Ejercicios SQL Bootcamp Data Engineer - EDVAI

Consignas:

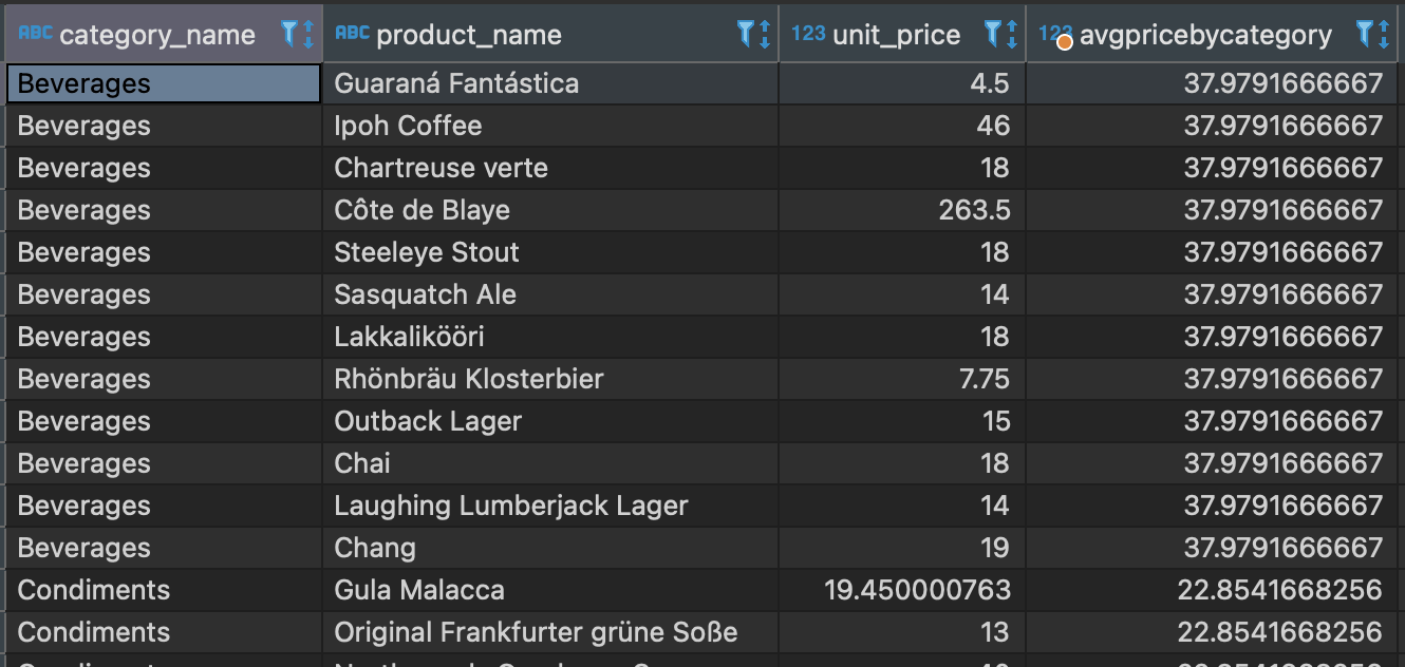
1. Escribir las querys/consultas necesarias para llegar al resultado (print), usando windows functions.
2. Las consultas deben ser subidas a un proyecto público de github y compartir el link al instructor.

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:

**Respuesta:**

**select** *c*.category\_name , *p*.product\_name , *p*.unit\_price ,

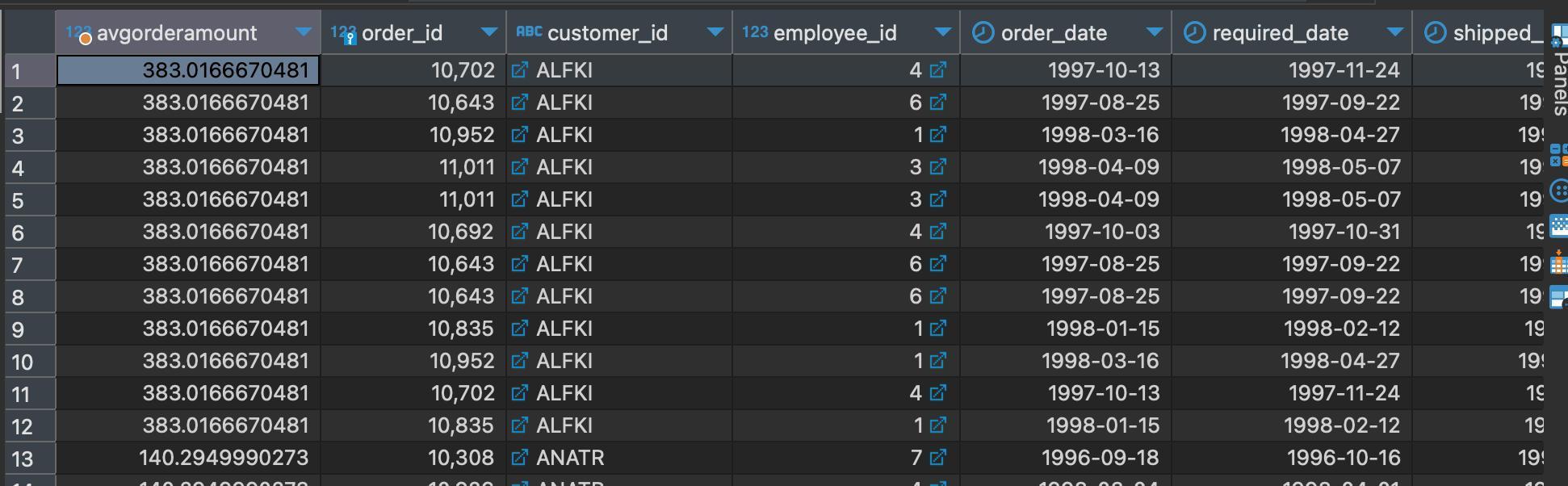
**avg**(*p*.unit\_price ) **over** (**partition** **by** c.category\_id) **as** *avgpricebycategory*

