00
3	Sergio	NULL	1	1200.00
4	Marta	NULL	1	1400.00
5	Elena	DS02	4	1700.00
6	Sebastián	NULL	4	1600.00
7	Sara	NULL	4	1300.00
8	Martín	DS03	1	1300.00
9	Laura	NULL	1	1200.00
10	Pedro	NULL	2	1200.00
11	Leopoldo	DS04	5	1700.00

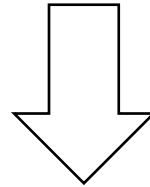


departamento	AVG(salario)
3	2700.000000
4	1533.333333
5	1700.000000

Agrupaciones

Obtén el identificador de aquellos empleados que hayan realizado más de una reserva.

```
SELECT * FROM reserva;
```

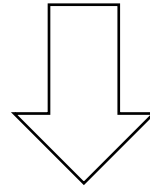


id	empleado	sala_reuniones	fecha_inicio	fecha_fin
1	1	SR01	2021-03-25 18:30:00	2021-03-25 19:30:00
2	2	SR01	2021-07-23 09:00:00	2021-07-23 10:30:00
3	2	SR02	2021-11-10 09:00:00	2021-11-10 10:00:00
4	3	SR03	2021-05-07 12:00:00	2021-05-07 13:00:00

Agrupaciones

Obtén el identificador de aquellos empleados que hayan realizado más de una reserva.

```
SELECT * FROM reserva;
```



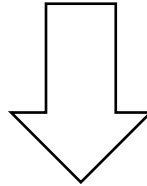
```
GROUP BY empleado
```

id	empleado	sala_reuniones	fecha_inicio	fecha_fin
1	1	SR01	2021-03-25 18:30:00	2021-03-25 19:30:00
2	2	SR01	2021-07-23 09:00:00	2021-07-23 10:30:00
3	2	SR02	2021-11-10 09:00:00	2021-11-10 10:00:00
4	3	SR03	2021-05-07 12:00:00	2021-05-07 13:00:00

Agrupaciones

Obtén el identificador de aquellos empleados que hayan realizado más de una reserva.

```
SELECT empleado, COUNT(*) FROM reserva GROUP BY empleado;
```

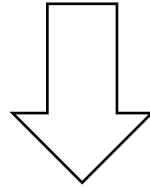


empleado	COUNT(*)
1	1
2	2
3	1

Agrupaciones

Obtén el identificador de aquellos empleados que hayan realizado más de una reserva.

```
SELECT empleado, COUNT(*) FROM reserva  
GROUP BY empleado HAVING COUNT(*) > 1;
```



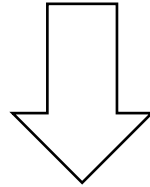
empleado	COUNT(*)
2	2

Agrupaciones

Obtén el identificador de aquellos empleados que hayan realizado más de una reserva.

Podemos utilizar los alias en la clausula HAVING

```
SELECT empleado, COUNT(*) AS num_reservas FROM reserva  
GROUP BY empleado HAVING num_reservas > 1;
```



empleado	num_reservas
2	2

Orden de las cláusulas

Recuerda que el orden de las cláusulas en una consulta es el siguiente:

SELECT - FROM - WHERE - GROUP BY - HAVING - ORDER BY - LIMIT

Donde las únicas cláusulas obligatorias son SELECT y FROM.

BASES DE DATOS

Consultas con agrupaciones

Introducción

Funciones de grupo

Agrupaciones

Restricciones de fila y grupo

Ejemplos

Restricciones de fila y de grupo

Hasta ahora hemos visto que podemos aplicar dos tipos de restricciones o condiciones.

- Las de la cláusula WHERE, que son restricciones de fila.
- Las de la cláusula HAVING, que son restricciones de grupo.

Restricciones de fila y de grupo

Obtén el id de los departamentos que tienen más de un empleado con un salario superior a 1300€.

1º Obtenemos los empleados con salario superior a 1300€:
restricción de fila.

```
SELECT * FROM empleado WHERE salario > 1300;
```

id	nombre	despacho	departamento	salario
2	Sara	DS01	3	2700.00
4	Marta	NULL	1	1400.00
5	Elena	DS02	4	1700.00
6	Sebastián	NULL	4	1600.00
11	Leopoldo	DS04	5	1700.00

Restricciones de fila y de grupo

Obtén el id de los departamentos que tienen más de un empleado con un salario superior a 1300€.

