



Pontificia Universidad Católica de Chile Escuela de Ingeniería Departamento de Ciencia de la Computación Procesamiento de Datos Masivos - IIC2440 – 2024-1

IIC2440 — Procesamiento de Datos Masivos — 1' 2024

Tarea - 1

1 Parte 2

- 2.1. Esta idea que tiene para realizar su analitica en los datos que tiene en su base de datos no es la más óptima ya que al ser una Base de datos OLTP la cual tiene funcionan la aplicación hacia los usuarios por lo tanto, al momento de realizar una consulta la cual pueda ser muy pesada y que consuma muchos recursos del servidor donde esta alojada la Base de Datos puede ocasionar que la aplicación funcione mal o simplemente esta se caiga y quede inoperativa.
- 2.2. En esta parte para las consultas de validación son bastante simples pero nos permiten ver que cada id de las ordenes tiene asignado un usuario. Primero, debemos tener en cuenta que nosotros le realizamos una reindexación a las compras id para asi estas puedan ser unicas y no tengamos problemas luego ya que, como por mes estos id se reseteaban nos generaba problemas.
- 2.3. las consultas al comienzo de estas todas tienen tablas realizadas con with para asi ir obteniendo lo necesario para poder realizar consulta, esto para no tener tantos join y nos quede una consulta muy compleja y dificil de comprender.
- 2.4. Aqui colocaremos las respuestas a nuestro analisis que obtenemos de las consultas en cuestion.
 - La región que va a registrar un crecimiento el año que sigue sera la dos, puesto que como es observable en la cantidad de gasto mensual en esta región cada vez se tienen más ingresos, cosa que en la región 1 no se ve pues cada mes se tiene una menor cantidad de compras por menores montos.
 - En cuanto a la demanda, la región dos supera por mucho en cantidades totales a la región 1, como es visible en el grafico "Cantidades Promedio por Región según Fruta", pues por cada fruta en promedio anual la región 2 supera a la 1, y en compras mensuales la región 1 llega a un poco más de 4000 unidades, mientras que la región 2 se alcanzan casi 14000 unidades.
 - Es notorio como en terminos generales por año para las regiones en conjunto, se notan las temporadas de las frutas, por ejemplo los melones se compran entre enero y marzo, luego se disminuye su venta a niveles bastante bajos hasta que a finales de año vuelven a repuntar. En el caso de las naranjas tambien se nota como su "temporada" es durante los meses de Mayo hasta el fin de año, otro caso con los kiwis que se venden mas en Abril y en Octubre se comienzan a vender menos.
 - Una forma de realizar una predición, es buscar si existe una relacion entre las ventas de kiwi con alguna de las demas frutas, es decir el kiwi tiene alguna correlación con que la gente compre algo más en conjunto con el kiwi.
 - De esta forma podriamos estimar en base la predicción, si existe alguna fruta que se compra con los kiwis, podriamos ver tendencias de compra para productos en conjunto, con esto se puede dar una idea general de la tendencia de compra para las frutas.

```
SELECT

*
FROM

tareal.compra_user'
```

2.1 consulta 2.2.1

```
1 SELECT
2   id_compra,
3   COUNT(usuario_id) AS cantidad_user
4 FROM
5   'tarea1.compra_user'
6 GROUP BY
7   id_compra
8 HAVING
9   cantidad_user > 1;
```

