

Avaluació Continua UF1845

Ferriol Baburés Morente

Pregunta 1

Calcula el total de ventas para cada producto, ordenado de mayor a menor.

```
-- Seleccionamos la tabla detalles y anexionamos (join) las tablas productos y pedidos para obtener el id_producto e id_pedido
-- los comparamos con los id correspondientes en la tabla detalles_pedidos. Realizamos la operacion para calcular el total de ventas de cada uno
-- los agrupamos por el nombre del producto y los ordenamos de mayor a menor
select pr.nombre as producto, sum(dp.cantidad * pr.precio) as total_ventas
from Detalles_Pedidos dp
join Productos pr on dp.id_producto = pr.id_producto
join Pedidos pe on dp.id_pedido = pe.id_pedido
group by pr.nombre
order by total_ventas desc;
```

Result Grid			Filter Rows:	Export:
	producto	total_ventas		
▶	Laptop	2400.00		
	Smartphone	1600.00		
	Monitor	1200.00		
	Impresora	1200.00		
	Escritorio	880.00		
	Cámara	500.00		
	Disco Duro Externo	360.00		
	Monitor Curvo	320.00		
	Tablet	250.00		
	Router	190.00		
	Silla ergonómica	180.00		
	Altavoces	140.00		
	Tedado Mecánico	130.00		
	Silla de oficina	120.00		
	Power Bank	100.00		
	Mochila	90.00		
	Micrófono	85.00		
	Mouse	50.00		
	Lámpara de escrit...	45.00		

```
-- En este codigo añadimos una clausula where para que no tenga en cuenta los productos de los pedidos cancelados
select pr.nombre as producto, sum(dp.cantidad * pr.precio) as total_ventas
from Detalles_Pedidos dp
join Productos pr on dp.id_producto = pr.id_producto
join Pedidos pe on dp.id_pedido = pe.id_pedido where pe.estado <> 'Cancelado'
group by pr.nombre
order by total_ventas desc;
```

Result Grid			Filter Rows:	Export:
	producto	total_ventas		
▶	Laptop	2400.00		
	Smartphone	1600.00		
	Monitor	1200.00		
	Impresora	1200.00		
	Escritorio	880.00		
	Cámara	500.00		
	Disco Duro Externo	360.00		
	Monitor Curvo	320.00		
	Tablet	250.00		
	Router	190.00		
	Silla ergonómica	180.00		
	Altavoces	140.00		
	Power Bank	100.00		
	Mochila	90.00		
	Micrófono	85.00		
	Lámpara de escrit...	45.00		

Pregunta 2

Identifica el último pedido realizado por cada cliente.

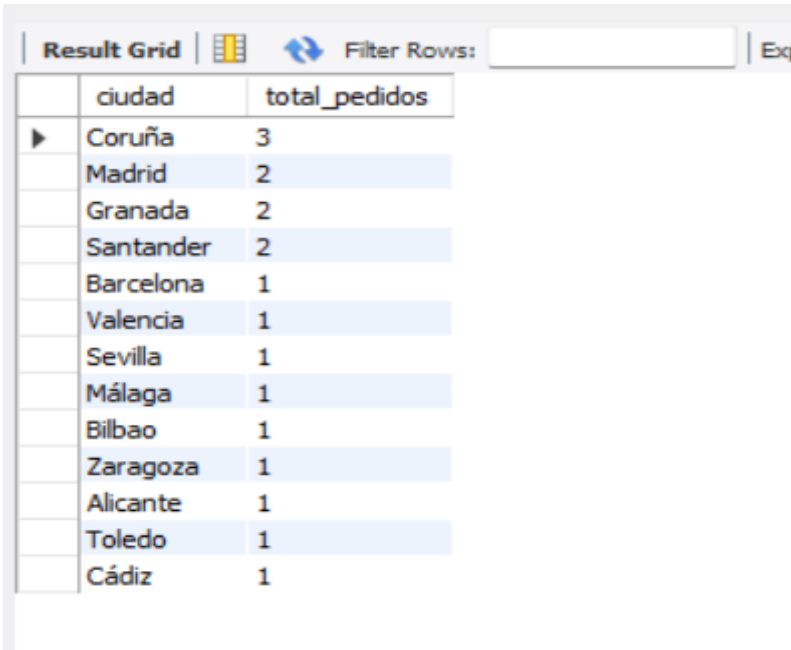
```
-- Obtenemos el id y nombre del cliente, e id y fecha del pedido. Anexionamos la tabla pedidos para obtener los pedidos de cada cliente (id que coincidan)
-- Hacemos una subconsulta que nos seleccione la fecha maxima para cada cliente (en pedidos) y solo tenga en cuenta ésta
select c.id_cliente, c.nombre as cliente, p.id_pedido, p.fecha_pedido from Clientes c
join Pedidos p on c.id_cliente = p.id_cliente
where p.fecha_pedido = (select max(p2.fecha_pedido) from Pedidos p2 where p2.id_cliente = c.id_cliente);
```

