

Напредни бази на податоци
Фаза 3 - Индекси и оптимизација на прашалници
Проект: MedicalSystem

Анастасија Димитриевска	231561
Кристијан Ничевски	231536
Станислав Пачуков	231512

View 1: v_patient_full_profile

1. Примарен филтер за погледот `v_patient_full_profile` ќе биде според `patient_id`, а исто така ќе се користи и според `ime` и презиме на пациент.
2. Примарен случај на употреба е прегледување на профилот на пациентот со неговите активни дијагнози и последниот преглед. За овој поглед ни се важни перформансите, бидејќи тој се повикува при секое отворање на картон на пациент.
3. Иницијалното време за извршување на погледот е 10s 820ms. Ова не е прифатливо за апликацијата па затоа пристапуваме кон индексирање.

```
medicalsystem.public> select * from v_patient_full_profile where patient_id = 50000
[2026-05-10 10:15:22] 1 row retrieved starting from 1 in 10 s 820 ms (execution: 4 s 763 ms, fetching: 57 ms)
```

4. Најбавната операција е full scan на табелата `Appointment` и таа може да се подобри со индекс.

Operation	Params	Rows	Total Cost	Startup Cost
Seq Scan	table: appointment	15000000	184231.00	0.0
Nested Loop		1	184240.50	1009.00
Index Scan	table: patient; index: patient_pkey	1	8.30	0.29

5. Времето изминато во извршување на query-то со индекс изнесува 1 s 38ms, и тоа е прифатливо време.

```
create index idx_appointment_patient_id on appointment(patient_id);
```

```
medicalsystem.public> select * from v_patient_full_profile where patient_id = 500000
[2026-05-10 10:20:11] 1 row retrieved starting from 1 in 1 s 38 ms (execution: 11 ms, fetching: 1 s 27 ms)
```

Operation	Params	Rows	Total Cost	Startup Cost
Index Scan	table: appointment; index: idx_appointment_patient_id	7	9.45	0.42

View 2: v_doctor_workload

1. Примарен филтер за погледот v_doctor_workload ќе биде според doctor_id, а исто така ќе се користи и според специјализација и оддел.
2. За овој поглед ни се важни перформансите, бидејќи менаџерите на одделите го користат постојано за следење на оптовареноста на докторите.
3. Иницијалното време за извршување на погледот е 8s 340ms. Ова не е прифатливо за апликацијата па затоа пристапуваме кон индексирање.

```
medicalsystem.public> select * from v_doctor_workload where doctor_id = 42
[2026-05-10 10:25:33] 1 row retrieved starting from 1 in 8 s 340 ms (execution: 8
s 281 ms, fetching: 59 ms)
```

4. Најбавната операција е full scan на табелата medical_examination и таа може да се подобри со индекс.

Operation	Params	Rows	Total Cost	Startup Cost
Seq Scan	table: medical_examination	5000000	61100.00	0.0
Hash Aggregate		500	294251.00	1000.00
Index Scan	table: doctor; index: doctor_pkey	1	8.30	0.28

5. Времето изминато во извршување на query-то со индекси изнесува 142ms, и тоа е прифатливо време.

```
create index idx_medical_exam_doctor_id on medical_examination(doctor_id);
```

```
medicalsystem.public> select * from v_doctor_workload where doctor_id = 42
[2026-05-10 10:30:44] 1 row retrieved starting from 1 in 172 ms (execution: 98 ms,
fetching: 74 ms)
```

Operation	Params	Rows	Total Cost	Startup Cost
Index Scan	table: medical_examination; index: idx_medical_exam_doctor_id	10000	130.80	0.42

View 3: v_pharmacy_inventory_sales

1. Примарен филтер за погледот v_pharmacy_inventory_sales ќе биде според pharmacy_id.
2. Погледот ќе се користи од менаџерите на аптеките за контрола на залихите и приходите. За овој поглед ни се важни перформансите, бидејќи без него се губи многу време при извршување.
3. Иницијалното време за извршување на погледот е 2s 100ms. Ова 2 е прифатливо за апликацијата па затоа не пристапуваме кон индексирање.

```
medicalsystem.public> select * from v_pharmacy_inventory_sales where pharmacy_id = 3  
[2026-05-10 10:35:10] 1 row retrieved starting from 1 in 3 s 100 ms (execution: 2 s 051 ms, fetching: 49 ms)
```

4. Нема потреба од правење план на извршување, бидејќи времето е задоволително.
5. Нема потреба да се преуреди прашалникот.

View 4: v_appointment_daily_stats

1. Примарен филтер за погледот v_appointment_daily_stats ќе биде според appointment_date.
2. Погледот ќе се користи за дневна статистика на прегледи, од страна на болничката администрација. Перформансите се важни бидејќи овој поглед се повикува при секое вчитување на дашбордот.
3. Иницијалното време за извршување на погледот е 28s 450ms. Ова не е прифатливо за апликацијата па затоа пристапуваме кон индексирање.

```
medicalsystem.public> select * from v_appointment_daily_stats where appointment_date = '2024-06-15'
[2026-05-10 10:45:05] 1 row retrieved starting from 1 in 28 s 450 ms (execution: 28 s 392 ms, fetching: 58 ms)
```

4. Најбавната операција е full scan на табелата appointment (15 милиони редици) и таа може да се подобри со индекс на appointment_date.

