

# Напредни бази на податоци

## Фаза 3 – Оптимизација на прашалници и погледи

### Име на проект: BankPaymentService

#### View 1: Најава на корисник (Login view)

1. **Филтри:** Примарен филтер за погледот `v_user_login` ќе биде според `username`, а дополнително се филтрира само на корисници со статус `ACTIVE`.
2. **Употреба:** е процесот на логирање на клиент во системот. Погледот обезбедува брз пристап до корисничкото име, и статусот на сметката.
3. **Иницијално време:** Иницијалното време за извршување на погледот е `0.315ms`. Ова е прифатливо време за апликацијата.

```
SELECT * FROM v_user_login WHERE username = 'client435156';
```

7	Planning Time: 43.319 ms
8	Execution Time: 0.315 ms

4. Најбавните операции се две Index Scan операции:  
Index Scan на табела `client_user` преку индексот `client_user_username_key` - Actual Total Time: `0.039ms`  
Index Scan на табела `client` преку индексот `client_pkey` - Actual Total Time: `0.013ms`  
Вкупниот Nested Loop завршува за само `0.056ms`. Бидејќи се работи за Index Scan (не Full Scan), операциите не можат значително да се подобрат со дополнителни индекси.

Operation	Params	Rows	Actual Rows	Total Cost	Actual Total Time	Startup Cost	Actual Startup Time	Raw Desc
Select								Planning Time ...
Nested Loop		1	1	16.88	0.056	0.84	0.054	Parallel Aware ...
Index Scan	table: client_user; index: client_user_username_key;	1	1	8.44	0.039	0.42	0.037	Parent Relation...
Index Scan	table: client; index: client_pkey;	1	1	8.44	0.013	0.42	0.013	Parent Relation...

Времето изминато во извршување на операциите `insert` и `update` изнесува

```
[2026-05-09 12:53:25] advdb_202526l_prj_bps_emj.public> EXPLAIN ANALYSE INSERT INTO Client_user (
    username,
    password_hash,
    last_login,
    status,
    client_id
)
VALUES (
    'client90593',
    '$2b$12$7sJ9KQ1rL8yN3vX2mPqE8uC1Wz4HkR5tY7uI2oP9aB6dF3gL0nM1Sa',
    CURRENT_TIMESTAMP,
    'ACTIVE',
    90593
)
[2026-05-09 12:53:25] 5 rows retrieved starting from 1 in 492 ms (execution: 73 ms, fetching: 419 ms)
```

Activate Windows

```
[2026-05-09 12:53:25] 5 rows retrieved starting from 1 in 492 ms (execution: 73 ms, fetching: 419 ms)
[2026-05-09 12:58:29] advdb_202526l_prj_bps_emj.public> UPDATE Client_user
SET
    password_hash = '$2b$12$7sJ9KQ1rL8yN3vX2mPqE8uC1Wz4HkR5tY7uI2oP9aB6dF3gL0nM1Ma'
WHERE username = 'client90593'
[2026-05-09 12:58:29] 1 row affected in 42 ms
```

5. Нема потреба да се преуреди прашалникот.
6. Време на извршување на операциите insert и update останува исто.

## View 2: Извештаи (Receipt view)

1. **Филтри:** Примарен филтер за погледот v\_client\_receipts ќе биде според client\_id, а дополнително може да се користи и според transaction\_date (по временски период - месец и година).
2. **Употреба:** Примарен случај на употреба е преглед на издадени сметки за клиент по одреден период. За овој поглед ни се важни перформансите, бидејќи без него се губи време при извршување.
3. **Иницијално време:** Иницијалното време за извршување на погледот е 705ms. Ова е прифатливо време за апликацијата.

```
SELECT * FROM v_client_receipts WHERE client_id = 60666;
24 Timing: Generation 10.914 ms (Deform 6.452 ms), Inlining
25 Execution Time: 705.433 ms
```

