

Tarea 4: Álgebra Relacional

Integrantes: Arcos Morales Ramón : 319541478
 Casarrubias Casarrubias Victor Manuel : 421003581
 Castillo Hernández Eduardo : 420003557
 López Asano Miguel Akira : 320219089
 Rivera Lara Sandra Valeria : 320039823

Fecha: 24/09/2024

Ejercicio 1. Cardinalidad de la consulta:

Considera las siguientes relaciones:

R =	A	B
	1	x
	2	y
	2	z
	3	x
	9	a

S =	B	C	D
	x	0	3
	y	2	1
	y	3	3
	w	3	0
	y	4	2

Para las siguientes expresiones de **álgebra relacional**, completa la tabla con el número de tuplas que cada una de ellas produce utilizando las relaciones **R** y **S**. Deberás indicar las **tablas resultantes** en cada caso.

Expresión	Cardinalidad del resultado
$R \times S$	25
$R \bowtie_{D > A} S$	7
$R \bowtie S$	7
$R \ltimes S$	6
$R \bowtie_{A = D} S$	5
$\rho_C \leftarrow A(R) \bowtie S$	1
$\pi_B(R) - \pi_B(\sigma_{C \geq 2}(S))$	3
$\pi_A(R) \cap \rho_A \leftarrow D(\pi_D(S))$	3
$\pi_D(S) \bowtie_{S.D > R.A} R$	7
$\gamma_A; count(B) \rightarrow t(R \ltimes S)$	5

1. Expresión $R \times S$

A	R.B	S.B	C	D
1	x	x	0	3
1	x	y	2	1
1	x	y	3	3
1	x	w	3	0
1	x	y	4	2
2	y	x	0	3
2	y	y	2	1
2	y	y	3	3
2	y	w	3	0
2	y	y	4	2
2	z	x	0	3
2	z	y	2	1
2	z	y	3	3
2	z	w	3	0
2	z	y	4	2
3	x	x	0	3
3	x	y	2	1
3	x	y	3	3
3	x	w	3	0
3	x	y	4	2
9	a	x	0	3
9	a	y	2	1
9	a	y	3	3
9	a	w	3	0
9	a	y	4	2

Tabla 1: $R \times S$ 2. Expresión $R \bowtie_{D > A} S$

A	R.B	S.B	C	D
1	x	x	0	3
1	x	y	3	3
1	x	y	4	2
2	y	x	0	3
2	y	y	3	3
2	z	x	0	3
2	z	y	3	3

Tabla 2: $R \bowtie_{D > A} S$ 3. Expresión $R \Join S$

R.A	R.B	S.C	S.D
1	x	0	3
2	y	2	1
2	y	3	3
2	y	4	2
2	z	null	null
3	x	0	3
9	a	null	null

Tabla 3: $R \Join S$

4. Expresión $R \bowtie S$

A	B	C	D
1	x	0	3
3	x	0	3
2	y	2	1
2	y	3	3
null	w	3	0
2	y	4	2

Tabla 4: $R \bowtie S$

5. Expresión $R \bowtie A = D S$

A	R.B	S.B	C	D
1	x	y	2	1
2	y	y	4	2
2	z	y	4	2
3	x	x	0	3
3	x	y	3	3

Tabla 5: $R \bowtie A = D S$

6. Expresión $\rho C \leftarrow A (R) \bowtie S$

C	B	D
2	y	1

Tabla 6: $\rho C \leftarrow A (R) \bowtie S$

7. Expresión $\pi B (R) - \pi B (\sigma_{C \geq 2} (S))$

B
x
z
a

Tabla 7: $\pi B (R) - \pi B (\sigma_{C \geq 2} (S))$

8. Expresión $\pi A (R) \cap \rho A \leftarrow D(\pi D (S))$

R.A
1
2
3

Tabla 8: $\pi A (R) \cap \rho A \leftarrow D(\pi D (S))$

9. Expresión $\pi D (S) \bowtie S.D > R.A R$

D	A	R.B
3	1	x
3	1	x
2	1	x
3	2	y
3	2	y
3	2	z
3	2	z

Tabla 9: $\pi D (S) \bowtie S.D > R.A R$

10. Expresión $\gamma A; count(B) \rightarrow t (R \bowtie S)$

A	B
1	1
2	4
3	1
9	1
<i>null</i>	1

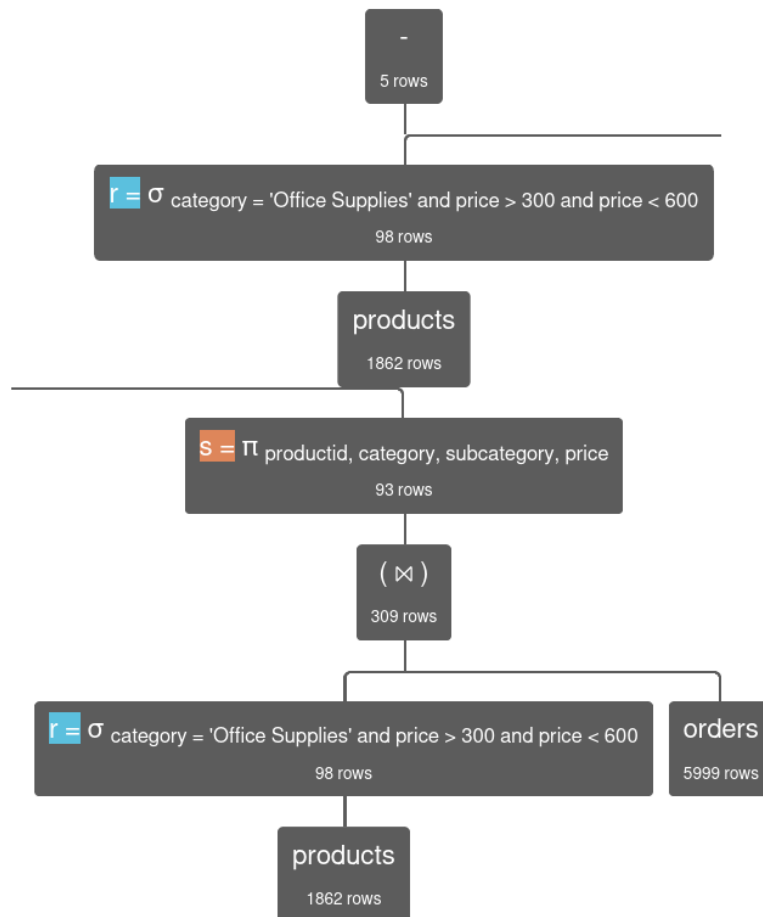
Tabla 10: $\gamma A; count(B) \rightarrow t (R \bowtie S)$

