

Ова е прифатливо време за апликацијата, па затоа не пристапуваме кон индексирање.

2. vw_arrivals_board

Време на извршување според AirportName (Destination), подредени според ArrivalTime.

```
1 Gather Merge (cost=17424.33..17785.56 rows=3096 width=242) (actual time=185.636..219.831 rows=87744 loops=1)
2   Workers Planned: 2
3   Workers Launched: 2
4   -> Sort (cost=16424.31..16428.18 rows=1548 width=242) (actual time=179.114..183.248 rows=29248 loops=3)
5     Sort Key: f.arrivaltime
6     Sort Method: external merge Disk: 3496kB
7     Worker 0: Sort Method: external merge Disk: 3448kB
8     Worker 1: Sort Method: external merge Disk: 3384kB
9     -> Hash Left Join (cost=9381.50..16342.29 rows=1548 width=242) (actual time=72.886..149.699 rows=29248 loops=3)
10       Hash Cond: (ar.taxiwayid = tx.taxiwayid)
11       -> Hash Join (cost=9379.83..16335.98 rows=1548 width=212) (actual time=72.795..142.453 rows=29248 loops=3)
12         Hash Cond: (ar.runwayid = r.runwayid)
13         -> Hash Join (cost=9378.67..16328.30 rows=1548 width=162) (actual time=72.637..135.758 rows=29248 loops=3)
14           Hash Cond: (g.terminalid = t.terminalid)
15           -> Hash Join (cost=9377.42..16321.42 rows=1548 width=102) (actual time=72.584..129.144 rows=29248 loops=3)
16             Hash Cond: (ar.gateid = g.gateid)
17             -> Hash Join (cost=9374.60..16314.32 rows=1548 width=98) (actual time=72.501..121.898 rows=29248 loops=3)
18               Hash Cond: (ar.originalairportid = a.orig.airportid)
19               -> Parallel Hash Join (cost=9372.14..16307.53 rows=1548 width=83) (actual time=72.426..114.516 rows=29248 loops=3)
20                 Hash Cond: (ar.flightid = f.flightid)
21                 -> Parallel Seq Scan on arrivals ar (cost=0.00..6551.77 rows=100578 width=48) (actual time=2.532..13.708 rows=80462 loops=3)
22                 -> Parallel Hash (cost=9332.11..9332.11 rows=3202 width=47) (actual time=69.664..69.667 rows=29248 loops=3)
23                   Buckets: 131072 (originally 8192) Batches: 1 (originally 1) Memory Usage: 10272kB
24                   -> Hash Join (cost=1.82..9332.11 rows=3202 width=47) (actual time=0.076..49.460 rows=29248 loops=3)
25                     Hash Cond: (f.destinationairportid = a_dest.airportid)
26                     -> Parallel Seq Scan on flights f (cost=0.00..8748.32 rows=288132 width=32) (actual time=0.025..23.917 rows=166585 loops=3)
27                     -> Hash (cost=1.81..1.81 rows=1 width=27) (actual time=0.023..0.024 rows=1 loops=3)
28                       Buckets: 1024 Batches: 1 Memory Usage: 9kB
29                       -> Seq Scan on airport a_dest (cost=0.00..1.81 rows=1 width=27) (actual time=0.010..0.017 rows=1 loops=3)
30                         Filter: ((airportname)::text = 'Skopje International Airport'::text)
31                         Rows Removed by Filter: 64
32                       -> Hash (cost=1.65..1.65 rows=65 width=27) (actual time=0.050..0.051 rows=65 loops=3)
33                         Buckets: 1024 Batches: 1 Memory Usage: 12kB
34                         -> Seq Scan on airport a_orig (cost=0.00..1.65 rows=65 width=27) (actual time=0.016..0.025 rows=65 loops=3)
35                       -> Hash (cost=1.81..1.81 rows=81 width=16) (actual time=0.055..0.055 rows=81 loops=3)
36                         Buckets: 1024 Batches: 1 Memory Usage: 12kB
37                         -> Seq Scan on gate g (cost=0.00..1.81 rows=81 width=16) (actual time=0.021..0.032 rows=81 loops=3)
38                       -> Hash (cost=1.11..1.11 rows=11 width=72) (actual time=0.026..0.027 rows=11 loops=3)
39                         Buckets: 1024 Batches: 1 Memory Usage: 9kB
40                         -> Seq Scan on terminal t (cost=0.00..1.11 rows=11 width=72) (actual time=0.013..0.015 rows=11 loops=3)
41                       -> Hash (cost=1.07..1.07 rows=7 width=62) (actual time=0.049..0.050 rows=7 loops=3)
42                         Buckets: 1024 Batches: 1 Memory Usage: 9kB
43                         -> Seq Scan on runway r (cost=0.00..1.07 rows=7 width=62) (actual time=0.040..0.042 rows=7 loops=3)
44                       -> Hash (cost=1.30..1.30 rows=30 width=42) (actual time=0.048..0.049 rows=30 loops=3)
45                         Buckets: 1024 Batches: 1 Memory Usage: 10kB
46                       -> Seq Scan on taxiway tx (cost=0.00..1.30 rows=30 width=42) (actual time=0.033..0.037 rows=30 loops=3)
47 Planning Time: 3.217 ms
48 Execution Time: 224.300 ms
```

Ова е прифатливо време за апликацијата, па затоа не пристапуваме кон индексирање.

