

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

TALLER DE BASES DE DATOS

JEAN RODRÍGUEZ

INFORME DE TALLER VI

DICIEMBRE, 2019

Índice

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

Lista de Figuras

1.	Primer ejercicio	2
2.	Segundo ejercicio	3
3.	Tercer ejercicio	4
4.	Cuarto ejercicio	5

1. Introducción

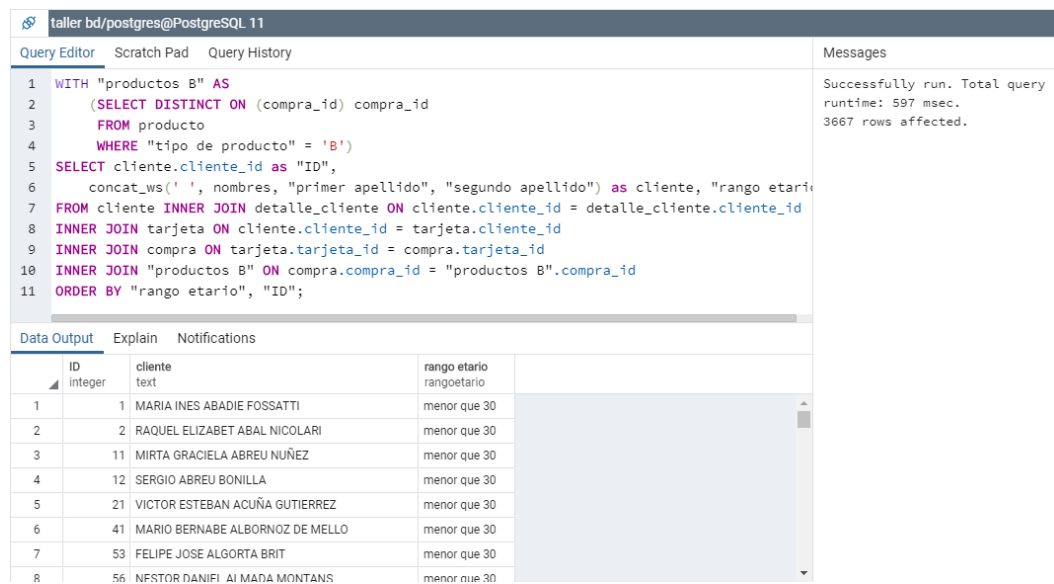
Los modelos de negocio actuales manejan cantidades enormes de datos de los clientes en cada empresa, sea grande o pequeña, lo que evidencia una clara necesidad de un software capaz de procesar las solicitudes requeridas. Cuando una entidad desea minimizar la complejidad de sus operaciones y mantener 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 en la cual separe por rango etario a los clientes que compran productos B.



The screenshot shows a PostgreSQL query editor interface. The top bar indicates the connection is to 'taller bd/postgres@PostgreSQL 11'. The interface includes tabs for 'Query Editor', 'Scratch Pad', and 'Query History', along with a 'Messages' panel on the right.

The SQL query in the editor is as follows:

```
1 WITH "productos B" AS
2 (SELECT DISTINCT ON (compra_id) compra_id
3  FROM producto
4  WHERE "tipo de producto" = 'B')
5 SELECT cliente.cliente_id as "ID",
6        concat_ws(' ', nombres, "primer apellido", "segundo apellido") as cliente, "rango etario"
7 FROM cliente INNER JOIN detalle_cliente ON cliente.cliente_id = detalle_cliente.cliente_id
8 INNER JOIN tarjeta ON cliente.cliente_id = tarjeta.cliente_id
9 INNER JOIN compra ON tarjeta.tarjeta_id = compra.tarjeta_id
10 INNER JOIN "productos B" ON compra.compra_id = "productos B".compra_id
11 ORDER BY "rango etario", "ID";
```

The 'Messages' panel on the right displays the execution status: 'Successfully run. Total query runtime: 597 msec. 3667 rows affected.'