- (b) Obtener una relación de los productos que pertenecen a la categoría **Office Supplies** con **precio mayor de \$300 y menor de \$600**, pero que **no hayan sido solicitados** en ninguna orden.

```

1 r = σ category = 'Office Supplies' ∧ price > 300 ∧ price < 600
  (products)
2 s = π productid, category, subcategory, price (r ⋈ orders)
3 r - s

```



products.productid	products.category	products.subcategory	products.price
'OFF-PA-10001593'	'Office Supplies'	'Paper'	563.4
'OFF-PA-10002109'	'Office Supplies'	'Paper'	505.18
'OFF-PA-10003205'	'Office Supplies'	'Paper'	478.48
'OFF-AP-10003278'	'Office Supplies'	'Appliances'	597.13
'OFF-AR-10003896'	'Office Supplies'	'Art'	462.56

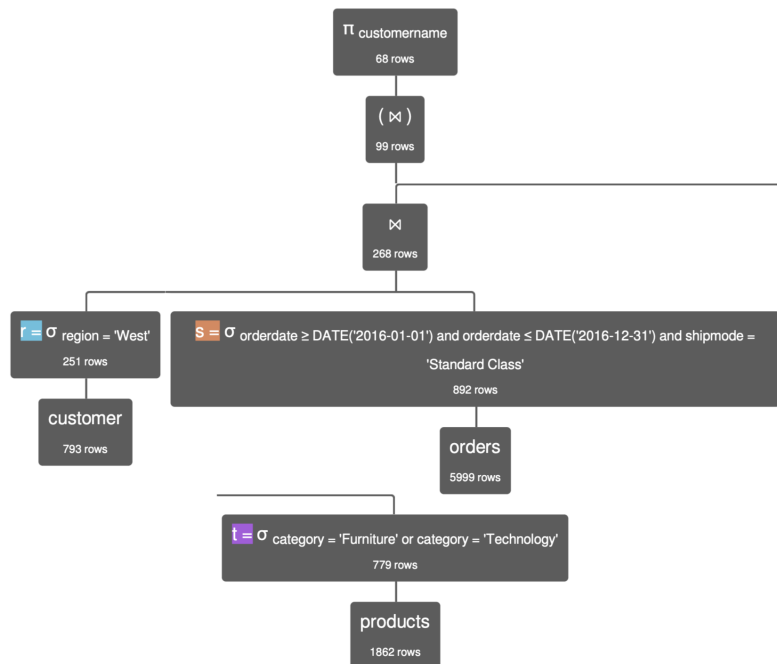
- (c) Obtener el **nombre** de todos los clientes que vivan en la región **West** y hayan solicitado productos de las categorías **Technology** o **Furniture**. El pedido debió de solicitarse en **2106** y el modo de envío debe ser **Standard Class**.

Se realiza la búsqueda de los pedidos tomando como año 2016, pues se considera como fe de errata el año 2106.

```

1 r = σ region = 'West' customer
2 s = σ orderdate ≥ date('2016-01-01') ∧ orderdate ≤ date('2016-12-31') ∧ shipmode = 'Standard Class' orders
3 t = σ category = 'Furniture' ∨ category = 'Technology' products
4 π customername (r ⋈ s ⋈ t)

```



$$\pi_{customername} \left(\left(\sigma_{region = 'West'} customer \bowtie \sigma_{orderdate \geq DATE('2016-01-01') \wedge orderdate \leq DATE('2016-12-31') \wedge shipmode = 'Standard Class'} orders \right) \bowtie \sigma_{category = 'Furniture' \vee category = 'Technology'} products \right)$$

customer.customername

'Brosina Hoffman'

'Zuschuss Donatelli'

'Emily Burns'

'Eric Hoffmann'

'Lena Creighton'

'Jonathan Doherty'

'Nora Paige'

'Chad Sievert'

'Jennifer Braxton'

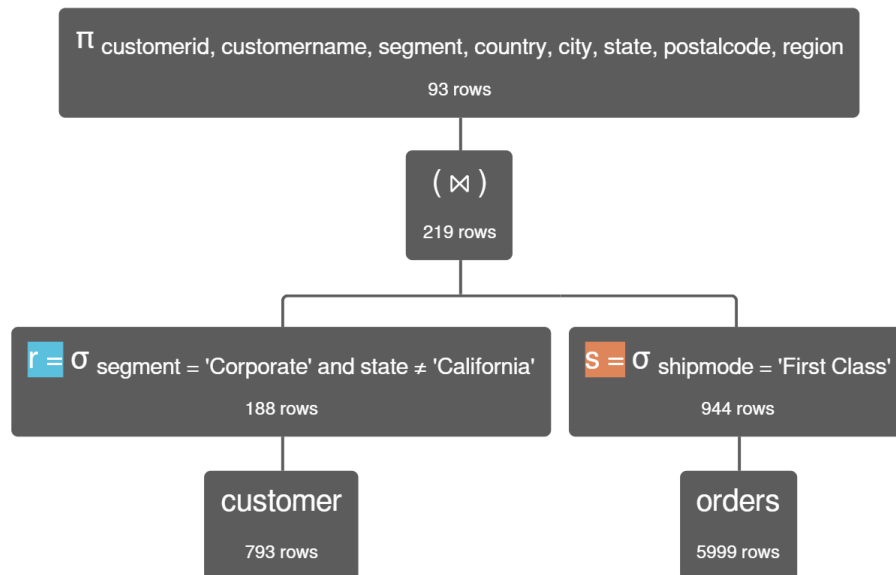
'Jonathan Howell'

