

## View 1. Head to head

Филтер кој ќе се користи за vw\_head\_to\_head ќе биде според id на двата тима. Случај на употреба е преглед на претходни средби помеѓу двата тима за еден натпревар.

Иницијалното време на извршување е 3s 123ms. Времето не е прифатливо.

```
[2026-05-03 23:20:44] pitchmap.public> select * from vw_head_to_head where team_a_id = 52 and team_b_id = 155
[2026-05-03 23:20:47] 1 row retrieved starting from 1 in 3 s 123 ms (execution: 2 s 799 ms, fetching: 324 ms)
```

| Operation                  | Params                  | Rows   | Total Cost | Startup Cost |
|----------------------------|-------------------------|--------|------------|--------------|
| Aggregate                  |                         | 1      | 540356.45  | 540356.39    |
| Aggregate                  |                         | 35226  | 538491.94  | 534237.54    |
| Full Scan (Seq Scan)       | table: event_type;      | 1      | 1.44       | 0            |
| Full Scan (Seq Scan)       | table: event_type;      | 1      | 1.44       | 0            |
| Unknown (Gather Merge)     |                         | 29356  | 537916.63  | 534234.66    |
| Aggregate                  |                         | 14678  | 533528.2   | 533234.64    |
| Sort                       |                         | 14678  | 533271.33  | 533234.64    |
| Hash Join                  |                         | 14678  | 532218.82  | 366119.73    |
| Full Scan (Seq Scan)       | table: event;           | 381522 | 165097.59  | 0            |
| Transformation (Hash)      |                         | 37501  | 365650.97  | 365650.97    |
| Full Scan (Seq Scan)       | table: event_attribute; | 37501  | 365650.97  | 0            |
| Sort                       |                         | 1      | 1864.46    | 1864.45      |
| Nested Loops (Nested Loop) |                         | 1      | 1864.44    | 1853.65      |
| Nested Loops (Nested Loop) |                         | 1      | 1859.07    | 1853.65      |
| Transformation (WindowAgg) |                         | 1      | 1853.69    | 1853.65      |
| Sort                       |                         | 1      | 1853.66    | 1853.65      |
| Merge Join                 |                         | 1      | 1853.64    | 972.97       |
| Access (CTE Scan)          | table: match_goals;     | 35226  | 704.52     | 0            |
| Sort                       |                         | 1      | 972.97     | 972.97       |
| Merge Join                 |                         | 1      | 972.96     | 92.29        |
| Access (CTE Scan)          | table: match_goals;     | 35226  | 704.52     | 0            |
| Sort                       |                         | 1      | 92.3       | 92.29        |
| Full Scan (Seq Scan)       | table: match;           | 1      | 92.28      | 0            |
| Full Scan (Seq Scan)       | table: team;            | 1      | 5.36       | 0            |
| Full Scan (Seq Scan)       | table: team;            | 1      | 5.36       | 0            |

Најскапите операции се full scan на табелите event и event\_attribute.

Прво се преуредува кверито да филтрира според полето event.event\_type\_id наместо според event\_type.type, со цел да се искористи индексот кој ќе биде поставен на Event табелата:

Before:

```
FROM Event e
```

```
JOIN Event_type et ON et.event_type_id = e.event_type_id
```

```
LEFT JOIN Event_attribute ea ON ea.event_id = e.event_id
```

```
AND ea.attribute_name = 'shot.outcome'
```

WHERE

(et.type = 'Shot' AND ea.attribute\_value = 'Goal')

OR et.type = 'Own Goal For'

After:

FROM Event e

LEFT JOIN Event\_attribute ea ON ea.event\_id = e.event\_id

AND ea.attribute\_name = 'shot.outcome'

WHERE

(e.event\_type\_id = (SELECT event\_type\_id FROM Event\_type WHERE type = 'Shot')

AND ea.attribute\_value = 'Goal')

OR e.event\_type\_id = (SELECT event\_type\_id FROM Event\_type WHERE type = 'Own Goal For')

```
[2026-05-03 23:21:55] pitchmap.public> INSERT INTO Event (match_id, team_id, player_id, event_type_id, minute, second)
VALUES (1, 1, 1, 32, 45, 30)
[2026-05-03 23:21:55] 1 row affected in 14 ms
[2026-05-03 23:22:16] pitchmap.public> UPDATE Event SET minute = 50 WHERE event_id = 1
[2026-05-03 23:22:16] 1 row affected in 5 ms
[2026-05-03 23:22:44] pitchmap.public> INSERT INTO Event_attribute (event_id, attribute_name, attribute_value)
VALUES (1, 'shot.outcome', 'Goal')
[2026-05-03 23:22:44] 1 row affected in 4 ms
[2026-05-03 23:22:55] pitchmap.public> UPDATE Event_attribute SET attribute_value = 'Saved'
WHERE event_id = 1 AND attribute_name = 'shot.outcome'
[2026-05-03 23:22:55] 1 row affected in 2 ms
```

Додаваме индекс на Event(event\_type\_id, match\_id, team\_id). По преуредувањето на погледот и користење на индексот на Event табелата времето се намалува на 738ms.

```
[2026-05-03 23:31:29] pitchmap.public> select * from vw_head_to_head where team_a_id = 52 and team_b_id = 155
[2026-05-03 23:31:30] 1 row retrieved starting from 1 in 738 ms (execution: 424 ms, fetching: 314 ms)
```

WHERE (et.type = 'Shot' AND ea.attribute\_value = 'Goal') OR et.type = 'Own Goal For' – во овој случај, едната гранка треба да дојде до event attribute, додека другата (за автогол) нема event\_attributes. Планерот мора да избере еден план и во случајов го бира да ги спои сите евенти со event\_attributes па потоа да го провери условот. Поради тоа овој услов го преуредуваме со UNION за да се оптимизира додатно и да се искористи индексот на event\_attribute табелата.

