

Напредни бази на податоци

Фаза 3 – Оптимизација на прашалници и погледи

Име на проект: BlinkBuy

View 1: Активен каталог на производи (v_active_product_catalog)

1. Филтри и употреба

- **Примарен филтер:** Филтрирањето се врши според `variant_id` (за приказ на поединечна понуда) и `category_name` (за листинг во продавницата).
- **Употреба:** Овој поглед е срцето на апликацијата. Се користи секој пат кога корисникот пребарува активни производи. Потребна е максимална брзина бидејќи директно влијае на корисничкото искуство.

2. Тест прашалник

```
[2026-05-17 23:39:28] Connected
[2026-05-17 23:39:28] advdb_202526l_prj_blin..., public> SELECT *
                                FROM v_active_product_catalog
                                WHERE variant_id = 234
[2026-05-17 23:39:29] 1 row retrieved starting from 1 in 1 s 194 ms (execution: 445 ms, fetching: 749 ms)
```

3. Состојба

- **Време на извршување:** 445 ms
- **Време на INSERT:** 12 ms
- **Време на UPDATE:** 13 ms

```
[2026-05-18 00:52:15] advdb_202526l_prj_blin..., public> EXPLAIN (ANALYZE, BUFFERS)
                                INSERT INTO inventory_items (variant_id, warehouse_id, quantity)
                                VALUES (
                                    12, 50, 52
                                )
[2026-05-18 00:52:16] 8 rows retrieved starting from 1 in 387 ms (execution: 12 ms, fetching: 375 ms)
```

```
[2026-05-18 00:52:50] advdb_202526l_prj_blin..., public> EXPLAIN ANALYZE
                                UPDATE inventory_items
                                SET quantity = 19
                                WHERE variant_id = 564
[2026-05-18 00:52:51] 5 rows retrieved starting from 1 in 393 ms (execution: 13 ms, fetching: 380 ms)
```

4. Заклучок

Времето на извршување на SELECT прашалникот е прифатливо. Постоечките индекси на примарните и надворешните клучеви обезбедуваат доволна брзина за редовното користење на погледот. Операциите INSERT и UPDATE се извршуваат со прифатлива брзина.

View 2: Централен административен преглед (v_order_command_center_full)

1. Филтри и употреба

- **Примарен филтер:** Примарниот филтер за погледот е според `order_id`, а дополнително може да се користи `order_status` за масовен преглед на нарачки во одредена фаза.
- **Употреба:** Погледот е наменет за администраторите и тимот за логистичката поддршка. Овозможува брз пристап до сите информации за една нарачка на едно место: податоци за купувачот, адресата, курирот и текстуална листа на сите нарачани ставки. Брзината е клучна за ефикасно работење на Help Desk центарот.

2. Тест прашалник

```
[2026-05-19 02:39:50] advdb_202526l_prj_blin..., public> SELECT *
                                FROM v_order_command_center_full
                                WHERE order_id = 1789
[2026-05-19 02:39:52] 1 row retrieved starting from 1 in 1 s 603 ms (execution: 1 s 278 ms, fetching: 325 ms)
```

3. Состојба ПРЕД индексирање

- **Иницијално време на извршување:** 1 s 278 ms
- **Најбавни операции (Bottlenecks):**
- Parallel Seq Scan на табелата `shipments`: Базата мора целосно да ја скенира табелата со пратки за да ја пронајде онаа што е поврзана со бараната нарачка.
- Subplan Scan на табелата `order_items`: Бидејќи погледот користи подпрашалник со `string_agg` за да ги излиста артиклите, недостатокот на индекс на `order_id` во ставките предизвикува бавно групирање на податоците.

Operation	Params	Rows	Total Cost	Startup Cost
↳ Select				
↳ Nested Loops (Nested Loop)		1	218549.9	2.55
↳ Nested Loops (Nested Loop)		1	27.44	2.13
↳ Nested Loops (Nested Loop)		1	18.99	1.7
↳ Nested Loops (Nested Loop)		1	18.49	1.28
↳ Nested Loops (Nested Loop)		1	17.43	1.28
↳ Nested Loops (Nested Loop)		1	16.89	0.85
↳ Index Scan	table: orders; index: orders_pkey;	1	8.45	0.43
↳ Index Scan	table: users; index: users_pkey;	1	8.44	0.42
↳ Index Scan	table: user_profiles; index: user_prof...	1	0.52	0.42
↳ Full Scan (Seq Scan)	table: shipping_methods;	3	1.03	0
↳ Index Scan (Index Only Scan)	table: addresses; index: idx_v2_addr...	1	0.48	0.42
↳ Index Scan	table: shipments; index: idx_v13_shi...	1	8.44	0.42
↳ Index Scan	table: user_profiles; index: user_prof...	1	8.44	0.42
↳ Aggregate		1	218514.01	218514
↳ Nested Loops (Nested Loop)		6	218513.9	0.85
↳ Nested Loops (Nested Loop)		6	218511.08	0.43
↳ Full Scan (Seq Scan)	table: order_items;	1	218460.41	0
↳ Index Scan	table: product_variants; index: produ...	1	8.45	0.43
↳ Index Scan	table: products; index: products_pkey;	1	0.47	0.42

- **Време на INSERT (Пред):** 14 ms

```
[2026-05-19 10:51:36] advdb_202526l_prj_blin..., public> EXPLAIN (ANALYZE, BUFFERS)
INSERT INTO orders (user_id, shipping_method_id, status, total_amount,
VALUES (
    1564,
    2,
    'PAID',
    2500.00,
    NOW()
)
[2026-05-19 10:51:36] 8 rows retrieved starting from 1 in 377 ms (execution: 14 ms, fetching: 363 ms)
```

- **Време на UPDATE (Пред): 14 ms**

```
[2026-05-19 10:51:07] advdb_202526l_prj_blin..., public> EXPLAIN (ANALYZE, BUFFERS)
UPDATE orders
SET
    status = 'PAID',
    total_amount = 1500.00,
    order_date = NOW()
WHERE order_id = 115
[2026-05-19 10:51:07] 7 rows retrieved starting from 1 in 365 ms (execution: 14 ms, fetching: 351 ms)
```

4. Воведување на индекси

Иницијалното време на извршување од 1 s 278 ms е неприфатливо за Help Desk центар кој треба да одговора во реално време. Parallel Seq Scan на табелата shipments и Subplan Scan на order_items се главните причини за бавноста.