3. vw_ticket_info

Време на извршување на погледот, според даден тикет.

```
Gather (cost=1001.86..397048.25 rows=25 width=162) (actual time=36.395..992.595 rows=14 loops=1)
  Workers Planned: 4
  Workers Launched: 4
  -> Nested Loop (cost=1.86..396045.75 rows=6 width=162) (actual time=508.508..947.492 rows=3 loops=5)
    -> Nested Loop (cost=1.43..396025.39 rows=6 width=136) (actual time=207.992..947.355 rows=6 loops=5)
      -> Nested Loop (cost=1.29..396024.44 rows=6 width=86) (actual time=207.942..947.258 rows=6 loops=5)
        -> Nested Loop (cost=0.86..396005.53 rows=6 width=58) (actual time=207.874..947.020 rows=6 loops=5)
          -> Nested Loop (cost=0.44..395986.53 rows=6 width=45) (actual time=207.773..946.848 rows=6 loops=5)
            -> Parallel Seq Scan on ticket (cost=0.00..395935.80 rows=6 width=4) (actual time=207.628..946.573 rows=6 loops=5)
              Filter: (passengerid = 10)
              Rows Removed by Filter: 4980014
            -> Index Scan using ticket_pkey on ticket t (cost=0.44..8.46 rows=1 width=45) (actual time=0.029..0.029 rows=1 loops=28)
              Index Cond: (ticketid = ticket.ticketid)
          -> Index Scan using passenger_pkey on passenger p (cost=0.42..3.17 rows=1 width=25) (actual time=0.024..0.024 rows=1 loops=28)
              Index Cond: (passengerid = t.passengerid)
        -> Index Scan using flights_pkey on flights f (cost=0.42..3.15 rows=1 width=40) (actual time=0.035..0.035 rows=1 loops=28)
              Index Cond: (flightid = t.flightid)
      -> Index Scan using airline_pkey on airline a (cost=0.14..0.16 rows=1 width=62) (actual time=0.010..0.010 rows=1 loops=28)
              Index Cond: (airlineid = f.airlineid)
    -> Index Scan using luggage_pkey on luggage l (cost=0.43..3.39 rows=1 width=34) (actual time=0.014..0.014 rows=0 loops=28)
              Index Cond: (luggageid = t.luggageid)
Planning Time: 1.474 ms
JIT:
  Functions: 135
  Options: Inlining false, Optimization false, Expressions true, Deforming true
  Timing: Generation 11.312 ms (Deform 5.772 ms), Inlining 0.000 ms, Optimization 4.741 ms, Emission 99.231 ms, Total 115.285 ms
Execution Time: 994.653 ms
```

Ова е прифатливо време за апликацијата, па затоа не пристапуваме кон индексирање.

Време на извршување на погледот, според FlightNumber.

```
Gather (cost=25680.00..422409.33 rows=50 width=162) (actual time=2487.226..2496.006 rows=50 loops=1)
  Workers Planned: 4
  Workers Launched: 4
  InitPlan 1
    -> Limit (cost=15408.53..15408.53 rows=1 width=12) (actual time=166.903..166.906 rows=1 loops=1)
      -> Sort (cost=15408.53..16657.32 rows=499516 width=12) (actual time=144.474..144.475 rows=1 loops=1)
        Sort Key: (random())
        Sort Method: top-N heapsort Memory: 25kB
        -> Seq Scan on flights (cost=0.00..12910.95 rows=499516 width=12) (actual time=0.035..75.250 rows=499516 loops=1)
      -> Nested Loop (cost=9271.46..405995.80 rows=12 width=162) (actual time=1189.192..1430.683 rows=10 loops=5)
        -> Hash Join (cost=9271.03..405990.04 rows=12 width=136) (actual time=1188.519..1188.899 rows=10 loops=5)
          Hash Cond: (f.airlineid = a.airlineid)
          -> Nested Loop (cost=9269.08..405988.06 rows=12 width=86) (actual time=1169.291..1169.660 rows=10 loops=5)
            -> Parallel Hash Join (cost=9268.66..405982.72 rows=12 width=73) (actual time=1164.925..1165.206 rows=10 loops=5)
              Hash Cond: (t.flightid = f.flightid)
              -> Parallel Seq Scan on ticket t (cost=0.00..380373.24 rows=6225024 width=45) (actual time=0.069..648.800 rows=4980019 loops=5)
              -> Parallel Hash (cost=9268.65..9268.65 rows=1 width=40) (actual time=9.538..9.539 rows=0 loops=5)
                Buckets: 1024 Batches: 1 Memory Usage: 40kB
                -> Parallel Seq Scan on flights f (cost=0.00..9268.65 rows=1 width=40) (actual time=24.024..47.653 rows=1 loops=1)
                  Filter: (flightnumber = (InitPlan 1).col1)
                  Rows Removed by Filter: 499515
            -> Index Scan using passenger_pkey on passenger p (cost=0.42..0.45 rows=1 width=25) (actual time=0.442..0.442 rows=1 loops=50)
              Index Cond: (passengerid = t.passengerid)
          -> Hash (cost=1.42..1.42 rows=42 width=62) (actual time=19.059..19.059 rows=42 loops=5)
            Buckets: 1024 Batches: 1 Memory Usage: 11kB
            -> Seq Scan on airline a (cost=0.00..1.42 rows=42 width=62) (actual time=19.021..19.031 rows=42 loops=5)
        -> Index Scan using luggage_pkey on luggage l (cost=0.43..0.48 rows=1 width=34) (actual time=24.176..24.176 rows=1 loops=50)
          Index Cond: (luggageid = t.luggageid)
Planning Time: 1.078 ms
JIT:
  Functions: 158
  Options: Inlining false, Optimization false, Expressions true, Deforming true
  Timing: Generation 12.082 ms (Deform 6.603 ms), Inlining 0.000 ms, Optimization 5.480 ms, Emission 112.140 ms, Total 129.702 ms
Execution Time: 2498.241 ms
```