Додаваме индекс на Event\_attribute(attribute\_value, event\_id) where attribute\_name = 'shot.outcome'

Времето се намалува на 485ms што е прифатливо.

```
[2026-05-03 23:53:52] pitchmap.public> INSERT INTO Event (match_id, team_id, player_id, event_type_id, minute, second)
VALUES (1, 1, 1, 32, 45, 30)
[2026-05-03 23:53:52] 1 row affected in 3 ms
[2026-05-03 23:53:57] pitchmap.public> UPDATE Event SET minute = 50 WHERE event_id = 1
[2026-05-03 23:53:57] 1 row affected in 3 ms
[2026-05-03 23:54:02] pitchmap.public> INSERT INTO Event_attribute (event_id, attribute_name, attribute_value)
VALUES (1, 'shot.outcome', 'Goal')
```

```
[2026-05-03 23:55:06] pitchmap.public> INSERT INTO Event_attribute (event_id, attribute_name, attribute_value)
VALUES (2, 'shot.outcome', 'Goal')
[2026-05-03 23:55:06] 1 row affected in 6 ms
[2026-05-03 23:55:15] pitchmap.public> UPDATE Event_attribute SET attribute_value = 'Saved'
WHERE event_id = 1 AND attribute_name = 'shot.outcome'
[2026-05-03 23:55:15] 1 row affected in 3 ms
```

```
[2026-05-04 00:06:43] pitchmap.public> select * from vw_head_to_head where team_a_id = 52 and team_b_id = 155
[2026-05-04 00:06:43] 1 row retrieved starting from 1 in 485 ms (execution: 167 ms, fetching: 318 ms)
```

План на евалуација по индексите:

| Operation                    | Params  | Rows   | Total Cost | Startup Cost |
|------------------------------|---|--------|------------|--------------|
| Aggregate                    |   | 1      | 24548.17   | 24548.12     |
| Aggregate                    |   | 37264  | 22479.29   | 22013.49     |
| Aggregate                    |   | 372639 | 17130.27   | 1.87         |
| Full Scan (Seq Scan)         | table: event_type;                                  | 1      | 1.44       | 0            |
| Index Scan (Index Only Scan) | table: event; index: h2h_idx;                       | 466804 | 9901.41    | 0.43         |
| Aggregate                    |   | 1      | 225.22     | 225.19       |
| Full Scan (Seq Scan)         | table: event_type;                                  | 1      | 1.44       | 0            |
| Sort                         |   | 1      | 223.76     | 223.76       |
| Nested Loops (Nested Lo      |   | 1      | 223.75     | 0.85         |
| Index Scan (Index Only       | table: event_attribute; index: idx_event_attr_sh... | 24     | 20.83      | 0.42         |
| Index Scan                   | table: event; index: event_pkey;                    | 1      | 8.46       | 0.43         |
| Sort                         |   | 1      | 2068.84    | 2068.83      |
| Nested Loops (Nested Loop)   |   | 1      | 2068.82    | 2058.03      |
| Nested Loops (Nested Loop)   |   | 1      | 2063.45    | 2058.03      |
| Transformation (WindowAgg    |   | 1      | 2058.07    | 2058.03      |
| Sort                         |   | 1      | 2058.04    | 2058.03      |
| Hash Join                    |   | 1      | 2058.02    | 1033.25      |
| Access (CTE Scz              | table: match_goals_final;                           | 37264  | 745.28     | 0            |
| Transformation (Ha:          |   | 1      | 1033.23    | 1033.23      |
| Hash Join                    |   | 1      | 1033.23    | 92.31        |
| Access (C'                   | table: match_goals_final;                           | 37264  | 745.28     | 0            |
| Transformati                 |   | 1      | 92.3       | 92.3         |
| Full Scz                     | table: match;                                       | 1      | 92.3       | 0            |
| Full Scan (Seq Scan)         | table: team;  | 1      | 5.36       | 0            |
| Full Scan (Seq Scan)         | table: team;  | 1      | 5.36       | 0            |

## View 2. Season overview

Филтер кој ќе се користи за vw\_season\_overview ќе биде според id на сезоната и дивизијата. Случај на употреба е преглед на статистики за една сезона во една лига (дивизија).

Иницијалното време на извршување е 4s 792ms. Времето не е прифатливо.

```
[2026-05-05 19:03:31] pitchmap.public> select * from vw_season_overview where season_id = 44 and division_id = 10
[2026-05-05 19:03:36] 1 row retrieved starting from 1 in 4 s 792 ms (execution: 4 s 474 ms, fetching: 318 ms)
```

```
[2026-05-04 00:25:37] pitchmap.public> INSERT INTO Ticket(match_id, is_scanned, seat_id, price) VALUES (1, true, 120, 15)
[2026-05-04 00:25:37] 1 row affected in 13 ms
```

|   | 37264   | 630208.91 | 567787.21 |
|---|---------|-----------|-----------|
| Aggregate   |         |           |           |
| Transformation (Merge Append)                     | 372640  | 626948.31 | 567787.21 |
| Aggregate   | 372639  | 228637.6  | 173202.93 |
| Full Scan (Seq Scan) table: event_type;           | 1       | 1.44      | 0         |
| Unknown (Gather Merge)                            | 389004  | 221992.25 | 173201.49 |
| Aggregate   | 194502  | 176091.51 | 172201.47 |
| Sort  | 194502  | 172687.72 | 172201.47 |
| Full Scan (Seq table: event;                      | 194502  | 152455.03 | 0         |
| Aggregate   | 1       | 394584.3  | 394584.27 |
| Full Scan (Seq Scan) table: event_type;           | 1       | 1.44      | 0         |
| Sort  | 1       | 394582.84 | 394582.84 |
| Unknown (Gather)                                  | 1       | 394582.83 | 1000.43   |
| Nested Loops (Nested Loop)                        | 1       | 393582.73 | 0.43      |
| Full Scan (Seq table: event_attribute;            | 10      | 393498.18 | 0         |
| Index Scan table: event; index: event_pkey;       | 1       | 8.46      | 0.43      |
| Nested Loops (Nested Loop)                        | 380     | 179426.14 | 179264.63 |
| Nested Loops (Nested Loop)                        | 1       | 10.09     | 0.14      |
| Full Scan (Seq Scan) table: season;               | 1       | 1.92      | 0         |
| Index Scan table: division; index: division_pkey; | 1       | 8.16      | 0.14      |
| Hash Join   | 380     | 179412.26 | 179264.48 |
| Aggregate   | 551     | 177432.66 | 177293.06 |
| Unknown (Gather Merge)                            | 1102    | 177421.64 | 177293.06 |
| Sort  | 551     | 176294.42 | 176293.04 |
| Aggregate   | 551     | 176267.95 | 176262.44 |
| Full Scan (Seq table: ticket;                     | 3513205 | 158696.42 | 0         |
| Transformation (Hash)                             | 380     | 1966.67   | 1966.67   |