Воведувањето на индекси директно ги таргетира овие операции и ги заменува со Index Scan и Index Only Scan, со цел времето да се намали на под 100 ms.

За да се елиминираат сериските скенирања при поврзување на логистичките податоци и ставките со главната нарачка, воведени се следните индекси:

```
CREATE INDEX idx_v2_shipments_order_id ON shipments(order_id);
CREATE INDEX idx_v2_order_items_order_id ON order_items(order_id); 5 s
```

5. Состојба ПОСЛЕ индексирање

- **Ново време на извршување: 22 ms**

Најбавни операции сега: Операциите за целосно скенирање се заменети со **Index Scan** и **Index Only Scan**. Базата сега веднаш ги лоцира поврзаните записи во shipments и order_items без да ги пребарува преостанатите стотици илјади редови. Времето на извршување падна значително.

```
[2026-05-18 01:53:22] advdb_202526l_prj_blin..., public> SELECT *
FROM v_order_command_center_full
WHERE order_id = 1789
[2026-05-18 01:53:23] 1 row retrieved starting from 1 in 385 ms (execution: 22 ms, fetching: 363 ms)
```

Operation	Params	Rows	Actual Ro...	Total Cost	Actual Total ...	Startup Cost	Actual Startu...	Property	Value
↳ Select								Operation	select
↳ Nested Loops (Nested Loop)		1	1	108.43	0.695	2.7	0.687	Planning Time	4.109
↳ Nested Loops (Nested Loop)		1	1	30.65	0.383	2.27	0.377	Triggers	[]
↳ Nested Loops (Nested Loop)		1	1	26.2	0.289	1.85	0.285	Execution TL...	0.948
↳ Nested Loops (Nested Loop)		1	1	25.7	0.251	1.42	0.248		
↳ Nested Loops (Nested Loop)		1	1	17.43	0.209	1.28	0.206		
↳ Nested Loops (Nested Loop)		1	1	16.89	0.144	0.85	0.142		
↳ Index Scan	table: orders; index: orders_pkey;	1	1	8.45	0.097	0.43	0.096		
↳ Index Scan	table: users; index: users_pkey;	1	1	8.44	0.041	0.42	0.041		
↳ Index Scan	table: user_profiles; index: user_profiles_u...	1	1	0.52	0.061	0.42	0.061		
↳ Index Scan	table: shipping_methods; index: shipping_...	1	1	8.16	0.039	0.14	0.039		
↳ Index Scan (Index Only Scan)	table: addresses; index: idx_v2_addresses...	1	1	0.48	0.035	0.42	0.035		
↳ Index Scan (Index Only Scan)	table: shipments; index: idx_v2_shipments...	1	1	4.44	0.092	0.42	0.091		
↳ Index Scan	table: user_profiles; index: user_profiles_u...	1	1	8.44	0.064	0.42	0.064		
↳ Aggregate		1	1	69.33	0.237	69.32	0.236		
↳ Nested Loops (Nested Loop)		6	1	69.22	0.227	1.41	0.225		
↳ Nested Loops (Nested Loop)		6	1	66.4	0.17	0.99	0.168		
↳ Index Scan (Index Only Scan)	table: order_items; index: idx_v2_order_ite...	6	1	15.73	0.109	0.56	0.108		
↳ Index Scan	table: product_variants; index: product_var...	1	1	8.45	0.058	0.43	0.058		
↳ Index Scan	table: products; index: products_pkey;	1	1	0.47	0.055	0.42	0.055		

6. Влијание врз запишувањето (Write-Impact Analysis)

За ова мерење се користеше симулација на креирање на нова нарачка во статус 'PAID' и ажурирање на постоечка нарачка.

- **Ново време на INSERT: 16 ms**

```
[2026-05-19 10:52:47] advdb_202526l_prj_blin..., public> EXPLAIN (ANALYZE, BUFFERS)
                INSERT INTO orders (user_id, shipping_method_id, status, total_amount,
                VALUES (
                    1564,
                    2,
                    'PAID',
                    2500.00,
                    NOW()
                )
[2026-05-19 10:52:48] 8 rows retrieved starting from 1 in 600 ms (execution: 30 ms, fetching: 570 ms)
```

- **Ново време на UPDATE: 32 ms**

```
[2026-05-19 10:53:44] advdb_202526l_prj_blin..., public> EXPLAIN (ANALYZE, BUFFERS)
                UPDATE orders
                SET
                    status = 'PAID',
                    total_amount = 1500.00,
                    order_date = NOW()
                WHERE order_id = 115
[2026-05-19 10:53:44] 7 rows retrieved starting from 1 in 395 ms (execution: 26 ms, fetching: 369 ms)
```

7. Заклучок

По воведувањето на индексите забележано е очекувано зголемување на времето потребно за операциите на запишување (INSERT и UPDATE). Сепак, ова зголемување од десетина милисекунди е прифатлив компромис, бидејќи брзината на прашалникот се подобри значително (од 1.278 s на 22 ms), со што Help Desk тимот добива инстантен пристап до информациите за нарачките.

View 3: Исполнување на ставки по продавач (v_seller_item_fulfillment)

1. Филтри и употреба

- **Примарен филтер:** Примарен филтер за погледот е seller_id (пребарување на сите ставки кои еден продавач треба да ги спакува), а дополнително може да се пребарува и по order_id.
- **Употреба:** Овој поглед е клучен за контролниот панел на продавачите (Seller Dashboard). Продавачот го користи за да види кои производи од неговите магацини се нарачани, каде треба да се подигнат и кој курир ќе ги преземе. Брзината по seller_id е критична за дневните операции на фирмите-продавачи.

