




SQL: calculando

aritmética y funciones de agregados sobre un conjunto de filas



columnas resultado de cálculos



numpedido	linea	IVA
1	1	91.35
1	2	137.34
2	1	293.16
2	2	2282.48
2	3	238.14
2	4	225.75
3	1	229.95
3	2	409.29
3	3	75.60
3	4	17.85

```
select numpedido, linea,  
importe*cantidad*0.21 IVA  
from linped
```

```
select numpedido, linea,  
round(importe*cantidad*0.21,2) IVA  
from linped
```

SQL: calculando

En las consultas podemos mostrar cálculos más o menos complejos. Eso incluye el uso de funciones de todo tipo que ofrece el motor de la base de datos.


En el primer ejemplo, para cada línea de pedido obtenemos el IVA correspondiente al importe de ese artículo (21%) por la cantidad pedida. El segundo ejemplo obtiene lo mismo, pero la salida es redondeada a dos decimales mediante una función ofrecida por MariaDB: round().

T08.015- Precio de venta al público del artículo más caro (PVP más alto)

select max(pvp) from articulo

cod	nombre	pvp
A01	lápiz electrónico	100
A02	lápiz analógico	125
A05	goma borrar nuclear	200
A07	papel gordo	30
A09	papel fino	25
A10	papel arrugao	15

max(pvp)
200

SQL: calculando fundamentos de las bases de datos 

Un ejemplo simple. Queremos conocer un único dato, cual es el mayor PVP que tenemos almacenado en la tabla ARTÍCULO. Para eso, utilizamos otra función que obtiene el máximo de una expresión, en este caso una columna: MAX().

T08.015- Precio de venta al público del artículo más caro (PVP más alto)

1 from
¿qué tablas necesito?

2 where
filtra filas

3 select
calcula | muestra

select max(pvp) from articulo
where cod>'A05'

max(pvp)

30

cod	nombre	pvp
A01	lápiz electrónico	100
A02	lápiz analógico	125
A05	goma borrar nuclear	200
A07	papel gordo	30
A09	papel fino	25
A10	papel arrugao	15

SQL: calculando

fundamentos de las bases de datos

Cómo afecta el filtrado de filas del *where*. El filtrado se hace antes de realizar el cálculo.

Como ya hemos dicho otras veces, debemos pensar —aunque no sea necesariamente cierto en la práctica— que *from* es lo primero que se ejecuta y, por tanto, partimos de una tabla completa o de un producto cartesiano entre varias tablas. Aquí solo necesitamos la tabla ARTÍCULO. A continuación, se "ejecutaría" *where*, es decir, se filtran por alguna condición las filas obtenidas en el paso anterior. Y, por último, se muestra la información solicitada en *select*.

En este ejemplo, modificamos la consulta anterior añadiendo una condición en *where*: queremos el máximo PVP, pero no tenemos en cuenta códigos de artículo anteriores a 'A06'. Dicho de otra forma, sobre los datos mostrados en la diapositiva, el máximo sólo se calcula sobre el PVP de los artículos A07, A09 y A10, los únicos que cumplen la condición en este momento y en esta base de datos.

T08.015- Importe máximo, mínimo y medio de las líneas de pedido que incluyen el artículo "Bravia KDL-32EX402"

```
from linped l, articulo a
where l.articulo=a.cod
```

```
linped (
  numPedido int(11),
  linea int(11),
  articulo varchar(7),
  importe decimal(9,2),
  cantidad int(11) )
CP (linea,numPedido)
CAj (articulo) -> articulo VNN
CAj (numPedido) -> pedido
VNN (importe)
```

```
articulo (
  cod varchar(7),
  nombre varchar(45),
  pvp decimal(7,2),
  marca varchar(15),
  imagen blob,
  urlimagen varchar(100),
  especificaciones text)
CP (cod)
CAj (marca) -> marca
```

SQL: calculando

fundamentos de las bases de datos



Pasamos a otro ejemplo. Los artículos se pueden pedir varias veces en LINPED. Y cada vez que aparecen en una fila de esa tabla pueden tener un importe de venta diferente, e incluso distinto al PVP en ARTÍCULO. Piensa que el catálogo de artículos (ARTÍCULO) es el de hoy, pero los pedidos pueden ser del mes pasado, y se vendieron más baratos porque el precio ha subido.

