

PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA INFORMÁTICA

TALLER DE BASES DE DATOS

JEAN RODRÍGUEZ

INFORME DE TALLER IV

OCTUBRE, 2019

Índice

1. Introducción	1
2. Resolución de ejercicios	2
3. Conclusiones	6

Lista de Figuras

1.	Primer ejercicio	2
2.	relación compra-producto	3
3.	Segundo ejercicio	4
4.	Tercer ejercicio	5

1. Introducción

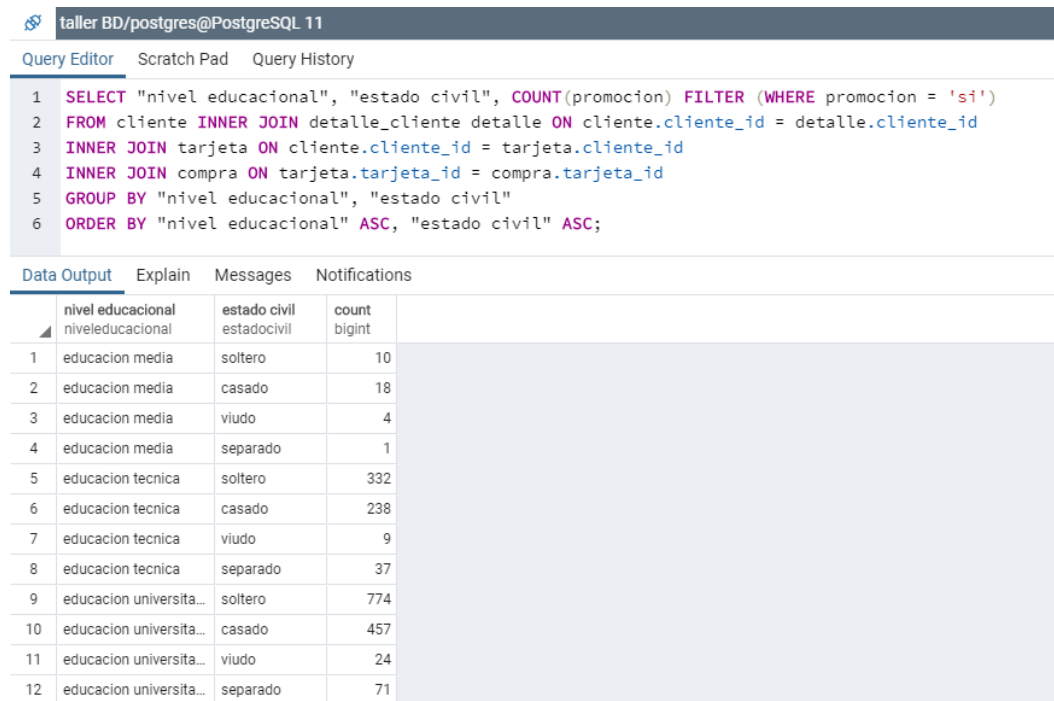
Los modelos de negocio actuales manejan cantidades enormes de datos de los clientes en cada empresa, sea grande o pequeña, lo que evidencia una clara necesidad de un software capaz de procesar las solicitudes requeridas. Cuando una entidad desea minimizar la complejidad de sus operaciones y mantener tiempos de respuesta aceptables, se debe dejar en claro un diseño simple y eficiente que sea mantenible y facilite la realización de cambios a futuro.

Con la intención de demostrar un correcto funcionamiento en la base de datos propuesta, se deja a disposición una serie de consultas para el motor PostgreSQL que entregan respuestas de manera rápida y efectiva, utilizando como base el script inicial para cargar la base de datos, y su correspondiente archivo de valores separado por comas.

2. Resolución de ejercicios

A continuación se presenta el conjunto de preguntas y respuestas correspondientes en el lenguaje PostgreSQL con los accesos necesarios a las tablas y sus llaves.

- 1.- Realice una consulta que muestre por cada tipo de Nivel educacional y por cada estado civil, la cantidad de productos en promoción comprados (por separado, indicando aquellas que han comprado productos en promoción).



The screenshot shows a PostgreSQL query editor interface. The title bar indicates the connection is 'taller BD/postgres@PostgreSQL 11'. The 'Query Editor' tab is active, displaying a SQL query. Below the query, the 'Data Output' tab shows the results of the query in a table format. The query counts the number of products in promotion for each combination of educational level and civil status.

```
1 SELECT "nivel educacional", "estado civil", COUNT(promocion) FILTER (WHERE promocion = 'si')
2 FROM cliente INNER JOIN detalle_cliente detalle ON cliente.cliente_id = detalle.cliente_id
3 INNER JOIN tarjeta ON cliente.cliente_id = tarjeta.cliente_id
4 INNER JOIN compra ON tarjeta.tarjeta_id = compra.tarjeta_id
5 GROUP BY "nivel educacional", "estado civil"
6 ORDER BY "nivel educacional" ASC, "estado civil" ASC;
```

	nivel educacional niveleducacional	estado civil estadocivil	count bigint
1	educacion media	soltero	10
2	educacion media	casado	18
3	educacion media	viudo	4
4	educacion media	separado	1
5	educacion tecnica	soltero	332
6	educacion tecnica	casado	238
7	educacion tecnica	viudo	9
8	educacion tecnica	separado	37
9	educacion universita...	soltero	774
10	educacion universita...	casado	457
11	educacion universita...	viudo	24
12	educacion universita...	separado	71

Figura 1: Primer ejercicio

2.- Realice una consulta donde aparezca el nombre de todos los clientes que han comprado productos en promoción A.

