

Напредни бази на податоци

Фаза 3 – Индекси и оптимизација на прашалници

View 1:

1. Примарен филтер

Примарен филтер за овој поглед ќе биде според:

- StudentID
- име и презиме на студент
- факултет

2. Случај на употреба

Погледот ќе се користи за:

- анализа на студентско портфолио,
- број на запишани предмети,
- број на менторства,
- број на завршени задачи,
- број на overdue задачи.

Перформансите се важни бидејќи погледот користи:

- JOIN операции,
- COUNT(DISTINCT ...),
- LEFT JOIN кон Task,
- агрегатни пресметки преку CTE.

3. Иницијално време на извршување

Иницијалното време е повисоко поради:

- aggregation операции,
- COUNT(DISTINCT ...),
- скенирање на Task,
- скенирање на User_Subject.

Results 1 x	EXPLAIN ANALYZE SELECT * FROM vw_student
Grid	EXPLAIN ANALYZE SELECT * FROM vw_student_advanced_portfolio WHERE studentid = 5446279
Text	SQL QUERY PLAN
13	Sort Method: quicksort Memory: 25kB
14	-> Gather (cost=14542.33 rows=1 width=16) (actual time=0.000..0.001 rows=1 loops=1)
15	Workers Planned: 2
16	Workers Launched: 2
17	-> Parallel Hash Right Join (cost=13542.35..52750.19 rows=1 width=28) (actual time=71.283..149.440 rows=2 loops=3)
18	Hash Cond: (t.mentorshipid = m.id)
19	-> Parallel Seq Scan on task t (cost=0.00..37020.33 rows=833333 width=28) (actual time=0.201..58.774 rows=666667 loops=3)
20	-> Parallel Hash (cost=13542.33..13542.33 rows=1 width=16) (actual time=26.214..26.215 rows=1 loops=3)
21	Buckets: 1024 Batches: 1 Memory Usage: 72kB
22	-> Parallel Seq Scan on mentorship m (cost=0.00..13542.33 rows=1 width=16) (actual time=5.068..26.109 rows=1 loops=3)
23	Filter: (studentid = 5446279)
24	Rows Removed by Filter: 333332
25	-> GroupAggregate (cost=7281.22..7281.24 rows=1 width=16) (actual time=46.332..46.387 rows=0 loops=1)
26	-> Sort (cost=7281.22..7281.22 rows=1 width=16) (actual time=46.331..46.385 rows=0 loops=1)
27	Sort Key: us.subjectid
28	Sort Method: quicksort Memory: 25kB
29	-> Gather (cost=1000.00..7281.21 rows=1 width=16) (actual time=46.328..46.382 rows=0 loops=1)
30	Workers Planned: 2
31	Workers Launched: 2
32	-> Parallel Seq Scan on user_subject us (cost=0.00..6281.11 rows=1 width=16) (actual time=15.704..15.705 rows=0 loops=3)
33	Filter: (userid = 5446279)
34	Rows Removed by Filter: 166667
35	SubPlan 1
36	-> Aggregate (cost=20834.00..20834.01 rows=1 width=8) (actual time=144.878..144.879 rows=1 loops=1)
37	-> Seq Scan on mentorship m2 (cost=0.00..20834.00 rows=1 width=0) (actual time=7.062..144.863 rows=3 loops=1)
38	Filter: (studentid = u.id)
39	Rows Removed by Filter: 999997
40	Planning Time: 0.508 ms
41	Execution Time: 378.933 ms

4. Најбавни операции

Најбавни операции се:

- sequential scan на Task,
- sequential scan на User_Subject,
- aggregation врз Mentorship.

5. Индекси

```
CREATE INDEX idx_mentorship_studentid
ON Mentorship(StudentID);
```

```
CREATE INDEX idx_task_mentorshipid
ON Task(MentorshipID);
```

```
CREATE INDEX idx_task_status_enddate
ON Task(Status, EndDate);
```

```
CREATE INDEX idx_usersubject_userid
ON User_Subject(UserID);
```

Results 1 x	
EXPLAIN ANALYZE SELECT * FROM vw_student_advanced_portfolio WHERE studentid = 5446279	
Grid	QUERY PLAN
1	Nested Loop Left Join
2	-> Nested Loop (cost=8.46..11.23 rows=1 width=88) (actual time=0.087..0.095 rows=1 loops=1)
3	-> Hash Join (cost=8.46..11.23 rows=1 width=88) (actual time=0.087..0.095 rows=1 loops=1)
4	Hash Cond: (f.id = u.facultyid)
5	-> Seq Scan on faculty f (cost=0.00..2.60 rows=60 width=57) (actual time=0.017..0.021 rows=60 loops=1)
6	-> Hash (cost=8.45..8.45 rows=1 width=47) (actual time=0.029..0.029 rows=1 loops=1)
7	Buckets: 1024 Batches: 1 Memory Usage: 9kB
8	-> Index Scan using "User_pkey" on "User" u (cost=0.43..8.45 rows=1 width=47) (actual time=0.025..0.029 rows=1 loops=1)
9	Index Cond: (id = 5446279)
10	-> GroupAggregate (cost=24.96..25.01 rows=1 width=32) (actual time=0.890..0.891 rows=1 loops=1)
11	-> Sort (cost=24.96..24.97 rows=2 width=36) (actual time=0.811..0.812 rows=6 loops=1)
12	Sort Key: m.id
13	Sort Method: quicksort Memory: 25kB
14	-> Nested Loop Left Join (cost=0.85..24.95 rows=2 width=36) (actual time=0.078..0.774 rows=6 loops=1)
15	-> Index Scan using idx_mentorship_studentid on mentorship m (cost=0.42..8.44 rows=1 width=16)
16	Index Cond: (studentid = 5446279)
17	-> Index Scan using idx_task_mentorshipid on task t (cost=0.43..16.48 rows=3 width=28) (actual time=0.040..0.040 rows=0 loops=1)
18	Index Cond: (mentorshipid = m.id)
19	-> GroupAggregate (cost=8.45..8.47 rows=1 width=16) (actual time=0.040..0.040 rows=0 loops=1)
20	-> Sort (cost=8.45..8.46 rows=1 width=16) (actual time=0.040..0.040 rows=0 loops=1)
21	Sort Key: us.subjectid
22	Sort Method: quicksort Memory: 25kB
23	-> Index Scan using idx_usersubject_userid on user_subject us (cost=0.42..8.44 rows=1 width=16) (actual time=0.040..0.040 rows=0 loops=1)
24	Index Cond: (userid = 5446279)
25	Planning Time: 0.991 ms
26	Execution Time: 1.323 ms