- (d) Toda la información de los clientes del segmento **Corporate** que realizaron una orden con modo de envío **First Class** y que no viven en **California**.

```

1 r = σ segment='Corporate' ∧ state≠'California' customer
2 s = σ shipmode='First Class' orders
3 π customerid, customername, segment, country, city, state, postcode, region (r ⋈ s)

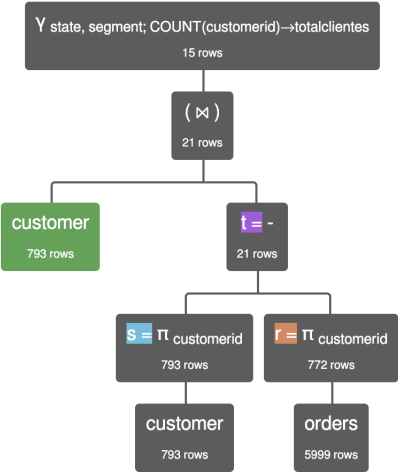
```



customer.customerid	customer.customername	customer.segment	customer.country	customer.city	customer.state	customer.postalcode	customer.region
'KB-16585'	'Ken Black'	'Corporate'	'United States'	'Fremont'	'Nebraska'	'68025'	'Central'
'GH-14485'	'Gene Hale'	'Corporate'	'United States'	'Richardson'	'Texas'	'75080'	'Central'
'LC-16930'	'Linda Cazamias'	'Corporate'	'United States'	'Naperville'	'Illinois'	'60540'	'Central'
'ES-14080'	'Erin Smith'	'Corporate'	'United States'	'Melbourne'	'Florida'	'32935'	'South'
'ON-18715'	'Odella Nelson'	'Corporate'	'United States'	'Eagan'	'Minnesota'	'55122'	'Central'
'KB-16600'	'Ken Brennan'	'Corporate'	'United States'	'Memphis'	'Tennessee'	'38109'	'South'
'SC-20770'	'Stewart Carmichael'	'Corporate'	'United States'	'Houston'	'Texas'	'77041'	'Central'
'CV-12805'	'Cynthia Voltz'	'Corporate'	'United States'	'Aurora'	'Colorado'	'80013'	'West'
'GG-14650'	'Greg Guthrie'	'Corporate'	'United States'	'Urbandale'	'Iowa'	'50322'	'Central'
'DW-13585'	'Dorothy Wardle'	'Corporate'	'United States'	'Bloomington'	'Illinois'	'61701'	'Central'

(e) Obtener el **estado**, **segmento** y el **total de clientes** que no han solicitado **ninguna orden**.

```
1 r= π customerid orders
2 s = π customerid customer
3 t = s-r
4 γ state,segment;count(customerid)→totalclientes (customer⋈t)
5
```



customer.state	customer.segment	totalclientes
'California'	'Consumer'	3
'Illinois'	'Home Office'	1
'Ohio'	'Corporate'	1
'New York'	'Corporate'	3
'Texas'	'Home Office'	1
'Delaware'	'Corporate'	2
'Tennessee'	'Corporate'	1
'California'	'Corporate'	1
'Pennsylvania'	'Corporate'	1
'Minnesota'	'Home Office'	1

- (f) Una lista que muestre la **región**, el **estado** y el total de clientes que se tienen, considerando que los clientes deben haber realizado órdenes con **al menos 6 productos** durante **2014 o 2015**. **Ordenar** la información por **región** y **estado**.

```

1 r = σ orderdate ≥ date('2014-01-01') and orderdate ≤
   date('2015-12-31') (σ quantity ≥ 6 (customer ⋈ orders))
2 x = γ region,state; count(customerid) → total_clients r
3 τ region,state x

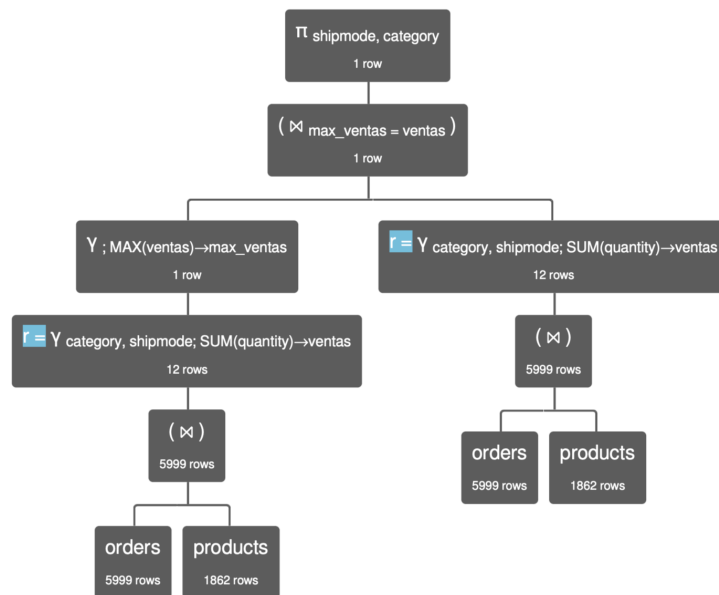
```



customer.region	customer.state	total_clients
'Central'	'Illinois'	20
'Central'	'Indiana'	8
'Central'	'Iowa'	4
'Central'	'Michigan'	17
'Central'	'Minnesota'	9
'Central'	'Missouri'	3
'Central'	'Nebraska'	4
'Central'	'Oklahoma'	4
'Central'	'Texas'	38
'Central'	'Wisconsin'	3

