



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Τμήμα Πληροφορικής



Εργασία Μαθήματος «Συστήματα διαχείρισης βάσεων δεδομένων»

Αριθμός εργασίας	Εργασία 2023-2024
------------------	-------------------



Πίνακας περιεχομένων

Πίνακας περιεχομένων	2
1 Ερώτημα 1 (30%)	4
1.1	5
1.2	6
1.3	7
1.4	8
1.5	9
1.5.1	9
1.5.2	10
2 Ερώτημα 2 (15%)	11
2.1	12
2.2	13
2.3	14
2.4	15
2.5	16
2.5.1	16
2.5.2	17
3 Ερώτημα 3 (15%)	18
3.1	19
3.2	20
3.3	21
3.4	22
3.5	23
3.5.1	23
3.5.2	24
4 Ερώτημα 4 (20%)	25
4.1	25
4.2	26
4.3	27



4.4	28
4.5	29
4.5.1	29
4.5.2	30
5 Ερώτημα 5 (20%)	31
5.1	32
5.2	33
5.3	34
5.4	35
5.5	36
5.5.1	36
5.5.2	37



1 Ερώτημα 1 (30%)

shared_buffers = '128MB'

max_parallel_workers_per_gather = 2

Explain *view_1*:

To *view_1* περιλαμβάνει στήλες από δύο πίνακες, "vessels" και "vesseltypes". Επιλέγει τη στήλη "description" από τον "vesseltypes" και τις στήλες "id" και "type" από τον "vessels". Τα δεδομένα ανακτώνται χρησιμοποιώντας μια LEFT OUTER JOIN, όπου η συνθήκη για την ένωση είναι ότι η στήλη "code" του "vesseltypes" πρέπει να ταιριάζει με τη στήλη "type" του "vessels".



1.1

Explain query:

Επιλέγει την στήλης "t" και χρησιμοποιεί την συνάρτηση date() η οποία αφαιρεί τον τμήμα της ώρας, αυτό μετονομάζεται σε "day". Επιπλέον, μετράει τον αριθμό των θέσεων για κάθε μοναδική ημερομηνία και ονομάζει τον αριθμό αυτό ως "num_positions".

Στη συνέχεια, τα δεδομένα ομαδοποιούνται με βάση τη στήλη "day" και ταξινομούνται με βάση το "num_positions" σε φθίνουσα σειρά.

Explain analyze:

```
--shared_buffers = '128MB'
--max_parallel_workers_per_gather = 2
--(1):
explain analyze
select date(t) as day, count(*) as num_positions
from positions
group by day
order by num_positions desc;
```

The screenshot shows the pgAdmin 4 interface with a query editor and a results pane. The results pane displays the query plan for the provided SQL statement. The plan includes details such as sort operations, group aggregation, and parallel execution statistics. The execution time is shown as 782.909 ms.

Total rows: 21 of 21 Query complete 00:00:00.820 Ln 13, Col 30

Planning Time: 0.094 ms

Execution Time: 782.909 ms

Query complete 00:00:00.820



1.2

Explain query:

Επιλέγει τις στήλες "description" και "type" από το εμφωλευμένο select "foo" και μετρά τον αριθμό των εγγραφών για κάθε ομάδα. Το ίδιο το εμφωλευμένο select ανακτά δεδομένα από τους πίνακες "vessels" και "vesseltypes" με μια LEFT OUTER JOIN.

Η συνθήκη για την ένωση είναι ότι η στήλη "code" στον πίνακα "vesseltypes" πρέπει να ταιριάζει με τη στήλη "type" στον πίνακα "vessels". Στη συνέχεια, τα αποτελέσματα φiltράρονται ώστε να περιλαμβάνουν μόνο τις γραμμές στις οποίες η στήλη "flag" στο "foo" είναι "Greece". Τέλος, τα δεδομένα ομαδοποιούνται με βάση τις στήλες "type" και "description" από το εμφωλευμένο select "foo".

Explain analyze:

The screenshot shows the pgAdmin 4 interface with the following details:

- Properties, SQL, Statistics, Dependencies, Processes, Dashboard** tabs are visible at the top.
- finalproject/postgres@PostgreSQL 15*** connection is selected.
- Query History** tab is active.
- Query** pane contains the SQL code:

```
1 --+-----+  
2 | EPOTHIMA 1 |  
3 +-----+  
4  
5 --shared_buffers = '128MB'  
6 --max_parallel_workers_per_gather = 2  
7  
8 --(if):  
9  
10 explain analyze  
11   select foo.description, foo.type, count()  
12   from (select vesseltypes.description, vessels.id, vessels.type, vessels.flag  
13     from vessels  
14     left outer join vesseltypes on vesseltypes.code = vessels.type  
15   ) as foo  
16  where foo.flag='Greece'  
17  group by foo.type, foo.description;
```
- Data Output** tab is active.
- QUERY PLAN** section displays the execution plan:

```
1 HashAggregate (cost=18.99..21.43 rows=244 width=42) (actual time=0.131..0.135 rows=29 loops=1)  
2   Group Key: vessels.type, vesseltypes.description  
3   Batches: 1 Memory Usage: 37kB  
4     > Hash Left Join (cost=3.39..17.16 rows=244 width=34) (actual time=0.033..0.092 rows=244 loops=1)  
5       Hash Cond: (vessels.type = vesseltypes.code)  
6         > Seq Scan on vessels (cost=0.00..13.11 rows=244 width=4) (actual time=0.010..0.044 rows=244 loops=1)  
7           Filter: ((flag).text = 'Greece'.text)  
8           Rows Removed by Filter: 245  
9         > Hash (cost=2.06..2.06 rows=106 width=34) (actual time=0.020..0.020 rows=106 loops=1)  
10        Buckets: 1024 Batches: 1 Memory Usage: 1kB  
11         > Seq Scan on vesseltypes (cost=0.00..2.06 rows=106 width=34) (actual time=0.004..0.009 rows=106 loops=1)  
12 Planning Time: 0.159 ms  
13 Execution Time: 0.165 ms
```
- Total rows: 13 of 13 Query complete 00:00:00.037** status message at the bottom.

Planning Time: 0.159 ms

Execution Time: 0.165 ms

Query complete 00:00:00.037



1.3

Explain query:

Επιλέγει τις στήλες "description" και "type" από την "view_1". Επιπλέων μετράει τον αριθμού των εγγραφών για κάθε μοναδική τιμή του "description". Η συνθήκη σύνδεσης είναι ότι η στήλη "vessel_id" στον πίνακα "positions" πρέπει να ταιριάζει με τη στήλη "id" στον πίνακα "view_1".

Τα αποτελέσματα φιλτράρονται για να συμπεριλάβουν μόνο τις γραμμές στις οποίες η στήλη "speed" στον πίνακα "positions" είναι μεγαλύτερη από 30. Στη συνέχεια, τα δεδομένα ομαδοποιούνται με βάση τις στήλες "type" και "description" από την "view_1".

Explain analyze:

```

Properties SQL Statistics Dependencies Processes Dashboard finalproject/postgres@PostgreSQL 15*
finalproject/postgres@PostgreSQL 15
No limit
Query History
Query Data Output Messages Notifications
QUERY PLAN
text
1 Finalize GroupAggregate (cost=170211.03..170731.08 rows=2013 width=42) (actual time=516.968..542.913 rows=10 loops=1)
2   Group Key: vessel.type, vesseltypes.description
3     -> Gather Merge (cost=170211.03..170680.76 rows=4026 width=42) (actual time=516.963..542.905 rows=23 loops=1)
4       Workers Planned: 2
5       Workers Launched: 2
6         -> Sort (cost=169211.00..169211.63 rows=2013 width=42) (actual time=494.779..494.781 rows=8 loops=3)
7           Sort Key: vessel.type, vesseltypes.description
8           Sort Method: quicksort, Memory: 25kB
9             Worker 0: Sort Method: quicksort, Memory: 25kB
10            Worker 1: Sort Method: quicksort, Memory: 25kB
11             -> Partial HashAggregate (cost=169080.41..169100.54 rows=2013 width=42) (actual time=494.736..494.747 rows=8 loops=3)
12               Group Key: vessel.type, vesseltypes.description
13               Batches: 1 Memory Usage: 121kB
14                 Worker 0: Batches: 1 Memory Usage: 121kB
15                 Worker 1: Batches: 1 Memory Usage: 121kB
16                   -> Hash Left Join (cost=21.39..168401.57 rows=90511 width=34) (actual time=1.324..480.787 rows=69914 loops=3)
17                     Hash Cond: (vessel.type = vesseltypes.description)
18                     -> Hash Join (cost=18.00..168151.88 rows=90511 width=4) (actual time=0.945..468.850 rows=69914 loops=3)
19                       Hash Cond: ((positions.vessel_id).text = (vessel.id).text)
20                         -> Parallel Seq Scan on positions (cost=0.00..167894.22 rows=90511 width=65) (actual time=0.535..453.853 rows=69914 loops=3)
21                           Filter: (speed > 30 double precision)
22                           Rows Removed by Filter: 2275637
23                         -> Hash (cost=11.89..11.89 rows=489 width=69) (actual time=0.393..0.393 rows=489 loops=3)
24                           Buckets: 1024 Batches: 1 Memory Usage: 57kB
25                             -> Seq Scan on vessels (cost=0.00..11.89 rows=489 width=69) (actual time=0.270..0.315 rows=489 loops=3)
26                             -> Hash (cost=2.06..2.06 rows=106 width=34) (actual time=0.363..0.363 rows=106 loops=3)
27                               Buckets: 1024 Batches: 1 Memory Usage: 19kB
28                                 -> Seq Scan on vesseltypes (cost=0.00..2.06 rows=106 width=34) (actual time=0.327..0.336 rows=106 loops=3)
29 Planning Time: 0.351 ms
30 Execution Time: 543.004 ms

```

Total rows: 30 of 30 Query complete 00:00:00.587 Ln 15, Col 43

Planning Time: 0.351 ms

Execution Time: 543.004 ms

Query complete 00:00:00.587



1.4

Explain query:

Επιλέγει την στήλης "t" και χρησιμοποιεί την συνάρτηση date() η οποία αφαιρεί τον τμήμα της ώρας, αυτό μετονομάζεται σε "day". Επιπλέον, μετράει τον αριθμό των εγγραφών για κάθε μοναδική ημερομηνία.

Η συνθήκη σύνδεσης είναι ότι η στήλη "vessel_id" στον πίνακα "positions" πρέπει να ταιριάζει με τη στήλη "id" στον πίνακα "view_1". Στη συνέχεια, τα αποτελέσματα φιλτράρονται για να συμπεριλάβουν μόνο τις γραμμές στις οποίες η στήλη "description" στον πίνακα "view_1" περιέχει την λέξη "Passenger" και η στήλη "t" στον πίνακα "positions" να ανήκει στο εύρος ημερομηνιών ("2019-08-14 00:00:00" έως "2019-08-18 23:59:59").

Τέλος, τα δεδομένα ομαδοποιούνται με βάση τη στήλη "day" και ταξινομούνται σε αύξουσα σειρά κατά "day".

Explain analyze:

Properties SQL Statistics Dependencies Processes Dashboard [finalproject/postgres@localhost:5432](#)

Final project/postgres@PostgreSQL 15

Query History

Query

```
1 --+-----+ EPITHIMA_1 +-----+
2
3
4
5 --shared_buffers = '128MB'
6 --max_parallel_workers_per_gather = 2
7
8 --(t);
9
10 explain analyze
11 select date(t) as day, count(*)
12   from view_1
13   join positions on positions.vessel_id = view_1.id
14 where view_1.description like '%Passenger%'
15   and positions.t between '2010-08-14 00:00:00' and '2010-08-18 23:59:59'
16 group by day
17 order by day;
```

