Ejercicios SQL Bootcamp Data Engineer - EDVAI

Consignas:

A) Escribir las querys/consultas necesarias para llegar al resultado (print), usando windows functions.
B) Las consultas deben ser subidas a un proyecto público de github y compartir el link al instructor.

Nota: el proyecto de github debe tener al menos dos commits (puede ser uno por el punto B y otro subir un archivo .sql con las consultas) y deberá ser compartido con el instructor.

AVG

1. Obtener el promedio de precios por cada categoría de producto. La cláusula OVER(PARTITION BY CategoryID) específica que se debe calcular el promedio de precios por cada valor único de CategoryID en la tabla.

Print:

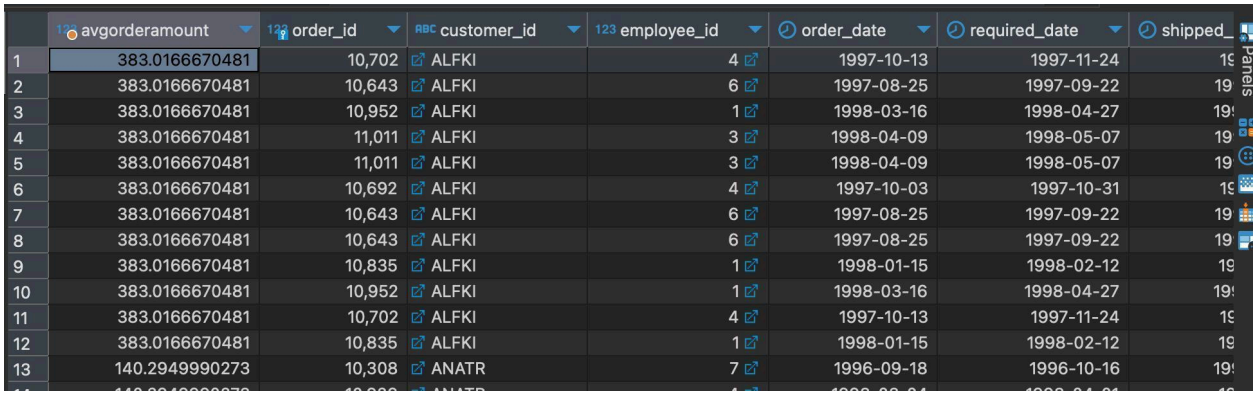
category_name	product_name	unit_price	avgpricebycategory
Beverages	Guaraná Fantástica	4.5	37.9791666667
Beverages	Ipoh Coffee	46	37.9791666667
Beverages	Chartreuse verte	18	37.9791666667
Beverages	Côte de Blaye	263.5	37.9791666667
Beverages	Steeleye Stout	18	37.9791666667
Beverages	Sasquatch Ale	14	37.9791666667
Beverages	Lakkalikööri	18	37.9791666667
Beverages	Rhönbräu Klosterbier	7.75	37.9791666667
Beverages	Outback Lager	15	37.9791666667
Beverages	Chai	18	37.9791666667
Beverages	Laughing Lumberjack Lager	14	37.9791666667
Beverages	Chang	19	37.9791666667
Condiments	Gula Malacca	19.450000763	22.8541668256
Condiments	Original Frankfurter grüne Soße	13	22.8541668256

Respuesta:

```
    select c.category_name , p.product_name , p.unit_price ,
avg(p.unit_price ) over (partition by c.category_id) as avgpricebycategory
from products p inner join categories c
    on c.category_id = p.category_id
```

2. Obtener el promedio de venta de cada cliente:

Print:



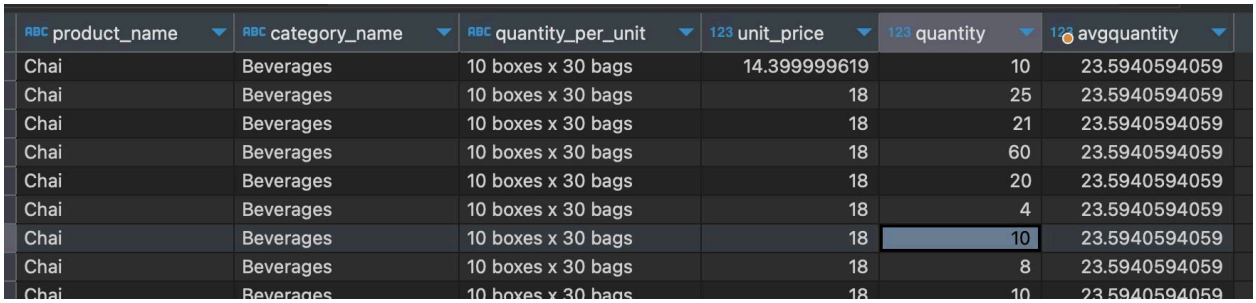
	avgorderamount	order_id	customer_id	employee_id	order_date	required_date	shipped_
1	383.0166670481	10,702	ALFKI	4	1997-10-13	1997-11-24	19
2	383.0166670481	10,643	ALFKI	6	1997-08-25	1997-09-22	19
3	383.0166670481	10,952	ALFKI	1	1998-03-16	1998-04-27	19
4	383.0166670481	11,011	ALFKI	3	1998-04-09	1998-05-07	19
5	383.0166670481	11,011	ALFKI	3	1998-04-09	1998-05-07	19
6	383.0166670481	10,692	ALFKI	4	1997-10-03	1997-10-31	19
7	383.0166670481	10,643	ALFKI	6	1997-08-25	1997-09-22	19
8	383.0166670481	10,643	ALFKI	6	1997-08-25	1997-09-22	19
9	383.0166670481	10,835	ALFKI	1	1998-01-15	1998-02-12	19
10	383.0166670481	10,952	ALFKI	1	1998-03-16	1998-04-27	19
11	383.0166670481	10,702	ALFKI	4	1997-10-13	1997-11-24	19
12	383.0166670481	10,835	ALFKI	1	1998-01-15	1998-02-12	19
13	140.2949990273	10,308	ANATR	7	1996-09-18	1996-10-16	19

Respuesta:

```
select avg(od.unit_price * od.quantity) over (partition by customer_id) as avgorderamount, *
from orders o inner join order_details od
on o.order_id = od.order_id
```

3. Obtener el promedio de cantidad de productos vendidos por categoría (product__name, quantity_per_unit, unit_price, quantity, avgquantity) y ordenarlo por nombre de la categoría y nombre del producto

Print:



product_name	category_name	quantity_per_unit	unit_price	quantity	avgquantity
Chai	Beverages	10 boxes x 30 bags	14.399999619	10	23.5940594059
Chai	Beverages	10 boxes x 30 bags	18	25	23.5940594059
Chai	Beverages	10 boxes x 30 bags	18	21	23.5940594059
Chai	Beverages	10 boxes x 30 bags	18	60	23.5940594059
Chai	Beverages	10 boxes x 30 bags	18	20	23.5940594059
Chai	Beverages	10 boxes x 30 bags	18	4	23.5940594059
Chai	Beverages	10 boxes x 30 bags	18	10	23.5940594059
Chai	Beverages	10 boxes x 30 bags	18	8	23.5940594059
Chai	Beverages	10 boxes x 30 bags	18	10	23.5940594059

Respuesta:

```
select p.product_name , c.category_name , p.quantity_per_unit , p.unit_price , od.quantity,
avg(od.quantity ) over (partition by c.category_id  ) as avgquantity
from products p inner join order_details od
on od.product_id = p.product_id
inner join categories c
on c.category_id = p.category_id
order by c.category_name, p.product_name
```