(g) Obtener el **modo de envío** y **categoría** que **más productos** ha vendido.

```
1 r = γ category, shipmode; sum(quantity) -> ventas (orders ⋈ products)
2 π shipmode, category (γ max(ventas) -> max_ventas r ⋈ max_ventas = ventas r)
```



$\pi_{\text{shipmode, category}} \left(\gamma_{\text{MAX(ventas) \rightarrow max_ventas}} \gamma_{\text{category, shipmode; SUM(quantity) \rightarrow ventas}} (\text{orders} \bowtie \text{products}) \bowtie \text{max_ventas = ventas} \gamma_{\text{category, shipmode; SUM(quantity) \rightarrow ventas}} (\text{orders} \bowtie \text{products}) \right)$

Tiempo de consulta 2 ms

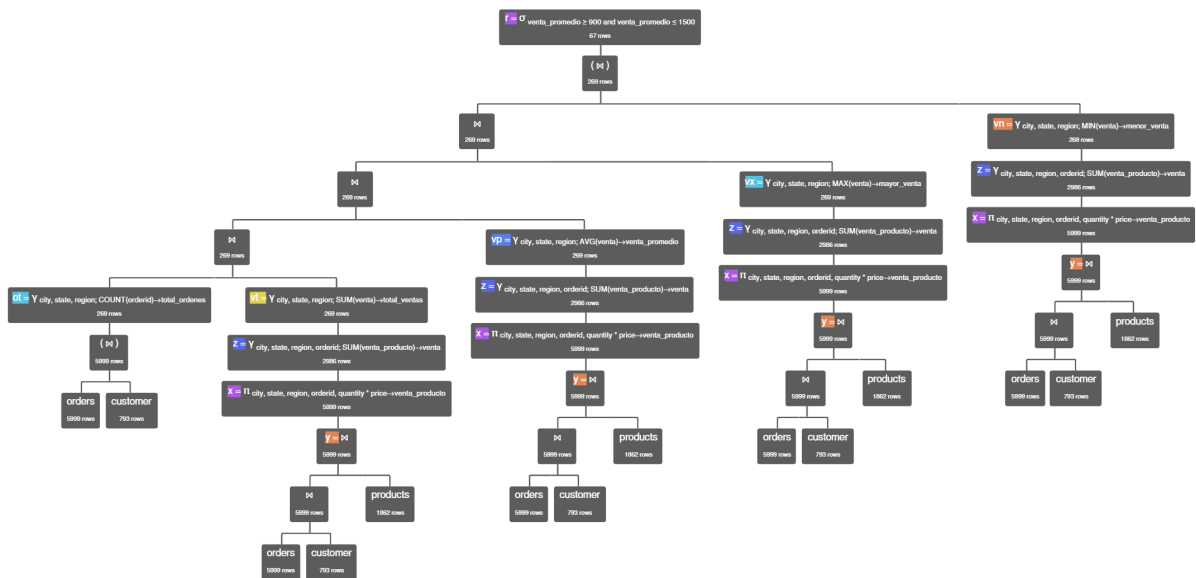
orders.shipmode	products.category
'Standard Class'	'Office Supplies'

- (h) Una tabla con la **venta promedio**, **venta total**, **mayor venta**, **menor venta**, y **total de órdenes**, por **región**, **estado** y **ciudad**. La **venta promedio** debe estar entre \$900 y \$1,500.

```

1 /* Obtener la venta de cada orden */
2 y = orders ⋈ customer ⋈ products
3 x = π city, state, region,orderid, venta_producto ← quantity*price y
4 z = γ city, state, region,orderid; sum(venta_producto) → venta x
5 /* Obtener avg, max, min, total y ordenes */
6 ot = γ city, state, region; count(orderid) → total_ordenes (orders ⋈ customer)
7 vt = γ city, state, region; sum(venta) → total_ventas z
8 vp = γ city, state, region; avg(venta) → venta_promedio z
9 vx = γ city, state, region; max(venta) → mayor_venta z
10 vn = γ city, state, region; min(venta) → menor_venta z
11 /* Mostrar resultado con restricciones dadas */
12 r = σ venta_promedio ≥ 900 ∧ venta_promedio ≤ 1500 (ot ⋈ vt ⋈ vp ⋈ vx ⋈ vn)
13 r