Ова НЕ е прифатливо време за апликацијата, па затоа пристапуваме кон индексирање.

```
CREATE INDEX idx_flights_flightnumber ON flights(flightnumber);
CREATE INDEX idx_ticket_flightid ON ticket(flightid);
```

После индексирање, се добиваат следните резултати:

1	Nested Loop (cost=15412.20..15490.00 rows=50 width=162) (actual time=194.493..194.498 rows=0 loops=1)
2	InitPlan 1
3	-> Limit (cost=15408.53..15408.53 rows=1 width=12) (actual time=150.882..150.884 rows=1 loops=1)
4	-> Sort (cost=15408.53..16657.32 rows=499516 width=12) (actual time=150.880..150.881 rows=1 loops=1)
5	Sort Key: (random())
6	Sort Method: top-N heapsort Memory: 25kB
7	-> Seq Scan on flights (cost=0.00..12910.95 rows=499516 width=12) (actual time=0.020..0.1552 rows=499516 loops=1)
8	-> Hash Join (cost=3.23..57.48 rows=50 width=136) (actual time=193.714..194.468 rows=50 loops=1)
9	Hash Cond: (f.airlineid = a.airlineid)
10	-> Nested Loop (cost=1.28..55.39 rows=50 width=86) (actual time=193.662..194.399 rows=50 loops=1)
11	-> Nested Loop (cost=0.86..33.13 rows=50 width=73) (actual time=151.012..151.052 rows=50 loops=1)
12	-> Index Scan using idx_flights_flightnumber on flights f (cost=0.42..8.44 rows=1 width=40) (actual time=150.947..150.948 rows=1 loops=1)
13	Index Cond: (flightnumber = (InitPlan 1).col1)
14	-> Index Scan using idx_ticket_flightid on ticket t (cost=0.44..20.62 rows=407 width=45) (actual time=0.059..0.085 rows=50 loops=1)
15	Index Cond: (flightid = f.flightid)
16	-> Index Scan using passenger_pkey on passenger p (cost=0.42..0.45 rows=1 width=25) (actual time=0.866..0.866 rows=1 loops=50)
17	Index Cond: (passengerid = t.passengerid)
18	-> Hash (cost=1.42..1.42 rows=42 width=62) (actual time=0.035..0.035 rows=42 loops=1)
19	Buckets: 1024 Batches: 1 Memory Usage: 11kB
20	-> Seq Scan on airline a (cost=0.00..1.42 rows=42 width=62) (actual time=0.017..0.023 rows=42 loops=1)
21	-> Index Scan using luggage_pkey on luggage l (cost=0.43..0.48 rows=1 width=34) (actual time=0.000..0.000 rows=0 loops=50)
22	Index Cond: (luggageid = t.luggageid)
23	Planning Time: 0.966 ms
24	Execution Time: 194.576 ms

24 rows ▾ ⋮

4. vw_boarding_pass_details

Време на извршување на погледот според TicketId и FlightNumber.

1	Gather (cost=1728873.81..2238619.30 rows=1 width=148) (actual time=136590.909..136601.068 rows=0 loops=1)
2	Workers Planned: 4
3	Workers Launched: 4
4	InitPlan 1
5	-> Limit (cost=753874.68..753874.68 rows=1 width=12) (actual time=8187.634..8187.637 rows=1 loops=1)
6	-> Sort (cost=753874.68..816124.92 rows=24900096 width=12) (actual time=7550.778..7550.780 rows=1 loops=1)
7	Sort Key: (random())
8	Sort Method: top-N heapsort Memory: 25kB
9	-> Seq Scan on ticket (cost=0.00..629374.20 rows=24900096 width=12) (actual time=0.081..4128.416 rows=24900097 loops=1)
10	InitPlan 2
11	-> Limit (cost=15408.53..15408.53 rows=1 width=12) (actual time=137.043..137.045 rows=1 loops=1)
12	-> Sort (cost=15408.53..16657.32 rows=499516 width=12) (actual time=137.001..137.001 rows=1 loops=1)
13	Sort Key: (random())
14	Sort Method: top-N heapsort Memory: 25kB
15	-> Seq Scan on flights (cost=0.00..12910.95 rows=499516 width=12) (actual time=0.025..68.496 rows=499516 loops=1)
16	-> Parallel Hash Semi Join (cost=958590.59..1468335.98 rows=1 width=148) (actual time=128220.904..128220.916 rows=0 loops=5)
17	Hash Cond: (bp.passid = bp_1.passid)
18	-> Hash Join (cost=771526.51..1272802.78 rows=3226332 width=148) (never executed)
19	Hash Cond: (g.terminalid = ter.terminalid)
20	-> Hash Join (cost=771525.27..1261069.41 rows=3226332 width=88) (never executed)
21	Hash Cond: (bp.gateid = g.gateid)
22	-> Parallel Hash Left Join (cost=771522.44..1252154.35 rows=3226332 width=84) (never executed)
23	Hash Cond: (t.luggageid = l.luggageid)
24	-> Parallel Hash Join (cost=542092.31..916411.09 rows=3226332 width=70) (never executed)
25	Hash Cond: (t.flightid = f.flightid)
26	-> Parallel Hash Join (cost=530742.34..896591.94 rows=3226332 width=66) (never executed)
27	Hash Cond: (t.passengerid = p.passengerid)
28	-> Parallel Hash Join (cost=506819.04..793314.48 rows=3226332 width=63) (never executed)
29	Hash Cond: (bp.ticketid = t.ticketid)
30	-> Parallel Seq Scan on boardingpass bp (cost=0.00..178981.32 rows=3226332 width=36) (never executed)
31	-> Parallel Hash (cost=380373.24..380373.24 rows=6225024 width=39) (never executed)
33	-> Parallel Hash (cost=18339.69..18339.69 rows=321169 width=15) (never executed)
34	-> Parallel Seq Scan on passenger p (cost=0.00..18339.69 rows=321169 width=15) (never executed)
35	-> Parallel Hash (cost=8748.32..8748.32 rows=208132 width=16) (never executed)
36	-> Parallel Seq Scan on flights f (cost=0.00..8748.32 rows=208132 width=16) (never executed)
37	-> Parallel Hash (cost=166562.95..166562.95 rows=3251295 width=26) (never executed)
38	-> Parallel Seq Scan on luggage l (cost=0.00..166562.95 rows=3251295 width=26) (never executed)
39	-> Hash (cost=1.81..1.81 rows=81 width=16) (never executed)
40	-> Seq Scan on gate g (cost=0.00..1.81 rows=81 width=16) (never executed)
41	-> Hash (cost=1.11..1.11 rows=11 width=72) (never executed)
42	-> Seq Scan on terminal ter (cost=0.00..1.11 rows=11 width=72) (never executed)
43	-> Parallel Hash (cost=187064.07..187064.07 rows=1 width=4) (actual time=128220.682..128220.685 rows=0 loops=5)
44	Buckets: 1024 Batches: 1 Memory Usage: 0kB
45	-> Nested Loop (cost=0.86..187064.07 rows=1 width=4) (actual time=128211.337..128211.339 rows=0 loops=5)
46	Join Filter: (t_1.flightid = f_1.flightid)
47	Rows Removed by Join Filter: 0
48	-> Nested Loop (cost=0.44..187055.61 rows=1 width=12) (actual time=106363.137..128211.305 rows=0 loops=5)
49	-> Parallel Seq Scan on boardingpass bp_1 (cost=0.00..187047.15 rows=1 width=12) (actual time=106321.103..128169.269 rows=0 loops=5)
50	Filter: (ticketid = (InitPlan 1).col1)
51	Rows Removed by Filter: 2581065
52	-> Index Scan using ticket_pkey on ticket t_1 (cost=0.44..8.46 rows=1 width=12) (actual time=210.030..210.031 rows=1 loops=1)
53	Index Cond: (ticketid = (InitPlan 1).col1)
54	-> Index Scan using idx_flights_flightnumber on flights f_1 (cost=0.42..8.44 rows=1 width=4) (actual time=0.127..0.129 rows=1 loops=1)
55	Index Cond: (flightnumber = (InitPlan 2).col1)
56	Planning Time: 63.547 ms
57	JIT:
58	Functions: 351
59	Options: Inlining true, Optimization true, Expressions true, Deforming true
60	Timing: Generation 28.330 ms (Deform 12.477 ms), Inlining 619.956 ms, Optimization 1582.613 ms, Emission 1084.346 ms, Total 3315.245 ms
61	Execution Time: 136604.847 ms