4. Нема потреба од правење план на извршување, бидејќи времето е задоволително. Времето на операциите insert и update е

```
[2026-05-09 13:35:14] advdb_202526l_prj_bps_emj> EXPLAIN ANALYSE INSERT INTO Transaction (
    amount,
    transaction_date,
    status,
    description,
    reference_number,
    account_id,
    transaction_type_id
)
VALUES (
    1250.50,
    '2025-04-15 11:32:00.654321',
    'COMPLETED',
    'Removing funds from an account',
    'TRX-0010000001',
    40000,
    2
)
[2026-05-09 13:35:14] 13 rows retrieved starting from 1 in 522 ms (execution: 77 ms, fetching: 445 ms)
```

```
[2026-05-09 13:44:21] advdb_202526l_prj_bps_emj.public> INSERT INTO Receipt (
    print_time,
    receipt_number,
    transaction_id
)
VALUES (
    '2025-04-15 11:50:00.654321',
    'RCPT-20250415-010000001',
    currval('transaction_transaction_id_seq')
)

[2026-05-09 13:44:21] 1 row affected in 126 ms
```

```
[2026-05-09 14:04:19] advdb_202526l_prj_bps_emj.public> UPDATE Transaction
SET
    amount = 1300.00
WHERE reference_number = 'TRX-0010000001'

[2026-05-09 14:04:19] 1 row affected in 44 ms
```

5. Нема потреба да се преуреди прашалникот.
6. Време на извршување на операциите insert и update останува исто.

### View 3: Рати (Loan installments view)

1. **Филтри:** Примарен филтер за погледот v\_loan\_installments ќе биде според loan\_id.
2. **Употреба:** Примарен случај на употреба е кога клиентот или банкарскиот персонал сака да ги прегледа ратите за конкретен кредит. За овој поглед ни се важни перформансите.
3. **Иницијално време:** Иницијалното време за извршување на погледот е 250.5ms. Ова е прифатливо време за апликацијата.

SELECT *	10	Planning Time: 0.725 ms
FROM v_loan_installments WHERE loan_id=46000;	11	Execution Time: 250.531 ms

4. Нема потреба од правење план на извршување, бидејќи времето е задоволително. Пребарувањето по loan\_id секогаш враќа мал број рати за конкретен кредит, па табелата не се пребарува со висока фреквенција истовремено. Времето на операциите insert и update изнесува

```
[2026-05-09 15:05:12] advdb_202526l_prj_bps_emj> INSERT INTO Loan (
    loan_type,
    amount,
    interest_rate,
    start_date,
    status,
    approved_date,
    client_id,
    employee_id
)
VALUES (
    'LOAN',
    114898.29,
    49,
    '2026-05-04',
    'APPROVED',
    '2026-05-04',
    318290,
    2242
)

[2026-05-09 15:05:12] 1 row affected in 252 ms
```

```
[2026-05-09 15:17:43] advdb_202526l_prj_bps_emj.public> UPDATE Loan
SET
    approved_date = '2026-05-03'
WHERE client_id = 318290
    AND employee_id = 2242
    AND start_date = '2026-05-04'

[2026-05-09 15:17:43] 1 row affected in 63 ms
```

5. Нема потреба да се преуреди прашалникот.
6. Времето на извршување на операциите insert и update останува исто.

#### View 4: Преглед на кредити

1. **Филтри:** Примарен филтер за погледот v\_loan\_details ќе биде според loan\_id, а исто така и според client\_id.
2. **Употреба:** Примарен случај на употреба е целосен преглед на кредитна апликација. За овој поглед ни се важни перформансите, бидејќи без него се губи многу време при извршување.
3. **Иницијално време:** Иницијалното време за извршување на погледот е 1s 735ms. Времето е побавно поради повеќе JOIN операции и full scan на табелите Account и Collateral. Ова не е прифатливо за апликацијата.

SELECT *	28	-> Seq Scan on branch b
FROM v_loan_details;	29	Planning Time: 5.148 ms
	30	Execution Time: 1735.876 ms

4. Најбавните операции се JOIN-овите на Account и Collateral преку loan\_id, кои немаат индекси на надворешниот клуч.