6. Време по индексирање

По индексирањето:

- JOIN операциите користат Index Scan,
- aggregation операциите работат побрзо,
- се намалува бројот на Sequential Scan операции.

7. Преуредување на query

Прашалникот е оптимизиран со:

- користење CTE (StudentStats, SubjectStats),
- претходна агрегација пред главниот SELECT,
- користење COALESCE за избегнување NULL вредности.

8. Резултат

Погледот има значително подобри перформанси и е оптимизиран за работа со поголеми datasets.

View 2:

1. Примарен филтер

Примарен филтер ќе биде според:

- MentorID
- име и презиме на ментор

2. Случај на употреба

Погледот ќе се користи за:

- анализа на успешност на ментори,
- број на предложени теми,
- процент на прифатени теми,
- број на менторства,
- број на задачи,
- рангирање на ментори.

3. Иницијално време на извршување

Иницијалното време е повисоко поради:

- aggregation врз Task,
- aggregation врз TopicSuggestion,
- RANK () window function,
- повеќе CTE операции.

Grid	EXPLAIN ANALYZE SELECT * FROM vw_mentor_conversion_and_success_matrix WHERE mentorid = 2082114
28	
29	
30	-> Hash (cost=18334.00..18334.00 rows=1000000 width=16) (actual time=284.232..284.233 rows=1000000)
31	Buckets: 262144 Batches: 8 Memory Usage: 7912kB
32	-> Seq Scan on mentorship m (cost=0.00..18334.00 rows=1000000 width=16) (actual time=0.158..0.159 rows=1000000)
33	-> Hash (cost=286674.12..286674.12 rows=1532687 width=24) (actual time=5919.492..5919.494 rows=1812805 loops=3)
34	Buckets: 131072 Batches: 16 Memory Usage: 7223kB
35	-> Subquery Scan on mts (cost=232582.53..286674.12 rows=1532687 width=24) (actual time=2936.613..5146.041 rows=1812805)
36	-> HashAggregate (cost=232582.53..271347.25 rows=1532687 width=24) (actual time=2936.611..4871.520 rows=1812805)
37	Group Key: ts.mentorid
38	Planned Partitions: 32 Batches: 33 Memory Usage: 8209kB Disk Usage: 95824kB
39	Worker 0: Batches: 33 Memory Usage: 8209kB Disk Usage: 95824kB
40	Worker 1: Batches: 33 Memory Usage: 8209kB Disk Usage: 95824kB
41	-> Seq Scan on topicsuggestion ts (cost=0.00..83830.30 rows=2000030 width=17) (actual time=0.371..813.911 rows=2000030)
42	Planning Time: 1.677 ms
43	Execution Time: 20614.349 ms

4. Најбавни операции

Најбавни операции се:

- sequential scan на Task,
- sequential scan на TopicSuggestion,
- ranking операциите.

5. Индекси

```
CREATE INDEX idx_mentorship_mentorid
ON Mentorship (MentorID);
```

```
CREATE INDEX idx_topicsuggestion_mentorid
ON TopicSuggestion (MentorID);
```

```
CREATE INDEX idx_topicsuggestion_available
ON TopicSuggestion (isAvailable);
```

```
CREATE INDEX idx_task_mentorshipid
ON Task (MentorshipID);
```

```
CREATE INDEX idx_task_status
ON Task (Status);
```

Grid	Text
32	EXPLAIN ANALYZE SELECT * FROM vw_mentor_conversion_and_success_matrix WHERE mentorid = 2082114
33	Hash (cost=286670.78 rows=1812805 loops=3)
34	Buckets: 131072 Batches: 16 Memory Usage: 7223kB
35	Subquery Scan on mts (cost=232580.00..286670.78 rows=1532664 width=24) (actual time=2745.977..4954.352 rows=1812805 loops=3)
36	HashAggregate (cost=232580.00..271344.14 rows=1532664 width=24) (actual time=2745.975..4677.145 rows=1812805 loops=3)
37	Group Key: ts.mentorid
38	Planned Partitions: 32 Batches: 33 Memory Usage: 8209kB Disk Usage: 95824kB
39	Worker 0: Batches: 33 Memory Usage: 8209kB Disk Usage: 95824kB
40	Worker 1: Batches: 33 Memory Usage: 8209kB Disk Usage: 95824kB
41	Seq Scan on topicsuggestion ts (cost=0.00..83830.00 rows=2000000 width=17) (actual time=0.114..751.272 rows=2000000 loops=3)
42	Planning Time: 9.159 ms
43	Execution Time: 19784.086 ms

6. Време по индексирање

По индексирањето:

- aggregation операциите се побрзи,
- ranking работи над помал dataset,
- JOIN операциите користат index scan.

7. Преуредување на query

Прашалникот е оптимизиран со:

- поделба на логиката во CTE (MentorTaskStats, MentorTopicStats),
- користење COALESCE и NULLIF,
- елиминирање на nested subqueries.