Се прави истото преуредување за агрегирање на головите како и во првиот поглед за да се искористат истите индекси на Event и Event\_attribute табелите. Дополнително се додава индекс на Ticket(match\_id) where is\_scanned = true

По реискористување на индексите од првиот поглед и додавање на новиот, времето се намалува на 1s 64ms што е прифатливо земајќи в предвид дека се агрегираат податоци помеѓу 3 табели со по ~15 милиони редици:

```
[2026-05-05 19:12:48] pitchmap.public> select * from vw_season_overview where season_id = 44 and division_id = 10
[2026-05-05 19:12:49] 1 row retrieved starting from 1 in 1 s 64 ms (execution: 741 ms, fetching: 323 ms)
```

```
[2026-05-04 01:02:50] pitchmap.public> INSERT INTO Ticket(match_id, is_scanned, seat_id, price) VALUES (44, true, 23, 15)
[2026-05-04 01:02:50] 1 row affected in 4 ms
```

```
[2026-05-04 01:04:10] pitchmap.public> UPDATE Ticket SET is_scanned = true WHERE match_id = 23 and seat_id = 120
[2026-05-04 01:04:10] completed in 2 ms
```

План на евалуација по оптимизациите:

| Operation                  | Params  | Rows    | Total Cost | Startup Cost |
|----------------------------|---|---------|------------|--------------|
| Aggregate                  |   | 1       | 148698.28  | 25531.04     |
| Aggregate                  |   | 37264   | 22479.29   | 22013.49     |
| Aggregate                  |   | 372639  | 17130.27   | 1.87         |
| Full Scan (Seq Scan)       | table: event_type;                                  | 1       | 1.44       | 0            |
| Index Scan (Index Only)    | table: event; index: h2h_idx;                       | 466804  | 9901.41    | 0.43         |
| Aggregate                  |   | 1       | 225.22     | 225.19       |
| Full Scan (Seq Scan)       | table: event_type;                                  | 1       | 1.44       | 0            |
| Sort                       |   | 1       | 223.76     | 223.76       |
| Nested Loops (Nested Loop) |   | 1       | 223.75     | 0.85         |
| Index Scan (Index Only)    | table: event_attribute; index: idx_event_attr_shot; | 24      | 20.83      | 0.42         |
| Index Scan                 | table: event; index: event_pkey;                    | 1       | 8.46       | 0.43         |
| Nested Loops (Nested Loop) |   | 380     | 126214.22  | 3051.75      |
| Nested Loops (Nested Loop) |   | 1       | 10.09      | 0.14         |
| Full Scan (Seq Scan)       | table: season;                                      | 1       | 1.92       | 0            |
| Index Scan                 | table: division; index: division_pkey;              | 1       | 8.16       | 0.14         |
| Hash Join                  |   | 380     | 126200.33  | 3051.6       |
| Aggregate                  |   | 551     | 124141.01  | 1000.46      |
| Unknown (Gather Merge)     |   | 1102    | 124129.99  | 1000.46      |
| Aggregate                  |   | 551     | 123002.77  | 0.43         |
| Index Scan (Index Only)    | table: ticket; index: idx_ticket_match_scanned;     | 3513205 | 105431.23  | 0.43         |
| Transformation (Hash)      |   | 380     | 2046.4     | 2046.4       |
| Hash Join                  |   | 380     | 2046.4     | 1018.63      |
| Access (CTE Scan)          | table: match_goals;                                 | 37264   | 745.28     | 0            |
| Transformation (Hash)      |   | 380     | 1012.93    | 1012.93      |
| Hash Join                  |   | 380     | 1012.93    | 72.01        |

### View 3. Stadium utilization

Филтер кој ќе се користи за vw\_utilization ќе биде според id на стадионот и сезоната. Случај на употреба е преглед на статистики за користење на стадион во една сезона.

Иницијалното време на извршување е 1s 571ms. Времето не е прифатливо.

```
[2026-05-05 19:17:47] pitchmap.public> select * from vw_utilization where stadium_id = 1 and season_id = 2
[2026-05-05 19:17:49] 1 row retrieved starting from 1 in 1 s 571 ms (execution: 1 s 237 ms, fetching: 334 ms)
```

| Operation                            | Params                                 | Rows     | Total Cost | Startup Cost |
|--------------------------------------|--|----------|------------|--------------|
| ↳ Select                             |  |          |            |              |
| ↳ Nested Loops (Nested Loop)         |  | 1        | 286151.1   | 75.16        |
| ↳ Nested Loops (Nested Loop)         |  | 1        | 91.95      | 0.14         |
| ↳ Nested Loops (Nested Loop)         |  | 1        | 83.7       | 0            |
| ↳ Nested Loops (Nested Loop)         |  | 1        | 81.92      | 0            |
| ↳ Full Scan (Seq Sc table: stadium;  |  | 1        | 6.91       | 0            |
| ↳ Aggregate                          |  | 1        | 74.99      | 0            |
| ↳ Full Scan (Seq table: match;       |  | 3        | 74.97      | 0            |
| ↳ Full Scan (Seq Scan table: season; |  | 1        | 1.76       | 0            |
| ↳ Index Scan                         | table: division; index: division_pkey; | 1        | 8.16       | 0.14         |
| ↳ Aggregate                          |  | 1        | 286059.09  | 75.01        |
| ↳ Hash Join                          |  | 13014    | 285961.47  | 75.01        |
| ↳ Full Scan (Seq Scan table: ticket; |  | 15031540 | 246380.4   | 0            |
| ↳ Transformation (Hash)              |  | 3        | 74.97      | 74.97        |
| ↳ Full Scan (Seq Sc table: match;    |  | 3        | 74.97      | 0            |

Од планот на евалуација гледаме дека најскапата операција е full scan на Ticket табелата. Додаваме индекс на Ticket(match\_id)

Оптимизираното време на извршување е 433ms што е прифатливо.

```
[2026-05-05 19:22:25] pitchmap.public> select * from vw_utilization where stadium_id = 1 and season_id = 2
[2026-05-05 19:22:25] 1 row retrieved starting from 1 in 433 ms (execution: 91 ms, fetching: 342 ms)
```

```
[2026-05-05 19:25:05] pitchmap.public> insert into Ticket(match_id, is_scanned, seat_id, price) values (44, true, 23, 15)
[2026-05-05 19:25:05] 1 row affected in 12 ms
```

```
[2026-05-05 19:26:40] pitchmap.public> update Ticket set is_scanned = false where match_id = 44 and seat_id = 23 and price = 15
[2026-05-05 19:26:40] 1 row affected in 3 ms
```