```



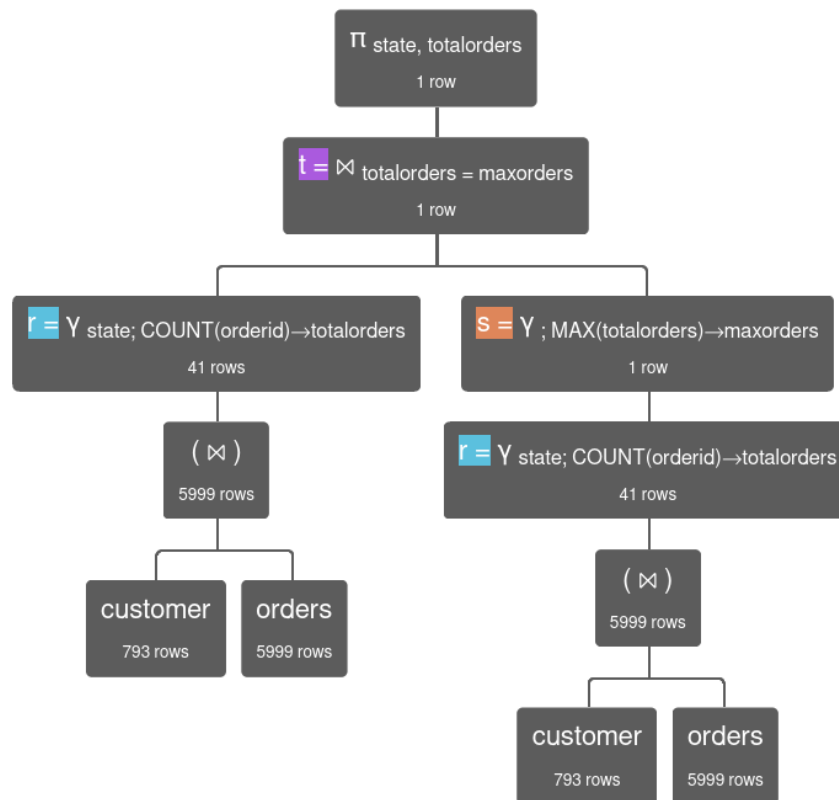
customer.city	customer.state	customer.region	total_ordenes	total_ventas	venta_promedio	mayor_venta	menor_venta
'Fort Worth'	'Texas'	'Central'	36	25551.49	1216.737619047619	13466.880000000001	26.76
'West Jordan'	'Utah'	'West'	2	2120.96	1060.48	2009.96	111
'Orem'	'Utah'	'West'	26	15460.620000000003	909.4482352941178	2872.7400000000002	45.42
'Melbourne'	'Florida'	'South'	14	4782.180000000001	956.4360000000003	3743.6800000000003	62.14
'Dover'	'Delaware'	'East'	14	10153.369999999999	1128.1522222222222	3798.5499999999997	22.959999999999997
'Chicago'	'Illinois'	'Central'	174	132752.07000000004	1427.441612903226	14502.82	1.68
'Columbia'	'South Carolina'	'South'	17	11920.630000000001	1324.5144444444445	6005.58	19.46
'Minneapolis'	'Minnesota'	'Central'	34	20877.899999999998	1159.8833333333332	3902.46	35.75
'Portland'	'Oregon'	'West'	12	9233.449999999999	1025.9388888888889	7080.66	12.8
'Pasadena'	'California'	'West'	8	5674.44	1134.888	2397.3599999999997	20.7

- (i) El estado que ha realizado la **mayor cantidad de órdenes**. Se debe mostrar también el **total de órdenes** que haya entregado.

```

1 r = γ state; count(orderid) → totalorders (customer ⋈ orders)
2 s = γ max(totalorders) → maxorders (r)
3 t = r ⋈ totalorders = maxorders s
4 π state, totalorders (t)

```



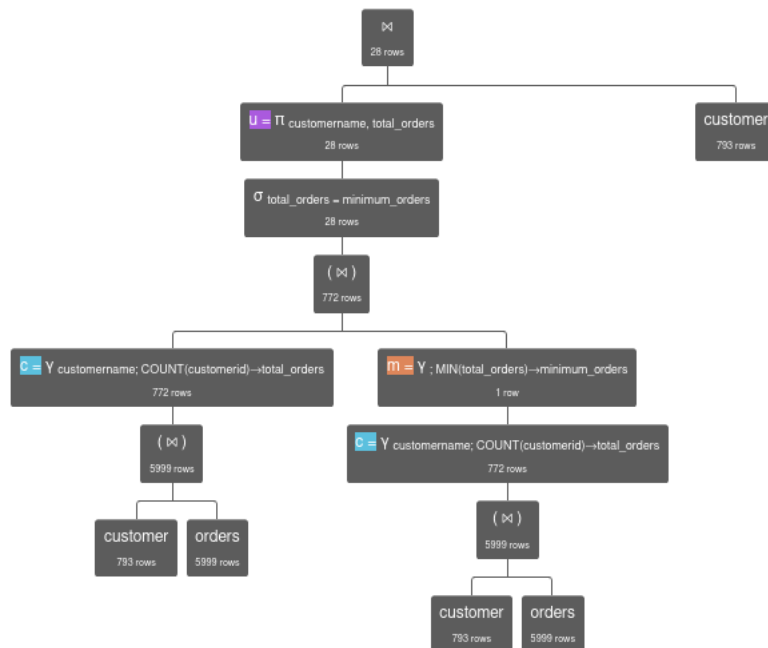
customer.state	totalorders
'California'	1200

- (j) La información del cliente que **menos órdenes** haya efectuado. Mostrar el **número de órdenes** que ha realizado.

```

1 c = γ customername; count(customerid) -> total_orders
  (customer ⋈ orders)
2 m = γ; min(total_orders) -> minimum_orders c
3 u = π customername, total_orders (σ total_orders =
  minimum_orders (c ⋈ m))
4 u ⋈ customer |

```



customer.customername	total_orders	customer.customerid	customer.segment	customer.c
'Henry MacAllister'	1	'HM-14980'	'Consumer'	'United St
'Katherine Hughes'	1	'KH-16360'	'Consumer'	'United St
'Thomas Brumley'	1	'TB-21190'	'Home Office'	'United St
'Paul MacIntyre'	1	'PM-18940'	'Consumer'	'United St
'Jocasta Rupert'	1	'JR-15700'	'Consumer'	'United St
'George Zrebassa'	1	'GZ-14545'	'Corporate'	'United St
'Pauline Webber'	1	'PW-19030'	'Corporate'	'United St
'Bobby Odegard'	1	'BO-11425'	'Consumer'	'United St
'Pamela Stobb'	1	'PS-18760'	'Consumer'	'United St
'Bobby Trafton'	1	'BT-11440'	'Consumer'	'United St

Operaciones de mantenimiento de datos: borrado, inserción y actualización

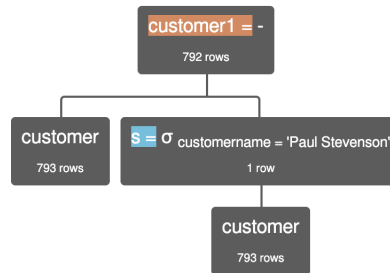
- (a) Borrar toda la información del cliente
- Paul Stevenson**
- .