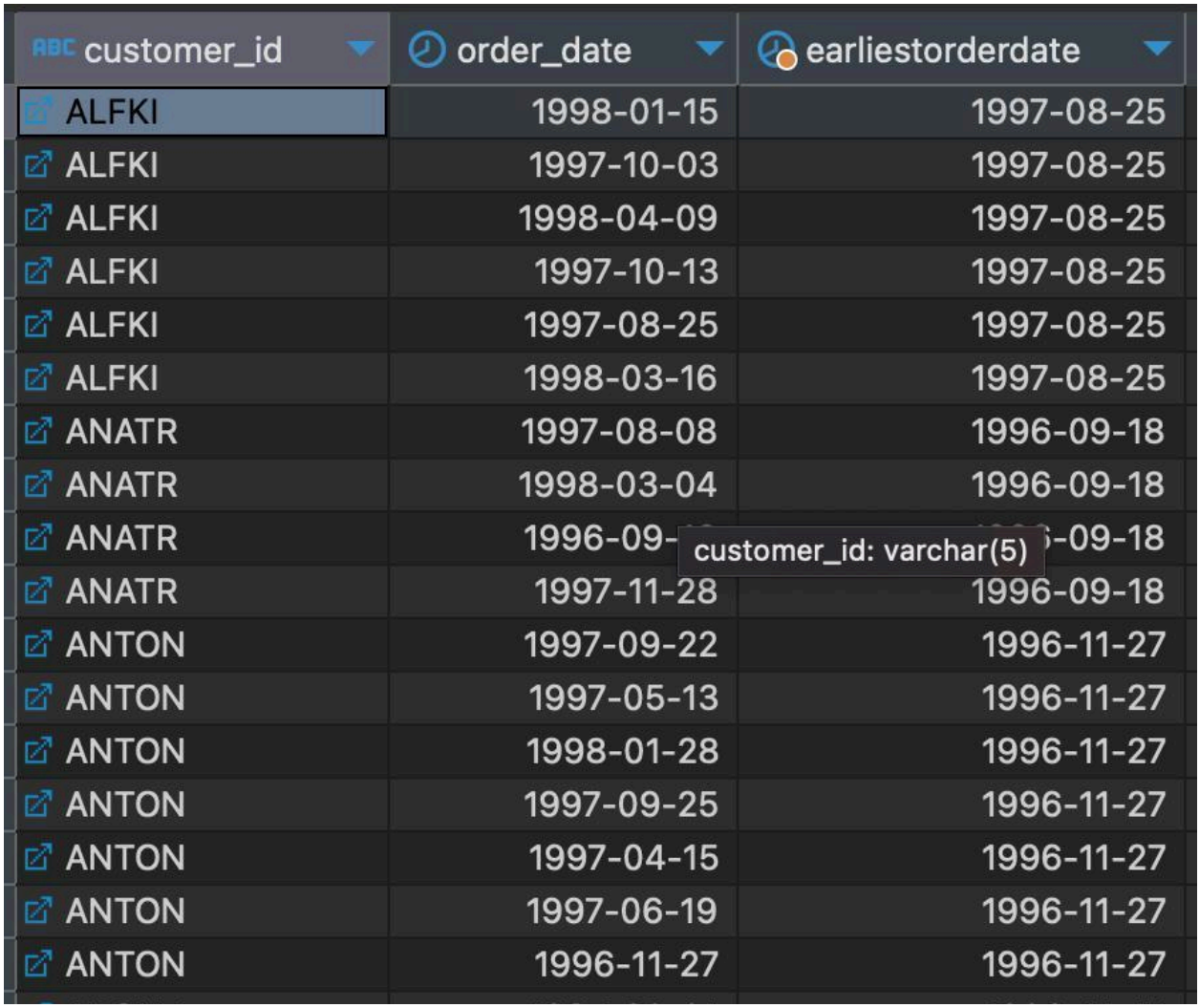
Consulta:

¿Qué es mejor hacer partition by por category_id porque es numérico ó category_name?

MIN

4 Selecciona el ID del cliente, la fecha de la orden y la fecha más antigua de la orden para cada cliente de la tabla 'Orders'.

Print:



customer_id	order_date	earliestorderdate
ALFKI	1998-01-15	1997-08-25
ALFKI	1997-10-03	1997-08-25
ALFKI	1998-04-09	1997-08-25
ALFKI	1997-10-13	1997-08-25
ALFKI	1997-08-25	1997-08-25
ALFKI	1998-03-16	1997-08-25
ANATR	1997-08-08	1996-09-18
ANATR	1998-03-04	1996-09-18
ANATR	1996-09-	-09-18
ANATR	1997-11-28	1996-09-18
ANTON	1997-09-22	1996-11-27
ANTON	1997-05-13	1996-11-27
ANTON	1998-01-28	1996-11-27
ANTON	1997-09-25	1996-11-27
ANTON	1997-04-15	1996-11-27
ANTON	1997-06-19	1996-11-27
ANTON	1996-11-27	1996-11-27

customer_id: varchar(5)

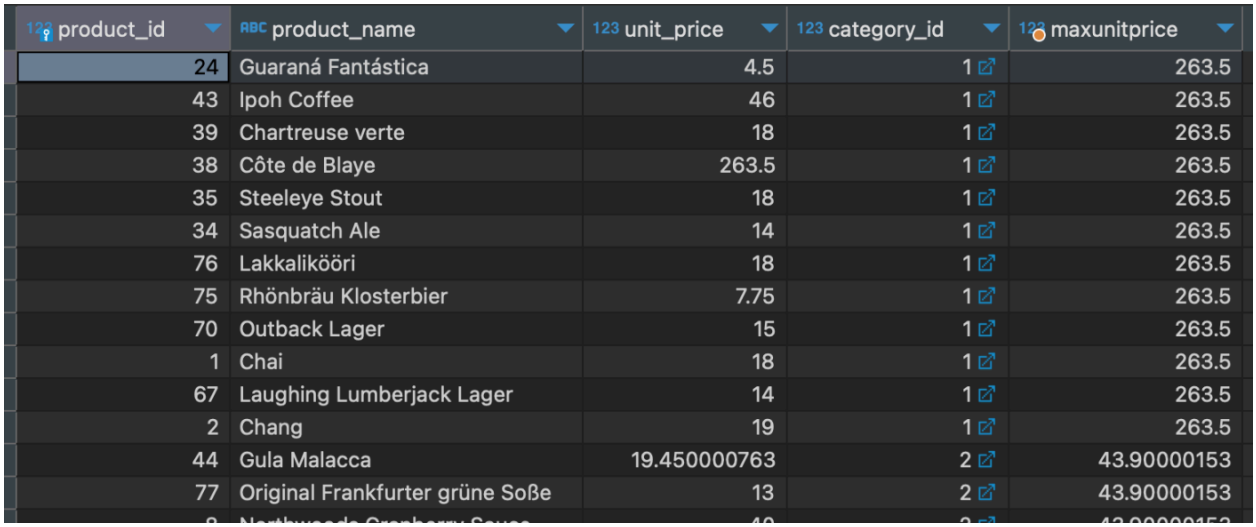
Respuesta:

```
select o.customer_id , o.order_date,
min(o.order_date) over (partition by o.customer_id) as  earliestorderdate
from orders o inner join customers c
on c.customer_id = o.customer_id
```

MAX

5. Seleccione el id de producto, el nombre de producto, el precio unitario, el id de categoría y el precio unitario máximo para cada categoría de la tabla Products.