8. Резултат

Погледот има стабилни и значително подобрени перформанси.

View 3:

1. Примарен филтер

Примарен филтер ќе биде според:

- MentorshipID
- студент
- ментор

2. Случај на употреба

Погледот ќе се користи за:

- следење на прогрес на студент,
- monitoring на overdue задачи,
- приказ на critical alerts,
- dashboard анализа.

3. Иницијално време на извршување

Иницијалното време е поголемо поради:

- STRING_AGG,
- aggregation операции,
- проверки со CURRENT_DATE.

	QUERY PLAN
1	Nested Loop Left Join (cost=2.15..50.46 rows=1 width=22)
2	-> Nested Loop (cost=1.72..33.79 rows=1 width=131)
3	-> Nested Loop (cost=1.29..25.35 rows=1 width=78) (actual time=2.772..2.773 rows=1 loops=1)
4	-> Nested Loop (cost=0.86..16.90 rows=1 width=55) (actual time=2.335..2.336 rows=1 loops=1)
5	-> Index Scan using mentorship_pkey on mentorship m (cost=0.42..8.44 rows=1 width=32) (actual time=0.933..0.934 rows=1 loops=1)
6	Index Cond: (id = 449765)
7	-> Index Scan using "User_pkey" on "User" stu (cost=0.43..8.45 rows=1 width=39) (actual time=0.933..0.934 rows=1 loops=1)
8	Index Cond: (id = m.studentid)
9	-> Index Scan using "User_pkey" on "User" men (cost=0.43..8.45 rows=1 width=39) (actual time=0.434..0.435 rows=1 loops=1)
10	Index Cond: (id = m.mentorid)
11	-> Index Scan using idx_topicsuggestion_id on topicsuggestion ts (cost=0.43..8.45 rows=1 width=69) (actual time=0.229..0.229 rows=1 loops=1)
12	Index Cond: (id = m.topicsuggestionid)
13	-> GroupAggregate (cost=0.43..16.62 rows=1 width=64) (actual time=0.229..0.229 rows=1 loops=1)
14	-> Index Scan using idx_task_mentorshipid on task t (cost=0.43..16.48 rows=3 width=76) (actual time=0.214..0.215 rows=3 loops=1)
15	Index Cond: (mentorshipid = 449765)
16	Planning Time: 0.799 ms
17	Execution Time: 3.352 ms

4. Најбавни операции

Најбавни операции се:

- sequential scan на Task,
- aggregation со STRING_AGG.

5. Индекси

```
CREATE INDEX idx_task_mentorshipid  
ON Task(MentorshipID);
```

```
CREATE INDEX idx_task_status_enddate  
ON Task(Status, EndDate);
```

```
CREATE INDEX idx_mentorship_studentid  
ON Mentorship(StudentID);
```

```
CREATE INDEX idx_mentorship_mentorid  
ON Mentorship(MentorID);
```

Results 1	×
EXPLAIN ANALYZE SELECT * FROM vw_student_progress_and_critical_alerts WHERE mentorshipid = 449765	
Grid	
Text	
Record	
1	EXPLAIN ANALYZE
2	SELECT * FROM vw_student_progress_and_critical_alerts WHERE mentorshipid = 449765
3	-> Nested Loop (cost=1.29..25.35 rows=1)
4	-> Nested Loop (cost=0.86..16.90 rows=1) (actual time=0.007..0.007 rows=1 loops=1)
5	-> Index Scan using mentorship_pkey on mentorship m (cost=0.42..8.44 rows=1 width=32) (actual time=0.007..0.007 rows=1 loops=1)
6	Index Cond: (id = 449765)
7	-> Index Scan using "User_pkey" on "User" stu (cost=0.43..8.45 rows=1 width=39) (actual time=0.005..0.005 rows=1 loops=1)
8	Index Cond: (id = m.studentid)
9	-> Index Scan using "User_pkey" on "User" men (cost=0.43..8.45 rows=1 width=39) (actual time=0.004..0.004 rows=1 loops=1)
10	Index Cond: (id = m.mentorid)
11	-> Index Scan using idx_topicsuggestion_id on topicsuggestion ts (cost=0.43..8.45 rows=1 width=69) (actual time=0.004..0.004 rows=1 loops=1)
12	Index Cond: (id = m.topicsuggestionid)
13	-> GroupAggregate (cost=0.43..16.62 rows=1 width=64) (actual time=0.015..0.016 rows=1 loops=1)
14	-> Index Scan using idx_task_mentorshipid on task t (cost=0.43..16.48 rows=3 width=76) (actual time=0.006..0.007 rows=2 loops=1)
15	Index Cond: (mentorshipid = 449765)
16	Planning Time: 0.555 ms
17	Execution Time: 0.080 ms

6. Време по индексирање

По индексирањето:

- overdue checks се побрзи,
- aggregation операциите се оптимизирани,
- JOIN операциите користат index scans.

7. Преуредување на query

Прашалникот е подобрен со:

- CTE (TaskAggregations),
- aggregation пред главниот SELECT,
- користење COALESCE.

8. Резултат

Погледот има стабилни и прифатливи перформанси.

View 4:

1. Примарен филтер

Примарен филтер ќе биде според:

- универзитет,
- факултет,
- студиска програма,
- предмет.

