# VISTAS

# Ejercicio

*-- últimos 5 productos publicados con los siguientes campos:*

*-- id, nombre producto, nombre categoria, precio, descuento, precio con descuento, nombre usuario*

**SELECT**

p.id,

p.nombre **AS** 'producto',

c.nombre **AS** 'categoria',

u.nombre **AS** 'usuario',

precio,

descuento,

**ROUND** ((precio-precio\*descuento/100), 2) **AS** 'precio\_con\_descuento'

**FROM**

producto p, categoria c, usuario u

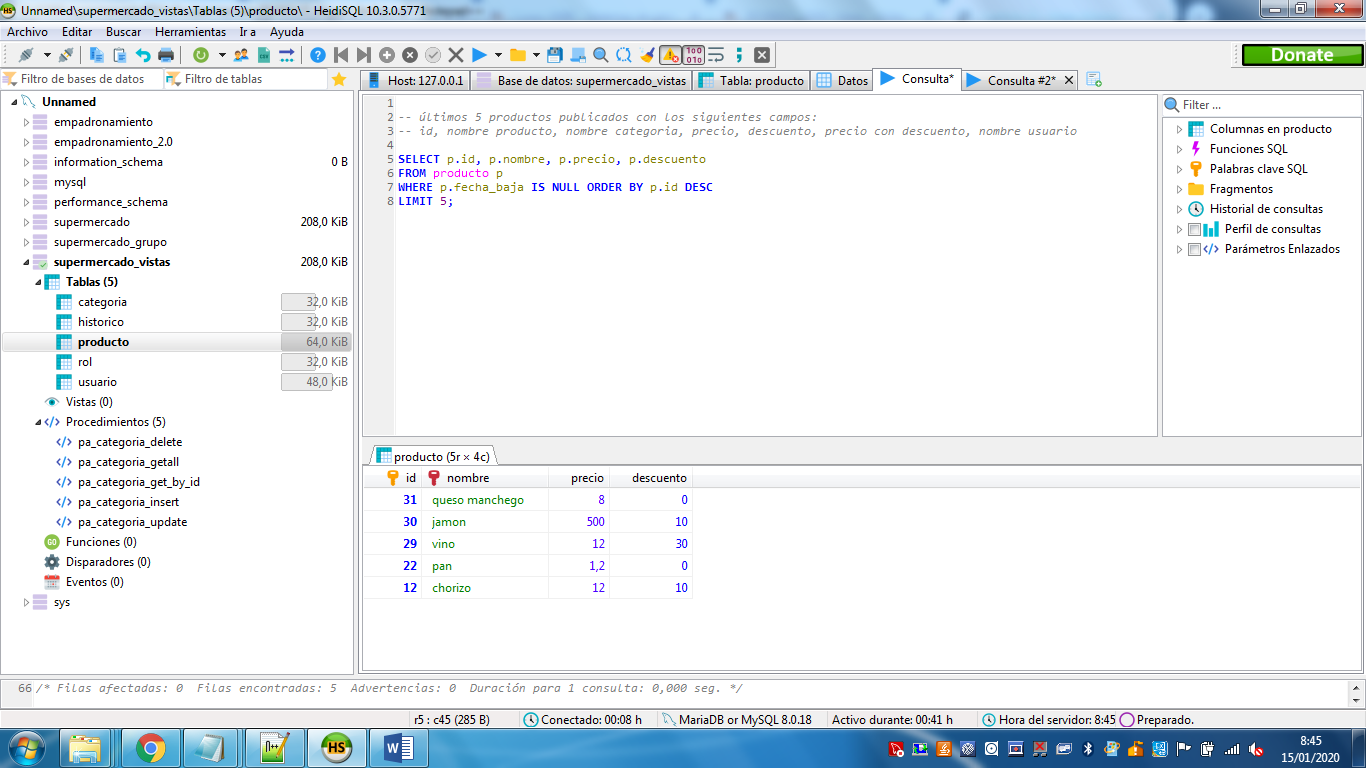
**WHERE**

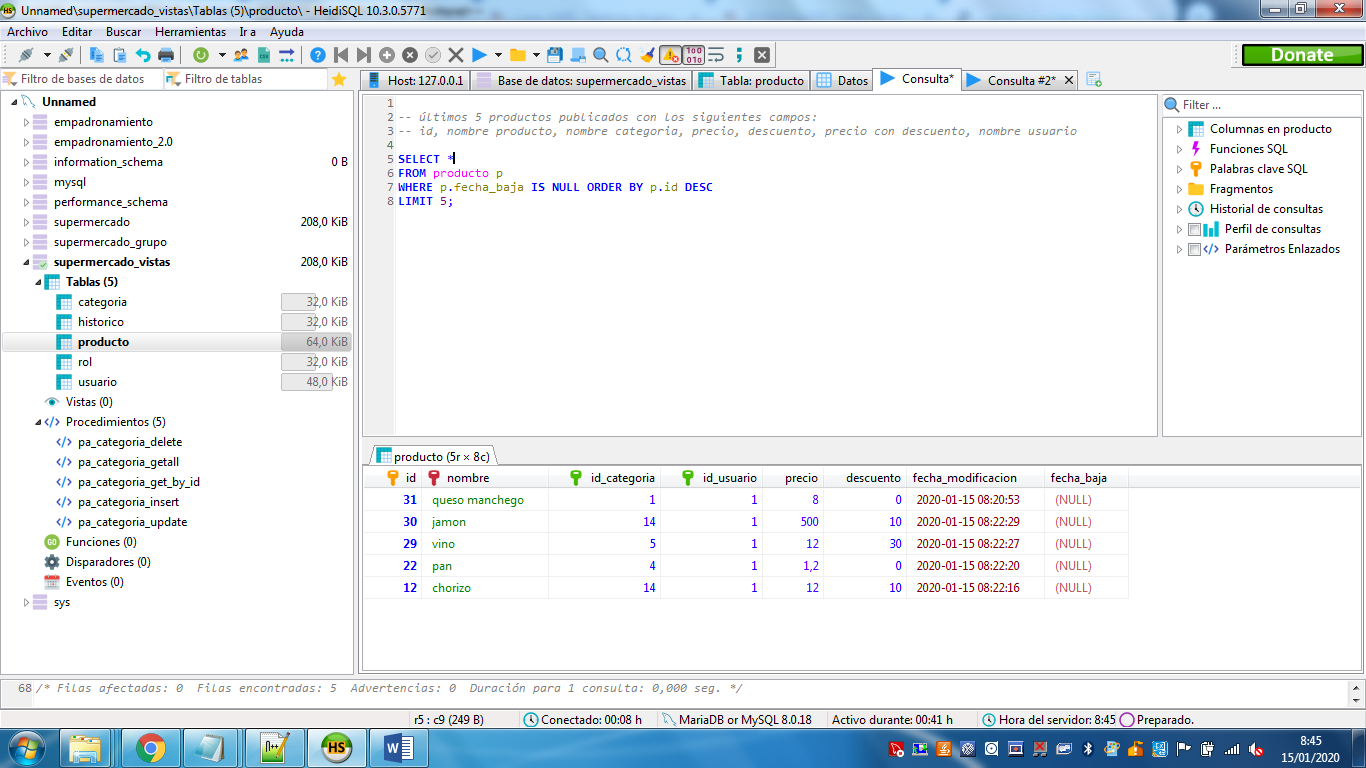
p.id\_categoria = c.id **AND**

p.id\_usuario = u.id **AND**

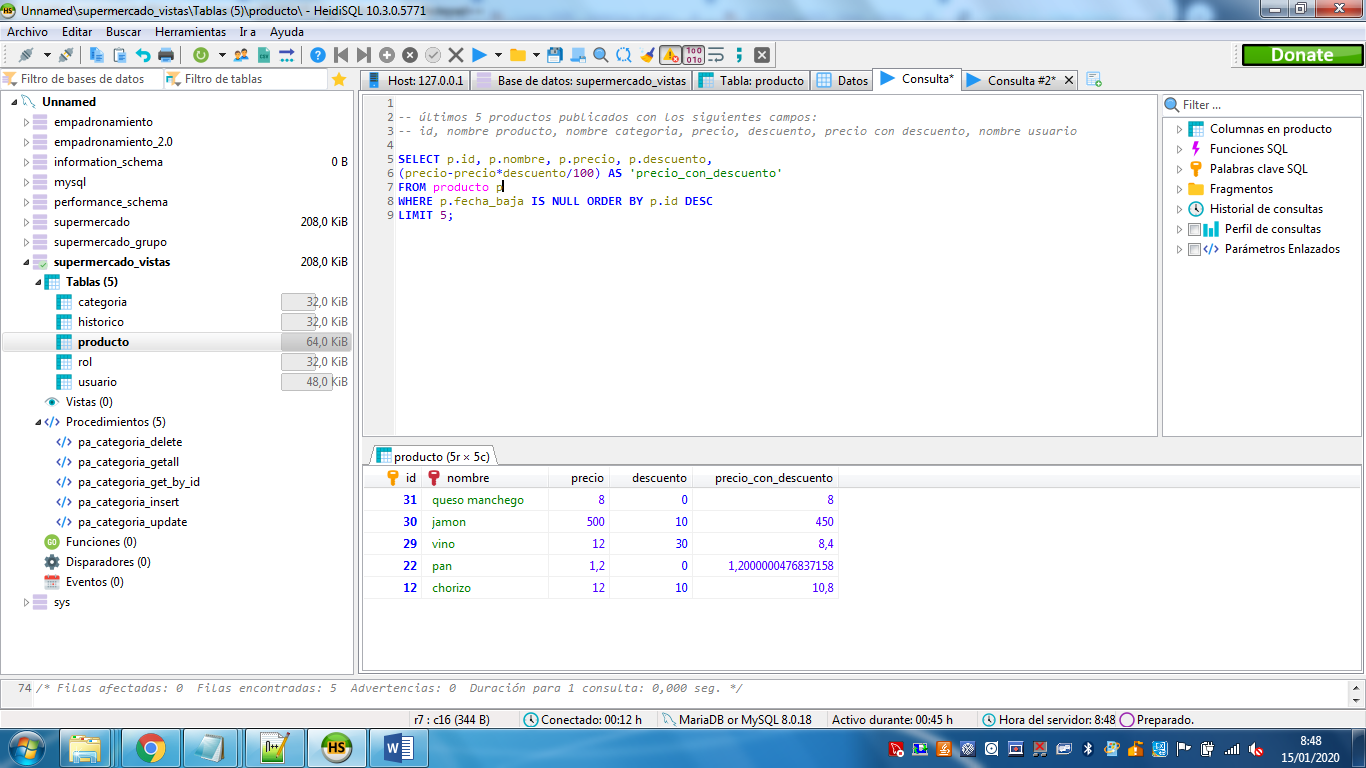
fecha\_baja **IS** **NULL** **ORDER** **BY** p.id **DESC**