   Print:



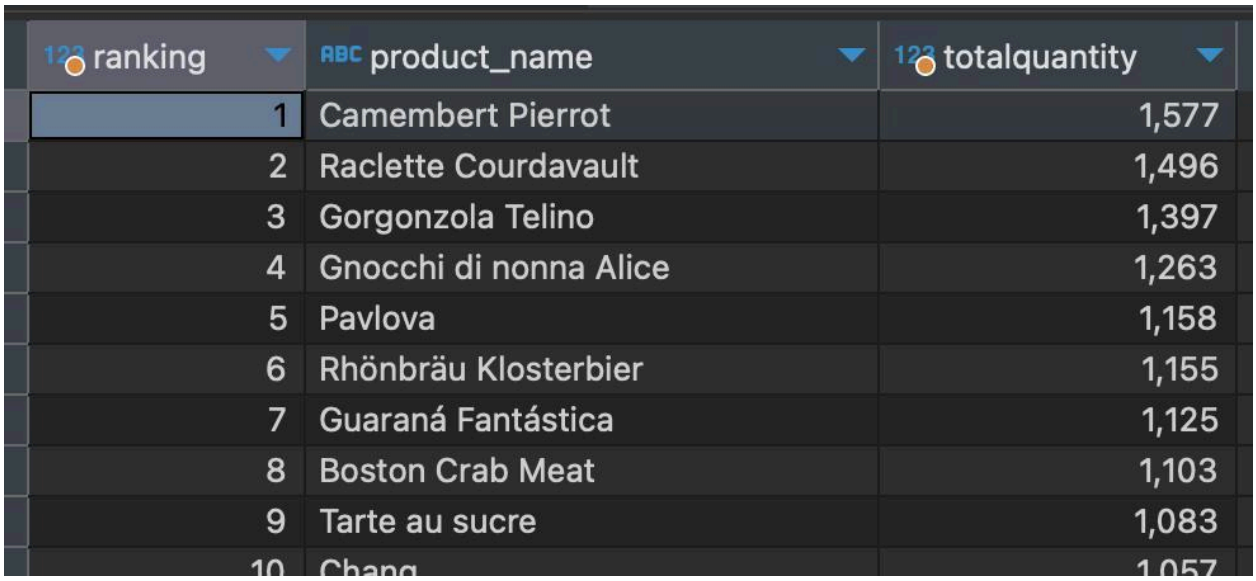
product_id	product_name	unit_price	category_id	maxunitprice
24	Guaraná Fantástica	4.5	1	263.5
43	Ipoh Coffee	46	1	263.5
39	Chartreuse verte	18	1	263.5
38	Côte de Blaye	263.5	1	263.5
35	Steeleye Stout	18	1	263.5
34	Sasquatch Ale	14	1	263.5
76	Lakkalikööri	18	1	263.5
75	Rhönbräu Klosterbier	7.75	1	263.5
70	Outback Lager	15	1	263.5
1	Chai	18	1	263.5
67	Laughing Lumberjack Lager	14	1	263.5
2	Chang	19	1	263.5
44	Gula Malacca	19.450000763	2	43.90000153
77	Original Frankfurter grüne Soße	13	2	43.90000153

Respuesta

```
select p.product_id , p.product_name , p.unit_price , c.category_id ,
max(p.unit_price) over (partition by p.category_id) as maxunitprice
from products p inner join categories c
on c.category_id = p.category_id
```

Row_number

6. Obtener el ranking de los productos más vendidos
   Print:

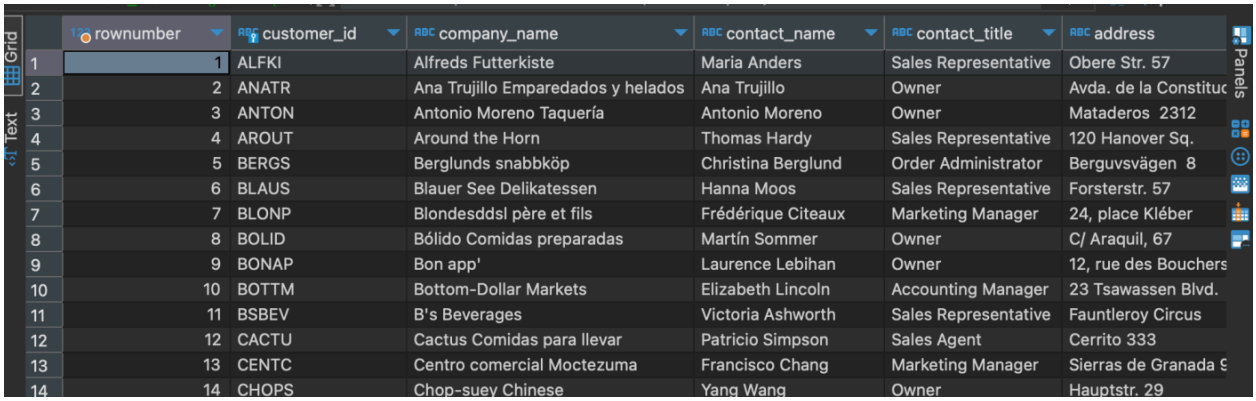


ranking	product_name	totalquantity
1	Camembert Pierrot	1,577
2	Raclette Courdavault	1,496
3	Gorgonzola Telino	1,397
4	Gnocchi di nonna Alice	1,263
5	Pavlova	1,158
6	Rhönbräu Klosterbier	1,155
7	Guaraná Fantástica	1,125
8	Boston Crab Meat	1,103
9	Tarte au sucre	1,083
10	Chang	1,057

Respuesta:

```
select row_number() over (order by sum(od.quantity) desc) as ranking, p.product_name,
sum(od.quantity) as totalquantity
from products p inner join order_details od
on p.product_id = od.product_id
group by p.product_id
```