2. Тест прашалник

```
[2026-05-18 02:47:39] advdb_202526l_prj_blin...myschema> SELECT *
                FROM v_seller_item_fulfillment
                WHERE seller_id = 333
[2026-05-18 02:47:42] 0 rows retrieved in 2 s 704 ms (execution: 2 s 344 ms, fetching: 360 ms)
```

3. Состојба ПРЕД индексирање

- **Иницијално време на извршување:** 2 s 704 ms
- **Најбавни операции (BottleNecks):**
- Seq Scan на табелата warehouses: Базата мора целосно да ја скенира табелата со магацини за да ги најде оние кои му припаѓаат на user_id (продавачот), бидејќи нема индекс на тој надворешен клуч.

- Hash Join и Nested Loops: Поради големиот број на спојувања (Products, Variants, Manufacturers, Warehouses, Instances), отсуството на индекси на клучните точки за поврзување предизвикува бавно вкрстување на податоците.
- Seq Scan на manufacturers: Иако табелата е помала, нејзиното постојано скенирање во циклус додава на вкупното време.

Operation	Params	R...	Act...	Tot...	Actu...	Star...	Actu...
▼ Nested Loops (Nested Loop)		2	0	5797...	695.338	3070...	695.3...
▼ Nested Loops (Nested Loop)		2	0	5797...	695.336	3070...	695.3...
▼ Hash Join		2	0	5797...	695.334	3070...	695.3...
▼ Nested Loops (Nested Loop)		219...	0	5746...	0	30251...	0
▼ Hash Join		219...	0	3707...	0	30251...	0
▼ Hash Join		219...	0	3701...	0	3024...	0
▼ Hash Join		219...	0	3559...	0	28886...	0
Full Scan (Seq Scan)	table: product...	375...	0	56908	0	0	0
▼ Transformation (Hash)		219...	0	2837...	0	28377...	0
▼ Hash Join		219...	0	2837...	0	97157...	0
▼ Hash Join		219...	0	2515...	0	71662...	0
▼ Hash Join		410...	0	1990...	0	28201...	0
Full Scan (Seq Scan)	table: order_it...	250...	0	118458	0	0	0
▼ Transformation (Hash)		529...	0	1898...	0	18989...	0
Full Scan (Seq Scan)	table: shipmen...	529...	0	1898...	0	0	0
▼ Transformation (Hash)		267...	0	3855...	0	38557...	0
Full Scan (Seq Scan)	table: orders;	267...	0	3855...	0	0	0
▼ Transformation (Hash)		322...	0	1957...	0	19571....	0
Full Scan (Seq Scan)	table: user_pr...	322...	0	1957...	0	0	0
▼ Transformation (Hash)		161...	0	11612.9	0	11612.9	0
Full Scan (Seq Scan)	table: products;	161...	0	11612.9	0	0	0
▼ Transformation (Hash)		600	0	12	0	12	0
Full Scan (Seq Scan)	table: manufac...	600	0	12	0	0	0
Index Scan	table: product...	1	0	0.93	0	0.43	0
▼ Transformation (Hash)		1	0	4531....	695.032	4531.27	695.0...
Full Scan (Seq Scan)	table: wareho...	1	0	4531....	680.594	0	680.5...

- **Време на INSERT (Пред): 91 ms**

```
[2026-05-19 11:35:11] advdb_202526l_prj_blin..., public> EXPLAIN (ANALYZE, BUFFERS)
INSERT INTO order_items (order_id, variant_id, price_history
SELECT
    78085,
    228416,
    863832,
    1,
    36923.88
FROM product_price_history
WHERE variant_id = 228416
LIMIT 1

[2026-05-19 11:35:11] 15 rows retrieved starting from 1 in 418 ms (execution: 91 ms, fetching: 327 ms)
```

- **Време на UPDATE (Пред): 15 ms**

```
[2026-05-19 11:35:53] advdb_202526l_prj_blin..., public> EXPLAIN (ANALYZE, BUFFERS)
UPDATE order_items
SET quantity = 2,
    unit_price = 36923.88
WHERE order_id = 78085 AND variant_id = 228416

[2026-05-19 11:35:53] 9 rows retrieved starting from 1 in 385 ms (execution: 15 ms, fetching: 370 ms)
```

4. Воведување на индекси

```
CREATE INDEX idx_v3_warehouses_user_id ON warehouses(user_id);
CREATE INDEX idx_v3_product_instances_warehouse_id ON product_instances(warehouse_id);
```

- **Ново време на извршување: 62 ms**

```
[2026-05-19 11:52:46] advdb_202526l_prj_blin..., public> SELECT *
FROM v_seller_item_fulfillment
WHERE seller_id = 333
[2026-05-19 11:52:47] 0 rows retrieved in 403 ms (execution: 62 ms, fetching: 341 ms)
```

- **Најбавни операции сега:**
Најголемото подобрување е кај примарниот случај на употреба. Со воведување на индексот на warehouses(user_id), времето на извршување падна од **2.7 секунди на само 62 милисекунди**. Операцијата Seq Scan на warehouses е заменета со ефикасен Index Scan.

♀ Index Scan	table: product_instances; inde...	1	0	0.93	0	0.43	0
Transformation (Hash)		2	0	12.3	0.059	12.3	0.058
Transformation (Hash)		2	0	12.3	0.058	4.44	0.057
♀ Bitmap Index Scan (Bitmap Heap Scan)	table: warehouses;	2	0	4.43	0.038	0	0.038
♀ Bitmap Index Scan	index: test_idx_wh_user;	2	0	0.47	0	0.42	0
♀ Index Scan	table: shipments; index: idx_v1...	1	0	0.59	0	0.42	0
♀ Index Scan	table: user_profiles; index: us...	1	0				

5. Влијание врз запишувањето (Write-Impact Analysis)

- **Ново време на INSERT: 122 ms**

```
[2026-05-19 11:54:13] advdb_202526l_prj_blin..., public> EXPLAIN (ANALYZE, BUFFERS)
INSERT INTO order_items (order_id, variant_id, price_history_id, quantity, unit_price)
SELECT
    78085,
    228416,
    863832,
    1,
    36923.88
FROM product_price_history
WHERE variant_id = 228416
LIMIT 1
[2026-05-19 11:54:13] 17 rows retrieved starting from 1 in 440 ms (execution: 122 ms, fetching: 318 ms)
```

- **Ново време на UPDATE: 15 ms**

```
[2026-05-19 11:54:37] advdb_202526l_prj_blin..., public> EXPLAIN (ANALYZE, BUFFERS)
UPDATE order_items
SET quantity = 2,
    unit_price = 36923.88
WHERE order_id = 78085 AND variant_id = 228416
[2026-05-19 11:54:37] 9 rows retrieved starting from 1 in 364 ms (execution: 15 ms, fetching: 349 ms)
```

6. Заклучок

Иако воведувањето на индексот врз warehouses ја оптимизираше влезната точка на пребарувањето, вкупниот трошок останува висок поради големината на табелите order_items (2.5M записи) и shipment_items. За потребите на агрегација (Revenue Report), базата користи Hash Join и Sequential Scan бидејќи обработува голем сет на податоци одеднаш, што е поефикасно од Index Scan во ова сценарио.

