Напредни бази на податоци

Фаза 3: Оптимизација на прашалници и погледи

Проект IRSON

- Александар Тодороски 231153
- Антонио Трајковски 231156
- Вељко Аџиќ 231267

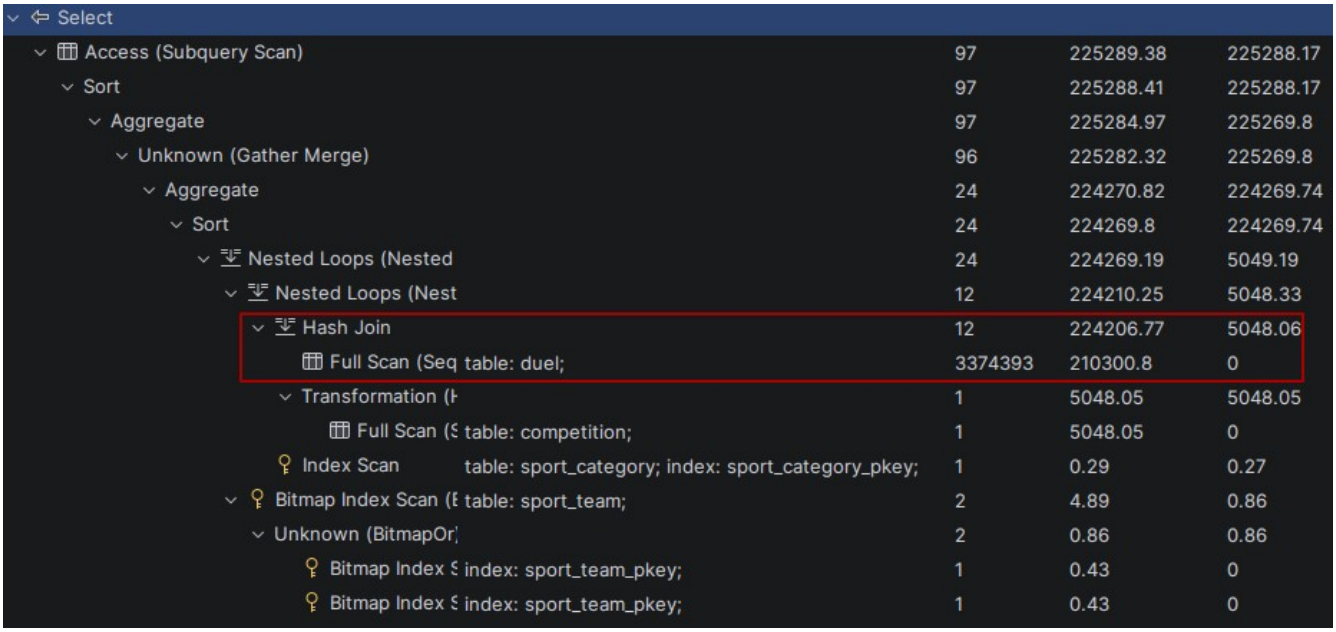
Поглед 1: season_standing

Овој поглед ни дава за секој тим број на натпревари, вкупно поени, и вкупно поени на противникот во секоја сезона и секој натпревар. Примарен случај кога ќе се користи погледот е при излистување на податоци за томовите во одредена сезона.

Иницијално време за извршување трае повеќе од 5 минути (беше прекинато извршувањето), што е неприфатливо за апликацијата.

```
advdb_202526l_prj_irson.public> select * from season_standing where season_id = 783
Cancelling…
[57014] ERROR: canceling statement due to user request
```

Најбавните операции се Full Scan, сортирање и агрегација



Со индексирање сортирање може да се подобри, но во овој поглед подредуваме според агрегација SUM, што нема да помогне. Индексирање на табелата duel ќе помогне бидејќи на неа се прави Full Scan.

```
CREATE INDEX IF NOT EXISTS duel_season_standing_indexes ON
    duel(home_team_id, away_team_id, start_time, competition_id, sport_category_id);
```

Со додавање на индекс се намалува времето на извршување на 4s 18ms.

```
advdb_202526l_prj_irson.public> select * from season_standing where season_id = 783
13 rows retrieved starting from 1 in 4 s 409 ms (execution: 4 s 18 ms, fetching: 391 ms)
```

Поглед 2: monthly_income_for_club:

Овој поглед за секој спортски клуб дава приход, трошоци, профит, активни спонзори и активни договори со играчи во последните 30 дена. Примарната намена за погледот е да овозможува транспарентен преглед на финасиските податоци за одреден клуб.

Извршување на квери со погледот трае 11s 397ms.

```
advdb_202526l_prj_irson.public> select * from monthly_income_for_club where id = 97426
1 row retrieved starting from 1 in 11 s 747 ms (execution: 11 s 397 ms, fetching: 350 ms)
```

Во споредба од претходното, ова е доста побрзо без оптимизација, но пак може да се забрза. Според квери планот можеме да ги видиме 3-те табели каде се извршува Full Scan.