7. Asignar numeros de fila para cada cliente, ordenados por customer_id
Print:



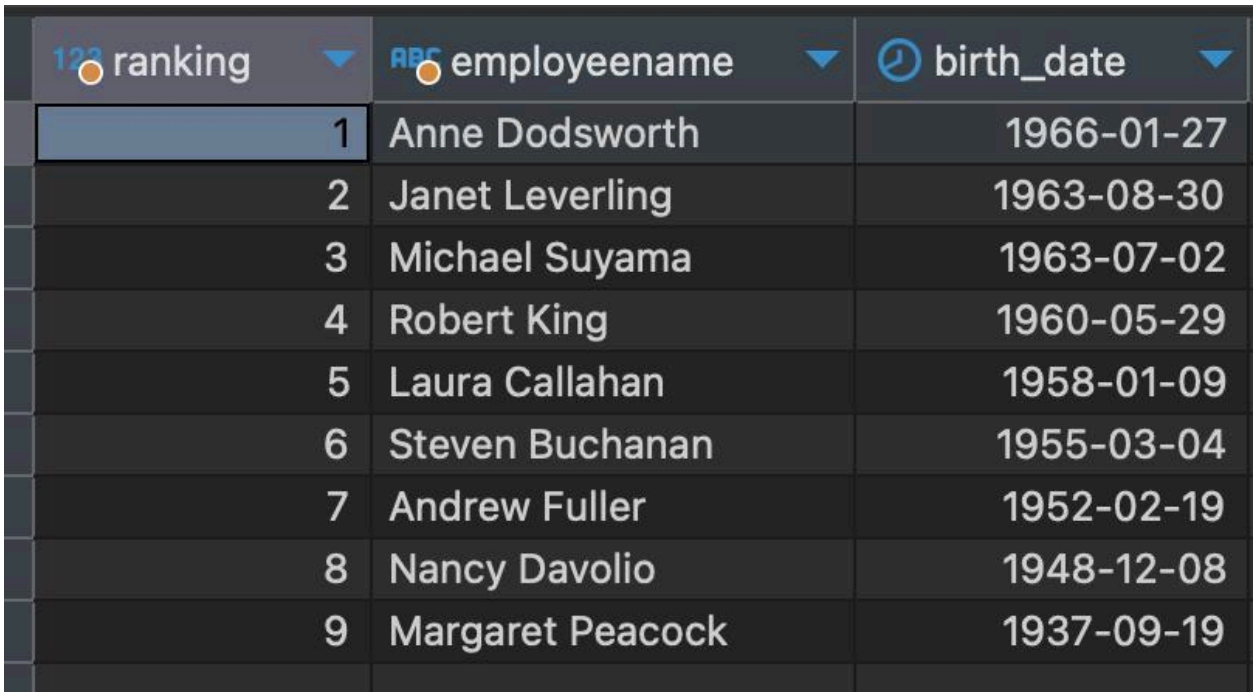
	rownumber	customer_id	company_name	contact_name	contact_title	address
1	1	ALFKI	Alfreds Futterkiste	Maria Anders	Sales Representative	Obere Str. 57
2	2	ANATR	Ana Trujillo Emparedados y helados	Ana Trujillo	Owner	Avda. de la Constituc
3	3	ANTON	Antonio Moreno Taquería	Antonio Moreno	Owner	Mataderos 2312
4	4	AROUT	Around the Horn	Thomas Hardy	Sales Representative	120 Hanover Sq.
5	5	BERGS	Berglunds snabbköp	Christina Berglund	Order Administrator	Berguvsvägen 8
6	6	BLAUS	Blauer See Delikatessen	Hanna Moos	Sales Representative	Forsterstr. 57
7	7	BLONP	Blondesddsl père et fils	Frédérique Citeaux	Marketing Manager	24, place Kléber
8	8	BOLID	Bólido Comidas preparadas	Martín Sommer	Owner	C/ Araquil, 67
9	9	BONAP	Bon app'	Laurence Lebihan	Owner	12, rue des Bouchers
10	10	BOTTM	Bottom-Dollar Markets	Elizabeth Lincoln	Accounting Manager	23 Tsawassen Blvd.
11	11	BSBEV	B's Beverages	Victoria Ashworth	Sales Representative	Fauntleroy Circus
12	12	CACTU	Cactus Comidas para llevar	Patricio Simpson	Sales Agent	Cerrito 333
13	13	CENTC	Centro comercial Moctezuma	Francisco Chang	Marketing Manager	Sierras de Granada 9
14	14	CHOPS	Chop-suey Chinese	Yang Wang	Owner	Hauptstr. 29

Respuesta:

```
select row_number() over (order by c.customer_id asc) as rownumber,*
from customers c
```

8. Obtener el ranking de los empleados más jóvenes () ranking, nombre y apellido del empleado, fecha de nacimiento)

Print:



ranking	employeename	birth_date
1	Anne Dodsworth	1966-01-27
2	Janet Leverling	1963-08-30
3	Michael Suyama	1963-07-02
4	Robert King	1960-05-29
5	Laura Callahan	1958-01-09
6	Steven Buchanan	1955-03-04
7	Andrew Fuller	1952-02-19
8	Nancy Davolio	1948-12-08
9	Margaret Peacock	1937-09-19

1er forma:

```
select row_number() over (order by e.birth_date  desc) as ranking,
CONCAT(e.first_name , ' ',e.last_name) AS employeename, e.birth_date
from employees e
```

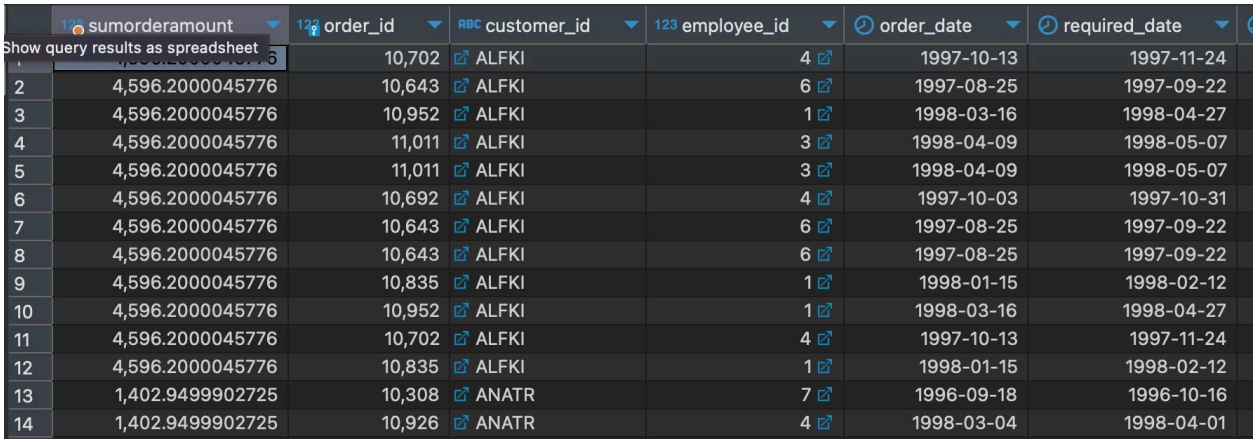
2da forma:

```
select
row_number() over(order by e.birth_date desc) as ranking,
e.first_name || ' ' || e.last_name as employeename, e.birth_date
from employees e
```

SUM

9. Obtener la suma de venta de cada cliente

Print:



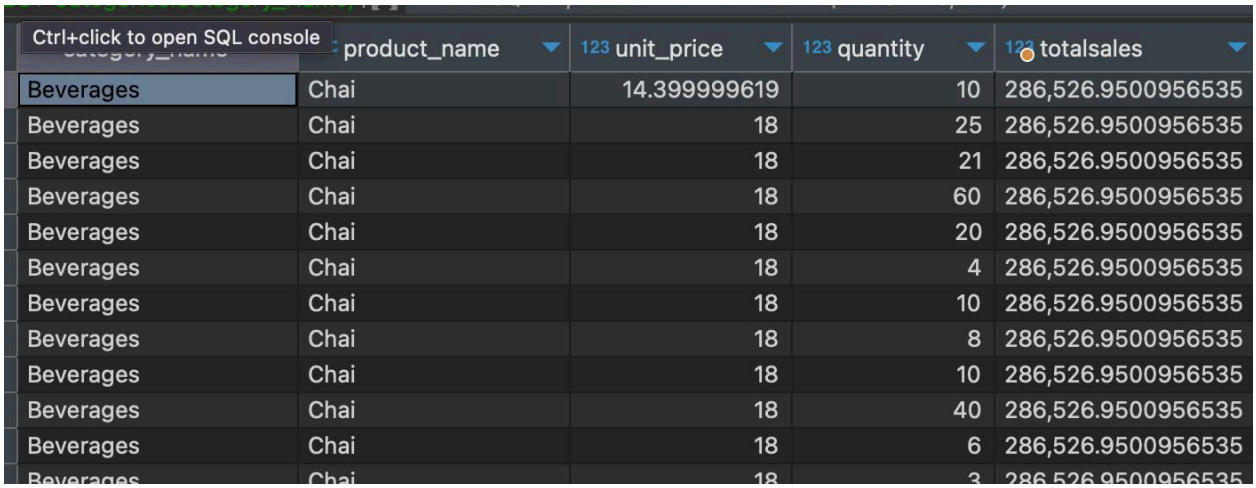
	sumorderamount	order_id	customer_id	employee_id	order_date	required_date
		10,702	ALFKI	4	1997-10-13	1997-11-24
2	4,596.2000045776	10,643	ALFKI	6	1997-08-25	1997-09-22
3	4,596.2000045776	10,952	ALFKI	1	1998-03-16	1998-04-27
4	4,596.2000045776	11,011	ALFKI	3	1998-04-09	1998-05-07
5	4,596.2000045776	11,011	ALFKI	3	1998-04-09	1998-05-07
6	4,596.2000045776	10,692	ALFKI	4	1997-10-03	1997-10-31
7	4,596.2000045776	10,643	ALFKI	6	1997-08-25	1997-09-22
8	4,596.2000045776	10,643	ALFKI	6	1997-08-25	1997-09-22
9	4,596.2000045776	10,835	ALFKI	1	1998-01-15	1998-02-12
10	4,596.2000045776	10,952	ALFKI	1	1998-03-16	1998-04-27
11	4,596.2000045776	10,702	ALFKI	4	1997-10-13	1997-11-24
12	4,596.2000045776	10,835	ALFKI	1	1998-01-15	1998-02-12
13	1,402.9499902725	10,308	ANATR	7	1996-09-18	1996-10-16
14	1,402.9499902725	10,926	ANATR	4	1998-03-04	1998-04-01