Operation	Params	Rows	Actual Rows	Total Cost	Actual Total Time	Startup Cost	Actual Startup Time	Raw Desc
▼ Select								Planning Time ...
▼ Hash Join		1000000	90504	75197.19	1468.617	36231.72	1074.986	Parallel Aware ...
▼ Hash Join		1000000	90504	72458.77	1449.734	36228.44	1074.915	Parent Relations...
Full Scan (Seq Scan)	table: account;	1000000	1000000	22464.0	242.133	0.0	0.039	Parent Relations...
Transformation (Hash)		86960	86961	33272.44	1074.751	33272.44	1074.742	Parent Relations...
▼ Hash Join		86960	86961	33272.44	974.072	18052.89	155.177	Parent Relations...
▼ Hash Join		86960	86961	29107.14	850.538	15083.29	96.958	Parent Relations...
▼ Merge Join		86960	86961	25744.86	756.884	11949.29	45.485	Parent Relations...
Index Scan	table: client; index: client_pkey;	500000	318291	69260.41	587.277	0.42	0.052	Parent Relations...
▼ Temporary Merge Join		86960	86961	12383.63	85.339	11948.83	45.42	Parent Relations...
▼ Sort		86960	86961	12166.23	73.195	11948.83	45.413	Parent Relations...
Full Scan (Seq Scan)	table: loan;	86960	86961	1839.6	15.899	0.0	0.037	Parent Relations...
Transformation (Hash)		100000	100000	1884.0	51.305	1884.0	51.303	Parent Relations...
Full Scan (Seq Scan)	table: employee;	100000	100000	1884.0	20.88	0.0	0.033	Parent Relations...
Transformation (Hash)		86960	86960	1882.6	57.149	1882.6	57.148	Parent Relations...
Full Scan (Seq Scan)	table: collateral;	86960	86960	1882.6	23.099	0.0	0.035	Parent Relations...
Transformation (Hash)		101	101	2.01	0.051	2.01	0.05	Parent Relations...
Full Scan (Seq Scan)	table: branch;	101	101	2.01	0.03	0.0	0.015	Parent Relations...

Времето изминато во извршување на операциите insert и update изнесува

```
[2026-05-09 17:43:31] advdb_202526l_prj_bps_emj> INSERT INTO Collateral (
    type,
    description,
    value,
    status,
    date_added,
    loan_id,
    client_id
)
VALUES (
    'REAL_ESTATE',
    'Apartment collateral',
    85000.00,
    'PLEGDED',
    '2026-05-09',
    86961,
    318290
)

[2026-05-09 17:43:31] 1 row affected in 106 ms
```

```
[2026-05-09 17:48:19] advdb_202526l_prj_bps_emj.public> UPDATE Collateral
SET
    value = 92000.00
WHERE loan_id = 86961

[2026-05-09 17:48:19] 1 row affected in 116 ms
```

5. Времето изминато во извршување на query-то со индекси изнесува 781ms, и тоа е прифатливо време.

```
CREATE INDEX idx_account_loan_id
ON Account(loan_id);
```

```
SELECT *
FROM v_loan_details;
```

33	Planning Time: 1.378 ms
34	Execution Time: 781.348 ms

Operation	Params	Rows	Actual Rows	Total Cost	Actual Total Time	Startup Cost	Actual Startup Time	Raw Desc
↳ Select								Planning Time...
↳ Merge Join		10000...	90504	51117.35	810.221	46948.99	698.641	Parallel Aware...
↳ Nested Loops (Nested Loop)		10000...	34837	92989.12	40.99	0.58	0.047	Parent Relatio...
↳ Index Scan	table: account; index: idx_account_loan...	10000...	34837	68118.37	21.032	0.42	0.019	Parent Relatio...
↳ Unknown (Memoize)		1	1	0.17	0.0	0.15	0.0	Parent Relatio...
↳ Index Scan	table: branch; index: branch_pkey;	1	1	0.16	0.001	0.14	0.001	Parent Relatio...
↳ Temporary (Materialize)		86961	90504	47383.22	729.639	46948.41	698.588	Parent Relatio...
↳ Sort		86961	86961	47165.82	717.359	46948.41	698.585	Parent Relatio...
↳ Hash Join		86961	86961	33272.6	604.135	18053.02	118.145	Parent Relatio...
↳ Hash Join		86961	86961	29107.26	513.261	15083.39	78.381	Parent Relatio...
↳ Merge Join		86961	86961	25744.98	439.949	11949.39	36.728	Parent Relatio...
↳ Index Scan	table: client; index: client_pkey;	500000	318291	69260.41	297.107	0.42	0.01	Parent Relatio...
↳ Temporary ( )		86961	86961	12383.73	71.696	11948.93	36.709	Parent Relatio...
↳ Transformation (Hash)		100000	100000	1884.0	41.519	1884.0	41.518	Parent Relatio...
↳ Full Scan (Seq S)	table: employee;	100000	100000	1884.0	19.894	0.0	0.031	Parent Relatio...
↳ Transformation (Hash)		86961	86961	1882.61	39.557	1882.61	39.556	Parent Relatio...
↳ Full Scan (Seq S)	table: collateral;	86961	86961	1882.61	19.122	0.0	0.019	Parent Relatio...