Select				
Aggregate		1	126716.54	63684.74
Nested Loops (Nested Loop)		4535	126671.17	63684.74
Unknown (Gather Merge)		84	62694.39	62684.45
Sort		27	61684.48	61684.41
Hash Join		27	61683.77	7774.42
Full Scan (Seq Sca	table: sponsorship;	1080780	51072.28	0
Transformation (Hash		5	7774.36	7774.36
Full Scan (Seq S	table: sport_team;	5	7774.36	0
Temporary (Materialize)		54	63920.22	1000.29
Nested Loops (Nested L		54	63919.95	1000.29
Index Scan	table: sport_club; index: sport_club_pkey;	1	8.31	0.29
Unknown (Gather)		54	63911.1	1000
Full Scan (Seq Sca	table: sportsperson_contract;	14	62905.7	0

```
CREATE INDEX IF NOT EXISTS sport_team_income_indexes
    on sport_team(club_id);
CREATE INDEX IF NOT EXISTS sponsorship_income_indexes
    on sponsorship(sport_team_id, start_date, end_date);
CREATE INDEX IF NOT EXISTS contract_income_indexes
    on sportsperson_contract(club_id, start_date, end_date);
```

Со овие индексирање на тие 3 табели можеме да добиеме време на извршување 20ms.

```
advdb_202526l_prj_irson.public> select * from monthly_income_for_club where id = 97426
1 row retrieved starting from 1 in 352 ms (execution: 20 ms, fetching: 332 ms)
```

Поглед 3: upcoming_duels

Погледот `upcoming_duels` содржи информации за идно закажани дуели, датумот и времето на дуелот, имиња на тимовите, спортот, име, капацитет и држава на локацијата каде се изигрува, и името на натпревар (доколку има).

Иницијално времен на извршување е 31s 773ms. Ова време е иницијално подобрено од индексирање од поглед 1.

```
advdb_202526l_prj_irson.public> select * from upcoming_duels
                                where match_date <= (now() + INTERVAL '7 days')::date
                                    and country like '%Macedonia'
                                order by match_date, kickoff, capacity DESC
500 rows retrieved starting from 1 in 32 s 126 ms (execution: 31 s 773 ms, fetching: 353 ms)
```

Сепак овој поглед главно филтрира дуели според `start_time`, кој не е опфатен од претходниот индекс. Затоа во планот пак се извршува Full Scan на табелата.

Select				
Sort		10348	269726.43	269700.56
Unknown (Gather Merge)		10348	269010.5	267771.49
Sort		2587	266777.9	266771.43
Nested Loops (Nested Loop)		2587	266624.79	1636.11
Hash Join		2587	265420.67	1635.68
Nested Loops (Nested Loc		2587	265396.21	1618.1
Nested Loops (Nested I		2587	264177.99	1617.68
Hash Join		2587	262959.77	1617.26
Full Scan (Seq Sc	table: duel;	213456	260516.17	0
Transformation (Hasl		1318	1600.78	1600.78
Nested Loops		1318	1600.78	0.42
Full Scan (S	table: country;	2	3.06	0
Index Scan	table: location; index: unique_name_per_country;	659	792.27	0.42
Index Scan	table: sport_team; index: sport_team_pkey;	1	0.47	0.42
Index Scan	table: sport_team; index: sport_team_pkey;	1	0.47	0.42
Transformation (Hash)		337	13.37	13.37
Full Scan (Seq Scan)	table: sport_category;	337	13.37	0
Index Scan	table: competition; index: competition_pkey;	1	0.46	0.42

Креирање на нов индекс каде прво се индексира `start_time`, ќе ги подобри перформансите.

```
CREATE INDEX IF NOT EXISTS duel_upcoming_indexes ON
    duel(
        start_time,
        home_team_id,
        away_team_id,
        sport_category_id,
        location_id,
        competition_id
    );
```

Со индексирање кверито се извршува за 10s 696ms.

```
advdb_202526l_prj_irson.public> select * from upcoming_duels
                                where match_date <= (now() + INTERVAL '7 days')::date
                                    and country like '%Macedonia'
                                order by match_date, kickoff, capacity DESC
500 rows retrieved starting from 1 in 11 s 59 ms (execution: 10 s 696 ms, fetching: 363 ms)
```

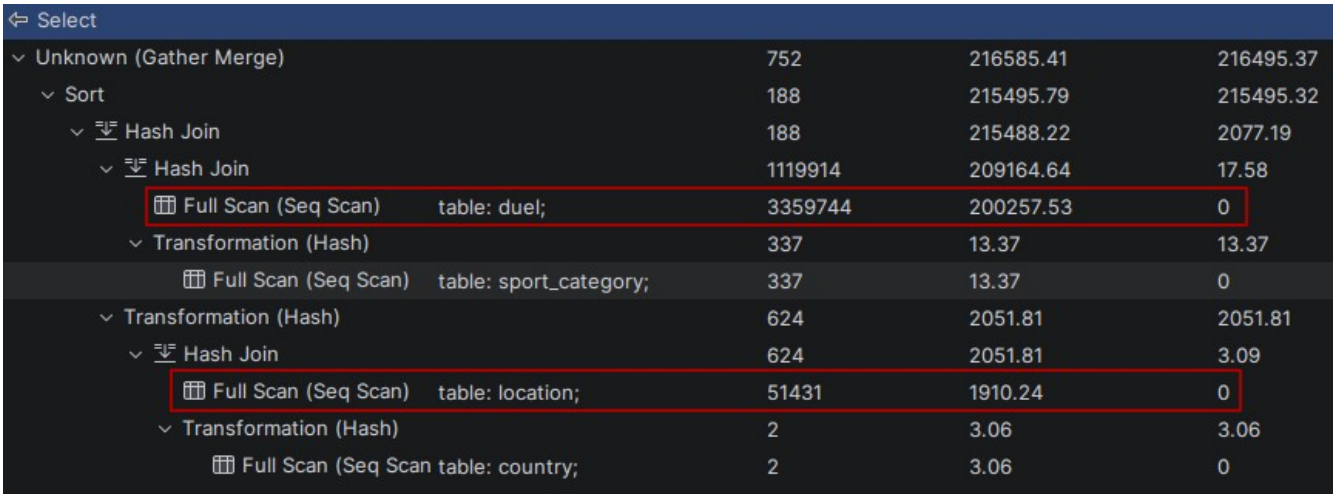
Поглед 4: free_locations

Погледот има главна примена да даде податоци за локации кои се слободни во одредена држава, во одреден временски период. Опционално може да се филтрира и по капацитет.

