

# Индекси – Library Management System

191513, 191515, 191517

## All reservations view

1. Со кои филтри најчесто ќе се извршува погледот?  
Овој поглед најчесто ќе се извршува со филтрирање по име на книга.
2. Кои се случаите на употреба на погледот? Дали ни се важни перформансите, или станува збор за аналитички поглед?  
За овој поглед важни се перформансите бидејќи библиотекарите го користат за да ги прегледаат сите резервации.
3. Кое е иницијалното време за извршување на погледот (со сите најчесто користени филтри)? Дали ова време е прифатливо за апликацијата?  
Пред индексирање, иницијалното време на извршување е околу 4 секунди, што може да биде оптимизирано.

```
[2023-05-18 20:34:01] Connected
postgres@public> SELECT t.*
                  FROM public.all_reservations_view t
                  LIMIT 501
[2023-05-18 20:34:06] 500 rows retrieved starting from 1 in 5 s 161 ms (execution: 4 s 663 ms, fetching: 498 ms)
```

### Планот на извршување пред индекс:

Operation	Params	Rows	Total Cost	Startup Cost	Raw Desc
Aggregate		2894994605	2.5590299845E8	616246.06	Strategy = SortedPartial Mode ...
Nested Loops (Nested Loop)		2894994605	1.6181567378E8	616246.06	Parent Relationship = OuterPara...
Nested Loops (Nested Loop)		2894994605	8.847380298E7	616245.77	Parent Relationship = OuterPara...
Unknown (Gather Merge)		2237530	876842.59	616245.35	Parent Relationship = OuterPara...
Sort		932304	617576.08	615245.32	Parent Relationship = OuterPara...
Hash Join		932304	19323.33	1676.67	Parent Relationship = OuterPara...
Hash Join		208237	8331.31	1569.48	Parent Relationship = OuterPara...
Hash Join		208237	7707.06	1493.02	Parent Relationship = OuterPara...
Full Scan (Seq Scan)	table: book_reservation;	208237	5667.37	0.0	Parent Relationship = OuterPara...
Transformation (Hash)		50001	868.01	868.01	Parent Relationship = InnerParall...
Full Scan (Seq Scan)	table: book_copy;	50001	868.01	0.0	Parent Relationship = OuterPara...
Transformation (Hash)		1976	51.76	51.76	Parent Relationship = InnerParall...
Full Scan (Seq Scan)	table: patron;	1976	51.76	0.0	Parent Relationship = OuterPara...
Transformation (Hash)		2252	79.03	79.03	Parent Relationship = InnerParall...
Hash Join		2252	79.03	34.47	Parent Relationship = OuterPara...
Hash Join		2252	59.79	21.32	Parent Relationship = OuterPara...
Full Scan (Seq Scan)	table: book_category;	2252	32.52	0.0	Parent Relationship = OuterPara...
Transformation (Hash)		503	15.03	15.03	Parent Relationship = InnerParall...
Full Scan (Seq Scan)	table: book;	503	15.03	0.0	Parent Relationship = OuterPara...
Transformation (Hash)		140	11.4	11.4	Parent Relationship = InnerParall...
Full Scan (Seq Scan)	table: category;	140	11.4	0.0	Parent Relationship = OuterPara...
Index Scan (Index Only Scan)	table: book_author; index: book_author_pkey;	1299	23.36	0.42	Parent Relationship = InnerParall...
Unknown (Memoize)		1	0.31	0.29	Parent Relationship = InnerParall...
Index Scan	table: author; index: author_pkey;	1	0.3	0.28	Parent Relationship = OuterPara...

4. Доколку не е, направете анализа на планот на извршување. Кои се најбавни операции? Како може да се подобрат? Пред додавање на индекси да се измери и времето на операции на insert i update на табелата во која ќе се додава индекс.

### Време и план на Insert пред индекс:

```
postgres@public> INSERT INTO book (id,title, publisher_id, publication_date, summary, book_request_id)
                  VALUES (53334,'The Great Gatsby', 1, '2022-05-10', 'A classic novel by F. Scott Fitzgerald', NULL)
[2023-05-18 20:43:57] 1 row affected in 6 ms
```

Operation	Params	Rows	Total Cost	Startup Cost	Raw Desc
Insert		1	0.01	0.0	Parent Relationship = OuterParallel Aware = f...
Value (Result)		1	0.01	0.0	Parent Relationship = OuterParallel Aware = f...

Време и план на Update пред индекс:

```
postgres.public> UPDATE book
                    SET title = 'New Title',
                        publisher_id = 2,
                        publication_date = '2023-01-01',
                        summary = 'Updated summary'
                    WHERE id = 1
[2023-05-18 20:45:36] 1 row affected in 4 ms
```

Operation	Params	Rows	Total Cost	Startup Cost	Raw Desc
Update					
Index Scan	table: book; index: book_pkey;	1	8.29	0.27	Parent Relationship = OuterParallel Aware

## 5. По индексирање:

```
CREATE INDEX idx_book_title ON book (title);
```

View

```
postgres.public> SELECT t.*
                  FROM public.all_reservations_view t
                  LIMIT 501
[2023-05-18 20:55:35] 500 rows retrieved starting from 1 in 4 s 45 ms (execution: 3 s 832 ms, fetching: 213 ms)
```

