



# SQL: agregación

cálculos de agregación



Por cada pedido, coste total para el usuario, para los pedidos 1 y 2


```
select * from lmped
where numpedido IN (1,2)
```

El coste total por pedido es la suma de (importe\*cantidad) de todas sus líneas

numPedido	linea	articulo	importe	cantidad
1	1	A1085	145,00	3
1	2	A0214	327,00	2
2	1	A1086	349,00	4
2	2	A0818	2717,24	4
2	3	A1079	1134,00	1
2	4	A0003	1075,00	1

QUIERO:

numpedido	coste
1	1089,00
2	14473,96

SQL: agregaciónfundamentos de las bases de datos

Nos piden un cálculo de agregación. Se dice así porque es información que se extrae de los datos en bruto, examinando las filas y agregando sus datos en un determinado cálculo, como por ejemplo una suma para obtener totales o un promedio por meses.

Ahora deseamos calcular para cada número de pedido el coste total, que se ha pagado en total por cada pedido. Lo hacemos solo con los pedidos 1 y 2 por claridad del ejemplo.

Por cada pedido, coste total para el usuario, para los pedidos 1 y 2

select l.\*, importe\*cantidad coste  
from linped l  
where numpedido IN (1,2)

numPedido	linea	articulo	importe	cantidad	coste
1	1	A1085	145,00	X 3 =	435,00
1	2	A0214	327,00	X 2 =	654,00
2	1	A1086	349,00	X 4 =	1396,00
2	2	A0818	2717,24	X 4 =	10868,96
2	3	A1079	1134,00	X 1 =	1134,00
2	4	A0003	1075,00	X 1 =	1075,00

QUIERO:

numpedido	coste
1	1089,00
2	14473,96

SQL: agregación

fundamentos de las bases de datos

Lo que de momento sí sabemos hacer es calcular, para cada fila, ese coste. Pero sigue sin ser lo que quiero, que es sumar esa cantidad por cada pedido.

Por cada pedido, coste total para el usuario, para los pedidos 1 y 2

```
select numpedido,
SUM(importe*cantidad) coste
from lmped
where numpedido IN (1,2)
```

numPedido	linea	articulo	importe	cantidad	coste
1	1	A1085	145,00	3	435,00
1	2	A0214	327,00	2	654,00
2	1	A1086	349,00	4	1396,00
2	2	A0818	2717,24	4	10868,96
2	3	A1079	1134,00	1	1134,00
2	4	A0003	1075,00	1	1075,00

QUIERO:

numpedido	coste
1	1089,00
2	14473,96



Casi lo tenemos, la función que necesitamos es SUM(), pero ¿cómo hago para asocie esos cálculos con los números de pedido? ¿Cómo le digo que el criterio de agregación es numPedido?

Por cada pedido, coste total para el usuario, para los pedidos 1 y 2

```
select numpedido,  
SUM(importe*cantidad) coste  
from linped  
where numpedido IN (1,2)
```

*/\* Error de SQL (1140): Mixing of GROUP columns (MIN(),MAX(),COUNT(),...) with no GROUP columns is illegal if there is no GROUP BY clause \*/*

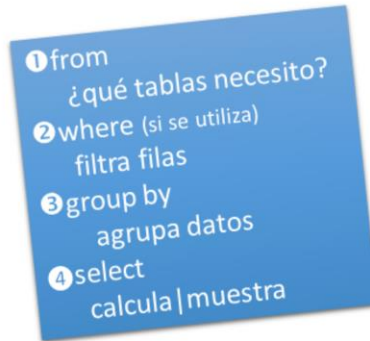
	numpedido	coste
QUIERO:	1	1089,00
	2	14473,96

SQL: agregación fundamentos de las bases de datos

En efecto, nos falta algo, pero el mensaje de error ya nos indica el qué: **la cláusula group by.**

Por cada pedido, coste total para el usuario, para los pedidos 1 y 2

```
select numpedido,
SUM(importe*cantidad) coste
from lmped
where numpedido IN (1,2)
group by numpedido
```



OBTENGO:

numpedido	coste
1	1089,00
2	14473,96

SQL: agregación

fundamentos de las bases de datos



Group by dirige el cálculo de las funciones de agregación, SUM() en nuestro caso. Siempre debe especificarse el criterio de agregación, numPedido aquí.