Иницијалното време на извршување на кверито е 6s 263ms.

```
advdb_202526l_prj_irson.public> SELECT *
                                from free_locations
                                WHERE
                                    capacity >= 10000
                                    and country LIKE '%Macedonia'
                                    and not exists(
                                        select 1
                                        from duel d
                                        join sport_category cat
                                            on cat.id = d.sport_category_id
                                        WHERE d.location_id = venue_id
                                          AND d.start_time < '2026-05-10 21:00:00'::timestamp -- end od range
                                          AND d.start_time +
                                              (interval '1 minute' * cat.duration_minutes) >
                                              '2026-05-10 16:00:00'::timestamp -- start of range
                                    )
                                ORDER BY capacity
454 rows retrieved starting from 1 in 6 s 616 ms (execution: 6 s 263 ms, fetching: 353 ms)
```

Од анализата може да видиме дека се извршува Full Scan на табелите `duel` и `location`.



Со индексирање можеме да го намалиме времето на извршување на 5s 433ms. Минимални добивки добиваме од индексирање, затоа овие индекси не се користат.

```
create index duel_free_locations_indexes on
    duel(start_time) INCLUDE (location_id, sport_category_id);
CREATE INDEX idx_location_capacity ON
    location(capacity, country_id) INCLUDE (name, address);
```

```
advdb_202526l_prj_irson.public> SELECT *
                                from free_locations
                                WHERE
                                    capacity >= 10000
                                    and country LIKE '%Macedonia'
                                    and not exists(
                                        select 1
                                        from duel d
                                        join sport_category cat
                                            on cat.id = d.sport_category_id
                                        WHERE d.location_id = venue_id
                                          AND d.start_time < '2026-05-10 21:00:00'::timestamp -- end od range
                                          AND d.start_time +
                                              (interval '1 minute' * cat.duration_minutes) >
                                              '2026-05-10 16:00:00'::timestamp -- start of range
                                    )
                                ORDER BY capacity
454 rows retrieved starting from 1 in 5 s 755 ms (execution: 5 s 433 ms, fetching: 322 ms)
```

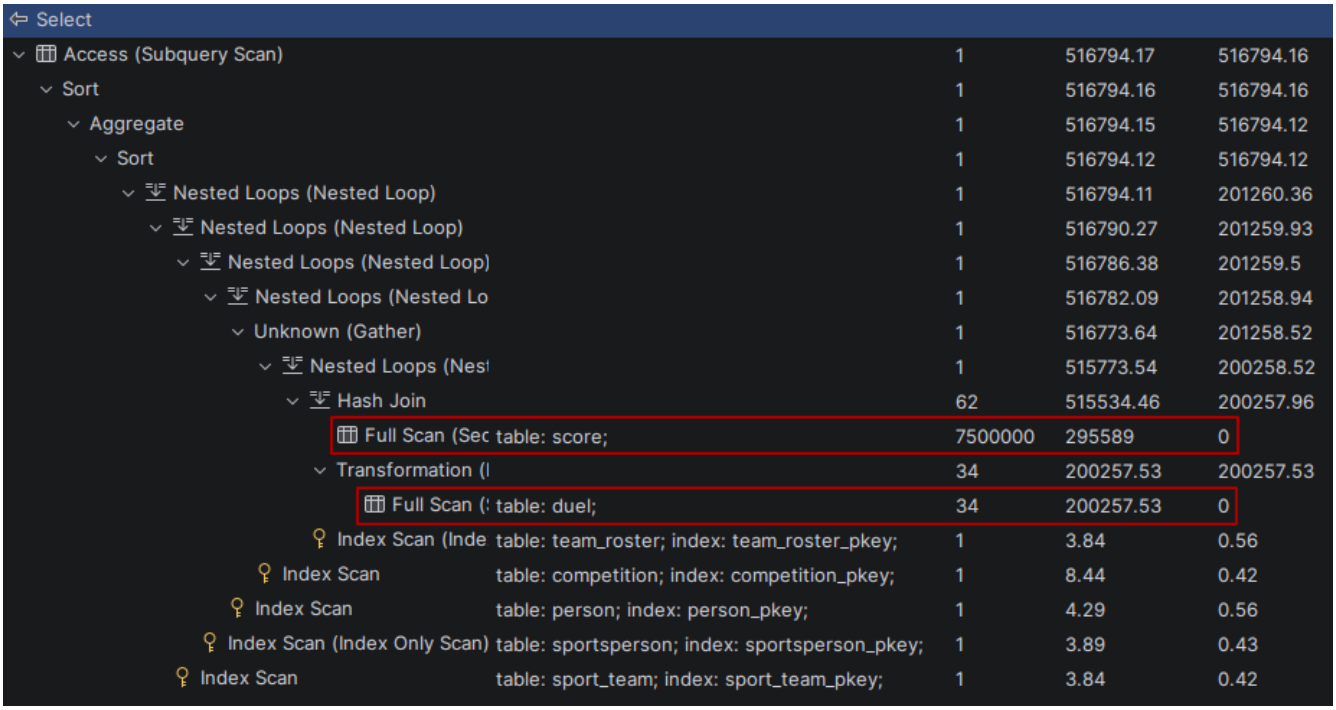
Поглед 5: top_scorers_on_competition

Овој поглед ни ги враќа сите играчи во даден натпревар рангирани според бројот на даден натпревар.