Operation	Params	Rows	Total Cost	Startup Cost	Raw Desc
Select					
Aggregate		2894994605	2.5590299845E8	616246.06	Strategy = SortedPartial Mode ...
Nested Loops (Nested Loop)		2894994605	1.6181567379E8	616246.06	Parent Relationship = OuterPara...
Nested Loops (Nested Loop)		2894994605	8.947380298E7	616245.77	Parent Relationship = OuterPara...
Unknown (Gather Merge)		2237530	876842.59	616245.35	Parent Relationship = OuterPara...
Sort		932304	617576.08	615243.32	Parent Relationship = OuterPara...
Hash Join		932304	19323.33	1676.67	Parent Relationship = OuterPara...
Hash Join		208237	8331.31	1569.48	Parent Relationship = OuterPara...
Hash Join		208237	7707.06	1493.02	Parent Relationship = OuterPara...
Full Scan (Seq Scan)	table: book_reservation;	208237	5667.37	0.0	Parent Relationship = OuterPara...
Transformation (Hash)		50001	868.01	868.01	Parent Relationship = InnerParall...
Full Scan (Seq Scan)	table: book_copy;	50001	868.01	0.0	Parent Relationship = OuterPara...
Transformation (Hash)		1976	51.76	51.76	Parent Relationship = InnerParall...
Full Scan (Seq Scan)	table: patron;	1976	51.76	0.0	Parent Relationship = OuterPara...
Transformation (Hash)		2252	79.03	79.03	Parent Relationship = InnerParall...
Hash Join		2252	79.03	34.47	Parent Relationship = OuterPara...
Hash Join		2252	59.79	21.32	Parent Relationship = OuterPara...
Full Scan (Seq Scan)	table: book_category;	2252	32.52	0.0	Parent Relationship = OuterPara...
Transformation (Hash)		503	15.03	15.03	Parent Relationship = InnerParall...
Full Scan (Seq Scan)	table: book;	503	15.03	0.0	Parent Relationship = OuterPara...
Transformation (Hash)		140	11.4	11.4	Parent Relationship = InnerParall...
Full Scan (Seq Scan)	table: category;	140	11.4	0.0	Parent Relationship = OuterPara...
Index Scan (Index Only Scan)	table: book_author; index: book_author_pkey;	1299	23.36	0.42	Parent Relationship = InnerParall...
Unknown (Memoize)		1	0.31	0.29	Parent Relationship = InnerParall...
Index Scan	table: author; index: author_pkey;	1	0.3	0.28	Parent Relationship = OuterPara...

Insert по индексирање:

```
postgres.public> INSERT INTO book (id,title, publisher_id, publication_date, summary, book_request_id)
                  VALUES (53335, 'The Great Gatsby', 1, '2022-05-10', 'A classic novel by F. Scott Fitzgerald', NULL)
[2023-05-18 20:47:52] 1 row affected in 11 ms
```

Operation	Params	Rows	Total Cost	Startup Cost	Raw Desc
Insert					
Value (Result)		1	0.01	0.0	Parent Relationship = OuterParallel Aware = f.

Update по индексирање:

```
postgres.public> UPDATE book
SET title = 'New Title',
    publisher_id = 2,
    publication_date = '2023-01-01',
    summary = 'Updated summary after index'
WHERE id = 2
[2023-05-18 20:49:37] 1 row affected in 7 ms
```

Operation	Params	Rows	Total Cost	Startup Cost	Raw Desc
Update					
Index Scan	table: book; index: book_pkey;	1	8.29	0.27	Parent Relationship = OuterParallel Aware.

## Book Availability view

1. Со кои филтри најчесто ќе се извршува погледот?  
Најчесто во овој поглед би се пребарувало по име на книгата.
2. Кои се случаите на употреба на погледот? Дали ни се важни перформансите, или станува збор за аналитички поглед?  
Овој поглед се користи кога треба да се провери дали книгата е достапна, затоа перформансите се важни.
3. Кое е иницијалното време за извршување на погледот (со сите најчесто користени филтри)? Дали ова време е прифатливо за апликацијата?  
Иницијалното време на извршување на погледот е 23 секунди што не е прифатливо за апликацијата.

book_title	edition	format	status	location
1 unknown	0	PAPERBACK	AVAILABLE	unknown--1
2 The James Dean Affair: A Neil Gulliver & Stevie Mar...	1	PAPERBACK	AVAILABLE	B-6
3 Die Korrekturen.	2	HARDCOVER	AVAILABLE	I-4
4 Possession: A Romance	1	PAPERBACK	AVAILABLE	X-1
5 Trãnsit (Area contemporãnia)	1	PAPERBACK	AVAILABLE	G-2
6 The Oneprince (The Redaemian Chronicles, Book 1)	1	HARDCOVER	AVAILABLE	Y-5
7 The Iron Giant	3	PAPERBACK	AVAILABLE	F-7

4. Доколку не е, направете анализа на планот на извршување. Кои се најбавни операции? Како може да се подобрат? Пред додавање на индекси да се измери и времето на операции на insert i update на табелата во која ќе се додава индекс.

Insert BookBorrow before index

Operation	Params	Rows	Actual Rows	Total Cost	Actual Total Time	Startup Cost	Actual Startup Time	Raw Desc
Insert								
Value (Result)		1	1	0.02	0.01	0.0	0.009	Planning Time = 0.026; Trigger... Parent Relationship = Outer; P...

Update BookBorrow before index

Operation	Params	Rows	Actual Rows	Total Cost	Actual Total Time	Startup Cost	Actual Startup Time	Raw Desc
Update								
Index Scan	table: book_borrow; index: book_borrow_pkey;	1	1	8.44	0.501	0.42	0.5	Planning Time = 0.064; Tr... Parent Relationship = Out...