Ова НЕ е прифатливо време за апликацијата, па затоа пристапуваме кон индексирање.

```
CREATE INDEX idx_boardingpass_ticketid ON boardingpass(ticketid);
```

После индексирање, се добиваат следните резултати:

```
1 Gather (cost=772551.90..1282297.38 rows=1 width=148) (actual time=685.553..692.092 rows=0 loops=1)
2   Workers Planned: 4
3   Workers Launched: 4
4   -> Hash Semi Join (cost=771551.90..1281297.28 rows=1 width=148) (actual time=605.056..605.063 rows=0 loops=5)
5     Hash Cond: (bp.passid = bp_1.passid)
6     -> Hash Join (cost=771526.51..1272802.78 rows=3226332 width=148) (never executed)
7       Hash Cond: (g.terminalid = ter.terminalid)
8       -> Hash Join (cost=771525.27..1261069.41 rows=3226332 width=88) (never executed)
9         Hash Cond: (bp.gateid = g.gateid)
10        -> Parallel Hash Left Join (cost=771522.44..1252154.35 rows=3226332 width=84) (never executed)
11          Hash Cond: (t.luggageid = l.luggageid)
12          -> Parallel Hash Join (cost=542092.31..916411.09 rows=3226332 width=70) (never executed)
13            Hash Cond: (t.flightid = f.flightid)
14            -> Parallel Hash Join (cost=530742.34..896591.94 rows=3226332 width=66) (never executed)
15              Hash Cond: (t.passengerid = p.passengerid)
16              -> Parallel Hash Join (cost=506819.04..793314.48 rows=3226332 width=63) (never executed)
15                Hash Cond: (bp.ticketid = t.ticketid)
18                -> Parallel Seq Scan on boardingpass bp (cost=0.00..178981.32 rows=3226332 width=36) (never executed)
19                -> Parallel Hash (cost=380373.24..380373.24 rows=6225024 width=39) (never executed)
20                  -> Parallel Seq Scan on ticket t (cost=0.00..380373.24 rows=6225024 width=39) (never executed)
21                  -> Parallel Hash (cost=18339.69..18339.69 rows=321169 width=15) (never executed)
22                  -> Parallel Seq Scan on passenger p (cost=0.00..18339.69 rows=321169 width=15) (never executed)
23                  -> Parallel Hash (cost=8748.32..8748.32 rows=208132 width=16) (never executed)
24                  -> Parallel Seq Scan on flights f (cost=0.00..8748.32 rows=208132 width=16) (never executed)
25                  -> Parallel Hash (cost=166562.95..166562.95 rows=3251295 width=26) (never executed)
26                  -> Parallel Seq Scan on luggage l (cost=0.00..166562.95 rows=3251295 width=26) (never executed)
27                  -> Hash (cost=1.81..1.81 rows=81 width=16) (never executed)
28                  -> Seq Scan on gate g (cost=0.00..1.81 rows=81 width=16) (never executed)
29                  -> Hash (cost=1.11..1.11 rows=11 width=72) (never executed)
30                  -> Seq Scan on terminal ter (cost=0.00..1.11 rows=11 width=72) (never executed)
31                  -> Hash (cost=25.37..25.37 rows=1 width=4) (actual time=604.921..604.923 rows=0 loops=5)
32                    Buckets: 1024 Batches: 1 Memory Usage: 8kB
33                    -> Nested Loop (cost=1.29..25.37 rows=1 width=4) (actual time=604.920..604.921 rows=0 loops=5)
34                      Join Filter: (t_1.flightid = f_1.flightid)
35                      Rows Removed by Join Filter: 1
36                      -> Nested Loop (cost=0.87..16.92 rows=1 width=12) (actual time=604.820..604.824 rows=1 loops=5)
37                        -> Index Scan using idx_boardingpass_ticketid on boardingpass bp_1 (cost=0.43..8.45 rows=1 width=12) (actual time=0.049..0.050 rows=1)
38                          Index Cond: (ticketid = 1502)
39                        -> Index Scan using ticket_pkey on ticket t_1 (cost=0.44..8.46 rows=1 width=12) (actual time=0.049..0.050 rows=1)
40                          Index Cond: (ticketid = 1502)
41                        -> Index Scan using idx_flights_flightnumber on flights f_1 (cost=0.42..8.44 rows=1 width=4) (actual time=0.076..0.077 rows=1)
42                          Index Cond: (flightnumber = 270795)
43 Planning Time: 2.915 ms
44 JIT:
45   Functions: 335
46   Options: Inlining true, Optimization true, Expressions true, Deforming true
47   Timing: Generation 22.051 ms (Deform 9.813 ms), Inlining 409.013 ms, Optimization 1558.677 ms, Emission 1055.732 ms, Total 3045.472 ms
48 Execution Time: 696.138 ms
```

