

Optimización de Consultas Distribuidas en PostgreSQL

Laboratorio 15



Integrantes:

Fabrizzio Vilchez

Jeffrey Antonio Monja Castro

Universidad de Ingeniería y Tecnología

CS2042 Base de Datos II

Docente: Sanchez Enriquez, Heider Ysaías

2024-I

Contents

1. P1. Creación de Tablas Fragmentadas	2
1.1. Población de las tablas	2
1.2. Fragmentación de las tablas	4
1.2.1. Fragmentación en la tabla Venta	4
1.2.2. Fragmentación en la tabla Reclamo	5
2. P2. Algoritmos distribuidos localmente	5
2.1. Consulta 1	5
2.1.1. Sentencia SQL de la implementación del algoritmo distribuido optimizado	6
2.1.2. Gráfico del plan de ejecución resultante	8
2.2. Consulta 2	9
2.2.1. Sentencia SQL de la implementación del algoritmo distribuido optimizado	9
2.2.2. Gráfico del plan de ejecución resultante	11
2.3. Consulta 3	11
2.3.1. Sentencia SQL de la implementación del algoritmo distribuido optimizado	12
2.3.2. Gráfico del plan de ejecución resultante	14
2.4. Consulta 4	14
2.4.1. Sentencia SQL de la implementación del algoritmo distribuido optimizado	15
2.4.2. Gráfico del plan de ejecución resultante	17
3. P3. Algoritmos distribuidos en al menos tres servidores.	18
3.1. Consulta 1	18
3.2. Consulta 2	22
3.3. Consulta 3	25
3.4. Consulta 4	27

1 P1. Creación de Tablas Fragmentadas

Para este laboratorio se crearon dos tablas principales:

- **Venta**(IdVenta, DNI_Cliente, FechaVenta, CodLocal, ImporteTotal, IdEmpleado)
- **Reclamo**(IdReclamo, DNI_Cliente, FechaReclamo, CodLocal, Descripcion, Estado)

Estas tablas se van a fragmentar para posteriormente aplicar algoritmos de consultas distribuidas eficientes. Las consultas a realizar son las siguientes:

Consulta 1

Recuperar todas las tuplas de la tabla **Venta** ordenándolas en base al atributo **ImporteTotal** de manera descendente

```
1 SELECT *
2 FROM venta
3 ORDER BY ImporteTotal DESC;
```

Consulta 2

Recuperar todos los DNI_Cliente distintos de la tabla **Venta**

```
1 SELECT DISTINCT DNI_Cliente
2 FROM venta;
```

Consulta 3

Recuperar el CodLocal y el promedio de ImporteTotal por CodLocal ordenado por el CodLocal en la tabla **Venta**

```
1 SELECT CodLocal, AVG(ImporteTotal) AS promedio_importe
2 FROM venta
3 GROUP BY CodLocal
4 ORDER BY CodLocal;
```

Consulta 4

Recuperar todas las tuplas de la tabla **Reclamo** donde el DNI_Cliente se encuentre en la tabla **Venta**

```
1 SELECT R.*
2 FROM reclamo R
3 JOIN venta V
4 ON R.DNI_Cliente = V.DNI_Cliente;
```

1.1 Población de las tablas

Para poblar las tablas se utilizó un Script de Python para generar 1 millón de datos aleatorios para cada atributo con la ayuda de la librería faker, y también se insertaron en cada tabla con ayuda de la librería de psycopg2. Cabe resaltar que los datos se han insertado de manera uniforme.

Además, se realizaron algunas restricciones sobre los atributos para posteriormente realizar particiones adecuadas:

- Los Ids de las tablas son incrementales
- El atributo DNI_Cliente es un número de 8 dígitos entre 70000000 y 77500000.
- El atributo FechaVenta de la tabla **Venta** y FechaReclamo de la tabla **Reclamo** tienen valores entre 2000-01-01 y 2025-01-01.
- El atributo CodLocal son strings de la forma "LOC[00-12]".
- El atributo ImporteTotal de la tabla **Venta** es un entero entre 1 y 1000.

Script de Python:

```
1 import psycopg2
2 from faker import Faker
3 import random
4 from datetime import date
5
6 # Configuración de la conexión a la base de datos
7 conn = psycopg2.connect(
8     host="localhost",
9     database="db2",
10    user="postgres",
11    password="123"
12 )
13
14 cursor = conn.cursor()
15 schema = "lab15"
16 cursor.execute(f"SET search_path TO {schema}")
17
18 fake = Faker('es_ES')
19
20 ventas = []
21 reclamos = []
22
23 # 12 códigos diferentes para CodLocal
24 cod_locales = [f"LOC{str(i).zfill(2)}" for i in range(1, 13)]
25 start_date = date(2000, 1, 1)
26 end_date = date(2025, 1, 1)
27 estados = ['Pendiente', 'En Proceso', 'Cerrado']
28
29 # Generar datos aleatorios con id incremental
30 for i in range(1000000):
31     dni_cliente = str(random.randint(70000000, 77500000))
32     fecha_venta = fake.date_between(start_date=start_date, end_date=
33         end_date)
34     cod_local = random.choice(cod_locales)
35     importe_total = random.randint(1, 1000)
36     id_empleado = f"EMP{str(fake.random_number(digits=3, fix_len=True)
37         ).zfill(3)}"
38
39     ventas.append((i + 1, dni_cliente, fecha_venta, cod_local,
40         importe_total, id_empleado))
41
42     dni_cliente = str(random.randint(70000000, 79999999))
43     fecha_reclamo = fake.date_between(start_date=start_date, end_date=
44         end_date)
45     cod_local = random.choice(cod_locales)
46     descripcion = fake.text(max_nb_chars=100)
47     estado = random.choice(estados)
48
49     reclamos.append((i + 1, dni_cliente, fecha_reclamo, cod_local,
50         descripcion, estado))
51
52 # Insertar las tuplas en la base de datos
53 insert_query_venta = """
54 INSERT INTO venta (IdVenta, DNI_Cliente, FechaVenta, CodLocal,
55     ImporteTotal, IdEmpleado)
```

```

50 VALUES (%s, %s, %s, %s, %s, %s)
51 """
52
53 insert_query_reclamo = """
54 INSERT INTO reclamo (IdReclamo, DNI_Cliente, FechaReclamo, CodLocal,
55     Descripcion, Estado)
56 VALUES (%s, %s, %s, %s, %s, %s)
57 """
58
59 # Insertar los datos en bloques de 10000 tuplas
60 batch_size = 10000
61 for i in range(0, len(reclamos), batch_size):
62     batch_v = ventas[i:i + batch_size]
63     batch_r = reclamos[i:i + batch_size]
64     try:
65         cursor.executemany(insert_query_venta, batch_v)
66         cursor.executemany(insert_query_reclamo, batch_r)
67         conn.commit()
68     except Exception as e:
69         print(f"Error al insertar los datos: {e}")
70         conn.rollback()
71
72 print("Datos insertados correctamente en la tabla venta y reclamo")
73
74 # Cerrar la conexion
75 cursor.close()
76 conn.close()

```

1.2 Fragmentación de las tablas

Las particiones mejoran significativamente el rendimiento. Al dividir grandes tablas en partes más pequeñas, se optimiza el acceso, reduciendo el tiempo de respuesta de las consultas y facilitando la escalabilidad de la base de datos.

1.2.1 Fragmentación en la tabla Venta

Se utilizó la técnica de partición por rango en el atributo FechaVenta para crear tres particiones. El vector de partición utilizado es: [2009-01-01, 2017-01-01]

```

1  -- Crear la tabla venta particionada
2  CREATE TABLE venta (
3      IdVenta SERIAL,
4      DNI_Cliente INT,
5      FechaVenta DATE,
6      CodLocal VARCHAR(5),
7      ImporteTotal FLOAT,
8      IdEmpleado VARCHAR(6)
9  ) PARTITION BY RANGE (FechaVenta);
10
11 -- Crear particiones para la tabla venta
12 CREATE TABLE venta_2000_2008 PARTITION OF venta
13     FOR VALUES FROM ('2000-01-01') TO ('2009-01-01');
14
15 CREATE TABLE venta_2009_2016 PARTITION OF venta
16     FOR VALUES FROM ('2009-01-01') TO ('2017-01-01');
17
18 CREATE TABLE venta_2017_2024 PARTITION OF venta

```

```
19 FOR VALUES FROM ('2017-01-01') TO ('2025-01-01');
```

1.2.2 Fragmentación en la tabla Reclamo

Se utilizó la técnica de partición por lista en el atributo `CodLocal` para crear tres particiones. Las particiones se realizaron de la siguiente forma:

- Partición 1: ['LOC01', 'LOC02', 'LOC03', 'LOC04']
- Partición 2: ['LOC05', 'LOC06', 'LOC07', 'LOC08']
- Partición 3: ['LOC09', 'LOC10', 'LOC11', 'LOC12']

```
1 CREATE TABLE reclamo (  
2     IdReclamo SERIAL,  
3     DNI_Cliente INT,  
4     FechaReclamo DATE,  
5     CodLocal VARCHAR(5),  
6     Descripcion VARCHAR(100),  
7     Estado VARCHAR(10)  
8 ) PARTITION BY LIST (CodLocal);  
9  
10 -- Crear particiones para la tabla reclamo  
11 CREATE TABLE reclamo_loc1 PARTITION OF reclamo  
12     FOR VALUES IN ('LOC01', 'LOC02', 'LOC03', 'LOC04');  
13  
14 CREATE TABLE reclamo_loc2 PARTITION OF reclamo  
15     FOR VALUES IN ('LOC05', 'LOC06', 'LOC07', 'LOC08');  
16  
17 CREATE TABLE reclamo_loc3 PARTITION OF reclamo  
18     FOR VALUES IN ('LOC09', 'LOC10', 'LOC11', 'LOC12');
```

2 P2. Algoritmos distribuidos localmente

En este apartado se asumió que cada fragmento se encuentra en sitios diferentes, pero en realidad la consulta se ejecuta en el mismo servidor.