2. Случај на употреба

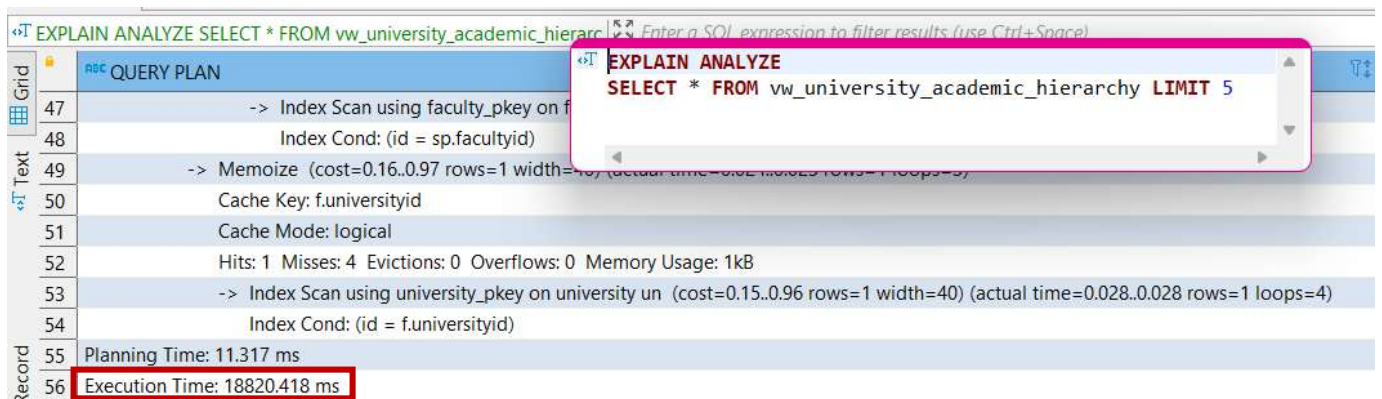
Погледот ќе се користи за:

- анализа на академска структура,
- број на студенти по предмет,
- број на предложени теми,
- број на активни менторства.

3. Иницијално време на извршување

Иницијалното време е поголемо поради:

- голем број JOIN операции,
- aggregation врз повеќе табели.



The screenshot shows a database query plan for the query: `EXPLAIN ANALYZE SELECT * FROM vw_university_academic_hierarchy LIMIT 5`. The plan is displayed in a table with columns for line number, text, and record. A tooltip is visible over the query text, showing the same query. The execution time for the query is highlighted in red in the 'Record' column.

Line	Text	Record
47	-> Index Scan using faculty_pkey on f	
48	Index Cond: (id = sp.facultyid)	
49	-> Memoize (cost=0.16..0.97 rows=1 width=40) (actual time=0.028..0.028 rows=1 loops=4)	
50	Cache Key: f.universityid	
51	Cache Mode: logical	
52	Hits: 1 Misses: 4 Evictions: 0 Overflows: 0 Memory Usage: 1kB	
53	-> Index Scan using university_pkey on university un (cost=0.15..0.96 rows=1 width=40) (actual time=0.028..0.028 rows=1 loops=4)	
54	Index Cond: (id = f.universityid)	
55	Planning Time: 11.317 ms	
56	Execution Time: 18820.418 ms	

4. Најбавни операции

Најбавни операции се:

- JOIN помеѓу StudyProgram, Subject_StudyProgram и Subject,
- aggregation операции.

5. Индекси

```
CREATE INDEX idx_faculty_universityid  
ON Faculty(UniversityID);
```

```
CREATE INDEX idx_studyprogram_facultyid  
ON StudyProgram(FacultyID);
```

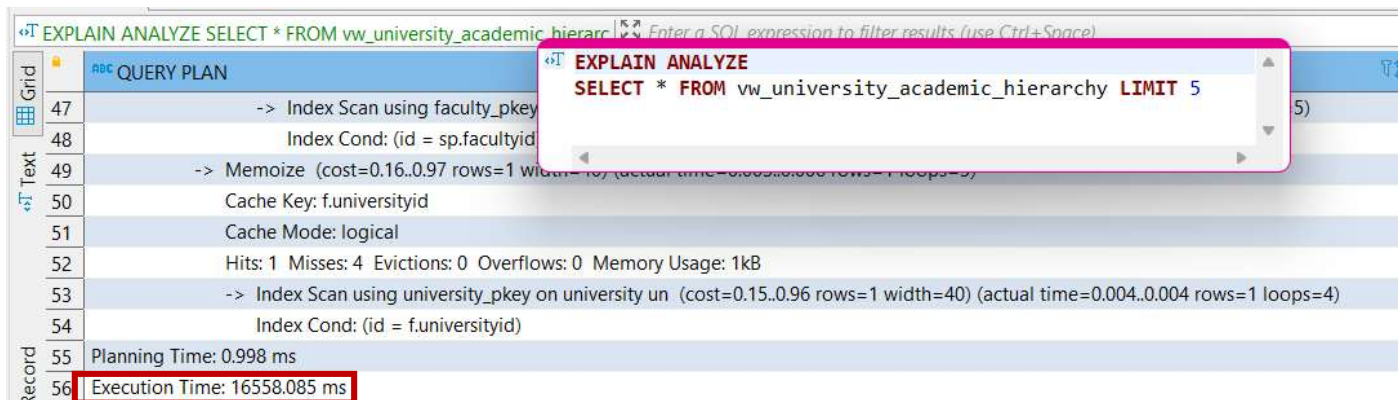
```
CREATE INDEX idx_subjectstudyprogram_subjectid  
ON Subject_StudyProgram(SubjectID);
```

```
CREATE INDEX idx_subjectstudyprogram_studyprogramid  
ON Subject_StudyProgram(StudyProgramID);
```

```
CREATE INDEX idx_usersubject_subjectid  
ON User_Subject(SubjectID);
```

```
CREATE INDEX idx_topicsuggestion_subjectid  
ON TopicSuggestion(SubjectID);
```

```
CREATE INDEX idx_mentorship_topicsuggestionid  
ON Mentorship(TopicSuggestionID);
```



The screenshot shows a database query plan for the query: `EXPLAIN ANALYZE SELECT * FROM vw_university_academic_hierarchy LIMIT 5`. The plan includes the following steps:

Step	Operation
47	-> Index Scan using faculty_pkey
48	Index Cond: (id = sp.facultyid)
49	-> Memoize (cost=0.16..0.97 rows=1 width=10; (actual time=0.003..0.004 rows=1 loops=1))
50	Cache Key: f.universityid
51	Cache Mode: logical
52	Hits: 1 Misses: 4 Evictions: 0 Overflows: 0 Memory Usage: 1kB
53	-> Index Scan using university_pkey on university un (cost=0.15..0.96 rows=1 width=40) (actual time=0.004..0.004 rows=1 loops=4)
54	Index Cond: (id = f.universityid)
55	Planning Time: 0.998 ms
56	Execution Time: 16558.085 ms

6. Време по индексирање

JOIN операциите работат значително побрзо поради користење index scans.