## Book Availability select before index

```

17 SELECT * from book_availability
18 WHERE book_title = 'Congo';
19

```

Operation	Params	Rows	Actual Rows	Total Cost	Actual Total Time	Startup Cost	Actual Startup Time	Raw Desc
Select								Planning Time = 0.417; Trigg...
Hash Join		83	100	2350649.7	7872.536	16179.44	535.286	Parallel Aware = false; Asyn...
Full Scan (Seq Scan)	table: book_copy;	50101	50000	870.01	4.637	0.0	0.009	Parent Relationship = Outer...
Transformation (Hash)		1	1	19.51	0.073	19.51	0.072	Parent Relationship = Inner...
Full Scan (Seq Scan)	table: book;	1	1	19.51	0.064	0.0	0.039	Parent Relationship = Outer...
Full Scan (Seq Scan)	table: book_borrow;	133967	123722	15825.0	72.022	0.0	0.08	Parent Relationship = SubPL...
Temporary (Materialize)		500000	475001	20279.0	43.17	0.0	0.004	Parent Relationship = SubPL...
Full Scan (Seq Scan)	table: book_borrow;	500000	475001	15825.0	68.087	0.0	0.021	Parent Relationship = Outer...
Hash Join		50000	0	17344.3	0.0	2.15	0.0	Parent Relationship = SubPL...
Full Scan (Seq Scan)	table: book_borrow;	50000	0	17075.0	0.0	0.0	0.0	Parent Relationship = Outer...
Transformation (Hash)		51	0	1.51	0.0	1.51	0.0	Parent Relationship = Inner; ...
Full Scan (Seq Scan)	table: librarian;	51	0	1.51	0.0	0.0	0.0	Parent Relationship = Outer...
Full Scan (Seq Scan)	table: book_location;	1	1	5.27	0.025	0.0	0.018	Parent Relationship = SubPL...

- По потреба, да се преуреди прашалникот и/или да се додадат индекси. Кој е планот на извршување по додадениот индекс? Кое е времето на извршување сега? Ако не е прифатливо се повторуваат чекор 4 и 5.

## Book Borrow index

```
CREATE INDEX book_borrow_id_index on book_borrow(book_copy_id);
```

## Book Availability select after BookBorrow index

```

16 SELECT * from book_availability
17 WHERE book_title = 'Congo';
18

```

Operation	Params	Rows	Actual Rows	Total Cost	Actual Total Time	Startup Cost	Actual Startup Time	Raw Desc
Select								Planning Time = 1.852; T...
Hash Join		99	100	2005934.95	7840.564	15911.96	286.08	Parallel Aware = false; As...
Full Scan (Seq Scan)	table: book_copy;	50001	50001	869.01	3.729	0.0	0.007	Parent Relationship = Ou...
Transformation (Hash)		1	1	18.32	0.088	18.32	0.087	Parent Relationship = Inn...
Full Scan (Seq Scan)	table: book;	1	1	18.32	0.085	0.0	0.048	Parent Relationship = Ou...
Full Scan (Seq Scan)	table: book_borrow;	127269	123724	15575.03	74.318	0.0	0.016	Parent Relationship = Su...
Temporary (Materialize)		475003	475003	14599.35	42.449	0.42	0.008	Parent Relationship = Su...
Index Scan (Index Only)	table: book_borrow; index: book_borrow_id_index;	475003	475003	10359.34	45.721	0.42	0.449	Parent Relationship = Ou...
Hash Join		47500	0	12207.24	0.0	534.7	0.0	Parent Relationship = Su...
Bitmap Index Scan (B)	table: book_borrow;	47500	0	11951.3	0.0	532.55	0.0	Parent Relationship = Ou...
Bitmap Index Scan	index: book_borrow_id_index;	47500	0	520.67	0.0	0.0	0.0	Parent Relationship = Ou...
Transformation (Hash)		51	0	1.51	0.0	1.51	0.0	Parent Relationship = Inn...
Full Scan (Seq Scan)	table: librarian;	51	0	1.51	0.0	0.0	0.0	Parent Relationship = Ou...
Full Scan (Seq Scan)	table: book_location;	1	1	5.27	0.024	0.0	0.016	Parent Relationship = Su...

## Insert BookBorrow after index

```

8
9 INSERT INTO book_borrow(user_id, book_copy_id, book_checkout, book_return, damage, checkout_librarian_id, return_librarian_id)
10 values (1, 488, now(), null, null, 3842, null);
11

```

Operation	Params	Rows	Actual Rows	Total Cost	Actual Total Time	Startup Cost	Actual Startup Time	Raw Desc
Insert								Planning Time = 0.026; Trigger...
Value (Result)		1	1	0.02	0.027	0.0	0.027	Parent Relationship = Outer; P...

## Update BookBorrow after index

```

12 UPDATE book_borrow
13 SET book_copy_id = 123
14 where id = 253;
15

```

Operation	Params	Rows	Actual Rows	Total Cost	Actual Total Time	Startup Cost	Actual Startup Time	Raw Desc
Update								Planning Time = 0.11; Trigg...
Index Scan	table: book_borrow; index: book_borrow_pkey;	1	1	8.44	0.017	0.42	0.016	Parent Relationship = Out...

## Insert BookCopy before index

```

27 INSERT INTO book_copy(book_id, location_id, edition, book_format)
28 values (123, 488, 1, 'PAPERBACK');
29

```

Operation	Params	Rows	Actual Rows	Total Cost	Actual Total Time	Startup Cost	Actual Startup Time	Raw Desc
Insert								Planning Time = 0.022; Trigger...
Value (Result)		1	1	0.01	0.008	0.0	0.007	Parent Relationship = Outer; P...

## Update BookCopy before index

```

31 UPDATE book_copy
32 SET book_format = 'UNKNOWN'
33 WHERE id = 567;

```

Operation	Params	Rows	Actual Rows	Total Cost	Actual Total Time	Startup Cost	Actual Startup Time	Raw Desc
Update								Planning Time = 0.054; Tr...
Index Scan	table: book_copy; index: book_copy_pkey;	1	1	8.31	0.01	0.29	0.009	Parent Relationship = Out...