2.1 Consulta 1

El atributo `ImporteTotal` se generó en el rango de 1 a 1000, por lo que se tomó el vector de partición [333,666] en el atributo `ImporteTotal`

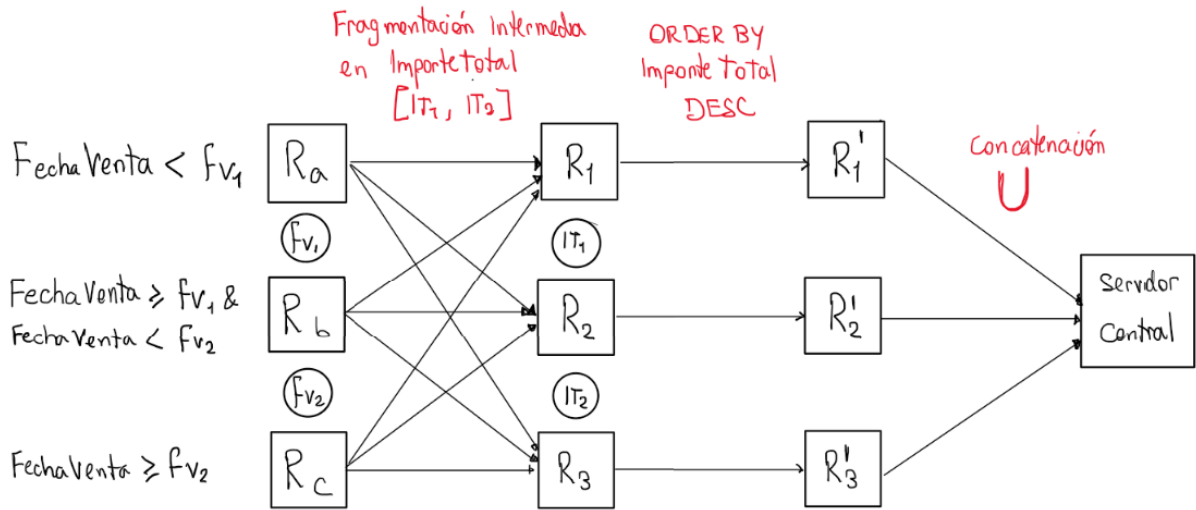


Figure 1: Optimización de la consulta 1 distribuida

2.1.1 Sentencia SQL de la implementación del algoritmo distribuido optimizado

```

1  SELECT *
2  FROM (
3      SELECT *
4      FROM (
5          SELECT V.*
6          FROM venta_2000_2008 V
7          WHERE V.ImporteTotal >= 666
8          UNION ALL
9          SELECT V.*
10         FROM venta_2009_2016 V
11         WHERE V.ImporteTotal >= 666
12         UNION ALL
13         SELECT V.*
14         FROM venta_2017_2024 V
15         WHERE V.ImporteTotal >= 666
16     ) AS temp_venta3
17     ORDER BY ImporteTotal DESC
18 ) AS temp_venta6
19
20 UNION ALL
21
22 SELECT *
23 FROM (
24     SELECT *
25     FROM (
26         SELECT V.*
27         FROM venta_2000_2008 V
28         WHERE V.ImporteTotal >= 333 AND V.ImporteTotal < 666
29         UNION ALL
30         SELECT V.*
31         FROM venta_2009_2016 V
32         WHERE V.ImporteTotal >= 333 AND V.ImporteTotal < 666
33         UNION ALL
34         SELECT V.*

```

```

35         FROM venta_2017_2024 V
36         WHERE V.ImporteTotal >= 333 AND V.ImporteTotal < 666
37     ) AS temp_venta2
38     ORDER BY ImporteTotal DESC
39 ) AS temp_venta5
40
41 UNION ALL
42
43 SELECT *
44 FROM (
45     SELECT *
46     FROM (
47         SELECT V.*
48         FROM venta_2000_2008 V
49         WHERE V.ImporteTotal < 333
50         UNION ALL
51         SELECT V.*
52         FROM venta_2009_2016 V
53         WHERE V.ImporteTotal < 333
54         UNION ALL
55         SELECT V.*
56         FROM venta_2017_2024 V
57         WHERE V.ImporteTotal < 333
58     ) AS temp_venta1
59     ORDER BY ImporteTotal DESC
60 ) AS temp_venta4;

```


2.1.2 Gráfico del plan de ejecución resultante

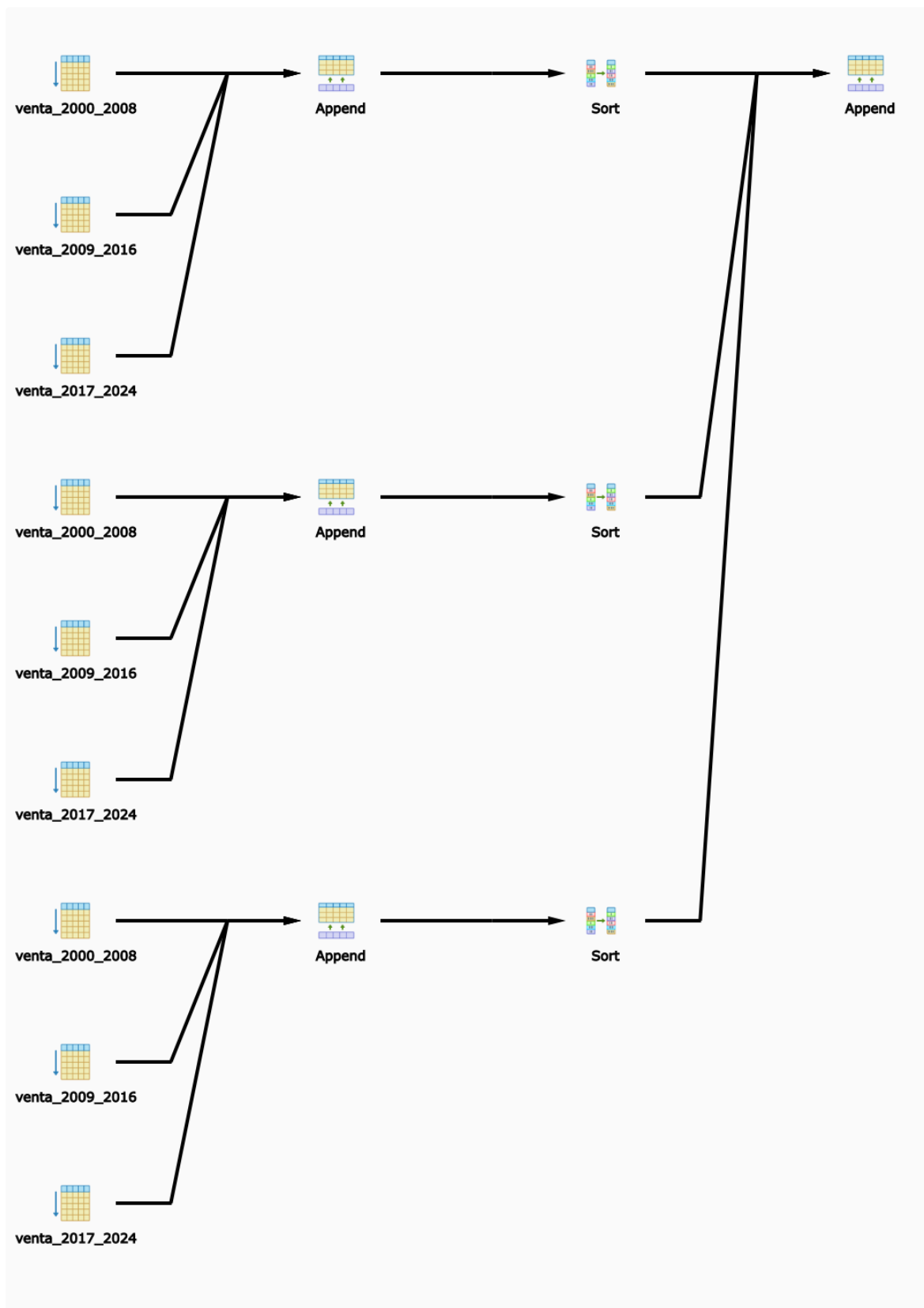


Figure 2: Grafico plan de ejecución en la consulta 1

2.2 Consulta 2

El atributo DNI_Cliente se generó en el rango de 70000000 a 77500000, por lo que se tomó el vector de partición [72500000, 75000000] en ese atributo.