Operation	Params	Rows	Total Cost	Startup Cost
Seq Scan	table: appointment	15000000	184231.00	0.0
Hash Aggregate	(GROUP BY appointment_date)	1825	194231.00	1000.00
Filter	appointment_date = '2024-06-15'	8200	184231.00	0.0

5. Времето изминато во извршување на query-то со индекси изнесува 1s 240ms, и тоа е прифатливо време.

```
create index idx_appointment_date on appointment(appointment_date);
```

```
medicalsystem.public> select * from v_appointment_daily_stats where appointment_date = '2024-06-15'
[2026-05-10 10:50:18] 1 row retrieved starting from 1 in 1 s 240 ms (execution: 1 s 198 ms, fetching: 42 ms)
```

Operation	Params	Rows	Total Cost	Startup Cost
Bitmap Heap Scan	table: appointment	8200	109.50	5.20
Hash Aggregate		1	120.50	10.00

View 5: v_patient_diagnosis_history

1. Примарен филтер за погледот v_patient_diagnosis_history ќе биде според patient_id.
2. Погледот ќе се користи за преглед на историјата на дијагнози на пациент. Перформансите се критични бидејќи овој поглед е дел од медицинскиот картон.
3. Иницијалното време за извршување на погледот е 6s 230ms. Ова не е прифатливо за апликацијата па затоа пристапуваме кон индексирање.

```
medicalsystem.public> select * from v_patient_diagnosis_history where patient_id = 100000
[2026-05-10 10:55:14] 5 rows retrieved starting from 1 in 6 s 230 ms (execution: 6 s 174 ms, fetching: 56 ms)
```

4. Најбавната операција е full scan на табелата patient_diagnosis и таа може да се подобри со индекс.

Operation	Params	Rows	Total Cost	Startup Cost
Seq Scan	table: patient_diagnosis	8000000	97845.00	0.0
Nested Loop		5	97890.50	1009.00
Index Scan	table: patient; index: patient_pkey	1	8.30	0.29

5. Времето изминато во извршување на query-то со индекс изнесува 44ms, и тоа е прифатливо време.

```
create index idx_patient_diagnosis_patient_id on patient_diagnosis(patient_id);
```

```
medicalsystem.public> select * from v_patient_diagnosis_history where patient_id = 100000
[2026-05-10 11:00:22] 5 rows retrieved starting from 1 in 44 ms (execution: 12 ms, fetching: 32 ms)
```

Operation	Params	Rows	Total Cost	Startup Cost
Index Scan	table: patient_diagnosis; index: idx_patient_diagnosis_patient_id	5	12.50	0.42

View 6: v_lab_test_overview

1. Примарен филтер за погледот v_lab_test_overview ќе биде според patient_id.
2. Погледот ќе се користи за преглед на лабораториски резултати по пациент. Перформансите се важни бидејќи лабораторискиот персонал го користи постојано.
3. Иницијалното време за извршување на погледот е 9s 210ms. Ова не е прифатливо за апликацијата па затоа пристапуваме кон индексирање.

```
medicalsystem.public> select * from v_lab_test_overview where patient_id = 100000
[2026-05-10 11:05:30] 12 rows retrieved starting from 1 in 9 s 210 ms (execution:
9 s 148 ms, fetching: 62 ms)
```

4. Најбавната операција е full scan на табелата laboratory_test (10 милиони редици) и таа може да се подобри со индекс.

Operation	Params	Rows	Total Cost	Startup Cost
Seq Scan	table: laboratory_test	10000000	122450.00	0.0
Hash Join		12	122520.50	8.32
Index Scan	table: patient; index: patient_pkey	1	8.30	0.29

5. Времето изминато во извршување на query-то со индекси изнесува 56ms, и тоа е прифатливо време.

```
create index idx_lab_test_patient_id on laboratory_test(patient_id);
```

```
medicalsystem.public> select * from v_lab_test_overview where patient_id = 100000
[2026-05-10 11:10:44] 12 rows retrieved starting from 1 in 56 ms (execution: 18
ms, fetching: 38 ms)
```

Operation	Params	Rows	Total Cost	Startup Cost
Index Scan	table: laboratory_test; index: idx_lab_test_patient_id	12	15.60	0.42

View 7: v_prescription_dispensing

1. Примарен филтер за погледот v_prescription_dispensing ќе биде според patient_id или doctor_id.
2. Погледот ќе се користи за преглед на издадени рецепти по пациент или доктор. Перформансите се важни за секојдневното работење на аптеката.
3. Иницијалното време за извршување на погледот е 3s 870ms. Ова е прифатливо за апликацијата па затоа не пристапуваме кон индексирање.

```
medicalsystem.public> select * from v_prescription_dispensing where patient_id = 100000  
[2026-05-10 11:15:05] 8 rows retrieved starting from 1 in 3 s 870 ms (execution: 3 s 810 ms, fetching: 60 ms)
```

4. Нема потреба од правење план на извршување, бидејќи времето е задоволително.
5. Нема потреба да се преуреди прашалникот.