**LIMIT** 5;

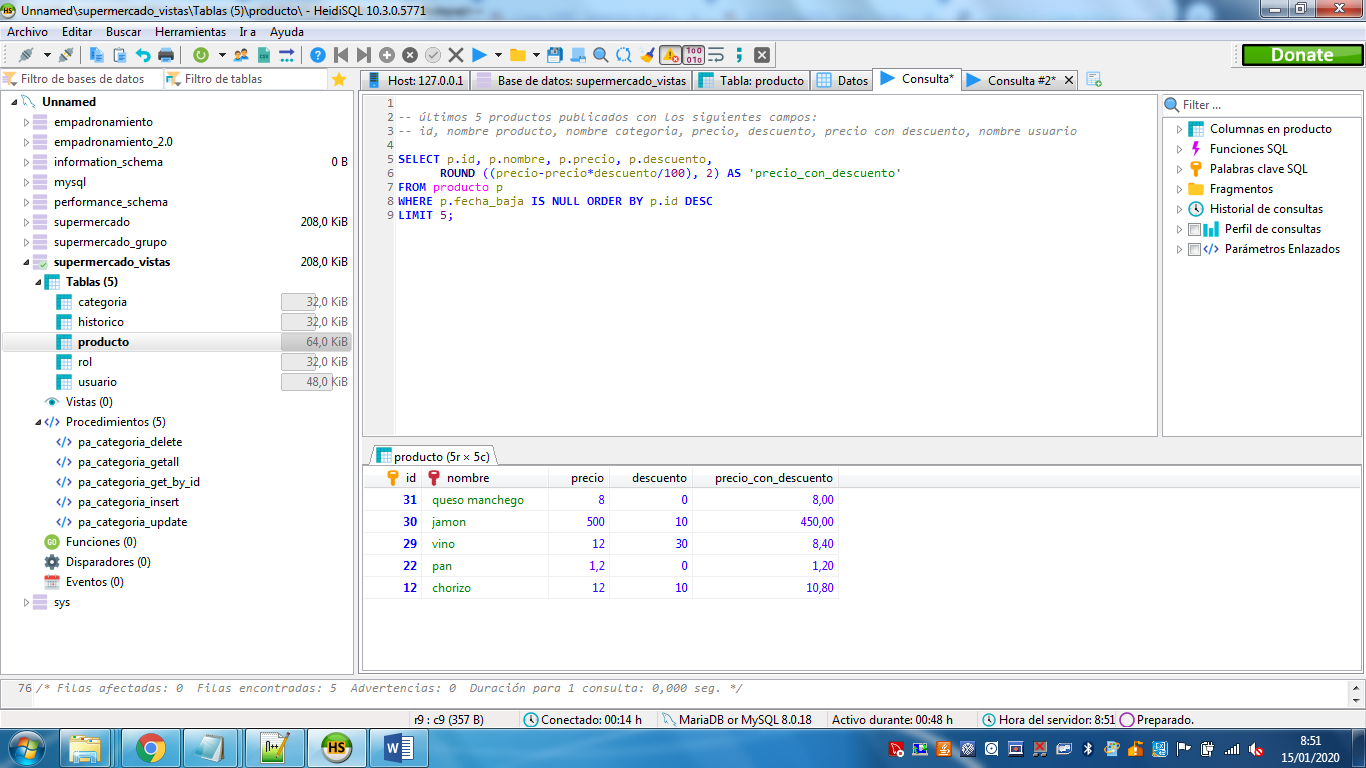




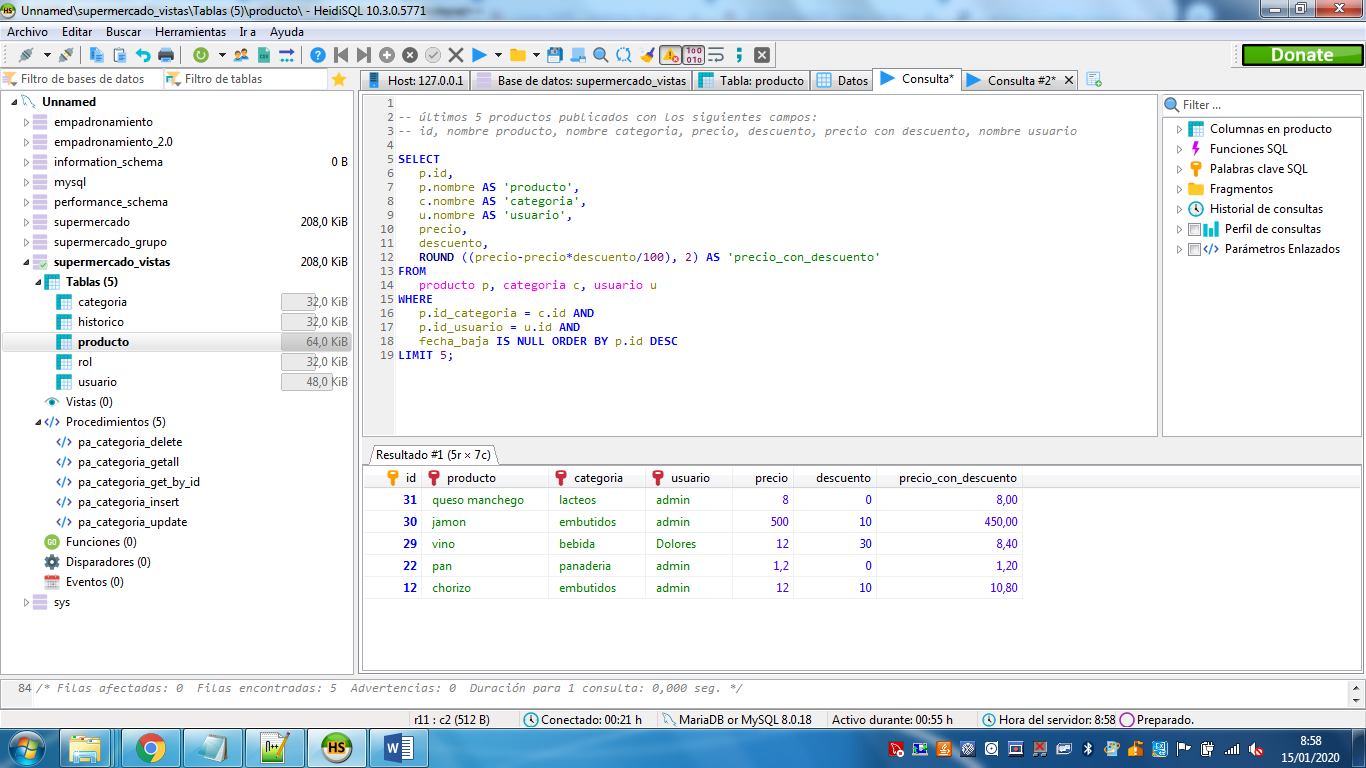




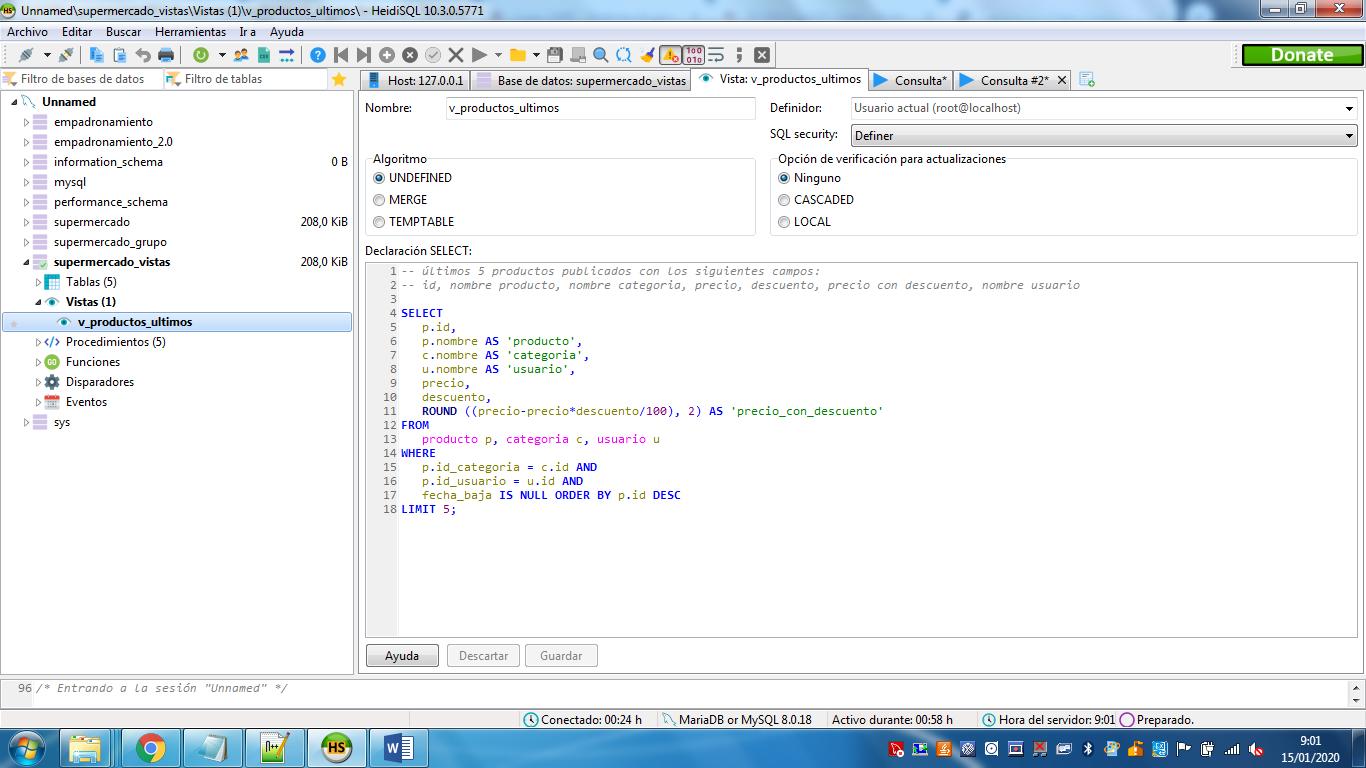
ROUND para redondear a los decimales que indiquemos en el 2º parámetro:



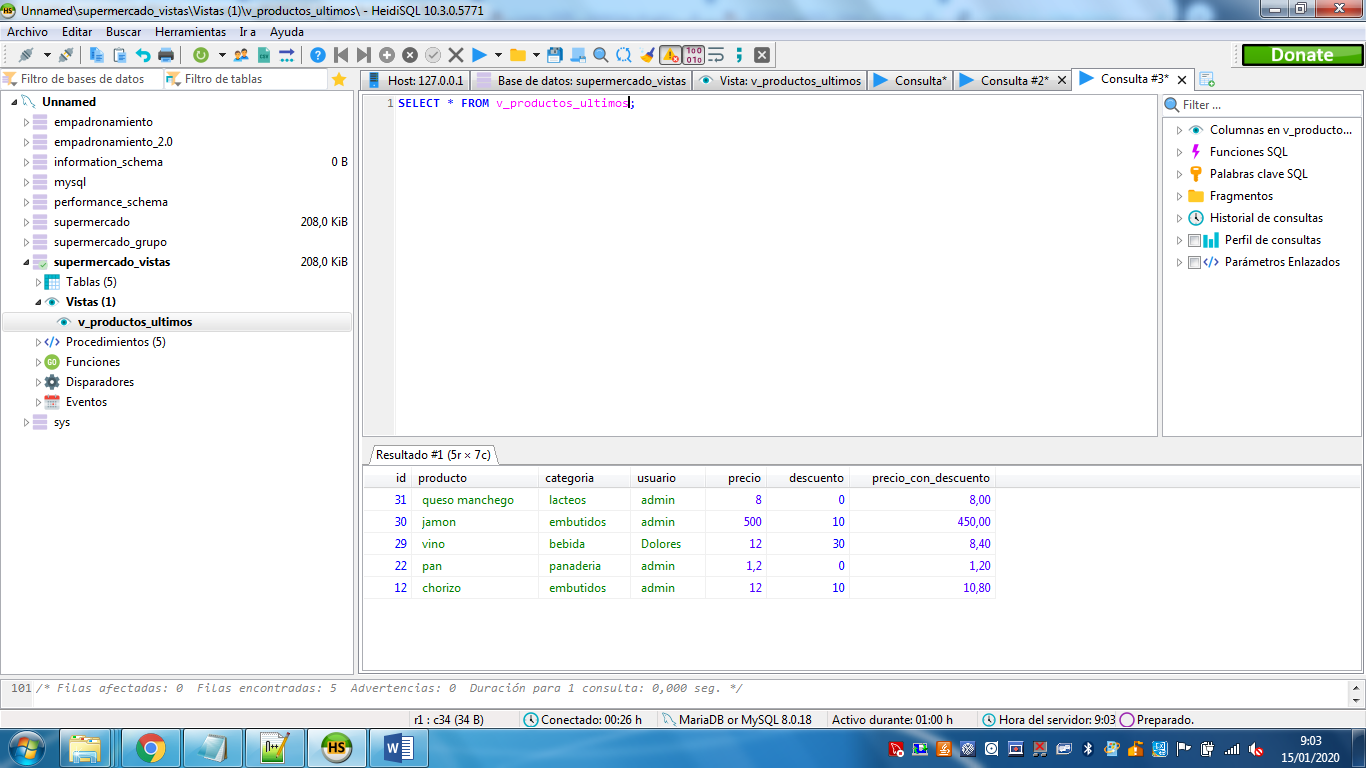
Usamos los alias para los nombres repetidos en distintas tablas:



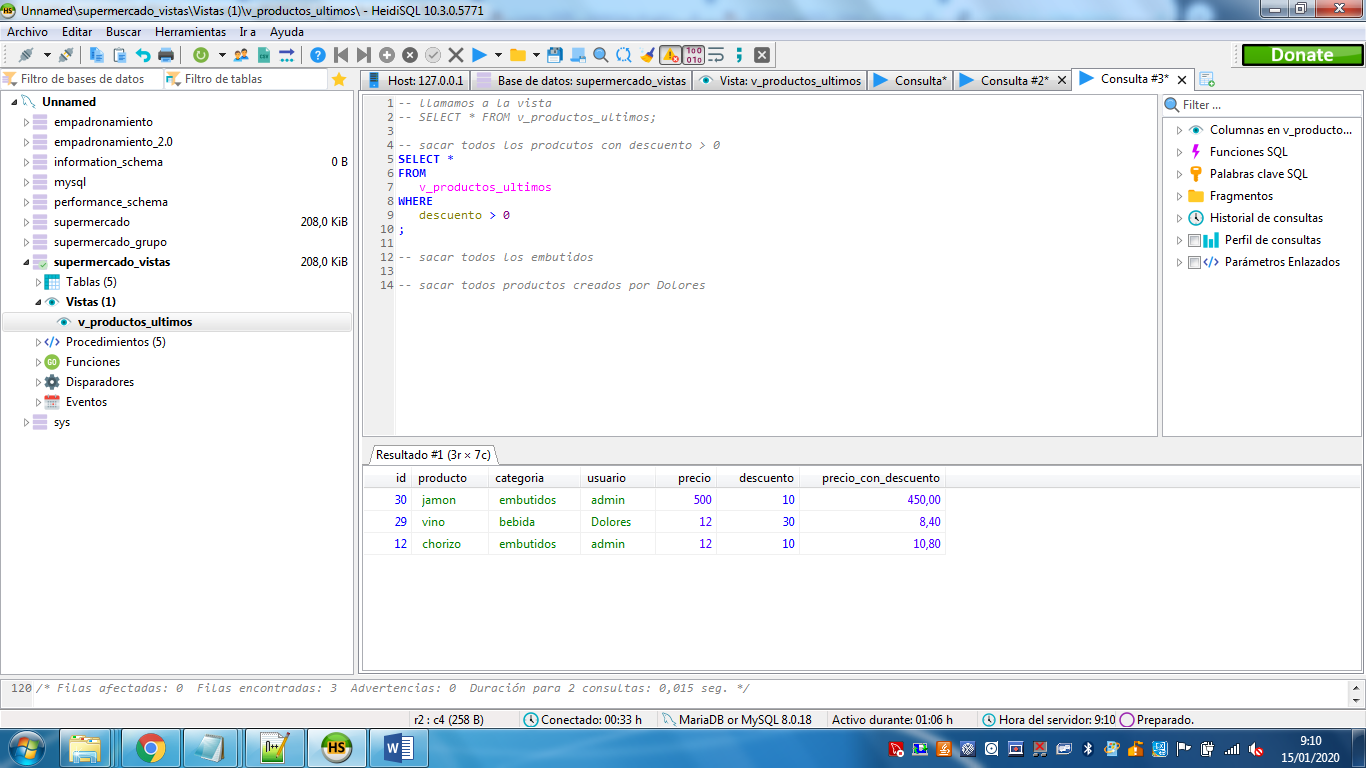
# 2.Pasando la consulta a una vista

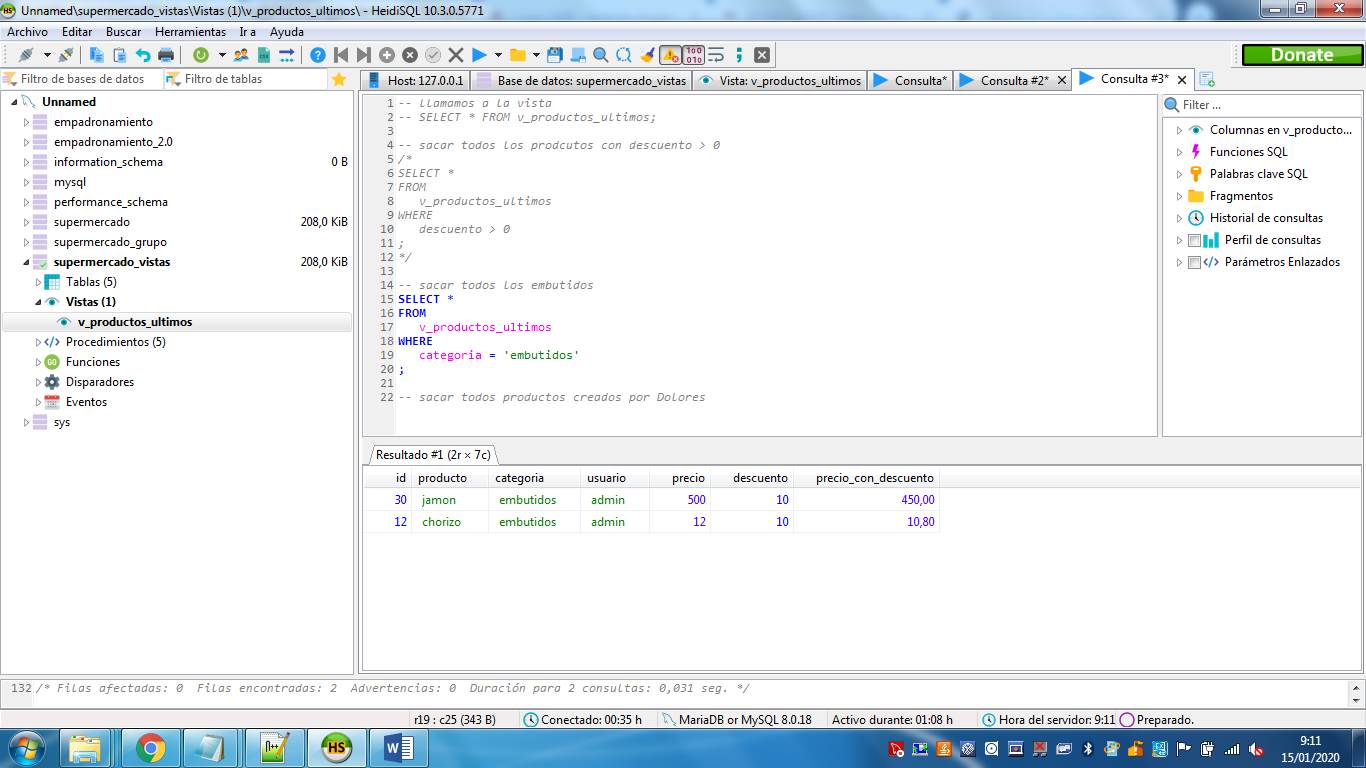


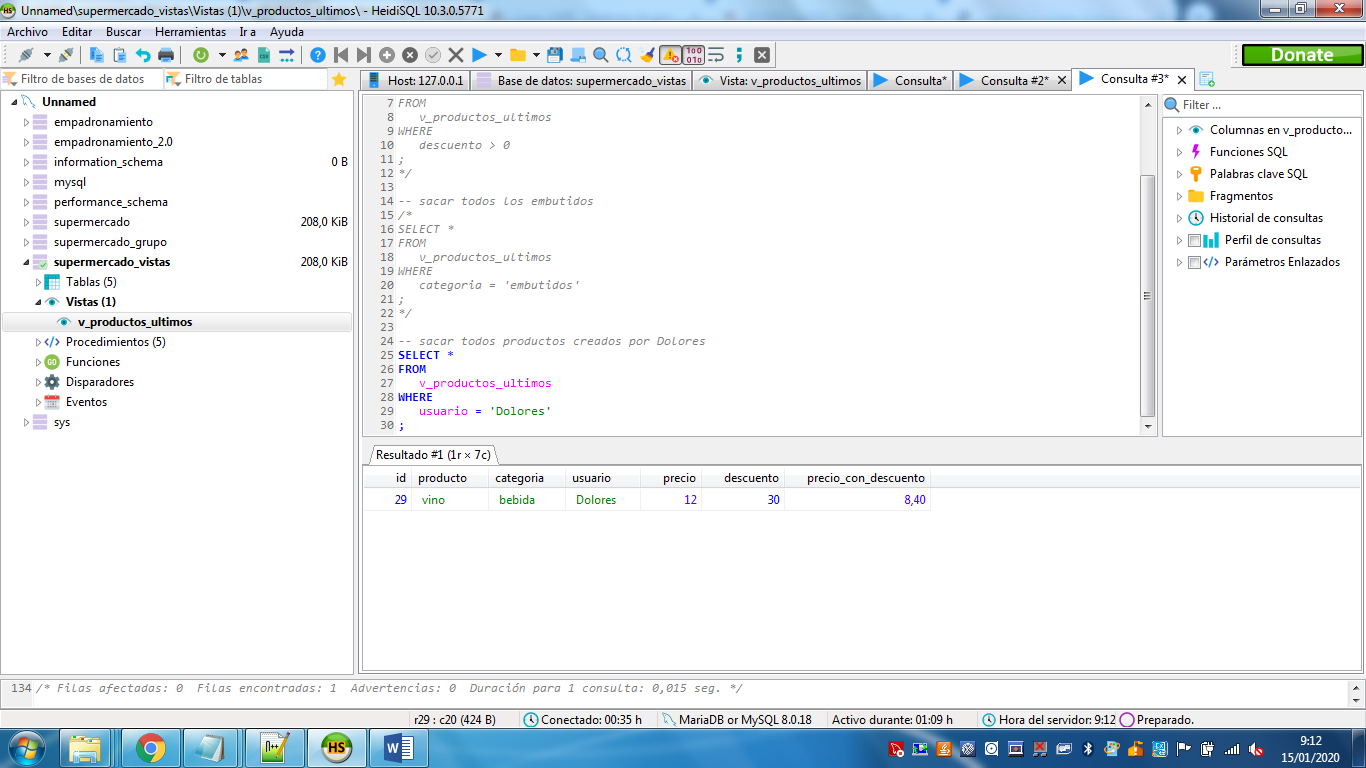
Llamamos a la consulta a través de la vista, es como hacer una tabla virtual