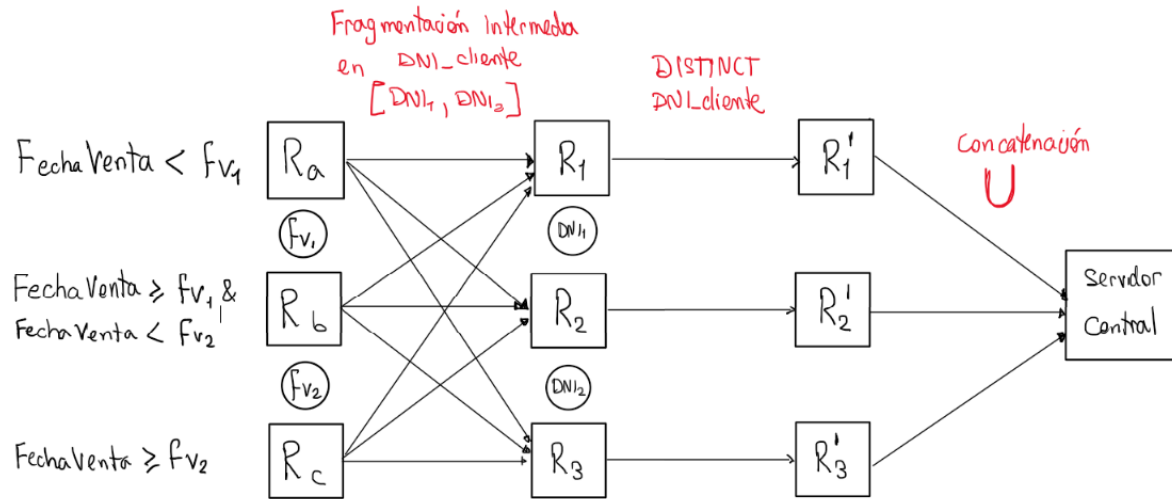


Figure 3: Optimización de la consulta 2 distribuida

2.2.1 Sentencia SQL de la implementación del algoritmo distribuido optimizado

```

1 SELECT *
2 FROM (
3     SELECT DISTINCT DNI_Cliente
4     FROM (
5         SELECT V.DNI_Cliente
6         FROM venta_2000_2008 V
7         WHERE V.DNI_Cliente >= 75000000
8         UNION ALL
9         SELECT V.DNI_Cliente
10        FROM venta_2009_2016 V
11        WHERE V.DNI_Cliente >= 75000000
12        UNION ALL
13        SELECT V.DNI_Cliente
14        FROM venta_2017_2024 V
15        WHERE V.DNI_Cliente >= 75000000
16    ) AS temp_venta3
17
18    UNION ALL
19
20    SELECT DISTINCT DNI_Cliente
21    FROM (
22        SELECT V.DNI_Cliente
23        FROM venta_2000_2008 V
24        WHERE V.DNI_Cliente >= 72500000 AND V.DNI_Cliente < 75000000
25        UNION ALL
26        SELECT V.DNI_Cliente
27        FROM venta_2009_2016 V
28        WHERE V.DNI_Cliente >= 72500000 AND V.DNI_Cliente < 75000000
29        UNION ALL

```

```

30      SELECT V.DNI_Cliente
31      FROM venta_2017_2024 V
32      WHERE V.DNI_Cliente >= 72500000 AND V.DNI_Cliente < 75000000
33  ) AS temp_venta2
34
35  UNION ALL
36
37  SELECT DISTINCT DNI_Cliente
38  FROM (
39      SELECT V.DNI_Cliente
40      FROM venta_2000_2008 V
41      WHERE V.DNI_Cliente < 72500000
42      UNION ALL
43      SELECT V.DNI_Cliente
44      FROM venta_2009_2016 V
45      WHERE V.DNI_Cliente < 72500000
46      UNION ALL
47      SELECT V.DNI_Cliente
48      FROM venta_2017_2024 V
49      WHERE V.DNI_Cliente < 72500000
50  ) AS temp_venta1
51  ) AS final_result;

```

2.2.2 Gráfico del plan de ejecución resultante

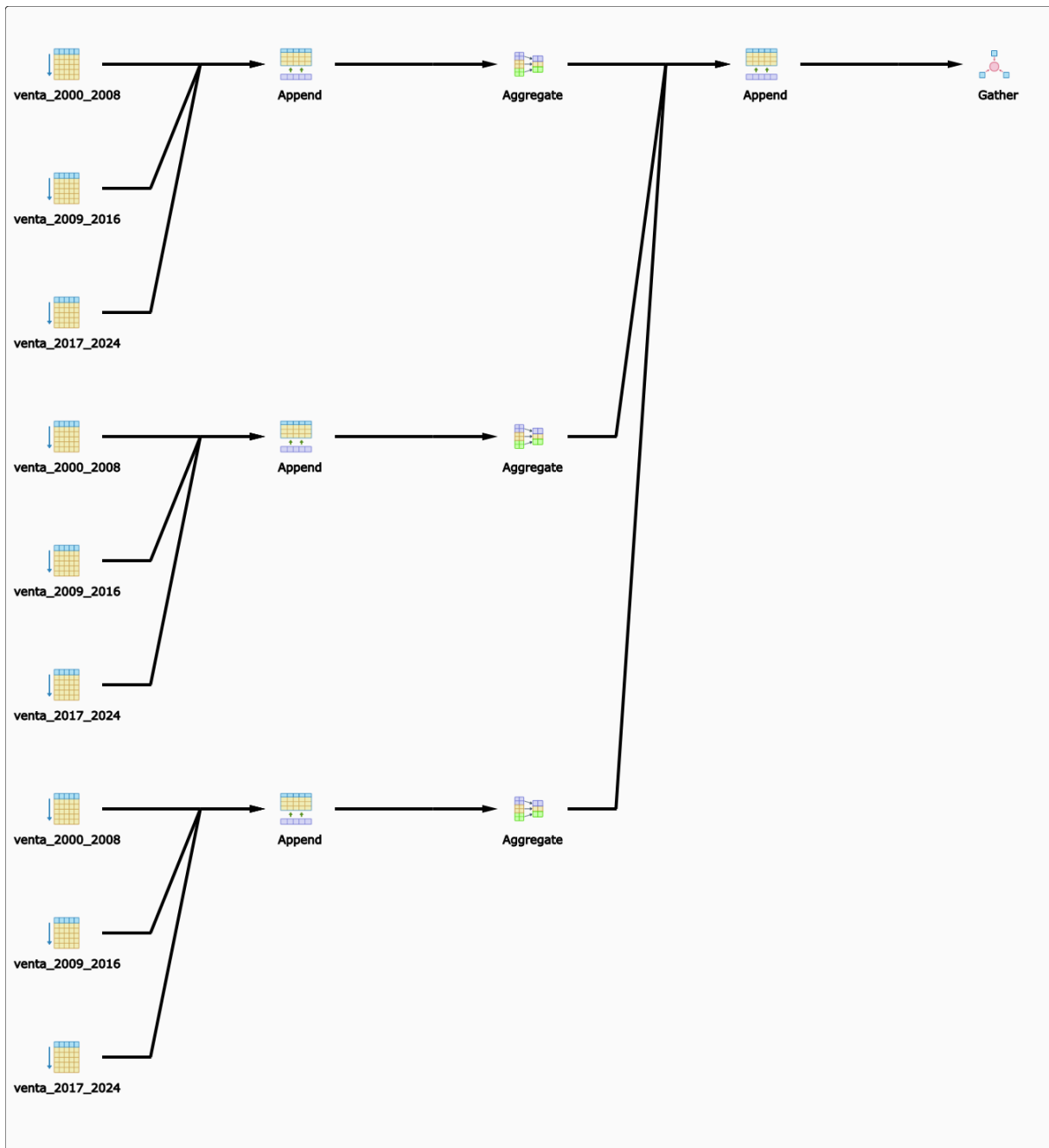


Figure 4: Grafico plan de ejecución en la consulta 2

2.3 Consulta 3

El atributo `CodLocal` se generó con los valores `LOC01` - `LOC12`, por lo que se dividió en 3 grupos de 4 valores cada uno: `[LOC01 a LOC04]`, `[LOC05 a LOC08]`, `[LOC09 a LOC12]`

