



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CIENCIAS

FUNDAMENTOS DE BASES DE DATOS - 7094

T A R E A 4

EQUIPO:

DEL MONTE ORTEGA MARYAM MICHELLE - 320083527

SOSA ROMO JUAN MARIO - 320051926

CASTILLO HERNÁNDEZ ANTONIO - 320017438

ERIK EDUARDO GÓMEZ LÓPEZ - 320258211

JULIO CÉSAR ISLAS ESPINO - 320340594

FECHA DE ENTREGA:
14 DE OCTUBRE DE 2024

PROFESOR:
M. EN I. GERARDO AVILÉS ROSAS

AYUDANTES:
LUIS ENRIQUE GARCÍA GÓMEZ
KEVIN JAIR TORRES VALENCIA
RICARDO BADILLO MACÍAS
ROCÍO AYLIN HUERTA GONZÁLEZ



Tarea 4

Preguntas

1. Cardinalidad de la consulta

Considera las siguientes relaciones:

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

Tabla 1: R

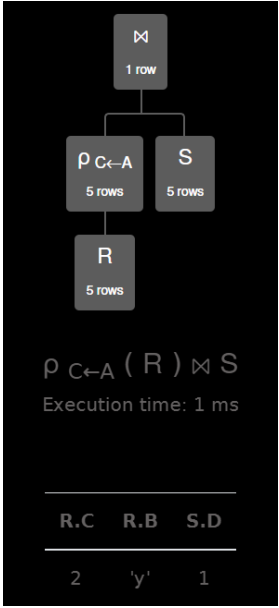
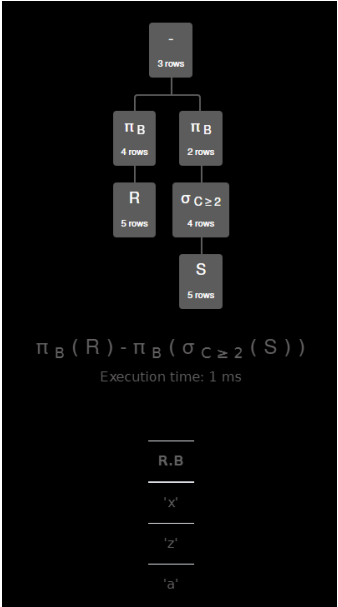
B	C	D
x	0	3
y	2	1
y	3	3
w	3	0
y	4	2

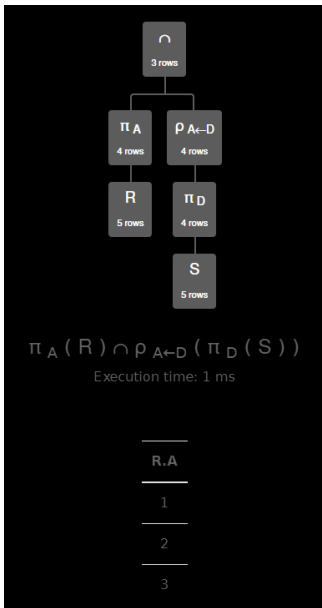
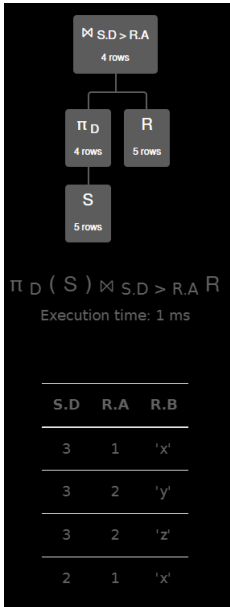
Tabla 2: S

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
$R \times S$	<p>Al ser un producto cartesiano se tienen $5 \times 5 = 25$ tuplas.</p> 