Respuesta:

```
select sum(od.quantity * od.unit_price) over (partition by customer_id) as sumorderamount, *
from orders o inner join order_details od
on o.order_id = od.order_id
```

10.Obtener la suma total de ventas por categoría de producto

Print:



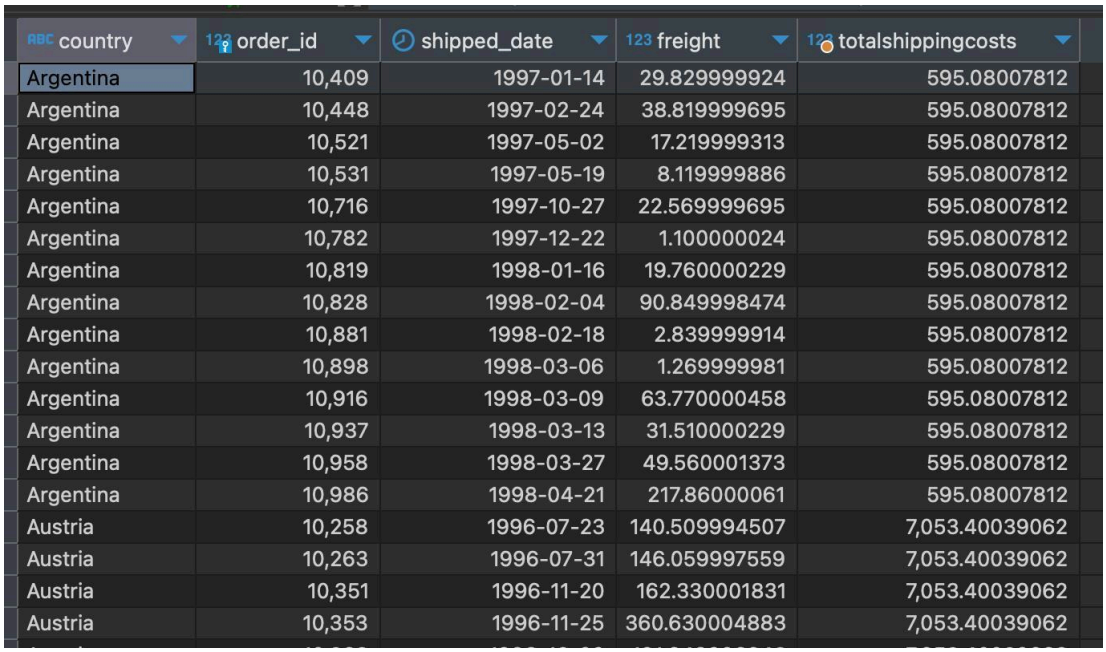
	product_name	123 unit_price	123 quantity	123 totalsales
Beverages	Chai	14.399999619	10	286,526.9500956535
Beverages	Chai	18	25	286,526.9500956535
Beverages	Chai	18	21	286,526.9500956535
Beverages	Chai	18	60	286,526.9500956535
Beverages	Chai	18	20	286,526.9500956535
Beverages	Chai	18	4	286,526.9500956535
Beverages	Chai	18	10	286,526.9500956535
Beverages	Chai	18	8	286,526.9500956535
Beverages	Chai	18	10	286,526.9500956535
Beverages	Chai	18	40	286,526.9500956535
Beverages	Chai	18	6	286,526.9500956535

Respuesta:

```
select c.category_name , p.product_name , od.unit_price , od.quantity,
sum (od.quantity * od.unit_price ) over (partition by c.category_name ) as totalsales
from order_details od inner join products p
on od.product_id = p.product_id inner join categories c
on c.category_id = p.category_id
order by c.category_id , p.product_name
```

11. Calcular la suma total de gastos de envío por país de destino, luego ordenarlo por país y por orden de manera ascendente

    Print:



country	order_id	shipped_date	freight	totalshippingcosts
Argentina	10,409	1997-01-14	29.829999924	595.08007812
Argentina	10,448	1997-02-24	38.819999695	595.08007812
Argentina	10,521	1997-05-02	17.219999313	595.08007812
Argentina	10,531	1997-05-19	8.119999886	595.08007812
Argentina	10,716	1997-10-27	22.569999695	595.08007812
Argentina	10,782	1997-12-22	1.100000024	595.08007812
Argentina	10,819	1998-01-16	19.760000229	595.08007812
Argentina	10,828	1998-02-04	90.849998474	595.08007812
Argentina	10,881	1998-02-18	2.839999914	595.08007812
Argentina	10,898	1998-03-06	1.269999981	595.08007812
Argentina	10,916	1998-03-09	63.770000458	595.08007812
Argentina	10,937	1998-03-13	31.510000229	595.08007812
Argentina	10,958	1998-03-27	49.560001373	595.08007812
Argentina	10,986	1998-04-21	217.86000061	595.08007812
Austria	10,258	1996-07-23	140.509994507	7,053.40039062
Austria	10,263	1996-07-31	146.059997559	7,053.40039062
Austria	10,351	1996-11-20	162.330001831	7,053.40039062
Austria	10,353	1996-11-25	360.630004883	7,053.40039062

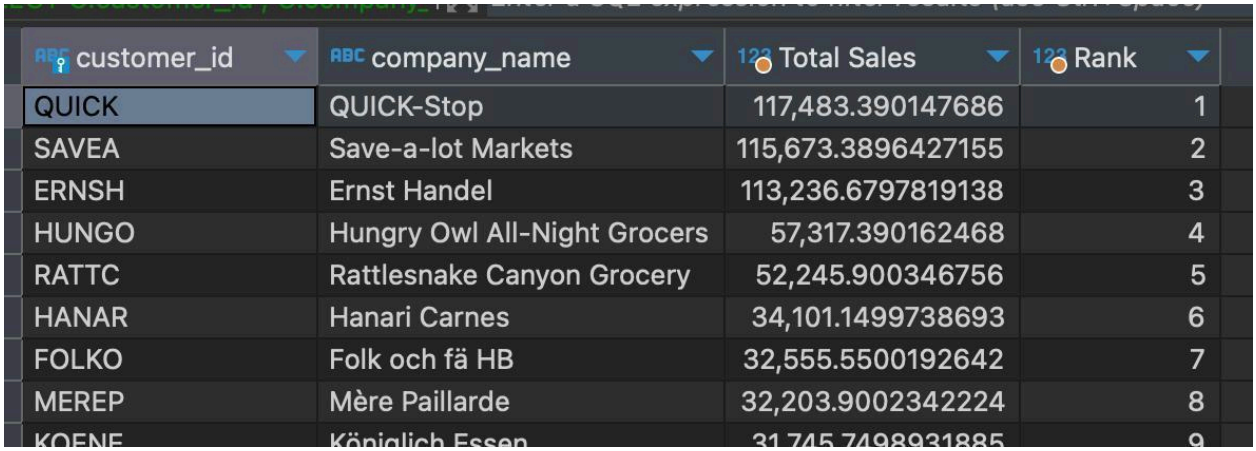
Respuesta:

```
select o.ship_country , o.order_id , o.shipped_date , o.freight ,
sum(o.freight ) over (partition by o.ship_country) as totalshippingcosts
from orders o
order by o.ship_country, o.order_id asc
```