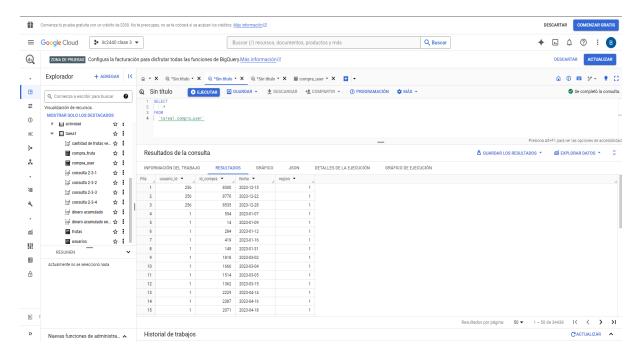


Figure 1: Consulta 2.2.1 junto a su ejecucion y resultado



Figure 2: resultado 2.2.1

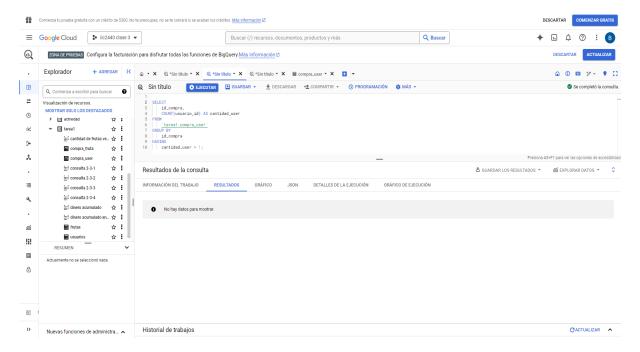


Figure 3: Consulta 2.2.2 junto a su ejecucion y resultado

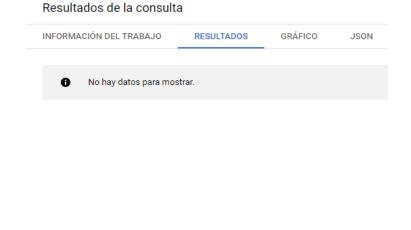


Figure 4: resultado 2.2.2

```
WITH Region as (
     SELECT
     FROM
       'tarea1.compra_user'
     WHERE
       region = 1
10
   compras as (
11
       SELECT *
12
       FROM
13
            'tarea1.compra_fruta' as compra_fruta, Region
14
       WHERE
15
           Region.id_compra = compra_fruta.id_compra
16
   ),
17
18
   frutas_fecha as (
19
       SELECT
20
           fruta,
21
           fecha,
22
           cantidad
23
       FROM
24
           compras,
25
            'tarea1.frutas' as frutas
26
       WHERE
27
           frutas.id_fruta = compras.id_fruta
28
   ),
29
30
   cantidad_frutas as (
31
       SELECT
32
           fruta,
33
           EXTRACT(YEAR FROM fecha) AS ano,
34
           EXTRACT(MONTH FROM fecha) AS mes,
           SUM(cantidad) AS total,
36
       FROM
37
           frutas_fecha
38
       GROUP BY
40
           fruta,
           ano,
41
           mes
42
       ORDER BY
           ano,
44
           mes
45
   )
46
47
   SELECT
48
       fruta,
49
       ano,
51
       mes,
       total
52
   FROM
53
       cantidad_frutas
   ORDER BY
55
       mes;
56
```

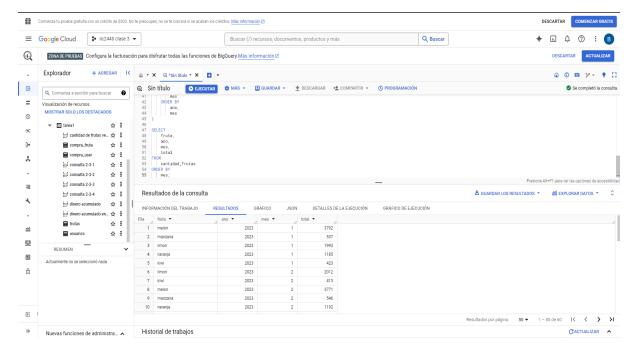


Figure 5: Consulta 2.3.1 junto a su ejecucion y resultado



Figure 6: resultado 2.3.1

```
WITH Region as (
     SELECT
     FROM
        'tarea1.compra_user'
     WHERE
       region = 2 AND usuario_id = 75
   compras as (
       SELECT
11
12
       FROM
13
           'tarea1.compra_fruta' as compra_fruta, Region
14
       WHERE
15
           Region.id_compra = compra_fruta.id_compra
16
   ),
17
18
   frutas_fecha as (
19
       SELECT
20
           fruta,
21
           fecha,
22
           cantidad,
23
           cantidad * precio as gasto
24
       FROM
25
           compras,
26
           'tarea1.frutas' as frutas
27
       WHERE
28
           frutas.id_fruta = compras.id_fruta
29
   )
30
31
   SELECT
32
       EXTRACT(MONTH FROM fecha) AS mes,
33
       sum(gasto) as gasto_total
34
   FROM
35
       frutas_fecha
36
   GROUP BY
37
       mes
38
   ORDER BY
39
       mes ASC
```