## 6. Времето изминато во извршување на операцијата update по индексирање изнесува

```
[2026-05-11 22:20:22] advdb_202526l_prj_bps_emj.public> UPDATE Collateral
SET
    date_added = '2026-05-05'
WHERE collateral_id = 86962

[2026-05-11 22:20:22] 1 row affected in 58 ms
```

## View 5: Трансфери меѓу сметки, трансакции

- Филтри:** Примарен филтер за погледот v\_client\_transfers ќе биде според sender\_client\_id, а дополнително и според transaction\_date.
- Употреба:** Примарен случај на употреба е кога клиентот сака да ги прегледа своите извршени трансфери. За овој поглед ни се важни перформансите.
- Иницијално време:** Иницијалното време за извршување на погледот е 68ms. Времето е прифатливо, бидејќи JOIN операциите се базираат на примарни и надворешни клучеви кои веќе имаат индекси.

```
SELECT *
FROM v_client_transfers WHERE sender_client_id=70004 ORDER BY transaction_date DESC;
32 Planning Time: 1.950 ms
33 Execution Time: 68.103 ms
```

- Најбавните операции не претставуваат значителен проблем. Двојниот JOIN на Account и Client (за испраќач и примач) се изведува ефикасно преку постојните индекси.

Operation	Params	Rows	Actual Rows	Total Cost	Actual Total Ti...	Startup Cost	Actual Startup Ti...	Raw Desc
↩ Select		18	20	17559.26	74.829	17557.13	66.289	Planning Tim...
Unknown (Gather Merge)		18	20	17559.26	74.829	17557.13	66.289	Parallel Awar...
Sort		6	5	16557.1	55.622	16557.09	55.615	Parent Relati...
Nested Loops (Nested Loop)		6	5	16557.01	55.479	2.0	39.626	Parent Relati...
Nested Loops (Nested Loop)		6	5	16556.03	55.462	1.85	39.618	Parent Relati...
Nested Loops (Nested Loop)		6	5	16555.06	55.442	1.71	39.608	Parent Relati...
Nested Loops (Nested Lo		6	5	16552.05	55.423	1.28	39.606	Parent Relati...
Nested Loops (Nested		6	5	16549.36	55.387	0.86	39.6	Parent Relati...
Nested Loops (Nest		1	0	16504.71	55.333	0.42	39.587	Parent Relati...
Full Scan (Seq S	table: account;	1	0	16496.26	55.316	0.0	39.572	Parent Relati...
Index Scan	table: client; index: client_pkey;	1	1	8.44	0.03	0.42	0.029	Parent Relati...
Index Scan	table: transaction; index: idx_transaction_accou...	10	10	44.55	0.101	0.43	0.027	Parent Relati...
Index Scan	table: account; index: account_pkey;	1	1	0.45	0.005	0.42	0.005	Parent Relati...
Index Scan	table: client; index: client_pkey;	1	1	0.5	0.003	0.42	0.003	Parent Relati...
Index Scan	table: transaction_type; index: transaction_type...	1	1	0.16	0.003	0.14	0.003	Parent Relati...
Index Scan	table: payment_method; index: payment_metho...	1	1	0.16	0.002	0.14	0.002	Parent Relati...

Времето на операциите insert и update изнесува

```

INSERT INTO Transaction (
  amount,
  transaction_date,
  status,
  description,
  reference_number,
  account_id,
  account_id2,
  transaction_type_id,
  payment_method_id,
  currency_id,
  currency_id2,
  emp_counter_id
)
VALUES (
  amount 2500.00,
  transaction_date '2026-04-26 15:12:00.000000',
  status 'PENDING',
  description 'Transferring funds between accounts',
  reference_number 'TRX-0010000002',
  account_id 1001,
  account_id2 1002,
  transaction_type_id 3,
  payment_method_id 4,
  currency_id 1,
  currency_id2 1,
  emp_counter_id 61735
);

```

[2026-05-09 19:47:28] 1 row affected in 835 ms

```

[2026-05-09 19:51:21] advdb_2025261_prj_bps_emj.public> UPDATE Transaction
SET
  status = 'COMPLETED'
WHERE reference_number = 'TRX-0010000002'

```

[2026-05-09 19:51:23] 1 row affected in 1 s 183 ms

- Нема потреба да се преуредува прашалникот.
- Времето на извршување на операциите insert и update останува исто.