RANK

12.Ranking de ventas por cliente

Print:



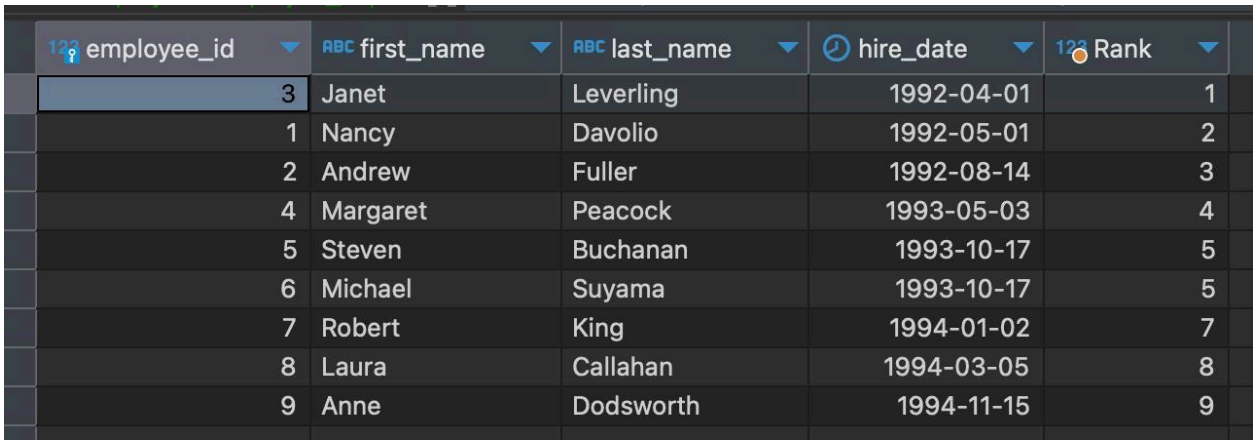
customer_id	company_name	Total Sales	Rank
QUICK	QUICK-Stop	117,483.390147686	1
SAVEA	Save-a-lot Markets	115,673.3896427155	2
ERNSH	Ernst Handel	113,236.6797819138	3
HUNGO	Hungry Owl All-Night Grocers	57,317.390162468	4
RATTC	Rattlesnake Canyon Grocery	52,245.900346756	5
HANAR	Hanari Carnes	34,101.1499738693	6
FOLKO	Folk och fä HB	32,555.5500192642	7
MEREP	Mère Paillarde	32,203.9002342224	8
KOENE	Königlich Essen	31.745.7498931885	9

Respuesta:

```
select o.customer_id , c.company_name , sum(od.unit_price * od.quantity ) as "Total Sales",
rank() over (order by sum(od.unit_price * od.quantity) desc )
from orders o inner join order_details od
on o.order_id = od.order_id inner join customers c
on c.customer_id = o.customer_id
group by o.customer_id , c.company_name
```

13.Ranking de empleados por fecha de contratación

Print:



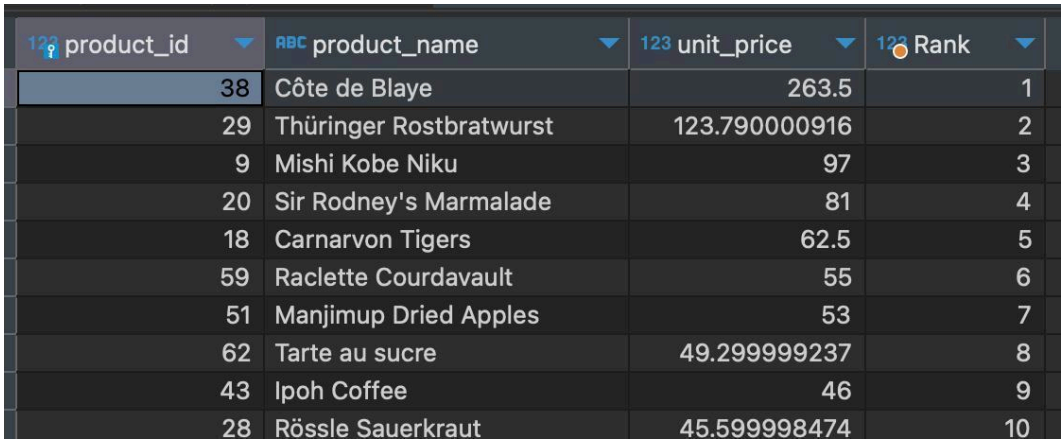
employee_id	first_name	last_name	hire_date	Rank
3	Janet	Leverling	1992-04-01	1
1	Nancy	Davolio	1992-05-01	2
2	Andrew	Fuller	1992-08-14	3
4	Margaret	Peacock	1993-05-03	4
5	Steven	Buchanan	1993-10-17	5
6	Michael	Suyama	1993-10-17	5
7	Robert	King	1994-01-02	7
8	Laura	Callahan	1994-03-05	8
9	Anne	Dodsworth	1994-11-15	9

Respuesta:

```
select e.employee_id , e.first_name , e.last_name , e.hire_date ,
rank() over (order by e.hire_date asc)
from employees e
```

14.Ranking de productos por precio unitario

Print:



product_id	product_name	unit_price	Rank
38	Côte de Blaye	263.5	1
29	Thüringer Rostbratwurst	123.790000916	2
9	Mishi Kobe Niku	97	3
20	Sir Rodney's Marmalade	81	4
18	Carnarvon Tigers	62.5	5
59	Raclette Courdavault	55	6
51	Manjimup Dried Apples	53	7
62	Tarte au sucre	49.299999237	8
43	Ipoh Coffee	46	9
28	Rössle Sauerkraut	45.599998474	10