План на евалуација по додавање индекси:

| Operation  | Params | Rows  | Total Cost | Startup Cost |
|--|--------|-------|------------|--------------|
| ↳ Select   |        |       |            |              |
| ↳ Nested Loops (Nested Loop)                         |        | 1     | 4906.08    | 0.43         |
| ↳ Nested Loops (Nested Loop)                         |        | 1     | 4892.22    | 0.43         |
| ↳ Nested Loops (Nested Loop)                         |        | 1     | 83.7       | 0            |
| ↳ Nested Loops (Nested Loop)                         |        | 1     | 81.92      | 0            |
| ↳ Full Scan (Seq Scan) table: stadium;               |        | 1     | 6.91       | 0            |
| ↳ Aggregate  |        | 1     | 74.99      | 0            |
| ↳ Full Scan (Seq Scan) table: match;                 |        | 3     | 74.97      | 0            |
| ↳ Full Scan (Seq Scan) table: season;                |        | 1     | 1.76       | 0            |
| ↳ Aggregate  |        | 1     | 4808.5     | 0.43         |
| ↳ Nested Loops (Nested Loop)                         |        | 13014 | 4710.88    | 0.43         |
| ↳ Full Scan (Seq Scan) table: match;                 |        | 3     | 74.97      | 0            |
| ↳ Index Scan table: ticket; index: idx_ticket_match; |        | 27280 | 1272.5     | 0.43         |
| ↳ Full Scan (Seq Scan) table: division;              |        | 170   | 11.7       | 0            |

#### View 4. vw\_match\_summary

Филтер кој ќе се користи за vw\_match\_summary ќе биде според match\_id или season\_id. Случај на употреба е преглед на резултати и статистики за натпревари — голови, картони, судија и стадион.

Иницијалното време на извршување е 8s 751ms. Времето не е прифатливо.

|                   |                 |                       |         |          |  |
|-------------------|-----------------|-----------------------|---------|----------|--|
| Hash Join         |                 | 348099.01 - 482153.94 | 3464    | 6084.255 | (m.referee_id = r.referee_id)                                  |
| Merge Join        |                 | 348084.43 - 482112.88 | 3464    | 6080.244 |  |
| Merge Join        |                 | 347819.25 - 481748.13 | 3464    | 6073.480 |  |
| Merge Join        |                 | 346810.23 - 348848.46 | 3464    | 1144.628 |  |
| Sort              |                 | 348.13 - 356.79       | 3464    | 74.893   |  |
| Hash Join         |                 | 40.23 - 144.48        | 3464    | 72.460   | (m.stadium_id = st.stadium_id)                                 |
| Hash Join         |                 | 30.19 - 125.22        | 3464    | 4.150    | (m.away_team_id = at.team_id)                                  |
| Hash Join         |                 | 23.19 - 108.99        | 3464    | 3.222    | (m.home_team_id = ht.team_id)                                  |
| Hash Join         |                 | 16.20 - 92.76         | 3464    | 2.355    | (s.division_id = d.division_id)                                |
| Hash              |                 | 2.37 - 69.62          | 3464    | 1.533    | (m.season_id = s.season_id)                                    |
| Hash              |                 | 11.70 - 11.70         | 21      | 0.041    |  |
| Hash              |                 | 4.22 - 4.22           | 269     | 0.103    |  |
| Seq Scan          | team            | 0.00 - 4.22           | 269     | 0.048    |  |
| Hash              |                 | 4.22 - 4.22           | 269     | 0.123    |  |
| Seq Scan          | team            | 0.00 - 4.22           | 269     | 0.064    |  |
| Hash              |                 | 6.13 - 6.13           | 313     | 67.458   |  |
| Seq Scan          | stadium         | 0.00 - 6.13           | 313     | 67.340   |  |
| Aggregate         |                 | 346462.09 - 348405.14 | 3290    | 1066.489 |  |
| Gather Merge      |                 | 346462.09 - 348232.09 | 4139    | 1062.824 |  |
| Aggregate         |                 | 345462.03 - 345583.08 | 828     | 1019.497 |  |
| Sort              |                 | 345462.03 - 345474.38 | 2822    | 1018.316 |  |
| Nested Loop       |                 | 0.43 - 345159.03      | 2822    | 1015.567 |  |
| Parallel Seq Scan | event_attribute | 0.00 - 309902.80      | 2822    | 952.761  | ((ea.attribute_name)::text = any ('{foul_committed.card,ba...  |
| Index Scan        | event           | 0.43 - 7.14           | 1       | 0.020    | (e.event_id = ea.event_id)                                     |
| Aggregate         |                 | 1009.02 - 132813.07   | 3464    | 4925.678 |  |
| Nested Loop       |                 | 1009.02 - 132735.13   | 10038   | 4921.141 |  |
| Index Scan        | match           | 0.28 - 126.24         | 3464    | 5.168    |  |
| Materialize       |                 | 1008.74 - 132556.93   | 9790    | 0.774    |  |
| Gather            |                 | 1008.74 - 132556.92   | 9790    | 1099.884 |  |
| Nested Loop       |                 | 8.74 - 131556.82      | 1938    | 1063.004 |  |
| Hash Join         |                 | 8.18 - 127619.66      | 17605   | 602.537  | (e_1.event_type_id = et.event_type_id)                         |
| Parallel Seq Scan | event           | 0.00 - 119584.15      | 2427379 | 268.561  |  |
| Hash              |                 | 8.17 - 8.17           | 1       | 11.852   |  |
| Index Scan        | event_attribute | 0.56 - 0.70           | 0       | 0.025    | ((ea_1.event_id = e_1.event_id)and ((ea_1.attribute_name)::... |
| Unique            |                 | 265.19 - 282.27       | 3281    | 3.956    |  |
| Sort              |                 | 265.19 - 273.73       | 3416    | 2.493    |  |
| Seq Scan          | referee_match   | 0.00 - 64.70          | 3416    | 0.715    |  |
| Hash              |                 | 8.70 - 8.70           | 470     | 0.206    |  |
| Seq Scan          | referee         | 0.00 - 8.70           | 470     | 0.113    |  |

Најскапите операции се Parallel Seq Scan на табелата event\_attribute — скенира ~26 милиони редици при барање картони, и Parallel Seq Scan на табелата event — скенира ~12 милиони редици при барање голови.