En cualquier caso, nos piden importes máximo, mínimo y promedio de las líneas de pedido donde aparece un determinado artículo. Fíjate que ya sabemos una tabla que debemos utilizar: LINPED. El criterio de búsqueda que nos exigen es, en realidad, un nombre de artículo; también necesitamos ARTÍCULO. Estas dos tablas tienen un enlace directo por la clave ajena alojada en LINPED (linped.articulo). Así, lo primero que hacemos es montar el *from* y enlazar las tablas en *where*.

T08.015- Importe máximo, mínimo y medio de las líneas de pedido que incluyen el artículo “Bravia KDL-32EX402”

from lnped l, articulo a
where l.articulo=a.cod and a.nombre = 'Bravia KDL-32EX402'

Resultado #1 (12x2)								
numPedido	linea	articulo	importe	pvp	marca	imagen	urlimagen	especificaciones
6	1	A0777	317.00	421.00	Sony	(NULL)	4945p.jpg	(NULL)
40	1	A0777	349.00	421.00	Sony	(NULL)	4945p.jpg	(NULL)

importe

317.00

349.00

máximo	349,00
mínimo	317,00
promedio	$(349+317)/2 = 333,00$

SQL: calculando

fundamentos de las bases de datos

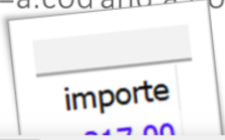


Ahora añadimos la condición que nos han pedido. Si examinas LINPED verás que solo 2 filas son de artículos que se llamen así, solo 2 valores de importe serán utilizados en los cálculos: 317,00€ y 349,00€. Recuerda: lnped.importe no es lo mismo que articulo.pvp.

Lo que esperamos es obtener 349,00€, 317€ y 333,00€ respectivamente.

T08.015- Importe máximo, mínimo y medio de las líneas de pedido que incluyen el artículo "Bravia KDL-32EX402"

```
select max(importe) max,min(importe) min,avg(importe) med
from linped l, articulo a
where l.articulo=a.cod and a.nombre = 'Bravia KDL-32EX402'
```



Resultado #1 (3x1)			
max	min	med	349,00
349.00	317.00	333.000000	317,00
promedio			$(349+317)/2 = 333,00$

SQL: calculando

fundamentos de las bases de datos



Para eso utilizamos funciones —de agregación, se llaman estas— que nos lo calculan: MAX(), MIN() y AVG() (de *average*: promedio o media).

T08.023- Código, nombre y marca del artículo más caro.

from **articulo**

```
articulo (  
  cod varchar(7),  
  nombre varchar(45),  
  pvp decimal(7,2),  
  marca varchar(15),  
  imagen blob,  
  urlimagen varchar(100),  
  especificaciones text)  
CP (cod)  
CAj (marca) -> marca
```

❶ from
¿qué tablas necesito?
❷ where
filtra filas
❸ select
calcula | muestra

SQL: calculando

fundamentos de las bases de datos



Otro ejemplo. En este caso preguntamos sobre ARTICULO. Recuerda, primero *from*, después *where* y, por último, *select*.

T08.023- Código, nombre y marca del artículo más caro.

```
from articulo  
where pvp = max(pvp)
```

```
articulo (  
  cod varchar(7),  
  nombre varchar(45),  
  pvp decimal(7,2),  
  marca varchar(15),  
  imagen blob,  
  urlimagen varchar(100),  
  especificaciones text)  
CP (cod)  
CAj (marca) -> marca
```

❶ from
¿qué tablas necesito?
❷ where
filtra filas
❸ select
calcula | muestra

SQL: calculando

fundamentos de las bases de datos



Aquí el problema es la condición "artículo más caro". Probemos esa expresión.

T08.023- Código, nombre y marca del artículo más caro.

```
select cod,nombre,marca
from articulo
where pvp = max(pvp)
```

articulo (
cod varchar(7),
nombre varchar(45),
 (7, 2)
 especifica
 CP (cod)
 CAj (marca) -> marca

1 from
 ¿qué tablas necesito?
2 where
 filtra filas
3 select
 calcula | muestra

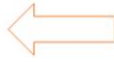
/* Error de SQL (1111): Invalid use of group function */

SQL: calculando fundamentos de las bases de datos

No!!! MariaDB no sabe qué hacer con eso. Max(pvp) tiene sentido cuando se comparan varios valores —el máximo de un conjunto de valores, por ejemplo, los de la una columna de una tabla—. MAX() pertenece a lo que se llaman funciones de agregados, y deben usarse junto con otro componente de la consulta *select*: *group-by*.