Result Grid					Filter Rows:	Export:
	id_cliente	cliente	id_pedido	fecha_pedido		
▶	1	Juan Pérez	1	2021-03-15		
	2	Ana Gómez	2	2021-03-17		
	3	Luisa Fernández	3	2021-03-20		
	4	Carlos Ruiz	4	2021-03-22		
	5	Marta López	5	2021-03-24		
	6	Andrea Jiménez	6	2021-03-25		
	7	Pablo Martínez	7	2021-03-28		
	8	Sofía Castro	8	2021-03-30		
	9	Raúl Navarro	9	2021-04-01		
	10	Irene Molina	16	2021-04-08		
	11	Jorge Esteban	17	2021-04-09		
	12	Clara Sanz	18	2021-04-10		
	13	Mario Redondo	13	2021-04-05		
	14	Lucía Marín	14	2021-04-06		

Pregunta 3

Determina el número total de pedidos realizados por clientes en cada ciudad.

```
-- Seleccionamos la ciudad de la tabla clientes, anexionamos la tabla pedidos para obtener los pedidos de cada cliente.  
-- Sumamos cada pedido y agrupamos por ciudad. Luego lo ordenamos de forma descendente  
select c.ciudad as ciudad, count(p.id_pedido) as total_pedidos from clientes c  
join pedidos p on c.id_cliente = p.id_cliente  
group by ciudad  
order by total_pedidos desc;
```



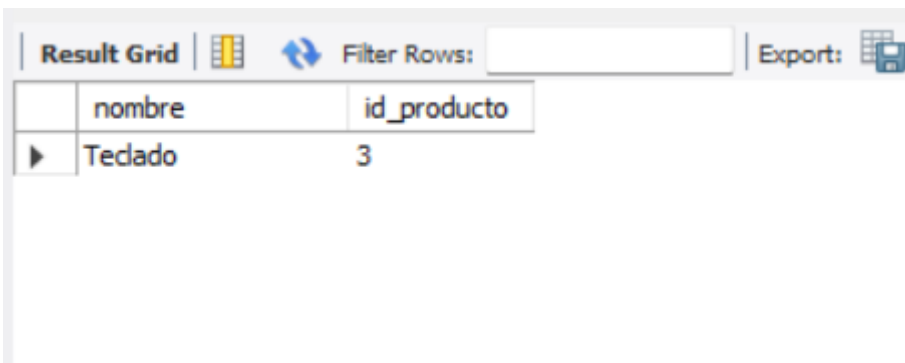
The screenshot shows a 'Result Grid' window with a toolbar containing icons for a grid, a refresh button, a 'Filter Rows' input field, and an 'Export' button. The table has two columns: 'ciudad' and 'total_pedidos'. The data is sorted in descending order of 'total_pedidos'.

	ciudad	total_pedidos
▶	Coruña	3
	Madrid	2
	Granada	2
	Santander	2
	Barcelona	1
	Valencia	1
	Sevilla	1
	Málaga	1
	Bilbao	1
	Zaragoza	1
	Alicante	1
	Toledo	1
	Cádiz	1

Pregunta 4

Lista todos los productos que nunca han sido parte de un pedido.

```
-- Seleccionamos el nombre e id de la tabla productos. Anexionamos la tabla detalles  
-- Y que nos devuelva solo aquellos productos que no tengan equivalencia (null)  
select p.nombre, p.id_producto from productos p  
left join detalles_pedidos dp on p.id_producto = dp.id_producto  
where dp.id_producto is null;
```



The screenshot shows a 'Result Grid' window with a toolbar containing icons for a grid, a refresh button, a 'Filter Rows' input field, and an 'Export' button. The table has two columns: 'nombre' and 'id_producto'. The data shows the product 'Teclado' with an 'id_producto' of 3.