**from** products *p* **inner** **join** categories *c*

**on** *c*.category\_id = *p*.category\_id

1. Obtener el promedio de venta de cada cliente:

Print:



**Respuesta:**

**select** **avg**(*od*.unit\_price \* *od*.quantity) **over** (**partition** **by** customer\_id) **as** *avgorderamount*, \*

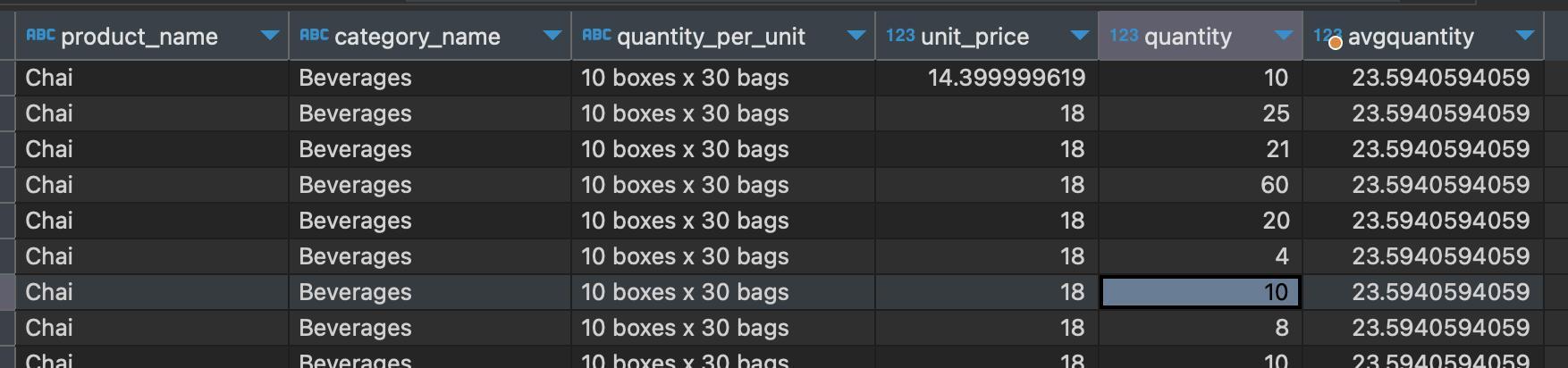
**from** orders *o* **inner** **join** order\_details *od*

**on** *o*.order\_id = *od*.order\_id

1. 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:



**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

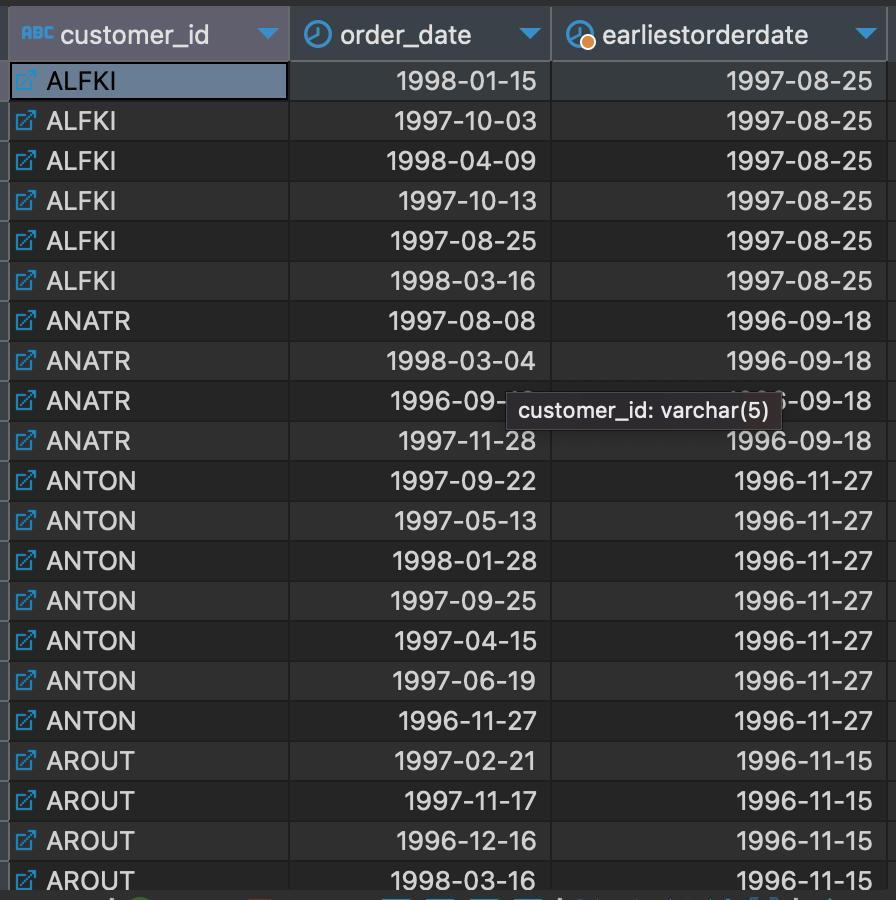
**order** **by** *c*.category\_name, *p*.product\_name

**Consulta:**

¿Qué es mejor hacer partition by por category\_id porque es numérico ó category\_name?