7. Преуредување на query

Прашалникот е оптимизиран со:

- користење CTE (SubjectStats),
- претходна агрегација пред главниот SELECT.

8. Резултат

Погледот е оптимизиран за работа со поголеми datasets.

View 5:

1. Примарен филтер

Примарен филтер ќе биде според:

- MentorshipID

2. Случај на употреба

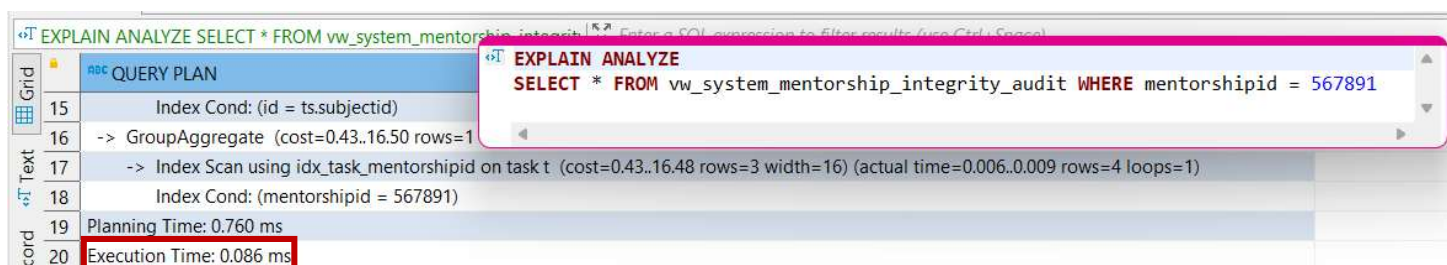
Погледот ќе се користи за:

- audit проверки,
- проверка на integrity на системот,
- детектирање ghost records,
- проверка на невалидни менторства.

3. Иницијално време на извршување

Иницијалното време е средно поради:

- повеќе LEFT JOIN,
- COALESCE,
- CASE проверки.



The screenshot shows a database query plan for the query: `EXPLAIN ANALYZE SELECT * FROM vw_system_mentorship_integrity_audit WHERE mentorshipid = 567891`. The plan is displayed in a table with the following rows:

Row	Operation
15	Index Cond: (id = ts.subjectid)
16	-> GroupAggregate (cost=0.43..16.50 rows=1)
17	-> Index Scan using idx_task_mentorshipid on task t (cost=0.43..16.48 rows=3 width=16) (actual time=0.006..0.009 rows=4 loops=1)
18	Index Cond: (mentorshipid = 567891)
19	Planning Time: 0.760 ms
20	Execution Time: 0.086 ms

4. Најбавни операции

Најбавни операции се:

- sequential scan на Task,
- aggregation операции.

5. Индекси

```
CREATE INDEX idx_mentorship_studentid
ON Mentorship(StudentID);

CREATE INDEX idx_mentorship_mentorid
ON Mentorship(MentorID);

CREATE INDEX idx_mentorship_topicsuggestionid
ON Mentorship(TopicSuggestionID);

CREATE INDEX idx_task_mentorshipid
ON Task(MentorshipID);
```

Grid	Text
15	Index Cond: (id = ts.subjectid)
16	-> GroupAggregate (cost=0.43..16.50 rows=1 width=)
17	-> Index Scan using idx_task_mentorshipid on task t (cost=0.43..16.48 rows=3 width=16) (actual time=0.005..0.007 rows=4 loops=1)
18	Index Cond: (mentorshipid = 567891)
19	Planning Time: 0.596 ms
20	Execution Time: 0.068 ms

6. Време по индексирање

JOIN операциите работат побрзо и се намалува бројот на sequential scans.

7. Преуредување на query

Прашалникот е оптимизиран со:

- CTE (TaskCount),
- aggregation пред главниот query.

8. Резултат

Погледот има прифатливи перформанси за административен audit систем.

View 6:

1. Примарен филтер

Примарен филтер ќе биде според:

- chat_id
- студент
- ментор

2. Случај на употреба

Погледот ќе се користи за:

- анализа на chat активност,
- приказ на последна порака,
- број на unread пораки,
- број на вкупни пораки.

3. Иницијално време на извршување

Иницијалното време е повисоко поради:

- LATERAL JOIN,
- aggregation врз Message,
- сортирање според timestamp.

Grid	Text
50	Worker 0: Sort M
51	Worker 1: Sort M
52	-> Partial HashAggregate (cost=442156.69..484019.99 rows=931122 width=16) (actual time=1519.883..2434.688 rows=940310 loops=3)
53	Group Key: message_1.chatid
54	Planned Partitions: 16 Batches: 17 Memory Usage: 8337kB Disk Usage: 48040kB
55	Worker 0: Batches: 17 Memory Usage: 8337kB Disk Usage: 112728kB
56	Worker 1: Batches: 17 Memory Usage: 8337kB Disk Usage: 112752kB
57	-> Parallel Seq Scan on message message_1 (cost=0.00..207781.67 rows=416667 width=8) (actual time=0.081..431.809 rows=333333 loops=3)
58	Planning Time: 0.994 ms
59	Execution Time: 19976.269 ms

4. Најбавни операции

Најбавни операции се:

- sequential scan на Message,
- sorting операции врз timestamp.