## BookCopy index

```
CREATE INDEX book_id_book_copy_index ON book_copy(book_id);
```

## Book Availability after BookCopy index

```
17 |
18 | SELECT * from book_availability
19 | WHERE book_title = 'Congo';
```

Operation	Params	Rows	Actual Rows	Total Cost	Actual Total Time	Startup Cost	Actual Startup Time	Raw Desc
▼ Select								Planning Time = 1.927; T...
▼ Nested Loops (Nested Loop)		99	100	2006165.76	7520.527	15898.69	280.509	Parallel/Aware = false; As...
▼ Full Scan (Seq Scan)	table: book;	1	1	18.32	0.06	0.0	0.039	Parent Relationship = Ou...
▼ Bitmap Index Scan (Bitm	table: book_copy;	100	100	231.19	0.238	5.07	0.033	Parent Relationship = Inn...
▼ Bitmap Index Scan	index: book_copy_id_index;	100	100	5.04	0.019	0.0	0.019	Parent Relationship = Ou...
▼ Full Scan (Seq Scan)	table: book_borrow;	127269	123726	15575.03	72.344	0.0	0.008	Parent Relationship = Su...
▼ Temporary (Materialize)		475003	475005	14590.35	40.853	0.42	0.003	Parent Relationship = Su...
▼ Index Scan (Index Onl	table: book_borrow; index: book_borrow_id_index;	475003	475005	10359.94	43.303	0.42	0.013	Parent Relationship = Ou...
▼ Hash Join		47500	0	12207.24	0.0	534.7	0.0	Parent Relationship = Su...
▼ Bitmap Index Scan (Bi	table: book_borrow;	47500	0	11951.3	0.0	532.55	0.0	Parent Relationship = Ou...
▼ Bitmap Index Scan	index: book_borrow_id_index;	47500	0	520.67	0.0	0.0	0.0	Parent Relationship = Ou...
▼ Transformation (Hash)		51	0	1.51	0.0	1.51	0.0	Parent Relationship = Inn...
▼ Full Scan (Seq Sca	table: librarian;	51	0	1.51	0.0	0.0	0.0	Parent Relationship = Ou...
▼ Full Scan (Seq Scan)	table: book_location;	1	1	5.27	0.023	0.0	0.015	Parent Relationship = Su...

## Insert BookCopy after index

```
74 | UPDATE book_copy
75 | SET book_format = 'UNKNOWN'
76 | WHERE id = 567;
```

Operation	Params	Rows	Actual Rows	Total Cost	Actual Total Time	Startup Cost	Actual Startup Time	Raw Desc
▼ Update								Planning Time = 0.059; Tri...
▼ Index Scan	table: book_copy; index: book_copy_pkey;	1	1	8.31	0.021	0.29	0.012	Parent Relationship = Out...

## Update BookCopy after index

```
71 | INSERT INTO book_copy(book_id, location_id, edition, book_format)
72 | VALUES(123, 486, 1, 'PAPERBACK');
```

Operation	Params	Rows	Actual Rows	Total Cost	Actual Total Time	Startup Cost	Actual Startup Time	Raw Desc
▼ Insert								Planning Time = 0.023; Trigger...
▼ Value (Result)		1	1	0.01	0.014	0.0	0.013	Parent Relationship = Outer; P...

## Unreturned books view

1. Со кои филтри најчесто ќе се извршува погледот?  
Овој погледно најчесто би се филтрирал по card\_number на членот.
2. Кои се случаите на употреба на погледот? Дали ни се важни перформансите, или станува збор за аналитички поглед?  
Овој поглед ќе се употребува кога библиотекарите сакаат да проверат кои книги не се вратени и кој член ги има позајмено.
3. Кое е иницијалното време за извршување на погледот (со сите најчесто користени филтри)? Дали ова време е прифатливо за апликацијата?  
Иницијалното време на извршување на погледот е 2 секунди, но бидејќи овој поглед ќе се користи многу често сметаме дека треба да е оптимизиран.

```
postgres@postgres> SELECT t.*
FROM public.unreturned_books t
LIMIT 501
[2023-05-18 21:03:22] 500 rows retrieved starting from 1 in 2 s 151 ms (execution: 1 s 862 ms, fetching: 289 ms)
```

Operation	Params	Rows	Total Cost	Startup Cost	Raw Desc
Aggregate		24502400	1,241435967E7	15514.4	Strategy = SortedPartial Mode = .
Nested Loops (Nested Loop)		171087682	9112972.44	15514.4	Parent Relationship = OuterParall.
Nested Loops (Nested Loop)		171087682	4837369.2	15514.11	Parent Relationship = OuterParall.
Unknown (Gather Merge)		132223	30914.4	15513.69	Parent Relationship = OuterParall.
Sort		55097	14651.41	14513.66	Parent Relationship = OuterParall.
Hash Join		55097	7913.86	98.08	Parent Relationship = OuterParall.
Nested Loops (Nested Loop)		55097	7746.69	76.76	Parent Relationship = OuterParall.
Hash Join		55097	6365.73	76.46	Parent Relationship = OuterParall.
Full Scan (Seq Scan)	table: book_borrow;	55097	6144.33	0.0	Parent Relationship = OuterParall.
Transformation (Hash)		1976	51.76	51.76	Parent Relationship = OuterParall.
Full Scan (Seq Scan)	table: patron;	1976	51.76	0.0	Parent Relationship = OuterParall.
Unknown (Memoize)		1	0.33	0.3	Parent Relationship = OuterParall.
Index Scan	table: book_copy; index: book_copy_pkey;	1	0.32	0.29	Parent Relationship = InnerParall.
Transformation (Hash)		503	15.03	15.03	Parent Relationship = InnerParall.
Full Scan (Seq Scan)	table: book;	503	15.03	0.0	Parent Relationship = InnerParall.
Index Scan (Index Only Scan)	table: book_author; index: book_author_pkey;	1299	23.36	0.42	Parent Relationship = InnerParall.
Unknown (Memoize)		1	0.31	0.29	Parent Relationship = InnerParall.
Index Scan	table: author; index: author_pkey;	1	0.3	0.28	Parent Relationship = OuterParall.