Иницијално извршување на квери со погледот трае повеќе од 8 минути, каде беше привремено прекинато. Ова е неприфатливо за апликацијата.

```
40:03] advdb_202526l_prj_irson.public> select * from top_scorers_on_competition where competition_id = 87756
48:09] Cancelling…
```

Од анализа на квери можеме да видиме дека имаме за 2 табели каде се извршува Full Scan. Овде може да се подобри со индексирање.



Select				
Access (Subquery Scan)		1	516794.17	516794.16
Sort		1	516794.16	516794.16
Aggregate		1	516794.15	516794.12
Sort		1	516794.12	516794.12
Nested Loops (Nested Loop)		1	516794.11	201260.36
Nested Loops (Nested Loop)		1	516790.27	201259.93
Nested Loops (Nested Loop)		1	516786.38	201259.5
Nested Loops (Nested Lo		1	516782.09	201258.94
Unknown (Gather)		1	516773.64	201258.52
Nested Loops (Nest		1	515773.54	200258.52
Hash Join		62	515534.46	200257.96
Full Scan (Sec	table: score;	7500000	295589	0
Transformation (I		34	200257.53	200257.53
Full Scan (S	table: duel;	34	200257.53	0
Index Scan (Inde	table: team_roster; index: team_roster_pkey;	1	3.84	0.56
Index Scan	table: competition; index: competition_pkey;	1	8.44	0.42
Index Scan	table: person; index: person_pkey;	1	4.29	0.56
Index Scan (Index Only Scan)	table: sportsperson; index: sportsperson_pkey;	1	3.89	0.43
Index Scan	table: sport_team; index: sport_team_pkey;	1	3.84	0.42

Ги дефинираме овие индекси со цел да се забрза извршување на кверито:

```
create index score_top_scorers_indexes ON
    score(duel_id) INCLUDE (player_ssn);
create index duel_top_scorers_indexes ON
    duel(competition_id)
    WHERE competition_id IS NOT NULL;
```

Со индексирање време на извршување на квери падна на 104ms.

```
advdb_202526l_prj_irson.public> select * from top_scorers_on_competition where competition_id = 87756
1 row retrieved starting from 1 in 428 ms (execution: 104 ms, fetching: 324 ms)
```

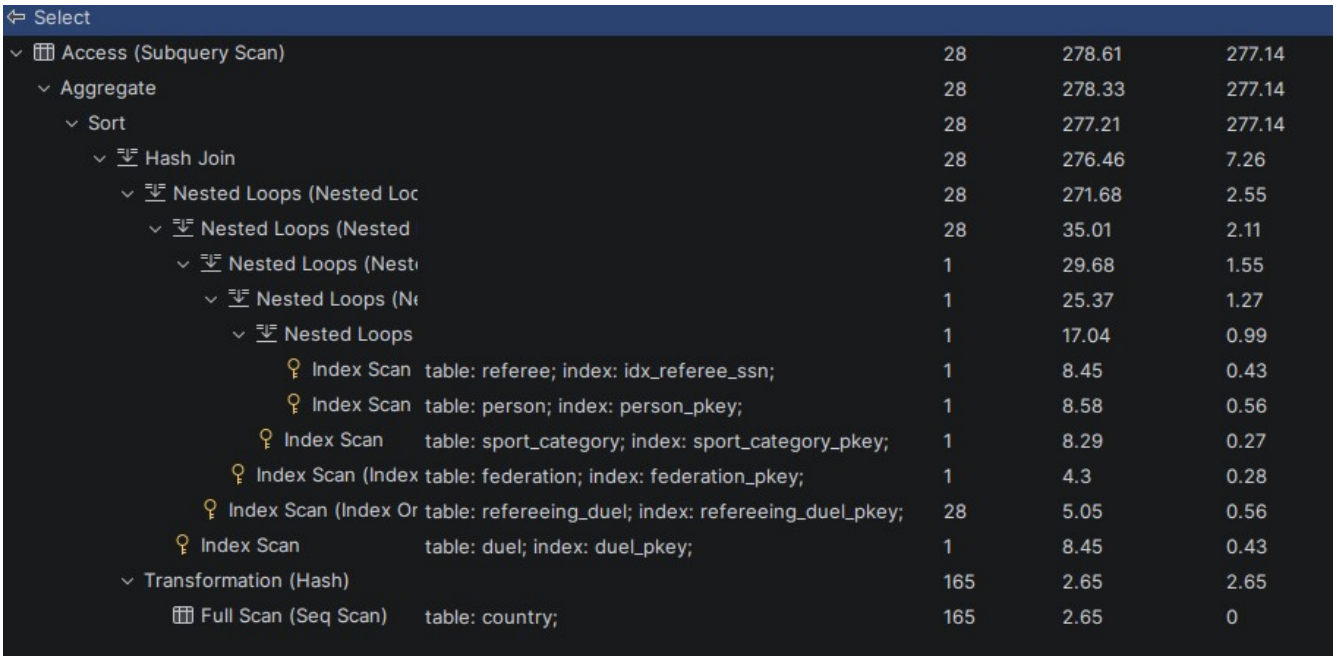
Поглед 6: referee_work

Погледот `referee_work` враќа податоци за судии: ЕМБГ, име и презиме, држава на потекло, спортска категорија, вкупен број дуели на кој судел, датумот на првиот дуел кој судел и бројот на идни дуели кои треба да суди.

Иницијално време на извршување е 1s 29ms, што е добро за апликацијата.

```
advdb_202526l_prj_irson.public> select * from referee_work where referee_ssn = '190495847040'
1 row retrieved starting from 1 in 1 s 360 ms (execution: 1 s 29 ms, fetching: 331 ms)
```

Ако извршиме анализа на кверито можеме да видиме дека скоро секаде имаме пребарување преку индекс. Единствена табела каде има Full Scan е `country`, но бидејќи имаме мал број редици индексирање ќе има мала промена на перформанси.