$R \bowtie_{D > A} S$	<p>Esta hace un join natural donde D es mayor que A, resulta en 7 tuplas :</p> <div><div><div><div><div>$\bowtie_{D > A}$</div><div>7 rows</div></div><div><div>R</div><div>5 rows</div></div><div><div>S</div><div>5 rows</div></div></div><div><div>$R \bowtie_{D > A} S$</div><div>Execution time: 1 ms</div></div><table><tr><th>R.A</th><th>R.B</th><th>S.B</th><th>S.C</th><th>S.D</th></tr><tr><td>1</td><td>'x'</td><td>'x'</td><td>0</td><td>3</td></tr><tr><td>1</td><td>'x'</td><td>'y'</td><td>3</td><td>3</td></tr><tr><td>1</td><td>'x'</td><td>'y'</td><td>4</td><td>2</td></tr><tr><td>2</td><td>'y'</td><td>'x'</td><td>0</td><td>3</td></tr><tr><td>2</td><td>'y'</td><td>'y'</td><td>3</td><td>3</td></tr><tr><td>2</td><td>'z'</td><td>'x'</td><td>0</td><td>3</td></tr><tr><td>2</td><td>'z'</td><td>'y'</td><td>3</td><td>3</td></tr></table></div></div>	R.A	R.B	S.B	S.C	S.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
R.A	R.B	S.B	S.C	S.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																																					
$R \bowtie S$	<p>Se selecciona la relación R y se junta con S usando su columna en comun, si no hay coincidencia en S se añade un null, resultando en 7 tuplas:</p> <div><div><div><div><div>\bowtie</div><div>7 rows</div></div><div><div>R</div><div>5 rows</div></div><div><div>S</div><div>5 rows</div></div></div><div><div>$R \bowtie S$</div><div>Execution time: 1 ms</div></div><table><tr><th>R.A</th><th>R.B</th><th>S.C</th><th>S.D</th></tr><tr><td>1</td><td>'x'</td><td>0</td><td>3</td></tr><tr><td>2</td><td>'y'</td><td>2</td><td>1</td></tr><tr><td>2</td><td>'y'</td><td>3</td><td>3</td></tr><tr><td>2</td><td>'y'</td><td>4</td><td>2</td></tr><tr><td>2</td><td>'z'</td><td>null</td><td>null</td></tr><tr><td>3</td><td>'x'</td><td>0</td><td>3</td></tr><tr><td>9</td><td>'a'</td><td>null</td><td>null</td></tr></table></div></div>	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								
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																																						

$\rho_{C \leftarrow A} R \bowtie S$	<p>Se renombra la columna A de R a C y se hace un join natural con S, se regresa donde C sea igual a A y como comparten B tambien debe ser igual, resultando en 1 tupla:</p> 
$\pi_B(R) - \pi_B(\sigma_{C \geq 2}(S))$	<p>Se selecciona de S las tuplas donde C es mayor o igual a 2, se seleccionan las diferentes B de la consulta anterior; se toman las diferentes B de R y se restan la primera consulta, resultando en 3 tuplas:</p> 

$\pi_A(R) \cap \rho_{A \leftarrow D} (\pi_D(S))$	<p>Selecciona las diferentes A de R y se intersecan con las diferentes D de S ahora renombradas a A, resultando en 3 tuplas:</p> <div><p>$\pi_A(R) \cap \rho_{A \leftarrow D}(\pi_D(S))$ Execution time: 1 ms</p><table><tr><th>R.A</th></tr><tr><td>1</td></tr><tr><td>2</td></tr><tr><td>3</td></tr></table></div>	R.A	1	2	3											
R.A																
1																
2																
3																
$\pi_D(S) \bowtie_{S.D > R.A} R$	<p>Selecciona las diferentes D de S y se hace un join natural con S, se seleccionan las tuplas donde D es mayor que A de R, resultando en 4 tuplas:</p> <div><p>$\pi_D(S) \bowtie_{S.D > R.A} R$ Execution time: 1 ms</p><table><tr><th>S.D</th><th>R.A</th><th>R.B</th></tr><tr><td>3</td><td>1</td><td>'x'</td></tr><tr><td>3</td><td>2</td><td>'y'</td></tr><tr><td>3</td><td>2</td><td>'z'</td></tr><tr><td>2</td><td>1</td><td>'x'</td></tr></table></div>	S.D	R.A	R.B	3	1	'x'	3	2	'y'	3	2	'z'	2	1	'x'
S.D	R.A	R.B														
3	1	'x'														
3	2	'y'														
3	2	'z'														
2	1	'x'														

$\gamma_{A; \text{count}(B)} \rightarrow t(R \bowtie S)$

Empezamos haciendo el natural join de R y S donde B es igual, si falta alguno ponemos null, agrupamos las filas resultantes por el atributo A de R usando la funcion de agregacion que cuenta el numero de ocurrencias de cada valor B en la relacion R (B era ambiguo pues ambos tienen B) por cada grupo de A, el resultado se guarda en una columna llamada t, al final salen 5 tuplas: (una por cada tipo de A)