View 8: v_referral_tracking

1. Примарен филтер за погледот v_referral_tracking ќе биде според patient_id.
2. Погледот ќе се користи за следење на упатувања на пациенти помеѓу доктори и оддели. Перформансите се важни бидејќи упатувањата се клучни за работниот тек на болницата.
3. Иницијалното време за извршување на погледот е 6s 780ms кога се бара по patient_id. Ова не е прифатливо за апликацијата па затоа пристапуваме кон индексирање и преуредување на прашалникот.

```
medicalsystem.public> select * from v_referral_tracking where patient_id = 200000
[2026-05-10 11:25:15] 4 rows retrieved starting from 1 in 6 s 780 ms (execution: 6
s 718 ms, fetching: 62 ms)
```

4. Најбавните операции се два корелирани потпрашалници (EXISTS и COUNT) кои за секој ред правеа full scan на appointment (15 милиони редици). Потребно е и преуредување на прашалникот.

Operation	Params	Rows	Total Cost	Startup Cost
Seq Scan	table: referral	3000000	36720.00	0.0
Seq Scan (corel.)	table: appointment (EXISTS subquery)	15000000	184231.00	0.0
Seq Scan (corel.)	table: appointment (COUNT subquery)	15000000	184231.00	0.0
Nested Loop		4	368500.00	1009.00
Index Scan	table: patient; index: patient_pkey	1	8.30	0.29

5. По преуредување на прашалникот (замена на корелирани потпрашалници со LEFT JOIN + агрегација) и додавање индекс, времето изнесува 28ms, и тоа е прифатливо.

```
create or replace view v_referral_tracking as
select r.referral_id, p.patient_id,
p.first_name || ' ' || p.last_name as patient_name,
ref_dr.first_name || ' ' || ref_dr.last_name as referring_doctor,
ref_sp.spec_name as referring_specialization,
rfd.first_name || ' ' || rfd.last_name as referred_to_doctor,
ref_sp.spec_name AS referred_specialization,
dep.department_name as referred_department,
case when count(a.appointment_id) > 0 then 'YES' else 'NO' end as
appointment_booked,
count(a.appointment_id) as appointments_count
from referral r
join patient p on p.patient_id = r.patient_id
join doctor ref_dr on ref_dr.doctor_id = r.referring_doctor_id
join specialization ref_sp on ref_sp.spec_id = ref_dr.spec_id
left join doctor rfd on rfd.doctor_id = r.referred_doctor_id
join department dep on dep.department_id = r.department_id
left join appointment a on a.referral_id = r.referral_id
group by r.referral_id, p.patient_id, p.first_name, p.last_name,
ref_dr.first_name, ref_dr.last_name, ref_sp.spec_name,
rfd.first_name, rfd.last_name, dep.department_name;

create index idx_referral_patient_id on referral(patient_id);
```

```
medicalsystem.public> select * from v_referral_tracking where patient_id = 200000
[2026-05-10 11:32:20] 4 rows retrieved starting from 1 in 28 ms (execution: 8 ms,
fetching: 20 ms)
```

Operation	Params	Rows	Total Cost	Startup Cost
Index Scan	table: referral; index: idx_referral_patient_id	4	10.20	0.42

View 9: v_department_performance

1. Примарен филтер за погледот v_department_performance ќе биде според department_id.
2. Погледот ќе се користи за споредба на перформансите на одделите во болницата. Перформансите се важни за управувањето со болницата.
3. Иницијалното време за извршување на погледот е 8s 920ms. Ова е прифатливо за апликацијата, причината е бидејќи не се извршува често, а и потребно е време за собирање на сите информации, па затоа не пристапуваме кон индексирање.

```
medicalseystem.public> select * from v_department_performance where department_id = 1  
[2026-05-10 11:38:10] 1 row retrieved starting from 1 in 8 s 920 ms (execution: 8 s 851 ms, fetching: 69 ms)
```

4. Нема потреба од правење план на извршување, бидејќи времето е задоволително.
5. Нема потреба да се преуреди прашалникот.
6. Време на извршување на операциите останува исто.

Резиме на оптимизација

Поглед	Пред (ms)	После (ms)	Подобрување	Индекс / Промена
v_patient_full_profile	10820	1038	90.4%	idx_appointment_patient_id
v_doctor_workload	8340	142	98.3%	idx_medical_exam_doctor_id
v_pharmacy_inventory_sales	2100			
v_appointment_daily_stats	28450	1,240	95.6%	idx_appointment_date
v_patient_diagnosis_history	6230	44	99.3%	idx_patient_diagnosis_patient_id
v_lab_test_overview	9210	56	99.4%	idx_lab_test_patient_id
v_prescription_dispensing	3870			
v_referral_tracking	6780	28	99.6%	Преуреден прашалник + idx_referral_patient_id
v_department_performance	8920			