Select				
Access (Subquery Scan)		28	278.61	277.14
Aggregate		28	278.33	277.14
Sort		28	277.21	277.14
Hash Join		28	276.46	7.26
Nested Loops (Nested Loc		28	271.68	2.55
Nested Loops (Nested		28	35.01	2.11
Nested Loops (Nest		1	29.68	1.55
Nested Loops (Ne		1	25.37	1.27
Nested Loops		1	17.04	0.99
Index Scan	table: referee; index: idx_referee_ssn;	1	8.45	0.43
Index Scan	table: person; index: person_pkey;	1	8.58	0.56
Index Scan	table: sport_category; index: sport_category_pkey;	1	8.29	0.27
Index Scan (Index	table: federation; index: federation_pkey;	1	4.3	0.28
Index Scan (Index Or	table: refereeing_duel; index: refereeing_duel_pkey;	28	5.05	0.56
Index Scan	table: duel; index: duel_pkey;	1	8.45	0.43
Transformation (Hash)		165	2.65	2.65
Full Scan (Seq Scan)	table: country;	165	2.65	0

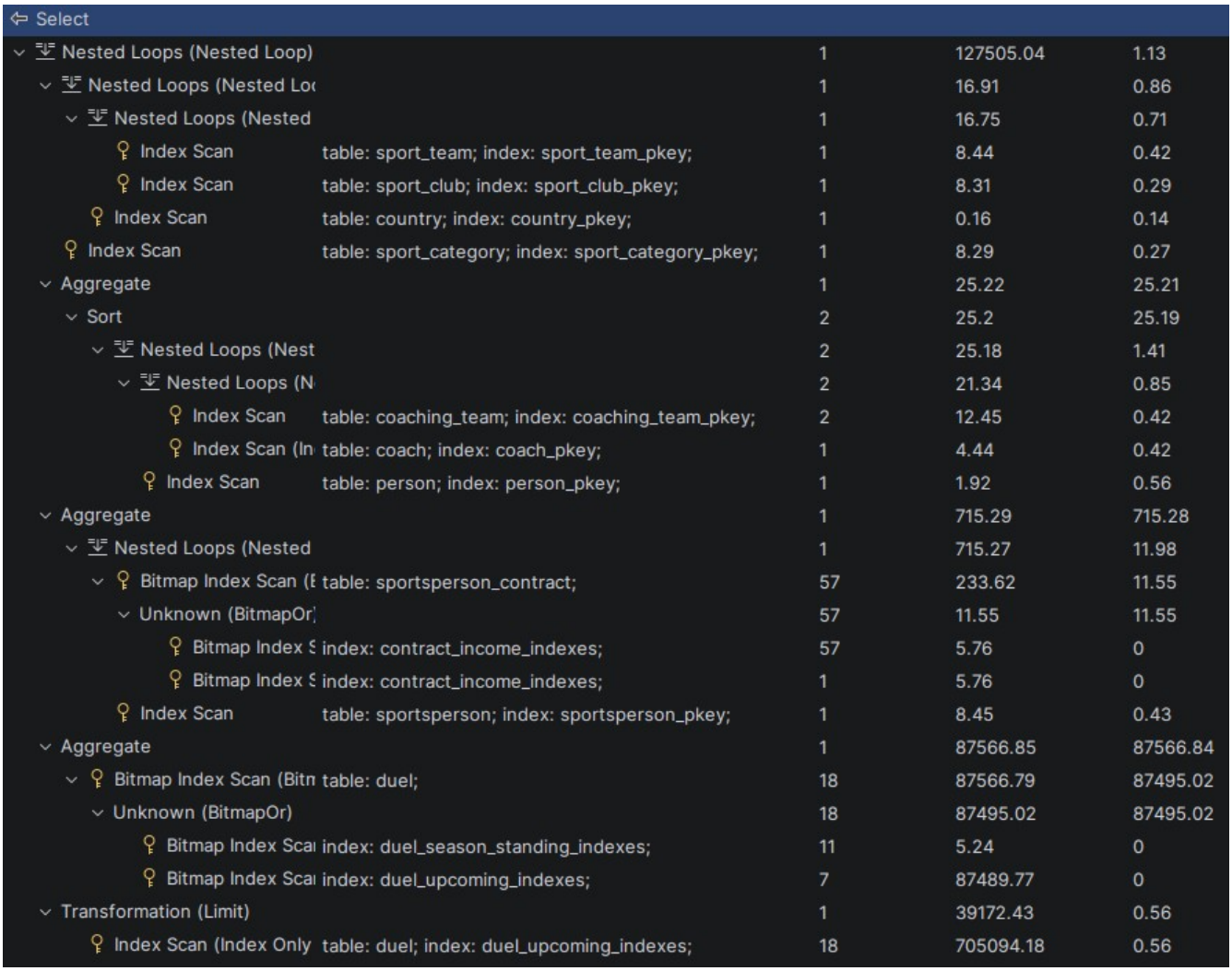
Поглед 7: team_stats

Овој поглед содржи податоци за секој тим: име на тимот, спортска категорија, држава на потекло, тренери на тимот, бројот на активни договори, бројот на закажани дуели во кои учествуваат и датум на следниот дуел во кој учествуваат.

Иницијално време на извршување на кверито е 12s 132ms.

```
advdb_202526l_prj_irson.public> select * from team_stats ts where ts.team_id = 205986
1 row retrieved starting from 1 in 12 s 471 ms (execution: 12 s 132 ms, fetching: 339 ms)
```

Од анализата може да видиме дека претходните индекси се искористуваат и доведува до побрзо извршување.



