

[Página Principal](#) / [Mis cursos](#) / [Base de Datos](#) / [UNIDAD 3](#) / [Lenguaje SQL](#)

Lenguaje SQL

Uso de funciones agregadas (II)

Veamos ahora un ejemplo para entender cómo funciona la suma y luego lo extenderemos al resto de las funciones agregadas citadas.



Si queremos saber cuánto se vendió en el mes de enero, o sea de las 3 ventas que vimos anteriormente, quisiéramos calcular su suma total, multiplicando el precio por la cantidad y sumando obtendríamos un solo número **\$1936.5** calculado así:

$$15*10 + 25*1.7 + 21*10 + 33*22 + 29*17 + 13*15 + 12*10$$

Seguramente te preguntarás ahora



¿Cómo haríamos este mismo cálculo con el lenguaje SQL? Y además ¿Cuál es la potencia de

Hacé clic en el botón para ver la respuesta.

Pues bien, para realizar una consulta que calcule esto mismo, usaremos la consulta interna del ejemplo anterior:



SQL

```
SELECT SUM(cant*pr.precio) FROM pedidos_productos pp
  JOIN productos pr ON pp.codproducto=pr.codigo
  JOIN pedidos p ON p.nro=pp.codpedido
 WHERE month(p.fecha)=1 and year(p.fecha)=2017
```

Observá que tenemos que usar la tabla pedidos para poder usar la condición del mes= enero y el año=2017 con la fecha, la tabla productos porque necesitamos los precios y la tabla pedidos_productos porque necesitamos la cantidad comprada de cada producto por cada pedido. Por eso necesitamos hacer esos **tres join**.

Además...

¡Claro que te habrás dado cuenta que la potencia está en que ahora podremos calcular una cuenta similar no solo para el mes de enero sino para todos los meses y podremos entonces saber la venta mensual, y a partir de allí la venta anual!

De manera similar, si quisiéramos saber cuántas unidades se vendieron de cada producto (no importa si son kg, litros o unidades), podríamos formularlo de la siguiente manera:



SQL

```
SELECT SUM(cant) FROM pedidos_productos pp
  JOIN productos pr ON pp.codproducto=pr.codigo
  JOIN pedidos p ON p.nro=pp.codpedido
 WHERE month(p.fecha)=1 and year(p.fecha)=2017
```

Y ahora, vamos a ingresar **el mes** para obtener la suma de las ventas de cada uno de los meses del año 2017, así veremos cómo trabaja el **GROUP BY**:



SQL

```
SELECT month(p.fecha) as Mes, SUM(cant) as Cantidad
  FROM pedidos_productos pp
  JOIN productos pr ON pp.codproducto=pr.codigo
  JOIN pedidos p ON p.nro=pp.codpedido
 WHERE year(p.fecha)=2017
 GROUP BY month(p.fecha)
```

Esta consulta nos retornará las siguientes **T-uplas**:

1	148
---	-----

3	174
---	-----

4	130
---	-----

Si hubiéramos querido saber los pedidos cuyo total fuera superior a \$ 1000 hubiéramos tenido que hacer lo siguiente:



SQL

```
SELECT p.Nro, SUM(cant*precio) as total FROM pedidos_productos pp
JOIN productos pr ON pp.codproducto=pr.codigo
JOIN pedidos p ON p.nro=pp.codpedido
GROUP BY p.nro
HAVING SUM(cant*precio)>1000
```

Antes de pasar a la siguiente actividad te proponemos ver el siguiente video. En él podrás ver algunos ejemplos adicionales de uso de funciones agregadas.

The video player shows a MySQL Workbench interface. The title bar reads "BD U3 - Funciones Agregadas from Buenos Aires Data". The SQL editor contains a query to calculate the total sales per product code, grouped by month. The results grid shows the following data:

mes	max(total)
3	192.90000012920929
4	123
10	592
11	595.9000001333786
12	602.0000004768372

The video player includes a play button, a progress bar, and a Vimeo logo in the bottom right corner. A timestamp "07:46" is visible in the bottom left corner of the video frame.

< Anterior

Siguiente >



Buenos Aires Ciudad

Dirección General de Gestión Digital
Ministerio de Modernización

Español - Internacional (es)

English (en)

Español - Internacional (es)