Borrar la información del cliente

```

1 s = σ customername = 'Paul Stevenson' customer
2 customer1 = customer - s
3 customer1

```

customer - σ customername = 'Paul Stevenson' customer

Execution time: 1 ms

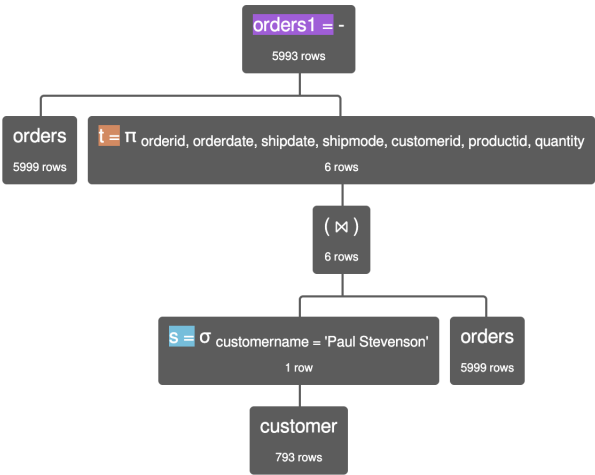
customer.customerid	customer.customername	customer.segment	customer.country	customer.city
'CG-12520'	'Claire Gute'	'Consumer'	'United States'	'Hendershot'
'DV-13045'	'Darrin Van Huff'	'Corporate'	'United States'	'Los Angeles'
'SO-20335'	'Sean ODonnell'	'Consumer'	'United States'	'Fort Lauderdale'
'BH-11710'	'Brosina Hoffman'	'Consumer'	'United States'	'Los Angeles'
'AA-10480'	'Andrew Allen'	'Consumer'	'United States'	'Concord'
'IM-15070'	'Irene Maddox'	'Consumer'	'United States'	'Seattle'
'HP-14815'	'Harold Pawlan'	'Home Office'	'United States'	'Fort Worth'
'PK-19075'	'Pete Kriz'	'Consumer'	'United States'	'Madison'
'AG-10270'	'Alejandro Grove'	'Consumer'	'United States'	'West Jordan'

Borrar las órdenes del cliente

```

5 t = π orderid,orderdate,shipdate,shipmode,customerid,productid,quantity (s ⋈ orders)
6 orders1 = orders - t
7 orders1

```



orders - π orderid, orderdate, shipdate, shipmode, customerid, productid, quantity (σ customername = 'Paul Stevenson' customer ⋈ orders)
Execution time: 3 ms

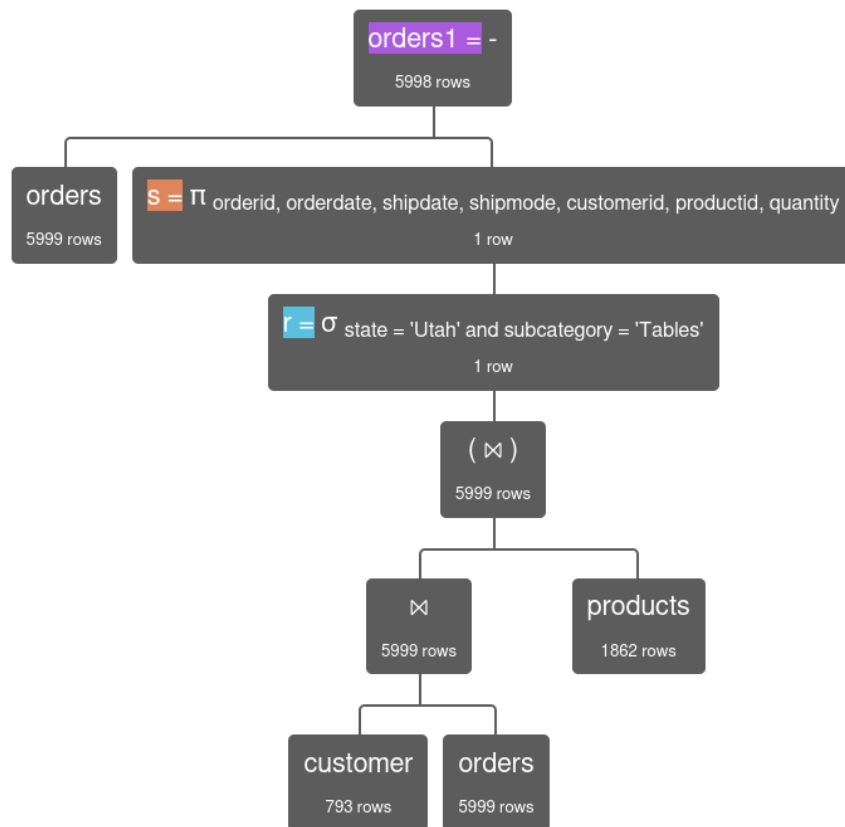
orders.orderid	orders.orderdate	orders.shipdate	orders.shipmode	orders.customerid	orders.
'CA-2016-152156'	2016-11-08	2016-11-11	'Second Class'	'CG-12520'	'FU 100
'CA-2016-152156'	2016-11-08	2016-11-11	'Second Class'	'CG-12520'	'FU 100
'CA-2016-138688'	2016-06-12	2016-06-16	'Second Class'	'DV-13045'	'OF 100
'US-2015-108966'	2015-10-11	2015-10-18	'Standard Class'	'SO-20335'	'FU 100
'US-2015-108966'	2015-10-11	2015-10-18	'Standard Class'	'SO-20335'	'OF 100
'CA-2014-115812'	2014-06-09	2014-06-14	'Standard Class'	'BH-11710'	'FU 100
'CA-2014-115812'	2014-06-09	2014-06-14	'Standard Class'	'BH-11710'	'OF 100
'CA-2014-115812'	2014-06-09	2014-06-14	'Standard Class'	'BH-11710'	'TE 100