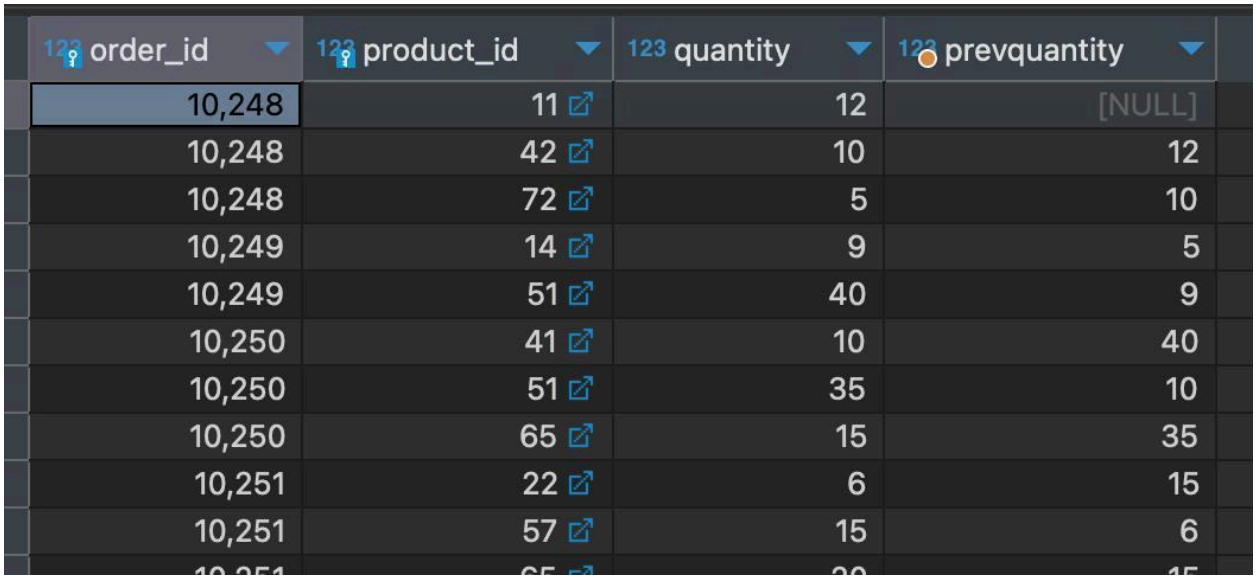
Respuesta:

```
select p.product_id , p.product_name , p.unit_price ,
rank() over (order by p.unit_price desc)
from products p
```

LAG

15. Mostrar por cada producto de una orden, la cantidad vendida y la cantidad vendida del producto previo.

Print:



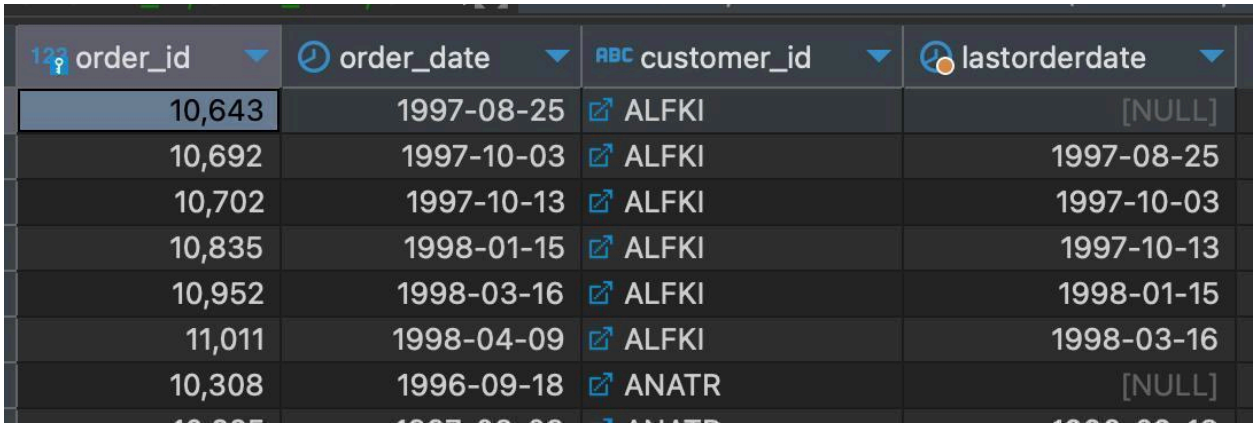
order_id	product_id	quantity	prevquantity
10,248	11	12	[NULL]
10,248	42	10	12
10,248	72	5	10
10,249	14	9	5
10,249	51	40	9
10,250	41	10	40
10,250	51	35	10
10,250	65	15	35
10,251	22	6	15
10,251	57	15	6

Respuesta:

```
select o.order_id , od.product_id , od.quantity ,
lag(od.quantity) over (order by o.order_id ) as prevquantity
from orders o inner join order_details od
on o.order_id = od.order_id
```

16. Obtener un listado de ordenes mostrando el id de la orden, fecha de orden, id del cliente y última fecha de orden.

    Print:



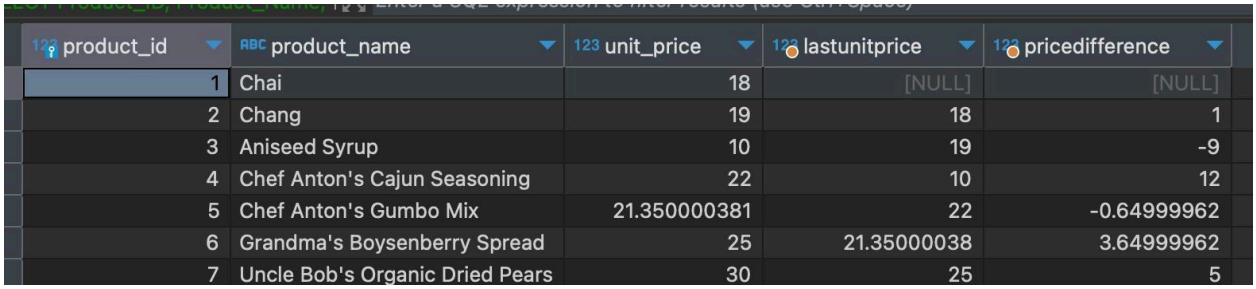
order_id	order_date	customer_id	lastorderdate
10,643	1997-08-25	ALFKI	[NULL]
10,692	1997-10-03	ALFKI	1997-08-25
10,702	1997-10-13	ALFKI	1997-10-03
10,835	1998-01-15	ALFKI	1997-10-13
10,952	1998-03-16	ALFKI	1998-01-15
11,011	1998-04-09	ALFKI	1998-03-16
10,308	1996-09-18	ANATR	[NULL]

Respuesta:

```
select o.order_id, o.order_date, o.customer_id,
lag(o.order_date) over (partition by o.customer_id order by o.order_date) as lastorderdate
from orders o
```

17. Obtener un listado de productos que contengan: id de producto, nombre del producto, precio unitario, precio del producto anterior, diferencia entre el precio del producto y precio del producto anterior.

    Print:



product_id	product_name	unit_price	lastunitprice	pricedifference
1	Chai	18	[NULL]	[NULL]
2	Chang	19	18	1
3	Aniseed Syrup	10	19	-9
4	Chef Anton's Cajun Seasoning	22	10	12
5	Chef Anton's Gumbo Mix	21.350000381	22	-0.64999962
6	Grandma's Boysenberry Spread	25	21.35000038	3.64999962
7	Uncle Bob's Organic Dried Pears	30	25	5