Data Output Messages Notifications

QUERY PLAN

text

```
1 Finalize GroupAggregate (cost=180852.46..190129.11 rows=76548 width=12) (actual time=407.365..307.652 rows=5 loops=1)
  2   Group Key {date(positions.t)}
  3     Gather Merge (cost=180852.46..180853.31 rows=63795 width=12) (actual time=405.547..507.649 rows=15 loops=1)
  4       Workers Planned: 2
  5       Workers Launched: 2
  6         Partial GroupAggregate (cost=179952.43..180490.33 rows=31895 width=12) (actual time=449.854..456.553 rows=5 loops=3)
  7           Group Key {date(positions.t)}
  8             Sort (cost=179952.43..179952.77 rows=31895 width=4) (actual time=449.182..453.082 rows=12048 loops=3)
  9               Sort Key {date(positions.t)}
10               Sort Method: quicksort Memory: 307368
11               Worker 0: Sort Method: quicksort Memory: 307368
12               Worker 1: Sort Method: quicksort Memory: 307368
13             Hash Join (cost=18.63..177466.32 rows=31895 width=4) (actual time=76.060..443.573 rows=312498 loops=3)
14               Hash Cond {positions.vessel_id = <vessel_id>.id}
15               Parallel Seq Scan on positions (cost=0.00..17524.07 rows=487396 width=4) (actual time=75.159..392.188 rows=312404 loops=3)
16                 Filter {((t >= '2010-08-14 00:00:00'::timestamp without time zone) AND ((t <= '2010-08-18 23:59:59'::timestamp without time zone))}
17                 Rows Removed by Filter: 955346
18               Hash (cost=45.83..15.50 rows=47 width=4) (actual time=4.722..2.724 rows=64 loops=3)
19               Backups: 1024 Batches: 1 Memory Usage: 156B
20               Hash Join (cost=24.15..15.63 rows=32 width=4) (actual time=4.638..8.708 rows=64 loops=3)
21                 Hash Cond {<vessel_id>.type = vessels.type}
22                 Seq Scan on vessels (cost=0.00..1.09 rows=489 width=4) (actual time=0.368..0.402 rows=489 loops=3)
23                 Hash (cost=2.33..2.33 rows=7 width=4) (actual time=0.228..0.237 rows=10 loops=3)
24                   Hash (cost=2.33..2.33 rows=7 width=4) (actual time=0.224..0.228 rows=10 loops=3)
25                     Filter {description ~~ '%Passenger%'::text}
26                     Seq Scan on vessels (cost=0.00..2.33 rows=7 width=4) (actual time=0.224..0.228 rows=10 loops=3)
27                     Rows Removed by Filter: 96
28 Planning Time: 0.336 ms
29 Execution Time: 507.904 ms
```

Total rows: 28 of 29 Query complete 00:00:00.360

Line 17, Col 1

Planning Time: 0.336 ms

Execution Time: 507.904 ms

Query complete 00:00:00.560



1.5

1.5.1

Explain query:

Επιλέγει τη στήλη "id" από τον πίνακα "view_1", τη στήλη "speed" από τον πίνακα "positions" και τη συνιστώσα ημερομηνίας την στήλης "t" η οποία χρησιμοποιεί την συνάρτηση date() όπου αφαιρεί τον τμήμα της ώρας, αυτό μετονομάζεται σε "day". Η συνθήκη σύνδεσης είναι ότι η στήλη "vessel_id" στον πίνακα "positions" πρέπει να ταιριάζει με τη στήλη "id" στον πίνακα "view_1".

Τα αποτελέσματα φιλτράρονται ώστε να περιλαμβάνουν μόνο γραμμές στις οποίες η στήλη "description" στον πίνακα "view_1" περιέχει την λέξη "Cargo", η στήλη "t" στον πίνακα "positions" να ανήκει στο εύρος ημερομηνιών ("2019-08-15 00:00:00" έως "2019-08-18 23:59:59") και η στήλη "speed" στον πίνακα "positions" είναι ίση με 0.

Στη συνέχεια, τα δεδομένα ομαδοποιούνται με βάση τις στήλες "day", "id" και "speed".

Explain analyze:

```

Properties SQL Statistics Dependencies Processes Dashboard finalproject/postgres@PostgreSQL 15*
finalproject/postgres@PostgreSQL 15
No limit
Query History
Data Output Messages Notifications
QUERY PLAN Text
1 Group (cost=18910.65..100311.91 rows=40139 width=77) (actual time=479.368..501.417 rows=47 loops=1)
  2 Group Key (date(positions.t), vessel_id, positions.speed)
  3 > Gather Merge (cost=18910.66..18990.69 rows=23465 width=77) (actual time=479.369..501.293 rows=196 loops=1)
    4   Workers Planned: 2
    5     Workers Launched: 2
    6       > Group (cost=18910.64..189119.70 rows=16/25 width=77) (actual time=440.473..442.017 rows=65 loops=1)
    7         Group Key (date(positions.t), vessel_id, positions.speed)
    8           > Sort (cost=18410.64..18492.45 rows=16/25 width=77) (actual time=440.469..440.698 rows=7/27 loops=3)
    9             Sort Key (date(positions.t), vessel_id)
    10            Sort Method: quicksort Memory: 11728
    11              Worker 0: Sort Method: quicksort Memory: 11458
    12              Worker 1: Sort Method: quicksort Memory: 11968
    13                > Hash Join (cost=16.03..18277.41 rows=16/25 width=77) (actual time=113.113..495.125 rows=7/27 loops=3)
    14                  Hash Cond: (positions.vessel_id = text.(vessel_id).text)
    15                    > Parallel Seq Scan on positions (cost=0.0..18253.91 rows=255573 width=81) (actual time=88.836..416.194 rows=197248 loops=3)
    16                      Filter: ((t > '2019-08-15 00:00:00'::timestamp without time zone) AND (t <= '2019-08-18 23:59:59'::timestamp without time zone) AND (speed = 0)::double precision)
    17                      Rows Removed by Filter: 214832
    18                      > Hash (cost=15.65..15.63 rows=32 width=65) (actual time=0.657..0.660 rows=104 loops=3)
    19                        Buckets: 1024 Batches: 1 Memory Usage: 18kB
    20                          > Hash Join (cost=2.41..15.63 rows=32 width=65) (actual time=0.572..0.638 rows=104 loops=3)
    21                            Hash Cond: (vessel_type = vesseltypes.code)
    22                              > Seq Scan on vessels (cost=0.0..0.118 rows=489 width=69) (actual time=0.334..0.365 rows=489 loops=3)
    23                              > Hash (cost=2.33..2.33 rows=6 width=6) (actual time=0.210..0.211 rows=10 loops=3)
    24                                Buckets: 1024 Batches: 1 Memory Usage: 9kB
    25                                  > Seq Scan on vesseltypes (cost=0.0..0.233 rows=7 width=4) (actual time=0.200..0.203 rows=10 loops=3)
    26                                  Filter: (description ~~ '%Cargo%'::text)
    27                                  Rows Removed by Filter: 96
    28 Planning Time: 0.354 ms
    29 Execution Time: 501.607 ms
  
```

Total rows: 29 of 29 Query complete 00:00:00.533 Ln 14, Col 3

Planning Time: 0.354 ms

Execution Time: 501.607 ms

Query complete 00:00:00.533



1.5.2

Explain query:

Επιλέγει τη στήλη "id" από τον πίνακα "view_1" και μετρά το άθροισμα όπου η ταχύτητα είναι 0 για κάθε "id". Η καταμέτρηση επιτυγχάνεται με τη χρήση μιας δήλωσης υπό όρους εντός της συνάρτησης count. Η συνθήκη είναι ότι εάν το "speed" στον πίνακα "positions" είναι 0, τότε την μετράει ως 1, διαφορετικά την μετράει ως NULL.

Τα αποτελέσματα φίλτραρονται ώστε να περιλαμβάνουν μόνο γραμμές στις οποίες η στήλη "description" στο "view_1" περιέχει την λέξη "Cargo" και η στήλη "t" στο "positions" να ανήκει στο εύρος ημερομηνιών ("2019-08-12 00:00:00" έως "2019-08-19 23:59:59").

Στη συνέχεια, τα δεδομένα ομαδοποιούνται με βάση τη στήλη "id" από το "view_1". Το clause having διασφαλίζει ότι περιλαμβάνονται μόνο οι εγγραφές στις οποίες ο αριθμός των θέσεων με ταχύτητα 0 είναι ίσος με τον αριθμό όλων των στιγμιότυπων για το συγκεκριμένο σκάφος. Αυτό καθορίζεται από την συνθήκη count(case when positions.speed = 0 then 1 else NULL end) = count(view_1.id).

Τέλος, τα αποτελέσματα διατάσσονται με βάση τη στήλη "id" από το "view_1".

Explain analyze:

```

Properties SQL Statistics Dependencies Processes Dashboard finalproject/postgres@PostgreSQL_15
File Edit Object Tools Help
finalproject/postgres@PostgreSQL_15
Query History
Query Analyze
Data Output Messages Notifications
QUERY PLAN
Finalize GroupAggregate (cost=176911.00..179441.99 rows=2 width=72) (actual time=478,501.496..240.240 rows=1 loops=1)
  Group Key: vessel_id
  Finalize GroupAggregate (cost=176911.00..179441.99 rows=2 width=72) (actual time=478,501.496..240.240 rows=1 loops=1)
    Group Key: vessel_id
    Filter: (positions.speed = 0) double precision < 0.01
    Rows Removed by Filter: 0
    →  Gather Merge (cost=176911.00..179441.99 rows=978 width=72) (actual time=478,424.416..214.214 rows=175 loops=1)
      Workers Planned: 2
      Workers Launched: 2
      + Sort (cost=78517.19..178519.22 rows=489 width=61) (actual time=422,422.422..427.427 rows=92 loops=1)
        Sort Key: vessel_id
        Sort Method: quicksort Memory: 31MB
        Workers: 0 (Sort Method: quicksort Memory: 31MB)
        Workers: 1 (Sort Method: quicksort Memory: 31MB)
        + Parallel HashAggregate (cost=176911.00..178519.22 rows=489 width=61) (actual time=472,214.422..280.280 rows=175 loops=1)
          Group Key: vessel_id
          Batches: 1 Memory Usage: 5 MB
          Worker 0: Batches: 1 Memory Usage: 5 MB
          Worker 1: Batches: 1 Memory Usage: 5 MB
          + Hash Join (cost=0.00..176710.68 rows=42050 width=72) (actual time=47,581.415..212 rows=401553 loops=1)
            Hash Cond: (positions.id = view_1.id)
            Hash Cond: (view_1.description like '%Cargo%')
            + Parallel Seq Scan on positions (cost=0.00..175271.07 rows=4457973 width=72) (actual time=37,770..365.115 rows=430768 loops=1)
              Filter: (t >= '2019-08-12 00:00:00'::timestamp without time zone) AND (t <= '2019-08-19 23:59:59'::timestamp without time zone)
              Rows Removed by Filter: 181407
            + Index Scan on view_1 (cost=0.00..15,530.00 rows=1 width=30) (actual time=1,021..1,023 rows=104 loops=1)
              Index Cond: (id = positions.id)
              + Hash Join (cost=24.41..15,630 rows=32 width=65) (actual time=920..3,998 rows=104 loops=1)
                Hash Cond: (view_1.type = vessel_type)
                + Seq Scan on vessel (cost=0.00..11,49 rows=489 width=65) (actual time=0..14..0.029 rows=489 loops=1)
                + Hash (cost=3.38 rows=1,030 width=4) (actual time=0..62..0.30 rows=104 loops=1)
                  Buckets: 128 Batches: 1 Memory Usage: 9 MB
                  + Seq Scan on vessel (cost=0.00..1,030 rows=1,030 width=4) (actual time=0..298..0.254 rows=104 loops=1)
                    Filter: (vessel_type = 'Automobile')
                    Rows Removed by Filter: 16
Planning Time: 0.391 ms
Execution Time: 496.377 ms

```

Planning Time: 0.391 ms

Execution Time: 496.327 ms

Query complete 00:00:00.528



2 Ερώτημα 2 (15%)

The screenshot shows a PostgreSQL terminal window. The left pane displays the following SQL commands:

```
1 --+-----+
2 |   ΕΡΩΤΗΜΑ 2   |
3 +-----+
4
5 --shared_buffers = '8GB'
6 --max_parallel_workers_per_gather = 2
7
8 alter system set shared_buffers to '8GB'
9 show shared_buffers
```

The right pane shows the results of the 'show shared_buffers' command, which returns a single row:

	shared_buffers
1	8GB

shared_buffers = '8GB'

max_parallel_workers_per_gather = 2

Παρατήρηση:

Η παράμετρος shared_buffers καθορίζει το μέγεθος της μνήμης που θα χρησιμοποιείται από την PostgreSQL για την προσωρινή αποθήκευση δεδομένων στην κοινή μνήμη. Υπάρχει ένα σημείο φθίνουσας απόδοσης όταν πρόκειται για αύξηση αυτής της παραμέτρου. Καθώς αυξάνεται το μέγεθος, οι σταδιακές βελτιώσεις της απόδοσης μπορεί να μειωθούν.

Συνεπώς με την αλλαγή των shared_buffers από 128MB σε 8GB παρατηρήσαμε ότι το Planning Time μειώνεται στις περισσότερες περιπτώσεις. Σε ορισμένες περίπτωσης όμως το Planning Time μειώνεται ελάχιστα λόγο της μεγάλης αύξησης της παραμέτρου. Ακόμα παρατηρήσαμε μείωση του Execution Time στις περιπτώσεις.

Σημείωση:

Κάθε query έχει εκτελεστεί πάνω από δύο φορές, ώστε να προλάβουν τα buffers να αρχικοποιηθούν.