Hacemos 3 consultas a partir de la vista:







*-- llamamos a la vista*

*-- SELECT \* FROM v\_productos\_ultimos;*

*-- sacar todos los prodcutos con descuento > 0*

**SELECT** \*

**FROM**

v\_productos\_ultimos

**WHERE**

descuento > 0

;

*-- sacar todos los embutidos*

**SELECT** \*

**FROM**

v\_productos\_ultimos

**WHERE**

categoria = 'embutidos'

;

*-- sacar todos productos creados por Dolores*

**SELECT** \*

**FROM**

v\_productos\_ultimos

**WHERE**

usuario = 'Dolores'

;

# 3.Creamos 3 vistas:

## v\_productos\_baja

*-- lista de los prodcutos dados de baja con los siguientes campos:*

*-- id, nombre producto, nombre categoria, precio, descuento, precio con descuento, nombre usuario*

**SELECT**

p.id,

p.nombre **AS** 'producto',

c.nombre **AS** 'categoria',

u.nombre **AS** 'usuario',

precio,

descuento,

**ROUND** ((precio-precio\*descuento/100), 2) **AS** 'precio\_con\_descuento'

**FROM**

producto p, categoria c, usuario u

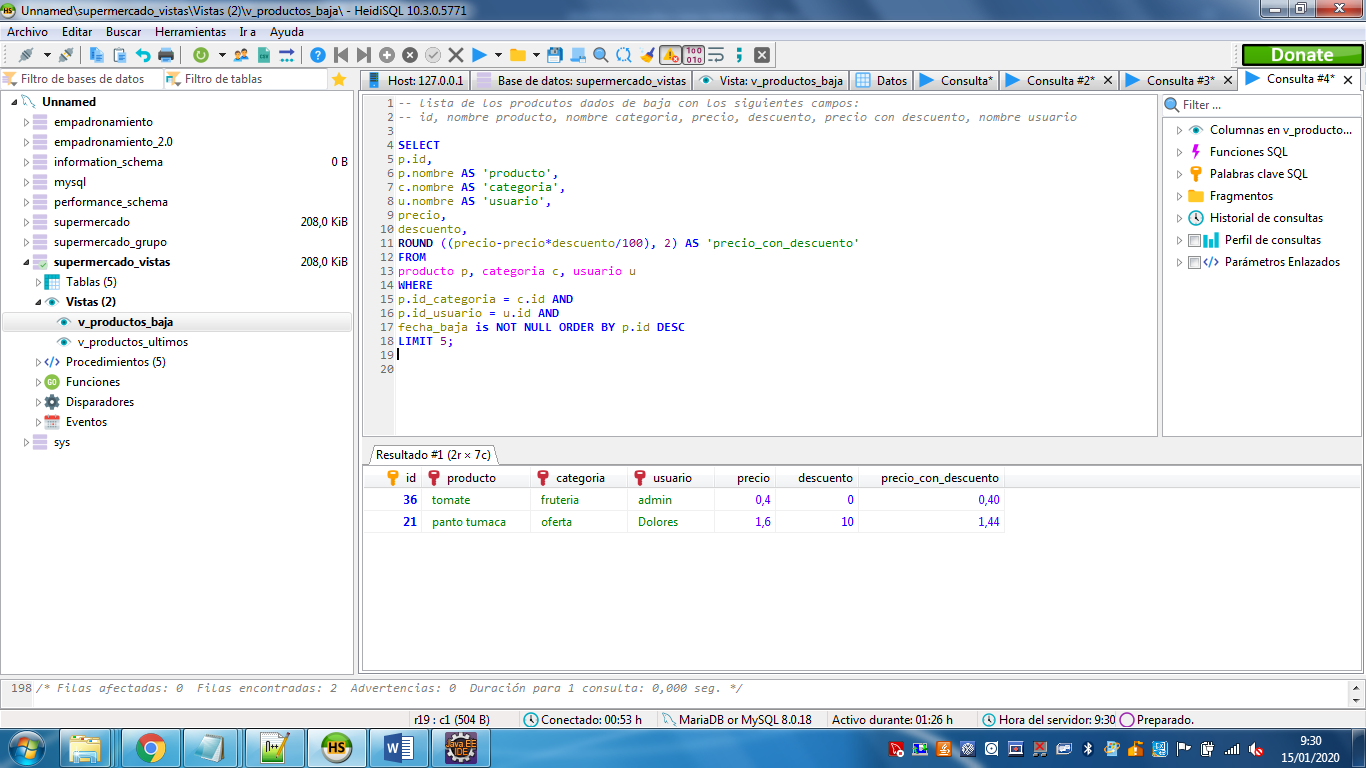
**WHERE**

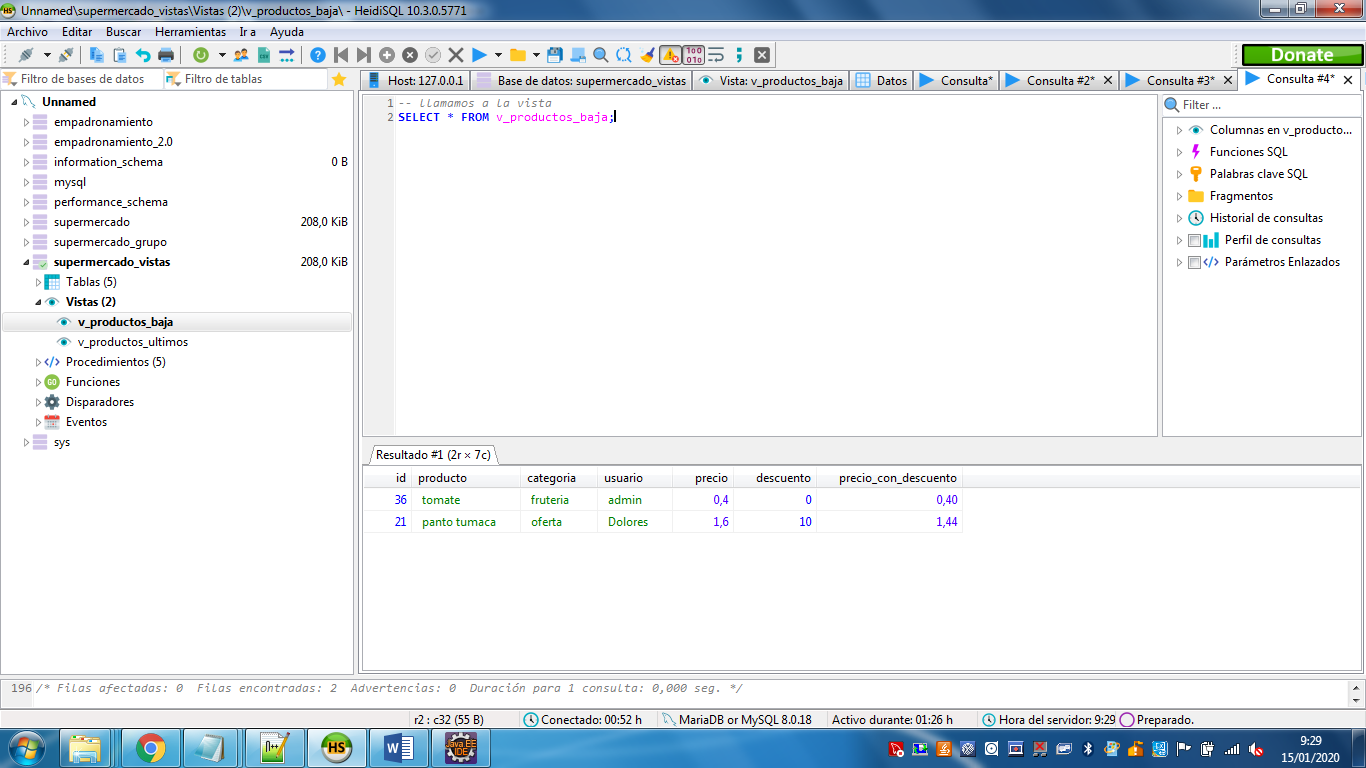
p.id\_categoria = c.id **AND**

p.id\_usuario = u.id **AND**

fecha\_baja **is** **NOT** **NULL** **ORDER** **BY** p.id **DESC**

**LIMIT** 5;





## v\_productos\_descuento

*-- lista de los prodcutos con descuento con los siguientes campos:*

*-- id, nombre producto, nombre categoria, precio, descuento, precio con descuento, nombre usuario*