4. Доколку не е, направете анализа на планот на извршување. Кои се најбавни операции? Како може да се подобрат? Пред додавање на индекси да се измери и времето на операции на insert и update на табелата во која ќе се додава индекс.

#### Insert

```
postgres.public> INSERT INTO patron (id, email, user_password, first_name, last_name, date_of_birth, address, phone_number, card_number)
VALUES (1234567, 'example@example.com', 'password123', 'John', 'Doe', '1990-01-01', '123 Main St, City', '1234567890')
[2023-05-18 21:19:20] 1 row affected in 5 ms
```

Operation	Params	Rows	Total Cost	Startup Cost	Raw Desc
Insert		1	0.01	0.0	Parent Relationship = OuterParallel Aware = .
Value (Result)		1	0.01	0.0	Parent Relationship = OuterParallel Aware = .

#### Update

```
postgres.public> UPDATE patron
SET membership_id = 2
WHERE id = 1234567
[2023-05-18 21:21:30] 1 row affected in 7 ms
```

Operation	Params	Rows	Total Cost	Startup Cost	Raw Desc
Update		1	8.29	0.28	Parent Relationship = OuterParallel Aware = .
Index Scan	table: patron; index: patron_pkey;	1	8.29	0.28	Parent Relationship = OuterParallel Aware = .

5. По потреба, да се преуреди прашалникот и/или да се додадат индекси. Кој е планот на извршување по додадениот индекс? Кое е времето на извршување сега? Ако не е прифатливо се повторуваат чекор 4 и 5.

```
CREATE INDEX idx_patron_card_number ON patron (card_number);
```

#### Време на извршување по креирање индекс:

```
postgres.public> SELECT t.*
FROM public.unreturned_books t
LIMIT 501
[2023-05-18 21:24:22] 500 rows retrieved starting from 1 in 2 s 100 ms (execution: 1 s 799 ms, fetching: 301 ms)
```

Operation	Params	Rows	Total Cost	Startup Cost	Raw Desc
Aggregate		24514800	1,241473169E7	15514.42	Strategy = SortedPartial Mode = .
Nested Loops (Nested Loop)		171087682	9112972.46	15514.42	Parent Relationship = OuterParall.
Nested Loops (Nested Loop)		171087682	4837369.22	15514.14	Parent Relationship = OuterParall.
Unknown (Gather Merge)		132233	30914.42	15513.71	Parent Relationship = OuterParall.
Sort		55097	14651.43	14513.69	Parent Relationship = OuterParall.
Hash Join		55097	7913.88	98.1	Parent Relationship = OuterParall.
Nested Loops (Nested Loop)		55097	7746.71	76.78	Parent Relationship = OuterParall.
Hash Join		55097	6365.76	76.48	Parent Relationship = OuterParall.
Full Scan (Seq Scan)	table: book_borrow;	55097	6144.33	0.0	Parent Relationship = OuterParall.
Transformation (Hash)		1977	51.77	51.77	Parent Relationship = InnerParall.
Full Scan (Seq Scan)	table: patron;	1977	51.77	0.0	Parent Relationship = OuterParall.
Unknown (Memoize)		1	0.33	0.3	Parent Relationship = InnerParall.
Index Scan	table: book_copy; index: book_copy_pkey;	1	0.32	0.29	Parent Relationship = OuterParall.
Transformation (Hash)		503	15.03	15.03	Parent Relationship = InnerParall.
Full Scan (Seq Scan)	table: book;	503	15.03	0.0	Parent Relationship = OuterParall.
Index Scan (Index Only Scan)	table: book_author; index: book_author_pkey;	1299	23.36	0.42	Parent Relationship = InnerParall.
Unknown (Memoize)		1	0.31	0.29	Parent Relationship = InnerParall.
Index Scan	table: author; index: author_pkey;	1	0.3	0.28	Parent Relationship = OuterParall.

## 6. Insert

```
postgres.public> INSERT INTO patron (id, email, us
VALUES (12345678, 'example1@examp
[2023-05-18 21:25:57] 1 row affected in 5 ms
```

Operation	Params	Rows	Total Cost	Startup Cost	Raw Desc
Insert		1	0.01	0.0	Parent Relationship = OuterParallel Aware = .
Value (Result)		1	0.01	0.0	Parent Relationship = OuterParallel Aware = .

## Update

```
postgres.public> UPDATE patron
SET membership_id = 3
WHERE id = 12345678
[2023-05-18 21:26:26] 1 row affected in 6 ms
```

Operation	Params	Rows	Total Cost	Startup Cost	Raw Desc
Update		1	8.29	0.28	Parent Relationship = OuterParallel Aware = .
Index Scan	table: patron; index: patron_pkey;	1	8.29	0.28	Parent Relationship = OuterParallel Aware = .

## Active reservations view

1. Со кои филтри најчесто ќе се извршува погледот?

Најчесто би се пребарувало по статус на резервација (reservation status) од book reservation табелата или по име, презиме и број на картичка на patron.

2. Кои се случаите на употреба на погледот? Дали ни се важни перформансите, или станува збор за аналитички поглед?

Овој поглед најчесто би се користел за да се покажат активните резервации на корисниците на UI.