Below the query editor, the 'Data Output' tab is active, showing a table with the following data:

ID	cliente	rango etario
1	MARIA INES ABADIE FOSSATTI	menor que 30
2	RAQUEL ELIZABET ABAL NICOLARI	menor que 30
3	MIRTA GRACIELA ABREU NUÑEZ	menor que 30
4	SERGIO ABREU BONILLA	menor que 30
5	VICTOR ESTEBAN ACUÑA GUTIERREZ	menor que 30
6	MARIO BERNABE ALBORNOZ DE MELLO	menor que 30
7	FELIPE JOSE ALGORTA BRIT	menor que 30
8	NFSTOR DANIFI AI MADA MONTANS	menor que 30

Figura 1: Primer ejercicio

2.- Realice una consulta en la cual dependiendo de la actividad de los clientes compran productos A.

taller bd/postgres@PostgreSQL 11

Query Editor Scratch Pad Query History Messages

```

1 WITH "productos A" AS
2   (SELECT DISTINCT ON (compra_id) compra_id
3    FROM producto
4    WHERE "tipo de producto" = 'A')
5 SELECT cliente.cliente_id as "ID",
6        concat_ws(' ', nombres, "primer apellido", "segundo apellido") as cliente, actividad
7 FROM cliente INNER JOIN detalle_cliente ON cliente.cliente_id = detalle_cliente.cliente_id
8 INNER JOIN tarjeta ON cliente.cliente_id = tarjeta.cliente_id
9 INNER JOIN compra ON tarjeta.tarjeta_id = compra.tarjeta_id
10 INNER JOIN "productos A" ON compra.compra_id = "productos A".compra_id
11 ORDER BY actividad, "ID";
12

```

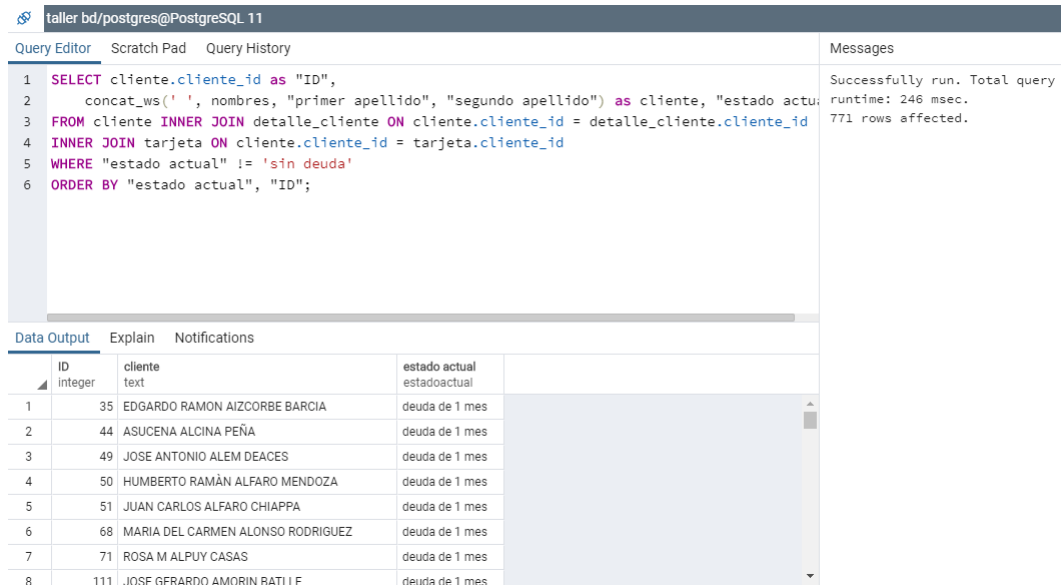
Successfully run. Total query runtime: 195 msec. 3621 rows affected.

Data Output Explain Notifications

	ID integer	cliente text	actividad actividad
1	3	MARÍA ROSARIO ABALDE MARTINEZ	dependiente
2	4	ALBERTO OSCAR ABALOS ROCHON	dependiente
3	5	ARIEL ABARNO SILVA	dependiente
4	6	WINSTON FRANKLIN ABASCAL BELOQUI	dependiente
5	7	PABLO DANIEL ABDALA SCHWARZ	dependiente
6	9	JORGE MARIA ABIN DE MARIA	dependiente
7	10	ALCIDES ABREU HERNANDEZ	dependiente
8	12	SFRGIN ARRFJ RIONII A	dependiente

Figura 2: Segundo ejercicio

3.- Realice una consulta en la cual me diga todos los clientes con deuda.