### View 6: Сомнителни трансакции

- Филтри:** Примарен филтер за погледот v\_suspicious\_transactions е amount > 9000 AND status IN ('PENDING', 'FAILED').

- Употреба:** Примарен случај на употреба е мониторинг на потенцијално сомнителна активност (AML). За овој поглед ни се важни перформансите, бидејќи без него се губи многу време при извршување.
- Иницијално време:** Иницијалното време за извршување на погледот е 8s 15ms. Времето за извршување е побавно поради full scan на табелата Transaction за да се пронајдат записите со висок износ. Ова не е прифатливо за апликацијата, па затоа пристапуваме кон индексирање.

```
SELECT *
FROM v_suspicious_transactions;
```

```
19 Timing: Generation 7.081 ms (Deform 4.309 ms), Inlinin
20 Execution Time: 8015.286 ms
```

- Најбавните операции се full scan на Transaction за филтрирање по amount и status, и JOIN со Account без индекс на client\_id.

Operation	Params	Rows	Actual Rows	Total Cost	Actual Total Time	Startup Cost	Actual Startup Time	Raw Desc
↳ Select								Planning Time = 0...
↳ Unknown (Gather)		54519	55416	260313.26	1211.29	228690.81	611.026	Parallel Aware = f...
↳ Nested Loops (Nested Lo		17587	13854	253861.36	920.61	227690.81	576.945	Parent Relationshi...
↳ Hash Join		17587	13854	245036.9	739.264	227690.39	576.843	Parent Relationshi...
↳ Full Scan (Seq Scai table: account;		322581	250000	15689.81	88.529	0.0	0.073	Parent Relationshi...
↳ Transformation (Hash)		13630	13854	227520.01	576.072	227520.01	576.071	Parent Relationshi...
↳ Full Scan (Seq S table: transaction;		13630	13854	227520.01	562.887	0.0	17.381	Parent Relationshi...
↳ Index Scan	table: client; index: client_pkey;	1	1	0.5	0.012	0.42	0.012	Parent Relationshi...

Времето изминато во извршување на insert и update изнесува

```
[2026-05-09 21:24:28] advdb_202526l_prj_bps_emj.public> INSERT INTO Transaction (
    amount,
    transaction_date,
    status,
    description,
    reference_number,
    account_id,
    transaction_type_id,
    payment_method_id,
    currency_id
)
VALUES (
    15000.00,
    CURRENT_TIMESTAMP,
    'PENDING',
    'Transferring funds between accounts',
    'TRX-0010000003',
    1001,
    3,
    4,
    1
)
[2026-05-09 21:24:28] 1 row affected in 63 ms
```

```
[2026-05-09 21:25:49] advdb_202526l_prj_bps_emj.public> UPDATE Transaction
SET status = 'FAILED'
WHERE reference_number = 'TRX-0010000003'
[2026-05-09 21:25:49] 1 row affected in 72 ms
```

- Времето изминато во извршување на query-то со индекси изнесува 669ms, и тоа е прифатливо време.

```
CREATE INDEX idx_transaction_amount_status ON Transaction(amount, status);
CREATE INDEX idx_account_client ON Account(client_id);
```

```

SELECT *
FROM v_suspicious_transactions;

```

20	Options: Inlining false, Optimization false, Expressions true, Deforming true
21	Timing: Generation 6.981 ms (Deform 4.226 ms), InLining 0.000 ms, Optimization 3.611 ms, Emission
22	Execution Time: 669.462 ms

Operation	Params	Rows	Actual Rows	Total Cost	Actual Total Time	Startup Cost	Actual Startup Time	Raw Desc
↳ Select								Planning Time = ...
↳ Unknown (Gather)		54519	55416	155084.77	1746.925	123462.32	351.668	Parallel Aware = ...
↳ Nested Loops (Nested Loo		17587	13854	148632.87	1670.541	122462.32	318.49	Parent Relations...
↳ Hash Join		17587	13854	139808.42	1568.883	122461.9	318.431	Parent Relations...
↳ Full Scan (Seq Scan) table: account;		322581	250000	15689.81	1176.29	0.0	0.041	Parent Relations...
↳ Transformation (Hash)		13630	13854	122291.52	317.899	122291.52	317.897	Parent Relations...
↳ Bitmap Index Sca table: transaction;		13630	13854	122291.52	302.997	3018.69	33.033	Parent Relations...
↳ Bitmap Index Index: idx_transaction_amount_status;		54519	55416	3005.06	36.935	0.0	36.934	Parent Relations...
↳ Index Scan table: client; index: client_pkey;		1	1	0.5	0.007	0.42	0.007	Parent Relations...