**SELECT**

p.id,

p.nombre **AS** 'producto',

c.nombre **AS** 'categoria',

u.nombre **AS** 'usuario',

precio,

descuento,

**ROUND** ((precio-precio\*descuento/100), 2) **AS** 'precio\_con\_descuento'

**FROM**

producto p, categoria c, usuario u

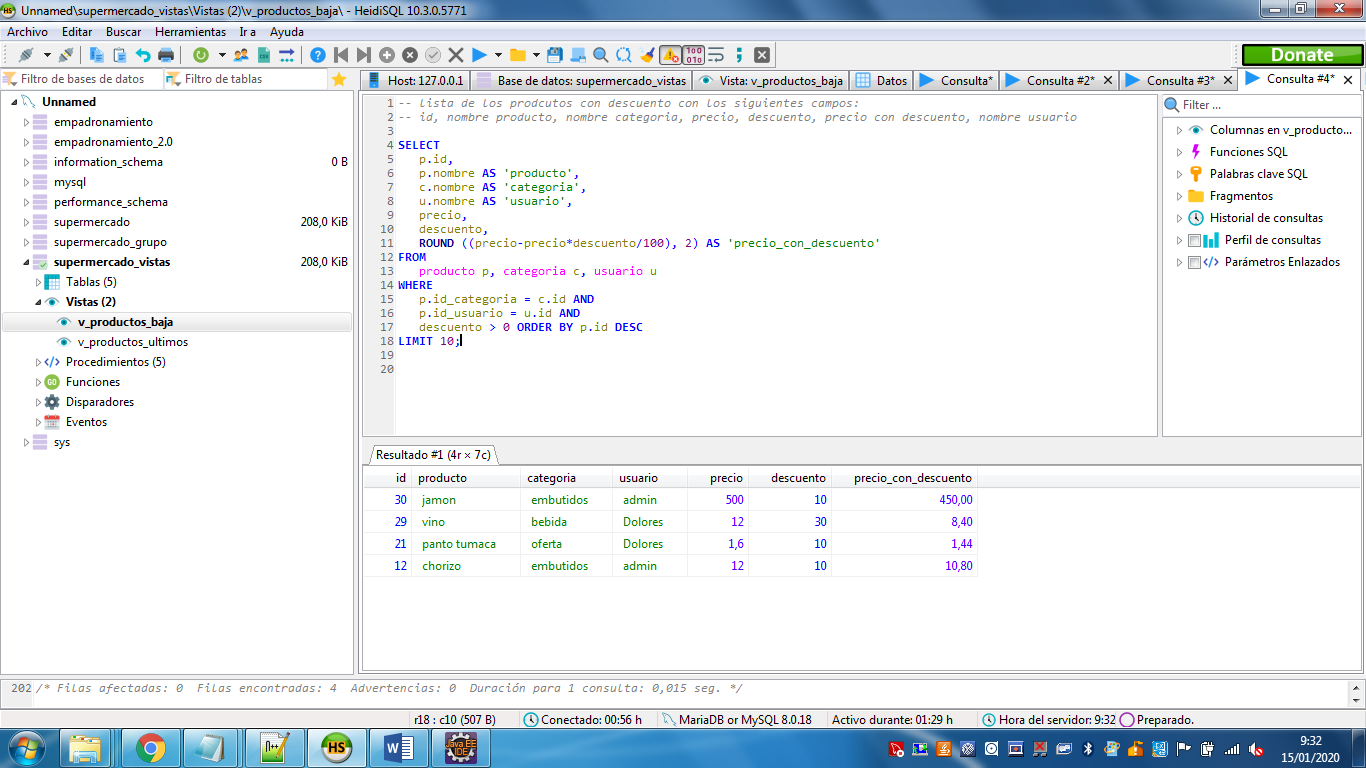
**WHERE**

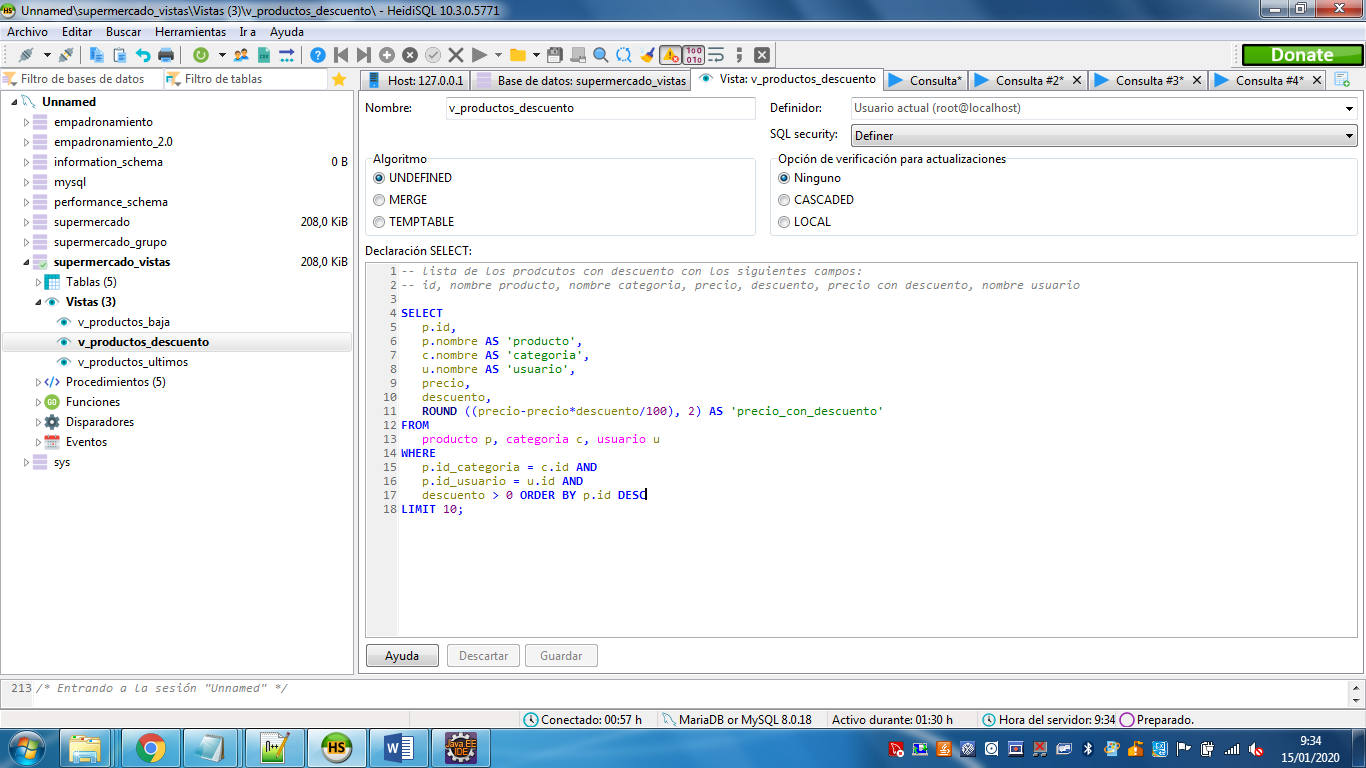
p.id\_categoria = c.id **AND**

p.id\_usuario = u.id **AND**

descuento > 0 **ORDER** **BY** p.id **DESC**

**LIMIT** 10;





## v\_productos\_activos

*-- lista de los prodcutos activos con los siguientes campos:*

*-- id, nombre producto, nombre categoria, precio, descuento, precio con descuento, nombre usuario*

**SELECT**

p.id,

p.nombre **AS** 'producto',

c.nombre **AS** 'categoria',

u.nombre **AS** 'usuario',

precio,

descuento,

**ROUND** ((precio-precio\*descuento/100), 2) **AS** 'precio\_con\_descuento'

**FROM**

producto p, categoria c, usuario u

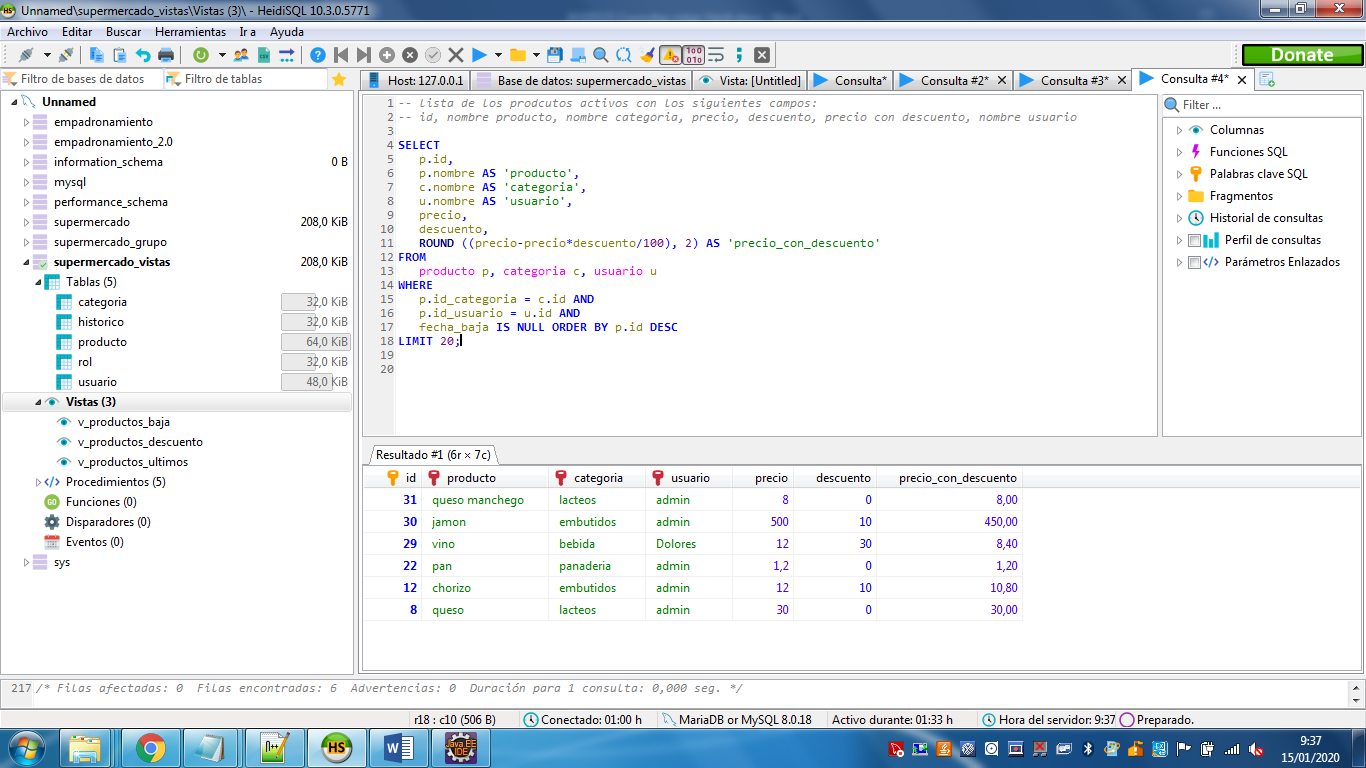
**WHERE**

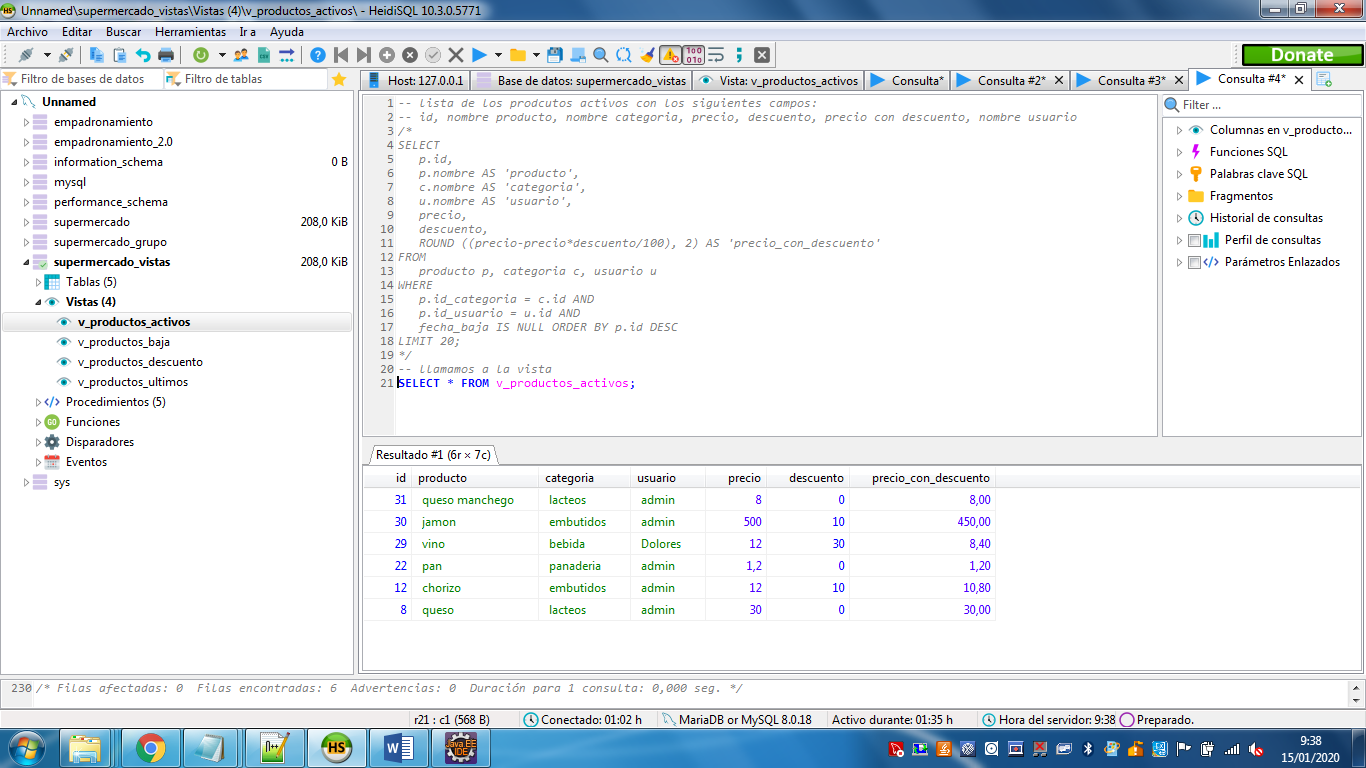
p.id\_categoria = c.id **AND**

p.id\_usuario = u.id **AND**

fecha\_baja **IS** **NULL** **ORDER** **BY** p.id **DESC**

**LIMIT** 20;





# 4. ALGORITHM, MERGE, TEMPTABLE, UNDEFINED:

Copiamos el mismo código en las vistas v\_merge y v\_temptable:

*-- últimos 5 productos publicados con los siguientes campos:*

*-- id, nombre producto, nombre categoria, precio, descuento, precio con descuento, nombre usuario*

**SELECT**

p.id,

p.nombre **AS** 'producto',

c.nombre **AS** 'categoria',

u.nombre **AS** 'usuario',

precio,

descuento,

**ROUND** ((precio-precio\*descuento/100), 2) **AS** 'precio\_con\_descuento'

**FROM**

producto p, categoria c, usuario u

**WHERE**

p.id\_categoria = c.id **AND**

p.id\_usuario = u.id **AND**

fecha\_baja **IS** **NULL** **ORDER** **BY** p.id **DESC**

**LIMIT** 5;

No se ven diferencias…

# Funciones de agregación:

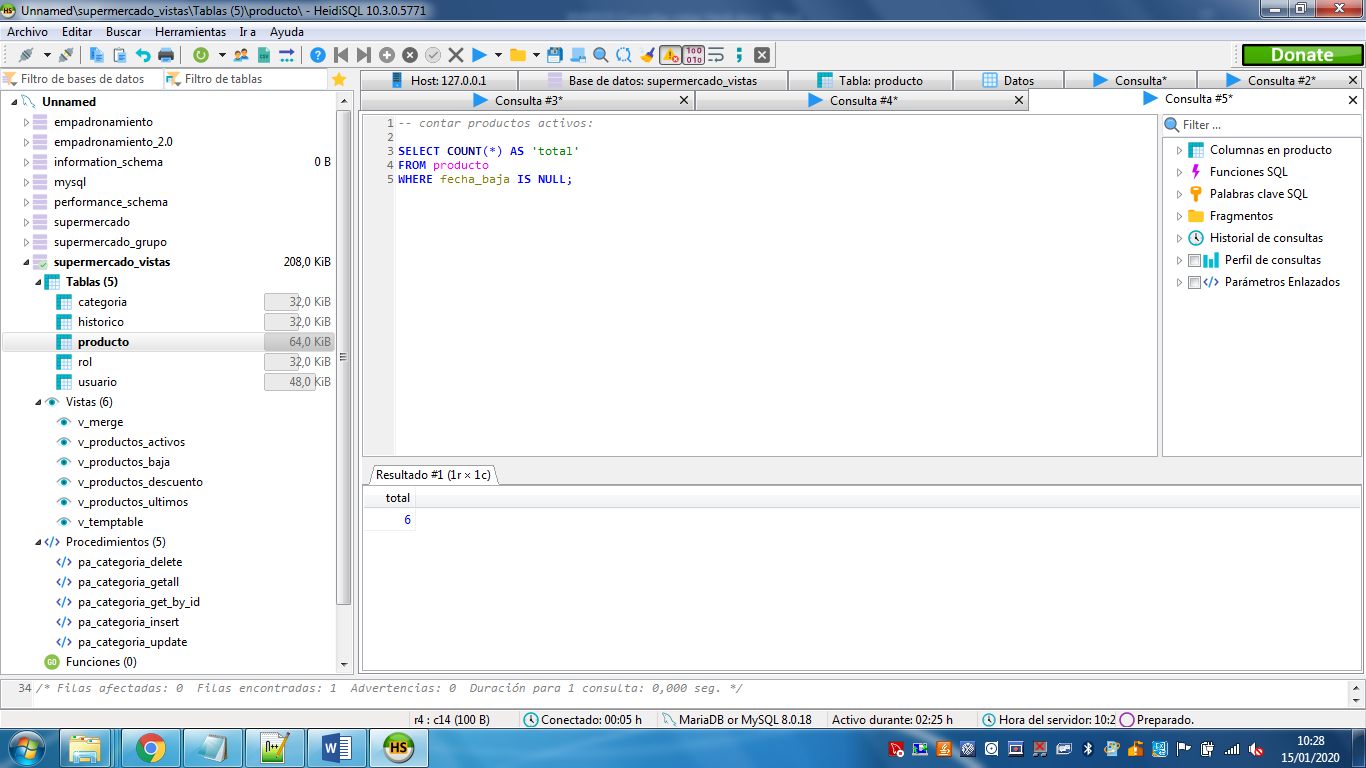
## COUNT

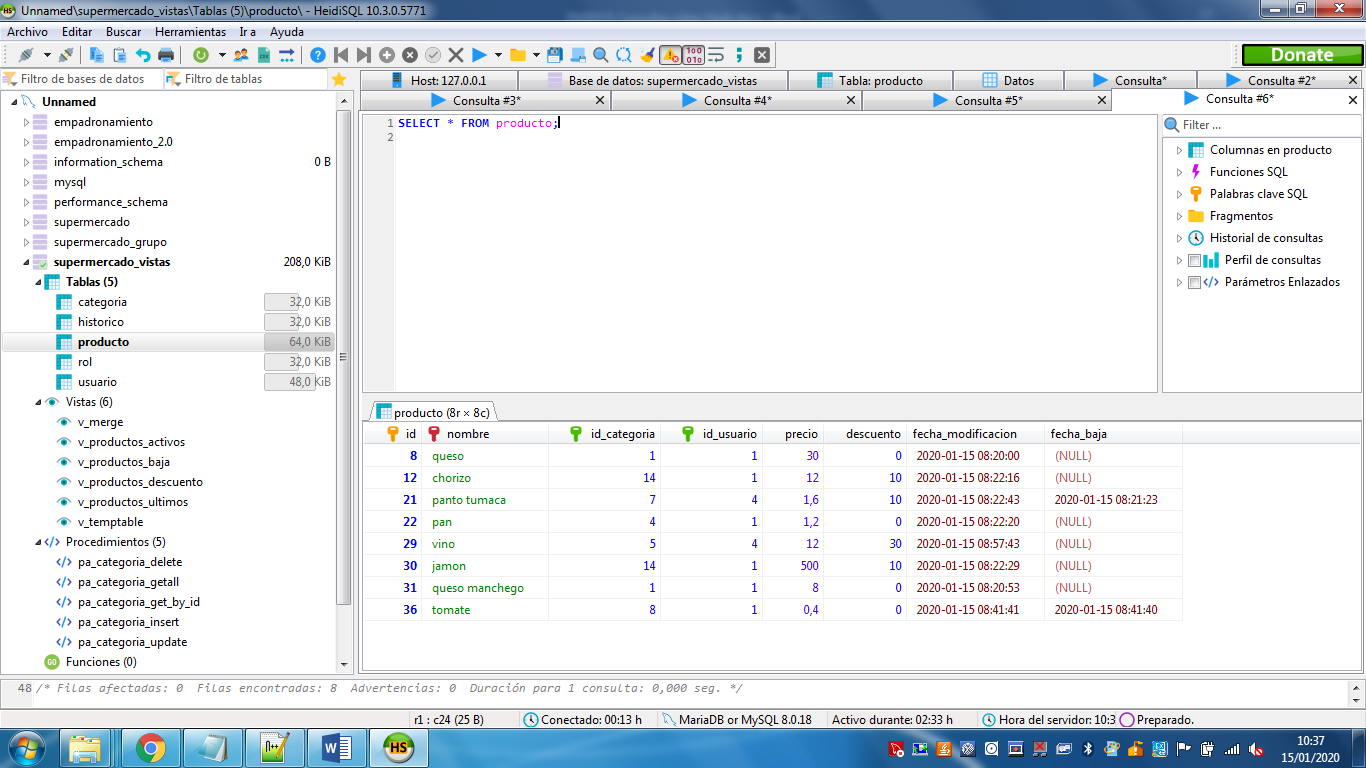
*-- contar productos activos:*

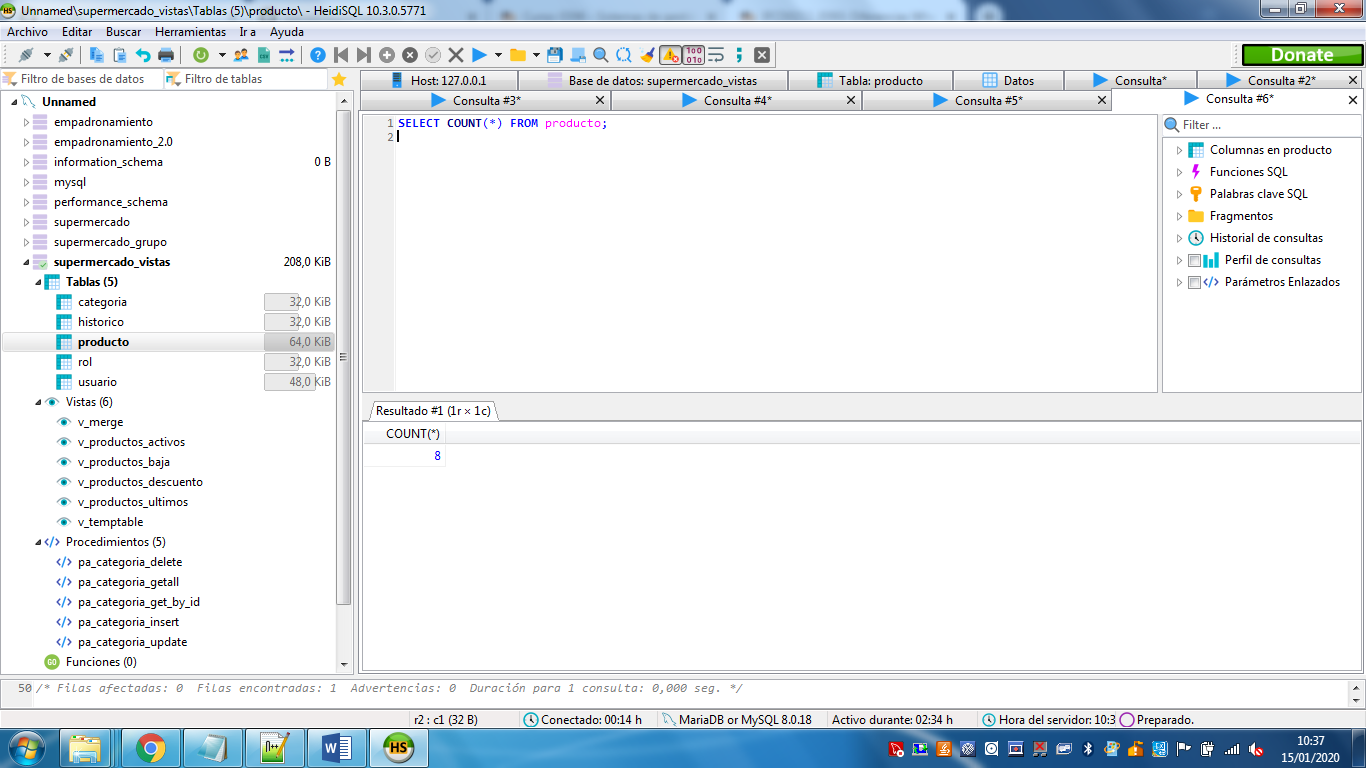
**SELECT** **COUNT**(\*) **AS** 'total'

**FROM** producto

**WHERE** fecha\_baja **IS** **NULL**;







## MIN, MAX, AVG

Son datos estadísticos

*-- contar productos activos, MIN, SUM total del precio de todos los productos, MAX y AVG:*