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:



**Respuesta:**

**select** *o*.customer\_id , *o*.order\_date,

**min**(*o*.order\_date) **over** (**partition** **by** o.customer\_id) **as** *earliestorderdate*

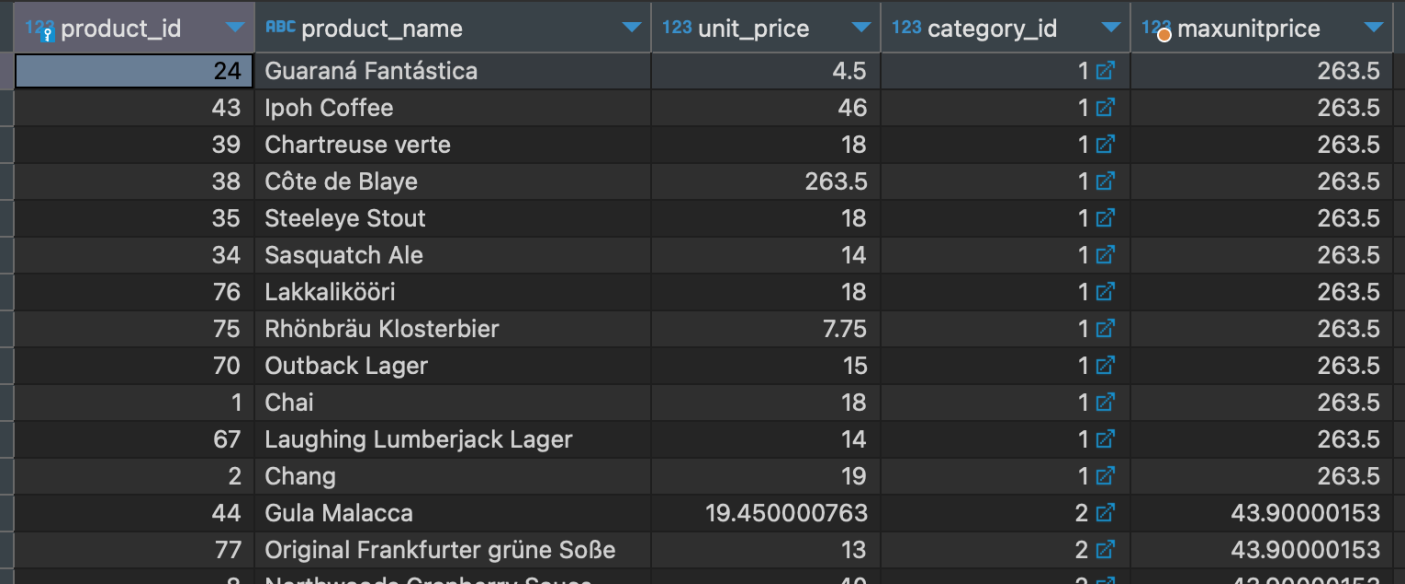
**from** orders *o* **inner** **join** customers *c*

**on** *c*.customer\_id = *o*.customer\_id

MAX

1. 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:



**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*

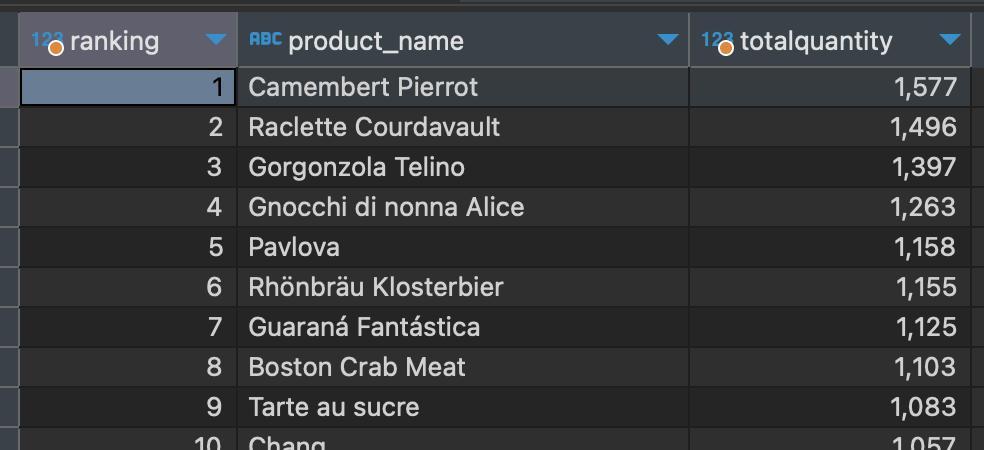
**from** products *p* **inner** **join** categories *c*

**on** *c*.category\_id = *p*.category\_id

Row number

1. Obtener el ranking de los productos más vendidos

Print:



**Respuesta:**

**select** **row\_number**() **over** (**order** **by** **sum**(od.quantity) **desc**) **as** *ranking*, *p*.product\_name,