2º Agrupamos por departamento y contabilizamos el número de filas (número de empleados).

```
SELECT departamento, COUNT(*) AS num_empleados  
FROM empleado  
WHERE salario > 1300 GROUP BY departamento;
```

departamento	num_empleados
1	1
3	1
4	2
5	1

Restricciones de fila y de grupo

Obtén el id de los departamentos que tienen más de un empleado con un salario superior a 1300€.

3º Obtenemos solo aquellas filas en las que el número de empleados es superior a 1: **restricción de grupo.**

```
SELECT departamento, COUNT(*) AS num_empleados  
FROM empleado  
WHERE salario > 1300  
GROUP BY departamento HAVING num_empleados > 1;
```

departamento	num_empleados
4	2

Restricciones de fila y de grupo

Obtén el salario medio de aquellos departamentos cuyo identificador es superior a 3.

1º Obtenemos los empleados de los departamentos cuyo identificador es superior a 3: **restricción de fila.**

```
SELECT * FROM empleado WHERE departamento > 3;
```

id	nombre	despacho	departamento	salario
5	Elena	DS02	4	1700.00
6	Sebastián	NULL	4	1600.00
7	Sara	NULL	4	1300.00
11	Leopoldo	DS04	5	1700.00

Restricciones de fila y de grupo

Obtén el salario medio de aquellos departamentos cuyo identificador es superior a 3.

2º Agrupamos por departamento y obtenemos el salario.

```
SELECT departamento, AVG(salario)  
FROM empleado WHERE departamento > 3  
GROUP BY departamento;
```

departamento	AVG(salario)
4	1533.333333
5	1700.000000

Restricciones de fila y de grupo

Obtén el salario medio de aquellos departamentos cuyo identificador es superior a 3.

También se puede resolver con restricción de grupo.

```
SELECT departamento, AVG(salario)
FROM empleado
GROUP BY departamento HAVING departamento > 3;
```

departamento	AVG(salario)
4	1533.333333
5	1700.000000

Restricciones de fila y de grupo

Algunas consultas se pueden resolver tanto con GROUP BY como con una condición WHERE.

Obtén el salario medio de departamento 1.



```
SELECT AVG(salario) FROM empleado  
GROUP BY departamento HAVING departamento = 1;
```



```
SELECT AVG(salario) FROM empleado  
WHERE departamento = 1;
```

En estos casos siempre será más óptimo utilizar WHERE.

BASES DE DATOS

Consultas con agrupaciones

Introducción

Funciones de grupo

Agrupaciones

Restricciones de fila y grupo

Ejemplos

Ejemplo 1

Obtén el sueldo medio según la edad de los empleados.

Empleados					
<u>dni</u>	nombre	edad	anyos	dept	sueldo
20000000	Pedro	24	2	ventas	1200
30000000	María	26	1	ventas	900
40000000	Luis	24	4	ventas	800
50000000	Sofía	24	2	técnico	600
60000000	María	26	4	técnico	1100

Ejemplo 1 - Solución

Obtén el sueldo medio según la edad de los empleados.

Si agrupamos por edad:

GROUP BY edad;

Empleados					
<u>dni</u>	nombre	edad	anyos	dept	sueldo
20000000	Pedro	24	2	ventas	1200
30000000	María	26	1	ventas	900
40000000	Luis	24	4	ventas	800
50000000	Sofía	24	2	técnico	600
60000000	María	26	4	técnico	1100

Obtendríamos 2 grupos.

Ejemplo 1 - Solución

Obtén el sueldo medio según la edad de los empleados.

Solo faltaría indicar en la cláusula SELECT la columna edad y utilizar la función de grupo AVG.

```
SELECT edad, AVG (sueldo) FROM empleados  
GROUP BY edad;
```

edad	avg
24	866,66
26	1000

Ejemplo 2

Obtener el sueldo máximo de los empleados que tienen 4 años de experiencia.