2.1

The screenshot shows the pgAdmin 4 interface with a query editor and a results pane. The query is:

```
1 -- | EPOTRMA 2 |
2 -- |
3 --
4
5 --shared_buffers = '8GB'
6 --max_parallel_workers_per_gather = 2
7 --();
8
9 explain analyze
10 select date(t) as day, count(*) as num_positions
11   from positions
12  group by day
13  order by num_positions desc;
```

The results pane displays the query plan (text format) and execution statistics:

Step	Operation	Cost	Time
1	Sort	(cost=582487.73..583638.07)	(actual time=694.969..734.327)
2	Sort Key: (count(*)) DESC		
3	Sort Method: quicksort Memory: 256kB		
4	> Finalize GroupAggregate (cost=413617.54..531343.42)	(actual time=694.936..734.320)	(rows=24 loops=1)
5	Group Key: (date(t))		
6	> Gather Merge (cost=413617.52..520990.34)	(actual time=694.930..734.308)	(rows=72 loops=1)
7	Workers Planned: 2		
8	Workers Launched: 2		
9	> Sort (cost=412617.52..413767.86)	(actual time=672.632..672.633)	(rows=24 loops=1)
10	Sort Key: (date())		
11	Sort Method: quicksort Memory: 20kB		
12	Worker 0: Sort Method: quicksort Memory: 264kB		
13	Worker 1: Sort Method: quicksort Memory: 264kB		
14	> Partial HashAggregate (cost=332815.74..361473.21)	(actual time=672.298..672.607)	(rows=24 loops=3)
15	Group Key: (date())		
16	Planned Partitions: 8 Batches: 1 Memory Usage: 3097kB		
17	Worker 0: Batches: 1 Memory Usage: 3097kB		
18	Worker 1: Batches: 1 Memory Usage: 3097kB		
19	> Parallel Seq Scan on positions (cost=0.00..167894.22)	(actual time=0.011..409.411)	(rows=2345550 loops=3)
20	Planning Time: 0.075 ms		
21	Execution Time: 734.891 ms		

Total rows: 21 of 21 | Query complete 00:00:00.774 | Ln 13, Col 30

Planning Time: 0.075 ms

Execution Time: 734.891ms

Query complete 00:00:00.774



2.2

The screenshot shows the pgAdmin 4 interface with a query editor and a results pane. The query is:

```
1 -- EPOTHMA 2
2 --shared_buffers = '1GB'
3 --max_parallel_workers_per_gather = 2
4 --(ii):
5
6 explain analyze
7
8
9
10    select foo.description, foo.type, count()
11   from (select vesseltypes.description, vessels.id, vesseltypes.type, vessels.flag
12          from vessels
13         left outer join vesseltypes on vesseltypes.code = vessels.type
14       ) as foo
15  where foo.flag='Greece'
16 group by foo.type, foo.description;
```

The results pane shows the query plan:

QUERY PLAN
text
1 HashAggregate (cost=18.99..21.43 rows=244 width=42) (actual time=0.145..0.149 rows=29 loops=1)
2 Group Key: vessel.type, vesseltypes.description
3 Batches: 1 Memory Usage: 37kB
4 > Hash Left Join (cost=3.39..17.16 rows=244 width=34) (actual time=0.034..0.098 rows=244 loops=1)
5 Hash Cond: (vessel.type = vesseltypes.code)
6 > Seq Scan on vessel (cost=0.00..13.11 rows=244 width=4) (actual time=0.010..0.046 rows=244 loops=1)
7 Filter: ((flag).text = 'Greece'.text)
8 Rows Removed by Filter: 245
9 > Hash (cost=2.06..2.06 rows=106 width=34) (actual time=0.021..0.021 rows=106 loops=1)
10 Buckets: 1024 Batches: 1 Memory Usage: 15kB
11 > Seq Scan on vesseltypes (cost=0.00..2.06 rows=106 width=34) (actual time=0.003..0.009 rows=106 loops=1)
12 Planning Time: 0.135 ms
13 Execution Time: 0.171 ms

Total rows: 13 of 13 | Query complete 00:00:00.072 | Ln 16, Col 37

Planning Time: 0.135 ms

Execution Time: 0.171 ms

Query complete 00:00:00.072



2.3

The screenshot shows the pgAdmin 4 interface with a query editor and a results pane. The query is an EXPLAIN ANALYZE command for a specific SQL query. The results pane displays the query plan, which includes details about the execution plan, cost, and actual time for each step. The plan shows parallel execution with two workers, including hash joins, gather merge, and partial aggregate operations. The total planning time is 0.328 ms and the execution time is 431.532 ms.

```
1 -- shared_buffers = '1GB'
2 -- max_parallel_workers_per_gather = 2
3 -- (ffff);
4
5 explain analyze
6 select view_1.description, view_1.type, count(*)
7   from view_1
8   join positions on positions.vessel_id = view_1.id
9   where speed > 30
10  group by view_1.type, view_1.description;
```

Planning Time: 0.328 ms

Execution Time: 431.532 ms

Query complete 00:00:00.463



2.4

Properties SQL Statistics Dependencies Processes Dashboard finalproject/postgres@PostgreSQL_15*

Query History

```

1 -- | EPOTHMA 2 |
2 -- |
3 --
4
5 --shared_buffers = '1GB'
6 --max_parallel_workers_per_gather = 2
7 --(iv):
8
9 explain analyze
10 select date(t) as day, count(*)
11   from view_1
12   join positions on positions.vessel_id = view_1.id
13  where view_1.description like '%Passenger%' and
14    positions.t between '2019-08-14 00:00:00' and '2019-08-18 23:59:59'
15 group by day
16 order by day;

```

Data Output Messages Notifications

QUERY PLAN text

1 Finalize GroupAggregate (cost=180852.46..190129.11 rows=76548 width=12) (actual time=371.148..405.419 rows=5 loops=1)
 2. Group Key {date(positions.t)}
 3. -> Gather Merge (cost=180852.46..188853.31 rows=63790 width=12) (actual time=368.578..405.405 rows=15 loops=1)
 4. Workers Planned: 2
 5. Workers Launched: 2
 6. -> Partial GroupAggregate (cost=179852.43..180490.33 rows=31895 width=12) (actual time=339.111..346.893 rows=5 loops=3)
 7. Group Key {date(positions.t)}
 8. -> Sort (cost=179852.43..179932.17 rows=31895 width=4) (actual time=338.624..341.558 rows=120498 loops=3)
 9. Sort Key {date(positions.t)}
 10. Sort Method: quicksort Memory: 4096kB
 11. Worker 0: Sort Method: quicksort Memory: 3073kB
 12. Worker 1: Sort Method: quicksort Memory: 3073kB
 13. -> Hash Join (cost=11.03..177464.52 rows=31895 width=4) (actual time=43.026..332.783 rows=120498 loops=3)
 14. Hash Cond: ((positions.vessel_id).text = (vessels.id).text)
 15. -> Parallel Seq Scan on positions (cost=0.00..175224.07 rows=487396 width=73) (actual time=41.983..276.026 rows=392404 loops=3)
 16. Filter: ((t >= '2019-08-14 00:00:00'::timestamp without time zone) AND (t <= '2019-08-18 23:59:59'::timestamp without time zone))
 17. Rows Removed by Filter: 1953146
 18. -> Hash (cost=15.63..15.63 rows=32 width=65) (actual time=0.948..0.950 rows=64 loops=3)
 19. Buckets: 1024 Batches: 1 Memory Usage: 15kB
 20. -> Hash Join (cost=2.41..15.63 rows=32 width=55) (actual time=0.874..0.937 rows=64 loops=3)
 21. Hash Cond: (vessels.type ~ vesseltypes.code)
 22. -> Seq Scan on vessels (cost=0.00..11.89 rows=489 width=69) (actual time=0.517..0.551 rows=489 loops=3)
 23. -> Hash (cost=2.33..2.33 rows=7 width=4) (actual time=0.324..0.325 rows=10 loops=3)
 24. Buckets: 1024 Batches: 1 Memory Usage: 9kB
 25. -> Seq Scan on vesseltypes (cost=0.00..0.23 rows=7 width=4) (actual time=0.312..0.317 rows=10 loops=3)
 26. Filter: (description ~~ '%Passenger%'.text)
 27. Rows Removed by Filter: 96
 28 Planning Time: 0.309 ms
 29 Execution Time: 405.598 ms

Total rows: 29 of 29 | Query complete 00:00:00.434 | Ln 14, Col 3

Planning Time: 0.309 ms

Execution Time: 405.598 ms

Query complete 00:00:00.434



2.5

2.5.1

Properties SQL Statistics Dependencies Processes Dashboard finalproject/postgres@PostgreSQL 15*

finalproject/postgres@PostgreSQL 15 No limit

Query History Data Output Messages Notifications

```

1 --+-----+
2 ---| EPOTHIMA_2 |
3 ---+-----+
4
5 --shared_buffers = '8GB'
6 --max_parallel_workers_per_gather = 2
7 --(v)
8 --a meros:
9
10 explain analyze
11 select view_1.id, positions.speed, date(t) as day
12 from view_1
13 join positions on positions.vessel_id = view_1.id
14 where view_1.description like '%Cargo%' and positions.t
15 between '2019-08-15 00:00:00' and '2019-08-18 23:59:59' and positions.speed = 0
16 group by day, view_1.id ,positions.speed;
17
18
19
20
21
22
23
24
25
26
27
28
29

```

QUERY PLAN

text

1 Group (cost=185910.66..190331.91 rows=40139 width=77) (actual time=361.372..399.083 rows=67 loops=1)

2 Group Key (date(positions.t), vessels.id, positions.speed)

3 -> Gather Merge (cost=185910.66..189980.69 rows=33450 width=77) (actual time=361.371..399.036 rows=196 loops=1)

4 Workers Planned: 2

5 Workers Launched: 2

6 -> Group (cost=184910.64..185119.70 rows=16725 width=77) (actual time=339.369..340.524 rows=65 loops=3)

7 Group Key (date(positions.t), vessels.id, positions.speed)

8 -> Sort (cost=164910.64..184952.45 rows=16725 width=77) (actual time=239.366..239.683 rows=7727 loops=3)

9 Sort Key (date(positions.t), vessels.id)

10 Sort Method: quicksort Memory: 1558kB

11 Worker 0: Sort Method: quicksort Memory: 1072kB

12 Worker 1: Sort Method: quicksort Memory: 1040kB

13 -> Hash Join (cost=16.03..183737.41 rows=16725 width=77) (actual time=73.220..334.673 rows=7727 loops=3)

14 Hash Cond: (positions.vessel_id = vessels.id).text

15 -> Parallel Seq Scan on positions (cost=0.00..182553.91 rows=255573 width=81) (actual time=52.286..314.012 rows=197248 loops=3)

16 Filter: (t >= '2019-08-15 00:00:00'::timestamp without time zone) AND (t <= '2019-08-18 23:59:59'::timestamp without time zone) AND (speed = '0':double prec..)

17 Rows Removed by Filter: 2148302

18 -> Hash (cost=15.63..15.63 rows=32 width=65) (actual time=0.878..0.880 rows=104 loops=3)

19 Buckets: 1024 Batches: 1 Memory Usage: 18kB

20 -> Hash Join (cost=2.41..15.63 rows=32 width=65) (actual time=0.788..0.857 rows=104 loops=3)

21 Hash Cond: (vessel_type = vesseltypes.code)

22 -> Seq Scan on vessels (cost=0.00..11.89 rows=489 width=69) (actual time=0.384..0.419 rows=489 loops=3)

23 -> Hash (cost=1.23..2.33 rows=7 width=4) (actual time=0.359..0.360 rows=10 loops=3)

24 Buckets: 1024 Batches: 1 Memory Usage: 9kB

25 -> Seq Scan on vesseltypes (cost=0.00..2.33 rows=7 width=4) (actual time=0.350..0.354 rows=10 loops=3)

26 Filter: (description ~~ '%Cargo%'.text)

27 Rows Removed by Filter: 96

28 Planning Time: 0.344 ms

29 Execution Time: 399.260 ms

Total rows: 29 of 29 Query complete 00:00:00.429

Ln 15, Col 3

Planning Time: 0.344 ms

Execution Time: 399.260 ms

Query complete 00:00:00.429



2.5.2

Properties SQL Statistics Dependencies Processes Dashboard finalproject/postgres@PostgreSQL 15*

Query History

```

1 --          +-----+
2 --          | EPOTHIMA 2 |
3 --          +-----+
4
5 --shared_buffers = '8GB'
6 --max_parallel_workers_per_gather = 2
7 --(V)
8 --b meros:
9
10 explain analyze
11 select view_1.id, count(case when positions.speed = 0 then 1 else NULL end) as count_zero_speed
12 from view_1
13 join positions on positions.vessel_id = view_1.id
14 where view_1.description like '%Cargo%' and
15       positions.t between '2019-08-12 00:00:00' and '2019-08-19 23:59:59'
16 group by view_1.id
17 having count(case when positions.speed = 0 then 1 else NULL end) = count(view_1.id)
18 order by view_1.id;