(b) Borrar todas las órdenes de la ciudad **Utah** que tengan artículos de la subcategoría **Tables**.

```

1 r = σ state = 'Utah' ∧ subcategory = 'Tables'(customer ⋈ orders ⋈
  products)
2 s = π orderid, orderdate, shipdate, shipmode, customerid, productid,
  quantity (r)
3 orders1 = orders - s
4 orders1

```



orders.orderid	orders.orderdate	orders.shipdate	orders.shipmode
'CA-2016-152156'	2016-11-08	2016-11-11	'Second Class'
'CA-2016-152156'	2016-11-08	2016-11-11	'Second Class'
'CA-2016-138688'	2016-06-12	2016-06-16	'Second Class'
'US-2015-108966'	2015-10-11	2015-10-18	'Standard Class'
'US-2015-108966'	2015-10-11	2015-10-18	'Standard Class'
'CA-2014-115812'	2014-06-09	2014-06-14	'Standard Class'
'CA-2014-115812'	2014-06-09	2014-06-14	'Standard Class'
'CA-2014-115812'	2014-06-09	2014-06-14	'Standard Class'

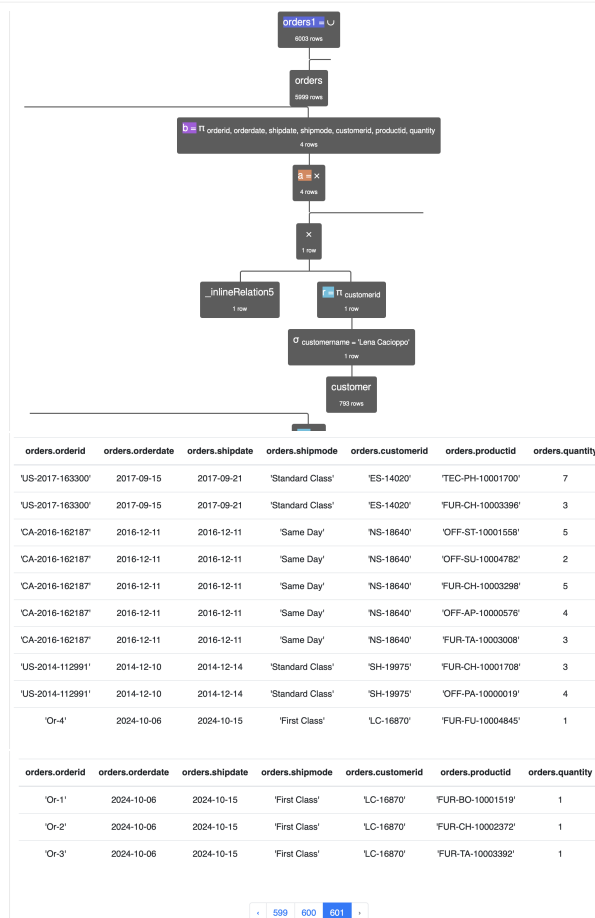
- (c) La clienta **Lena Cacioppo** compró un producto de cada subcategoría de **Furniture**. Deberás elegir los productos que desees e indicar como parte de esta consulta, la información que se agregará en cada caso.

Se decidió que los productos seleccionados fueran los de mayor precio. Además, la fecha de entrega será 9 días posterior a la fecha de pedido, la cantidad de producto por orden es 1 y el tipo de envío es First Class.

```

1 -- Se obtiene el customerid de Lena Cacioppo
2 r =  $\pi$  customerid  $\sigma$  customername = 'Lena Cacioppo' customer
3 -- Se agrupan las subcategorías de Furniture según el precio más alto
4 s =  $\gamma$  subcategory; max(price)-> max_price  $\sigma$  category = 'Furniture' products
5 -- Se obtiene la información de los productos más caros por subcategoría, de aquí se obtiene su id
6 t =  $\pi$  max_price s  $\bowtie$  max_price = price products
7 -- Por cada producto más caro de su correspondiente subcategoría se hace el producto cartesiano con un id.
8 v = {orderid:string
9     Or-1}  $\times$  ( $\pi$  productid  $\sigma$  subcategory = 'Bookcases' t)
10 w = {orderid:string
11     Or-2}  $\times$  ( $\pi$  productid  $\sigma$  subcategory = 'Chairs' t)
12 z = {orderid:string
13     Or-3}  $\times$  ( $\pi$  productid  $\sigma$  subcategory = 'Tables' t)
14 -- Se hace la unión de todos los idproducto con los idorder
15 y = {orderid:string
16     Or-4}  $\times$  ( $\pi$  productid  $\sigma$  subcategory = 'Furnishings' t)  $\cup$  v  $\cup$  w  $\cup$  z
17 -- Se genera la relacional de las 4 nuevas ordene, con la informcaión correspondiente
18 a = {orderid:date, shipdate:date, shipmode:string, quantity:number
19     2024-10-06, 2024-10-15, 'First Class' ,1
20 }  $\times$  r  $\times$  y
21 -- Se realiza la proyeccion para acomodar la información según se presenta en orders
22 b =  $\pi$  orderid,orderdate,shipdate,shipmode,customerid,productid,quantity a
23 -- Se realiza la unión
24 orders1 = orders  $\cup$  b
25 orders1