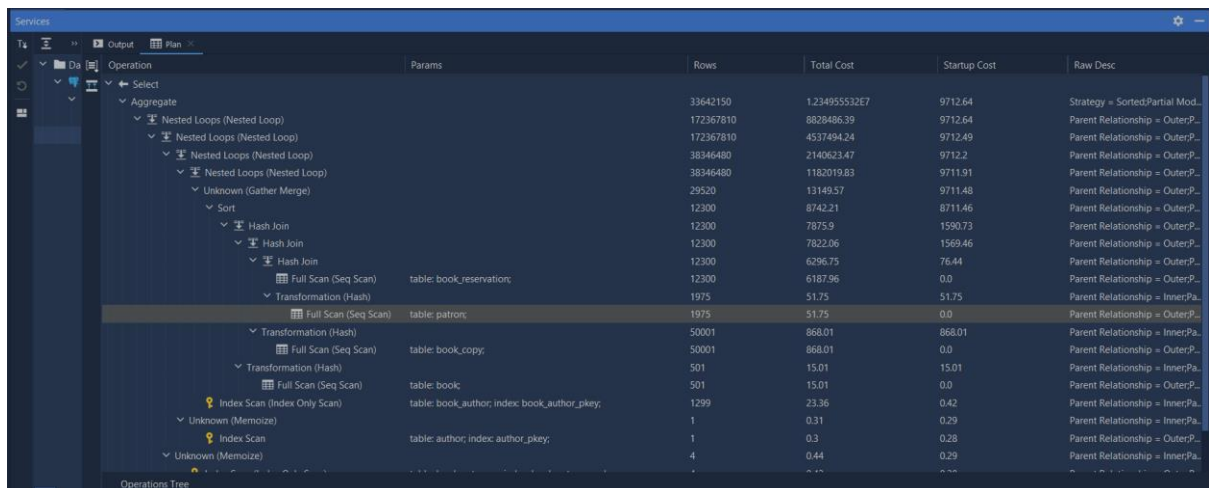
За овој случај важни ни се перформансите бидејќи погледот соединува 6 табели за да може да ги даде потребните информации.

3. Кое е иницијалното време за извршување на погледот (со сите најчесто користени филтри)? Дали ова време е прифатливо за апликацијата?

Иницијалното време на извршување беше нешто помалку од 8 секунди што не би било прифатливо за апликацијата бидејќи се користи на UI каде што крајните корисници би требало во real-time да ги имаат овие информации.

4. Доколку не е, направете анализа на планот на извршување. Кои се најбавни операции? Како може да се подобрат? Пред додавање на индекси да се измери и времето на операции на insert и update на табелата во која ќе се додава индекс.

Планот пред индекс:

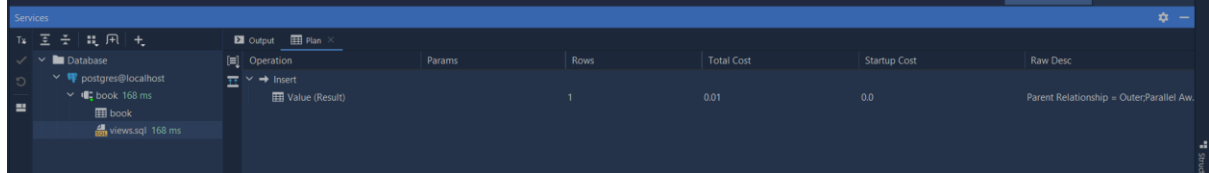


Operation	Params	Rows	Total Cost	Startup Cost	Raw Desc
Aggregate		33642150	1.23495532E7	9712.64	Strategy = Sorted/Partial Mod...
Nested Loops (Nested Loop)		172367810	8828486.39	9712.64	Parent Relationship = OuterP...
Nested Loops (Nested Loop)		172367810	4537494.24	9712.49	Parent Relationship = OuterP...
Nested Loops (Nested Loop)		38346480	2140623.47	9712.2	Parent Relationship = OuterP...
Nested Loops (Nested Loop)		38346480	1182019.83	9711.91	Parent Relationship = OuterP...
Unknown (Gather Merge)		29520	13149.57	9711.48	Parent Relationship = OuterP...
Sort		12300	8742.21	8711.46	Parent Relationship = OuterP...
Hash Join		12300	7875.9	1590.73	Parent Relationship = OuterP...
Hash Join		12300	7822.06	1569.46	Parent Relationship = OuterP...
Hash Join		12300	6296.75	76.44	Parent Relationship = OuterP...
Full Scan (Seq Scan)	table: book_reservation;	12300	6187.96	0.0	Parent Relationship = OuterP...
Transformation (Hash)		1975	51.75	51.75	Parent Relationship = InnerPa...
Full Scan (Seq Scan)	table: patron;	1975	51.75	0.0	Parent Relationship = OuterP...
Transformation (Hash)		50001	868.01	868.01	Parent Relationship = InnerPa...
Full Scan (Seq Scan)	table: book_copy;	50001	868.01	0.0	Parent Relationship = InnerPa...
Transformation (Hash)		501	15.01	15.01	Parent Relationship = InnerPa...
Full Scan (Seq Scan)	table: book;	501	15.01	0.0	Parent Relationship = OuterP...
Index Scan (Index Only Scan)	table: book_author; index: book_author_pk;	1299	23.36	0.42	Parent Relationship = InnerPa...
Unknown (Memoize)		1	0.31	0.29	Parent Relationship = InnerPa...
Index Scan	table: author; index: author_pk;	1	0.3	0.28	Parent Relationship = OuterP...
Unknown (Memoize)		4	0.44	0.29	Parent Relationship = InnerPa...



## Insert пред индекс:

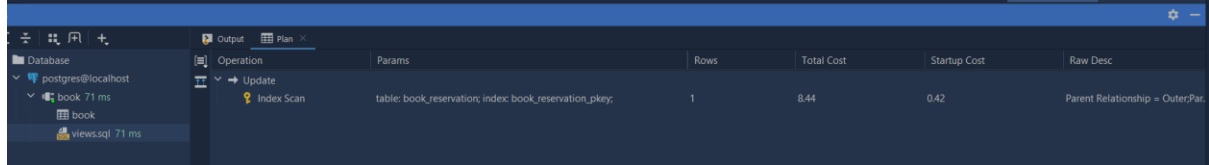
```
52 INSERT INTO book_reservation (user_id, book_copy_id, reservation_date, reservation_status)
53 VALUES (1, 100, '2023-05-19', 'ACTIVE');
54
55
```



Operation	Params	Rows	Total Cost	Startup Cost	Raw Desc
Insert		1	0.01	0.0	Parent Relationship = OuterParallel Aw...