## 6. Времето изминато во извршување на операцијата update по индексирање изнесува

```

[2026-05-12 00:18:06] advdb_202526l_prj_bps_emj.public> UPDATE Transaction
SET amount = 22020.00
WHERE reference_number = 'TRX-0010000004'

[2026-05-12 00:18:06] 1 row affected in 59 ms

```

## View 7: Вкупна состојба на клиент

- Филтри:** Примарен филтер за погледот v\_account\_balance ќе биде според client\_id.
- Употреба:** Примарен случај на употреба е кога клиентот сака да го провери балансот на своите сметки. Овде се битни перформансите, затоа што сакаме апликацијата да функционира без застој.
- Иницијално време:** Иницијалното време за извршување на овој поглед изнесува 88ms. Времето е мало и прифатливо, бидејќи JOIN-овите се на примарни клучеви.

```

SELECT *
FROM v_account_balance WHERE client_id=445004;

```

14	Planning Time: 0.380 ms
15	Execution Time: 88.018 ms

- Нема потреба од правење план на извршување, бидејќи времето е задоволително. Времето на операции insert и update изнесува

```
[2026-05-09 20:16:24] advdb_202526l_prj_bps_emj.public> INSERT INTO Account (
    account_number,
    account_type,
    balance,
    currency_id,
    open_date,
    status,
    client_id,
    branch_id
)
VALUES (
    'MK000001000001',
    'CHECKING',
    5000.00,
    1,
    CURRENT_DATE,
    'ACTIVE',
    445004,
    18
)

[2026-05-09 20:16:24] 1 row affected in 397 ms
```

```
[2026-05-09 20:16:24] 1 row affected in 397 ms
[2026-05-09 20:27:31] advdb_202526l_prj_bps_emj.public> UPDATE Account
    SET status = 'CLOSED'
    WHERE account_number = 'MK000001000001'

[2026-05-09 20:27:31] 1 row affected in 183 ms
```

- 5. Нема потреба да се преуреди прашалникот.
- 6. Време на извршување на операциите insert и update останува исто.

### View 8: Следење на доцнења (Loan Default Monitoring)

- 1. **Филтри:** Примарен филтер за погледот v\_loan\_default\_monitoring ќе биде според installment\_status IN ('PENDING','LATE') и due\_date, со услов paid\_date IS NULL.
- 2. **Употреба:** Примарен случај на употреба е кога банкарскиот персонал сака да ги идентификува клиентите кои задоцниле со плаќање. За овој поглед ни се важни перформансите, бидејќи без него се губи многу време при извршување.
- 3. **Иницијално време:** Иницијалното време за извршување на погледот е 9s 289ms. Иницијалното време за извршување на погледот беше побавно поради full scan на табелата Loan\_installment. Ова не е прифатливо време за апликацијата па затоа пристапуваме кон индексирање.

<pre>SELECT * FROM v_loan_default_monitoring WHERE installment_status IN ('PENDING', 'LATE') AND paid_date IS NULL AND due_date BETWEEN '2026-05-01' /</pre>	<pre>16 JIT: 17 Functions: 49 18 Options: Inlining false, Optimization false, Expressions true, Deforming true 19 Timing: Generation 5.377 ms (Deform 3.226 ms), Inlining 0.000 ms, Optimization 20 Execution Time: 9289.482 ms</pre>
--	---

- 4. Најбавните операции се full scan на Loan\_installment без филтрирање на платените рати, и JOIN со Account без covering индекс.