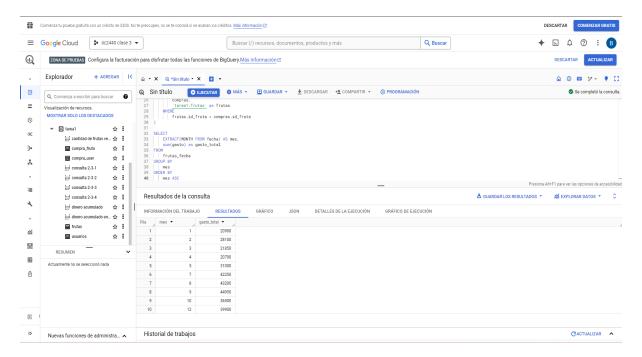


Figure 7: Consulta 2.3.2 junto a su ejecucion y resultado

Resultados de la consulta					
INFORMACIÓN DEL TRABAJO RESULTADOS					
Fila	mes ▼	gasto_total ▼			
1	1	20900			
2	2	28100			
3	3	21850			
4	4	20700			
5	5	31000			
6	7	42250			
7	8	43200			
8	9	44950			
9	10	36500			
10	12	39900			

Figure 8: resultado 2.3.2

```
WITH compras as (
       SELECT
       FROM
           'tareal.compra_fruta' as compra_fruta,
           'tarea1.compra_user' as Region
       WHERE
           Region.id_compra = compra_fruta.id_compra
   ),
10
   frutas_fecha as (
11
       SELECT
           fruta,
13
           fecha,
14
           cantidad,
           region
16
       FROM
17
           compras,
18
           'tarea1.frutas' as frutas
19
       WHERE
           frutas.id_fruta = compras.id_fruta
21
   ),
22
23
   cantidad_frutas as (
24
       SELECT
25
           fruta,
26
           EXTRACT(YEAR FROM fecha) AS ano,
27
           EXTRACT (MONTH FROM fecha) AS mes,
28
           SUM(cantidad) AS total,
29
           region
30
       FROM
31
           frutas_fecha
32
       GROUP BY
33
           region,
34
           fruta,
35
           ano,
36
           mes
37
       ORDER BY
38
39
           ano,
           mes
40
   )
41
42
   SELECT
43
       region,
44
       fruta,
45
       ano,
46
       mes,
47
48
       ROW_NUMBER() OVER (PARTITION BY region, ano, mes ORDER BY total DESC) AS ranking
49
   FROM
51
       cantidad_frutas
   QUALIFY
52
       ranking=1
53
   ORDER BY
55
       mes,
       region
56
```