Додаваме индекс на Event\_attribute(attribute\_value, event\_id) where attribute\_name = 'shot.outcome' за да се забрза филтрирањето на голови:

```
CREATE INDEX idx_event_attr_shot
ON public.event_attribute USING btree (attribute_value, event_id)
WHERE ((attribute_name)::text = 'shot.outcome'::text);
```

Додаваме индекс на Event\_attribute(attribute\_value, event\_id) where attribute\_name IN ('foul\_committed.card', 'bad\_behaviour.card') за да се елиминира Seq Scan при броење картони. Планерот претходно мораше да скенира ~26 милиони редици за да ги најде

записите за картони. Со парцијалниот индекс директно се пристапува само до релевантните редици:

```
CREATE INDEX idx_event_attr_cards
ON public.event_attribute USING btree (attribute_value, event_id)
WHERE ((attribute_name)::text = ANY ({'foul_committed.card,bad_behaviour.card'}));
```

Додаваме индекс на Event(event\_type\_id, match\_id, team\_id) за да се забрза спојувањето на Event табелата:

```
CREATE INDEX h2h_idx
ON public.event USING btree (event_type_id, match_id, team_id);
```

По додавањето на индексите времето се намалува на 373ms што е прифатливо (~23x побрзо).

План на евалуација по индексите:

|                           |                     |                       |       |         |                                 |
|---------------------------|---------------------|-----------------------|-------|---------|---------------------------------|
| Hash Join                 |                     | 108003.98 - 109729.88 | 3464  | 251.315 | (m.referee_id = r.referee_id)   |
| Merge Join                |                     | 107989.40 - 109688.81 | 3464  | 249.991 |                                 |
| Merge Join                |                     | 107724.22 - 109324.06 | 3464  | 246.633 |                                 |
| Sort                      |                     | 1086.22 - 1094.88     | 3464  | 144.064 |                                 |
| Hash Join                 |                     | 757.58 - 882.57       | 3464  | 141.201 | (m.stadium_id = st.stadium_id)  |
| Hash Join                 |                     | 747.53 - 863.31       | 3464  | 74.096  | (m.away_team_id = at.team_id)   |
| Hash Join                 |                     | 740.54 - 847.08       | 3464  | 73.149  | (m.home_team_id = ht.team_id)   |
| Hash Join                 |                     | 733.54 - 830.85       | 3464  | 72.212  | (s.division_id = d.division_id) |
| Hash Join                 |                     | 719.72 - 807.71       | 3464  | 71.366  | (m.season_id = s.season_id)     |
| Hash Join                 |                     | 717.35 - 795.73       | 3464  | 70.487  | (m_1.match_id = m.match_id)     |
| Hash                      |                     | 1.61 - 1.61           | 75    | 0.051   |                                 |
| Hash                      |                     | 11.70 - 11.70         | 21    | 0.031   |                                 |
| Seq Scan                  | division            | 0.00 - 11.70          | 21    | 0.015   |                                 |
| Hash                      |                     | 4.22 - 4.22           | 269   | 0.105   |                                 |
| Seq Scan                  | team                | 0.00 - 4.22           | 269   | 0.049   |                                 |
| Hash                      |                     | 4.22 - 4.22           | 269   | 0.124   |                                 |
| Seq Scan                  | team                | 0.00 - 4.22           | 269   | 0.068   |                                 |
| Hash                      |                     | 6.13 - 6.13           | 313   | 66.090  |                                 |
| Seq Scan                  | stadium             | 0.00 - 6.13           | 313   | 65.965  |                                 |
| Aggregate                 |                     | 106638.00 - 108142.65 | 3290  | 100.579 |                                 |
| Gather Merge              |                     | 106638.00 - 108004.21 | 5866  | 98.658  |                                 |
| Aggregate                 |                     | 105637.96 - 105784.16 | 1466  | 56.273  |                                 |
| Sort                      |                     | 105637.96 - 105653.90 | 3528  | 54.959  |                                 |
| Nested Loop               |                     | 304.22 - 105234.97    | 3528  | 53.604  |                                 |
| Parallel Bitmap Heap Scan | event_attribute     | 303.78 - 59717.77     | 3528  | 15.022  |                                 |
| Bitmap Index Scan         | idx_event_attr_c... | 0.00 - 298.84         | 14111 | 6.394   |                                 |
| Index Scan                | event               | 0.43 - 7.14           | 1     | 0.007   | (e_1.event_id = ea_1.event_id)  |
| Unique                    |                     | 265.19 - 282.27       | 3281  | 2.110   |                                 |
| Sort                      |                     | 265.19 - 273.73       | 3416  | 1.564   |                                 |
| Seq Scan                  | referee_match       | 0.00 - 64.70          | 3416  | 0.624   |                                 |
| Hash                      |                     | 8.70 - 8.70           | 470   | 0.184   |                                 |
| Seq Scan                  | referee             | 0.00 - 8.70           | 470   | 0.092   |                                 |

## View 5. vw\_team\_season\_table

Филтер кој ќе се користи за vw\_team\_season\_table ќе биде според id на сезоната и дивизијата. Случај на употреба е преглед на табела за лига — победи, реми, поразии, голови и бодови за секој тим во сезона.

Иницијалното време на извршување е 6s 250ms. Времето не е прифатливо.