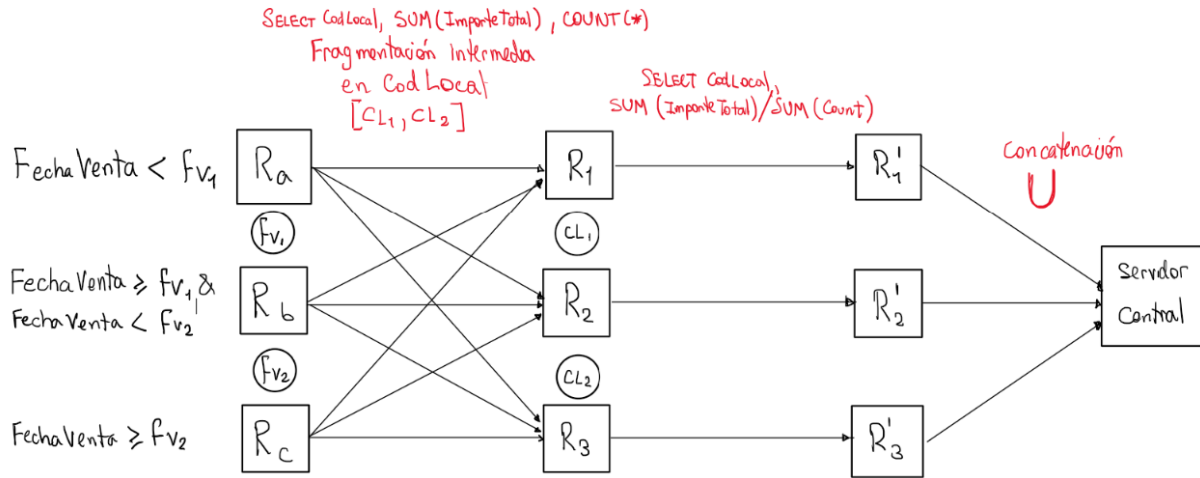


Figure 5: Optimización de la consulta 3 distribuida

2.3.1 Sentencia SQL de la implementación del algoritmo distribuido optimizado

```

1 SELECT *
2 FROM (
3     SELECT CodLocal, SUM(SumaImporte) / SUM(CantVentas) AS
4     PromedioImporte
5     FROM (
6         SELECT CodLocal, SUM(ImporteTotal) AS SumaImporte, COUNT(*) AS
7         CantVentas
8         FROM (
9             SELECT V.CodLocal, V.ImporteTotal
10            FROM venta_2000_2008 V
11            WHERE V.CodLocal IN ('LOC01', 'LOC02', 'LOC03', 'LOC04')
12            UNION ALL
13            SELECT V.CodLocal, V.ImporteTotal
14            FROM venta_2009_2016 V
15            WHERE V.CodLocal IN ('LOC01', 'LOC02', 'LOC03', 'LOC04')
16            UNION ALL
17            SELECT V.CodLocal, V.ImporteTotal
18            FROM venta_2017_2024 V
19            WHERE V.CodLocal IN ('LOC01', 'LOC02', 'LOC03', 'LOC04')
20        ) AS temp_vental
21        GROUP BY CodLocal
22    ) AS aggregated_temp_vental
23    GROUP BY CodLocal
24
25 UNION ALL
26
27 SELECT CodLocal, SUM(SumaImporte) / SUM(CantVentas) AS
28 PromedioImporte
29 FROM (
30     SELECT CodLocal, SUM(ImporteTotal) AS SumaImporte, COUNT(*) AS
31     CantVentas
32     FROM (
33         SELECT V.CodLocal, V.ImporteTotal
34         FROM venta_2000_2008 V
35         WHERE V.CodLocal IN ('LOC05', 'LOC06', 'LOC07', 'LOC08')

```

```

32         UNION ALL
33         SELECT V.CodLocal, V.ImporteTotal
34         FROM venta_2009_2016 V
35         WHERE V.CodLocal IN ('LOC05', 'LOC06', 'LOC07', 'LOC08')
36         UNION ALL
37         SELECT V.CodLocal, V.ImporteTotal
38         FROM venta_2017_2024 V
39         WHERE V.CodLocal IN ('LOC05', 'LOC06', 'LOC07', 'LOC08')
40     ) AS temp_venta2
41     GROUP BY CodLocal
42 ) AS aggregated_temp_venta2
43 GROUP BY CodLocal
44
45 UNION ALL
46
47 SELECT CodLocal, SUM(SumaImporte) / SUM(CantVentas) AS
PromedioImporte
48 FROM (
49     SELECT CodLocal, SUM(ImporteTotal) AS SumaImporte, COUNT(*) AS
CantVentas
50     FROM (
51         SELECT V.CodLocal, V.ImporteTotal
52         FROM venta_2000_2008 V
53         WHERE V.CodLocal IN ('LOC09', 'LOC10', 'LOC11', 'LOC12')
54         UNION ALL
55         SELECT V.CodLocal, V.ImporteTotal
56         FROM venta_2009_2016 V
57         WHERE V.CodLocal IN ('LOC09', 'LOC10', 'LOC11', 'LOC12')
58         UNION ALL
59         SELECT V.CodLocal, V.ImporteTotal
60         FROM venta_2017_2024 V
61         WHERE V.CodLocal IN ('LOC09', 'LOC10', 'LOC11', 'LOC12')
62     ) AS temp_venta3
63     GROUP BY CodLocal
64 ) AS aggregated_temp_venta3
65 GROUP BY CodLocal
66 ) AS final_result;

```

2.3.2 Gráfico del plan de ejecución resultante

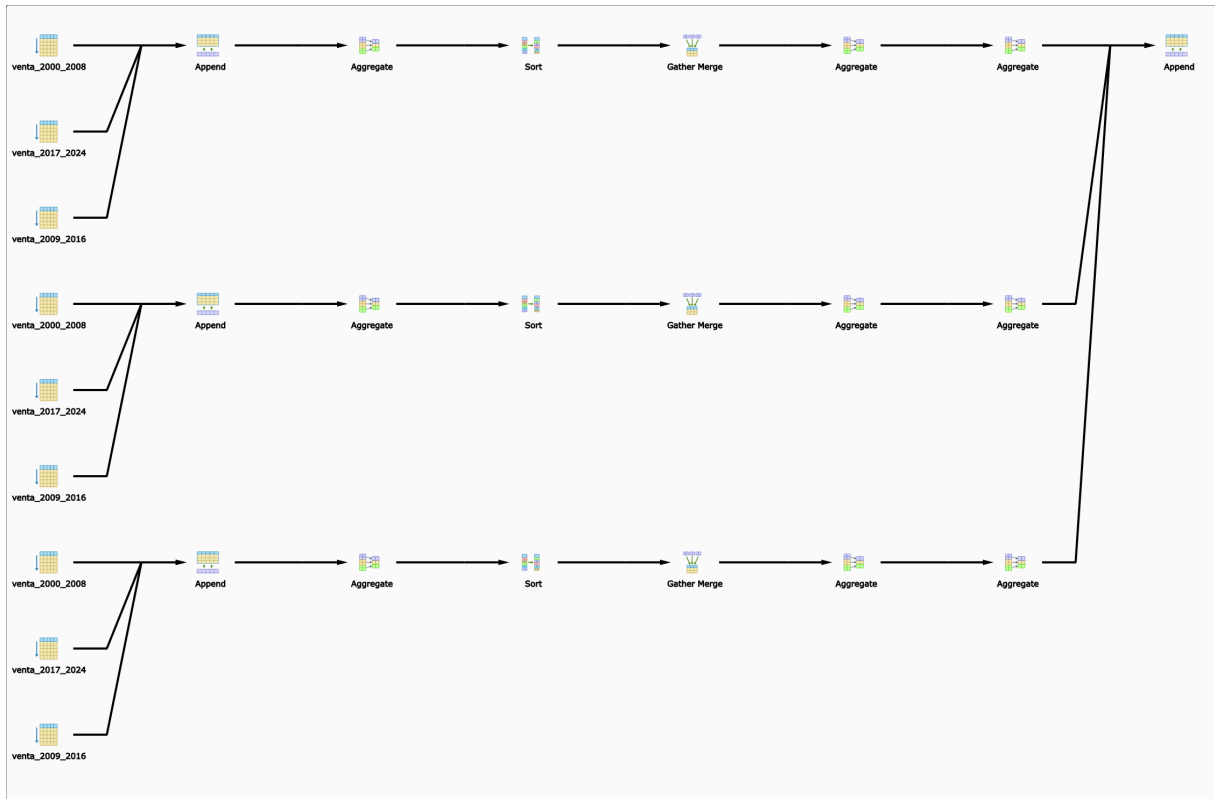


Figure 6: Grafico plan de ejecución en la consulta 3

2.4 Consulta 4

El atributo `DNI_Cliente` se generó en el rango de 70000000 a 77500000, por lo que se tomó el vector de partición `[72500000, 75000000]` en ese atributo.