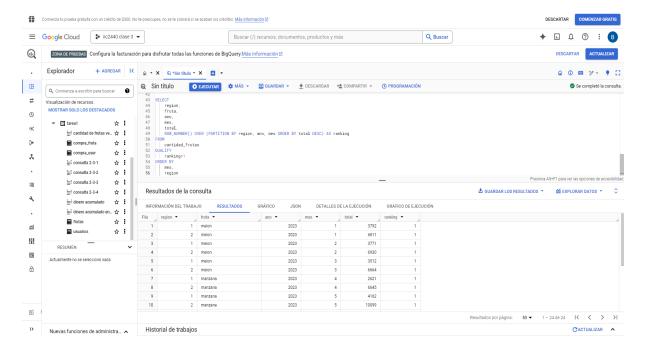


Figure 9: Consulta 2.3.3 junto a su ejecucion y resultado

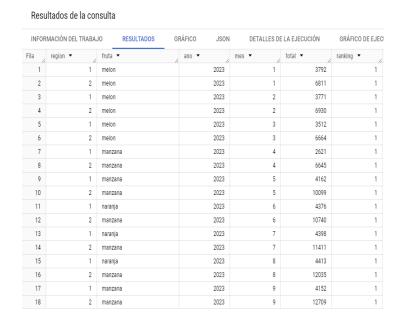


Figure 10: resultado 2.3.3

```
WITH compras as (
       SELECT *
       FROM
3
        'tarea1.compra_fruta' as compra_fruta,
       'tarea1.compra_user' as Region
       WHERE
       Region.id_compra = compra_fruta.id_compra
   ),
   frutas_fecha as (
       SELECT
10
           fruta,
11
           fecha,
           cantidad,
13
           region,
14
           precio
15
           FROM
16
       compras,
17
           'tarea1.frutas' as frutas
18
19
           frutas.id_fruta = compras.id_fruta
20
   ),
21
   cantidad_frutas as (
22
       SELECT
23
           fruta,
24
           EXTRACT(WEEK FROM fecha) AS week,
25
           SUM(cantidad * precio) AS total,
26
           region
27
       FROM
28
           frutas_fecha
29
       GROUP BY
30
           region,
31
           fruta,
32
           week
33
       ORDER BY
34
           week
   ),
36
   semanas as (
37
       SELECT
38
39
           sum(total) as entrante
40
       FROM
41
           cantidad_frutas
42
       GROUP BY
           week
44
       ORDER BY
45
           week)
46
47
   SELECT
48
       week,
49
50
       SUM(entrante) OVER (ORDER BY week ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW
51
            ) AS ganancias_acumuladas
   FROM
52
       semanas
   ORDER BY
54
       week;
55
```

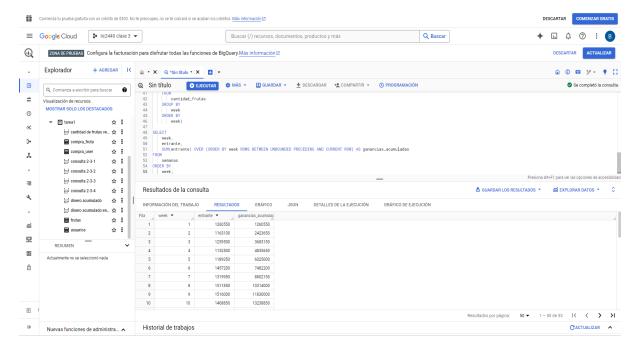


Figure 11: Consulta 2.3.4 junto a su ejecucion y resultado

Resultados de la consulta					
INFO	RMACIÓN DEL TRABA	JO RESULTADO	os GRÁFICO		
Fila	week ▼	entrante ▼	ganancias_acumulac		
1	1	1260550	1260550		
2	2	1163100	2423650		
3	3	1259500	3683150		
4	4	1152500	4835650		
5	5	1189350	6025000		
6	6	1457200	7482200		
7	7	1319950	8802150		
8	8	1511850	10314000		
9	9	1516000	11830000		
10	10	1408850	13238850		
11	11	1370300	14609150		
12	12	1425350	16034500		
13	13	1457200	17491700		
14	14	1552850	19044550		
15	15	1533650	20578200		
16	16	1455800	22034000		
17	17	1646150	23680150		
10	10	1000550	05400700		