| Operator        | Estimated Rows        | Estimated Cost | Actual Rows | Actual Cost | Comments                                 |
|-----------------|-----------------------|----------------|-------------|-------------|--|
| Aggregate       | 133895.43 - 133948.25 | 858            | 5098.257    |             |  |
| Subquery Scan   | 1012.30 - 133571.09   | 3464           | 5018.011    |             |  |
| Nested Loop     | 1012.30 - 133536.45   | 3464           | 5016.860    |             |  |
| Hash Join       | 1012.14 - 133437.87   | 3464           | 5011.089    |             | (m.season_id = s_1.season_id)            |
| Merge Join      | 1009.77 - 133425.89   | 3464           | 5008.229    |             |  |
| Nested Loop     | 0.75 - 526.23         | 3464           | 21.685      |             |  |
| Nested Loop     | 0.59 - 386.21         | 3464           | 16.292      |             |  |
| Nested Loop     | 0.44 - 256.23         | 3464           | 10.175      |             |  |
| Index Scan      | 0.28 - 126.24         | 3464           | 3.467       |             | match                                    |
| Memoize         | 0.15 - 0.18           | 1              | 0.001       |             |  |
| Index Only Scan | 0.14 - 0.17           | 1              | 0.005       |             | (ht.team_id = m.home_team_id)            |
| Memoize         | 0.15 - 0.18           | 1              | 0.001       |             |  |
| Index Only Scan | 0.14 - 0.17           | 1              | 0.004       |             | (at.team_id = m.away_team_id)            |
| Memoize         | 0.16 - 0.18           | 1              | 0.001       |             |  |
| Index Only Scan | 0.15 - 0.17           | 1              | 0.003       |             | (st.stadium_id = m.stadium_id)           |
| Aggregate       | 1009.02 - 132813.07   | 3464           | 4983.535    |             |  |
| Nested Loop     | 1009.02 - 132735.13   | 10038          | 4977.726    |             |  |
| Index Scan      | 0.28 - 126.24         | 3464           | 3.512       |             | match                                    |
| Materialize     | 1008.74 - 132556.93   | 9790           | 0.781       |             |  |
| Gather          | 1008.74 - 132556.92   | 9790           | 1102.287    |             |  |
| Nested Loop     | 8.74 - 131556.82      | 1958           | 1061.759    |             |  |
| Hash            | 1.61 - 1.61           | 75             | 0.060       |             |  |
| Seq Scan        | 0.00 - 1.61           | 75             | 0.038       |             | season                                   |
| Memoize         | 0.15 - 0.96           | 1              | 0.001       |             |  |
| Index Only Scan | 0.14 - 0.95           | 1              | 0.007       |             | (d_1.division_id = s_1.division_id)      |
| Hash Join       | 23.19 - 260.95        | 6928           | 5089.770    |             | (match_results.home_team_id = t.team_id) |
| Hash Join       | 16.20 - 248.29        | 6928           | 5030.094    |             | (s.division_id = d.division_id)          |
| Hash Join       | 2.37 - 228.79         | 6928           | 5027.435    |             | (match_results.season_id = s.season_id)  |
| Append          | 0.00 - 207.84         | 6928           | 5024.055    |             |  |
| CTE Scan        | 0.00 - 86.60          | 3464           | 5022.169    |             | match_results                            |
| CTE Scan        | 0.00 - 86.60          | 3464           | 0.796       |             | match_results                            |
| Hash            | 1.61 - 1.61           | 75             | 0.061       |             |  |
| Seq Scan        | 0.00 - 1.61           | 75             | 0.035       |             | season                                   |
| Hash            | 11.70 - 11.70         | 21             | 0.034       |             |  |
| Seq Scan        | 0.00 - 11.70          | 21             | 0.021       |             | division                                 |
| Hash            | 4.22 - 4.22           | 269            | 56.479      |             |  |
| Seq Scan        | 0.00 - 4.22           | 269            | 56.376      |             | team                                     |

Овој поглед внатрешно го повикува vw\_match\_summary преку CTE match\_results, па ги наследува истите проблеми — Parallel Seq Scan на event и event\_attribute табелите.

Се реискористуваат истите индекси додадени за vw\_match\_summary: idx\_event\_attr\_shot за голови, idx\_event\_attr\_cards за картони и h2h\_idx за event табелата. Бидејќи vw\_team\_season\_table целосно се базира на vw\_match\_summary, нема потреба од додатни индекси.

По реискористување на индексите од првиот поглед времето се намалува на 164ms што е прифатливо (~38x побрзо).

## План на евалуација по оптимизациите:

|               |  |               |                   |       |        |  |
|---------------|--|---------------|-------------------|-------|--------|--|
| Aggregate     |  |               | 1241.55 - 1294.37 | 858   | 90.229 |  |
| Subquery Scan |  |               | 757.58 - 917.21   | 3464  | 76.250 |  |
| Hash Join     |  |               | 757.58 - 882.57   | 3464  | 75.730 | (m.stadium_id = st.stadium_id)           |
| Hash Join     |  |               | 747.53 - 863.31   | 3464  | 74.177 | (m.away_team_id = at.team_id)            |
| Hash Join     |  |               | 740.54 - 847.08   | 3464  | 73.156 | (m.home_team_id = ht.team_id)            |
| Hash Join     |  |               | 733.54 - 830.85   | 3464  | 72.102 | (s_1.division_id = d_1.division_id)      |
| Hash Join     |  |               | 719.72 - 807.71   | 3464  | 71.200 | (m.season_id = s_1.season_id)            |
| Hash Join     |  |               | 717.35 - 795.73   | 3464  | 70.180 | (m_1.match_id = m.match_id)              |
| Aggregate     |  |               | 616.41 - 651.05   | 3464  | 67.584 |  |
| Hash Join     |  |               | 502.47 - 573.11   | 10038 | 63.903 | (m_1.match_id = e.match_id)              |
| Hash          |  |               | 57.64 - 57.64     | 3464  | 1.238  |  |
| Seq Scan      |  | match         | 0.00 - 57.64      | 3464  | 0.592  |  |
| Hash          |  |               | 1.61 - 1.61       | 75    | 0.032  |  |
| Seq Scan      |  | season        | 0.00 - 1.61       | 75    | 0.018  |  |
| Hash          |  |               | 11.70 - 11.70     | 21    | 0.017  |  |
| Seq Scan      |  | division      | 0.00 - 11.70      | 21    | 0.013  |  |
| Hash          |  |               | 4.22 - 4.22       | 269   | 0.076  |  |
| Seq Scan      |  | team          | 0.00 - 4.22       | 269   | 0.039  |  |
| Hash          |  |               | 4.22 - 4.22       | 269   | 0.076  |  |
| Seq Scan      |  | team          | 0.00 - 4.22       | 269   | 0.040  |  |
| Hash          |  |               | 6.13 - 6.13       | 313   | 0.100  |  |
| Seq Scan      |  | stadium       | 0.00 - 6.13       | 313   | 0.055  |  |
| Hash Join     |  |               | 23.19 - 260.95    | 6928  | 85.537 | (match_results.home_team_id = t.team_id) |
| Hash Join     |  |               | 16.20 - 248.29    | 6928  | 83.452 | (s.division_id = d.division_id)          |
| Hash Join     |  |               | 2.37 - 228.79     | 6928  | 81.661 | (match_results.season_id = s.season_id)  |
| Append        |  |               | 0.00 - 207.84     | 6928  | 79.709 |  |
| CTE Scan      |  | match_results | 0.00 - 86.60      | 3464  | 78.267 |  |
| CTE Scan      |  | match_results | 0.00 - 86.60      | 3464  | 0.684  |  |
| Hash          |  |               | 1.61 - 1.61       | 75    | 0.044  |  |
| Seq Scan      |  | season        | 0.00 - 1.61       | 75    | 0.028  |  |
| Hash          |  |               | 11.70 - 11.70     | 21    | 0.018  |  |
| Seq Scan      |  | division      | 0.00 - 11.70      | 21    | 0.012  |  |
| Hash          |  |               | 4.22 - 4.22       | 269   | 0.107  |  |
| Seq Scan      |  | team          | 0.00 - 4.22       | 269   | 0.056  |  |