```

Data Output Messages Notifications

QUERY PLAN

```

1 Finalize GroupAggregate (cost=179518.00..179645.56 rows=2 width=73) (actual time=370.865..416.226 rows=1 loops=1)
2   Group Key: vessel.id
3   Filter: (count(CASE WHEN (positions.speed = 0::double precision) THEN 1 ELSE NULL::integer END) = count(vessel.id))
4   Rows Removed by Filter: 52
5   > Gather Merge (cost=179518.00..179632.11 rows=478 width=81) (actual time=370.774..416.185 rows=155 loops=1)
6     Workers Planned: 2
7     Workers Launched: 2
8     > Sort (cost=178517.98..178519.20 rows=489 width=81) (actual time=347.135..347.140 rows=52 loops=3)
9       Sort Key: vessel.id
10      Sort Method: quicksort Memory: 31kB
11      Worker 0: Sort Method: quicksort Memory: 31kB
12      Worker 1: Sort Method: quicksort Memory: 31kB
13      > Partial HashAggregate (cost=178491.24..178496.13 rows=489 width=81) (actual time=346.995..347.008 rows=52 loops=3)
14        Group Key: vessel.id
15        Batches: 1 Memory Usage: 57kB
16        Worker 0: Batches: 1 Memory Usage: 57kB
17        Worker 1: Batches: 1 Memory Usage: 57kB
18        > Hash Join (cost=16.03..178070.68 rows=42056 width=73) (actual time=45.783..339.497 rows=31533 loops=3)
19          Hash Cond: (positions.vessel_id = (vessel.id).text)
20          > Parallel Seq Scan on positions (cost=0.00..175224.07 rows=642673 width=73) (actual time=41.013..281.410 rows=530768...)
21          Filter: ((t ~> 2019-08-12 00:00:00 timestamp without time zone) AND (t <= 2019-08-19 23:59:59 timestamp without time zone))
22          Rows Removed by Filter: 1814783
23          > Hash (cost=15.63..15.63 rows=32 width=65) (actual time=0.824..0.826 rows=104 loops=3)
24            Buckets: 1024 Batches: 1 Memory Usage: 18kB
25            > Hash Join (cost=2.41..41.15.63 rows=32 width=65) (actual time=0.736..0.806 rows=104 loops=3)
26              Hash Cond: (vessel.type = vesseltypes.code)
27              > Seq Scan on vessels (cost=0.00..11.89 rows=489 width=69) (actual time=0.401..0.435 rows=489 loops=3)
28              > Hash (cost=2.33..2.33 rows=7 width=4) (actual time=0.293..0.293 rows=10 loops=3)
29                Buckets: 1024 Batches: 1 Memory Usage: 9kB
30                > Seq Scan on vesseltypes (cost=0.00..2.33 rows=7 width=4) (actual time=0.288..0.288 rows=10 loops=3)
31                  Filter: (description ~~ '%Cargo%::text')
32                  Rows Removed by Filter: 96
33 Planning Time: 0.334 ms
34 Execution Time: 416.295 ms

```

Total rows: 34 of 34 Query complete 00:00:00.445 Ln 8, Col 11

Planning Time: 0.334 ms

Execution Time: 416.295 ms

Query complete 00:00:00.445



3 Ερώτημα 3 (15%)

The screenshot shows a database query interface with two panes. The left pane, titled 'Query History', contains the following SQL code:

```
1 --  
2 -- ΕΡΩΤΗΜΑ 3  
3 --  
4  
5 --shared_buffers = '8GB'  
6 --max_parallel_workers_per_gather = 4  
7  
8 set max_parallel_workers_per_gather=4;  
9 show max_parallel_workers_per_gather
```

The right pane, titled 'Data Output', shows the result of the 'show' command:

	max_parallel_workers_per_gather
1	4

shared_buffers = '8GB'

max_parallel_workers_per_gather = 4

Παρατήρηση:

Μια υψηλότερη τιμή για την παράμετρο max_parallel_workers_per_gather επιτρέπει σε περισσότερους parallel workers (παράλληλους εργάτες) να συμμετέχουν στην εκτέλεση ενός μόνο κόμβου gather κατά την εκτέλεση ερωτημάτων.

Ο παραλληλισμός μπορεί να οδηγήσει σε ταχύτερους χρόνους εκτέλεσης ερωτημάτων, ιδίως για ερωτήματα που μπορούν να επωφεληθούν από την παράλληλη επεξεργασία, όπως μεγάλες σαρώσεις πινάκων ή σύνθετες συνδέσεις (Join).

Συνεπώς με την αλλαγή της παραμέτρου max_parallel_workers_per_gather από 2 σε 4 παρατηρήσαμε ότι το Execution Time μειώνεται σε όλες τις περιπτώσεις. Ακόμα παρατηρήσαμε μικρές προσαυξήσεις στο Planning Time.

Σημείωση:

Κάθε query έχει εκτελεστεί πάνω από δύο φορές, ώστε να προλάβουν τα buffers να αρχικοποιηθούν.



3.1

The screenshot shows the pgAdmin 4 interface with a query editor and a results pane. The query is:

```
--shared_buffers = '8GB'  
--max_parallel_workers_per_gather = 4  
--()  
  
explain analyze  
select date() as day, count(*) as num_positions  
from positions  
group by day  
order by num_positions desc;
```

The results pane displays the query plan:

Step	Operation	Cost	Time
1	Sort	(cost=610303.10..611453.44 rows=460137 width=12)	(actual time=546.658..582.520 rows=24 loops=1)
2	Sort Key (count*) DESC		
3	Sort Method: quicksort Memory: 264B		
4	→ Finalize GroupAggregate (cost=323826.99..559158.80 rows=460137 width=12)	(actual time=546.620..582.520 rows=24 loops=1)	
5	Group Key (date())		
6	→ Gather Merge (cost=323826.99..544204.34 rows=1840548 width=12)	(actual time=546.613..582.499 rows=118 loops=1)	
7	Workers Planned: 4		
8	Workers Launched: 4		
9	→ Sort (cost=322826.93..32997.72 rows=460137 width=12)	(actual time=507.947..507.949 rows=24 loops=5)	
10	Sort Key (date())		
11	Sort Method: quicksort Memory: 264B		
12	Worker 0: Sort Method: quicksort Memory: 264B		
13	Worker 1: Sort Method: quicksort Memory: 264B		
14	Worker 2: Sort Method: quicksort Memory: 264B		
15	Worker 3: Sort Method: quicksort Memory: 264B		
16	→ Partial HashAggregate (cost=252187.45..271682.63 rows=460137 width=12)	(actual time=507.452..507.922 rows=24 loops=5)	
17	Group Key (date())		
18	Planned Partitions: 8 Batches: 1 Memory Usage: 309kB		
19	Worker 0: Batches: 1 Memory Usage: 309kB		
20	Worker 1: Batches: 1 Memory Usage: 309kB		
21	Worker 2: Batches: 1 Memory Usage: 309kB		
22	Worker 3: Batches: 1 Memory Usage: 309kB		
23	→ Parallel Seq Scan on positions (cost=0.00..153234.54 rows=1759163 width=4)	(actual time=0.013..305.366 rows=1407330 loops=5)	
24	Planning Time: 0.069 ms		
25	Execution Time: 583.064 ms		

Total rows: 25 of 25 Query complete 00:00:00.643 Ln 13, Col 30

Planning Time: 0.069 ms

Execution Time: 583.064 ms

Query complete 00:00:00.643



3.2

The screenshot shows the pgAdmin 4 interface with a query editor and a results pane. The query is:

```
-- EPOTIMA 3
--shared_buffers = '1GB'
--max_parallel_workers_per_gather = 4
--(11);
explain analyze
select foo.description, foo.type, count()
from (select vesseltypes.description, vessels.id, vessels.type, vessels.flag
      from vessels
      left outer join vesseltypes on vesseltypes.code = vessels.type
     ) as foo
   where foo.flag='Greece'
 group by foo.type, foo.description;
```

The results pane displays the execution plan:

QUERY PLAN	text
1	HashAggregate (cost=18.99..21.43 rows=244 width=42) (actual time=0.132..0.135 rows=29 loops=1)
2	Group Key: vessels.type, vesseltypes.description
3	Batches: 1 Memory Usage: 37kB
4	-> Hash Left Join (cost=4.39..17.16 rows=244 width=34) (actual time=0.031..0.094 rows=244 loops=1)
5	Hash Cond: (vesseltypes.type = vesseltypes.code)
6	-> Seq Scan on vesseltypes (cost=0.00..13.11 rows=244 width=4) (actual time=0.009..0.047 rows=244 loops=1)
7	Filter: ((t1aa).text = 'Greece'.text)
8	Rows Removed by Filter: 245
9	-> Hash (cost=2.06..2.06 rows=106 width=34) (actual time=0.019..0.019 rows=106 loops=1)
10	Buckets: 1024 Batches: 1 Memory Usage: 19kB
11	-> Seq Scan on vessels (cost=0.00..2.06 rows=106 width=34) (actual time=0.003..0.008 rows=106 loops=1)
12	Planning Time: 0.132 ms
13	Execution Time: 0.157 ms

Total rows: 13 of 13 | Query complete 00:00:00.036 | Ln 16, Col 37

Planning Time: 0.132 ms

Execution Time: 0.157 ms

Query complete 00:00:00.037



3.3

Properties SQL Statistics Dependencies Processes Dashboard finalproject/postgres@PostgreSQL_15*

Query History

```

1 -- | EPOTIMA 3 |
2 -- |
3 --
4
5 --shared_buffers = '1GB'
6 --max_parallel_workers_per_gather = 4
7 --(ffff);
8
9 explain analyze
10 select view_1.description, view_1.type, count(*)
11   from view_1
12  join positions on positions.vessel_id = view_1.id
13 where speed > 30
14 group by view_1.type, view_1.description;

```

Data Output Messages Notifications

QUERY PLAN

text
1 Finalize GroupAggregate (cost=155085.44..156130.06 rows=2013 width=42) (actual time=327.814..365.226 rows=10 loops=1)
2 Group Key: vessels.type, vesseltypes.description
3 -> Gather Merge (cost=155085.44..156049.54 rows=8052 width=42) (actual time=327.807..365.199 rows=30 loops=1)
4 Workers Planned: 4
5 Workers Launched: 4
6 -> Sort (cost=154083.38..154090.41 rows=2013 width=42) (actual time=286.414..286.416 rows=6 loops=5)
7 Sort Key: vessels.type, vesseltypes.description
8 Sort Method: quicksort Memory: 25kB
9 Worker 0: Sort Method: quicksort Memory: 25kB
10 Worker 1: Sort Method: quicksort Memory: 25kB
11 Worker 2: Sort Method: quicksort Memory: 25kB
12 Worker 3: Sort Method: quicksort Memory: 25kB
13 -> Partial HashAggregate (cost=153954.79..153974.92 rows=2013 width=42) (actual time=286.346..286.368 rows=6 loops=5)
14 Group Key: vessels.type, vesseltypes.description
15 Batches: 1 Memory Usage: 121kB
16 Worker 0: Batches: 1 Memory Usage: 121kB
17 Worker 1: Batches: 1 Memory Usage: 121kB
18 Worker 2: Batches: 1 Memory Usage: 121kB
19 Worker 3: Batches: 1 Memory Usage: 121kB
20 -> Hash Left Join (cost=21.39..153547.49 rows=54306 width=34) (actual time=1.457..275.108 rows=41948 loops=5)
21 Hash Cond: (vessels.type = vesseltypes.code)
22 -> Hash Join (cost=18.00..153956.33 rows=54306 width=4) (actual time=1.113..266.852 rows=41948 loops=5)
23 Hash Cond: ((positions.vessel_id).text = (vessels.id).text)
24 -> Parallel Seq Scan on positions (cost=0.00..15324.53 rows=54306 width=65) (actual time=0.577..254.622 rows=41948 loops=5)
25 Filter: (speed > 30::double precision)
26 Rows Removed by Filter: 1365382
27 -> Hash (cost=11.89..11.89 rows=489 width=59) (actual time=0.509..0.509 rows=489 loops=5)
28 Buckets: 1024 Batches: 1 Memory Usage: 57kB
29 -> Seq Scan on vessels (cost=0.00..11.89 rows=489 width=69) (actual time=0.309..0.393 rows=489 loops=5)
30 -> Hash (cost=2.06..2.06 rows=106 width=34) (actual time=0.318..0.318 rows=106 loops=5)
31 Buckets: 1024 Batches: 1 Memory Usage: 15kB
32 -> Seq Scan on vesseltypes (cost=0.00..2.06 rows=106 width=34) (actual time=0.280..0.292 rows=106 loops=5)
33 Planning Time: 0.345 ms
34 Execution Time: 365.328 ms

Total rows: 34 of 34 | Query complete 00:00:00.424 | Ln 14, Col 43

Planning Time: 0.345 ms

Execution Time: 365.328 ms

Query complete 00:00:00.424



3.4

Properties SQL Statistics Dependencies Processes Dashboard finalproject/postgres@PostgreSQL 15*

finalproject/postgres@PostgreSQL 15

No limit

Query History

```

1 --+
2 --+ | EPOTHMA 3 |
3 --+
4
5 --shared_buffers = '1GB'
6 --max_parallel_workers_per_gather = 4
7 --(iv):
8
9 explain analyze
10 select date(t) as day, count(*)
11   from view_1
12  join positions on positions.vessel_id = view_1.id
13 where view_1.description like '%Passenger%' and positions.t
14   between '2019-08-14 00:00:00' and '2019-08-18 23:59:59'
15 group by day
16 order by day;