5. vw_passenger_flight_history

Време на извршување на погледот според PassportNumber.

1	Gather (cost=20143.04..416861.76 rows=25 width=73) (actual time=10916.684..12443.411 rows=29 loops=1)
2	Workers Planned: 4
3	Workers Launched: 4
4	-> Nested Loop (cost=19143.04..415859.26 rows=6 width=73) (actual time=10988.620..12397.787 rows=6 loops=5)
5	-> Parallel Hash Join (cost=19142.62..415856.62 rows=6 width=53) (actual time=10959.771..12312.882 rows=6 loops=5)
6	Hash Cond: (t.passengerid = p.passengerid)
7	-> Parallel Seq Scan on ticket t (cost=0.00..380373.24 rows=6225024 width=40) (actual time=0.050..663.410 rows=4980019..)
8	-> Parallel Hash (cost=19142.61..19142.61 rows=1 width=25) (actual time=10813.683..10813.684 rows=0 loops=5)
9	Buckets: 1024 Batches: 1 Memory Usage: 40kB
10	-> Parallel Seq Scan on passenger p (cost=0.00..19142.61 rows=1 width=25) (actual time=8647.131..10813.120 rows=..)
11	Filter: ((passportnumber)::text = 'ZAF5450551'::text)
12	Rows Removed by Filter: 199124
13	-> Index Scan using flights_pkey on flights f (cost=0.42..0.44 rows=1 width=32) (actual time=14.625..14.625 rows=1 loops=29)
14	Index Cond: (flightid = t.flightid)
15	Planning Time: 0.692 ms
16	JIT:
17	Functions: 90
18	Options: Inlining false, Optimization false, Expressions true, Deforming true
19	Timing: Generation 9.841 ms (Deform 6.307 ms), Inlining 0.000 ms, Optimization 4.767 ms, Emission 87.520 ms, Total 102.128 ms
20	Execution Time: 12446.009 ms

Ова НЕ е прифатливо време за апликацијата, па затоа пристапуваме кон индексирање.

```
CREATE INDEX idx_passenger_passportnumber ON passenger(passportnumber);
```

После индексирање, се добиваат следните резултати:

1	Gather (cost=1008.88..397727.60 rows=25 width=73) (actual time=434.709..1555.356 rows=29 loops=1)
2	Workers Planned: 4
3	Workers Launched: 4
4	-> Nested Loop (cost=8.88..396725.10 rows=6 width=73) (actual time=546.272..1511.445 rows=6 loops=5)
5	-> Hash Join (cost=8.46..396722.45 rows=6 width=53) (actual time=546.154..1511.110 rows=6 loops=5)
6	Hash Cond: (t.passengerid = p.passengerid)
7	-> Parallel Seq Scan on ticket t (cost=0.00..380373.24 rows=6225024 width=40) (actual time=0.066..649.079 rows=4980019..)
8	-> Hash (cost=8.44..8.44 rows=1 width=25) (actual time=18.514..18.515 rows=1 loops=5)
9	Buckets: 1024 Batches: 1 Memory Usage: 9kB
10	-> Index Scan using idx_passenger_passportnumber on passenger p (cost=0.42..8.44 rows=1 width=25) (actual time=1..)
11	Index Cond: ((passportnumber)::text = 'ZAF5450551'::text)
12	-> Index Scan using flights_pkey on flights f (cost=0.42..0.44 rows=1 width=32) (actual time=0.048..0.048 rows=1 loops=29)
13	Index Cond: (flightid = t.flightid)
14	Planning Time: 0.546 ms
15	JIT:
16	Functions: 90
17	Options: Inlining false, Optimization false, Expressions true, Deforming true
18	Timing: Generation 9.200 ms (Deform 5.908 ms), Inlining 0.000 ms, Optimization 4.639 ms, Emission 87.730 ms, Total 101.569 ms
19	Execution Time: 1557.227 ms