Y A; COUNT(R.B)→t
5 rows

(\bowtie)
8 rows

R
5 rows

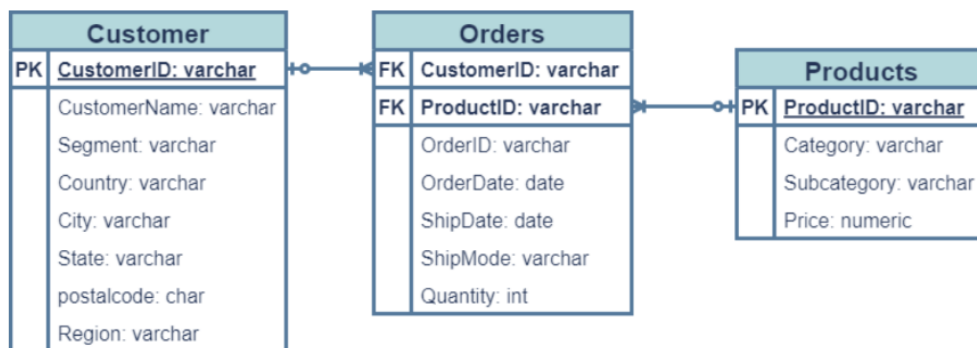
S
5 rows

Y A; COUNT(R.B)→t (R \bowtie S)
Execution time: 1 ms

R.A	t
1	1
2	4
3	1
9	1
null	0

2. Tienda de productos en línea.

Tienes el siguiente esquema de una base de datos para una tienda en línea (ID gist: 31074567738afef8c497f6ca89335782)



Escribe una expresión de álgebra relacional para responder las siguientes consultas. Deberás comprobar cada una ellas en la calculadora Relax y agregar para cada inciso la expresión en álgebra relacional y una captura de pantalla con el resultado obtenido (no es necesario mostrar todas las tuplas):

- a. Obtener toda la información de los clientes que viven en Seattle o en San Francisco, que pertenezcan al segmento corporate que hayan solicitado una orden en el segundo trimestre de 2014. Mostrar la información ordenada por la cantidad solicitada.

La verdad es que la sintaxis de la pregunta deja poco claro lo que se necesita pero yo lo interprete de la siguiente manera:

```
1 C = σ segment='Corporate' AND (city='Seattle' OR city='San Francisco') (customer)
2 D = σ orderdate >= date('2014-04-01') and orderdate <= date('2014-06-30') orders
3 A = (C ⋈ D)
4 O = τ quantity A
5 π customerid, customername, segment, country, city, state, postalcode, region O
6
```

La idea es la siguiente, comenzamos por seleccionar a los clientes que viven en Seattle o en San Francisco, que pertenezcan al segmento corporate, eso es una mini consulta. Luego seleccionamos las ordenes que se hicieron en el segundo trimestre de 2014, eso es otra mini consulta. Luego unimos ambas consultas, finalmente ordenamos por la cantidad solicitada, aqui es donde no estoy seguro de si ahi acaba nuestra consulta o, como dice que solo quiere la información de los clientes tenemos que solo regresarle las columnas relevantes (si no hacemos el siguiente paso se van a repetir clientes porque tienen varios pedidos, igual otra opcion es agrupar por cliente), entonces decidi incluir esta ultima aunque si no es necesario podriamos parar ahi, entonces solo proyectamos sobre las columnas de customer sobre la consulta anterior.

Esto nos regresa la siguiente tabla:

customerid	customername	customersegment	customercountry	customercity	customerstate	customerpostalcode	customerregion
'KL-16555'	'Kelly Lampkin'	'Corporate'	'United States'	'San Francisco'	'California'	'94110'	'West'
'JM-15655'	'Jim Mitchum'	'Corporate'	'United States'	'Seattle'	'Washington'	'98103'	'West'
'ML-17395'	'Marina Lichtenstein'	'Corporate'	'United States'	'San Francisco'	'California'	'94109'	'West'
'MH-17785'	'Maja Herman'	'Corporate'	'United States'	'Seattle'	'Washington'	'98105'	'West'
'GP-14740'	'Guy Phoney'	'Corporate'	'United States'	'Seattle'	'Washington'	'98103'	'West'

Como se ve esto regresa toda la información de los clientes en la forma que se solicita con el detalle de no incluir detalle de la orden.

- b.
- c.
- 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**.