View 4: Матрица за избор на варијанти (v_product_selection_matrix)

1. Филтри и употреба

- **Примарен филтер:** product_id (за приказ на сите опции на еден модел) и variant_id (за поединечен приказ на SKU).
- **Употреба:** Овој поглед се повикува на самата страница на производот (Product Detail Page). Корисникот кога ќе кликне на одреден лаптоп, погледот ги повлекува сите достапни бои, RAM конфигурации и мемории за тој модел.

2. Тест прашалници

```
[2026-05-19 12:42:15] advdb_202526l_prj_blin..., public> SELECT *
                                FROM v_product_selection_matrix
                                WHERE product_id = 222
[2026-05-19 12:42:16] 3 rows retrieved starting from 1 in 373 ms (execution: 14 ms, fetching: 359 ms)
[2026-05-19 12:42:31] advdb_202526l_prj_blin..., public> SELECT *
                                FROM v_product_selection_matrix
                                WHERE variant_id = 322
[2026-05-19 12:42:31] 1 row retrieved starting from 1 in 365 ms (execution: 16 ms, fetching: 349 ms)
```

3. Состојба

- **Иницијално време (4А):** 14 ms
- **Иницијално време (4Б):** 16 ms
- **Време на INSERT:** 15 ms
- **Време на UPDATE:** 19 ms

```
[2026-05-19 12:43:56] advdb_202526l_prj_blin..., public> EXPLAIN (ANALYZE, BUFFERS)
                                INSERT INTO product_attribute_values (variant_id, attribute_id, attr_value)
                                VALUES (228416, 999, 'Test')
[2026-05-19 12:43:57] 7 rows retrieved starting from 1 in 405 ms (execution: 15 ms, fetching: 390 ms)
```

```
[2026-05-19 12:45:13] advdb_202526l_prj_blin..., public> EXPLAIN (ANALYZE, BUFFERS)
                                UPDATE product_attribute_values
                                SET attr_value = 'Intel i5'
                                WHERE variant_id = 1 AND attribute_id = 1
[2026-05-19 12:45:13] 7 rows retrieved starting from 1 in 381 ms (execution: 19 ms, fetching: 362 ms)
```

4. Заклучок

По извршените тестови, заклучено е дека нема потреба од додавање дополнителни индекси за овој поглед. Иницијалното време на извршување од 14–16 ms е одлично. Базата веќе ги користи постоечките индекси на примарните и надворешните клучеви кои беа дефинирани во Фаза 2.

View 5: Просечна оцена на производ (v_product_ratings_avg)

1. Филтри и употреба

- **Примарен филтер:** product_id (за приказ на просечна оцена на еден производ)

2. Тест прашалник

```
[2026-05-25 13:25:38] advdb_202526l_prj_blin..., public> select *
                                from v_product_ratings_avg
                                where product_id=3
[2026-05-25 13:25:39] 1 row retrieved starting from 1 in 469 ms (execution: 84 ms, fetching: 385 ms)
```

3. Состојба

- **Иницијално време на извршување:** 84 ms
- **Време на INSERT:** 320 ms

```
[2026-05-25 14:05:15] advdb_202526l_prj_blin..., public> INSERT INTO reviews VALUES
                                                                (300001, 374403, 8, 4, 'fine', now())
[2026-05-25 14:05:15] 1 row affected in 320 ms
```

- **Време на UPDATE:** 91 ms

```
[2026-05-25 14:13:28] advdb_202526l_prj_blin..., public> EXPLAIN (ANALYZE , BUFFERS )
                                UPDATE reviews
                                SET comment_text='fine'
                                where review_id=68
[2026-05-25 14:13:28] 9 rows retrieved starting from 1 in 476 ms (execution: 91 ms, fetching: 385 ms)
```

4. Заклучок

Иницијалното SELECT време на погледот е прифатливо. Воведувањето на дополнителни индекси не е потребно.

View 6: Преглед на месечни продажби(v_monthly_sales_report)

1. Филтри и употреба

- **Примарен филтер:** sales_month (за приказ на продажбите за еден конкретен месец)
- **2. Тест прашалник**

```
[2026-05-25 14:19:04] advdb_202526l_prj_blin..., public> select *
                                from v_monthly_sales_report
                                where sales_month='2026-05'
[2026-05-25 14:19:05] 1 row retrieved starting from 1 in 910 ms (execution: 445 ms, fetching: 465 ms)
```

3. Состојба ПРЕД индексирање

- **Иницијално време на извршување:** 445 ms.
- **Време на INSERT (Пред):** 96 ms

```
[2026-05-25 14:37:34] advdb_202526l_prj_blin..., public> INSERT INTO orders VALUES
                                                                (2000002,44,1,now(),20,'SHIPPED')
[2026-05-25 14:37:34] 1 row affected in 96 ms
```

- **Време на UPDATE (Пред):** 600 ms

```
[2026-05-25 14:39:02] advdb_202526l_prj_blin..., public> EXPLAIN (ANALYZE, BUFFERS)
                                UPDATE orders
                                SET status='PLACED'
                                WHERE order_id=1575099
[2026-05-25 14:39:03] 8 rows retrieved starting from 1 in 955 ms (execution: 600 ms, fetching: 355 ms)
```

4. Заклучок

Иницијалното SELECT време на погледот е прифатливо. Воведувањето на дополнителни индекси не е оправдано.