```

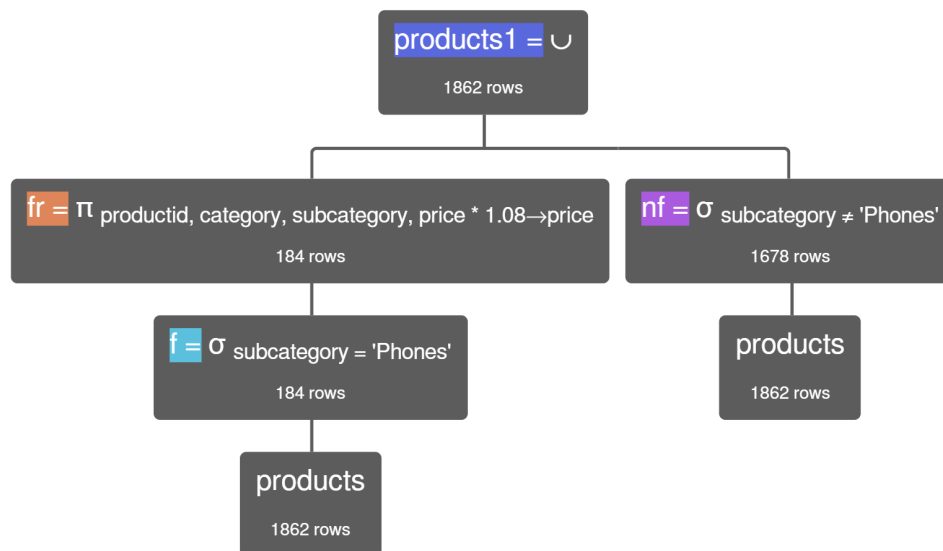


(d) Aumentar los precios de productos de la subcategoría **Phones** en un 8%.

```

1 f = σ subcategory='Phones' products
2 nf = σ subcategory≠'Phones' products
3 fr = π productid, category, subcategory, price ← price*1.08 f
4 products1 = fr ∪ nf
5 products1

```



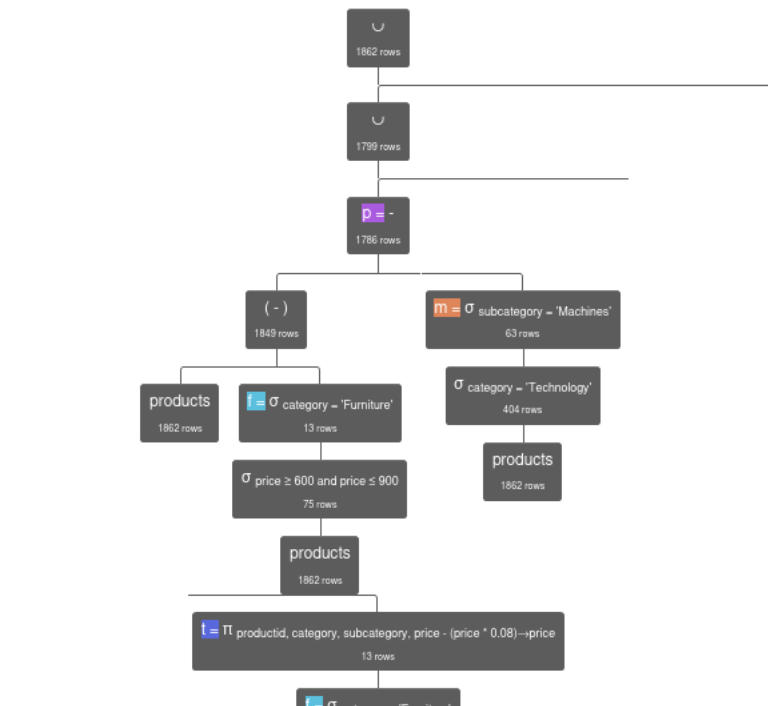
products.productid	products.category	products.subcategory	price
'TEC-PH-10002275'	'Technology'	'Phones'	979.7220000000001
'TEC-PH-10002033'	'Technology'	'Phones'	984.3336
'TEC-PH-10001949'	'Technology'	'Phones'	230.5584
'TEC-PH-10004977'	'Technology'	'Phones'	31.8276
'TEC-PH-10000486'	'Technology'	'Phones'	229.02480000000003
'TEC-PH-10004093'	'Technology'	'Phones'	400.8636
'TEC-PH-10003988'	'Technology'	'Phones'	48.6
'TEC-PH-10002447'	'Technology'	'Phones'	16.480800000000002
'TEC-PH-10002726'	'Technology'	'Phones'	230.16960000000003
'TEC-PH-10002844'	'Technology'	'Phones'	110.5488

- (e) Disminuir **8 %** los precios de los productos de la categoría **Furniture** cuyo precio sea de **\$600** a **\$900**. Aumentar en un **5 %** los precios de los productos de la categoría **Technology** y subcategoría **Machines**.

```

f = σ category = 'Furniture' (σ price ≥ 600 and price ≤ 900 products)
m = σ subcategory = 'Machines' (σ category = 'Technology' products)
p = (products - f) - m
t = π productid, category, subcategory, price ← price - (price*0.08) f
r = π productid, category, subcategory, price ← price + (price*0.05) m
p ∪ t ∪ r

```



products.productid	products.category	products.subcategory	products.price
'FUR-BO-10001798'	'Furniture'	'Bookcases'	261.96
'OFF-LA-10000240'	'Office Supplies'	'Labels'	14.62
'FUR-TA-10000577'	'Furniture'	'Tables'	957.58
'OFF-ST-10000760'	'Office Supplies'	'Storage'	22.37
'FUR-FU-10001487'	'Furniture'	'Furnishings'	48.86
'OFF-AR-10002833'	'Office Supplies'	'Art'	7.28
'TEC-PH-10002275'	'Technology'	'Phones'	907.15
'OFF-BI-10003910'	'Office Supplies'	'Binders'	18.5
'OFF-AP-10002892'	'Office Supplies'	'Appliances'	114.9
'FUR-TA-10001539'	'Furniture'	'Tables'	1706.18