Operation	Params	Rows	Actual Rows	Total Cost	Actual Total Time	Startup Cost	Actual Startup Time	Raw Desc
Select								Planning Time...
Hash Join		200033	28261	144790.87	1439.682	118510.16	425.022	Parallel Aware
Full Scan (Seq Scan)	table: account;	1000002	1000002	22464.02	840.099	0.0	0.013	Parent Relatio...
Transformation (Hash)		17395	27138	118292.72	425.064	118292.72	424.909	Parent Relatio...
Unknown (Gather)		17395	27138	118292.72	416.967	1000.29	17.763	Parent Relatio...
Nested Loops (N)		4349	5428	115553.22	387.09	0.29	12.006	Parent Relatio...
Full Scan (Seq)	table: loan_installment;	4349	5428	112994.96	364.194	0.0	11.918	Parent Relatio...
Index Scan	table: loan; index: loan_pkey; 1	1	1	0.59	0.003	0.29	0.003	Parent Relatio...

Времето изминато во извршување на INSERT и UPDATE пред изнесува

```
[2026-05-10 13:35:15] advdb_202526l_prj_bps_emj.public> INSERT INTO Loan_installment (
    installment_number,
    due_date,
    amount,
    status,
    paid_date,
    loan_id
)
VALUES (
    1,
    '2026-06-01',
    114898.29,
    'LATE',
    NULL,
    86961
)

[2026-05-10 13:35:33] 1 row affected in 17 s 811 ms
```

```
[2026-05-10 13:42:01] advdb_202526l_prj_bps_emj.public> UPDATE Loan_installment
SET
    status = 'PAID',
    paid_date = CURRENT_DATE
WHERE installment_id = 9147194

[2026-05-10 13:42:08] 1 row affected in 7 s 68 ms
```

5. Времето изминато во извршување на query-то со индекси изнесува 597.9ms, и тоа е прифатливо време.

```
CREATE INDEX idx_installment_monitoring
ON loan_installment (due_date, loan_id)
WHERE paid_date IS NULL
AND status IN ('PENDING', 'LATE');

CREATE INDEX idx_account_loan_cover
ON account (loan_id)
INCLUDE (account_id, account_number, account_type, balance);
```

SELECT *	17	-> Seq
FROM v_loan_default_monitoring	18	Planning Time: 30.624 ms
WHERE installment_status IN ('PENDING', 'LATE')	19	Execution Time: 597.910 ms
AND paid_date IS NULL AND due_date BETWEEN '2026-05-01' AND CURRENT_DATE;		

Merge Join	200033	28261	46247.68	466.604	44417.03	446.39	Parallel Aware = ...	
Index Scan (Index Or	table: account; index: idx_account_loan_cover;	1000002	34836	47812.46	8.416	0.42	0.016	Parent Relations...
Sort		17395	28261	44460.09	448.049	44416.6	446.364	Parent Relations...
Hash Join		17395	27138	43191.44	427.248	3297.36	54.604	Parent Relations...
Bitmap Index	table: loan_installment;	17395	27138	40219.16	359.947	370.73	12.438	Parent Relations...
Bitmap Indx	index: idx_installment_monitoring;	17395	27138	366.38	7.376	0.0	7.375	Parent Relations...
Transformation (f)		86961	86961	1839.61	42.075	1839.61	42.074	Parent Relations...
Full Scan (f)	table: loan;	86961	86961	1839.61	21.934	0.0	0.028	Parent Relations...

6. Времето изминато во извршување на операциите insert и update по индексирање изнесува

```
[2026-05-10 22:59:27] advdb_202526l_prj_bps_emj.public> INSERT INTO Loan_installment (
    installment_number,
    due_date,
    amount,
    status,
    paid_date,
    loan_id
)
VALUES (
    2,
    '2026-06-10',
    45679.29,
    'LATE',
    NULL,
    86961
)

[2026-05-10 22:59:38] 1 row affected in 10 s 678 ms
```

```
[2026-05-12 17:13:21] advdb_202526l_prj_bps_emj.public> UPDATE Loan_installment
SET
    status = 'PAID',
    paid_date = CURRENT_DATE
WHERE installment_id = 9147196

[2026-05-12 17:13:24] 1 row affected in 2 s 751 ms
```

## View 9: Accounts (сметки)

1. **Филтри:** Примарен филтер за погледот v\_accounts ќе биде според client\_id.
2. **Употреба:** Примарен случај на употреба е преглед на сите сметки на клиент при услуги на шалтер. Овде се битни перформансите.
3. **Иницијално време:** Иницијалното време за извршување на овој поглед е 100ms, што е прифатливо, па затоа нема да имаме потреба од индексирање.

```
SELECT *
FROM v_accounts WHERE client_id=40000;
```