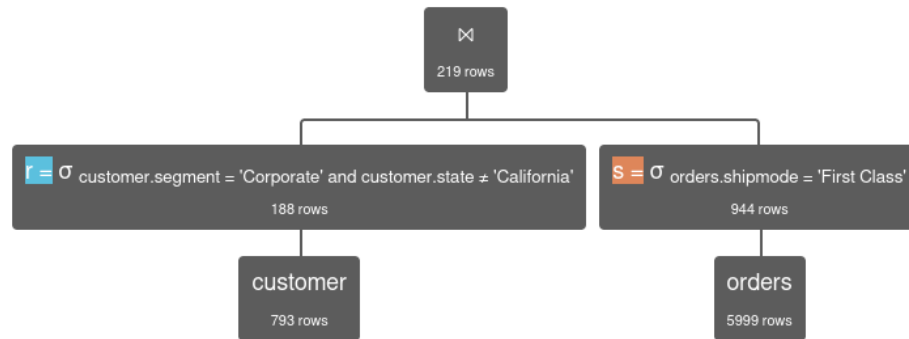
En primera instancia, *seleccionamos* los clientes que pertenecen al segmento **Corporate** y que no viven en **California** de la relación *customer*, almacenando dichas tuplas en una relación temporal *r* . Luego, seleccionamos las órdenes cuyo modo de envío es **First Class** de la relación *orders*. Finalmente, realizamos un natural join entre las dos relaciones resultantes para obtener lo que nos pide el inciso.

```

1 r = σ customer.segment = 'Corporate' ∧ customer.state ≠ 'California' (customer)
2 s = σ orders.shipmode = 'First Class' (orders)
3
4 r ⋈ s

```

Ahora el arbol de la consulta se ve de la siguiente manera:



y parte de la tabla resultante es la siguiente:

customer.customerid	customer.customername	customer.segment	customer.country	customer.city	customer.state
'KB-16585'	'Ken Black'	'Corporate'	'United States'	'Fremont'	'Nebraska'
'KB-16585'	'Ken Black'	'Corporate'	'United States'	'Fremont'	'Nebraska'
'KB-16585'	'Ken Black'	'Corporate'	'United States'	'Fremont'	'Nebraska'
'GH-14485'	'Gene Hale'	'Corporate'	'United States'	'Richardson'	'Texas'
'GH-14485'	'Gene Hale'	'Corporate'	'United States'	'Richardson'	'Texas'
'LC-16930'	'Linda Cazamias'	'Corporate'	'United States'	'Naperville'	'Illinois'
'LC-16930'	'Linda Cazamias'	'Corporate'	'United States'	'Naperville'	'Illinois'
'ES-14080'	'Erin Smith'	'Corporate'	'United States'	'Melbourne'	'Florida'
'ES-14080'	'Erin Smith'	'Corporate'	'United States'	'Melbourne'	'Florida'
'ES-14080'	'Erin Smith'	'Corporate'	'United States'	'Melbourne'	'Florida'

customer.state	customer.postalcode	customer.region	orders.orderid	orders.orderdate	orders.shipdate	orders
Nebraska	'68025'	'Central'	'CA-2015-138674'	2015-11-14	2015-11-17	'Firm'
Nebraska	'68025'	'Central'	'CA-2015-138674'	2015-11-14	2015-11-17	'Firm'
Nebraska	'68025'	'Central'	'CA-2015-138674'	2015-11-14	2015-11-17	'Firm'
Texas	'75080'	'Central'	'CA-2016-117590'	2016-12-08	2016-12-10	'Firm'
Texas	'75080'	'Central'	'CA-2016-117590'	2016-12-08	2016-12-10	'Firm'
Illinois	'60540'	'Central'	'CA-2016-152289'	2016-08-26	2016-08-28	'Firm'
Illinois	'60540'	'Central'	'CA-2016-152289'	2016-08-26	2016-08-28	'Firm'
Florida	'32935'	'South'	'CA-2016-136924'	2016-07-14	2016-07-17	'Firm'
Florida	'32935'	'South'	'CA-2014-108189'	2014-10-02	2014-10-05	'Firm'
Florida	'32935'	'South'	'CA-2014-108189'	2014-10-02	2014-10-05	'Firm'

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

Dado esto, *seleccionamos* todos los clientes de la relación *customer*, almacenando dichas tuplas en una relación temporal *clientes*. Luego, seleccionamos los IDs de los clientes que han realizado órdenes de la relación *orders*, almacenando dichas tuplas en una relación temporal *clientes_con_ordenes*. A continuación, obtenemos los IDs de los clientes que no han realizado ninguna orden mediante la operación de diferencia de conjuntos entre *clientes* y *clientes_con_ordenes*, almacenando el resultado en relación temporal *clientes_sin_ordenes*. Finalmente, utilizamos el operador de proyección para obtener el **estado**, **segmento** y el **total de clientes** que no han solicitado **ninguna orden**.