Empleados					
<u>dni</u>	nombre	edad	anyos	dept	sueldo
20000000	Pedro	24	2	ventas	1200
30000000	María	26	1	ventas	900
40000000	Luis	24	4	ventas	800
50000000	Sofía	24	2	técnico	600
60000000	María	26	4	técnico	1100

Ejemplo 2 - Solución

Obtener el sueldo máximo de los empleados que tienen 4 años de experiencia.

¿Es necesario GROUP BY? ¡Solo necesitamos las filas de aquellos empleados que tienen 4 años de experiencia!

WHERE años = 4;

Empleados					
<u>dni</u>	nombre	edad	anyos	dept	sueldo
20000000	Pedro	24	2	ventas	1200
30000000	María	26	1	ventas	900
40000000	Luis	24	4	ventas	800
50000000	Sofía	24	2	técnico	600
60000000	María	26	4	técnico	1100

Ejemplo 2 - Solución

Obtener el sueldo máximo de los empleados que tienen 4 años de experiencia.

Ahora solo nos falta añadir el SELECT indicando que queremos obtener el sueldo máximo de las filas seleccionadas.

```
SELECT MAX(sueldo) FROM empleados WHERE anyos = 4;
```

max(Sueldo)
1100

Ejemplo 3

Obtener los años de experiencia, la edad media y el sueldo máximo de los empleados según la cantidad de años de experiencia que tengan.

Empleados					
dni	nombre	edad	anyos	dept	sueldo
20000000	Pedro	24	2	ventas	1200
30000000	María	26	1	ventas	900
40000000	Luis	24	4	ventas	800
50000000	Sofía	24	2	técnico	600
60000000	María	26	4	técnico	1100

Ejemplo 3 - Solución

Obtener los años de experiencia, la edad media y el sueldo máximo de los empleados según la cantidad de años de experiencia que tengan.

Si agrupamos por años de experiencia:

GROUP BY años;

Empleados					
dni	nombre	edad	anyos	dept	sueldo
20000000	Pedro	24	2	ventas	1200
30000000	María	26	1	ventas	900
40000000	Luis	24	4	ventas	800
50000000	Sofía	24	2	técnico	600
60000000	María	26	4	técnico	1100

Obtenemos 3 grupos.

Ejemplo 3 - Solución

Obtener los años de experiencia, la edad media y el sueldo máximo de los empleados según la cantidad de años de experiencia que tengan.

```
SELECT anyos, AVG(edad), MAX(sueldo)
FROM empelados
GROUP BY anyos;
```

anyos	avg(edad)	max(sueldo)
1	26	900
2	24	1200
4	25	1100

Ejemplo 4

Obtener el sueldo medio según la edad de aquellos empleados que trabajen en el departamento de ventas.

Empleados					
dni	nombre	edad	anyos	dept	sueldo
20000000	Pedro	24	2	ventas	1200
30000000	María	26	1	ventas	900
40000000	Luis	24	4	ventas	800
50000000	Sofía	24	2	técnico	600
60000000	María	26	4	técnico	1100

Ejemplo 4 - Solución

Obtener el sueldo medio según la edad de aquellos empleados que trabajen en el departamento de ventas.

Primero nos tendremos que quedar con los empleados que trabajan en el departamento de ventas.

```
WHERE dept = 'ventas'
```

Empleados					
dni	nombre	edad	anyos	dept	sueldo
20000000	Pedro	24	2	ventas	1200
30000000	María	26	1	ventas	900
40000000	Luis	24	4	ventas	800
50000000	Sofía	24	2	técnico	600
60000000	María	26	4	técnico	1100

Ejemplo 4 - Solución

Obtener el sueldo medio según la edad de aquellos empleados que trabajen en el departamento de ventas.

Pero también tenemos que agrupar por edad:

```
WHERE dept='ventas' GROUP BY edad;
```

Empleados					
dni	nombre	edad	anyos	dept	sueldo
20000000	Pedro	24	2	ventas	1200
30000000	María	26	1	ventas	900
40000000	Luis	24	4	ventas	800

Ejemplo 4 - Solución

Obtener el sueldo medio según la edad de aquellos empleados que trabajen en el departamento de ventas.

```
SELECT edad, AVG(sueldo) FROM empleados  
WHERE dept = 'ventas'  
GROUP BY edad;
```

edad	avg(sueldo)
24	1000
26	900