Dado que en modelo relacional establece a los productos como una entidad separada de las compras, como lo muestra la figura 2; además se requiere que las consultas no incluyan subqueries, por lo que es necesario realizar un conteo con filtro en una clausura HAVING, puesto que sirve para evaluar la función de agregación COUNT.

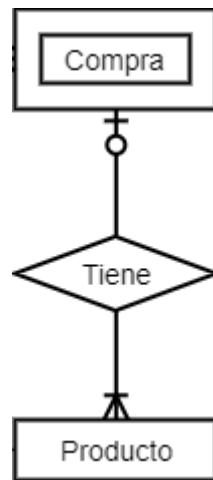


Figura 2: relación compra-producto

taller BD/postgres@PostgreSQL 11

Query Editor Scratch Pad Query History

```

1 SELECT concat_ws(' ', nombres, "primer apellido", "segundo apellido") AS "cliente"
2 FROM cliente INNER JOIN tarjeta ON cliente.cliente_id = tarjeta.cliente_id
3 INNER JOIN compra ON tarjeta.tarjeta_id = compra.tarjeta_id
4 INNER JOIN producto ON compra.compra_id = producto.compra_id
5 GROUP BY "cliente", promocion
6 HAVING (COUNT("tipo de producto") FILTER (WHERE "tipo de producto" = 'A')) > 0
7 AND promocion = 'si';

```

Data Output Explain Messages Notifications

	cliente text
1	MARIA MIRTHA SABINA PIRIZ CAZULO
2	KARIM DINORAH BLANCO RAMOS
3	MABEL MARÍA GUIDO PESCE
4	FERNANDO RAUL CARDINAL PIEGAS
5	DANIEL ALDECOSEA PEDROZZA
6	CARLOS FERNANDO ENCISO CHRISTIANSEN
7	EDUARDO ARBULO
8	JOSE ALFREDO DE MATTOS DE MELLO
9	DARÍO CONSTANTINO MENONI GONZALEZ
10	GUSTAVO ADRIAN ESPINOSA MARMOL
11	MARÍA TERESA LARROSA ---

Figura 3: Segundo ejercicio

3.- Realice una consulta que muestre los clientes que han realizado más de 5 compras en el año.

taller BD/postgres@PostgreSQL 11			
Query Editor Scratch Pad Query History			
<pre> 1 SELECT concat_ws(' ', nombres, "primer apellido", "segundo apellido") AS "cliente", 2 concat_ws('-', cliente.rut, verificador) AS "rut", tarjeta_id 3 FROM cliente INNER JOIN tarjeta ON cliente.cliente_id = tarjeta.cliente_id 4 WHERE "compras promedio" > 5 5 order by "cliente"; </pre>			
Data Output Explain Messages Notifications			
	cliente text	rut text	tarjeta_id integer
1	HAROLDO ANTONIO CANONIERO MARR	2577157-7	486
2	HAROLDO ANTONIO CANONIERO MARR	2577157-7	3154
3	JOSE PEDRO PEZZATTI PIAGGIO	3078930-9	1951
4	ABEL CESAR VANNI ROSAS	3024909-0	2540
5	ABILIO HUGO BRIZ LUCAS	1732335-0	3062
6	ACIDO ARCANGEL ROUS LEITES	3549064-0	2228
7	ADA CRISTIANA PASTRO BONFIGLIO	1109102-2	1857
8	ADAN CUELLO ROSA	1539346-2	3372
9	ADAN CUELLO ROSA	1539346-2	704
10	ADEL MIRZA PERPIGNANI	1391024-0	1654
11	ADELA ISABEL HOUNIE GONZALEZ	1211157-6	1286
12	ADELA ISABEL HOUNIE GONZALEZ	1211157-6	3956
13	ADELA ISABEL HOUNIE GONZALEZ	1211157-6	1288

Figura 4: Tercer ejercicio

3. Conclusiones

Del presente informe se obtuvo en detalle el proceso de organización necesario para conformar la base de datos propuesta para el cliente ficticio, de tal manera que facilita la construcción del software que debe utilizar las variables necesarias para que el negocio funcione de manera exitosa, dejándose expuesta una interpretación clara del modelo definido para entregar tiempos de respuesta breves en futuras consultas.

Además, se obtuvo resultados satisfactorios para consultas que relacionan múltiples tablas para entregar respuestas concisas y que el cliente necesitaría hacer en la base de datos, lo que asegura integridad al momento de implementar este en sistemas que requieran un procesamiento rápido y seguro de la información.