Select				
Nested Loops (Nested Loop)		1	127505.04	1.13
Nested Loops (Nested Lo		1	16.91	0.86
Nested Loops (Nested		1	16.75	0.71
Index Scan	table: sport_team; index: sport_team_pkey;	1	8.44	0.42
Index Scan	table: sport_club; index: sport_club_pkey;	1	8.31	0.29
Index Scan	table: country; index: country_pkey;	1	0.16	0.14
Index Scan	table: sport_category; index: sport_category_pkey;	1	8.29	0.27
Aggregate		1	25.22	25.21
Sort		2	25.2	25.19
Nested Loops (Nest		2	25.18	1.41
Nested Loops (N		2	21.34	0.85
Index Scan	table: coaching_team; index: coaching_team_pkey;	2	12.45	0.42
Index Scan (In	table: coach; index: coach_pkey;	1	4.44	0.42
Index Scan	table: person; index: person_pkey;	1	1.92	0.56
Aggregate		1	715.29	715.28
Nested Loops (Nested		1	715.27	11.98
Bitmap Index Scan (E	table: sportsperson_contract;	57	233.62	11.55
Unknown (BitmapOr)		57	11.55	11.55
Bitmap Index S	index: contract_income_indexes;	57	5.76	0
Bitmap Index S	index: contract_income_indexes;	1	5.76	0
Index Scan	table: sportsperson; index: sportsperson_pkey;	1	8.45	0.43
Aggregate		1	87566.85	87566.84
Bitmap Index Scan (Bitn	table: duel;	18	87566.79	87495.02
Unknown (BitmapOr)		18	87495.02	87495.02
Bitmap Index Scai	index: duel_season_standing_indexes;	11	5.24	0
Bitmap Index Scai	index: duel_upcoming_indexes;	7	87489.77	0
Transformation (Limit)		1	39172.43	0.56
Index Scan (Index Only	table: duel; index: duel_upcoming_indexes;	18	705094.18	0.56

Сепак ако додадеме индекс на `coaching_team` можеме да постигнеме време на извршување од 1s 339ms.

```
CREATE INDEX coaching_team_team_stats_indexes ON
    coaching_team(team_id) WHERE end_date IS NULL;
```

```
advdb_202526l_prj_irson.public> select * from team_stats ts where ts.team_id = 205986
1 row retrieved starting from 1 in 1 s 658 ms (execution: 1 s 339 ms, fetching: 319 ms)
```