**sum**(*od*.quantity) **as** *totalquantity*

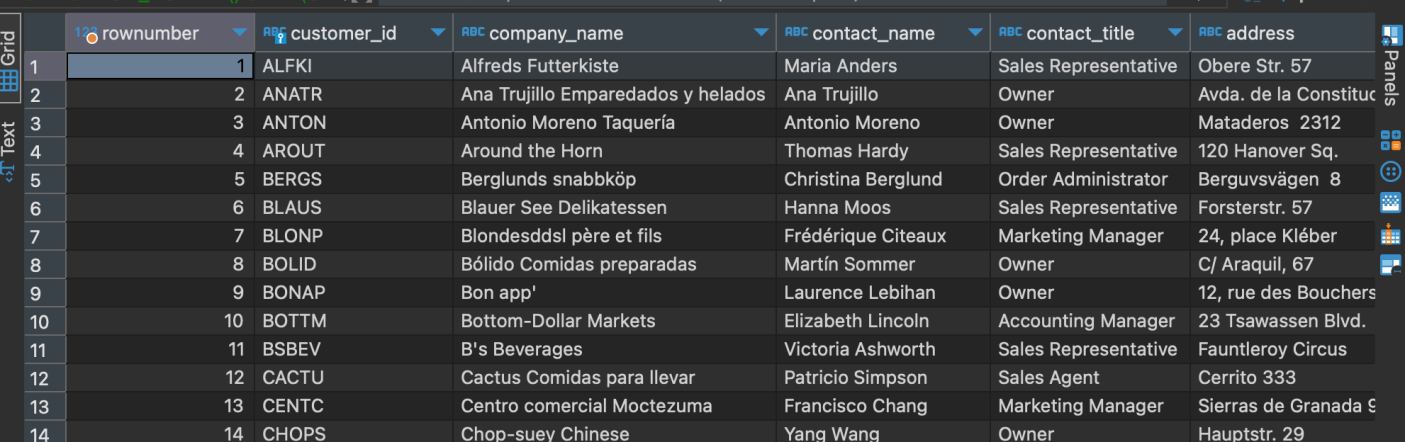
**from** products *p* **inner** **join** order\_details *od*

**on** *p*.product\_id = *od*.product\_id

**group** **by** *p*.product\_id

1. Asignar numeros de fila para cada cliente, ordenados por customer\_id

Print:



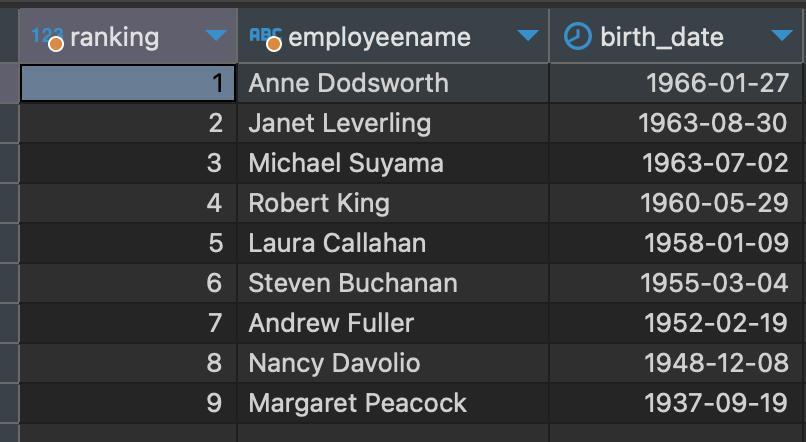
**Respuesta:**

**select** **row\_number**() **over** (**order** **by** c.customer\_id **asc**) **as** *rownumber*,\*

**from** customers *c*

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

Print:



**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*,

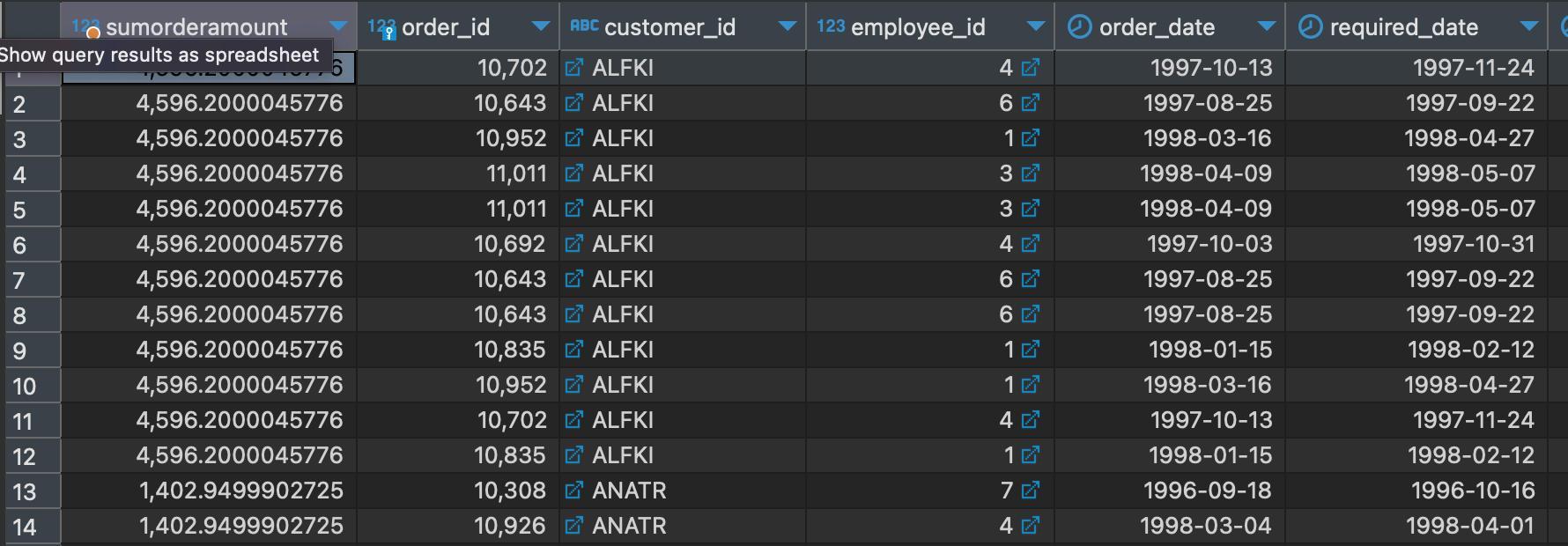
*e*.first\_name || ' ' || *e*.last\_name **as** *employeename*, *e*.birth\_date