	nombre	id_producto
▶	Teclado	3

Pregunta 5

Encuentra los productos más vendidos en términos de cantidad total vendida.

```
-- Seleccionamos el nombre de la tabla productos, adjuntamos la tabla detalles y sumamos los id_producto que coincidan.  
-- Los sumamos a traves del atributo cantidad que encuentre en la tabla detalles por cada producto  
-- Los agrupamos por el nombre del producto y los ordenamos de forma descendente. En este caso solo mostramos los 5 primeros  
select p.nombre, sum(dp.cantidad) as cantidad_total from productos p  
join detalles_pedidos dp on p.id_producto = dp.id_producto  
group by p.nombre  
order by cantidad_total desc  
limit 5;
```

	nombre	cantidad_total
▶	Impresora	8
	Monitor	4
	Escritorio	4
	Disco Duro Externo	3
	Smartphone	2

```
-- En esta opción hacemos lo mismo pero descartamos los pedidos cancelados  
select p.nombre, sum(dp.cantidad) as cantidad_total from productos p  
join detalles_pedidos dp on p.id_producto = dp.id_producto  
join pedidos pe on dp.id_pedido = pe.id_pedido  
where pe.estado <> 'Cancelado'  
group by p.nombre  
order by cantidad_total desc  
limit 5;
```

	nombre	cantidad_total
▶	Impresora	8
	Monitor	4
	Escritorio	4
	Disco Duro Externo	3
	Laptop	2

Pregunta 6

Identifica a los clientes que han realizado compras en más de una categoría de producto.

```
-- Seleccionamos id y nombre de la tabla clientes, adjuntamos las tablas de pedidos, detalles y productos
-- Esto lo hacemos para poder sacar que categoria(producto) tiene cada producto(detalles pedido) de cada pedido(cliente) de cada cliente(pedido)
-- agrupamos por cliente y mostramos solo aquellos que hayan comprado en mas de una categoria(ultima linea)
select c.id_cliente, c.nombre as cliente from clientes c
join pedidos pe on c.id_cliente = pe.id_cliente
join detalles_pedidos dp on pe.id_pedido = dp.id_pedido
join productos p on dp.id_producto = p.id_producto
group by c.id_cliente, cliente
having count(distinct p.categoria) > 1;
```

	id_cliente	cliente
▶	5	Marta López
	9	Raúl Navarro
	10	Irene Molina
	11	Jorge Esteban
	12	Clara Sanz

Pregunta 7

Muestra las ventas totales agrupadas por mes y año.

```
-- Seleccionamos el año y mes de la tabla pedidos(fecha) y realizamos la suma de cantidad producto(detallesP) por el precio (producto)
-- hacemos join para obtener los detalles de los pedidos y la cantidad de productos donde tengan coincidencia entre tablas
-- Los agrupamos por mes y año
select year(pe.fecha_pedido) as año, month(pe.fecha_pedido) as mes, sum(dp.cantidad * p.precio) as total_ventas from pedidos pe
join detalles_pedidos dp on pe.id_pedido = dp.id_pedido
join productos p on dp.id_producto = p.id_producto
group by año, mes
order by mes, año;
```

	año	mes	total_ventas
▶	2021	3	5890.00
	2021	4	3950.00

```
-- Aquí hacemos lo mismo pero descartando aquellos pedidos con estado cancelado
select year(pe.fecha_pedido) as año, month(pe.fecha_pedido) as mes, sum(dp.cantidad * p.precio) as total_ventas from pedidos pe
join detalles_pedidos dp on pe.id_pedido = dp.id_pedido
join productos p on dp.id_producto = p.id_producto
where pe.estado <> 'Cancelado'
group by mes, año
order by mes, año;
```

Result Grid			
Filter Rows:			
	año	mes	total_ventas
▶	2021	3	5720.00
	2021	4	3820.00

Pregunta 8

Calcula la cantidad promedio de productos por pedido.