View 7: Приходи по продавач (v_seller_revenue_report)

1. Филтри и употреба

- **Примарен филтер:** seller_id (за пронаоѓање конкретен продавач)
- **Употреба:** Овој преглед ги сумира приходите и продажбата на секој продавач.

2. Тест прашалник

```
[2026-05-26 21:07:39] advdb_202526l_prj_blin..., public> select *
                        from v_seller_revenue_report
                        where seller_id = 414352
[2026-05-26 21:07:41] 1 row retrieved starting from 1 in 2 s 650 ms (execution: 2 s 319 ms, fetching: 331 ms)
```

2. Состојба

- **Иницијално време на извршување:** 2s 319 ms
- **Најбавни операции (Bottlenecks):**
- Најбавна операција е агрегацијата:

Operation	Par...	Ro...	Total ...	S...
Select				
Sort		1	1835.52	18...
Aggregate		1	1835.5	18...
Sort		4	1835.43	18...
Nested Loops (Nested Loop)		4	1835.38	13.16
Nested Loops (Nested Loop)		4	1833.61	12...
Nested Loops (Nested Loop)		4	1831.73	12.31
Nested Loops (Nested Loop)		13	1825.67	11.88
Nested Loops (Nested Loop)		13	1815.96	11...
Nested Loops (Nested Loop)		106	1768.18	11.02
Nested Loops (Nested Loop)		2	20.76	4.86
Index Scan	table:...	1	8.44	0.42
Bitmap Index Scan (Bitmap Heap Scan)	table:...	2	12.3	4.44
Bitmap Index Scan	index:...	2	4.43	0
Bitmap Index Scan (Bitmap Heap Scan)	table:...	222	871.49	6.16
Bitmap Index Scan	index:...	222	6.1	0
Index Scan	table:...	1	0.45	0.43

- **Време на INSERT:** 4s 493 ms

```
[2026-05-26 21:08:49] advdb_202526l_prj_blin..., public> INSERT INTO orders
                        VALUES (2000003,44,1,now(),20,'SHIPPED')
[2026-05-26 21:08:54] 1 row affected in 4 s 493 ms
```

- **Време на UPDATE:** 1.6s

```
[2026-05-26 21:15:26] advdb_202526l_prj_blin..., public> EXPLAIN (ANALYZE, BUFFERS)
                        UPDATE orders
                        SET status='PLACED'
                        WHERE order_id=1575099
[2026-05-26 21:15:27] 8 rows retrieved starting from 1 in 1 s 183 ms (execution: 854 ms, fetching: 329 ms)
```

4. Заклучок

Иако агрегацијата на приходите бара обработка на голем број поврзани записи, дополнително индексирање не е оправдано. Планот покажува дека базата веќе

оптимално ги користи постоечките индекси за сите JOIN операции. Најголемиот дел од преостанатиот трошок отпаѓа на пресметковните операции (AGGREGATE и SORT), кои се извршуваат во CPU и не можат да се забрзаат со индекси. Со ова е постигнат баланс помеѓу брзината на известувањето и перформансите на DML операциите (INSERT/UPDATE) кои се клучни за непречено купување.

View 8: Следење на активни гаранции (v_active_warranty_tracker)

1. Филтри и употреба

- **Примарен филтер:** serial_number (за пронаоѓање конкретен уред) и warranty_status (за филтрирање на активни гаранции).
- **Употреба:** Се користи од корисничка поддршка, овозможува веднаш да се провери дали одреден сериски број има важечка гаранција, кога е купен производот и кој е сопственикот, со цел да се одобри или одбие сервисна интервенција.

2. Тест прашалник

```
[2026-05-18 03:58:53] advdb_202526l_prj_blin...myschema> SELECT *
FROM v_active_warranty_tracker
WHERE serial_number = 'SN-232668-2-29084'
[2026-05-18 03:58:54] 1 row retrieved starting from 1 in 464 ms (execution: 131 ms, fetching: 333 ms)
```

3. Состојба ПРЕД индексирање

- **Иницијално време на извршување:** 131 ms
- **Најбавни операции (Bottlenecks):**
- **Nested Loops:** Погледот поврзува дури 8 табели. Главниот застој се случува при пребарување на серискиот број во табелата product_instances, каде поради недостаток на индекс, се врши скенирање на голем број страници со податоци за да се најде соодветната инстанца.
- **Време на INSERT (Пред):** 16 ms
- **Време на UPDATE (Пред):** 13 ms

Operation	Params	Rows	Actua...	Total ...	Actual ...	Startup...	Actual S...
↳ Select							
↳ Nested Loops (Nested Loop)		1	1	35.02	0.15	3.98	0.145
↳ Nested Loops (Nested Loop)		1	1	33.52	0.124	3.56	0.12
↳ Nested Loops (Nested Loop)		1	1	25.07	0.114	3.13	0.111
↳ Nested Loops (Nested Loop)		1	1	23.57	0.105	2.71	0.101
↳ Nested Loops (Nested Loop)		1	1	22.03	0.095	2.29	0.092
↳ Nested Loops (Nested Loop)		1	1	20.46	0.086	1.86	0.083
↳ Nested Loops (Nested Loop)		1	1	18.53	0.076	1.42	0.073
↳ Nested Loops (Nested Loop)		1	1	17.03	0.064	1	0.062
↳ Index Scan	table: product_instances; index...	1	1	8.58	0.039	0.56	0.038
↳ Index Scan	table: warranties; index: warra...	1	1	8.45	0.012	0.43	0.012
↳ Index Scan	table: shipment_items; index: u...	1	1	1.5	0.011	0.43	0.011
↳ Index Scan	table: order_items; index: orde...	1	1	1.93	0.009	0.43	0.009
↳ Index Scan	table: orders; index: orders_pk...	1	1	1.57	0.008	0.43	0.008
↳ Index Scan	table: users; index: users_pkey;	1	1	1.54	0.009	0.42	0.009
↳ Index Scan	table: user_profiles; index: use...	1	1	1.5	0.008	0.42	0.008
↳ Index Scan	table: product_variants; index: ...	1	1	8.45	0.009	0.43	0.008
↳ Index Scan	table: products; index: product...	1	1	1.46	0.01	0.42	0.01

```
[2026-05-19 22:22:22] advdb_202526l_prj_blin..., public> EXPLAIN (ANALYZE, BUFFERS)
          INSERT INTO warranties (instance_id, duration_months, term
          VALUES (6280, 24, 'Performance Test Warranty')
[2026-05-19 22:22:22] 7 rows retrieved starting from 1 in 398 ms (execution: 16 ms, fetching: 382 ms)
[2026-05-19 23:20:29] advdb_202526l_prj_blin...myschema> EXPLAIN (ANALYZE, BUFFERS)
          UPDATE product_instances
          SET status = 'DEFECTIVE'
          WHERE serial_number = 'SN-232668-2-29084'
[2026-05-19 23:20:29] 7 rows retrieved starting from 1 in 384 ms (execution: 13 ms, fetching: 371 ms)
```