## Update пред индекс:

```
UPDATE book_reservation
SET reservation_status = 'INACTIVE'
WHERE id = 1;
```

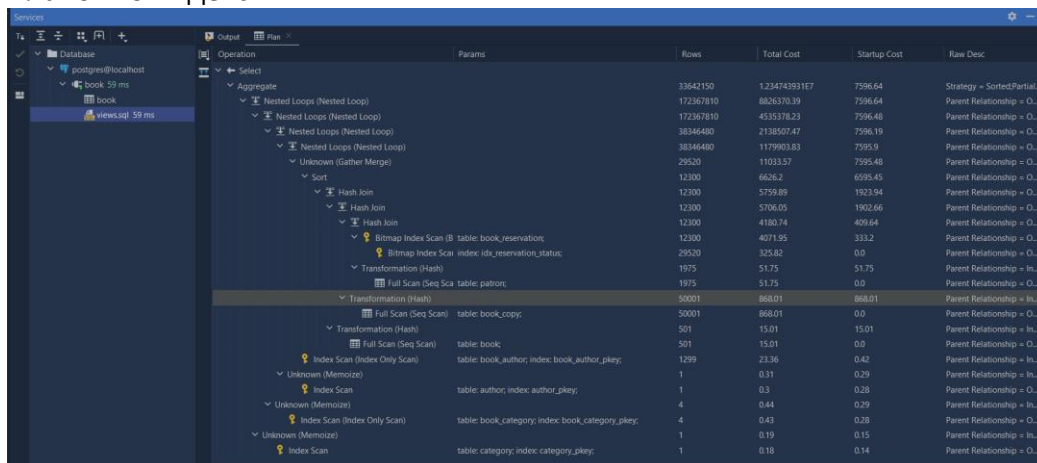


Operation	Params	Rows	Total Cost	Startup Cost	Raw Desc
Update		1	8.44	0.42	Parent Relationship = OuterPar...

Креираме индекс на табелата `book_reservation`. Бидејќи `reservation_status` се користи во `WHERE` предикатот на погледот, индексирањето на таа колона го забрзува погледот. Неговото време на извршување по индексот е 50тина милисекунди.

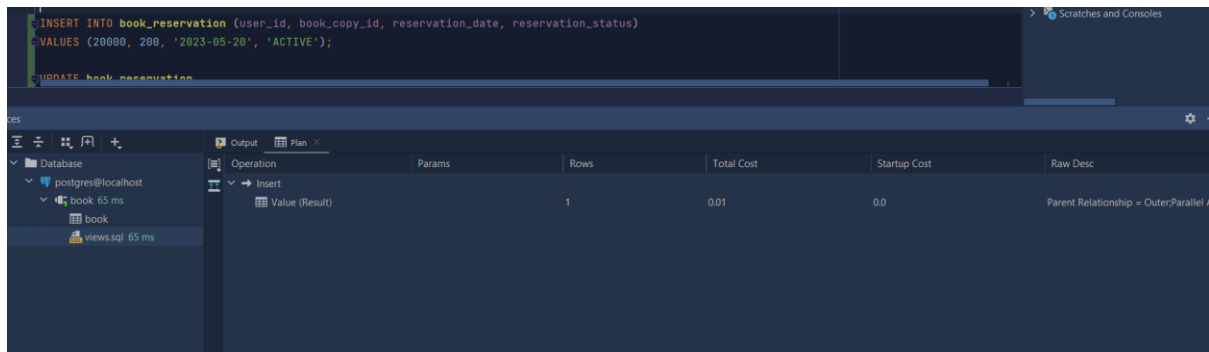
```
CREATE INDEX idx_reservation_status ON book_reservation (reservation_status);
```