```

Data Output Messages Notifications

QUERY PLAN text

1	GroupAggregate (cost=161345.42..171850.45 rows=76548 width=12) (actual time=292.203..389.946 rows=5 loops=1)
2	Group Key (date(positions.t))
3	-> Gather Merge (cost=161345.42..170510.86 rows=76548 width=4) (actual time=288.625..372.524 rows=361493 loops=1)
4	Workers Planned: 4
5	Workers Launched: 4
6	-> Sort (cost=160345.36..160393.20 rows=19137 width=4) (actual time=252.133..254.194 rows=72299 loops=5)
7	Sort Key: (date(positions.t))
8	Sort Method: quicksort Memory: 3073kB
9	Worker 0: Sort Method: quicksort Memory: 3073kB
10	Worker 1: Sort Method: quicksort Memory: 3073kB
11	Worker 2: Sort Method: quicksort Memory: 3073kB
12	Worker 3: Sort Method: quicksort Memory: 3073kB
13	-> Hash Join (cost=14.03..158998.33 rows=19137 width=4) (actual time=39.537..247.646 rows=72299 loops=5)
14	Hash Cond: ((positions.vessel_id)=text -> (vessels.id)=text)
15	-> Parallel Seq Scan on positions (cost=0.00..157632.44 rows=292438 width=73) (actual time=28.382..203.763 rows=235442 loops=5)
16	Filter: (t >= '2019-08-14 00:00:00'::timestamp without time zone) AND (t <= '2019-08-18 23:59:59'::timestamp without time zone)
17	Rows Removed by Filter: 1171888
18	-> Hash (cost=15.63..15.63 rows=32 width=65) (actual time=0.943..0.956 rows=64 loops=5)
19	Buckets: 1024 Batches: 1 Memory Usage: 15kB
20	-> Hash Join (cost=2.41..15.63 rows=32 width=65) (actual time=0.863..0.928 rows=64 loops=5)
21	Hash Cond: (vessels.type = vesseltypes.code)
22	-> Seq Scan on vessels (cost=0.00..11.89 rows=489 width=69) (actual time=0.518..0.545 rows=489 loops=5)
23	-> Hash (cost=2.33..2.33 rows=7 width=4) (actual time=0.315..0.316 rows=10 loops=5)
24	Buckets: 1024 Batches: 1 Memory Usage: 9kB
25	-> Seq Scan on vesseltypes (cost=0.00..2.33 rows=7 width=4) (actual time=0.304..0.308 rows=10 loops=5)
26	Filter: (description ~~ '%Passenger%')::text
27	Rows Removed by Filter: 96
28	Planning Time: 0.337 ms
29	Execution Time: 390.119 ms

Total rows: 29 of 29 | Query complete 00:00:00.429 | Ln 14, Col 3

Planning Time: 0.337 ms

Execution Time: 390.119 ms

Query complete 00:00:00.429



3.5

3.5.1

Properties SQL Statistics Dependencies Processes Dashboard finalproject/postgres@PostgreSQL 15*

finalproject/postgres@PostgreSQL 15 No limit ▾

Query History Data Output Messages Notifications

```

1 --+-----+
2 --| EPOTHMA 3 |
3 --+-----+
4
5 --shared_buffers = '8GB'
6 --max_parallel_workers_per_gather = 4
7 --(v)
8 --a meros;
9
10 explain analyze
11   select view_1.id, positions.speed, date(t) as day
12   from view_1
13   join positions on positions.vessel_id = view_1.id
14   where view_1.description like '%Cargo%' and positions.t
15     between '2019-08-15 00:00:00' and '2019-08-18 23:59:59' and positions.speed = 0
16   group by day, view_1.id ,positions.speed;
17

```

QUERY PLAN text

1 Group (cost=164413.88..169621.30 rows=40139 width=77) (actual time=333.385..371.265 rows=67 loops=1)

2 Group Key (date(positions.t), vessels.id, positions.speed)

3 → Gather Merge (cost=164413.88..169219.91 rows=40139 width=77) (actual time=333.384..369.512 rows=23182 loops=1)

4 Workers Planned: 4

5 Workers Launched: 4

6 → Sort (cost=163413.82..164348.91 rows=10035 width=77) (actual time=272.947..273.121 rows=4636 loops=5)

7 Sort Key: (date(positions.t)), vessels.id

8 Sort Method: quicksort Memory: 809kB

9 Worker 0: Sort Method: quicksort Memory: 841kB

10 Worker 1: Sort Method: quicksort Memory: 777kB

11 Worker 2: Sort Method: quicksort Memory: 592kB

12 Worker 3: Sort Method: quicksort Memory: 755kB

13 → Hash Join (cost=14.03..162746.85 rows=10035 width=77) (actual time=65.977..268.661 rows=4636 loops=5)

14 Hash Cond: ((positions.vessel_id).text = (vessels.id).text)

15 → Parallel Seq Scan on positions (cost=0.00..162030.35 rows=153344 width=81) (actual time=54.936..250.090 rows=118349 loops=5)

16 Filter: (t >= 2019-08-15 00:00:00 AND t <= 2019-08-18 23:59:59 AND timestamp without time zone AND (speed = 0 OR double precision > 0))

17 Rows Removed by Filter: 1228881

18 → Hash (cost=15.63..15.63 rows=32 width=65) (actual time=1.003..1.004 rows=104 loops=5)

19 Buckets: 1024 Batches: 1 Memory Usage: 18kB

20 → Hash Join (cost=0.00..15.63 rows=32 width=65) (actual time=0.875..0.976 rows=104 loops=5)

21 Hash Cond: (vessels.type = vesseltypes.code)

22 → Seq Scan on vessels (cost=0.00..11.89 rows=489 width=69) (actual time=0.522..0.572 rows=489 loops=5)

23 → Hash (cost=4.23..2.33 rows=7 width=4) (actual time=0.397..0.308 rows=10 loops=5)

24 Buckets: 1024 Batches: 1 Memory Usage: 9kB

25 → Seq Scan on vesseltypes (cost=0.00..2.33 rows=7 width=4) (actual time=0.293..0.298 rows=10 loops=5)

26 Filter: (description ~~ '%Cargo%'.text)

27 Rows Removed by Filter: 96

28 Planning Time: 0.328 ms

29 Execution Time: 371.430 ms

Total rows: 29 of 29 Query complete 00:00:00.403 Lr 15, Col 3

Planning Time: 0.328 ms

Execution Time: 371.430 ms

Query complete 00:00:00.530



3.5.2

Properties SQL Statistics Dependencies Processes Dashboard finalproject/postgres@PostgreSQL 15*

finalproject/postgres@PostgreSQL 15* No limit ▾

Query History

```

1 --+
2 --+ | EPOTHMA 3 |
3 --+-----+
4
5 --shared_buffers = '8GB'
6 --max_parallel_workers_per_gather = 4
7 --(V)
8 --b meros:
9
10 explain analyze
11 select view_1.id, count(case when positions.speed = 0 then 1 else NULL end) as count_zero_speed
12 from view_1
13 join positions on positions.vessel_id = view_1.id
14 where view_1.description like '%Cargo%' and
15     positions.t between '2019-08-12 00:00:00' and '2019-08-19 23:59:59'
16 group by view_1.id
17 having count(case when positions.speed = 0 then 1 else NULL end) = count(view_1.id)
18 order by view_1.id;
19

```

Data Output Messages Notifications

QUERY PLAN text

```

1 Finalize GroupAggregate (cost=160625.96..160880.94 rows=2 width=73) (actual time=293.981..337.149 rows=1 loops=1)
  2   Group Key: vessels.id
  3   Filter: (count(CASE WHEN (positions.speed = 0) THEN 1 ELSE NULL END) - count(vessels.id)) > 0
  4   Rows Removed by Filter: 52
  5   -> Gather Merge (cost=160625.96..160880.16 rows=159627.12 width=81) (actual time=293.700..337.070 rows=253 loops=1)
  6     Workers Planned: 4
  7     Workers Launched: 4
  8       -> Sort (cost=159625.90..159627.12 rows=489 width=81) (actual time=255.654..255.660 rows=51 loops=5)
  9         Sort Key: vessels.id
 10        Sort Method: quicksort Memory: 31kB
 11        Worker 0: Sort Method: quicksort Memory: 31kB
 12        Worker 1: Sort Method: quicksort Memory: 31kB
 13        Worker 2: Sort Method: quicksort Memory: 31kB
 14        Worker 3: Sort Method: quicksort Memory: 31kB
 15       -> Partial HashAggregate (cost=159599.17..159604.06 rows=489 width=81) (actual time=255.473..255.491 rows=51 loops=5)
 16         Group Key: vessels.id
 17         Batches: 1 Memory Usage: 57kB
 18         Worker 0: Batches: 1 Memory Usage: 57kB
 19         Worker 1: Batches: 1 Memory Usage: 57kB
 20         Worker 2: Batches: 1 Memory Usage: 57kB
 21         Worker 3: Batches: 1 Memory Usage: 57kB
 22       -> Hash Join (cost=16.03..199346.83 rows=252234 width=73) (actual time=41.521..249.757 rows=18920 loops=5)
 23         Hash Cond: ((positions.vessel_id)=text AND (vessels.id)=text)
 24         -> Parallel Seq Scan on positions (cost=0.00..157632.44 rows=385604 width=73) (actual time=31.615..205.471 rows=318461 loops=5)
 25           Filter: (t >= '2019-08-12 00:00:00'::timestamp without time zone) AND (t <= '2019-08-19 23:59:59'::timestamp without time zone)
 26           Rows Removed by Filter: 1088870
 27         -> Hash (cost=15.63..15.63 rows=32 width=65) (actual time=0.951..0.953 rows=104 loops=5)
 28           Buckets: 1024 Batches: 1 Memory Usage: 19kB
 29           -> Hash Join (cost=4.21..15.63 rows=32 width=65) (actual time=0.837..0.929 rows=104 loops=5)
 30             Hash Cond: (vessels.type = vesseltypes.code)
 31             -> Seq Scan on vessels (cost=0.00..11.89 rows=489 width=69) (actual time=0.431..0.479 rows=489 loops=5)
 32             -> Hash (cost=2.33..2.33 rows=7 width=4) (actual time=0.350..0.351 rows=10 loops=5)
 33             Buckets: 1024 Batches: 1 Memory Usage: 9kB
 34             -> Seq Scan on vesseltypes (cost=0.00..2.33 rows=7 width=4) (actual time=0.336..0.341 rows=10 loops=5)
 35               Filter: (description ~~ '%Cargo%::text')
 36               Rows Removed by Filter: 56
 37   Planning Time: 0.353 ms
 38   Execution Time: 337.218 ms

```

Total rows: 38 of 38 Query complete 00:00:00.366 Ln 15, Col 3

Planning Time: 0.353 ms

Execution Time: 337.218 ms

Query complete 00:00:00.366



4 Ερώτημα 4 (20%)

shared_buffers = '8GB'

`max_parallel_workers_per_gather = 2`

4.1

Δημιουργήσαμε ευρετήριο τύπου B-Tree στη στήλη date(t) του πίνακα positions (idx_positions_date).

Planning Time: 0.083 ms

Execution Time: 750.487 ms

Query complete 00:00:00.778

The screenshot shows the pgAdmin 4 interface with a query editor containing a PostgreSQL script. The status bar at the bottom right displays the message "pg_size = 83 MB".

```
1 --  
2 --| EPICENTER 4  
3 --  
4  
5 set enable_seqscan = off;  
6  
7 --shared_buffers = '8GB'  
8 --max_parallel_workers_per_gather = 4  
9 [;]  
10  
11 ----index---  
12 --drop index idx_positions_date;  
13  
14 create index idx_positions_date  
15 on positions using btree (t);  
16  
17 select pg_size.pretty(pg_relation_size('idx_positions_date'))  
18  
explain analyze  
19 select count(*) as day, count(*) as num_positions  
20 from positions  
21 group by day  
22 order by num_positions desc;
```

Στην συγκεκριμένη περίπτωση το ευρετήριο δεν βοηθάει στην βελτίωση της απόδοσης, γιατί υπάρχουν πολλές γραμμές με τις ίδιες ή παρόμοιες τιμές.

Ρυθμίζοντας την εντολή enable_seqscan ώστε να είναι απενεργοποιημένη (SET enable_seqscan = off), ο query planner είναι αναγκασμένος να χρησιμοποιήσει τα ευρετήρια που δημιουργήσαμε.