## View 6. vw\_player\_season\_stats

Филтер кој ќе се користи за vw\_player\_season\_stats ќе биде според id на играчот и сезоната. Случај на употреба е преглед на статистики за играч во сезона — настапи, голови, асистенции, картони и проценка на изиграни минути.

Иницијалното време на извршување е 29s. Времето не е прифатливо.

|                   |                 |                       |        |          |   |
|-------------------|-----------------|-----------------------|--------|----------|---|
| Hash Join         |                 | 936643.68 - 950791.27 | 19738  | 8861.491 | (lineup_per_player_season.team_id = t.team_id)              |
| Merge Join        |                 | 936636.68 - 950782.23 | 19738  | 7724.873 |   |
| Merge Join        |                 | 609027.46 - 623171.95 | 19738  | 6727.709 |   |
| Merge Join        |                 | 281400.77 - 295542.34 | 19738  | 5488.316 |   |
| Merge Join        |                 | 148843.54 - 162984.06 | 19738  | 1503.213 |   |
| Nested Loop       |                 | 16286.31 - 30425.78   | 19738  | 235.589  |   |
| Nested Loop       |                 | 16286.03 - 30230.78   | 19738  | 194.680  |   |
| Subquery Scan     |                 | 16285.89 - 30115.09   | 19738  | 161.025  | (lineup_per_player_season.m = 1)                            |
| WindowAgg         |                 | 16285.89 - 28468.75   | 19738  | 158.823  |   |
| Incremental Sort  |                 | 16285.89 - 25834.61   | 19943  | 146.251  |   |
| Index Scan        | season          | 0.14 - 0.18           | 1      | 0.001    | (s.season_id = lineup_per_player_season.season_id)          |
| Index Scan        | player          | 0.29 - 0.87           | 1      | 0.002    | (p.player_id = lineup_per_player_season.player_id)          |
| Aggregate         |                 | 132557.23 - 132557.25 | 3611   | 1262.652 |   |
| Sort              |                 | 132557.23 - 132557.24 | 9790   | 1259.630 |   |
| Nested Loop       |                 | 1009.02 - 132557.22   | 9790   | 1252.546 |   |
| Gather            |                 | 1008.74 - 132556.92   | 9790   | 1224.087 |   |
| Nested Loop       |                 | 8.74 - 131556.82      | 1958   | 1206.009 |   |
| Index Scan        | match           | 0.28 - 0.30           | 1      | 0.002    | (m_1.match_id = e.match_id)                                 |
| Aggregate         |                 | 132557.23 - 132557.25 | 2952   | 3980.518 |   |
| Sort              |                 | 132557.23 - 132557.23 | 6354   | 3978.355 |   |
| Nested Loop       |                 | 1009.02 - 132557.22   | 6354   | 3974.016 |   |
| Gather            |                 | 1008.74 - 132556.92   | 6354   | 3955.234 |   |
| Nested Loop       |                 | 8.74 - 131556.82      | 1271   | 3928.862 |   |
| Hash Join         |                 | 8.18 - 127619.66      | 677552 | 853.364  | (e_1.event_type_id = et_1.event_type_id)                    |
| Index Scan        | event_attribute | 0.56 - 0.70           | 0      | 0.004    | ((e_1.event_id = e_1.event_id)and ((e_1.attribute_name)=... |
| Index Scan        | match           | 0.28 - 0.30           | 1      | 0.002    | (m_2.match_id = e_1.match_id)                               |
| Aggregate         |                 | 327626.69 - 327628.40 | 5864   | 1234.170 |   |
| Gather Merge      |                 | 327626.69 - 327628.18 | 9982   | 1231.251 |   |
| Aggregate         |                 | 326626.63 - 326626.69 | 1996   | 1187.289 |   |
| Sort              |                 | 326626.63 - 326626.64 | 2754   | 1186.347 |   |
| Nested Loop       |                 | 0.72 - 326626.61      | 2754   | 1183.208 |   |
| Nested Loop       |                 | 0.43 - 326625.72      | 2754   | 1167.090 |   |
| Parallel Seq Scan | event_attribute | 0.00 - 326600.36      | 2754   | 1073.504 | ((e_2.attribute_name)=text = any ('{foul_committed.card...  |
| Index Scan        | event           | 0.43 - 8.45           | 1      | 0.032    | (e_2.event_id = ea_2.event_id)                              |
| Index Scan        | match           | 0.28 - 0.30           | 1      | 0.004    | (m_3.match_id = e_2.match_id)                               |
| Aggregate         |                 | 327609.22 - 327609.24 | 329    | 993.176  |   |
| Sort              |                 | 327609.22 - 327609.23 | 342    | 992.820  |   |
| Nested Loop       |                 | 1000.71 - 327609.21   | 342    | 992.418  |   |
| Nested Loop       |                 | 1000.43 - 327608.91   | 342    | 990.669  |   |
| Gather            |                 | 1000.00 - 327600.46   | 342    | 983.598  |   |
| Parallel Seq Scan | event_attribute | 0.00 - 326600.36      | 68     | 949.125  | ((e_3.attribute_name)=text = any ('{foul_committed.card...  |
| Index Scan        | event           | 0.43 - 8.45           | 1      | 0.019    | (e_3.event_id = ea_3.event_id)                              |
| Index Scan        | match           | 0.28 - 0.30           | 1      | 0.004    | (m_4.match_id = e_3.match_id)                               |

Од планот на евалуација гледаме дека најскапите операции се Parallel Seq Scan на event табелата за Pass тип (асистенции) — скенира ~12 милиони редици × 5 паралелни работници, Parallel Seq Scan на event\_attribute × 2 пати за жолти и црвени картони, и Seq Scan на lineup табелата со 131,707 редици без индекс со external merge sort на диск (2968kB).