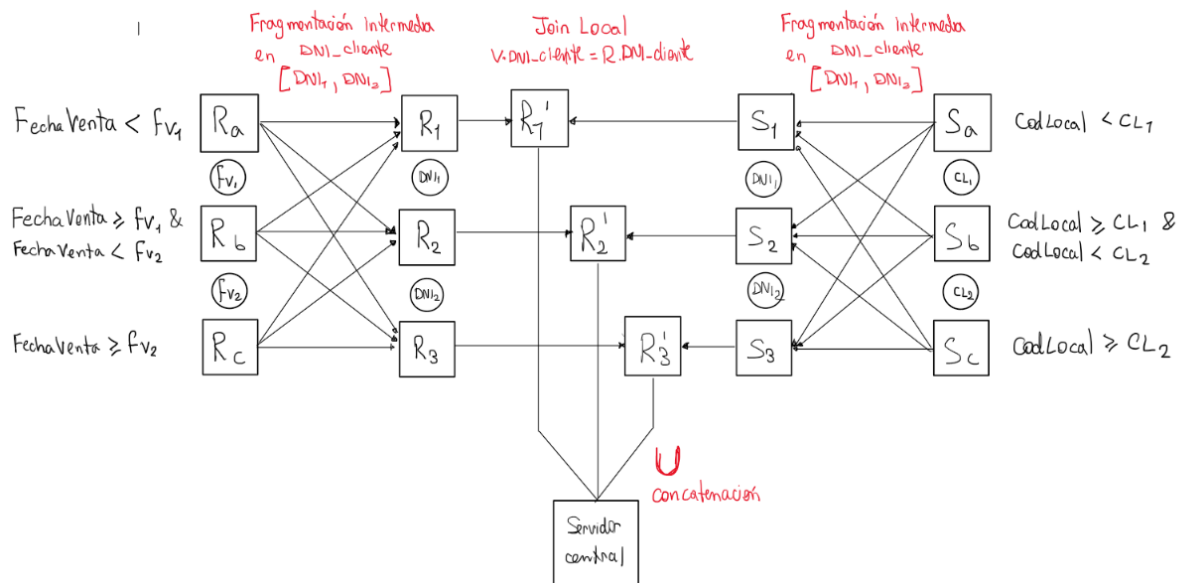


Figure 7: Optimización de la consulta 4 distribuida

2.4.1 Sentencia SQL de la implementación del algoritmo distribuido optimizado

```
1 SELECT *
2 FROM (
3     SELECT *
4     FROM (
5         SELECT R.*
6         FROM reclamo_loc1 R
7         WHERE R.DNI_Cliente < 72500000
8         UNION ALL
9         SELECT R.*
10        FROM reclamo_loc2 R
11        WHERE R.DNI_Cliente < 72500000
12        UNION ALL
13        SELECT R.*
14        FROM reclamo_loc3 R
15        WHERE R.DNI_Cliente < 72500000
16    ) AS temp_reclamo1
17    JOIN (
18        SELECT DNI_Cliente
19        FROM venta_2000_2008 V
20        WHERE V.DNI_Cliente < 72500000
21        UNION ALL
22        SELECT DNI_Cliente
23        FROM venta_2009_2016 V
24        WHERE V.DNI_Cliente < 72500000
25        UNION ALL
26        SELECT DNI_Cliente
27        FROM venta_2017_2024 V
28        WHERE V.DNI_Cliente < 72500000
29    ) AS temp_venta1
30    ON temp_reclamo1.DNI_Cliente = temp_venta1.DNI_Cliente
31
32    UNION ALL
33
34    SELECT *
35    FROM (
36        SELECT R.*
37        FROM reclamo_loc1 R
38        WHERE R.DNI_Cliente >= 72500000 AND R.DNI_Cliente < 75000000
39        UNION ALL
40        SELECT R.*
41        FROM reclamo_loc2 R
42        WHERE R.DNI_Cliente >= 72500000 AND R.DNI_Cliente < 75000000
43        UNION ALL
44        SELECT R.*
45        FROM reclamo_loc3 R
46        WHERE R.DNI_Cliente >= 72500000 AND R.DNI_Cliente < 75000000
47    ) AS temp_reclamo2
48    JOIN (
49        SELECT DNI_Cliente
50        FROM venta_2000_2008 V
51        WHERE V.DNI_Cliente >= 72500000 AND V.DNI_Cliente < 75000000
52        UNION ALL
53        SELECT DNI_Cliente
54        FROM venta_2009_2016 V
```



```

55         WHERE V.DNI_Cliente >= 72500000 AND V.DNI_Cliente < 75000000
56     UNION ALL
57     SELECT DNI_Cliente
58     FROM venta_2017_2024 V
59     WHERE V.DNI_Cliente >= 72500000 AND V.DNI_Cliente < 75000000
60 ) AS temp_venta2
61 ON temp_reclamo2.DNI_Cliente = temp_venta2.DNI_Cliente
62
63 UNION ALL
64
65 SELECT *
66 FROM (
67     SELECT R.*
68     FROM reclamo_loc1 R
69     WHERE R.DNI_Cliente >= 75000000
70     UNION ALL
71     SELECT R.*
72     FROM reclamo_loc2 R
73     WHERE R.DNI_Cliente >= 75000000
74     UNION ALL
75     SELECT R.*
76     FROM reclamo_loc3 R
77     WHERE R.DNI_Cliente >= 75000000
78 ) AS temp_reclamo3
79 JOIN (
80     SELECT DNI_Cliente
81     FROM venta_2000_2008 V
82     WHERE V.DNI_Cliente >= 75000000
83     UNION ALL
84     SELECT DNI_Cliente
85     FROM venta_2009_2016 V
86     WHERE V.DNI_Cliente >= 75000000
87     UNION ALL
88     SELECT DNI_Cliente
89     FROM venta_2017_2024 V
90     WHERE V.DNI_Cliente >= 75000000
91 ) AS temp_venta3
92 ON temp_reclamo3.DNI_Cliente = temp_venta3.DNI_Cliente
93 ) AS final_result;

```

2.4.2 Gráfico del plan de ejecución resultante

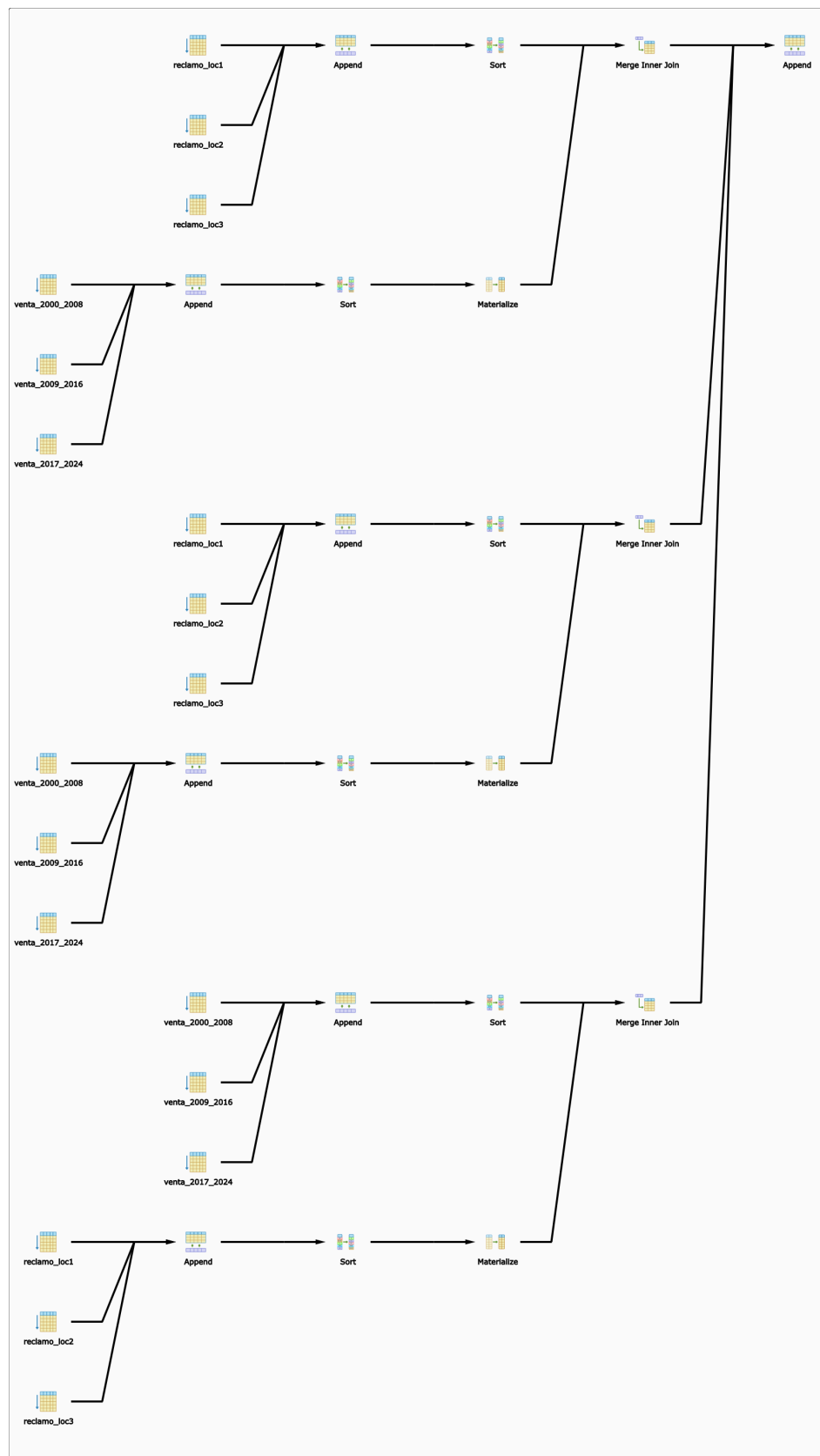


Figure 8: Grafico plan de ejecución en la consulta 4

3 P3. Algoritmos distribuidos en al menos tres servidores

Para este caso, se estan usando 3 contenedores de Docker para simular 3 servidores remotos y ejecutar todas las consultas ahí.

```
version: '3.8'

services:
  postgres1:
    image: postgres:latest
    container_name: postgres1
    environment:
      POSTGRES_USER: user1
      POSTGRES_PASSWORD: password1
      POSTGRES_DB: db1
    ports:
      - "5433:5432"
    networks:
      - postgres-network

  postgres2:
    image: postgres:latest
    container_name: postgres2
    environment:
      POSTGRES_USER: user2
      POSTGRES_PASSWORD: password2
      POSTGRES_DB: db2
    ports:
      - "5434:5432"
    networks:
      - postgres-network

  postgres3:
    image: postgres:latest
    container_name: postgres3
    environment:
      POSTGRES_USER: user3
      POSTGRES_PASSWORD: password3
      POSTGRES_DB: db3
    ports:
      - "5435:5432"
    networks:
      - postgres-network

networks:
  postgres-network:
    driver: bridge
```