17	-> seq scan on current
18	Planning Time: 0.738 ms
19	Execution Time: 100.166 ms

4. Нема потреба од правење план на извршување, бидејќи времето е задоволително. Времето на операциите insert и update е

```
[2026-05-09 20:39:10] advdb_2025261_prj_bps_emj.public> INSERT INTO Account (
    account_number,
    account_type,
    balance,
    currency_id,
    open_date,
    status,
    client_id,
    branch_id
)
VALUES (
    'MK000001000002',
    'SAVINGS',
    10000.00,
    1,
    CURRENT_DATE,
    'ACTIVE',
    445046,
    18
)

[2026-05-09 20:39:16] 1 row affected in 5 s 844 ms
```

```
[2026-05-09 20:40:48] advdb_2025261_prj_bps_emj.public> UPDATE Account
    SET status = 'CLOSED'
    WHERE account_number = 'MK000001000002'

[2026-05-09 20:40:48] 1 row affected in 309 ms
```

5. Нема потреба да се преуредува прашалникот.
6. Времето на извршување на операциите insert и update останува исто.

### View 10: Камати по штедни сметки

1. **Филтри:** Примарен филтер за погледот v\_savings\_interest\_payments ќе биде според client\_id.
2. **Употреба:** Примарен случај на употреба е кога клиентот сака да ги прегледа добиените камати по период. Овде се важни перформансите.
3. **Иницијално време:** Иницијалното време за извршување на овој поглед е 214ms, што е прифатливо, па затоа нема да имаме потреба од индексирање.

SELECT *	14	Index Cond: (a
FROM v_savings_interest_payments WHERE client_id=5;	15	Planning Time: 0.639 ms
	16	Execution Time: 214.638 ms

4. Нема потреба од правење план на извршување, бидејќи времето е задоволително. Времето на операциите insert и update е

```
[2026-05-09 20:52:33] advdb_202526l_prj_bps_emj.public> INSERT INTO InterestPayment (
    amount,
    payment_date,
    period_start,
    period_end,
    account_id,
    transaction_id
)
VALUES (
    85.50,
    CURRENT_DATE,
    '2026-04-01',
    '2026-04-30',
    1001,
    50001
)

[2026-05-09 20:52:33] 1 row affected in 267 ms
```

```
[2026-05-09 20:56:14] advdb_202526l_prj_bps_emj.public> UPDATE InterestPayment
    SET payment_date = '2026-05-15'
    WHERE interest_payment_id = 2001021

[2026-05-09 20:56:14] 1 row affected in 72 ms
```

5. Нема потреба да се преуредува прашалникот.
6. Времето на извршување на операциите insert и update останува исто.

### View 11: Дневен извештај на филијала

1. **Филтри:** Примарен филтер за погледот v\_daily\_branch\_report ќе биде според branch\_id, а дополнително и според report\_date.
2. **Употреба:** Примарен случај на употреба е кога раководството на филијалата или централниот менаџмент сака да ги следи дневните активности. Овде се битни перформансите.
3. **Иницијално време:** Иницијалното време за извршување на погледот е 385ms. Ова е прифатливо време за апликацијата.

SELECT *	7	Planning Time: 1.425 ms
FROM v_daily_branch_report;	8	Execution Time: 385.409 ms

4. Нема потреба од правење план на извршување, бидејќи времето е задоволително. Времето на операциите insert и update е

```
[2026-05-09 21:01:31] advdb_202526l_prj_bps_emj.public> INSERT INTO Daily_report (
    report_date,
    total_transactions,
    total_amount,
    branch_id
)
VALUES (
    CURRENT_DATE,
    120,
    458900.50,
    50
)

[2026-05-09 21:01:31] 1 row affected in 169 ms
```

```
[2026-05-09 21:03:29] advdb_202526l_prj_bps_emj.public> UPDATE Daily_report
SET
    total_transactions = 125,
    total_amount = 470000.00
WHERE report_id = 100001

[2026-05-09 21:03:29] 1 row affected in 57 ms
```

5. Нема потреба да се преуредува прашалникот.
6. Времето на извршување на операциите insert и update останува исто.

Членови на тим:  
Михаела Ковчегарска 231068  
Јована Мечева 231124  
Емилија Костова 231107