4. Заклучок

Иницијалното SELECT време на погледот е прифатливо. Воведувањето на дополнителни индекси не е оправдано.

View 9: Ревизија и синхронизација на залиха (v_inventory_drift_sync_report)

1. Филтри

- **Примарен филтер:** Примарен филтер за погледот е според variant_id. Дополнително, погледот се користи за идентификација на сите производи каде постои отстапување (drift) помеѓу вкупната и физичката залиха.

2. Употреба

Погледот служи за внатрешна контрола и ревизија на интегритетот на податоците. Со него менаџерите на магацини проверуваат дали бројката на артикли во главниот каталог се поклопува со збирот на количините низ сите магацини и со бројот на активни сериски броеви.

```
[2026-05-19 23:38:36] advdb_202526l_prj_blin...myschema> SELECT *
          FROM v_inventory_drift_sync_report
          WHERE variant_id = 4
[2026-05-19 23:38:37] 1 row retrieved starting from 1 in 361 ms (execution: 17 ms, fetching: 344 ms)
```

3. Состојба

- **Иницијално време на извршување:** 17 ms

За мерење на времето на запишување беа извршени трансакции врз табелите product_instances и inventory_items.

```
[2026-05-19 23:46:12] advdb_202526l_prj_blin...myschema> EXPLAIN (ANALYZE, BUFFERS)
          INSERT INTO product_instances (variant_
          VALUES (
            4, -- I
            1, -- I
            'SN-DRIFT-TEST-' || random(), -- \
            'AVAILABLE'
          )
[2026-05-19 23:46:13] 8 rows retrieved starting from 1 in 373 ms (execution: 28 ms, fetching: 34
```

```
[2026-05-19 23:46:39] advdb_202526l_prj_blin...myschema> EXPLAIN (ANALYZE, BUFFERS)
UPDATE inventory_items
SET quantity = quantity + 1
WHERE variant_id = 4 AND warehouse_id = 1;
[2026-05-19 23:46:39] 9 rows retrieved starting from 1 in 365 ms (execution: 15 ms, fetching: 350 ms)
```

6. Заклучок

Времето на извршување на SELECT прашалникот е одлично и без дополнителна оптимизација. Операциите INSERT и UPDATE се извршуваат со прифатлива брзина. Системот веќе користи оптимални патеки за пристап до податоците дефинирани преку примарните клучеви.

View 10: Моментална вредност на кошничка (v_cart_current_value)

1. Филтри

- **Примарен филтер:** order_id за приказ на конкретна кошничка, а дополнително се користи user_id за приказ на активната кошничка на одреден купувач.

```
[2026-05-19 23:57:19] advdb_202526l_prj_blin...myschema> SELECT *
FROM v_cart_current_value
WHERE order_id = 3;
[2026-05-19 23:57:19] 1 row retrieved starting from 1 in 377 ms (execution: 17 ms, fetching: 360 ms)
```

```
[2026-05-19 23:58:33] advdb_202526l_prj_blin...myschema> SELECT *
FROM v_cart_current_value
WHERE user_id = 5;
[2026-05-19 23:58:33] 1 row retrieved starting from 1 in 462 ms (execution: 142 ms, fetching: 320 ms)
```

2. Употреба

Погледот се користи во реално време на Front-end апликацијата за да му ја прикаже на купувачот вкупната сума што треба да ја плати пред да ја финализира нарачката. Бидејќи табелата ORDERS содржи NULL за сумата додека статусот е 'CART', овој поглед е единствениот извор на информација за вредноста на кошничката.

3. Состојба

- **Иницијално време на извршување:** под 30 ms (просечно)
- **Време на INSERT:** 18 ms
- **Време на UPDATE:** 17 ms

```
[2026-05-20 00:01:02] advdb_202526l_prj_blin...myschema> EXPLAIN (ANALYZE, BUFFERS)
INSERT INTO order_items (order_id, price_history_id, variant_id, quantity, unit_price)
VALUES (
  3,
  863832,
  228416,
  1,
  16026.58
);
[2026-05-20 00:01:02] 9 rows retrieved starting from 1 in 385 ms (execution: 18 ms, fetching: 367 ms)
```

```
[2026-05-19 23:56:47] advdb_202526l_prj_blin...myschema> EXPLAIN (ANALYZE, BUFFERS)
UPDATE order_items
SET quantity = quantity + 1
WHERE order_id = 3 AND variant_id = 228416;
[2026-05-19 23:56:48] 11 rows retrieved starting from 1 in 364 ms (execution: 17 ms, fetching: 347 ms)
```

5. Заклучок

Погледот е типичен агрегациски преглед кој користи GROUP BY и математички функции врз под-сет на податоци (само статус 'CART'). Постоечките индекси на примарните клучеви (order_id) веќе обезбедуваат максимална брзина за филтрирање.

View 11: Статус на исполнување на ставки (v_order_item_fulfillment_status)

1. Филтри и употреба

- **Примарен филтер:** order_id. Погледот овозможува преглед на тоа дали секоја поединечна ставка од нарачката е целосно, делумно или воопшто не е испратена (FULLY_SHIPPED, PARTIALLY_SHIPPED, NOT_SHIPPED).
- **Употреба:** Овој поглед е клучен за логистиката и за корисничкиот интерфејс. Купувачот и администраторот можат точно да видат колку парчиња од нарачаната количина се веќе во пат, а колку доцнат.