Се реискористуваат индексите idx\_event\_attr\_shot, idx\_event\_attr\_cards и h2h\_idx од претходните погледи. Дополнително се додава индекс на Event\_attribute(attribute\_value, event\_id) where attribute\_name = 'pass.goal\_assist' за да го замениме Seq Scan за асистенции со Index Only Scan:

```
CREATE INDEX idx_event_attr_assist
ON public.event_attribute USING btree (attribute_value, event_id)
WHERE ((attribute_name)::text = 'pass.goal_assist'::text);
```

Додаваме индекс на Lineup(player\_id, match\_id, team\_id, is\_starter) за да го елиминираме external merge sort на диск при групирање на настани по играч и сезона:

```
CREATE INDEX idx_lineup_player_match
```

```
ON public.lineup USING btree (player_id, match_id, team_id, is_starter);
```

По реискористувањето на постоечките индекси и додавањето на новите, времето се намалува на 324ms што е прифатливо (~89x побрзо).

План на евалуација по додавање индекси:

|                  |                 |  |                    |       |         |  |
|------------------|-----------------|--|--------------------|-------|---------|--|
| Hash Join        |                 |  | 5507.88 - 34093.28 | 19738 | 602.726 | (lineup_per_player_season.team_id = t.team_id)     |
| Merge Join       |                 |  | 5500.89 - 34084.24 | 19738 | 593.635 |  |
| Merge Join       |                 |  | 5483.83 - 34066.12 | 19738 | 585.732 |  |
| Merge Join       |                 |  | 5345.05 - 33925.88 | 19738 | 467.642 |  |
| Merge Join       |                 |  | 505.73 - 29085.51  | 19738 | 406.008 |  |
| Nested Loop      |                 |  | 2.97 - 28581.69    | 19738 | 312.221 |  |
| Nested Loop      |                 |  | 2.69 - 28386.69    | 19738 | 272.369 |  |
| Subquery Scan    |                 |  | 2.55 - 28271.00    | 19738 | 238.255 | (lineup_per_player_season.n = 1)                   |
| WindowAgg        |                 |  | 2.55 - 26624.66    | 19738 | 235.448 |  |
| Incremental Sort |                 |  | 2.55 - 23990.52    | 19943 | 220.209 |  |
| Index Scan       | season          |  | 0.14 - 0.18        | 1     | 0.001   | (s.season_id = lineup_per_player_season.season_id) |
| Index Scan       | player          |  | 0.29 - 0.97        | 1     | 0.002   | (p.player_id = lineup_per_player_season.player_id) |
| Aggregate        |                 |  | 502.76 - 502.78    | 3611  | 88.024  |  |
| Sort             |                 |  | 502.76 - 502.77    | 9790  | 84.869  |  |
| Nested Loop      |                 |  | 1.29 - 502.75      | 9790  | 79.938  |  |
| Nested Loop      |                 |  | 1.01 - 502.45      | 9790  | 61.877  |  |
| Nested Loop      |                 |  | 0.85 - 496.35      | 9790  | 57.511  |  |
| Memoize          |                 |  | 0.16 - 0.18        | 1     | 0.000   |  |
| Index Scan       | match           |  | 0.28 - 0.30        | 1     | 0.001   | (m_1.match_id = e.match_id)                        |
| Aggregate        |                 |  | 4839.32 - 4839.34  | 2952  | 56.169  |  |
| Sort             |                 |  | 4839.32 - 4839.33  | 6354  | 53.962  |  |
| Nested Loop      |                 |  | 1.15 - 4839.31     | 6354  | 50.733  |  |
| Nested Loop      |                 |  | 0.87 - 4839.01     | 6354  | 38.834  |  |
| Index Scan       | event_type      |  | 0.15 - 8.17        | 1     | 0.024   | ((et_1.type)::text = 'Pass':text)                  |
| Nested Loop      |                 |  | 0.72 - 4824.56     | 6354  | 37.797  |  |
| Index Only Scan  | event_attribute |  | 0.28 - 588.95      | 6354  | 3.444   | (ea_1.attribute_value = any ({}::text[]))          |
| Index Scan       | event           |  | 0.43 - 8.42        | 1     | 0.005   | (e_1.event_id = ea_1.event_id)                     |
| Index Scan       | match           |  | 0.28 - 0.30        | 1     | 0.001   | (m_2.match_id = e_1.match_id)                      |
| Aggregate        |                 |  | 138.77 - 139.03    | 5864  | 111.760 |  |
| Sort             |                 |  | 138.77 - 138.80    | 13769 | 107.251 |  |
| Nested Loop      |                 |  | 1.00 - 138.53      | 13769 | 100.288 |  |
| Nested Loop      |                 |  | 0.72 - 134.66      | 13769 | 74.822  |  |
| Index Only Scan  | event_attribute |  | 0.29 - 24.78       | 13769 | 7.185   | (ea_2.attribute_value = any ({}::text[]))          |
| Index Scan       | event           |  | 0.43 - 8.45        | 1     | 0.005   | (e_2.event_id = ea_2.event_id)                     |
| Index Scan       | match           |  | 0.28 - 0.30        | 1     | 0.001   | (m_3.match_id = e_2.match_id)                      |
| Aggregate        |                 |  | 17.06 - 17.09      | 329   | 3.432   |  |
| Sort             |                 |  | 17.06 - 17.07      | 342   | 3.152   |  |
| Nested Loop      |                 |  | 1.00 - 17.05       | 342   | 2.951   |  |
| Nested Loop      |                 |  | 0.72 - 16.76       | 342   | 2.225   |  |
| Index Only Scan  | event_attribute |  | 0.29 - 8.30        | 342   | 0.253   | (ea_3.attribute_value = 'Red Card':text)           |
| Index Scan       | event           |  | 0.43 - 8.45        | 1     | 0.005   | (e_3.event_id = ea_3.event_id)                     |
| Index Scan       | match           |  | 0.28 - 0.30        | 1     | 0.002   | (m_4.match_id = e_3.match_id)                      |