Figure 12: resultado 2.3.4

```
WITH compras as (
       SELECT
       FROM
            'tarea1.compra_fruta' as compra_fruta,
           'tarea1.compra_user' as Region
       WHERE
           Region.id_compra = compra_fruta.id_compra
   ),
9
10
   frutas_fecha as (
11
       SELECT
12
           fruta,
           fecha,
14
           cantidad,
15
           region
16
       FROM
           compras,
18
           'tarea1.frutas' as frutas
19
       WHERE
20
           frutas.id_fruta = compras.id_fruta
21
   ),
22
23
   cantidad_frutas as (
24
       SELECT
25
           fruta,
26
           EXTRACT (YEAR FROM fecha) AS ano,
27
           EXTRACT(MONTH FROM fecha) AS mes,
28
           SUM(cantidad) AS total,
           region
30
       FROM
31
           frutas_fecha
32
       GROUP BY
33
           region,
34
           fruta,
35
           ano,
37
           mes
       ORDER BY
38
           ano,
39
           mes
   )
41
42
   SELECT
43
       region,
44
       fruta,
45
       mes,
46
       total,
47
   FROM
       cantidad_frutas
49
   ORDER BY
50
       mes,
51
       fruta,
52
       region
53
```

```
WITH compras as (
       SELECT
       FROM
           'tareal.compra_fruta' as compra_fruta,
           'tarea1.compra_user' as Region
       WHERE
           Region.id_compra = compra_fruta.id_compra
   ),
   frutas_fecha as (
10
       SELECT
11
           fruta,
12
           fecha,
13
           cantidad,
14
           region,
15
           precio
16
       FROM
17
           compras, 'tarea1.frutas' as frutas
18
       WHERE
19
           frutas.id_fruta = compras.id_fruta
20
   ),
21
   cantidad_frutas as (
22
       SELECT
23
           fruta,
24
           EXTRACT (MONTH FROM fecha) AS mes,
25
           SUM(cantidad*precio) AS total,
26
           region
27
       FROM
28
           frutas_fecha
29
       GROUP BY
30
           region,
31
           fruta,
32
           mes
33
       ORDER BY
34
           mes
   ),
36
   semanas as (
37
       SELECT
38
39
           sum(total) as entrante
40
       FROM
41
           cantidad_frutas
42
       GROUP BY
           mes
44
       ORDER BY
45
           mes
46
   )
47
   SELECT
48
       mes,
49
50
       SUM(entrante) OVER (ORDER BY mes ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW)
51
             AS ganancias_acumuladas
   FROM
52
       semanas
   ORDER BY
54
       mes;
55
```

```
WITH compras as (
       SELECT
       FROM
            'tarea1.compra_fruta' as compra_fruta,
           'tarea1.compra_user' as Region
       WHERE
           Region.id_compra = compra_fruta.id_compra
   ),
9
10
   frutas_fecha as (
11
       SELECT
12
           fruta,
           fecha,
14
           cantidad,
15
           region,
16
           precio,
           usuario_id
18
       FROM
19
           compras,
20
           'tarea1.frutas' as frutas
21
       WHERE
22
           frutas.id_fruta = compras.id_fruta
23
   ),
24
25
   cantidad_frutas as (
26
       SELECT
27
           usuario_id,
28
           fruta,
           EXTRACT(MONTH FROM fecha) AS mes,
30
           SUM(cantidad*precio) AS total,
31
           region
32
       FROM
33
           frutas_fecha
34
       GROUP BY
35
           region,
36
           usuario_id,
37
           fruta,
38
           mes
39
       ORDER BY
40
           mes
41
   ),
42
43
   semanas as (
44
       SELECT
45
           usuario_id,
46
           region,
47
           mes,
           sum(total) as entrante
49
       FROM
50
           cantidad_frutas
51
       GROUP BY
52
           region,
53
           usuario_id,
54
           mes
55
       ORDER BY
```

```
mes
57
58 )
  SELECT
60
      usuario_id,
61
     region,
62
     mes,
63
      entrante
64
65 FROM
      semanas
67 ORDER BY
      usuario_id,
68
      region,
69
   mes;
70
```

```
WITH compras as (
       SELECT
       FROM
            'tarea1.compra_fruta' as compra_fruta,
           'tarea1.compra_user' as Region
       WHERE
           Region.id_compra = compra_fruta.id_compra
   ),
9
10
   frutas_fecha as (
11
       SELECT
12
           fruta,
           fecha,
14
15
               cantidad,
16
           region,
           precio,
18
           usuario_id
19
       FROM
20
           compras,
21
            'tarea1.frutas' as frutas
22
23
           frutas.id_fruta = compras.id_fruta
24
   ),
25
26
   cantidad_frutas as (
27
       SELECT
28
           usuario_id,
29
           fruta,
30
           EXTRACT(MONTH FROM fecha) AS mes,
31
           SUM(cantidad*precio) AS total,
32
           region
33
       FROM
34
           frutas_fecha
35
       GROUP BY
36
37
           region,
           usuario_id,
38
           fruta,
39
           mes
       ORDER BY
41
           mes
42
   ),
43
44
   semanas as (
45
       SELECT
46
           usuario_id,
47
           region,
           fruta,
49
50
           sum(total) as entrante
51
       FROM
52
           cantidad_frutas
53
       GROUP BY
54
           region,
55
           usuario_id,
```

```
fruta,
57
           mes
58
       ORDER BY
           mes
60
  )
61
62
   SELECT
63
       usuario_id,
64
       region,
65
       fruta,
66
       mes,
67
       entrante
68
   FROM
69
       semanas
70
   ORDER BY
71
       usuario_id,
72
       region,
73
       mes;
```