6. vw_airport_infrastructure

1	Subquery Scan on vw_airport_infrastructure (cost=9.78..10.56 rows=6 width=64) (actual time=0.662..0.937 rows=6 loops=1)
2	-> GroupAggregate (cost=9.78..10.50 rows=6 width=68) (actual time=0.660..0.934 rows=6 loops=1)
3	Group Key: ap.airportid
4	-> Sort (cost=9.78..9.89 rows=44 width=52) (actual time=0.619..0.664 rows=664 loops=1)
5	Sort Key: ap.airportid, t.terminalid
6	Sort Method: quicksort Memory: 75kB
7	-> Hash Right Join (cost=6.10..8.58 rows=44 width=52) (actual time=0.135..0.288 rows=664 loops=1)
8	Hash Cond: (t.airportid = ap.airportid)
9	-> Hash Right Join (cost=1.25..3.35 rows=81 width=16) (actual time=0.025..0.056 rows=83 loops=1)
10	Hash Cond: (g.terminalid = t.terminalid)
11	-> Seq Scan on gate g (cost=0.00..1.81 rows=81 width=12) (actual time=0.007..0.012 rows=81 ..
12	-> Hash (cost=1.11..1.11 rows=11 width=12) (actual time=0.012..0.013 rows=11 loops=1)
13	Buckets: 1024 Batches: 1 Memory Usage: 9kB
14	-> Seq Scan on terminal t (cost=0.00..1.11 rows=11 width=12) (actual time=0.007..0.00..
15	-> Hash (cost=4.78..4.78 rows=6 width=44) (actual time=0.101..0.102 rows=37 loops=1)
16	Buckets: 1024 Batches: 1 Memory Usage: 11kB
17	-> Hash Left Join (cost=3.37..4.78 rows=6 width=44) (actual time=0.068..0.091 rows=37 loops=
18	Hash Cond: (ap.airportid = r.airportid)
19	-> Hash Right Join (cost=2.21..3.59 rows=6 width=40) (actual time=0.051..0.065 rows=3..
20	Hash Cond: (tx.airportid = ap.airportid)
21	-> Seq Scan on taxiway tx (cost=0.00..1.30 rows=30 width=12) (actual time=0.007..
22	-> Hash (cost=2.14..2.14 rows=6 width=36) (actual time=0.036..0.036 rows=6 loop=
23	Buckets: 1024 Batches: 1 Memory Usage: 9kB
24	-> Seq Scan on airport ap (cost=0.00..2.14 rows=6 width=36) (actual time=..
25	Filter: (airportid = ANY ('{1,2,3,5,8,17}'::integer[]))
26	Rows Removed by Filter: 59
27	-> Hash (cost=1.07..1.07 rows=7 width=12) (actual time=0.011..0.011 rows=7 loops=1)
28	Buckets: 1024 Batches: 1 Memory Usage: 9kB
29	-> Seq Scan on runway r (cost=0.00..1.07 rows=7 width=12) (actual time=0.007..0..
30	Planning Time: 0.682 ms
31	Execution Time: 1.024 ms

Ова е прифатливо време за апликацијата, па затоа не пристапуваме кон индексирање.

7. vw_maintenance_overview

Време на извршување на погледот според статус и помеѓу даден временски период (StartDate и EndDate):

1	Gather (cost=22326.36..34961.91 rows=11510 width=119) (actual time=80.355..1058.425 rows=17046 loops=1)
2	Workers Planned: 2
3	Workers Launched: 2
4	-> Hash Left Join (cost=21326.36..32810.91 rows=4796 width=119) (actual time=74.567..187.042 rows=5682 loops=3)
5	Hash Cond: (wo.employeeid = e.employeeid)
6	-> Hash Left Join (cost=21172.21..32620.18 rows=4796 width=95) (actual time=72.590..182.666 rows=5682 loops=3)
7	Hash Cond: (am.airportid = ap.airportid)
8	-> Hash Join (cost=21169.75..32604.30 rows=4796 width=80) (actual time=72.491..180.959 rows=5682 loops=3)
9	Hash Cond: (am.aircraftid = ac.aircraftid)
10	-> Parallel Hash Right Join (cost=21152.50..32574.35 rows=4796 width=74) (actual time=72.047..178.494 rows=5682 loops=3)
11	Hash Cond: (wo.maintenanceid = am.maintenanceid)
12	-> Parallel Seq Scan on workson wo (cost=0.00..10250.67 rows=446167 width=16) (actual time=0.019..49.603 rows=356933 loops=3)
13	-> Parallel Hash (cost=21109.16..21109.16 rows=3467 width=66) (actual time=71.719..71.719 rows=3432 loops=3)
14	Buckets: 16384 Batches: 1 Memory Usage: 1216kB
15	-> Parallel Seq Scan on aircraftmaintenance am (cost=0.00..21109.16 rows=3467 width=66) (actual time=0.040..68.988 rows=3432 loops=3)
16	Filter: ((startdate >= '2024-01-01 00:00:00'::timestamp without time zone) AND (startdate <= '2024-06-30 00:00:00'::timestamp without time zone) AND ((status)::text = 'In Progress'::text))
17	Rows Removed by Filter: 329901
18	-> Hash (cost=11.00..11.00 rows=500 width=18) (actual time=0.277..0.278 rows=500 loops=3)
19	Buckets: 1024 Batches: 1 Memory Usage: 34kB
20	-> Seq Scan on aircraft ac (cost=0.00..11.00 rows=500 width=18) (actual time=0.042..0.151 rows=500 loops=3)
21	-> Hash (cost=1.65..1.65 rows=65 width=27) (actual time=0.057..0.057 rows=65 loops=3)
22	Buckets: 1024 Batches: 1 Memory Usage: 12kB
23	-> Seq Scan on airport ap (cost=0.00..1.65 rows=65 width=27) (actual time=0.028..0.036 rows=65 loops=3)
24	-> Hash (cost=104.62..104.62 rows=3962 width=19) (actual time=1.939..1.939 rows=3962 loops=3)
25	Buckets: 4096 Batches: 1 Memory Usage: 239kB
26	-> Seq Scan on employee e (cost=0.00..104.62 rows=3962 width=19) (actual time=0.018..0.920 rows=3962 loops=3)
27	Planning Time: 1.071 ms
28	Execution Time: 1059.307 ms

Ова е прифатливо време за апликацијата, па затоа не пристапуваме кон индексирање.