4.2

Δημιουργήσαμε ευρετήριο τύπου B-Tree στη στήλη type του πίνακα vessels (idx_vessels_flag_type_id).

```

Properties SQL Statistics Dependencies Processes Dashboard finalproject/postgres@PostgreSQL 15
finalproject/postgres@PostgreSQL 15
No Hint
Query History
Data Output Messages Explain Notifications
Query Plan Cost
1 HashAggregate (cost=34.4 35.88 rows=244 width=42) (actual time=0.140 0.152 rows=20 loops=1)
  2 Group By: vessels.type, vessels.description
  3 Range: 1 Memory Usage: 376B
  4 > Nested Loop Left Join (cost=53.227 rows=244 width=24) (actual time=0.177 0.170 rows=244 loops=1)
    5 > Index Scan using idx_vessels_flag_type_id on vessels using btree (type) (cost=0.0 0.0 rows=244 width=4) (actual time=0.011 0.027 rows=244 loops=1)
    6 Recheck Cond: (flag.flag = 'Greece')
    7 Heap Block: read=5
      8 > Bitmap Index Scan on idx_vessels_flag_type_id (cost=0.0 0.37 rows=244 width=0) (actual time=0.007 0.007 rows=244 loops=1)
      9 > Memorder (cost=0.0 0.22 rows=1 width=4) (actual time=0.000 0.000 rows=1 loops=244)
    10 > Index Only Scan using idx_vessels_flag_type_id on vessels using btree (flag) (cost=0.0 0.0 rows=1 width=4) (actual time=0.000 0.000 rows=1 loops=244)
    11 Cache Miss: 0
    12 HLR: 215. Misses: 25. Evictions: 0. Memory Usage: 41B
    13 > Index Scan using vessels_types_code on vesseltypes (cost=0.11 0.21 rows=1 width=3) (actual time=0.001 0.001 rows=1 loops=244)
    14 > Index Cond: (code = vessels.type)
    15 Planning Time: 0.184 ms
    16 Execution Time: 0.182 ms
    17
    18
    19
    20
    21
    22
    23
    24
    25
    26
    27
  select pg_size_pretty(pg_relation_size('idx_vessels_flag_type_id'));
  explain analyze
  select foo.description, foo.type, count(*)
  from vesseltypes.description, vessels.id, vessels.type, vessels.flag
  left outer join vesseltypes on vesseltypes.code = vessels.type
  ) as foo
  where foo.flag='Greece'
  group by foo.type, foo.description;
  group by foo.type, foo.description;

```

Total rows: 16 of 18 | Query complete 00:00:00.040

Planning Time: 0.184 ms

Execution Time: 0.182 ms

Query complete 00:00:00.040

```

Properties SQL Statistics Dependencies Processes Dashboard finalproject/postgres@PostgreSQL 15
finalproject/postgres@PostgreSQL 15
No Hint
Query History
Data Output Messages Explain Notifications
Query Plan Cost
1 HashAggregate (cost=34.4 35.88 rows=244 width=42) (actual time=0.140 0.152 rows=20 loops=1)
  2 Group By: vessels.type, vessels.description
  3 Range: 1 Memory Usage: 376B
  4 > Nested Loop Left Join (cost=53.227 rows=244 width=24) (actual time=0.177 0.170 rows=244 loops=1)
    5 > Index Scan using idx_vessels_flag_type_id on vessels using btree (type) (cost=0.0 0.0 rows=244 width=4) (actual time=0.011 0.027 rows=244 loops=1)
    6 Recheck Cond: (flag.flag = 'Greece')
    7 Heap Block: read=5
      8 > Bitmap Index Scan on idx_vessels_flag_type_id (cost=0.0 0.37 rows=244 width=0) (actual time=0.007 0.007 rows=244 loops=1)
      9 > Memorder (cost=0.0 0.22 rows=1 width=4) (actual time=0.000 0.000 rows=1 loops=244)
    10 > Index Only Scan using idx_vessels_flag_type_id on vessels using btree (flag) (cost=0.0 0.0 rows=1 width=4) (actual time=0.000 0.000 rows=1 loops=244)
    11 Cache Miss: 0
    12 HLR: 215. Misses: 25. Evictions: 0. Memory Usage: 41B
    13 > Index Scan using vessels_types_code on vesseltypes (cost=0.11 0.21 rows=1 width=3) (actual time=0.001 0.001 rows=1 loops=244)
    14 > Index Cond: (code = vessels.type)
    15 Planning Time: 0.184 ms
    16 Execution Time: 0.184 ms
    17
    18
    19
    20
    21
    22
    23
    24
    25
    26
    27
  select pg_size_pretty(pg_relation_size('idx_vessels_flag_type_id'));
  explain analyze
  select foo.description, foo.type, count(*)
  from vesseltypes.description, vessels.id, vessels.type, vessels.flag
  left outer join vesseltypes on vesseltypes.code = vessels.type
  ) as foo
  where foo.flag='Greece'
  group by foo.type, foo.description;
  group by foo.type, foo.description;

```

pg_size = 16 KB

Το ευρετήριο περιλαμβάνει μια συνθήκη `where flag = 'Greece'`. Αυτό σημαίνει ότι περιλαμβάνει μόνο τις γραμμές στις οποίες η στήλη flag έχει την τιμή 'Greece'. Η συνθήκη μειώνει τον αριθμό των γραμμών που πρέπει να σαρωθούν κατά την εκτέλεση του ερωτήματος, όμως το σύνολο των εγγραφών στον πίνακα 'vessels' είναι μικρό οπότε στην συγκεκριμένη περίπτωση το ευρετήριο δεν βοηθάει στην βελτίωση της απόδοσης.

Ρυθμίζοντας την εντολή `enable_seqscan` ώστε να είναι απενεργοποιημένη (SET `enable_seqscan = off`), ο query planner είναι αναγκασμένος να χρησιμοποιήσει τα ευρετήρια που δημιουργήσαμε.



4.3

Δημιουργήσαμε ευρετήριο τύπου B-Tree στις στήλες vessel_id, speed του πίνακα positions (idx_positions_vessel_id_speed).

```

1 --+
2 --+ | PSQL1984_4 |
3 --
4
5 set enable_seqscan = off;
6 max_parallel_workers_per_gather = 4
7 shared_buffers = '1000'
8 max_parallel_workers = 1000
9 max_parallel_workers_per_gather = 4
10 --(1);
11 --<--Index<-->
12 drop index idx_positions_vessel_id_speed;
13 create index idx_positions_vessel_id_speed
14 on positions using btree (vessel_id, speed)
15 where speed > 30;
16
17 select pg_size_pretty(pg_relation_size('idx_positions_vessel_id_speed'));
18
19 explain analyze
20 select view_1.description, view_1.type, count(*)
21   from view_1
22   join positions on positions.vessel_id = view_1.id
23  where speed > 30
24  group by view_1.type, view_1.description;

```

Planning Time: 0.438 ms

Execution Time: 135.706 ms

Query complete 00:00:00.174

```

1 --+
2 --+ | PSQL1984_4 |
3 --
4
5 set enable_seqscan = off;
6 max_parallel_workers_per_gather = 4
7 shared_buffers = '1000'
8 max_parallel_workers = 1000
9 max_parallel_workers_per_gather = 4
10 --(1);
11 --<--Index<-->
12 drop index idx_positions_vessel_id_speed;
13 create index idx_positions_vessel_id_speed
14 on positions using btree (vessel_id, speed)
15 where speed > 30;
16
17 select pg_size_pretty(pg_relation_size('idx_positions_vessel_id_speed'));
18
19 explain analyze
20 select view_1.description, view_1.type, count(*)
21   from view_1
22   join positions on positions.vessel_id = view_1.id
23  where speed > 30
24  group by view_1.type, view_1.description;

```

pg_size = 21 MB

Το ευρετήριο περιλαμβάνει μια συνθήκη where speed > 30. Αυτό σημαίνει ότι περιλαμβάνει μόνο τις γραμμές στις οποίες η στήλη speed έχει τιμή μεγαλύτερη του 30. Η συνθήκη μειώνει τον αριθμό των γραμμών που πρέπει να σαρωθούν κατά την εκτέλεση του ερωτήματος.

Ακόμα το ευρετήριο περιλαμβάνει μόνο τις στήλες (vessel_id, speed) που απαιτούνται για το ερώτημα. Αυτό έχει ως αποτέλεσμα να μην χρειάζεται να προσπελάσει τις πραγματικές γραμμές του πίνακα.

Άρα στην συγκεκριμένη περίπτωση το ευρετήριο βοηθάει στην βελτίωση της απόδοσης.

Ρυθμίζοντας την εντολή enable_seqscan ώστε να είναι απενεργοποιημένη (SET enable_seqscan = off), ο query planner είναι αναγκασμένος να χρησιμοποιήσει τα ευρετήρια που δημιουργήσαμε.



4.4

Δημιουργήσαμε ευρετήριο τύπου B-Tree στη στήλη t του πίνακα positions (dx_positions_t).

```

Properties SQL Statistics Dependencies Processes Dashboard finalproject/postgres@PostgreSQL_15
Query History Explain Notifications
Query Data Output Messages Explain Notifications
1 --+-----+
2 | EPDTHM4 |
3 +-----+
4
5 set enable_seqscan = off;
6 --shared_buffers = '8GB'
7 --max_parallel_workers_per_gather = 4
8 ('t');
9
10 ----Index-----
11 --drop index idx_positions_t;
12 --drop index idx_positions_l;
13
14 create index idx_positions_t
15 on positions using btree (t)
16 where t between '2019-08-14 00:00:00' and '2019-08-18 23:59:59';
17
18 select pg_size_pretty(pg_relation_size('idx_positions_t'));
19
20
21 explain analyze
22 select date(t) as day, count(*)
23 from view_l
24 join positions on positions.vessel_id = view_l.id
25 where view_l.description like '%Passenger%' and
26 positions.t between '2019-08-14 00:00:00' and '2019-08-18 23:59:59'
27 group by day
28 order by day;
29
30 Planning Time: 0.374 ms
31 Execution Time: 225.800 ms
32
33 Total rows: 30 of 30 - Query complete 00:00:00.259
34
35

```

Planning Time: 0.374 ms

Execution Time: 225.800 ms

Query complete 00:00:00.259

```

Properties SQL Statistics Dependencies Processes Dashboard finalproject/postgres@PostgreSQL_15
Query History Explain Notifications
Query Data Output Messages Explain Notifications
1 --+-----+
2 | EPDTHM4 |
3 +-----+
4
5 set enable_seqscan = off;
6 --shared_buffers = '8GB'
7 --max_parallel_workers_per_gather = 4
8 ('t');
9
10 ----Index-----
11 --drop index idx_positions_t;
12 --drop index idx_positions_l;
13
14 create index idx_positions_t
15 on positions using btree (t)
16 where t between '2019-08-14 00:00:00' and '2019-08-18 23:59:59';
17
18 select pg_size_pretty(pg_relation_size('idx_positions_l'));
19
20
21 explain analyze
22 select date(t) as day, count(*)
23 from view_l
24 join positions on positions.vessel_id = view_l.id
25 where view_l.description like '%Passenger%' and
26 positions.t between '2019-08-14 00:00:00' and '2019-08-18 23:59:59'
27 group by day
28 order by day;
29
30
31 pg_size = 16 MB
32
33 Planning Time: 0.274 ms
34 Execution Time: 215.800 ms
35
36

```

Το ευρετήριο περιλαμβάνει μια συνθήκη where t between '2019-08-14 00:00:00' and '2019-08-18 23:59:59'. Αυτό σημαίνει ότι περιλαμβάνει μόνο τις γραμμές στις οποίες η στήλη t έχει τιμή μεγαλύτερη '2019-08-14 00:00:00' και μικρότερη ή ίση με '2019-08-18 23:59:59'. Η συνθήκη μειώνει τον αριθμό των γραμμών που πρέπει να σαρωθούν κατά την εκτέλεση του ερωτήματος.

Ρυθμίζοντας την εντολή enable_seqscan ώστε να είναι απενεργοποιημένη (SET enable_seqscan = off), ο query planner είναι αναγκασμένος να χρησιμοποιήσει τα ευρετήρια που δημιουργήσαμε.



4.5

4.5.1

Δημιουργήσαμε ευρετήριο τύπου B-Tree στη στήλη t του πίνακα positions (idx_positions_t_a).

Planning Time: 0.378 ms

Execution Time: 144.351 ms

Query complete 00:00:00.191



The screenshot shows the pgAdmin 4 interface with a query editor containing a complex SQL script. The script includes configuration settings like shared_buffers and max_parallel_workers_per_gather, and performs various operations such as creating indexes and analyzing data. A large result set is displayed in the Data Output tab, showing a single row with a size of 15 MB.

```
Properties SQL Statistics Dependencies Processes Dashboard / finalproject/postgres@PostgreSQL_15*
```

```
finalproject/postgres@PostgreSQL_15
```

```
Query Query History No limit ▾
```

```
Data Output Messages Explain Notifications
```

```
pg_size.pretty
```

```
1 15 MB
```

```
1 --  
2 --  
3 -- PostgreSQL 4  
4 --  
5 set enable_seqscan = off;  
6 --shared_buffers = '96GB'  
7 --max_parallel_workers_per_gather = 4  
8 --  
9 --> max_parallel_workers;  
10 --> max_parallel_workers;  
11  
12 ---- index ----  
13 --drop index idx.positions_t_a;  
14  
15 create index idx.positions_t_a  
on positions using btree()  
where t between '2019-08-10 00:00:00' and '2019-08-18 23:59:59';  
16  
17 select pg_size.pretty(pg_relation_size('idx.positions_t_a'));  
18  
19 explain analyze  
20 select view1.id, positions.speed, date(t) as day  
21 from view1  
22 join positions on positions.view1_id = view1.id  
23 where view1.description like '%Cargo%' and positions.t  
24 between '2019-08-10 00:00:00' and '2019-08-18 23:59:59' and positions.speed >  
25 group by day, view1.id, positions.speed;
```

Το ευρετήριο περιλαμβάνει μια συνθήκη where t between '2019-08-15 00:00:00' and '2019-08-18 23:59:59'. Αυτό σημαίνει ότι περιλαμβάνει μόνο τις γραμμές στις οποίες η στήλη t έχει τιμή μεγαλύτερη '2019-08-15 00:00:00' και μικρότερη ή ίση με '2019-08-18 23:59:59'. Η συνθήκη μειώνει τον αριθμό των γραμμών που πρέπει να σαρωθούν κατά την εκτέλεση του ερωτήματος.