5. Индекси

```
CREATE INDEX idx_message_chatid
ON Message(ChatID);
```

```
CREATE INDEX idx_message_timestamp
ON Message(Timestamp DESC);
```

```
CREATE INDEX idx_message_isread
ON Message(isRead);
```

```
CREATE INDEX idx_chat_studentid
ON Chat(StudentID);
```

```
CREATE INDEX idx_chat_mentorid
ON Chat(MentorID);
```

EXPLAIN ANALYZE SELECT * FROM vw_chat_analytics LIMIT 5	
Grid	ABC QUERY PLAN
21	Group Key: message.chatid
22	-> Index Scan using idx_message_chatid
23	Filter: (NOT isread)
24	Rows Removed by Filter: 18
25	-> GroupAggregate (cost=0.43..293867.65 rows=931122 width=16) (actual time=0.313..0.326 rows=5 loops=1)
26	Group Key: message_1.chatid
27	-> Index Only Scan using idx_message_chatid on message message_1 (cost=0.43..234556.43 rows=1000000 width=8) (actual time=0.300..0.31
28	Heap Fetches: 3
29	Planning Time: 1.770 ms
Record	0 Execution Time: 1.292 ms

6. Време по индексирање

По индексирањето:

- пребарувањето на последна порака е побрзо,
- unread checks се оптимизирани,
- aggregation операциите работат побрзо.

7. Преуредување на query

Прашалникот е оптимизиран со:

- користење LATERAL JOIN,
- поделба на aggregation логика во посебни subqueries.

8. Резултат

Погледот има стабилни и добри перформанси за realtime chat analytics.

View 7:

1. Примарен филтер

Примарен филтер ќе биде според:

- MentorID

2. Случај на употреба

Погледот ќе се користи за:

- анализа на workload на ментори,
- број на pending задачи,
- број на активни менторства.

3. Иницијално време на извршување

Иницијалното време е повисоко поради aggregation врз Task.

Line	Operation
14	-> Index Scan using idx_task_mentorshipid on Task (cost=4.45..4.46 rows=1 width=8) (actual time=0.012..0.012 rows=1 loops=1)
15	Index Cond: (mentorshipid = m_1.id)
16	Filter: (status = 0)
17	SubPlan 1
18	-> Aggregate (cost=4.45..4.46 rows=1 width=8) (actual time=0.012..0.012 rows=1 loops=1)
19	-> Index Only Scan using idx_mentorship_mentorid on mentorship m2 (cost=0.42..4.44 rows=1 width=0) (actual time=0.009..0.010 rows=2 loops=1)
20	Index Cond: (mentorid = m.id)
21	Heap Fetches: 0
22	Planning Time: 0.429 ms
23	Execution Time: 6.914 ms

4. Најбавни операции

Најбавна операција е sequential scan на Task.

5. Индекси

```
CREATE INDEX idx_mentorship_mentorid  
ON Mentorship (MentorID);
```

```
CREATE INDEX idx_task_mentorshipid  
ON Task (MentorshipID);
```

```
CREATE INDEX idx_task_status  
ON Task (Status);
```

Line	Operation
14	-> Index Scan using idx_task_mentorshipid on Task (cost=4.45..4.46 rows=1 width=8) (actual time=0.004..0.004 rows=1 loops=1)
15	Index Cond: (mentorshipid = m_1.id)
16	Filter: (status = 0)
17	SubPlan 1
18	-> Aggregate (cost=4.45..4.46 rows=1 width=8) (actual time=0.004..0.004 rows=1 loops=1)
19	-> Index Only Scan using idx_mentorship_mentorid on mentorship m2 (cost=0.42..4.44 rows=1 width=0) (actual time=0.003..0.004 rows=2 loops=1)
20	Index Cond: (mentorid = m.id)
21	Heap Fetches: 0
22	Planning Time: 0.396 ms
23	Execution Time: 0.087 ms

6. Време по индексирање

Pending task checks се извршуваат побрзо.

7. Преуредување на query

Прашалникот е подобрен со:

- користење СТЕ (`PendingTasks`),
- aggregation пред главниот query.

8. Резултат

Погледот има стабилни и прифатливи перформанси.

View 8:

1. Примарен филтер

Примарен филтер ќе биде според:

- предмет

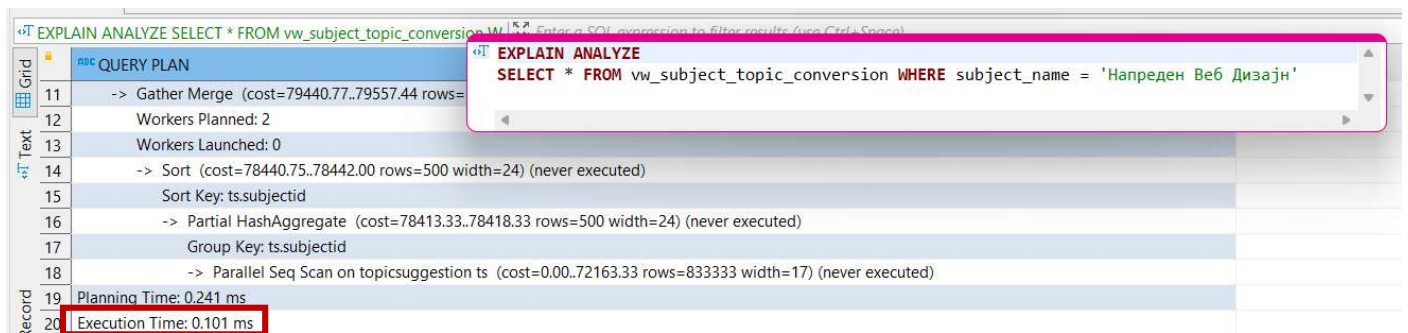
2. Случај на употреба

Погледот ќе се користи за:

- анализа на conversion rate на теми,
- број на предложени теми,
- број на прифатени теми.