3.1 Consulta 1

```
1 SELECT *
2 FROM (
3     SELECT *
4     FROM (
5         SELECT V.*
6         FROM dblink('dbname=db1 user=user1 password=password1 host=
7                     localhost port=5433',
                     'SELECT * FROM venta_2000_2008 WHERE ImporteTotal
                     >= 666')
```

```

8      AS V(IdVenta INT, DNI_Cliente INT, FechaVenta DATE, CodLocal
9          VARCHAR, ImporteTotal FLOAT, IdEmpleado VARCHAR)
10     UNION ALL
11     SELECT V.*
12     FROM dblink('dbname=db2 user=user2 password=password2 host=
13         localhost port=5434',
14         'SELECT * FROM venta_2009_2016 WHERE ImporteTotal
15             >= 666')
16     AS V(IdVenta INT, DNI_Cliente INT, FechaVenta DATE, CodLocal
17         VARCHAR, ImporteTotal FLOAT, IdEmpleado VARCHAR)
18     UNION ALL
19     SELECT V.*
20     FROM dblink('dbname=db3 user=user3 password=password3 host=
21         localhost port=5435',
22         'SELECT * FROM venta_2017_2024 WHERE ImporteTotal
23             >= 666')
24     AS V(IdVenta INT, DNI_Cliente INT, FechaVenta DATE, CodLocal
25         VARCHAR, ImporteTotal FLOAT, IdEmpleado VARCHAR)
26 ) AS temp_venta3
27 ORDER BY ImporteTotal DESC
28 ) AS temp_venta6
29
30 UNION ALL
31
32 SELECT *
33 FROM (
34     SELECT *
35     FROM (
36         SELECT V.*
37         FROM dblink('dbname=db1 user=user1 password=password1 host=
38             localhost port=5433',
39             'SELECT * FROM venta_2000_2008 WHERE ImporteTotal
40                 >= 333 AND ImporteTotal < 666')
41         AS V(IdVenta INT, DNI_Cliente INT, FechaVenta DATE, CodLocal
42             VARCHAR, ImporteTotal FLOAT, IdEmpleado VARCHAR)
43         UNION ALL
44         SELECT V.*
45         FROM dblink('dbname=db2 user=user2 password=password2 host=
46             localhost port=5434',
47             'SELECT * FROM venta_2009_2016 WHERE ImporteTotal
48                 >= 333 AND ImporteTotal < 666')
49         AS V(IdVenta INT, DNI_Cliente INT, FechaVenta DATE, CodLocal
50             VARCHAR, ImporteTotal FLOAT, IdEmpleado VARCHAR)
51         UNION ALL
52         SELECT V.*
53         FROM dblink('dbname=db3 user=user3 password=password3 host=
54             localhost port=5435',
55             'SELECT * FROM venta_2017_2024 WHERE ImporteTotal
56                 >= 333 AND ImporteTotal < 666')
57         AS V(IdVenta INT, DNI_Cliente INT, FechaVenta DATE, CodLocal
58             VARCHAR, ImporteTotal FLOAT, IdEmpleado VARCHAR)
59     ) AS temp_venta2
60     ORDER BY ImporteTotal DESC
61 ) AS temp_venta5
62
63 UNION ALL

```

```

48 SELECT *
49 FROM (
50     SELECT *
51     FROM (
52         SELECT V.*
53         FROM dblink('dbname=db1 user=user1 password=password1 host=
54                     localhost port=5433',
55                     'SELECT * FROM venta_2000_2008 WHERE ImporteTotal
56                     < 333')
57         AS V(IdVenta INT, DNI_Cliente INT, FechaVenta DATE, CodLocal
58             VARCHAR, ImporteTotal FLOAT, IdEmpleado VARCHAR)
59         UNION ALL
60         SELECT V.*
61         FROM dblink('dbname=db2 user=user2 password=password2 host=
62                     localhost port=5434',
63                     'SELECT * FROM venta_2009_2016 WHERE ImporteTotal
64                     < 333')
65         AS V(IdVenta INT, DNI_Cliente INT, FechaVenta DATE, CodLocal
66             VARCHAR, ImporteTotal FLOAT, IdEmpleado VARCHAR)
67         UNION ALL
68         SELECT V.*
69         FROM dblink('dbname=db3 user=user3 password=password3 host=
70                     localhost port=5435',
71                     'SELECT * FROM venta_2017_2024 WHERE ImporteTotal
72                     < 333')
73         AS V(IdVenta INT, DNI_Cliente INT, FechaVenta DATE, CodLocal
74             VARCHAR, ImporteTotal FLOAT, IdEmpleado VARCHAR)
75     ) AS temp_venta1
76     ORDER BY ImporteTotal DESC
77 ) AS temp_venta4
78 ORDER BY ImporteTotal DESC;

```



Figure 9: Optimización de la consulta 1 distribuida en diferentes servidores

3.2 Consulta 2

```
1 SELECT DISTINCT DNI_Cliente
2 FROM (
3     SELECT *
4     FROM (
5         SELECT V.DNI_Cliente
6         FROM dblink('dbname=db1 user=user1 password=password1 host=
7                     localhost port=5433',
8                     'SELECT DNI_Cliente FROM venta_2000_2008 WHERE
9                     DNI_Cliente >= 75000000')
10        AS V(DNI_Cliente INT)
11     UNION ALL
12     SELECT V.DNI_Cliente
13     FROM dblink('dbname=db2 user=user2 password=password2 host=
14                 localhost port=5434',
15                 'SELECT DNI_Cliente FROM venta_2009_2016 WHERE
16                 DNI_Cliente >= 75000000')
17        AS V(DNI_Cliente INT)
18     UNION ALL
19     SELECT V.DNI_Cliente
20     FROM dblink('dbname=db3 user=user3 password=password3 host=
21                 localhost port=5435',
22                 'SELECT DNI_Cliente FROM venta_2017_2024 WHERE
23                 DNI_Cliente >= 75000000')
24        AS V(DNI_Cliente INT)
25    ) AS temp_venta3
26 ) AS temp_venta6
27
28 UNION ALL
29
30 SELECT DISTINCT DNI_Cliente
31 FROM (
32     SELECT *
33     FROM (
34         SELECT V.DNI_Cliente
35         FROM dblink('dbname=db1 user=user1 password=password1 host=
36                     localhost port=5433',
37                     'SELECT DNI_Cliente FROM venta_2000_2008 WHERE
38                     DNI_Cliente >= 72500000 AND DNI_Cliente <
39                     75000000')
40        AS V(DNI_Cliente INT)
41     UNION ALL
42     SELECT V.DNI_Cliente
43     FROM dblink('dbname=db2 user=user2 password=password2 host=
44                 localhost port=5434',
45                 'SELECT DNI_Cliente FROM venta_2009_2016 WHERE
46                 DNI_Cliente >= 72500000 AND DNI_Cliente <
47                 75000000')
48        AS V(DNI_Cliente INT)
49     UNION ALL
50     SELECT V.DNI_Cliente
51     FROM dblink('dbname=db3 user=user3 password=password3 host=
52                 localhost port=5435',
53                 'SELECT DNI_Cliente FROM venta_2017_2024 WHERE
54                 DNI_Cliente >= 72500000 AND DNI_Cliente <
```

```

41         75000000')
42     AS V(DNI_Cliente INT)
43 ) AS temp_venta2
44 ) AS temp_venta5
45 UNION ALL
46
47 SELECT DISTINCT DNI_Cliente
48 FROM (
49     SELECT *
50     FROM (
51         SELECT V.DNI_Cliente
52         FROM dblink('dbname=db1 user=user1 password=password1 host=
53             localhost port=5433',
54             'SELECT DNI_Cliente FROM venta_2000_2008 WHERE
55                 DNI_Cliente < 72500000')
56         AS V(DNI_Cliente INT)
57     UNION ALL
58     SELECT V.DNI_Cliente
59     FROM dblink('dbname=db2 user=user2 password=password2 host=
60         localhost port=5434',
61         'SELECT DNI_Cliente FROM venta_2009_2016 WHERE
62             DNI_Cliente < 72500000')
63     AS V(DNI_Cliente INT)
64     UNION ALL
65     SELECT V.DNI_Cliente
66     FROM dblink('dbname=db3 user=user3 password=password3 host=
67         localhost port=5435',
68         'SELECT DNI_Cliente FROM venta_2017_2024 WHERE
69             DNI_Cliente < 72500000')
70     AS V(DNI_Cliente INT)
71 ) AS temp_venta1
72 ) AS temp_venta4;