2. Тест прашалник

```
[2026-05-20 00:06:45] advdb_202526l_prj_blin...myschema> SELECT *
                                FROM v_order_item_fulfillment_status
                                WHERE order_id = 38087
[2026-05-20 00:06:46] 1 row retrieved starting from 1 in 537 ms (execution: 215 ms, fetching: 322 ms)
```

3. Состојба

- **Време на извршување:** 215 ms
- **Време на INSERT:** 12 ms

```
[2026-05-20 00:23:54] advdb_202526l_prj_blin...myschema> INSERT INTO shipment_items (shipment_item_id, shipment_id, order_item_id, instance_id, quantity_shipped)
                                VALUES (
                                (SELECT COALESCE(MAX(shipment_item_id), 0) + 1 FROM shipment_items),
                                15525,
                                38087,
                                4399804,
                                1
                                )
[2026-05-20 00:23:54] 1 row affected in 12 ms
```

- **Време на UPDATE:** 14 ms

```
[2026-05-20 00:24:43] advdb_202526l_prj_blin...myschema> EXPLAIN (ANALYZE, BUFFERS)
                                UPDATE shipment_items
                                SET quantity_shipped = 1
                                WHERE shipment_item_id = 38087
[2026-05-20 00:24:43] 7 rows retrieved starting from 1 in 432 ms (execution: 14 ms, fetching: 418 ms)
```

5. Заклучок

Иако SEQ_SCAN врз 1.6 милиони редови е релативно бавна операција, времето на извршување од 215 ms е во рамките на прифатливото за агрегациски/summary поглед. Погледот веќе го користи постоечкиот индекс idx_v2_order_items_order_id на табелата order_items, што овозможува брзо пронаоѓање на основните ставки.

View 12: Преглед на пратки и плаќања (v_shipment_payment_summary)

1. Филтри и употреба

- **Примарен филтер:** order_id и shipment_id.
- **Употреба:** Овој поглед се користи од секторот за финансии и логистика за да се потврди дека секоја физичка пратка е покриена со успешна трансакција (плаќање) пред да биде предадена на купувачот.

2. Тест прашалници

12A:

```
2026-05-18 04:04:00] advdb_202526l_prj_blin...myschema> SELECT * FROM v_shipment_payment_summary
WHERE shipment_id = 1000
2026-05-18 04:04:01] 1 row retrieved starting from 1 in 438 ms (execution: 84 ms, fetching: 354 ms)
```

Operation	Params	Rows	Total Cost	Startup Cost
↳ Select				
↳ Nested Loops (Neste		1	16.91	0.86
↳ Index Scan	table: shipments; index: idx_v2_shipments_order_id;	1	8.44	0.42
↳ Index Scan	table: payments; index: payments_transaction_id_key;	1	8.46	0.44

12B:

```
[2026-05-18 04:05:10] advdb_202526l_prj_blin...myschema> SELECT * FROM v_shipment_payment_summary
WHERE order_id = 78085
[2026-05-18 04:05:11] 1 row retrieved starting from 1 in 370 ms (execution: 42 ms, fetching: 328 ms)
```

Operation	Params	Rows	Total Cost	Startup Cost	Property	Value
↳ Select					Operation	select
↳ Nested Loops (Neste		1	16.91	0.86		
↳ Index Scan	table: shipments; index: shipments_pkey;	1	8.44	0.42		
↳ Index Scan	table: payments; index: payments_transaction_id_key;	1	8.46	0.44		

3. Состојба

- **Иницијално време (12A):** 84 ms
- **Иницијално време (12Б):** 42 ms
- **Најбавни операции:** Nested Loop спојување. Иако се користеа постоечките индекси на примарните клучеви, спојувањето беше поспоро поради комплексниот услов за поврзување на трансакциите со нарачките.
- **Време на INSERT (Пред):** 36 ms

```
[2026-05-18 04:10:02] advdb_202526l_prj_blin...myschema> EXPLAIN (ANALYZE, BUFFERS)
INSERT INTO payments (order_id, payment_method, a
VALUES (
78085,
'CARD',
2500.00,
'SUCCESS',
'TEN-78985-MAIN',
NOW()
)
[2026-05-18 04:10:02] 7 rows retrieved starting from 1 in 396 ms (execution: 36 ms, fetching: 360 ms)
```

- **Време на UPDATE (Пред):** 251 ms

```
[2026-05-18 04:06:18] advdb_202526l_prj_blin...myschema> EXPLAIN ANALYZE
UPDATE payments
SET payment_status = 'FAIL'
WHERE order_id = 78085 AND transaction_id LIKE '%'
[2026-05-18 04:06:19] 6 rows retrieved starting from 1 in 589 ms (execution: 251 ms, fetching: 338 ms)
```

4. Заклучок

Времето на извршување е во рамките на прифатливото. Системот покажува задоволителна ефикасност при финансиски проверки. Не е потребно додавање на нови индекси.

View 13: Следење на нарачки за купувач (v_customer_order_tracking)

1. Филтри и употреба

- **Примарен филтер:** order_id и customer_id.
- **Употреба:** Ова е примарниот поглед кој го користат купувачите во мобилната апликација за да го следат статусот на својата нарачка во реално време, вклучувајќи ги ставките, продавачите и информациите од курирската служба.