3. Иницијално време на извршување

Aggregation врз `TopicSuggestion` предизвикува поголемо време на извршување.



Grid	SQL QUERY PLAN
11	-> Gather Merge (cost=79440.77..79557.44 rows=
12	Workers Planned: 2
13	Workers Launched: 0
14	-> Sort (cost=78440.75..78442.00 rows=500 width=24) (never executed)
15	Sort Key: ts.subjectid
16	-> Partial HashAggregate (cost=78413.33..78418.33 rows=500 width=24) (never executed)
17	Group Key: ts.subjectid
18	-> Parallel Seq Scan on topicsuggestion ts (cost=0.00..72163.33 rows=833333 width=17) (never executed)
19	Planning Time: 0.241 ms
20	Execution Time: 0.101 ms

4. Најбавни операции

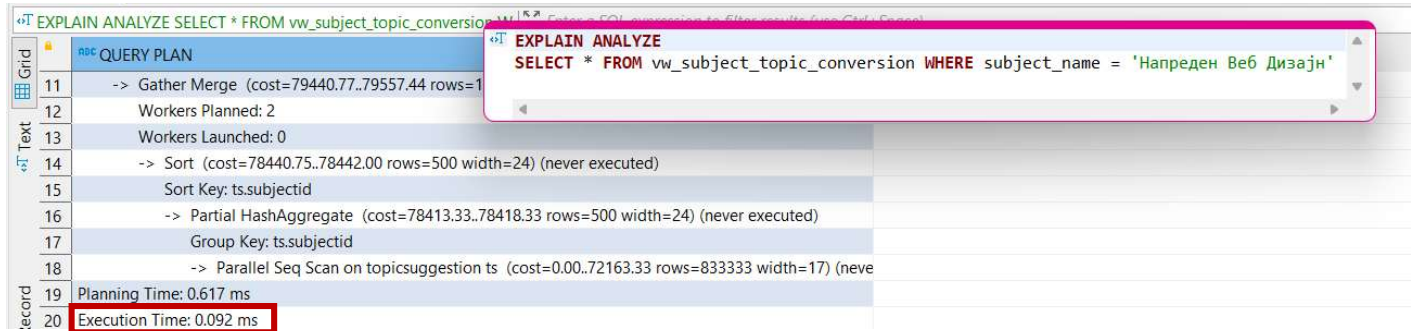
Најбавна операција е scan на `TopicSuggestion`.

5. Индекси

```
CREATE INDEX idx_topicsuggestion_subjectid  
ON TopicSuggestion(SubjectID);
```

```
CREATE INDEX idx_topicsuggestion_available  
ON TopicSuggestion(isAvailable);
```

```
CREATE INDEX idx_topicsuggestion_subject_available  
ON TopicSuggestion(SubjectID, isAvailable);
```



Step	Operation	Cost	Rows	Width	Notes
11	-> Gather Merge	(cost=79440.77..79557.44)	rows=1		
12	Workers Planned: 2				
13	Workers Launched: 0				
14	-> Sort	(cost=78440.75..78442.00)	rows=500	width=24	(never executed)
15	Sort Key: ts.subjectid				
16	-> Partial HashAggregate	(cost=78413.33..78418.33)	rows=500	width=24	(never executed)
17	Group Key: ts.subjectid				
18	-> Parallel Seq Scan on topicsuggestion ts	(cost=0.00..72163.33)	rows=833333	width=17	(never executed)
19	Planning Time: 0.617 ms				
20	Execution Time: 0.092 ms				

6. Време по индексирање

Aggregation операциите се извршуваат побрзо.

7. Преуредување на query

Прашалникот е оптимизиран со:

- CTE (TopicCounts),
- користење COALESCE и NULLIF.

8. Резултат

Погледот има подобрени перформанси.

View 9:

1. Примарен филтер

Примарен филтер ќе биде според:

- универзитет,
- факултет.

2. Случај на употреба

Погледот ќе се користи за:

- приказ на академска статистика,
- број на студиски програми по факултет.

3. Иницијално време на извршување

Иницијалното време е релативно мало бидејќи погледот користи мал број табели.

Step	Operation	Cost	Actual Time
11	Sort Key: sp.facultyid, s		
12	Sort Method: quicksort		
13	Seq Scan on studyprogram sp	(cost=0.00.6.00 rows=200 width=16)	(actual time=0.006.0.00)
14	Hash	(cost=2.75.2.75 rows=3 width=65)	(actual time=0.017.0.017 rows=3 loops=1)
15	Buckets: 1024 Batches: 1 Memory Usage: 9kB		
16	Seq Scan on faculty f	(cost=0.00.2.75 rows=3 width=65)	(actual time=0.009.0.014 rows=3 loop=1)
17	Filter: (name = 'Факултет за информатички науки и компјутерско инженерство':text)		
18	Rows Removed by Filter: 57		
19	Planning Time: 0.218 ms		
20	Execution Time: 0.176 ms		

4. Најбавни операции

Најбавна операција е aggregation врз StudyProgram.