8. vw_aircraft_status

Време на извршување на погледот според статусот на самиот авион.

1	Hash Left Join (cost=18.49..29480.41 rows=217116 width=73) (act...
2	Hash Cond: (ac.aircrafttypeid = at2.aircrafttypeid)
3	-> Hash Right Join (cost=16.32..28861.86 rows=217116 width=6...
4	Hash Cond: (am.aircraftid = ac.aircraftid)
5	-> Seq Scan on aircraftmaintenance am (cost=0.00..2796...
6	Filter: ((status)::text = ANY ('{Scheduled,"In Pro...
7	Rows Removed by Filter: 666774
8	-> Hash (cost=12.25..12.25 rows=326 width=42) (actual ...
9	Buckets: 1024 Batches: 1 Memory Usage: 33kB
10	-> Seq Scan on aircraft ac (cost=0.00..12.25 row...
11	Filter: ((status)::text = 'Active'::text)
12	Rows Removed by Filter: 174
13	-> Hash (cost=1.52..1.52 rows=52 width=21) (actual time=0.02...
14	Buckets: 1024 Batches: 1 Memory Usage: 11kB
15	-> Seq Scan on aircrafttype at2 (cost=0.00..1.52 rows=...
16	Planning Time: 0.962 ms
17	Execution Time: 389.555 ms

Ова е прифатливо време за апликацијата, па затоа не пристапуваме кон индексирање.

9.vw_gateoccupancy

Време на извршување на погледот според GateCode.

```
1 Gather (cost=1010.57..11880.55 rows=1420 width=252) (actual time=7.752..4428.124 rows=2099 loops=1)
2   Workers Planned: 2
3   Workers Launched: 2
4   -> Hash Join (cost=10.57..10738.55 rows=592 width=252) (actual time=32.533..4390.845 rows=700 loops=3)
5     Hash Cond: (d.destinationairportid = ap_dest.airportid)
6     -> Hash Join (cost=8.10..10734.43 rows=592 width=237) (actual time=32.359..4390.297 rows=700 loops=3)
7       Hash Cond: (f.airlineid = al.airlineid)
8       -> Nested Loop (cost=6.16..10730.78 rows=592 width=187) (actual time=32.142..4389.577 rows=700 loops=3)
9         -> Hash Join (cost=5.74..10304.82 rows=592 width=179) (actual time=2.239..3829.631 rows=700 loops=3)
10           Hash Cond: (t.airportid = ap.airportid)
11           -> Hash Join (cost=3.27..10300.71 rows=592 width=164) (actual time=2.186..3829.249 rows=700 loops=3)
12             Hash Cond: (g.terminalid = t.terminalid)
13             -> Hash Join (cost=2.03..10297.30 rows=592 width=54) (actual time=2.121..3828.841 rows=700 loops=3)
14               Hash Cond: (d.gateid = g.gateid)
15               -> Parallel Seq Scan on departures d (cost=0.00..10162.87 rows=47933 width=41) (actual time=2..
16                 Filter: ((status)::text = ANY ('{Scheduled,"Go to gate",Boarding} '::text[]))
17                 Rows Removed by Filter: 47740
18                 -> Hash (cost=2.01..2.01 rows=1 width=21) (actual time=0.037..0.038 rows=1 loops=3)
19                   Buckets: 1024 Batches: 1 Memory Usage: 9kB
20                   -> Seq Scan on gate g (cost=0.00..2.01 rows=1 width=21) (actual time=0.027..0.032 rows=
21                     Filter: (gatecode = 206)
22                     Rows Removed by Filter: 80
23                     -> Hash (cost=1.11..1.11 rows=11 width=118) (actual time=0.039..0.040 rows=11 loops=3)
24                       Buckets: 1024 Batches: 1 Memory Usage: 9kB
25                       -> Seq Scan on terminal t (cost=0.00..1.11 rows=11 width=118) (actual time=0.029..0.032 rows=
26                         -> Hash (cost=1.65..1.65 rows=65 width=27) (actual time=0.032..0.034 rows=65 loops=3)
27                           Buckets: 1024 Batches: 1 Memory Usage: 12kB
28                           -> Seq Scan on airport ap (cost=0.00..1.65 rows=65 width=27) (actual time=0.007..0.016 rows=65 loop
29                         -> Index Scan using flights_pkey on flights f (cost=0.42..0.72 rows=1 width=16) (actual time=0.799..0.799 rows=
30                           Index Cond: (flightid = d.flightid)
31                         -> Hash (cost=1.42..1.42 rows=42 width=62) (actual time=0.192..0.194 rows=42 loops=3)
32                           Buckets: 1024 Batches: 1 Memory Usage: 11kB
33                         -> Seq Scan on airline al (cost=0.00..1.42 rows=42 width=62) (actual time=0.171..0.177 rows=42 loops=3)
34                         -> Hash (cost=1.65..1.65 rows=65 width=27) (actual time=0.064..0.065 rows=65 loops=3)
35                           Buckets: 1024 Batches: 1 Memory Usage: 12kB
36                           -> Seq Scan on airport ap_dest (cost=0.00..1.65 rows=65 width=27) (actual time=0.035..0.043 rows=65 loops=3)
37 Planning Time: 2.333 ms
38 Execution Time: 4428.398 ms
```