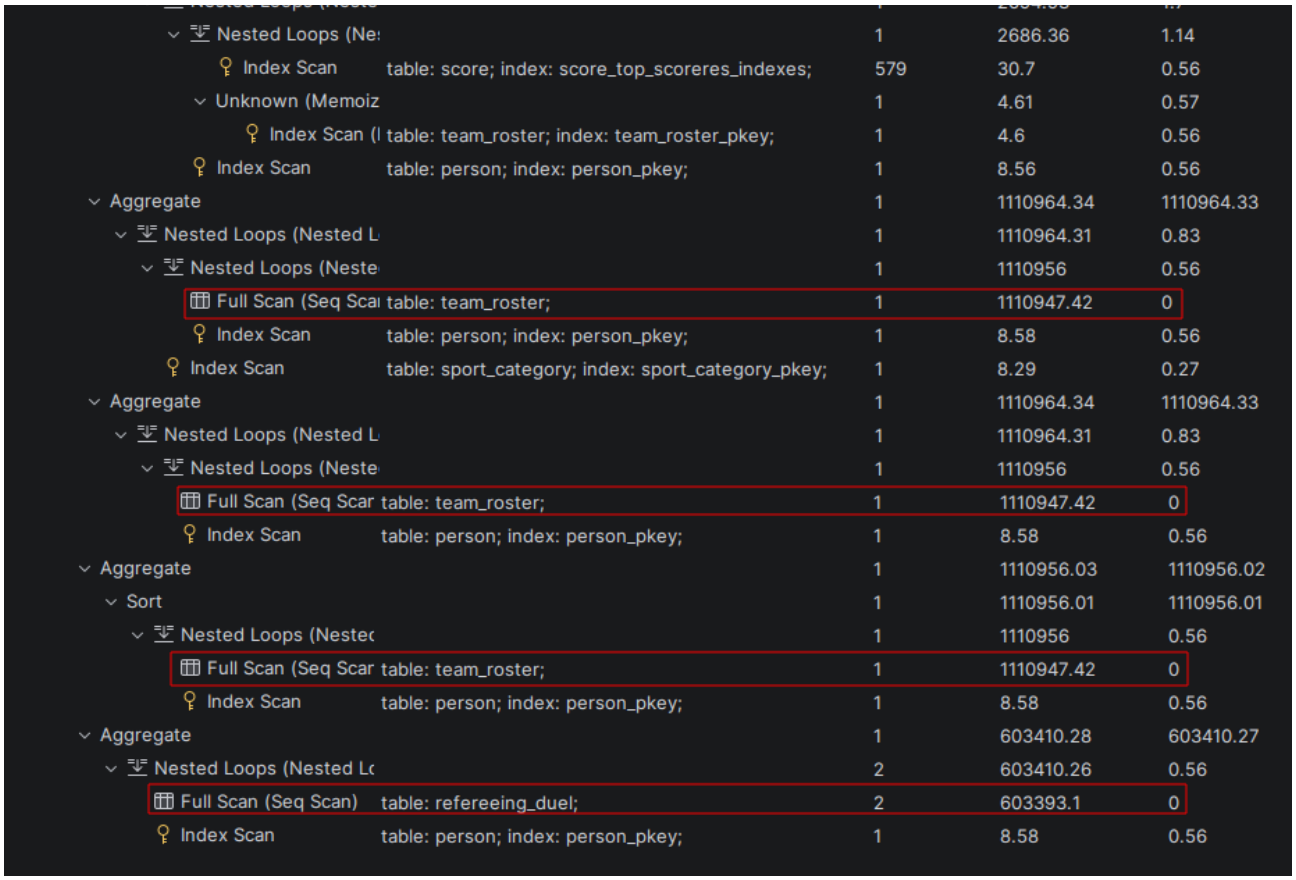
Поглед 8: duel_history

Овој поглед ни дава податоци за секој дуел: кога почнал, на која локација се изигрува, имиња на тимовите, освоени поени од тимовите, резултатот, црвени картони за тимовите, кои играчи играат во секој тим, и кои судии го судат дуелот.

Иницијално квери трае повеќе од 8 минути, каде беше привремено прекинато. За ова квери ни треба оптимизирање.

```
01:07] advdb_202526l_prj_irson.public> select * from duel_history where duel_id = 23
09:38] Cancelling…
```

Од анализата можеме да видиме дека голем дел операции користат индекси, но на 4 места се извршува Full Scan на големи табели, што доведува до долгото време на извршување.



Nested Loops (Ne		1	2686.36	1.14
Index Scan	table: score; index: score_top_scoreres_indexes;	579	30.7	0.56
Unknown (Memoiz		1	4.61	0.57
Index Scan (I	table: team_roster; index: team_roster_pkey;	1	4.6	0.56
Index Scan	table: person; index: person_pkey;	1	8.56	0.56
Aggregate		1	1110964.34	1110964.33
Nested Loops (Nested L		1	1110964.31	0.83
Nested Loops (Neste		1	1110956	0.56
Full Scan (Seq Sca	table: team_roster;	1	1110947.42	0
Index Scan	table: person; index: person_pkey;	1	8.58	0.56
Index Scan	table: sport_category; index: sport_category_pkey;	1	8.29	0.27
Aggregate		1	1110964.34	1110964.33
Nested Loops (Nested L		1	1110964.31	0.83
Nested Loops (Neste		1	1110956	0.56
Full Scan (Seq Scar	table: team_roster;	1	1110947.42	0
Index Scan	table: person; index: person_pkey;	1	8.58	0.56
Aggregate		1	1110956.03	1110956.02
Sort		1	1110956.01	1110956.01
Nested Loops (Nestec		1	1110956	0.56
Full Scan (Seq Scar	table: team_roster;	1	1110947.42	0
Index Scan	table: person; index: person_pkey;	1	8.58	0.56
Aggregate		1	603410.28	603410.27
Nested Loops (Nested Lc		2	603410.26	0.56
Full Scan (Seq Scan)	table: refereeing_duel;	2	603393.1	0
Index Scan	table: person; index: person_pkey;	1	8.58	0.56

Креирани се овие два индекси за табелите на кои им се извршува Full Scan.

```
CREATE INDEX roster_duel_history_indexes ON
    team_roster(duel_id, team_id) INCLUDE (player_ssn, end_time);
CREATE INDEX idx_refereeing_duel_duel_id ON
    refereeing_duel(duel_id) INCLUDE (referee_ssn);
```

Со овие индекси добивме време на извршување 1s 113ms.

```
advdb_202526l_prj_irson.public> select * from duel_history where duel_id = 23
1 row retrieved starting from 1 in 1 s 456 ms (execution: 1 s 113 ms, fetching: 343 ms)
```