5. Индекси

```
CREATE INDEX idx_faculty_universityid
ON Faculty(UniversityID);
```

```
CREATE INDEX idx_studyprogram_facultyid
ON StudyProgram(FacultyID);
```

Step	Operation	Cost	Actual Time
11	Sort Key: sp.facultyid, sp.id		
12	Sort Method: quicksort Memory: 31kB		
13	Seq Scan on studyprogram sp	(cost=0.00.6.00 rows=200 width=16)	(actual time=0.009.0.025 rows=3 loops=1)
14	Hash	(cost=2.75.2.75 rows=3 width=65)	(actual time=0.046.0.046 rows=3 loops=1)
15	Buckets: 1024 Batches: 1 Memory Usage: 9kB		
16	Seq Scan on faculty f	(cost=0.00.2.75 rows=3 width=65)	(actual time=0.036.0.041 rows=3 loops=1)
17	Filter: (name = 'Факултет за информатички науки и компјутерско инженерство':text)		
18	Rows Removed by Filter: 57		
19	Planning Time: 0.327 ms		
20	Execution Time: 0.223 ms		

6. Време по индексирање

Aggregation операциите работат побрзо.

7. Преуредување на query

Прашалникот е оптимизиран со:

- користење CTE (ProgramCounts),
- aggregation пред главниот query.

8. Резултат

Погледот има стабилни и прифатливи перформанси.

View 10:

1. Примарен филтер

Примарен филтер ќе биде според:

- корисник,
- notification status,
- timestamp.

2. Случај на употреба

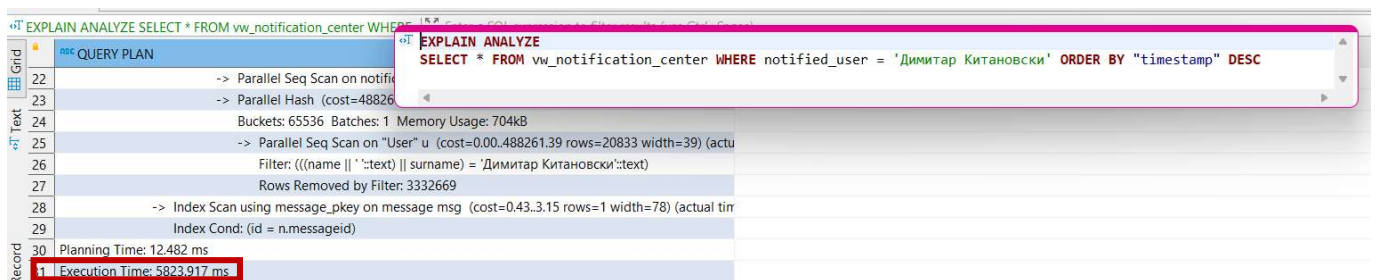
Погледот ќе се користи за:

- приказ на notifications,
- unread notifications,
- поврзување со пораки и коментари,
- monitoring на активности.

3. Иницијално време на извршување

Иницијалното време е повисоко поради:

- повеќе LEFT JOIN операции,
- поврзување со Message и CommentOpinion,
- пребарување според timestamp.



22	EXPLAIN ANALYZE SELECT * FROM vw_notification_center WHERE notified_user = 'Димитар Китановски' ORDER BY "timestamp" DESC
23	QUERY PLAN
24	-> Parallel Seq Scan on notification_center
25	-> Parallel Hash (cost=488261.39 rows=1 width=78) (actual time=0.00..0.01 rows=1 width=78)
26	Buckets: 65536 Batches: 1 Memory Usage: 704kB
27	-> Parallel Seq Scan on "User" u (cost=0.00..488261.39 rows=20833 width=39) (actual time=0.00..0.01 rows=1 width=39)
28	Filter: (((name ' '::text) surname) = 'Димитар Китановски '::text)
29	Rows Removed by Filter: 3332669
30	-> Index Scan using message_pkey on message msg (cost=0.43..3.15 rows=1 width=78) (actual time=0.00..0.01 rows=1 width=78)
31	Index Cond: (id = n.messageid)
32	Planning Time: 12.482 ms
33	Execution Time: 5823.917 ms

4. Најбавни операции

Најбавни операции се:

- sequential scan на Notification,

- JOIN операции со Message,
- JOIN операции со CommentOpinion.

5. Индекси

```
CREATE INDEX idx_notification_usertonotifyid
ON Notification(UserToNotifyID);
```

```
CREATE INDEX idx_notification_timestamp
ON Notification(Timestamp);
```

```
CREATE INDEX idx_notification_isread
ON Notification(isRead);
```

```
CREATE INDEX idx_notificationtype_notificationid
ON Notification_Type(NotificationID);
```

```
CREATE INDEX idx_message_id
ON Message(ID);
```

```
CREATE INDEX idx_commentopinion_id
ON CommentOpinion(ID);
```

Step	Operation	Cost	Rows	Width	Actual Time	Actual Rows	Loops
20	Nested Loop	0.43..665.00	2	88	0.085..0.100	2	1
21	Parallel Seq Scan on "UserToNotifyID"						
22	Filter: (((name '':text) surname) = 'Димитар Китановски':text)						
23	Rows Removed by Filter: 3332669						
24	Index Scan using idx_notification_usertonotifyid on notification n	0.43..8.50	2	88	0.085..0.100	2	1
25	Index Cond: (usertonotifyid = u.id)						
26	Index Scan using message_pkey on message msg	0.43..3.15	1	78	0.454..0.454	1	1966
27	Index Cond: (id = n.messageid)						
28	Planning Time: 3.344 ms						
29	Execution Time: 11373.919 ms						

6. Време по индексирање

По индексирањето:

- notifications се пребаруваат побрзо,
- unread checks се оптимизирани,
- JOIN операциите користат index scan.

7. Преуредување на query

Нема потреба од дополнително преуредување бидејќи логиката е веќе оптимизирана преку директни JOIN операции.

8. Резултат

Погледот има стабилни и прифатливи перформанси.