Ова НЕ е прифатливо време за апликацијата, па затоа пристапуваме кон индексирање.

```
CREATE INDEX idx_departures_status ON departures(status);
```

После индексирање, се добиваат следните резултати:

```

1 Gather (cost=1010.57..11880.55 rows=1420 width=252) (actual time=0.913..67.784 rows=2099 loops=1)
2   Workers Planned: 2
3   Workers Launched: 2
4   -> Hash Join (cost=10.57..10738.55 rows=592 width=252) (actual time=0.574..33.457 rows=700 loops=3)
5     Hash Cond: (d.destinationairportid = ap_dest.airportid)
6     -> Hash Join (cost=8.10..10734.43 rows=592 width=237) (actual time=0.399..33.017 rows=700 loops=3)
7       Hash Cond: (f.airlineid = al.airlineid)
8       -> Nested Loop (cost=6.16..10730.78 rows=592 width=187) (actual time=0.331..32.634 rows=700 loops=3)
9         -> Hash Join (cost=5.74..10304.82 rows=592 width=179) (actual time=0.246..26.976 rows=700 loops=3)
10          Hash Cond: (t.airportid = ap.airportid)
11          -> Hash Join (cost=3.27..10300.71 rows=592 width=164) (actual time=0.192..26.702 rows=700 loops=3)
12            Hash Cond: (g.terminalid = t.terminalid)
13            -> Hash Join (cost=2.03..10297.30 rows=592 width=54) (actual time=0.132..26.400 rows=700 loops=3)
14              Hash Cond: (d.gateid = g.gateid)
15              -> Parallel Seq Scan on departures d (cost=0.00..10162.87 rows=47933 width=41) (actual time=0...
16                Filter: ((status)::text = ANY ('{Scheduled,"Go to gate",Boarding} '::text[]))
17                Rows Removed by Filter: 47740
18              -> Hash (cost=2.01..2.01 rows=1 width=21) (actual time=0.035..0.036 rows=1 loops=3)
19                Buckets: 1024 Batches: 1 Memory Usage: 9kB
20              -> Seq Scan on gate g (cost=0.00..2.01 rows=1 width=21) (actual time=0.025..0.031 rows=1...
21                Filter: (gatecode = 206)
22                Rows Removed by Filter: 80
23              -> Hash (cost=1.11..1.11 rows=11 width=118) (actual time=0.036..0.036 rows=11 loops=3)
24                Buckets: 1024 Batches: 1 Memory Usage: 9kB
25              -> Seq Scan on terminal t (cost=0.00..1.11 rows=11 width=118) (actual time=0.026..0.028 rows=1...
26              -> Hash (cost=1.65..1.65 rows=65 width=27) (actual time=0.032..0.033 rows=65 loops=3)
27                Buckets: 1024 Batches: 1 Memory Usage: 12kB
28              -> Seq Scan on airport ap (cost=0.00..1.65 rows=65 width=27) (actual time=0.006..0.014 rows=65 loops...
29              -> Index Scan using flights_pkey on flights f (cost=0.42..0.72 rows=1 width=16) (actual time=0.007..0.007 rows=1...
30                Index Cond: (flightid = d.flightid)
31              -> Hash (cost=1.42..1.42 rows=42 width=62) (actual time=0.039..0.039 rows=42 loops=3)
32                Buckets: 1024 Batches: 1 Memory Usage: 11kB
33              -> Seq Scan on airline al (cost=0.00..1.42 rows=42 width=62) (actual time=0.018..0.024 rows=42 loops=3)
34              -> Hash (cost=1.65..1.65 rows=65 width=27) (actual time=0.065..0.065 rows=65 loops=3)
35                Buckets: 1024 Batches: 1 Memory Usage: 12kB
36              -> Seq Scan on airport ap_dest (cost=0.00..1.65 rows=65 width=27) (actual time=0.036..0.044 rows=65 loops=3)
37 Planning Time: 1.365 ms
38 Execution Time: 68.028 ms

```

10. vw_flight_overview

Време на извршување на погледот според OperatingStatus, за одреден временски период.

1	Nested Loop (cost=1000.27..11324.60 rows=1 width=161) (actual t...
2	-> Nested Loop (cost=1000.00..11316.27 rows=1 width=155) (ac...
3	Join Filter: (al.airlineid = f.airlineid)
4	-> Nested Loop (cost=1000.00..11314.33 rows=1 width=10...
5	Join Filter: (a_dest.airportid = f.destinationairp...
6	-> Nested Loop (cost=1000.00..11311.87 rows=1 wi...
7	Join Filter: (a_orig.airportid = f.originair...
8	-> Gather (cost=1000.00..11309.40 rows=1 w...
9	Workers Planned: 2
10	Workers Launched: 2
11	-> Parallel Seq Scan on flights f (c...
12	Filter: ((departuretime >= '2024...
13	Rows Removed by Filter: 166505
14	-> Seq Scan on airport a_orig (cost=0.00...
15	-> Seq Scan on airport a_dest (cost=0.00..1.65 r...
16	-> Seq Scan on airline al (cost=0.00..1.42 rows=42 wid...
17	-> Index Scan using aircraft_pkey on aircraft ac (cost=0.27...
18	Index Cond: (aircraftid = f.aircraftid)
19	Planning Time: 1.129 ms
20	Execution Time: 73.183 ms

Ова е прифатливо време за апликацијата, па затоа не пристапуваме кон индексирање.