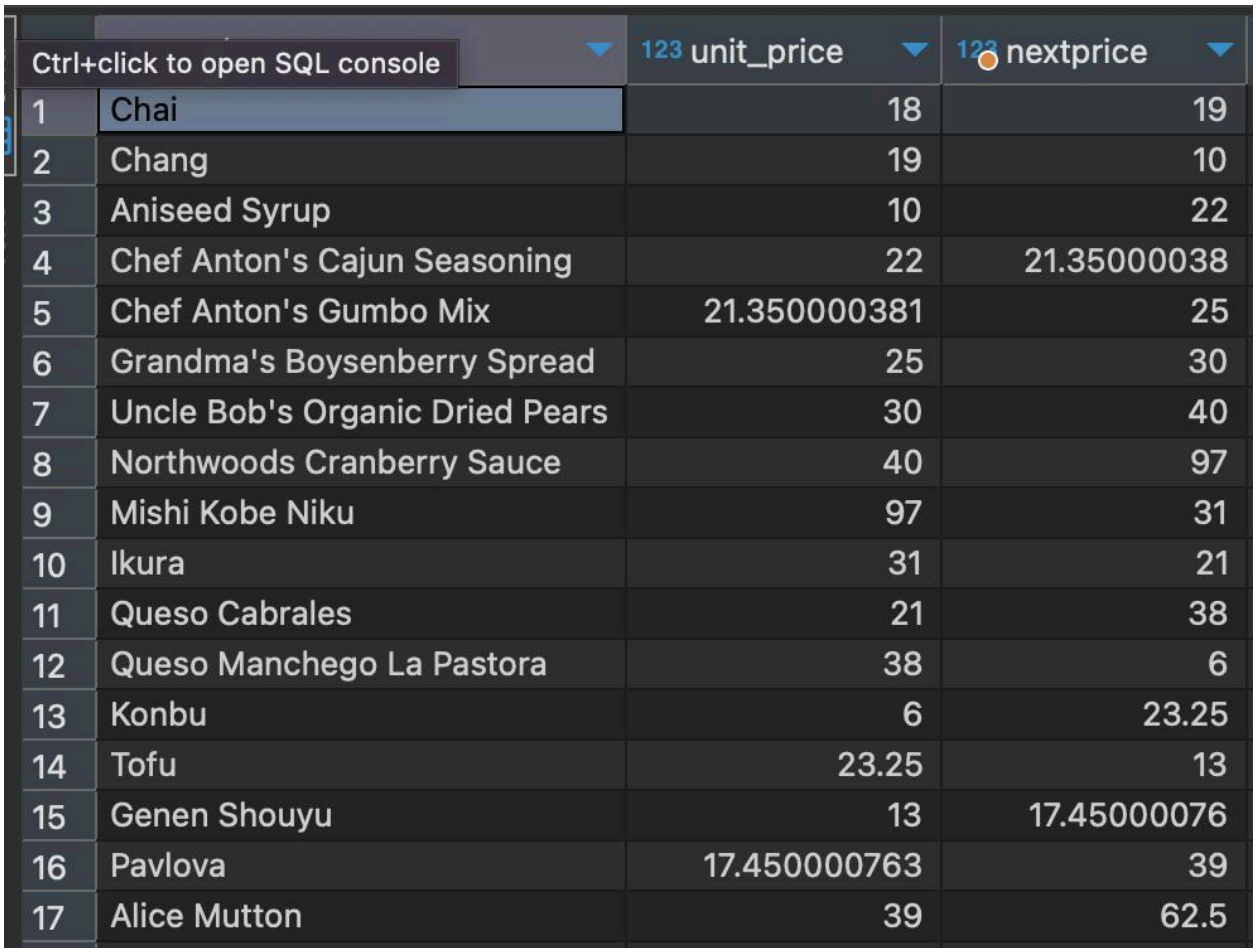
Respuesta:

```
select p.product_id, p.product_name , p.unit_price,
lag(p.unit_price ) over (order by p.product_id ) as lastunitprice,
(p.unit_price - lag(p.unit_price ) over (order by p.product_id) ) as pricedifference
from products p
```

LEAD

18. Obtener un listado que muestra el precio de un producto junto con el precio del producto siguiente:

Print:



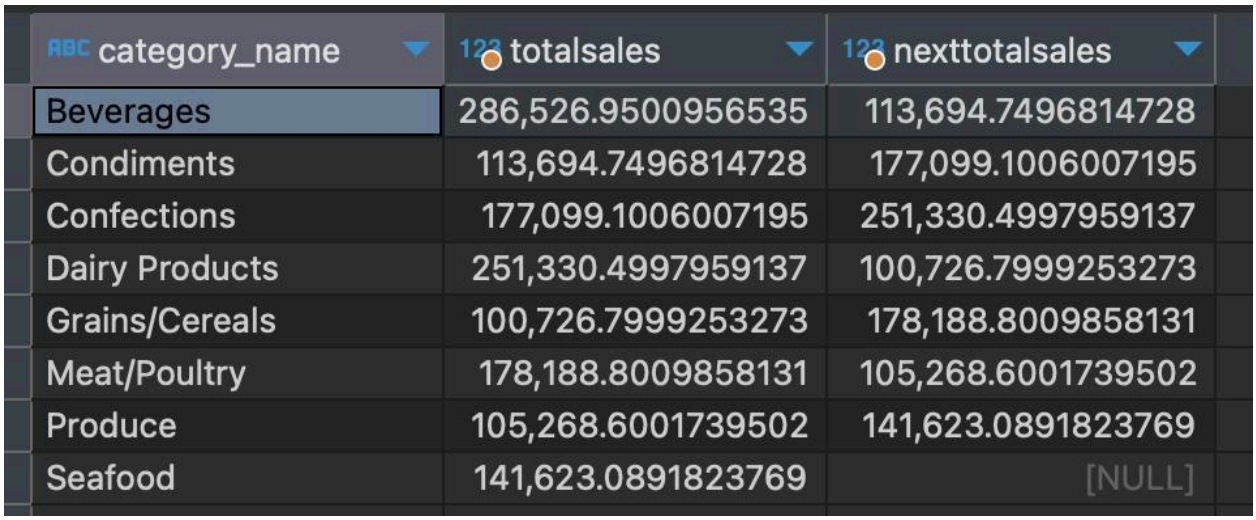
		123 unit_price	123 nextprice
	Ctrl+click to open SQL console		
1	Chai	18	19
2	Chang	19	10
3	Aniseed Syrup	10	22
4	Chef Anton's Cajun Seasoning	22	21.35000038
5	Chef Anton's Gumbo Mix	21.350000381	25
6	Grandma's Boysenberry Spread	25	30
7	Uncle Bob's Organic Dried Pears	30	40
8	Northwoods Cranberry Sauce	40	97
9	Mishi Kobe Niku	97	31
10	Ikura	31	21
11	Queso Cabrales	21	38
12	Queso Manchego La Pastora	38	6
13	Konbu	6	23.25
14	Tofu	23.25	13
15	Genen Shouyu	13	17.45000076
16	Pavlova	17.450000763	39
17	Alice Mutton	39	62.5

Respuesta:

```
select p.product_name ,p.unit_price ,
lead(p.unit_price ) over (order by p.product_id ) as nextprice
from products p
```

19. Obtener un listado que muestra el total de ventas por categoría de producto junto con el total de ventas de la categoría siguiente

Print:



category_name	totalsales	nexttotalsales
Beverages	286,526.9500956535	113,694.7496814728
Condiments	113,694.7496814728	177,099.1006007195
Confections	177,099.1006007195	251,330.4997959137
Dairy Products	251,330.4997959137	100,726.7999253273
Grains/Cereals	100,726.7999253273	178,188.8009858131
Meat/Poultry	178,188.8009858131	105,268.6001739502
Produce	105,268.6001739502	141,623.0891823769
Seafood	141,623.0891823769	[NULL]

Respuesta:

```
select c.category_name ,
sum(od.unit_price * od.quantity) as totalsales,
lead (sum(od.unit_price * od.quantity)) over (order by c.category_name  )  as nexttotalsales
--lead (c.category_name  ) over (order by c.category_id )  as nexttotalsales
from order_details od inner join products p
on od.product_id = p.product_id inner join categories c
on c.category_id = p.category_id
group by c.category_name
```

Consulta

Para este caso agrupamos por category_name, y comparando con el print de pantalla esperado se ve OK. Pero en la vida real,¿está bien que sea el filtro por name ó es mejor por category_id?