***from*** *employees e*

SUM

1. Obtener la suma de venta de cada cliente

Print:



**Respuesta:**

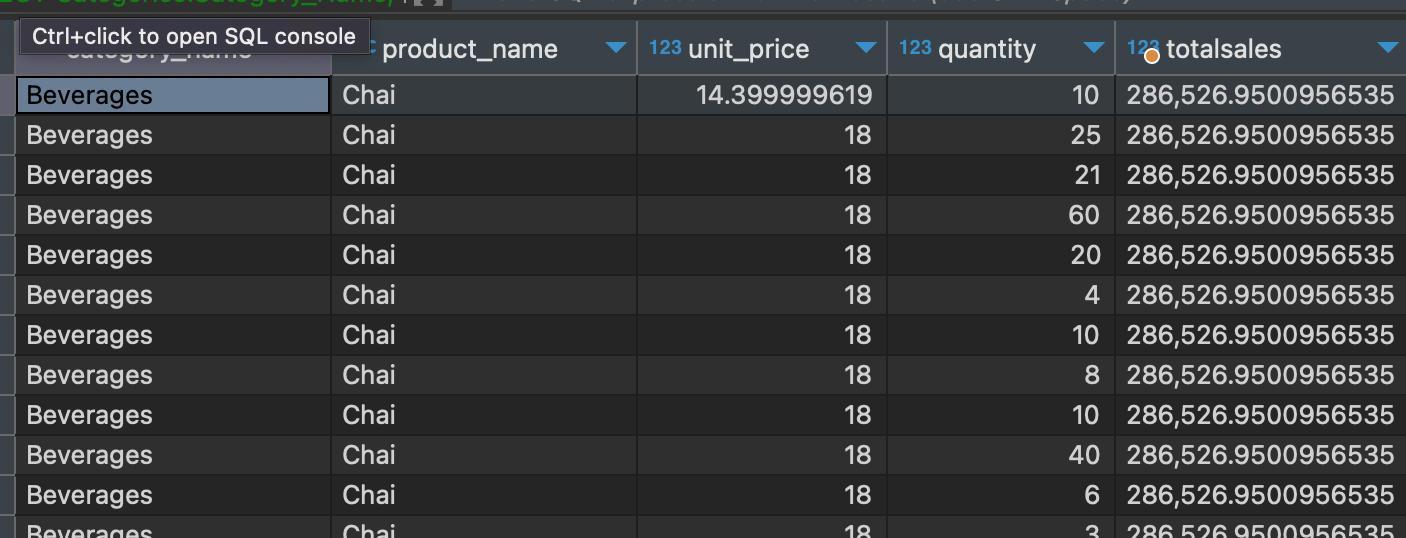
**select** **sum**(*od*.quantity \* *od*.unit\_price) **over** (**partition** **by** customer\_id) **as** *sumorderamount*, \*

**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:



**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*

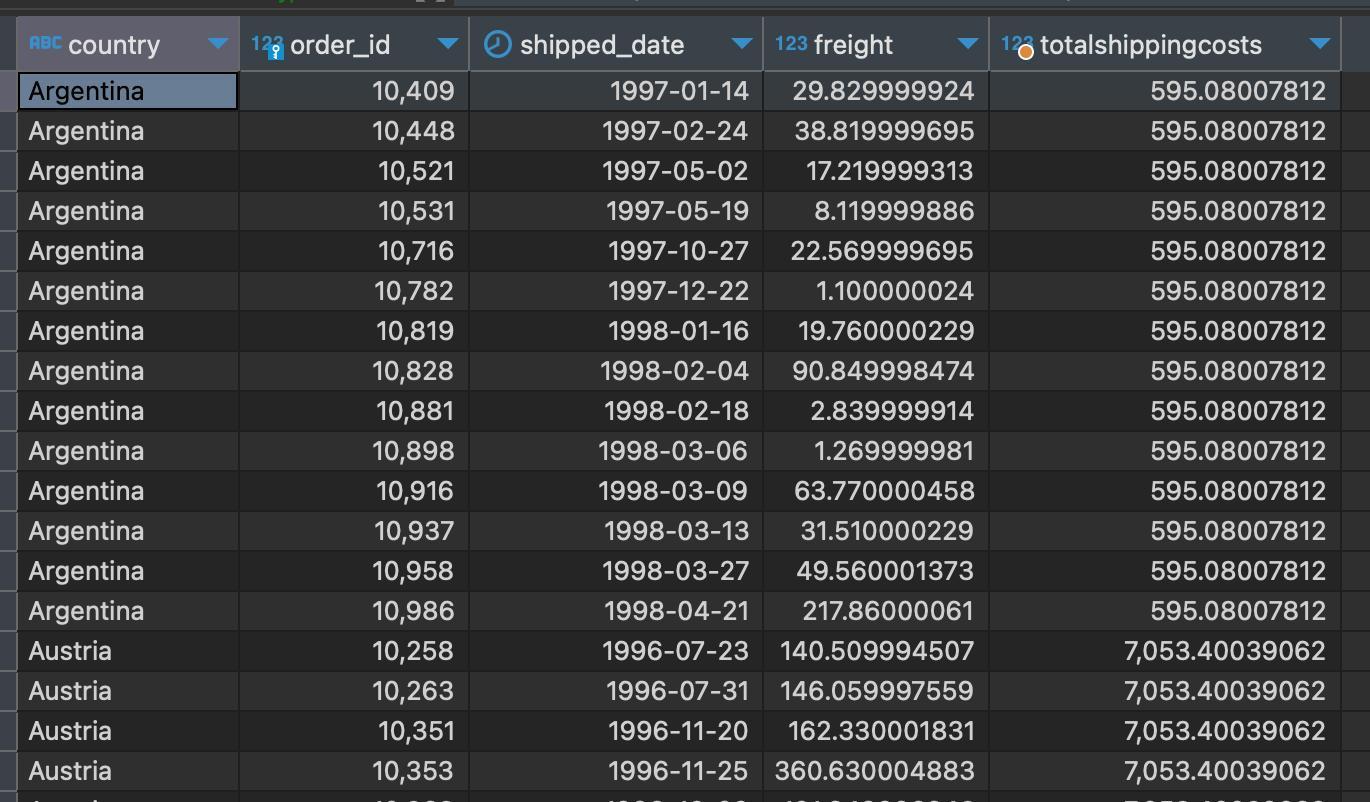
**on** *od*.product\_id = *p*.product\_id **inner** **join** categories *c*