```




Figure 10: Optimización de la consulta 2 distribuida en diferentes servidores

3.3 Consulta 3

```
1 SELECT CodLocal, SUM(SumaImporte) / SUM(CantVentas) AS PromedioImporte
2 FROM (
3     SELECT CodLocal, SUM(ImporteTotal) AS SumaImporte, COUNT(*) AS
4     CantVentas
5     FROM (
6         SELECT V.CodLocal, V.ImporteTotal
7         FROM dblink('dbname=db1 user=user1 password=password1 host=
8             localhost port=5433',
9             'SELECT CodLocal, ImporteTotal FROM
10                 venta_2000_2008 WHERE CodLocal IN (''LOC01'', '
11                 'LOC02'', ''LOC03'', ''LOC04'')')
12         AS V(CodLocal VARCHAR, ImporteTotal FLOAT)
13     UNION ALL
14     SELECT V.CodLocal, V.ImporteTotal
15     FROM dblink('dbname=db2 user=user2 password=password2 host=
16         localhost port=5434',
17         'SELECT CodLocal, ImporteTotal FROM
18             venta_2009_2016 WHERE CodLocal IN (''LOC01'', '
19             'LOC02'', ''LOC03'', ''LOC04'')')
20     AS V(CodLocal VARCHAR, ImporteTotal FLOAT)
21     UNION ALL
22     SELECT V.CodLocal, V.ImporteTotal
23     FROM dblink('dbname=db3 user=user3 password=password3 host=
24         localhost port=5435',
25         'SELECT CodLocal, ImporteTotal FROM
26             venta_2017_2024 WHERE CodLocal IN (''LOC01'', '
27             'LOC02'', ''LOC03'', ''LOC04'')')
28     AS V(CodLocal VARCHAR, ImporteTotal FLOAT)
29 ) AS temp_venta1
30 GROUP BY CodLocal
31 ) AS aggregated_temp_venta1
32 GROUP BY CodLocal
33
34 UNION ALL
35
36 SELECT CodLocal, SUM(SumaImporte) / SUM(CantVentas) AS PromedioImporte
37 FROM (
38     SELECT CodLocal, SUM(ImporteTotal) AS SumaImporte, COUNT(*) AS
39     CantVentas
40     FROM (
41         SELECT V.CodLocal, V.ImporteTotal
42         FROM dblink('dbname=db1 user=user1 password=password1 host=
43             localhost port=5433',
44             'SELECT CodLocal, ImporteTotal FROM
45                 venta_2000_2008 WHERE CodLocal IN (''LOC05'', '
46                 'LOC06'', ''LOC07'', ''LOC08'')')
47         AS V(CodLocal VARCHAR, ImporteTotal FLOAT)
48     UNION ALL
49     SELECT V.CodLocal, V.ImporteTotal
50     FROM dblink('dbname=db2 user=user2 password=password2 host=
51         localhost port=5434',
52         'SELECT CodLocal, ImporteTotal FROM
53             venta_2009_2016 WHERE CodLocal IN (''LOC05'', '
54             'LOC06'', ''LOC07'', ''LOC08'')')
```

```

38     AS V(CodLocal VARCHAR, ImporteTotal FLOAT)
39     UNION ALL
40     SELECT V.CodLocal, V.ImporteTotal
41     FROM dblink('dbname=db3 user=user3 password=password3 host=
42                 localhost port=5435',
43                 'SELECT CodLocal, ImporteTotal FROM
44                  venta_2017_2024 WHERE CodLocal IN (''LOC05'', '
45                  'LOC06'', ''LOC07'', ''LOC08'')')
46     AS V(CodLocal VARCHAR, ImporteTotal FLOAT)
47 ) AS temp_venta2
48     GROUP BY CodLocal
49 ) AS aggregated_temp_venta2
50     GROUP BY CodLocal
51 UNION ALL
52 SELECT CodLocal, SUM(SumaImporte) / SUM(CantVentas) AS PromedioImporte
53 FROM (
54     SELECT CodLocal, SUM(ImporteTotal) AS SumaImporte, COUNT(*) AS
55     CantVentas
56 FROM (
57     SELECT V.CodLocal, V.ImporteTotal
58     FROM dblink('dbname=db1 user=user1 password=password1 host=
59                 localhost port=5433',
60                 'SELECT CodLocal, ImporteTotal FROM
61                  venta_2000_2008 WHERE CodLocal IN (''LOC09'', '
62                  'LOC10'', ''LOC11'', ''LOC12'')')
63     AS V(CodLocal VARCHAR, ImporteTotal FLOAT)
64     UNION ALL
65     SELECT V.CodLocal, V.ImporteTotal
66     FROM dblink('dbname=db2 user=user2 password=password2 host=
67                 localhost port=5434',
68                 'SELECT CodLocal, ImporteTotal FROM
69                  venta_2009_2016 WHERE CodLocal IN (''LOC09'', '
70                  'LOC10'', ''LOC11'', ''LOC12'')')
71     AS V(CodLocal VARCHAR, ImporteTotal FLOAT)
72     UNION ALL
73     SELECT V.CodLocal, V.ImporteTotal
74     FROM dblink('dbname=db3 user=user3 password=password3 host=
75                 localhost port=5435',
76                 'SELECT CodLocal, ImporteTotal FROM
77                  venta_2017_2024 WHERE CodLocal IN (''LOC09'', '
78                  'LOC10'', ''LOC11'', ''LOC12'')')
79     AS V(CodLocal VARCHAR, ImporteTotal FLOAT)
80 ) AS temp_venta3
81     GROUP BY CodLocal
82 ) AS aggregated_temp_venta3
83     GROUP BY CodLocal;

```

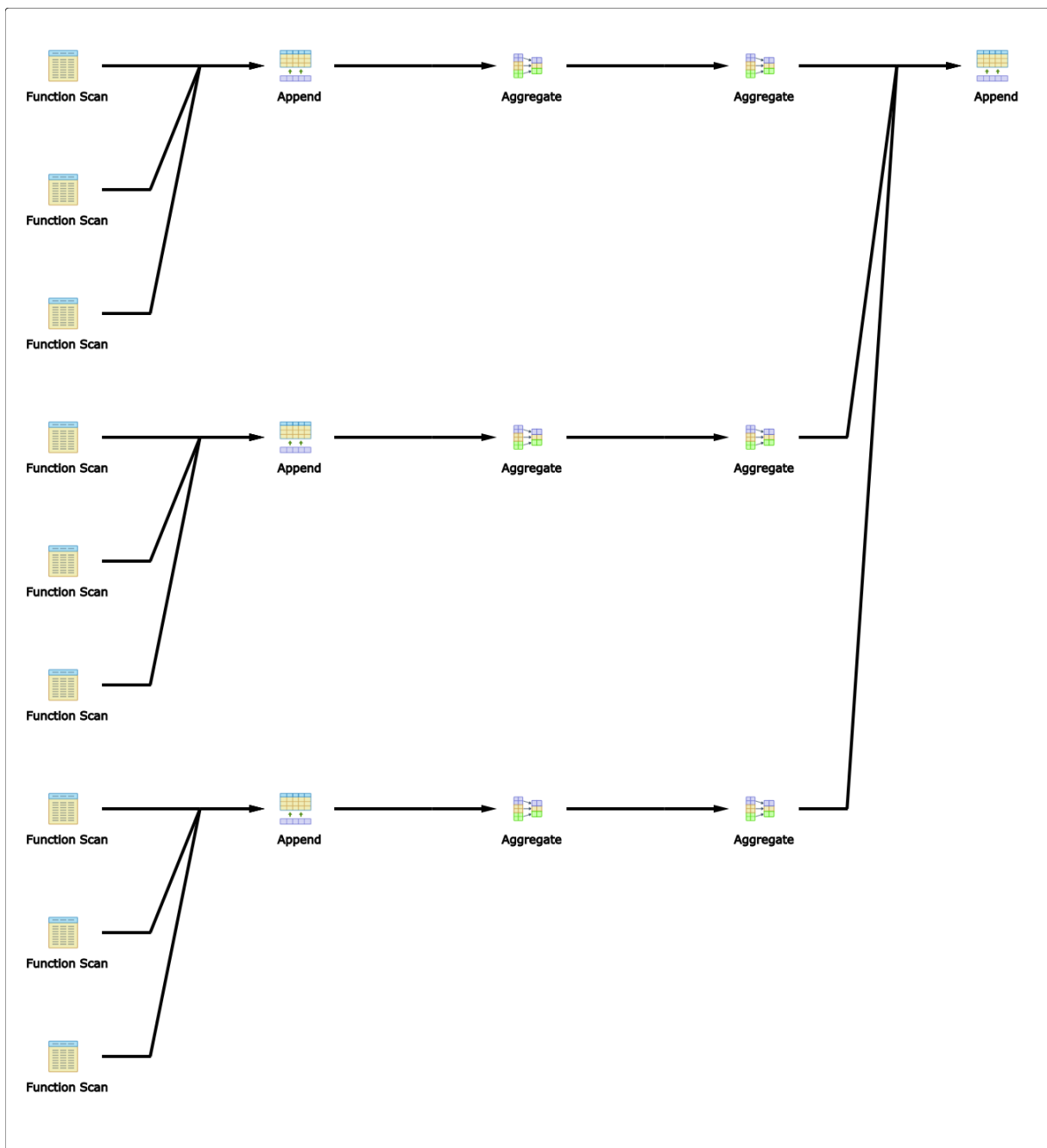


Figure 11: Optimización de la consulta 3 distribuida en diferentes servidores

3.4 Consulta 4

```

1 SELECT *
2 FROM (
3     SELECT *
4     FROM (
5         SELECT R.*
6         FROM dblink('dbname=db1 user=user1 password=password1 host=
7                  localhost port=5433',
                   'SELECT * FROM reclamo_loc1_loc4 WHERE DNI_Cliente
                   < 72500000')

```

```

8      AS R(IdReclamo INT, DNI_Cliente INT, FechaReclamo DATE,
9          CodLocal VARCHAR, Descripcion VARCHAR, Estado VARCHAR)
10     UNION ALL
11     SELECT R.*
12     FROM dblink('dbname=db2 user=user2 password=password2 host=
13         localhost port=5434',
14         'SELECT * FROM reclamo_loc5_loc8 WHERE DNI_Cliente
15             < 72500000')
16     AS R(IdReclamo INT, DNI_Cliente INT, FechaReclamo DATE,
17         CodLocal VARCHAR, Descripcion VARCHAR, Estado VARCHAR)
18     UNION ALL
19     SELECT R.*
20     FROM dblink('dbname=db3 user=user3 password=password3 host=
21         localhost port=5435',
22         'SELECT * FROM reclamo_loc9_loc12 WHERE
23             DNI_Cliente < 72500000')
24     AS R(IdReclamo INT, DNI_Cliente INT, FechaReclamo DATE,
25         CodLocal VARCHAR, Descripcion VARCHAR, Estado VARCHAR)
26 ) AS temp_reclamo1
27 JOIN (
28     SELECT DNI_Cliente
29     FROM dblink('dbname=db1 user=user1 password=password1 host=
30         localhost port=5433',
31         'SELECT DNI_Cliente FROM venta_2000_2008 WHERE
32             DNI_Cliente < 72500000')
33     AS V(DNI_Cliente INT)
34     UNION ALL
35     SELECT DNI_Cliente
36     FROM dblink('dbname=db2 user=user2 password=password2 host=
37         localhost port=5434',
38         'SELECT DNI_Cliente FROM venta_2009_2016 WHERE
39             DNI_Cliente < 72500000')
40     AS V(DNI_Cliente INT)
41     UNION ALL
42     SELECT DNI_Cliente
43     FROM dblink('dbname=db3 user=user3 password=password3 host=
44         localhost port=5435',
45         'SELECT DNI_Cliente FROM venta_2017_2024 WHERE
46             DNI_Cliente < 72500000')
47     AS V(DNI_Cliente INT)
48 ) AS temp_venta1
49 ON temp_reclamo1.DNI_Cliente = temp_venta1.DNI_Cliente
50 ) AS result1
51
52 UNION ALL
53
54 SELECT *
55 FROM (
56     SELECT *
57     FROM (
58         SELECT R.*
59         FROM dblink('dbname=db1 user=user1 password=password1 host=
60             localhost port=5433',
61             'SELECT * FROM reclamo_loc1_loc4 WHERE DNI_Cliente
62                 >= 72500000 AND DNI_Cliente < 75000000')