Sé consciente, nuevamente, del orden en que se "ejecutan" las cláusulas: primero *from*, luego *where*, a continuación *group by* y, por último, *select*.

## Group by

```
select x, y, z, f(exp), g(exp) from tabla
group by x, y, z
```

- En *group by*

- al menos TODAS las columnas (expresiones) no agregadas
- pero puedo poner más

```
select x, y, z, f(exp), g(exp) from tabla
group by x, y, z, a
```

```
select nombre,apellidos, count(*) from usuario group by nombre,apellidos
```

```
select nombre,apellidos, count(*) from usuario group by email,nombre,apellidos
```

```
select count(*) from usuario group by email
```

SQL: agregación

fundamentos de las bases de datos



En el estándar de SQL es obligatorio poner en *group by* TODAS las expresiones —normalmente columnas— no agregadas. Sin embargo, podrían ponerse más.

En el ejemplo de abajo agrupamos por nombre y apellidos de usuario. Pero estos datos no son el identificador, puede haber varios usuarios con el mismo nombre y apellidos. Lo que los distingue es el email, único para cada uno. Evidentemente, es más habitual incorporar el email a la salida, pero dependerá de nuestras necesidades.

Incluso podría no haber nada más que el cálculo de agregación en *select*. No suele ser lo más útil en la mayoría de las ocasiones, pero se puede hacer.

Por cada mes, cantidad de filas de cestas, cantidad de usuarios y cantidad de usuarios diferentes

- La función count() tiene dos usos diferentes
  - count(\*) cuenta filas
  - count(expresión) cuenta valores distintos de *null*

```
select month(fecha) mes, articulo,
usuario
from cesta
```

mes	articulo	usuario
10	A0024	bmm@agwab.com
10	A0024	emr@hotmail.com
10	A0046	bmm@agwab.com
10	A0046	emr@hotmail.com
11	A0072	arc@colegas.com
10	A0072	emr@hotmail.com
1	A0362	gmm@agwab.com
10	A0686	bmm@agwab.com
11	A0686	emr@hotmail.com
12	A0703	amp@colegas.com
10	A0778	bmm@agwab.com
11	A0778	emr@hotmail.com
11	A0954	emr@hotmail.com
11	A1076	emr@hotmail.com
11	A1077	jra@colegas.com



La función count() es algo diferente a las demás funciones de agregación. Tiene dos usos: contar filas y contar valores. La consulta que mostramos son los datos de partida para obtener lo que queremos: "por cada mes, cantidad de cestas, cantidad de usuarios y cantidad de usuarios diferentes".

Cuando utilizamos count(\*) queremos contar filas. Por ejemplo, cuenta manualmente las filas que tienen un valor 11 en "mes": te saldrá 6.



Por cada mes, cantidad de filas de cestas, cantidad de usuarios y cantidad de usuarios diferentes

- La función count() tiene dos usos diferentes
  - count(\*) cuenta filas
  - count(expresión) cuenta valores distintos de null

mes	articulo	usuario
10	A0024	bmm@agwab.com
10	A0024	emr@hotmail.com
10	A0046	bmm@agwab.com
10	A0046	emr@hotmail.com
11	A0072	arc@colegas.com
10	A0072	emr@hotmail.com
1	A0362	gmm@agwab.com
10	A0686	bmm@agwab.com
11	A0686	emr@hotmail.com
12	A0703	amp@colegas.com
10	A0778	bmm@agwab.com
11	A0778	emr@hotmail.com
11	A0954	emr@hotmail.com
11	A1076	emr@hotmail.com
11	A1077	jra@colegas.com

```
select month(fecha) mes,  
count(*) filas  
from cesta  
group by month(fecha)
```

mes	filas
1	1
10	7
11	6
12	1



Utilizando group by estamos pidiendo que calcule cuántas filas hay en CESTA por cada mes. Comprueba que, efectivamente, el cálculo es correcto.