**on** *c*.category\_id = *p*.category\_id

**order** **by** *c*.category\_id , *p*.product\_name

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

Print:



**Respuesta:**

**select** *o*.ship\_country , *o*.order\_id , *o*.shipped\_date , *o*.freight ,

**sum**(*o*.freight ) **over** (**partition** **by** o.ship\_country) **as** *totalshippingcosts*

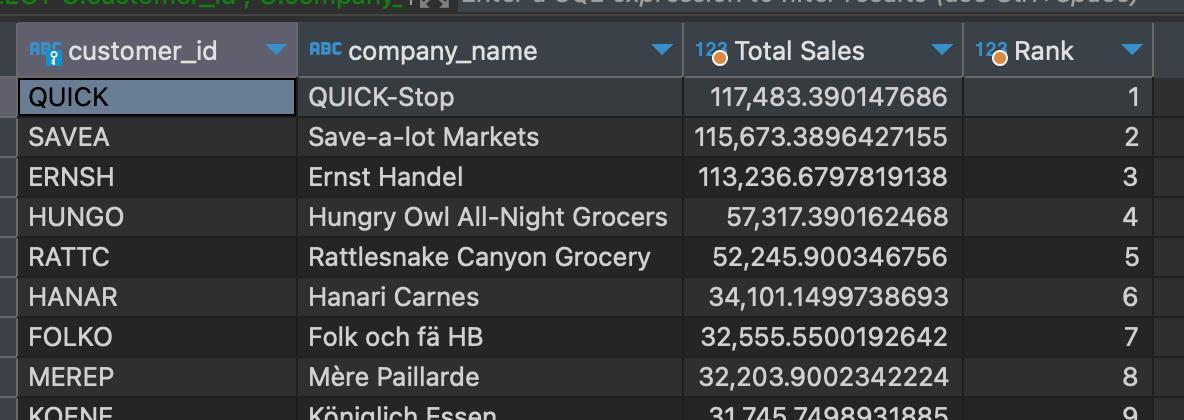
**from** orders *o*

**order** **by** *o*.ship\_country, *o*.order\_id **asc**

RANK

12.Ranking de ventas por cliente

Print:



**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*

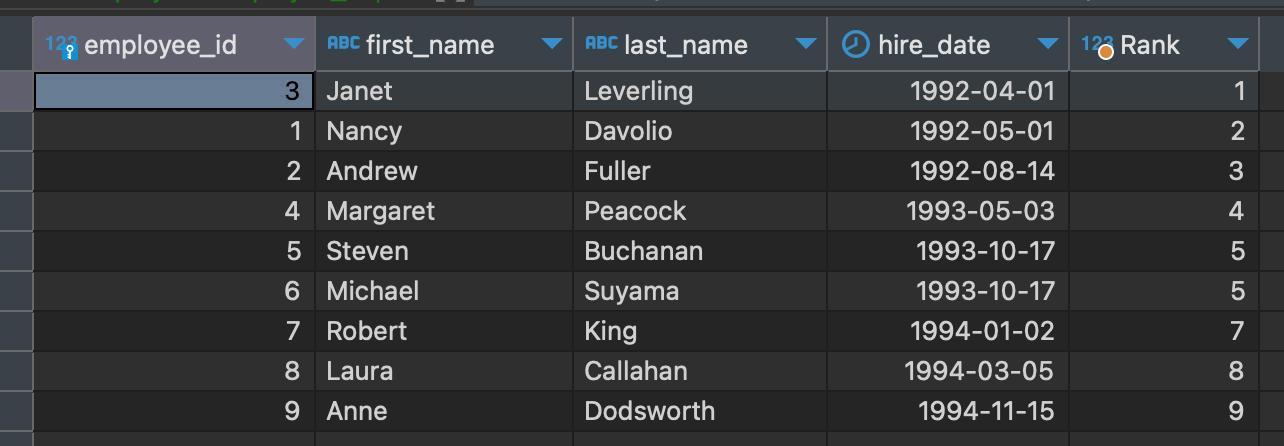
**on** *o*.order\_id = *od*.order\_id **inner** **join** customers *c*

**on** *c*.customer\_id = *o*.customer\_id

**group** **by** *o*.customer\_id , *c*.company\_name

13.Ranking de empleados por fecha de contratación

Print:



**Respuesta:**

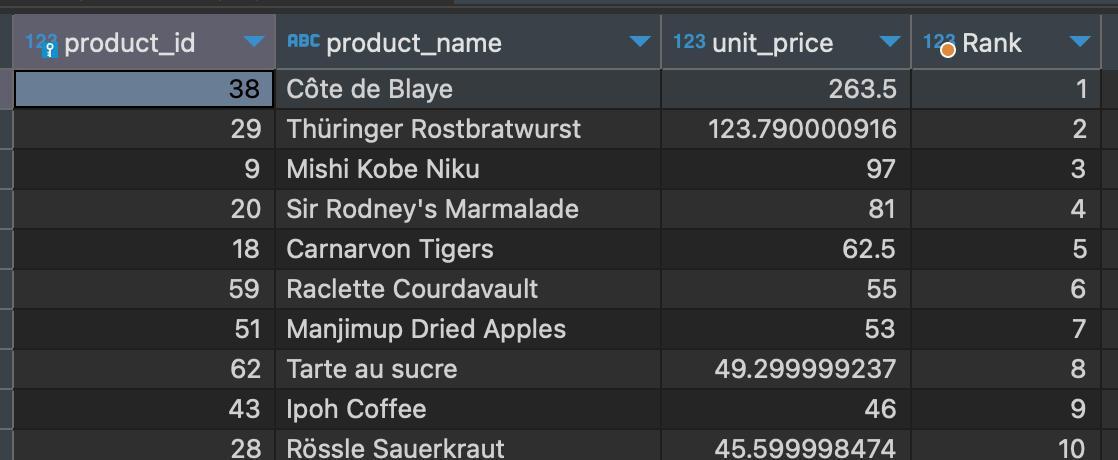
**select** *e*.employee\_id , *e*.first\_name , *e*.last\_name , *e*.hire\_date ,

**rank**() **over** (**order** **by** e.hire\_date **asc**)

**from** employees *e*

14.Ranking de productos por precio unitario

Print:



**Respuesta:**

**select** *p*.product\_id , *p*.product\_name , *p*.unit\_price ,

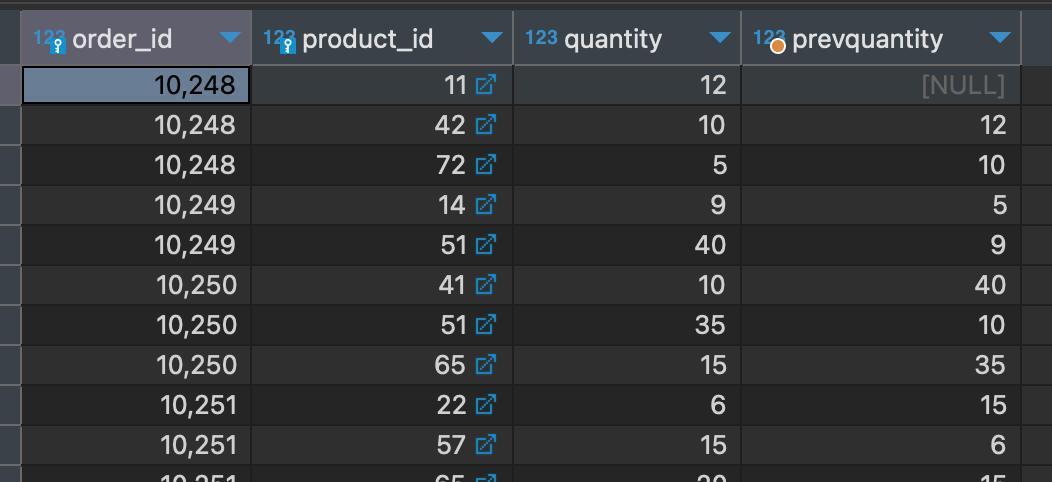
**rank**() **over** (**order** **by** p.unit\_price **desc**)

**from** products *p*

LAG

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

Print:



**Respuesta:**

**select** *o*.order\_id , *od*.product\_id , *od*.quantity ,

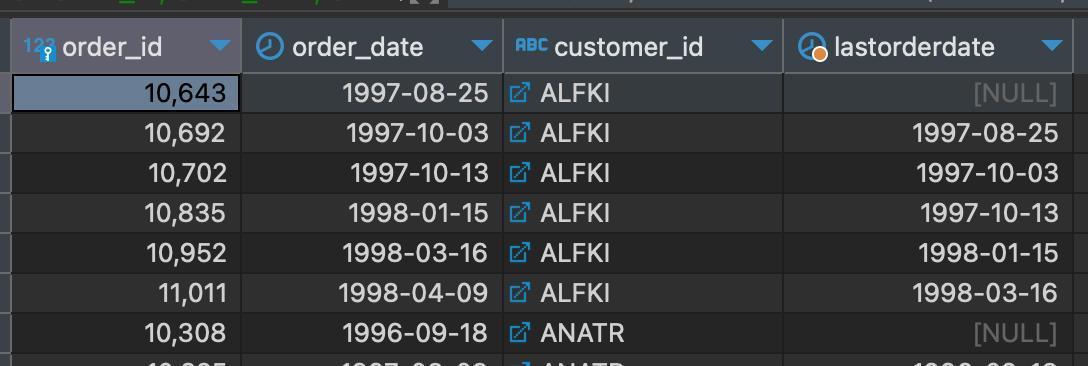
**lag**(*od*.quantity) **over** (**order** **by** o.order\_id ) **as** *prevquantity*