2. Тест прашалник

```
[2026-05-18 04:23:14] advdb_202526l_prj_blin...myschema> SELECT *
                                FROM v_customer_order_tracking
                                WHERE order_id = 24090
[2026-05-18 04:23:15] 1 row retrieved starting from 1 in 1 s 56 ms (execution: 707 ms, fetching: 349 ms)
```

3. Состојба ПРЕД индексирање

- **Иницијално време на извршување:** 707 ms
- **Најбавни операции (Bottlelenecks):**
- Parallel Seq Scan врз табелите orders и shipments. Поради големиот број на JOIN операции (вкупно 8 табели) и подпрашалници за string_agg, базата троши многу време на пребарување на поврзаните записи без индексна поддршка.
- **Време на INSERT (Пред):** 61 ms

```
[2026-05-20 01:10:09] advdb_202526l_prj_blin...myschema> EXPLAIN (ANALYZE, BUFFERS)
                                INSERT INTO orders (user_id, shipping_method_id, ord
                                VALUES (
                                888,
                                1,
                                NOW(),
                                15400.00,
                                'PAID'
                                )
[2026-05-20 01:10:09] 10 rows retrieved starting from 1 in 387 ms (execution: 61 ms, fetching: 326 ms)
```

- **Време на UPDATE (Пред):** 428 ms

```
[2026-05-20 01:10:53] advdb_202526l_prj_blin...myschema> EXPLAIN (ANALYZE, BUFFERS)
                                UPDATE orders
                                SET
                                status = 'PAID',
                                order_date = NOW(),
                                total_amount = 5500.00
                                WHERE user_id = 888
                                AND status = 'CART'
[2026-05-20 01:10:54] 8 rows retrieved starting from 1 in 751 ms (execution: 428 ms, fetching: 323 ms)
```

4. Воведување на индекси

Иницијалното SELECT време од 707 ms е директно неприфатливо за примарниот поглед на мобилната апликација, каде купувачот очекува инстантен одговор.

Дополнително, UPDATE времето пред индексирање изнесуваше 428 ms — и тоа е само поради недостаток на индексна поддршка при JOIN операциите, не поради природата на операцијата. Овие вредности јасно укажуваат дека базата врши скапи Seq Scan операции на критични табели.

За да се оптимизира поврзувањето на логистиката со трансакциите, воведени се следните индекси:

```
CREATE INDEX idx_v13_shipments_order_id ON shipments(order_id);
CREATE INDEX idx_v13_shipment_items_lookup ON shipment_items(order_item_id);
```

5. Состојба ПОСЛЕ индексирање

- **Ново време на извршување: 25 ms**

```
[2026-05-20 01:20:01] advdb_202526l_prj_blin...myschema> SELECT *
                    FROM v_customer_order_tracking
                    WHERE order_id = 24090
[2026-05-20 01:20:01] 1 row retrieved starting from 1 in 366 ms (execution: 25 ms, fetching: 341 ms)
```

Најбавни операции сега: Операцијата Seq Scan на shipment_items е целосно заменета со **Index Scan**. Базата сега врши директен пристап до потребните ставки, со што времето на извршување падна за **99%** во споредба со почетната состојба.

Operation	Params	Rows	Actual ...	Total ...	Actual Tot...	Startup...	Actual Star...
↳ Select							
↳ Sort		1	1	149.23	233.58	149.23	233.574
↳ Nested Loops (Nested Loop)		1	1	149.22	233.571	1.28	233.563
↳ Nested Loops (Nested Loop)		1	1	16.9	233.086	0.85	233.083
↳ Index Scan	table: orders; index: orders_pkey;	1	1	8.45	0.085	0.43	0.084
↳ Index Scan	table: shipments; index: idx_v2_ship...	1	1	8.44	232.994	0.42	232.993
↳ Index Scan	table: user_profiles; index: user_pro...	1	1	8.44	0.044	0.42	0.044
↳ Aggregate		1	1	62.28	0.219	62.27	0.217
↳ Nested Loops (Nested Loop)		6	1	62.18	0.205	1.29	0.2
↳ Nested Loops (Nested Loop)		6	1	59.36	0.139	0.86	0.135
↳ Index Scan	table: order_items; index: idx_v2_or...	6	1	8.69	0.044	0.43	0.042
↳ Index Scan	table: product_variants; index: prod...	1	1	8.45	0.089	0.43	0.088
↳ Index Scan	table: products; index: products_pk...	1	1	0.47	0.061	0.42	0.061
↳ Aggregate		1	1	61.59	0.205	61.58	0.203
↳ Sort		1	1	61.57	0.197	61.56	0.195
↳ Nested Loops (Nested Loop)		1	1	61.55	0.182	2.14	0.178
↳ Nested Loops (Nested Loop)		1	1	60.79	0.141	1.72	0.137
↳ Nested Loops (Nested Loop)		1	1	60.35	0.105	1.3	0.102
↳ Nested Loops (Nested Loop)		1	1	59.42	0.053	0.86	0.051
↳ Index Scan	table: order_items; index: idx_v2_or...	6	1	8.69	0.009	0.43	0.008
↳ Index Scan	table: shipment_items; index: idx_v1...	1	1	8.45	0.041	0.43	0.04
↳ Index Scan	table: product_instances; index: pro...	1	1	0.93	0.048	0.43	0.048
↳ Index Scan	table: warehouses; index: warehous...	1	1	0.44	0.032	0.42	0.032
↳ Index Scan	table: user_profiles; index: user_pro...	1	1	0.77	0.036	0.42	0.036

6. Влијание врз запишувањето (Write-Impact Analysis)

- **Ново време на INSERT: 93 ms**

```
[2026-05-20 01:41:46] advdb_202526l_prj_blin...myschema> EXPLAIN (ANALYZE, BUFFERS)
                    INSERT INTO orders (user_id, shipping_method_id, order_date, total_amount, status)
                    VALUES (
                        888,
                        1,
                        NOW(),
                        15400.00,
                        'PAID'
                    )
[2026-05-20 01:41:47] 8 rows retrieved starting from 1 in 410 ms (execution: 93 ms, fetching: 317 ms)
```

- **Ново време на UPDATE: 20 ms**

```
[2026-05-18 04:31:13] advdb_202526l_prj_blin...myschema> EXPLAIN (ANALYZE, BUFFERS)
UPDATE orders
SET
    status = 'PAID',
    order_date = NOW(),
    total_amount = 5500.00
WHERE user_id = 888
AND status = 'CART'
[2026-05-18 04:31:13] 9 rows retrieved starting from 1 in 413 ms (execution: 20 ms, fetching: 393 ms)
```

7. Заклучок

Постои значително подобрување кај SELECT прашалникот (од 707 ms на 25 ms) и драстично забрзување кај UPDATE операцијата (од 428 ms на 20 ms). Забавувањето на INSERT е прифатливо (од 61 ms на 93 ms) и е очекуван резултат на одржувањето на новите индексни структури. Овој компромис е јасно оправдан со оглед на критичната улога на погледот во мобилната апликација.