```

```

48      AS R(IdReclamo INT, DNI_Cliente INT, FechaReclamo DATE,
49          CodLocal VARCHAR, Descripcion VARCHAR, Estado VARCHAR)
50      UNION ALL
51      SELECT R.*
52      FROM dblink('dbname=db2 user=user2 password=password2 host=
53          localhost port=5434',
54          'SELECT * FROM reclamo_loc5_loc8 WHERE DNI_Cliente
55              >= 72500000 AND DNI_Cliente < 75000000')
56      AS R(IdReclamo INT, DNI_Cliente INT, FechaReclamo DATE,
57          CodLocal VARCHAR, Descripcion VARCHAR, Estado VARCHAR)
58      UNION ALL
59      SELECT R.*
60      FROM dblink('dbname=db3 user=user3 password=password3 host=
61          localhost port=5435',
62          'SELECT * FROM reclamo_loc9_loc12 WHERE
63              DNI_Cliente >= 72500000 AND DNI_Cliente <
64              75000000')
65      AS R(IdReclamo INT, DNI_Cliente INT, FechaReclamo DATE,
66          CodLocal VARCHAR, Descripcion VARCHAR, Estado VARCHAR)
67      ) AS temp_reclamo2
68      JOIN (
69          SELECT DNI_Cliente
70          FROM dblink('dbname=db1 user=user1 password=password1 host=
71              localhost port=5433',
72              'SELECT DNI_Cliente FROM venta_2000_2008 WHERE
73                  DNI_Cliente >= 72500000 AND DNI_Cliente <
74                  75000000')
75          AS V(DNI_Cliente INT)
76          UNION ALL
77          SELECT DNI_Cliente
78          FROM dblink('dbname=db2 user=user2 password=password2 host=
79              localhost port=5434',
80              'SELECT DNI_Cliente FROM venta_2009_2016 WHERE
81                  DNI_Cliente >= 72500000 AND DNI_Cliente <
82                  75000000')
83          AS V(DNI_Cliente INT)
84          UNION ALL
85          SELECT DNI_Cliente
86          FROM dblink('dbname=db3 user=user3 password=password3 host=
87              localhost port=5435',
88              'SELECT DNI_Cliente FROM venta_2017_2024 WHERE
89                  DNI_Cliente >= 72500000 AND DNI_Cliente <
90                  75000000')
91          AS V(DNI_Cliente INT)
92      ) AS temp_venta2
93      ON temp_reclamo2.DNI_Cliente = temp_venta2.DNI_Cliente
94      ) AS result2
95
96      UNION ALL
97
98      SELECT *
99      FROM (
100          SELECT *
101          FROM (
102              SELECT R.*

```

```

86      FROM dblink('dbname=db1 user=user1 password=password1 host=
      localhost port=5433',
87      'SELECT * FROM reclamo_loc1_loc4 WHERE DNI_Cliente
      >= 75000000')
88      AS R(IdReclamo INT, DNI_Cliente INT, FechaReclamo DATE,
      CodLocal VARCHAR, Descripcion VARCHAR, Estado VARCHAR)
89      UNION ALL
90      SELECT R.*
91      FROM dblink('dbname=db2 user=user2 password=password2 host=
      localhost port=5434',
92      'SELECT * FROM reclamo_loc5_loc8 WHERE DNI_Cliente
      >= 75000000')
93      AS R(IdReclamo INT, DNI_Cliente INT, FechaReclamo DATE,
      CodLocal VARCHAR, Descripcion VARCHAR, Estado VARCHAR)
94      UNION ALL
95      SELECT R.*
96      FROM dblink('dbname=db3 user=user3 password=password3 host=
      localhost port=5435',
97      'SELECT * FROM reclamo_loc9_loc12 WHERE
      DNI_Cliente >= 75000000')
98      AS R(IdReclamo INT, DNI_Cliente INT, FechaReclamo DATE,
      CodLocal VARCHAR, Descripcion VARCHAR, Estado VARCHAR)
99  ) AS temp_reclamo3
100  JOIN (
101      SELECT DNI_Cliente
102      FROM dblink('dbname=db1 user=user1 password=password1 host=
      localhost port=5433',
103      'SELECT DNI_Cliente FROM venta_2000_2008 WHERE
      DNI_Cliente >= 75000000')
104      AS V(DNI_Cliente INT)
105      UNION ALL
106      SELECT DNI_Cliente
107      FROM dblink('dbname=db2 user=user2 password=password2 host=
      localhost port=5434',
108      'SELECT DNI_Cliente FROM venta_2009_2016 WHERE
      DNI_Cliente >= 75000000')
109      AS V(DNI_Cliente INT)
110      UNION ALL
111      SELECT DNI_Cliente
112      FROM dblink('dbname=db3 user=user3 password=password3 host=
      localhost port=5435',
113      'SELECT DNI_Cliente FROM venta_2017_2024 WHERE
      DNI_Cliente >= 75000000')
114      AS V(DNI_Cliente INT)
115  ) AS temp_venta3
116      ON temp_reclamo3.DNI_Cliente = temp_venta3.DNI_Cliente
117  ) AS result3;

```

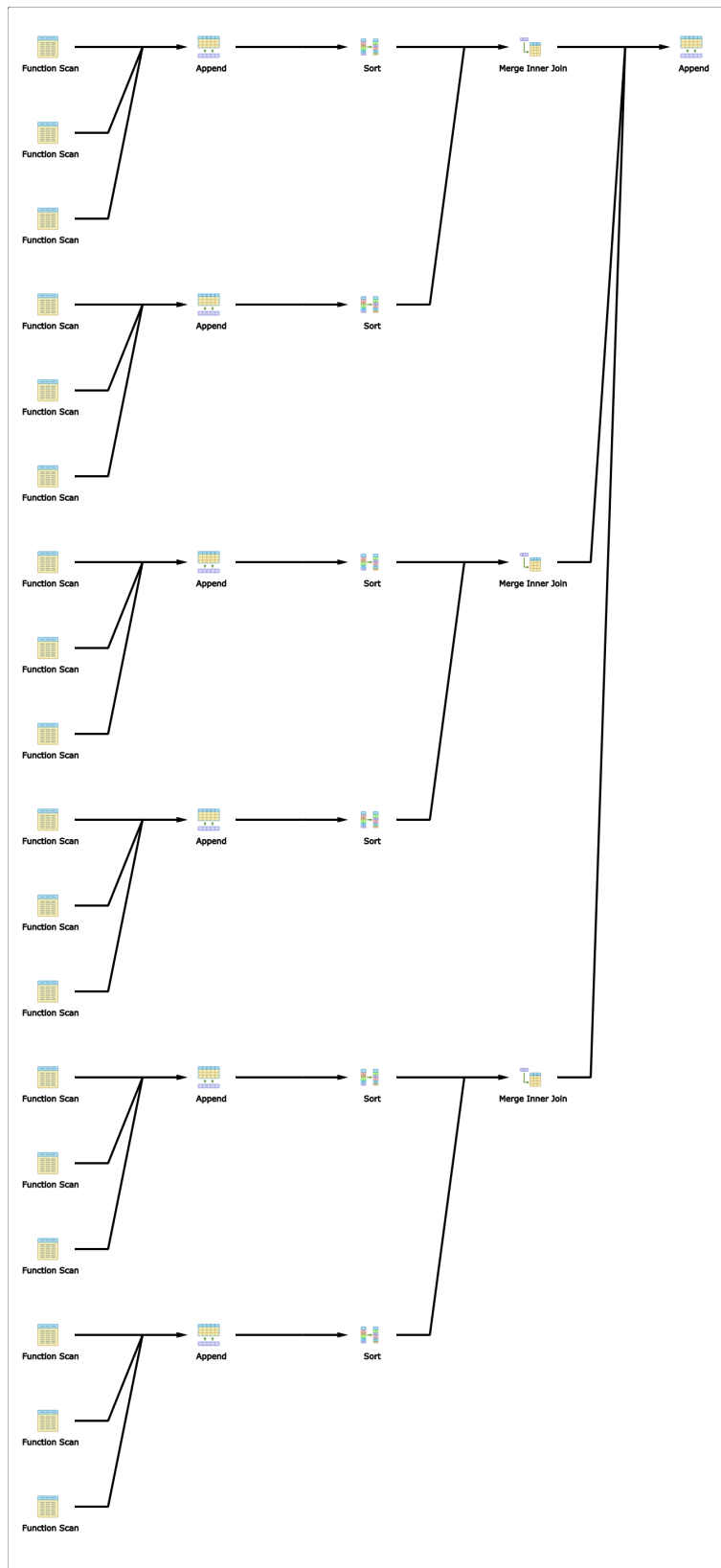


Figure 12: Optimización de la consulta 4 distribuida en diferentes servidores