The screenshot shows a PostgreSQL Query Editor interface. The top bar indicates the user is 'taller bd/postgres@PostgreSQL 11'. The interface is divided into three main sections: 'Query Editor', 'Scratch Pad', and 'Query History'. The 'Query Editor' contains the following SQL query:

```

1 SELECT cliente.cliente_id as "ID",
2       concat_ws(' ', nombres, "primer apellido", "segundo apellido") as cliente, "estado actual" as estado_actual
3 FROM cliente INNER JOIN detalle_cliente ON cliente.cliente_id = detalle_cliente.cliente_id
4 INNER JOIN tarjeta ON cliente.cliente_id = tarjeta.cliente_id
5 WHERE "estado actual" != 'sin deuda'
6 ORDER BY "estado actual", "ID";

```

To the right of the query editor, the 'Messages' pane shows the execution status: 'Successfully run. Total query runtime: 246 msec. 771 rows affected.'

Below the query editor, the 'Data Output' pane displays the results of the query in a table format:

ID	cliente	estado actual
integer	text	estadoactual
1	35 EDGARDO RAMON AIZCORBE BARCIA	deuda de 1 mes
2	44 ASUCENA ALCINA PEÑA	deuda de 1 mes
3	49 JOSE ANTONIO ALEM DEACES	deuda de 1 mes
4	50 HUMBERTO RAMÁN ALFARO MENDOZA	deuda de 1 mes
5	51 JUAN CARLOS ALFARO CHIAPPA	deuda de 1 mes
6	68 MARIA DEL CARMEN ALONSO RODRIGUEZ	deuda de 1 mes
7	71 ROSA M ALPUY CASAS	deuda de 1 mes
8	111 JOSE GFRARDO AMORIN RATTI F	deuda de 1 mes

Figura 3: Tercer ejercicio

Cabe recordar que los valores permitidos en el estado actual de la deuda registrada en la tarjeta corresponden a:

- sin deuda
- deuda de 1 mes
- deuda de 2 meses

al ser un tipo de dato registrado como enumeración, se permite hacer una comparación directa de los valores en vez de utilizar la cláusula LIKE como se haría para comparar cadenas de caracteres.

4.- Necesito saber los clientes que tiene un porcentaje de uso de cupo sobre 40%.

taller bd/postgres@PostgreSQL 11

Query Editor Scratch Pad Query History Messages

```

1 SELECT cliente.cliente_id as "ID",
2       concat_ws(' ', nombres, "primer apellido", "segundo apellido") as cliente, "pct uso cupo"
3 FROM cliente INNER JOIN detalle_cliente ON cliente.cliente_id = detalle_cliente.cliente_id
4 INNER JOIN tarjeta ON cliente.cliente_id = tarjeta.cliente_id
5 WHERE "pct uso cupo" > 40
6 ORDER BY "pct uso cupo", "ID";

```

Successfully run. Total query runtime: 215 msec. 1104 rows affected.

Data Output Explain Notifications

	ID integer	cliente text	pct uso cupo real
1	827	LAURA ISABEL DOGLIO CASULLO	40.2
2	1827	PATRICIA MACARENA PACHECO CABRERA	40.2
3	2827	MARIA ELIDA ARGUELLO WETHERALL	40.2
4	3827	NANNER GOITIÑO ARIGON	40.2
5	369	FREDY JOSE BORBA SILVA	40.7
6	787	GABRIEL EDUARDO DEL RIO NOBLE	40.7
7	1369	JULIO LAMATH PEREYRA	40.7
8	1787	HFCTOR RAIJI OI IVFRA RANGFI	40.7

Figura 4: Cuarto 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 a la hora de aplicar en su conjunto el conocimiento aplicado en las entregas previas, como subqueries, joins, agrupacion de datos, entre otros.