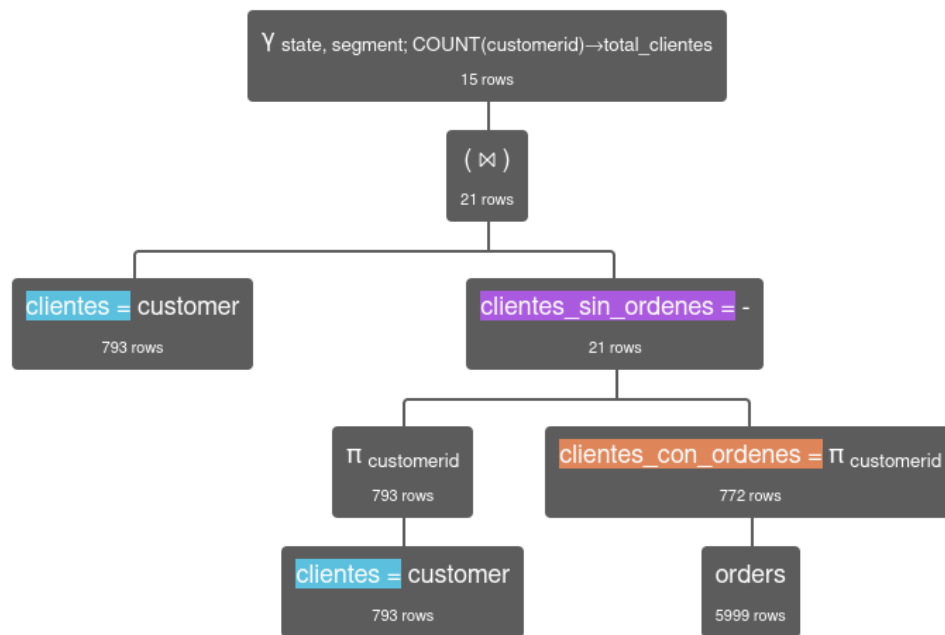
La consulta en álgebra relacional se vería mas o menos de la siguiente manera:

```

1 clientes = customer
2 clientes_con_ordenes =  $\pi$  customerid (orders)
3
4 clientes_sin_ordenes =  $\pi$  customerid (clientes) - clientes_con_ordenes
5
6  $\gamma$  state, segmento; COUNT(customerid)  $\rightarrow$  total_clientes (clientes  $\bowtie$  clientes_sin_ordenes)

```

El árbol de la consulta:



y la tabla resultante:

customer.state	customer.segment	total_clientes
'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

g.

- h.
- i.
- j.

3. Operaciones de mantenimiento de datos: borrado, inserción y actualización

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

En este caso, lo primero que hacemos es realizar un natural join entre *orders*, *customer* y *products*, almacenando dicho join en una relación temporal *r*. Luego, seleccionamos las tuplas que sean de la ciudad **Utah** y que tengan artículos de la subcategoría **Tables**, guardando estas tuplas en otra relación temporal *s*. A continuación, hacemos una proyección de los IDs de la relación *s* creada anteriormente (llamamos a esto *ordenes_a_borrar*). Finalmente, hacemos una diferencia entre *orders* y el join natural derivado de *orders* y *ordenes_a_borrar*.

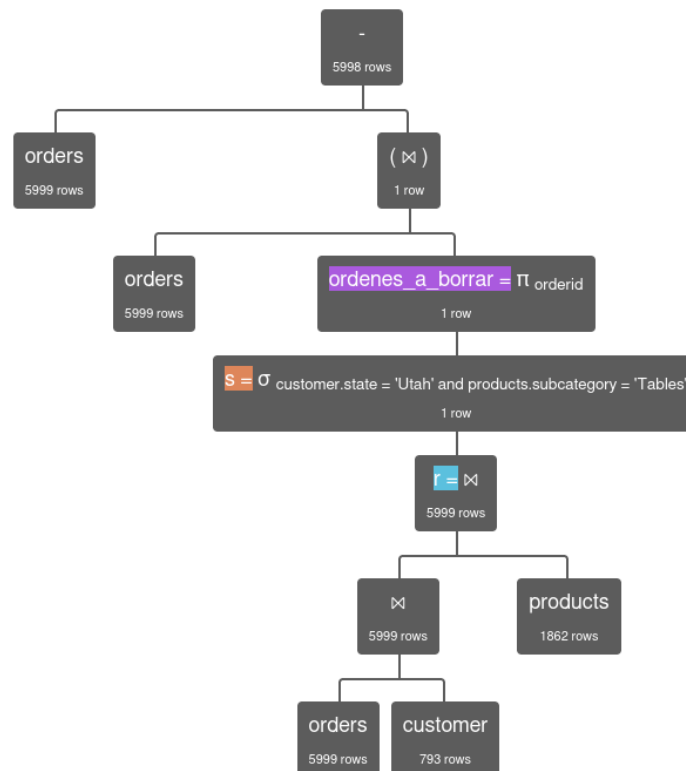
La consulta en algebra relacional se veria de la siguiente manera:

```

1 r = orders ⋈ customer ⋈ products
2 s = σ customer.state = 'Utah' ∧ products.subcategory = 'Tables' (r)
3 ordenes_a_borrar = π orderid (s)
4 orders - (orders ⋈ ordenes_a_borrar)