**SELECT**

**COUNT**(\*) **AS** 'total',

**ROUND**(**MIN**(precio), 3) **AS** 'precio\_mas\_bajo',

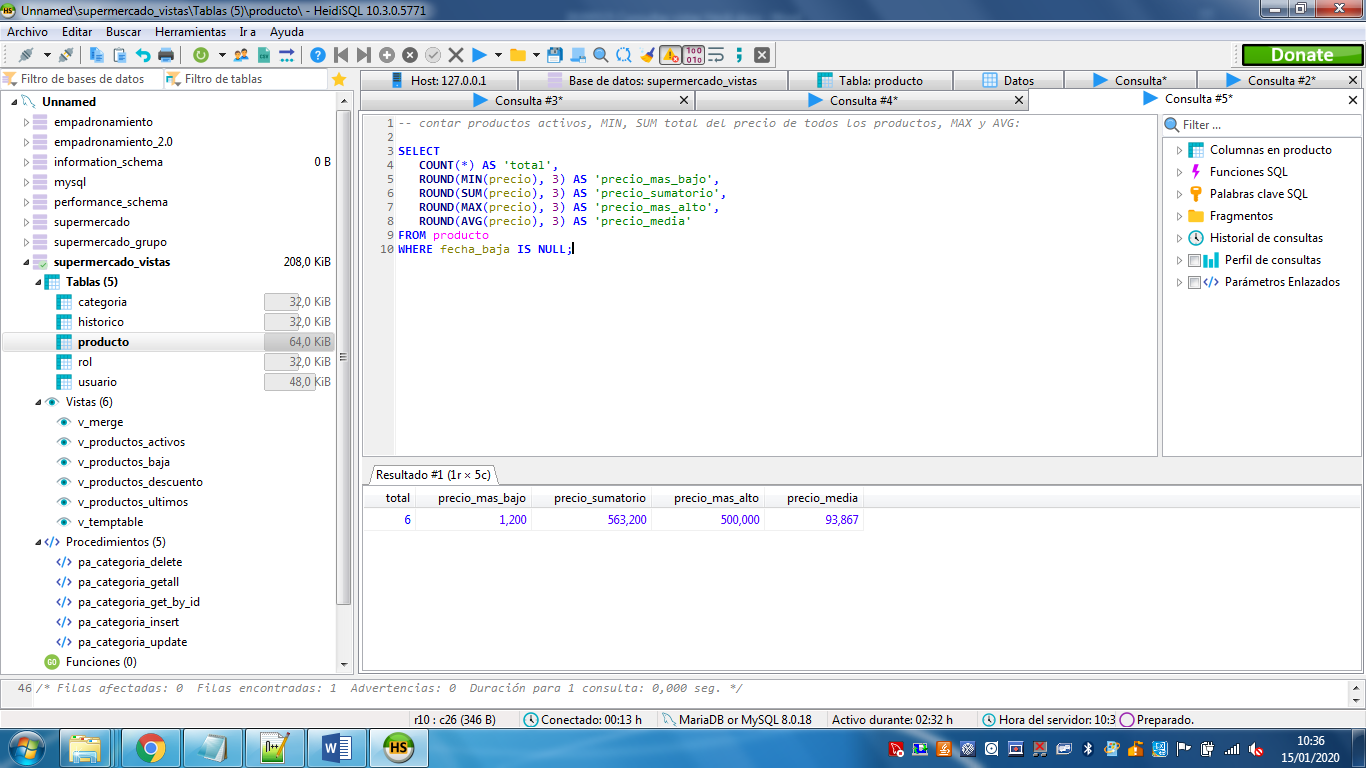
**ROUND**(**SUM**(precio), 3) **AS** 'precio\_sumatorio',