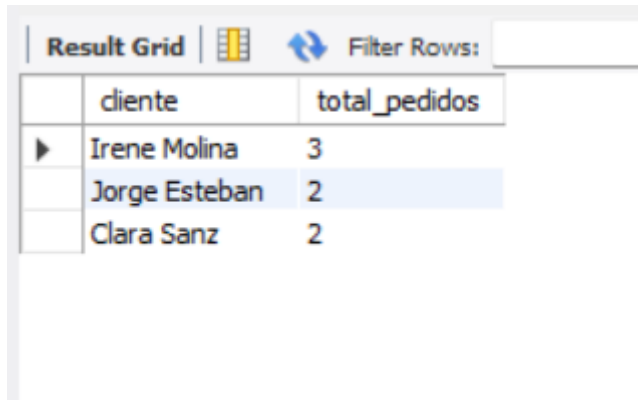
```
-- Primero se realiza una consulta interna para que sume la cantidad de productos de cada pedido(detalles).
-- Se pone 'as pedidos' porque toda tabla derivada necesita un alias
-- Finalmente realizamos la consulta principal con un promedio dividiendo la suma (cantidad_producto_pedido) por el numero de pedidos
select avg(cantidad_producto_pedido) as promedio from
(
select sum(cantidad) as cantidad_producto_pedido
from detalles_pedidos
group by id_pedido) as pedidos;
```

Result Grid	
Filter Rows:	
	promedio
▶	2.1667

Pregunta 9

Determina cuántos clientes han realizado pedidos en más de una ocasión. Saca el nombre del cliente

```
-- Seleccionamos el nombre del cliente de la tabla clientes, anexionamos la tabla pedidos cogiendo los id de cliente que coincida con id del cliente en la tabla clientes
-- agrupamos por cliente y sacamos solo aquellos que tengan mas de un pedido
select c.nombre as cliente, count(p.id_pedido) as total_pedidos from clientes c
join pedidos p on c.id_cliente = p.id_cliente
group by cliente
having count(p.id_pedido) > 1;
```



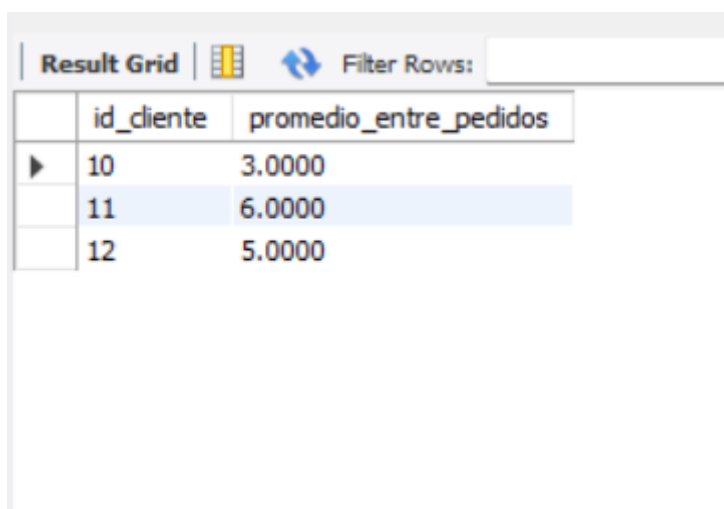
The screenshot shows a database interface with a 'Result Grid' tab. The grid displays the results of a SQL query. The columns are 'cliente' and 'total_pedidos'. The data rows are:

cliente	total_pedidos
Irene Molina	3
Jorge Esteban	2
Clara Sanz	2

Pregunta 10

Calcula el tiempo promedio que pasa entre pedidos para cada cliente.

```
-- El primer select coge el id del cliente y hace el promedio con la diferencia de dias entre el pedido de dicho cliente y el anterior pedido del mismo
-- para obtener el pedido anterior, utilizamos la funcion lag, hacemos una particion(partition by) de la tabla para cada cliente y la ordenamos con order by según la fecha
-- la informacion necesaria la obtenemos de la tabla pedidos. Con el where le indicamos que si es null, no coja dicho pedido
-- (significa que es el primero y en consecuencia no tiene anterior pedido)
select id_cliente, avg(datediff(fecha_pedido, lag_fecha_pedido)) as promedio_entre_pedidos from
(
    select id_pedido, id_cliente, fecha_pedido,
    lag(fecha_pedido) over (partition by id_cliente order by fecha_pedido) as lag_fecha_pedido
    from Pedidos
)
as pedidos_con_lag
where lag_fecha_pedido is not null
group by id_cliente;
```



The screenshot shows a database interface with a 'Result Grid' tab. The grid displays the results of a SQL query. The columns are 'id_cliente' and 'promedio_entre_pedidos'. The data rows are:

id_cliente	promedio_entre_pedidos
10	3.0000
11	6.0000
12	5.0000