Por cada mes, cantidad de filas de cestas, cantidad de usuarios y cantidad de usuarios diferentes

- La función count() tiene dos usos diferentes
  - count(\*) cuenta filas
  - count(expresión) cuenta valores distintos de null

mes	articulo	usuario
10	A0024	bmm@agwab.com
10	A0024	emr@hotmail.com
10	A0046	bmm@agwab.com
10	A0046	emr@hotmail.com
11	A0072	arc@colegas.com
10	A0072	emr@hotmail.com
1	A0362	gmm@agwab.com
10	A0686	bmm@agwab.com
11	A0686	emr@hotmail.com
12	A0703	amp@colegas.com
10	A0778	bmm@agwab.com
11	A0778	emr@hotmail.com
11	A0954	emr@hotmail.com
11	A1076	emr@hotmail.com
11	A1077	jra@colegas.com

```
select month(fecha) mes,  
count(*) filas,  
count(usuario) usuarios  
from cesta  
group by month(fecha)
```

mes	filas	usuarios
1	1	1
10	7	7
11	6	6
12	1	1



Añadimos el otro uso de COUNT(), la cuenta de usuarios por cada mes. En este caso coincide con la cuenta de filas, porque no hay ninguna fila que tenga un usuario nulo. **Count(columna) cuenta los valores distintos de null en "columna".**

Por cada mes, cantidad de filas de cestas, cantidad de usuarios y cantidad de usuarios diferentes

- La función count() tiene dos usos diferentes
  - count(\*) cuenta filas
  - count(expresión) cuenta valores distintos de null

```
select month(fecha) mes,  
count(*) filas,  
count(usuario) usuarios,  
count(distinct usuario) diferentes  
from cesta  
group by month(fecha)
```

mes	articulo	usuario
10	A0024	bmm@agwab.com
10	A0024	emr@hotmail.com
10	A0046	bmm@agwab.com
10	A0046	emr@hotmail.com
11	A0072	arc@colegas.com
10	A0072	emr@hotmail.com
1	A0362	gmm@agwab.com
10	A0686	bmm@agwab.com
11	A0686	emr@hotmail.com
12	A0703	amp@colegas.com
10	A0778	bmm@agwab.com
11	A0778	emr@hotmail.com
11	A0954	emr@hotmail.com
11	A1076	emr@hotmail.com
11	A1077	jra@colegas.com

mes	filas	usuarios	diferentes
1	1	1	1
10	7	7	2
11	6	6	3
12	1	1	1



Si añadimos el modificador *distinct* a la cuenta de valores, entonces lo que nos muestra es la cantidad de valores distintos de usuario por cada mes. Fíjate en los datos sin procesar que, efectivamente, los usuarios se repiten en cada mes.

Por cada mes, cantidad de filas de cestas, cantidad de usuarios y cantidad de usuarios diferentes

- count(\*) y count(columna) no siempre coinciden
  - si tuviéramos estos datos...

```
select month(fecha) mes,  
count(*) filas,  
count(usuario) usuarios,  
count(distinct usuario) diferentes  
from cesta  
group by month(fecha)
```

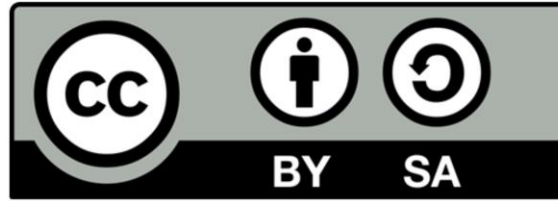
mes	articulo	usuario
10	A0024	null
10	A0024	emr@hotmail.com
10	A0046	bmm@agwab.com
10	A0046	emr@hotmail.com
11	A0072	arc@colegas.com
10	A0072	emr@hotmail.com
1	A0362	gmm@agwab.com
10	A0686	bmm@agwab.com
11	A0686	emr@hotmail.com
12	A0703	amp@colegas.com
10	A0778	bmm@agwab.com
11	A0778	emr@hotmail.com
11	A0954	emr@hotmail.com
11	A1076	emr@hotmail.com
11	A1077	jra@colegas.com

mes	filas	usuarios	diferentes
1	1	1	1
10	7	6	2
11	6	6	3
12	1	1	1



Cuidado con los enunciados y lo que nos pidan, a veces coincidirá la cuenta de filas, a veces tendremos que contar valores.

## licencias



Todos los logotipos y marcas registradas mostrados en este sitio son propiedad de sus respectivos propietarios y NO están bajo la licencia mencionada.