**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:



**Respuesta:**

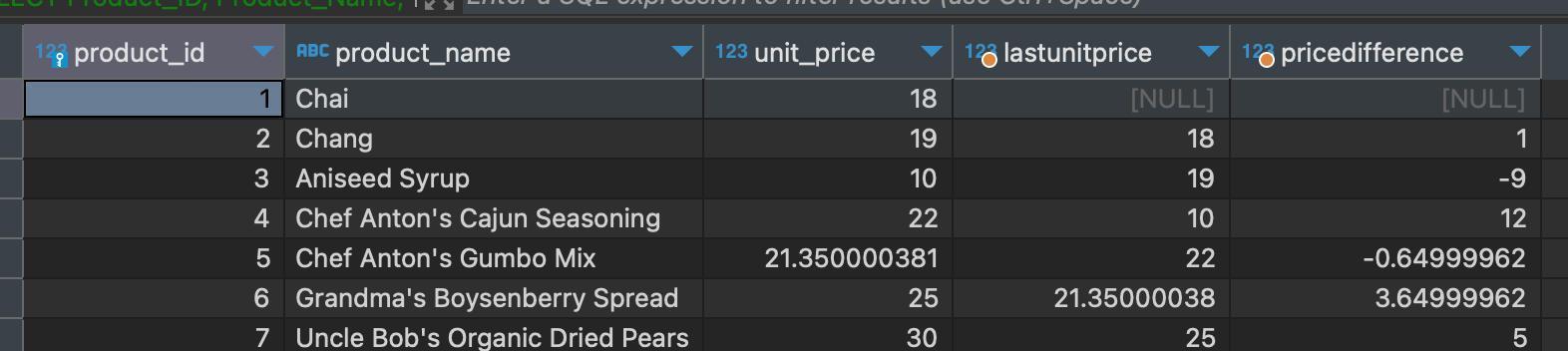
**select** *o*.order\_id, *o*.order\_date, *o*.customer\_id,

**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:



**Respuesta:**

**select** *p*.product\_id, *p*.product\_name , *p*.unit\_price,

**lag**(*p*.unit\_price ) **over** (**order** **by** p.product\_id ) **as** *lastunitprice*,

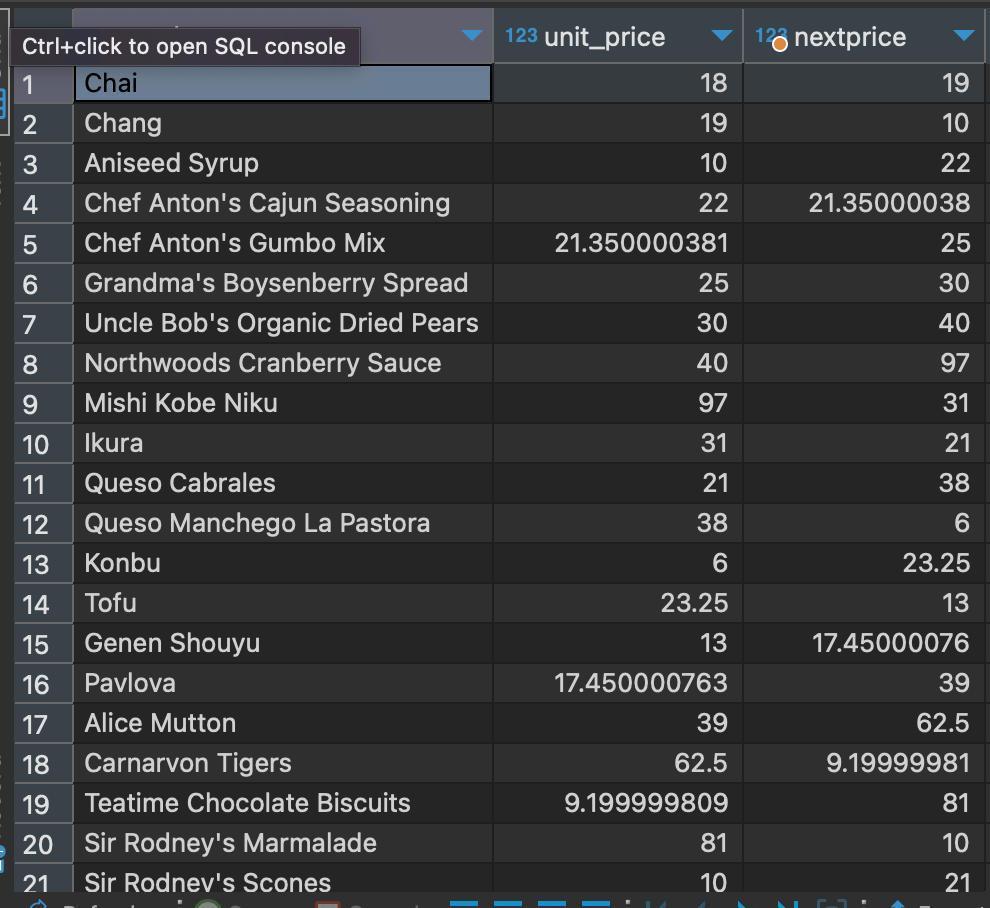
(*p*.unit\_price - **lag**(*p*.unit\_price ) **over** (**order** **by** p.product\_id) ) **as** *pricedifference*

**from** products *p*

LEAD

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

Print:



**Respuesta:**

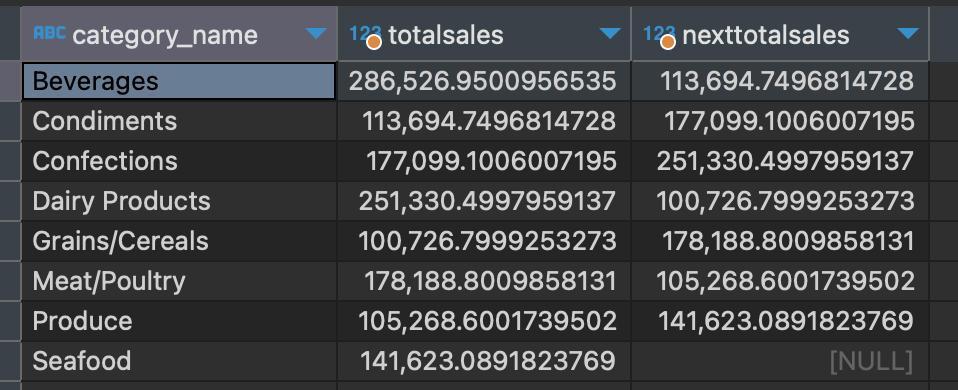
**select** *p*.product\_name ,*p*.unit\_price ,

**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:



**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?