**ROUND**(**MAX**(precio), 3) **AS** 'precio\_mas\_alto',

**ROUND**(**AVG**(precio), 3) **AS** 'precio\_media'

**FROM** producto

**WHERE** fecha\_baja **IS** **NULL**;



## GROUP BY

**SELECT**

\*

**FROM** producto

**GROUP** **BY** id\_categoria;

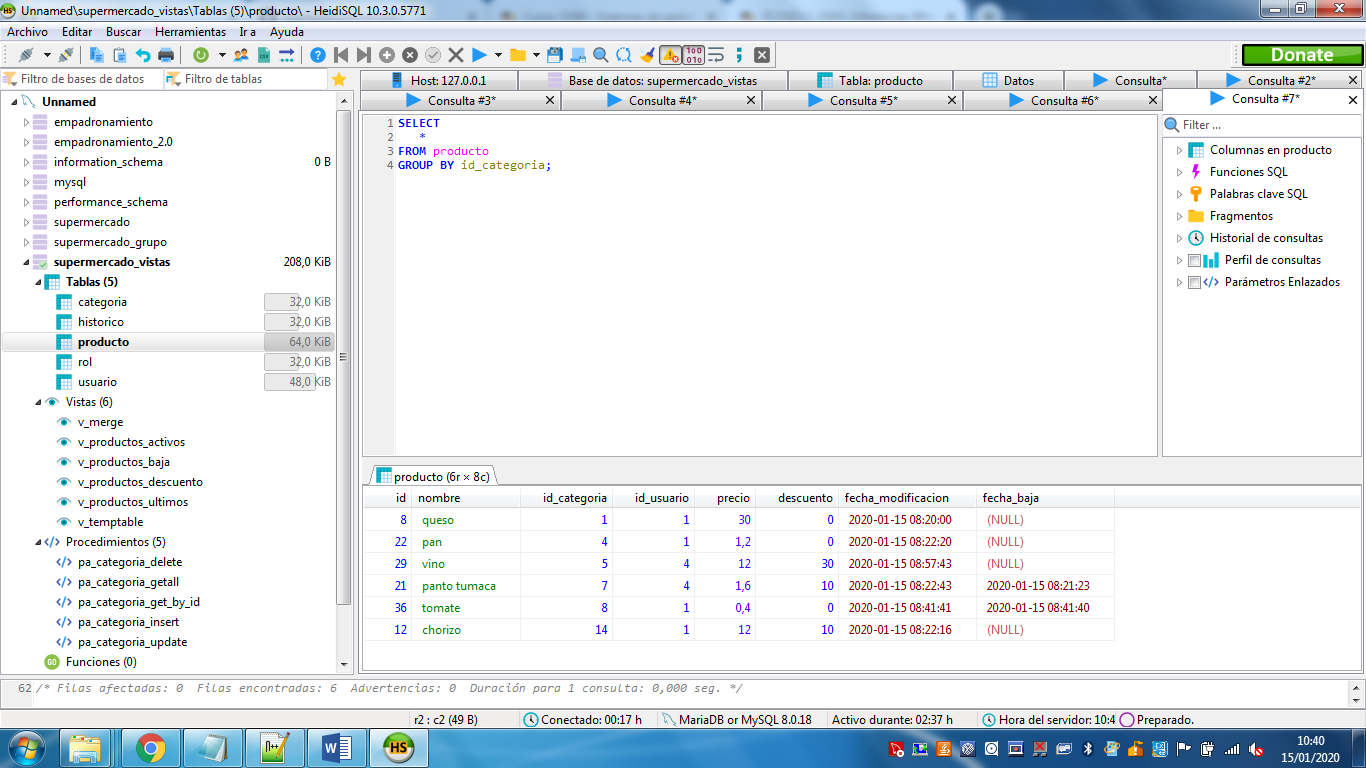
**SELECT**

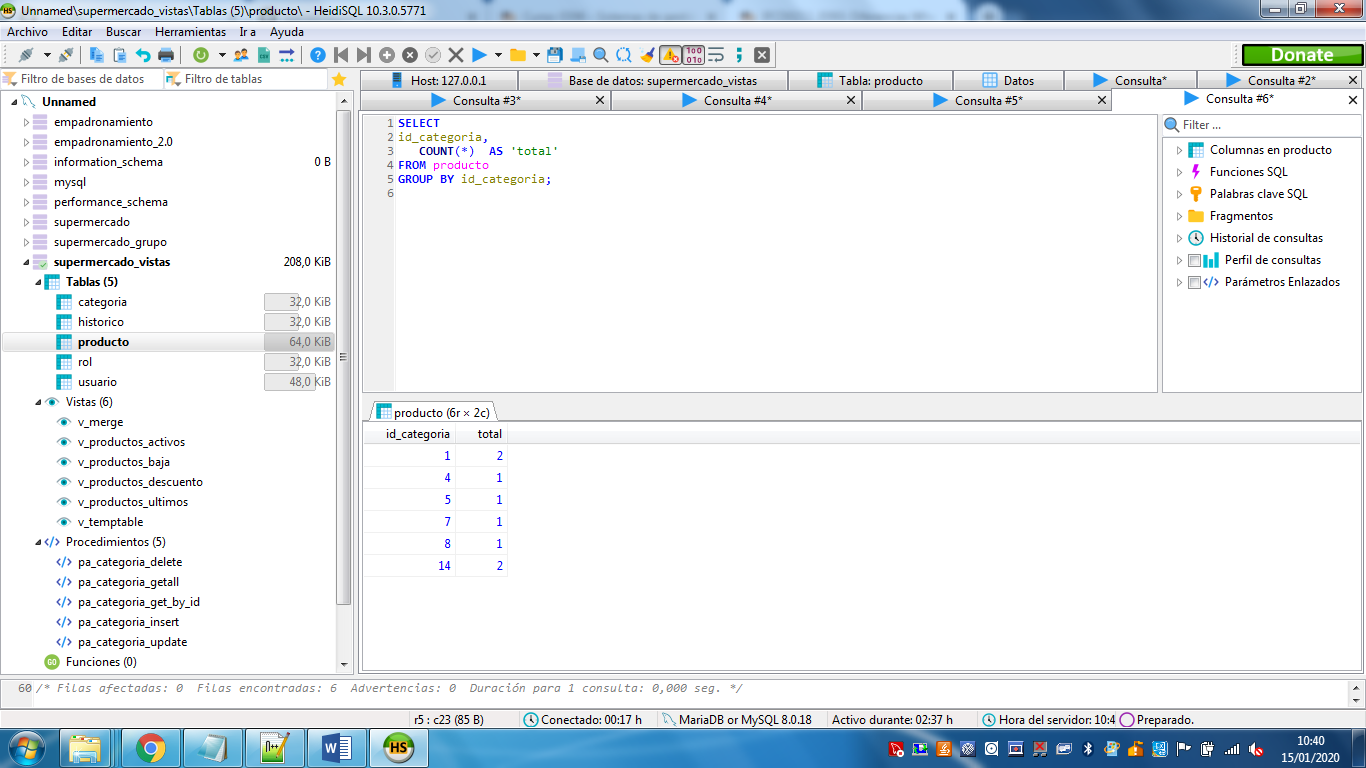
id\_categoria,

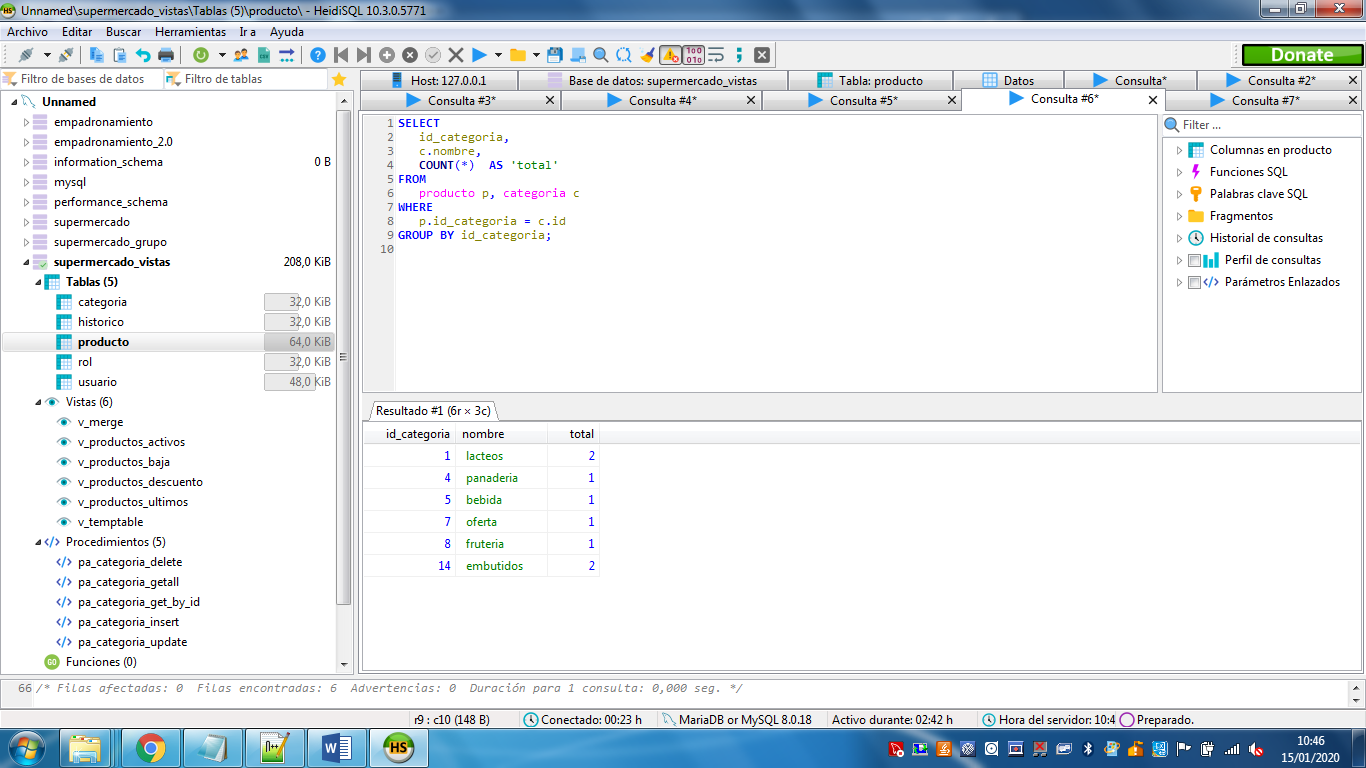
**COUNT**(\*) **AS** 'total'

**FROM** producto

**GROUP** **BY** id\_categoria;







**SELECT**

id\_categoria,

c.nombre,

**COUNT**(\*) **AS** 'total'

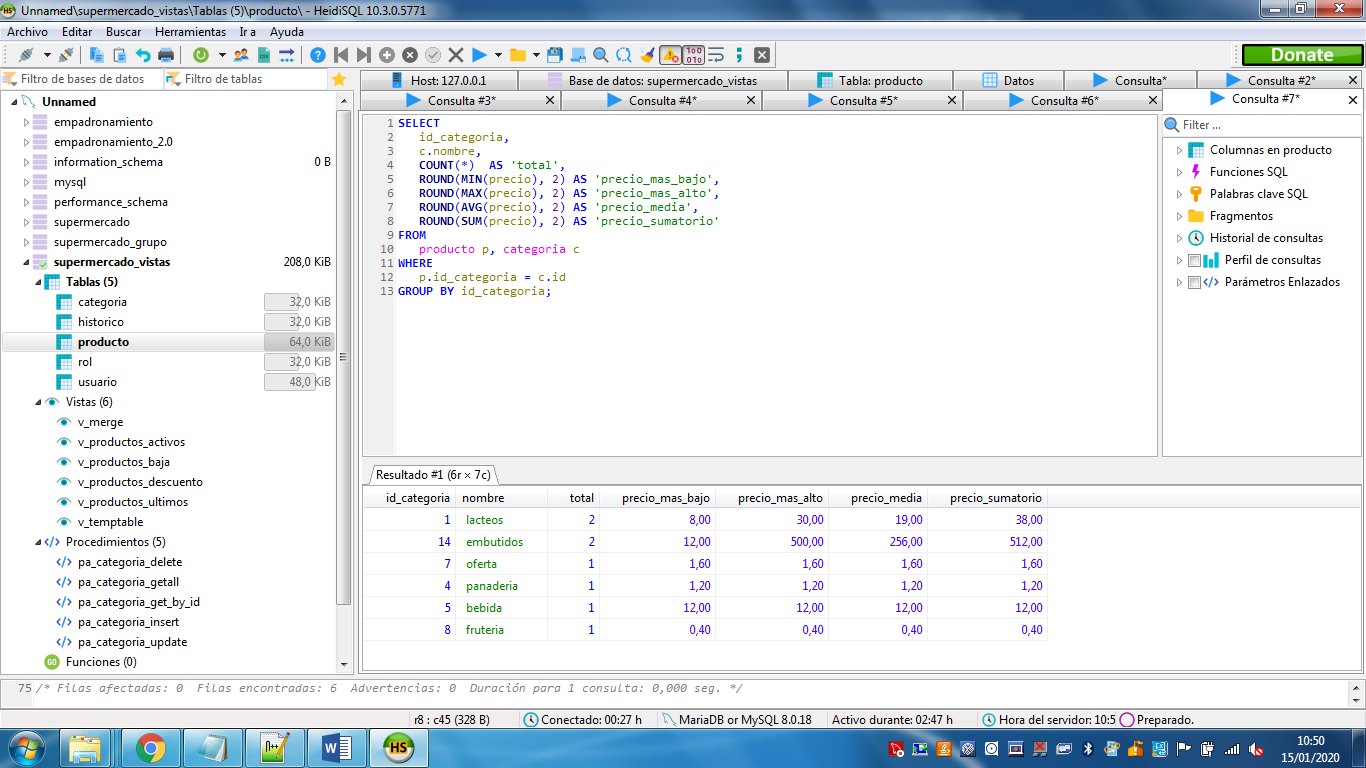
**FROM**

producto p, categoria c

**WHERE**

p.id\_categoria = c.id

**GROUP** **BY** id\_categoria;



**SELECT**

id\_categoria,

c.nombre,

**COUNT**(\*) **AS** 'total',

**ROUND**(**MIN**(precio), 2) **AS** 'precio\_mas\_bajo',

**ROUND**(**MAX**(precio), 2) **AS** 'precio\_mas\_alto',

**ROUND**(**AVG**(precio), 2) **AS** 'precio\_media',

**ROUND**(**SUM**(precio), 2) **AS** 'precio\_sumatorio'

**FROM**

producto p, categoria c

**WHERE**

p.id\_categoria = c.id

**GROUP** **BY** id\_categoria;

## WHERE y HAVING

**Where** filtra antes de hacer el select (la agrupación)

1.filtra sobre la tabla productos

2.luego hace el where

**SELECT**

id\_categoria,

c.nombre,

**COUNT**(\*) **AS** 'total',

**ROUND**(**MIN**(precio), 2) **AS** 'precio\_mas\_bajo',

**ROUND**(**MAX**(precio), 2) **AS** 'precio\_mas\_alto',

**ROUND**(**AVG**(precio), 2) **AS** 'precio\_media',

**ROUND**(**SUM**(precio), 2) **AS** 'precio\_sumatorio'

**FROM**

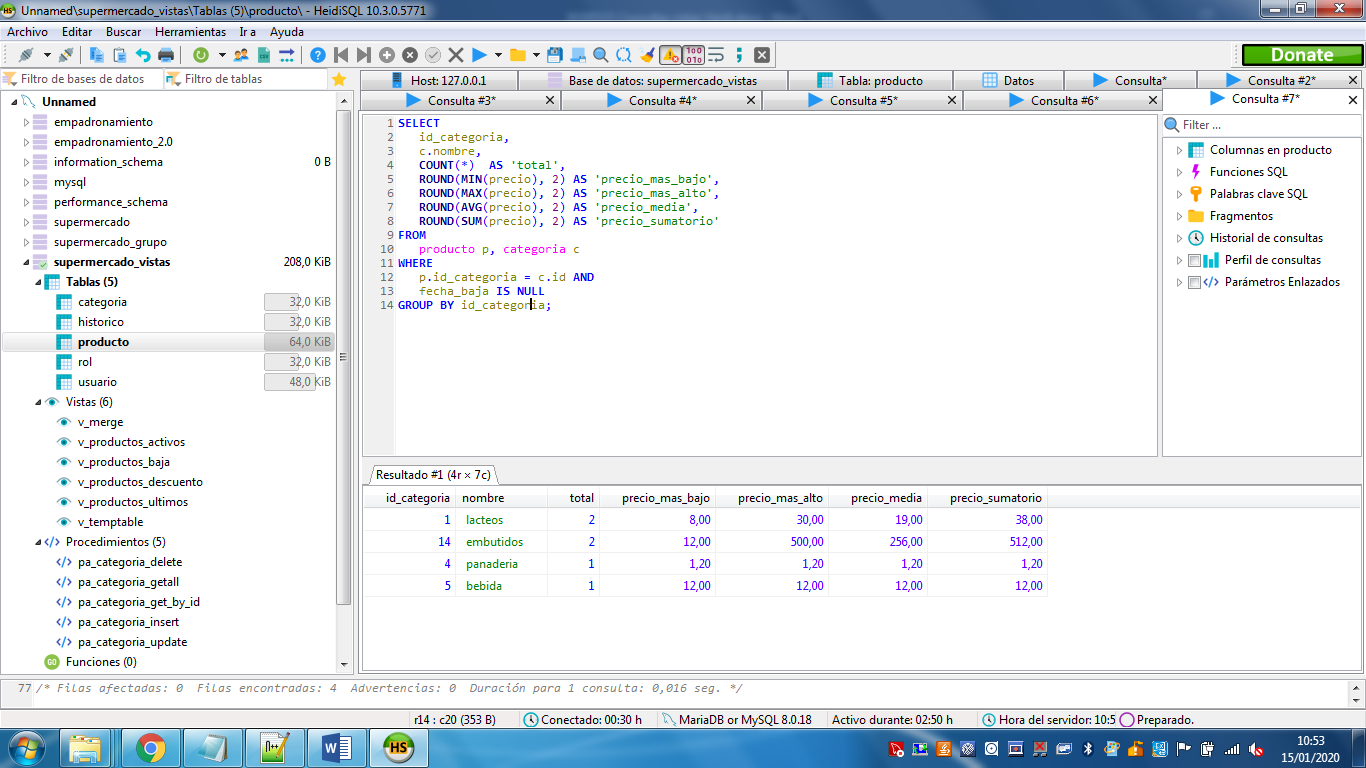
producto p, categoria c

**WHERE**

p.id\_categoria = c.id **AND**

fecha\_baja **IS** **NULL**

**GROUP** **BY** id\_categoria;



**Having** selecciona las filas después de que hayan sido agrupadas. Es una especie de where con las flas agrupadas

1.agrupa (quedaría la tabla anterior)

2.luego hace el filtro sobre esa tabla y tendríamos la siguiente tabla

**SELECT**

id\_categoria,

c.nombre,

**COUNT**(\*) **AS** 'total',

**ROUND**(**MIN**(precio), 2) **AS** 'precio\_mas\_bajo',

**ROUND**(**MAX**(precio), 2) **AS** 'precio\_mas\_alto',

**ROUND**(**AVG**(precio), 2) **AS** 'precio\_media',

**ROUND**(**SUM**(precio), 2) **AS** 'precio\_sumatorio'

**FROM**

producto p, categoria c

**WHERE**

p.id\_categoria = c.id **AND**

fecha\_baja **IS** **NULL**

**GROUP** **BY** id\_categoria

**HAVING** total > 1;