Άρα στην συγκεκριμένη περίπτωση το ευρετήριο Βοηθάει στην βελτίωση της απόδοσης.

Ρυθμίζοντας την εντολή enable_seqscan ώστε να είναι απενεργοποιημένη (SET enable_seqscan = off), ο query planner είναι αναγκασμένος να χρησιμοποιήσει τα ευρετήρια που δημιουργήσαμε.



4.5.2

Δημιουργήσαμε ευρετήριο τύπου B-Tree στη στήλη `t` του πίνακα `positions` (`idx_positions_t_b`).

1 `\i`

2 `\c test`

3 `\d`

4 `\o /tmp/test/postgres/PostgreSQL.1`

5 `set enable_seqscan = off;`

6 `--shared_buffers = 1GB`

7 `--max_parallel_workers_per_gather = 4`

8 `--(v)`

9 `0 errors`

10 `---finidx---`

11 `create index idx_positions_t_1;`

12 `create index idx_positions_t_b`
on positions using btree t;

13 `where t between '2019-08-12 00:00:00' and '2019-08-10 23:00:00';`

14 `select pg_stx.pretty(pg.relation_stx('idx_positions_t_b'));`

15 `analyze idx_positions_t_b;`

16 `analyze idx_positions_t_1;`

17 `analyze idx_positions_t_b;`

18 `analyze idx_positions_t_1;`

19 `analyze idx_positions_t_b;`

20 `analyze idx_positions_t_1;`

21 `analyze idx_positions_t_b;`

22 `analyze idx_positions_t_1;`

23 `analyze idx_positions_t_b;`

24 `analyze idx_positions_t_1;`

25 `analyze idx_positions_t_b;`

26 `analyze idx_positions_t_1;`

27 `analyze idx_positions_t_b;`

28 `analyze idx_positions_t_1;`

29 `analyze idx_positions_t_b;`

30 `analyze idx_positions_t_1;`

31 `analyze idx_positions_t_b;`

32 `analyze idx_positions_t_1;`

33 `analyze idx_positions_t_b;`

34 `analyze idx_positions_t_1;`

35 `analyze idx_positions_t_b;`

36 `analyze idx_positions_t_1;`

37 `analyze idx_positions_t_b;`

38 `analyze idx_positions_t_1;`

39 `analyze idx_positions_t_b;`

Planning Thrust: 0.387 ms

Execution Time: 7.671 ms

Total rows: 39 of 39 Query complete: 00:00:00.207

1 `QUERY PLAN`

2 `Table Scan using idx_positions_t_b on test.t (cost=17421.30..11866.28 rows=2 width=73) (actual time=10714.16 ms) (0 loops=1)`

3 `Filter: (pos < 10000 AND pos > 0) -> (width=73) (actual time=10714.16 ms) (0 loops=1)`

4 `Rows Removed by Filter: 0`

5 `-> Gather Merge (cost=17421.34..17445.50 rows=1856 width=73) (actual time=15145.16 ms) (0 loops=1)`

6 `Workers Planned: 4`

7 `Workers Launched: 4`

8 `Sort Key: pos::int`

9 `Sort Key: pos::int`

10 `Sort Method: quicksort Memory: 2768`

11 `Worker 0 Sort Method: quicksort Memory: 3116`

12 `Worker 1 Sort Method: quicksort Memory: 3116`

13 `Worker 2 Sort Method: quicksort Memory: 3116`

14 `Worker 3 Sort Method: quicksort Memory: 3116`

15 `-> Parallel HashAggregate (cost=173154.51..173189.40 rows=489 width=49) (actual time=122384.12 ms) (0 loops=1)`

16 `Group Key: view_id::int`

17 `Batches: 1 Memory Usage: 376B`

18 `Worker 0: 1 Memory Usage: 376B`

19 `Worker 1: 1 Memory Usage: 376B`

20 `Worker 2: 1 Memory Usage: 376B`

21 `Worker 3: 1 Memory Usage: 376B`

22 `Hash Total (cost=173154.51..173189.40 rows=489 width=49) (actual time=435.117 ms) (0 loops=1)`

23 `-> Item Count (positions_view_id, t_id) (actual time=435.117 ms) (0 loops=1)`

24 `-> Parallel Bitmap Index Scan on positions (cost=16652.11..16652.50 rows=385064 width=73) (actual time=251.847..73.373 ms) (0 loops=1)`

25 `Bitmap Index Scan on positions (cost=16652.11..16652.50 rows=385064 width=73) (actual time=251.847..73.373 ms) (0 loops=1)`

26 `-> Block Bitmap Scan (cost=16652.11..16652.50 rows=385064 width=73) (actual time=251.847..73.373 ms) (0 loops=1)`

27 `-> Bitmap Index Scan on idx_positions_t_b (cost=0..66 rows=15643 width=0) (actual time=17.685..67.686 ms) (0 loops=1)`

28 `-> Hash (cost=1024.82..911 rows=236.327 width=104 loops=1)`

29 `Buckets: 1024 Batches: 1 Memory Usage: 181B`

30 `-> Hash Join (cost=0..39.327 rows=1 width=104) (actual time=16.708..0.747 ms) (0 loops=1)`

31 `-> Seq Scan on positions (cost=0..0 rows=104 width=104) (actual time=0.000..0.000 ms) (0 loops=1)`

32 `-> Index Scan using view_id_index on view (cost=8.27..87.70 rows=104 width=104) (actual time=0.030..0.189 ms) (0 loops=1)`

33 `-> Hash (cost=84.80..104 rows=104 width=104) (actual time=203.358 ms) (0 loops=10 loops=1)`

34 `-> Hash (cost=104.81..104 rows=104 width=104) (actual time=0.047 ms) (0 loops=1)`

35 `-> Filter (actual_time = 0.047 ms) (actual_time = 0.047 ms) (0 loops=1)`

36 `-> Seq Scan on resources (cost=14..14.00 rows=1 width=4) (actual time=0.039..0.047 ms) (0 loops=1)`

37 `-> Filter (actual_time = 0.047 ms) (actual_time = 0.047 ms) (0 loops=1)`

38 `-> Rows Removed by Filter: 90`

Planning Time: 0.387 ms

Execution Time: 176.178 ms

Query complete 00:00:00.207

The screenshot shows the pgAdmin 4 interface with a query editor containing a complex SQL script. The script includes various PostgreSQL configuration parameters, index creation, and a query to calculate the size of a specific index. The results pane on the right displays the output of the 'pg_size.pretty(pg_relation_size('idx_positions_t_b'))' command, showing a single row with a value of 21 MB.

```
Properties SQL Statistics Dependencies Processes Dashboard / mailproject/postgres@PostgreSQL_15
finalproject/postgres/PostgreSQL_15
Query History
Query
1
2 --+
3 --+ | EPOTHMA 4 |
4 --+
5
6 set enable_nosync = off;
7
8 --shared_buffers = '6GB'
9 --max_parallel_workers_per_gather = 4
10
11 --n
12 --+-- neros;
13
14 --+-- Index+--+
15 --+drop_index idx_positions_t_b;
16
17 create index idx_positions_t_b
18 on positions using btree (t)
19 where t between '2019-08-12 00:00:00' and '2019-08-19 23:59:59';
20
21 select pg_size.pretty(pg_relation_size('idx_positions_t_b'));
22
23
24 explain analyze
25 select view_1.id, count(case when positions.speed < 0 then 1 else NULL end) as count_zero_speed
26 from view_1
27 join positions on positions.vessel_id = view_1.id
28 where positions.date_time like '%Cargo%' and
29 positions.t between '2019-08-12 00:00:00' and '2019-08-19 23:59:59'
30 group by view_1.id
31 having count(case when positions.speed < 0 then 1 else NULL end) > count(view_1.id)
32 order by view_1.id;
```

Data Output Messages Explain Notifications

pg_size.pretty
test 21 MB

Το ευρετήριο περιλαμβάνει μια συνθήκη where t between '2019-08-12 00:00:00' and '2019-08-19 23:59:59'. Αυτό σημαίνει ότι περιλαμβάνει μόνο τις γραμμές στις οποίες η στήλη t έχει τιμή μεγαλύτερη '2019-08-12 00:00:00' και μικρότερη ή ίση με '2019-08-19 23:59:59'. Η συνθήκη μειώνει τον αριθμό των γραμμών που πρέπει να σαρωθούν κατά την εκτέλεση του ερωτήματος.

Άρα στην συγκεκριμένη περίπτωση το ευρετήριο βοηθάει στην βελτίωση της απόδοσης.

Ρυθμίζοντας την εντολή enable_seqscan ώστε να είναι απενεργοποιημένη (SET enable_seqscan = off), ο query planner είναι αναγκασμένος να χρησιμοποιήσει τα ευρετήρια που δημιουργήσαμε.



5 Ερώτημα 5 (20%)

shared_buffers = '128MB'

max_parallel_workers_per_gather = 2

Σπάσαμε τον πίνακα 'positions' σε partitions χρησιμοποιώντας την μέθοδο διαμέριση μέσω κληρονομικότητας μεταξύ πινάκων. Ο πίνακας 'positions' παρέμεινε ο ίδιος με μοναδική διαφορά ότι ως πρωτεύον κλειδί προστέθηκε η στήλη t καθώς η διαμέριση έγινε με βάση αυτή την μεταβλητή. Ενώ οι πίνακες 'vessels' και 'vesseltypes' δεν διαμερίζονται.

Δημιουργήσαμε 24 partitions καθένα από τα οποία αντιστοιχεί σε μία ημερολογιακή ημέρα που περιέχει ο πίνακας 'positions', ο χωρισμός αυτός έγινε με range {CHECK (t >= '2019-08-01' AND t < '2019-08-02')}.

Ακόμα δημιουργήσαμε μια συνάρτηση ενεργοποίησης (positions_insert_trigger) για τη δρομολόγηση των εισερχόμενων δεδομένων στο κατάλληλο partition με βάση την ημερομηνία της στήλης t.

Επίσης δημιουργήσαμε ένα trigger (insert_positions_trigger) για την εκτέλεση της συνάρτησης positions_insert_trigger πριν από κάθε πράξη INSERT στον κύριο πίνακα θέσεων.

Τέλος χρησιμοποιούμε την εντολή COPY για τη φόρτωση δεδομένων στον πίνακα 'positions'.

Επιλέξαμε να χωρίσουμε τα partitions με βάση την στήλη t του πίνακα 'positions', καθώς περιέχεται σε ένα μεγάλο αριθμό των επερωτήσεων και ο πίνακας 'positions' περιέχει τα περισσότερα δεδομένα συγκριτικά με τους άλλους 2 πίνακες ('vessels', 'vesseltypes'). Ενώ ο τρόπος με τον οποίο χωρίστηκαν τα partitions επιλέχθηκε για να μπορεί να συμπεριληφθεί μια ολόκληρη μέρα σε κάθε partition καθώς λόγο του τύπου timestamp περιέχεται εκτός από την ημερομηνία και η ώρα, για αυτόν τον λόγο δεν ήταν δυνατή η υλοποίηση με κάποιον άλλο τρόπο.



5.1

Παρατήρηση:

To partitioning δεν βοηθάει στην βελτίωση εκτέλεσης του query καθώς χρησιμοποιούνται όλα τα partitions.

```

1
2
3
4
5 --shared_buffers = '128MB'
6 --max_parallel_workers_per_gather = 2
7
8 --();
9
10 explain analyze
11 select date() as day, count(*) as num_positions
12   from positions
13   group by day
14   order by num_positions desc;
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47

```

Planning Time: 0.479 ms

Execution Time: 1114.177 ms

Query complete 00:00:01.150



5.2

Παρατήρηση:

Το query δεν επηρεάζεται από το partition καθώς δεν υπάρχει ο πίνακας positions σε αυτό.