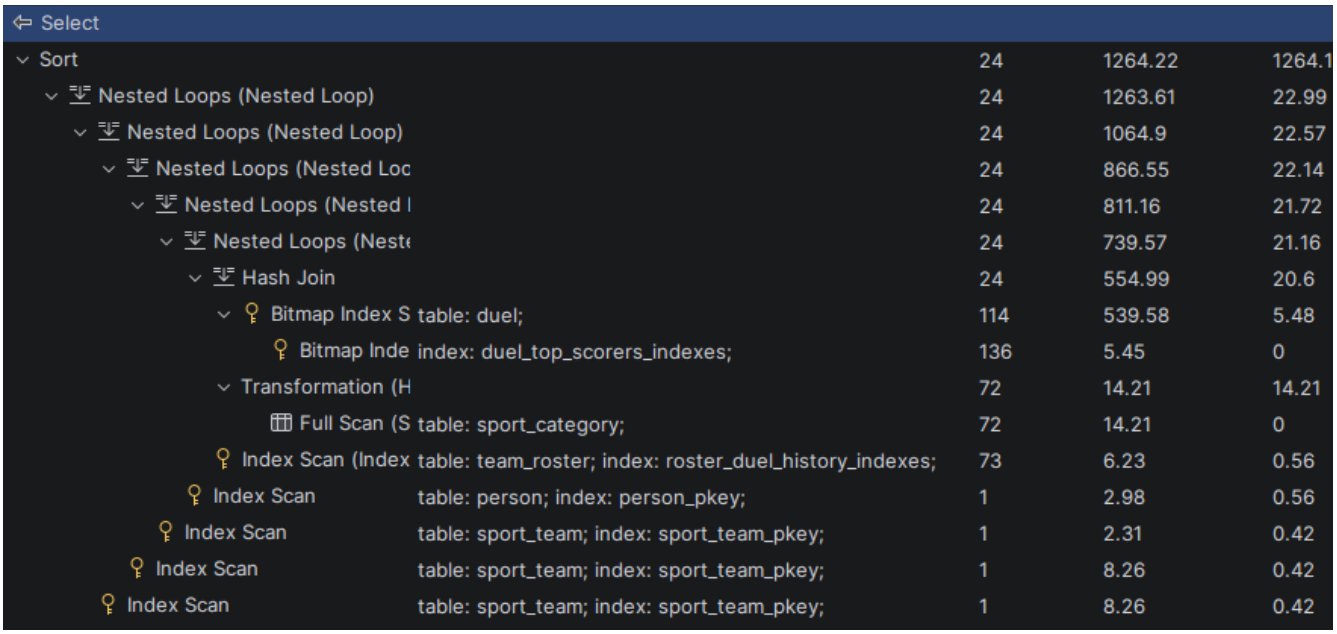
Поглед 9: get_red_cards

Помошен поглед кој ги враќа сите играчи што добиле црвен картон за секој дуел. Враќа податоци за дуелот (id, имиња на тимовите, спортска категорија и натпреварот) и информации за играчот (име, за кој тим игра, кога почнал, кога излегол од играта и колку минути пропуштил).

Иницијалното време на извршување квери е 20ms.

```
advdb_202526l_prj_irson.public> select * from get_red_cards WHERE competition_id = 877
7 rows retrieved starting from 1 in 378 ms (execution: 20 ms, fetching: 358 ms)
```

Од анализата можеме да видиме дека во скоро сите операции се користи индекс. Единствената табела каде што се има Full Scan е `sport_category`, но бидејќи табелата има малку редици индексирање нема да доведе до поголемо подобрување.



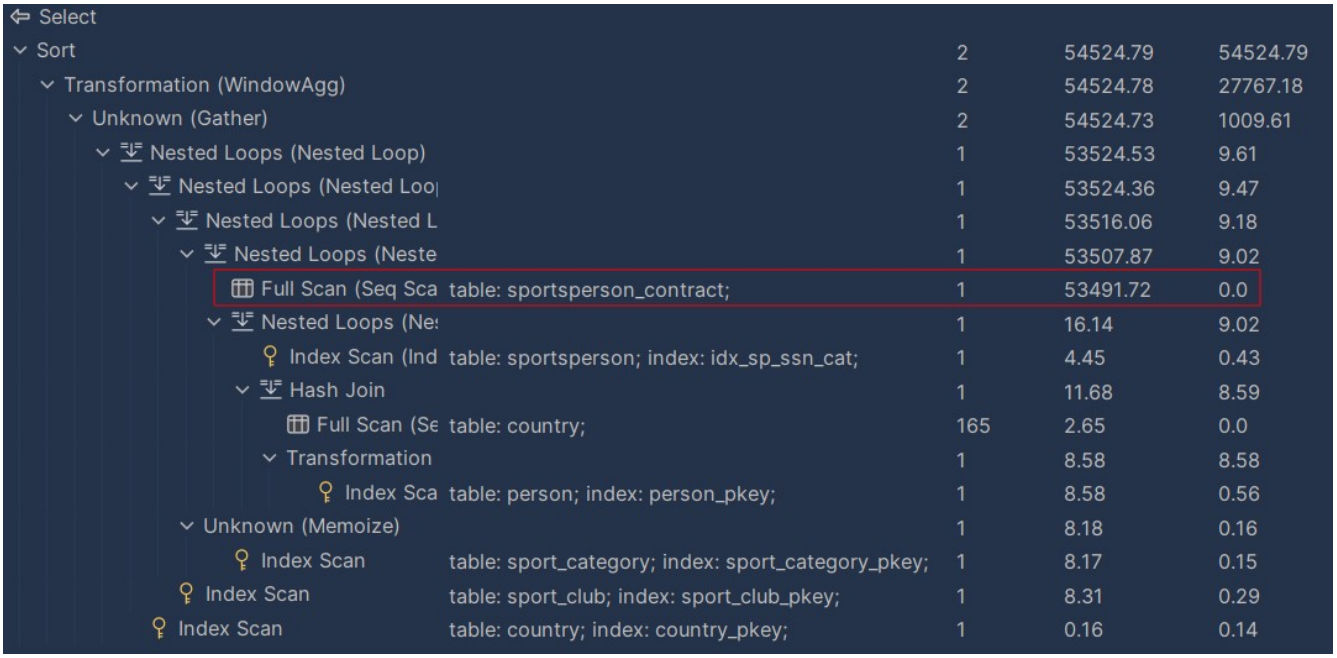
Поглед 10: player_career_history

Овој поглед враќа податоци за сите договори кој склучил спортист. Го враќа ЕМБГ на спортистот, името, датум на раѓање, национална припадност, спортска категорија, име на клубот со кој склучува договр, државата на клубот, почеток и крај на договорот, и статус на договорот.

Иницијалната имплементација извршува за 881ms.

```
postgres.public> select * from player_career_history where ssn = '220598645596'
2 rows retrieved starting from 1 in 1 s 390 ms (execution: 881 ms, fetching: 509 ms)
```

Од анализата може да видиме дека има 2 табели што се извршуваат со Full Scan: `country` и `sportsperson_contract`. Табелата за држави е мала, зато кај неа индексирање нема да врати многу подобрувања во перформанси, но табелата за договори и треба.



По креирање на индексот гледаме дека има минимално забрзување на брзината, односно ни се намали на 18ms. Затоа овој индекс нема да се користи.

```
CREATE INDEX idx_sportsperson_contract_player_ssn_start_date
    ON sportsperson_contract (player_ssn, start_date);
```

```
select * from player_career_history where ssn = '220598645596'
starting from 1 in 1 s 60 ms (execution: 18 ms, fetching: 1 s 42 ms)
```