T08.023- Código, nombre y marca del artículo más caro.

```
select cod,nombre,marca  
from articulo  
where pvp = x
```



where = comparación
en cada fila

y=x → espera valores
escalares **a izquierda y**
derecha

(1 columna, 1 fila) es
un valor escalar

SQL: calculando

fundamentos de las bases de datos



El problema es que la comparación del pvp espera a la derecha un valor escalar y espera encontrarlo en cada fila individualmente. Lo que queremos es también un valor escalar, pero calculado sobre toda la tabla, no en cada fila, hay que encontrarlo...

T08.023- Código, nombre y marca del artículo más caro.

```
select cod,nombre,marca
from articulo
where pvp = (select max(pvp) from articulo)
```

↑
(1 columna, 1 fila)


← subconsulta
independiente
compleja como
se quiera

SQL: calculando fundamentos de las bases de datos

...¿por qué no obtener ese valor de otra consulta? Solo hay que tener cuidado en devolver una única fila y una única columna. Y aquí se da ese hecho, el valor máximo del PVP es un único valor. La consulta general recorrerá fila a fila ARTICULO, y en todas ellas comparará el valor almacenado de PVP con el máximo calculado previamente. Puedes pensar que esta subconsulta es lo primero que se ejecuta, se obtiene el valor máximo, y este se guarda en una "variable" para ser utilizado en todas las comparaciones que se generen —tantas como filas tenga ARTICULO—.

T08.023- Código, nombre y marca del artículo más caro.

```
select cod,nombre,marca  
from articulo  
where pvp = (select max(pvp) from articulo)
```



Resultado #1 (1x1)	
max(pvp)	6450.41

SQL: calculando

fundamentos de las bases de datos



El motor de base de datos ejecutará primero la subconsulta, obtendrá el valor...

T08.C22

select
from
where

articulo (4x1.291)

cod	nombre	marca	pvp
A0215	D300s	Nikon	1330.00
A0216	D5000	Nikon	489.00
A0217	Coolpix L19	Nikon	(NULL)
A0218	Coolpix L20	Nikon	(NULL)
A0219	Coolpix L100	Nikon	(NULL)
A0220	Coolpix S220	Nikon	99.00
A0221	Coolpix S230	Nikon	(NULL)
A0222	Coolpix S620	Nikon	(NULL)
A0223	Coolpix P90	Nikon	(NULL)
A0224	Coolpix S630	Nikon	6450.41
A0225	D3X	Nikon	671.00
A0226	D90	Nikon	(NULL)
A0227	Coolpix P6000	Nikon	(NULL)
A0228	Coolpix S60	Nikon	(NULL)
A0229	Coolpix S710	Nikon	(NULL)
A0230	Coolpix S610c	Nikon	(NULL)
A0231	Coolpix S610	Nikon	(NULL)
A0232	Coolpix S560	Nikon	(NULL)

Resultado #1 (1x1)

max(pvp)

6450.41


SQL: calculando

fundamentos de las bases de datos

... y lo utilizará para buscarlo entre los valores de la columna PVP de ARTICULO.

T08.023- Código, nombre y marca del artículo más caro.

```
select cod,nombre,marca
from articulo
where pvp = (select max(pvp) from articulo)
```



The screenshot shows a database interface. On the left, a table named 'articulo (3x1)' is displayed with the following data:

cod	nombre	marca
A0225	D3X	Nikon

On the right, a query result window titled 'Resultado #1 (1x1)' shows the result of the SQL query. It displays the value 'max(pvp)' as '6450.41'. An arrow points from this value to the 'pvp' column of the 'articulo' table, highlighting the row with 'A0225' and 'D3X'.

SQL: calculando

fundamentos de las bases de datos



Recordemos que nuestro objetivo era encontrar todos los datos de ese o esos artículos con el PVP máximo. Obviamente, depende del estado de la tabla, de los datos que contenga, el que me salga solo un artículo o varios. El caso es que

1. encontramos ese valor máximo
2. y después lo comparamos con todas las filas para encontrar los artículos deseados.

licencias



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