The screenshot shows the pgAdmin 4 interface with a query editor containing the following SQL code:

```
1 -- EPOTHMIA 5
2 --
3 --
4
5 --shared_buffers = '128MB'
6 --max_parallel_workers_per_gather = 2
7
8 --(11):
9
10 explain analyze
11 select foo.description, foo.type, count(*)
12   from (select vesseltypes.description, vessels.id, vessels.type, vessels.flag
13          from vesseltypes
14          left outer join vesseltypes on vesseltypes.code = vessels.type
15         ) as foo
16   where foo.flag='Greece'
17   group by foo.type, foo.description;
```

The Data Output tab displays the EXPLAIN ANALYZE results:

QUERY PLAN	Cost	Time
Final Aggregate (cost=18.99..21.43 rows=244 width=40) (actual time=0.152..0.157 rows=29 loops=1)	1	
Output Key Vessels type, vesseltypes.description	2	
Batches 1 Memory Usage 27kB	3	
-> Hash Left Join (cost=3.59..17.18 rows=244 width=34) (actual time=0.038..0.112 rows=244 loops=1)	4	
Hash Cond: (vesseltypes.type = vessels.type)	5	
-> Seq Scan on vesseltypes (cost=0.00..13.11 rows=244 width=4) (actual time=0.009..0.045 rows=244 loops=1)	6	
-> Filter (flag = 'Greece' AND)	7	
Rows Removed by Filter: 245	8	
-> Hash (cost=2.06..2.36 rows=106 width=34) (actual time=0.024..0.024 rows=106 loops=1)	9	
Buckets 1024 Batches 1 Memory Usage 15kB	10	
-> Seq Scan on vessels (cost=0..10.26 rows=106 width=34) (actual time=0.003..0.013 rows=106 loops=1)	11	
Planning Time: 0.142 ms	12	
Execution Time: 0.179 ms	13	

Total rows: 19 of 13 Query complete 00:00:00.045 Ln 17, Col 37

Planning Time: 0.142 ms

Execution Time: 0.179 ms

Query complete 00:00:00.045



5.3

Παρατήρηση:

To partitioning δεν βοηθάει στην βελτίωση εκτέλεσης του query καθώς χρησιμοποιούνται όλα τα partitions.

```

1 -- EPOTHMA 5
2
3
4 --shared_buffers = '128MB'
5 --max_parallel_workers_per_gather = 2
6
7
8 --(11):
9
10 explain analyze
11 select view_1.description, view_1.type, count(*)
12 from view_1
13 join positions on positions.vessel_id = view_1.id
14 where speed > 30
15 group by view_1.type, view_1.description;
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102

```

Total rows: 102 of 102 Query complete 00:00:00.579 Log 15 Col 42

Planning Time: 0.378 ms

Execution Time: 144.351 ms

Query complete 00:00:00.191



5.4

Παρατήρηση:

Ο χρόνος εκτέλεσης του query βελτιώνεται καθώς χρησιμοποιούνται μόνο τα partitions που αναφέρονται στις ζητούμενες ημέρες.

```

1 -- shared_buffers = '128MB'
2 --max_parallel_workers_per_gather = 2
3 --(iv):
4 explain analyze
5 select date_trunc('day', count(*)
6   from positions
7   join positions on positions.vessel_id = view_1.id
8   where view_1.description like '%Passenger%' and positions.t
9   between '2019-08-14 00:00:00' and '2019-08-18 23:59:59'
10 group by day
11 order by day;
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42

```

The screenshot shows the pgAdmin 4 interface with the following details:

- Object Explorer:** Shows the database schema with tables, procedures, and triggers.
- Properties:** Shows the connection details to the 'finalproject/postgres@PostgreSQL_15' database.
- Query:** The SQL query being executed.
- Data Output:** Shows the results of the query.
- Messages:** Shows any messages or errors.
- Explain:** Shows the detailed query plan.
- Notifications:** Shows any notifications.
- QUERY PLAN:**
 - Workers Launched: 2
 - Sort (cost=34402.77..34402.77 rows=200 width=12) (actual time=173.453..173.458 rows=2 loops=2)
 - Sort Key: (date(positions.t))
 - Sort Method: quicksort Memory: 256B
 - Worker 0: Sort Method: quicksort Memory: 256B
 - Worker 1: Sort Method: quicksort Memory: 256B
 - Group Key: date(positions.t)
 - Batches: 1 Memory Usage: 40KB
 - Worker 0: Batches: 1 Memory Usage: 40KB
 - Worker 1: Batches: 1 Memory Usage: 40KB
 - Hash Join (cost=16.63..34231.67 rows=32092 width=4) (actual time=1.089..158.178 rows=130498 loops=2)
 - Hash Cond: (positions.vessel_id = vessel.id)
 - Hash: 1 Memory Usage: 40KB
 - Parallel HashAggregate (cost=34392.13..34394.63 rows=200 width=12) (actual time=173.457..173.442 rows=2 loops=2)
 - Sort Method: quicksort Memory: 256B
 - Parallel Scan on positions_2019_08_16 positions_4 (cost=0..8722.12 rows=76.914 rows=350259 loops=1)
 - Filter (t ~ 2019-08-14 00:00:00..timestamp without time zone) AND (t <= 2019-08-18 23:59:59 .. timestamp without time zone)
 - Parallel Seq Scan on positions_2019_08_17 positions_3 (cost=0..0 rows=126592 width=72) (actual time=0.406..56.471 rows=30385 loops=1)
 - Filter (t ~ 2019-08-14 00:00:00..timestamp without time zone) AND (t <= 2019-08-18 23:59:59 .. timestamp without time zone)
 - Parallel Seq Scan on positions_2019_08_15 positions_3 (cost=0..0 rows=103881 width=72) (actual time=0.413..19.020 rows=47223 loops=1)
 - Filter (t ~ 2019-08-14 00:00:00..timestamp without time zone) AND (t <= 2019-08-18 23:59:59 .. timestamp without time zone)
 - Parallel Seq Scan on positions_2019_08_18 positions_8 (cost=0..0 rows=84110 width=72) (actual time=0.004..22.390 rows=261904 loops=1)
 - Filter (t ~ 2019-08-14 00:00:00..timestamp without time zone) AND (t <= 2019-08-18 23:59:59 .. timestamp without time zone)
 - Parallel Seq Scan on positions_2019_08_14 positions_2 (cost=0..0 rows=20996 width=72) (actual time=0..1331 rows=72994 loops=1)
 - Filter (t ~ 2019-08-14 00:00:00..timestamp without time zone) AND (t <= 2019-08-18 23:59:59 .. timestamp without time zone)
 - Parallel Seq Scan on positions_2019_08_13 positions_1 (cost=0..0 rows=1 width=154) (actual time=0..001..0.001 rows=1 loops=1)
 - Filter (t ~ 2019-08-14 00:00:00..timestamp without time zone) AND (t <= 2019-08-18 23:59:59 .. timestamp without time zone)
 - Hash (cost=15.53..15.53 rows=32 width=6) (actual time=0.758..0.759 rows=64 loops=2)
 - Buckets: 1024 Batches: 1 Memory Usage: 19KB
 - Hash Join (cost=2.31..2.31 rows=7 width=5) (actual time=0.334..0.335 rows=10 loops=2)
 - Buckets: 1024 Batches: 1 Memory Usage: 9KB
 - Seq Scan on vesseltypes (cost=0..0 rows=23 width=4) (actual time=0.315..0.321 rows=10 loops=1)
 - Filter: description ~ '%Passenger%'
 - Rows Removed by Filter: 96
- Planning Time:** 0.580 ms
- Execution Time:** 213.250 ms
- Query complete:** 00:00:00.254



5.5

5.5.1

Παρατήρηση:

Ο χρόνος εκτέλεσης του query βελτιώνεται καθώς χρησιμοποιούνται μόνο τα partitions που αναφέρονται στις ζητούμενες ημέρες.

```

File Object Tools Help
Object Explorer Properties SQL Statistics Dependencies Processes Dashboard finalproject/postgres@PostgreSQL 15
Tables (27)
positions
    Columns (8)
    Constraints (2)
    Indexes
    MLS Policies
    Rules
    Triggers (1)
        Insert_positions_trigger
            positions_insert_trigger()
    views
        view_1
            columns
            constraints
            indexes
            triggers
            types
            views
    sequences
    functions
    materializedviews
    subscriptions
    tables
    tablespaces
    roles
    tablespaces
    Postgres 16
    total rows: 40 of 40 | Query complete 00:00:00.195 | Ln 17, Col 4
    
```

Query

```

1 -- 
2 -- 
3 -- 
4 --shared_buffers = '128MB'
5 --max_parallel_workers_per_gather = 2
6 --max_parallel_workers = 2
7 
8 --(v)
9 --> merges;
10 
11 explain analyse
12 select vms.id, positions.speed, date(t) as day
13 from view_1
14 join positions on positions.vessel_id = view_1.id
15 where view_1.description like '%cargo%' and positions.t
16 between '2019-08-15 00:00:00' and '2019-08-18 23:59:59'
17 and positions.speed >
18 group by day, view_1.id, positions.speed;
19 
```

QUERY PLAN

```

1 BatchScan 1 Memory Usage: 156KB
2 -> Gather (cost=32153.02 38571.08 rows=32170 width=77) (actual time=154.201 159.222 rows=102 loops=1)
3   Workers Planned: 2
4   Workers Launched: 2
5   Hash Join (cost=14.03 31032.00 31154.08 rows=16085 width=77) (actual time=131.116 131.196 rows=34 loops=1)
6     Group Key: date(positions.id), vessels.id, positions.speed
7     Batches: 1 Memory Usage: 793B
8     Worker 0: Batches: 1 Memory Usage: 793B
9     Worker 1: Batches: 1 Memory Usage: 793B
10    -> Hash Join (cost=14.03 31032.00 31154.08 rows=16085 width=77) (actual time=143.123 259 rows=7727 loops=3)
11      Hash Cond: (positions.vessel_id = vessels.id) AND (vessels.id < 10000)
12      -> Parallel HashAggregate (cost=31153.02 31154.08 rows=16085 width=77) (actual time=131.116 131.196 rows=34 loops=3)
13        Group Key: date(positions.id), vessels.id, positions.speed
14        Batches: 1 Memory Usage: 793B
15        -> Parallel Append (cost=0.00 23982.56 rows=245794 width=81) (actual time=0.403 102.930 rows=197246 loops=3)
16          Parallel Seq Scan on positions_2019_08_08 (cost=0.00 9058.97 rows=77270 width=81) (actual time=0.413 32.174 rows=568951 loops=3)
17          Filter: (t >= 2019-08-15 00:00:00 AND t <= 2019-08-18 23:59:59) AND (timestamp without time zone) AND (speed = 0) double precision)
18          Rows Removed by Filter: 57862
19          Parallel Seq Scan on positions_2019_08_09 (cost=0.00 7894.83 rows=49508 width=81) (actual time=0.546 81.640 rows=167191 loops=1)
20          Filter: (t >= 2019-08-15 00:00:00 AND t <= 2019-08-18 23:59:59) AND (timestamp without time zone) AND (speed = 0) double precision)
21          Rows Removed by Filter: 13686
22          Parallel Seq Scan on positions_2019_08_10 (cost=0.00 6430.57 rows=56523 width=81) (actual time=0.314 34.928 rows=66461 loops=2)
23          Filter: (t >= 2019-08-15 00:00:00 AND t <= 2019-08-18 23:59:59) AND (timestamp without time zone) AND (speed = 0) double precision)
24          Rows Removed by Filter: 55623
25          Parallel Seq Scan on positions_2019_08_11 (cost=0.00 5242.22 rows=40402 width=81) (actual time=0.265 38.379 rows=111123 loops=1)
26          Filter: (t >= 2019-08-15 00:00:00 AND t <= 2019-08-18 23:59:59) AND (timestamp without time zone) AND (speed = 0) double precision)
27          Rows Removed by Filter: 50771
28          Parallel Seq Scan on positions_2019_08_12 (cost=0.00 4490.57 rows=56523 width=81) (actual time=0.201 34.928 rows=66461 loops=1)
29          Filter: (t >= 2019-08-15 00:00:00 AND t <= 2019-08-18 23:59:59) AND (timestamp without time zone) AND (speed = 0) double precision)
30          Rows Removed by Filter: 55623
31          Hash Join (cost=15.63 15.63 rows=32 width=65) (actual time=0.769 0.843 rows=104 loops=1)
32            Hash Cond: vessels.type = vesseltypes.type
33            -> Hash Scan on vessels (cost=0.00 11.89 rows=489 width=65) (actual time=0.396 0.416 rows=489 loops=1)
34            -> Hash Scan on vesseltypes (cost=0.00 2.73 rows=7 width=4) (actual time=0.257 0.258 rows=10 loops=1)
35            -> Hash Join (cost=15.63 15.63 rows=32 width=65) (actual time=0.769 0.843 rows=104 loops=1)
36              Hash Cond: vesseltypes.type = vesseltypes.type
37              Filter: (description = 'cargo') AND (vessel_id < 10000)
38              Rows Removed by Filter: 96
39 Planning Time: 0.742 ms
40 Execution Time: 160.321 ms

```

Planning Time: 0.742 ms

Execution Time: 160.321 ms

Query complete 00:00:00.195



5.5.2

Παρατήρηση:

Ο χρόνος εκτέλεσης του query βελτιώνεται καθώς χρησιμοποιούνται μόνο τα partitions που αναφέρονται στις ζητούμενες ημέρες.

Planning Time: 0.617 ms

Execution Time: 257.926 ms

Query complete 00:00:00.299