```

Ahora el árbol de la consulta:



y forma de la tabla resultante:

orders.orderid	orders.orderdate	orders.shipdate	orders.shipmode	orders.customerid	orders.productid
'CA-2016-152156'	2016-11-08	2016-11-11	'Second Class'	'CG-12520'	'FUR-BO-10001798'
'CA-2016-152156'	2016-11-08	2016-11-11	'Second Class'	'CG-12520'	'FUR-CH-10000454'
'CA-2016-138688'	2016-06-12	2016-06-16	'Second Class'	'DV-13045'	'OFF-LA-10000240'
'US-2015-108966'	2015-10-11	2015-10-18	'Standard Class'	'SO-20335'	'FUR-TA-10000577'
'US-2015-108966'	2015-10-11	2015-10-18	'Standard Class'	'SO-20335'	'OFF-ST-10000760'
'CA-2014-115812'	2014-06-09	2014-06-14	'Standard Class'	'BH-11710'	'FUR-FU-10001487'
'CA-2014-115812'	2014-06-09	2014-06-14	'Standard Class'	'BH-11710'	'OFF-AR-10002833'
'CA-2014-115812'	2014-06-09	2014-06-14	'Standard Class'	'BH-11710'	'TEC-PH-10002275'
'CA-2014-115812'	2014-06-09	2014-06-14	'Standard Class'	'BH-11710'	'OFF-BI-10003910'
'CA-2014-115812'	2014-06-09	2014-06-14	'Standard Class'	'BH-11710'	'OFF-AP-10002892'

d.

- 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.

Para esta consulta use el video '07 Álgebra Relacional | Actualización | Operaciones de mantenimiento de datos', al final use esta consulta:

```

1 -- Vamos a ubicar a los que van a ser actualizados
2 F =  $\sigma$  category='Furniture' and price $\geq$ 600 and price $\leq$ 900 products
3 M =  $\sigma$  category='Technology' and subcategory='Machines' products
4
5 -- Actualizamos cadda uno
6 A =  $\pi$  productid, category,subcategory,newprice $\leftarrow$ price*0.92 (F)
7 B =  $\pi$  productid, category,subcategory,newprice $\leftarrow$ price*1.05 (M)
8
9 -- Unimos y renombramos
10 products1 = A $\cup$ B
11  $\pi$  productid, category, subcategory, price $\leftarrow$ newprice products1

```

De manera que me quedo la siguiente Actualización:

products.productid	products.category	products.subcategory	price
'FUR-CH-10000454'	'Furniture'	'Chairs'	673.3848
'FUR-FU-10003773'	'Furniture'	'Furnishings'	568.2840000000001
'FUR-TA-10004289'	'Furniture'	'Tables'	827.2088
'FUR-FU-10002960'	'Furniture'	'Furnishings'	576.2420000000001
'FUR-CH-10001891'	'Furniture'	'Chairs'	667.7728000000001
'FUR-BO-10001601'	'Furniture'	'Bookcases'	797.088
'FUR-FU-10000221'	'Furniture'	'Furnishings'	595.0376
'FUR-FU-10001095'	'Furniture'	'Furnishings'	595.0008
'FUR-FU-10004270'	'Furniture'	'Furnishings'	664.562
'FUR-FU-10002088'	'Furniture'	'Furnishings'	592.5536000000001

< 1 2 3 >

Ojo que esta solo contiene la tabla de los productos con los precios actualizados como se ve en el video, no contiene toda la lista de productos pero seria tan facil como eliminar las tuplas que se estan seleccionando, e insertar estas nuevas, no lo hice pues no lo hizo el profe :v.