## Планот по индекс:

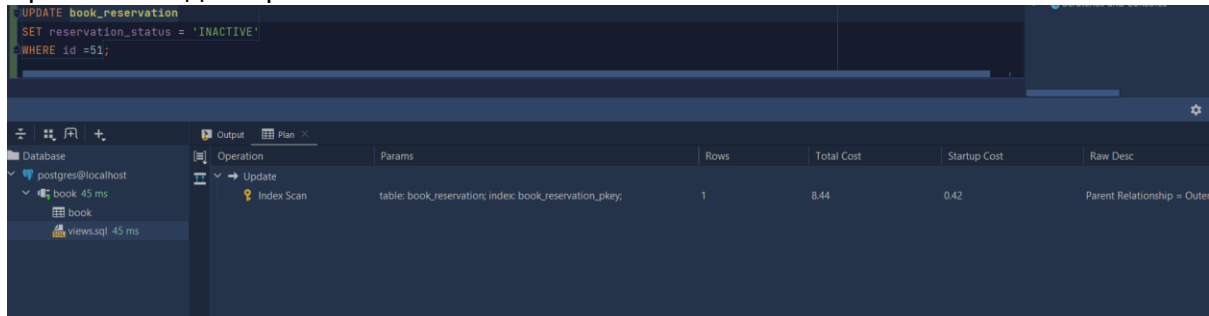


Operation	Params	Rows	Total Cost	Startup Cost	Raw Desc
Aggregate		33642150	1.234743931E7	7596.64	Strategy = Sorted/Partial...
Nested Loops (Nested Loop)		172367810	8826370.39	7596.64	Parent Relationship = O...
Nested Loops (Nested Loop)		172367810	4535378.23	7596.48	Parent Relationship = O...
Nested Loops (Nested Loop)		38346480	2138507.47	7596.19	Parent Relationship = O...
Nested Loops (Nested Loop)		38346480	1179903.83	7595.9	Parent Relationship = O...
Unknown (Gather Merge)		29520	11033.57	7595.48	Parent Relationship = O...
Sort		12300	6626.2	6595.45	Parent Relationship = O...
Hash Join		12300	5759.89	1923.94	Parent Relationship = O...
Hash Join		12300	5706.05	1902.66	Parent Relationship = O...
Hash Join		12300	4180.74	409.64	Parent Relationship = O...
Bitmap Index Scan (B...	table: book_reservation	12300	4071.95	333.2	Parent Relationship = O...
Bitmap Index Scan	index: idx_reservation_status	29520	325.82	0.0	Parent Relationship = In...
Transformation (Hash)		1975	51.75	51.75	Parent Relationship = In...
Full Scan (Seq Sca...	table: patron	1975	51.75	0.0	Parent Relationship = O...
Transformation (Hash)		50001	868.01	868.01	Parent Relationship = In...
Full Scan (Seq Scan)	table: book_copy	50001	868.01	0.0	Parent Relationship = O...
Transformation (Hash)		501	15.01	15.01	Parent Relationship = In...
Full Scan (Seq Scan)	table: book	501	15.01	0.0	Parent Relationship = O...
Index Scan (Index Only Scan)	table: book_author; index: book_author_pkey;	1299	23.36	0.42	Parent Relationship = In...
Index Scan	table: author; index: author_pkey;	1	0.31	0.29	Parent Relationship = In...
Unknown (Memoize)		1	0.3	0.28	Parent Relationship = O...
Index Scan (Index Only Scan)	table: book_category; index: book_category_pkey;	4	0.44	0.29	Parent Relationship = In...
Unknown (Memoize)		4	0.43	0.28	Parent Relationship = O...
Index Scan	table: category; index: category_pkey;	1	0.19	0.15	Parent Relationship = In...
Unknown (Memoize)		1	0.18	0.14	Parent Relationship = O...

## Insert по индексирање



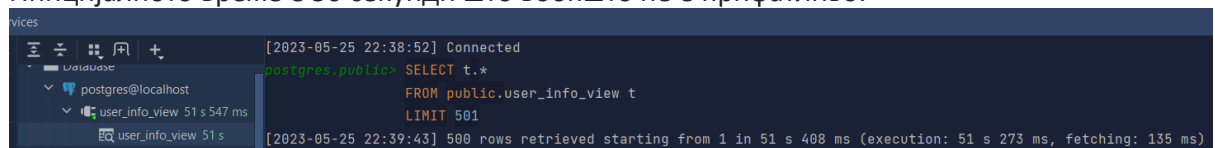
## Update по индексирање



## Info User view

1. Со кои филтри најчесто ќе се извршува погледот?  
Овој поглед најчесто би се извршувал по име и презиме на корисникот
2. Кои се случаите на употреба на погледот? Дали ни се важни перформансите, или станува збор за аналитички поглед?  
- Кога библиотекарот сака да ги провери податоците за корисникот, па перформансите се важни.
3. Кое е иницијалното време за извршување на погледот (со сите најчесто користени филтри)? Дали ова време е прифатливо за апликацијата?

Иницијалното време е 50 секунди што воопшто не е прифатливо.



4. Доколку не е, направете анализа на планот на извршување. Кои се најбавни операции? Како може да се подобрат? Пред додавање на индекси да се измери и времето на операции на insert i update на табелата во која ќе се додава индекс.

## User Info View before index

```
18
19 SELECT * from user_info_view
20 WHERE last_name = 'Lanternman';
21
```

Operation	Params	Rows	Actual Rows	Total Cost	Actual Total Time	Startup Cost	Actual Startup Time	Raw Desc
Select								Planning Time = 1.245; Tri...
Access (Subquery Scan)		1801	2	24197.41	210.337	23910.24	201.298	Parallel Aware = false; Asy...
Aggregate		1801	2	24179.4	210.334	23910.24	201.296	Strategy = Sorted; Partial ...
Unknown (Gather Merge)		1500	2	24109.63	210.305	23910.24	200.857	Parent Relationship = Out...
Aggregate		750	1	22936.47	191.709	22910.22	191.565	Strategy = Sorted; Partial ...
Sort		750	400	22912.1	191.484	22910.22	191.431	Parent Relationship = Out...
Merge Join		750	400	22874.41	190.993	22853.33	190.744	Parent Relationship = Out...
Sort		750	400	22676.85	190.405	22674.98	190.36	Parent Relationship = Out...
Hash Join		750	400	22630.16	190.253	20597.27	181.91	Parent Relationship = Out...
Transformation (Append)		3	1	100.41	0.427	0.0	0.264	Parent Relationship = Out...
Full Scan (Seq Scan) table: library_user;		1	0	52.6	0.24	0.0	0.144	Parent Relationship = Mem...
Full Scan (Seq Scan) table: patron;		1	0	46.52	0.259	0.0	0.245	Parent Relationship = Mem...
Full Scan (Seq Scan) table: librarian;		1	0	1.38	0.026	0.0	0.025	Parent Relationship = Mem...
Transformation (Hash)		494012	394801	12492.12	142.227	12492.12	142.226	Parent Relationship = Inne...
Full Scan (Seq Scan) table: payment;		494012	394801	12492.12	60.313	0.0	0.082	Parent Relationship = Out...
Sort		1975	1482	183.29	1.229	178.35	1.095	Parent Relationship = Inne...
Hash Join		1975	1976	70.24	0.781	13.15	0.03	Parent Relationship = Out...
Full Scan (Seq Scan) table: patron;		1975	1976	51.75	0.199	0.0	0.012	Parent Relationship = Out...
Transformation (Hash)		140	4	11.4	0.012	11.4	0.011	Parent Relationship = Inne...
Full Scan (Seq Scan) table: membership;		140	4	11.4	0.008	0.0	0.006	Parent Relationship = Out...

## Insert Payment before index

```
32 INSERT INTO payment(payment_date, membership_id, payment_method, user_id)
33 VALUES (now(), 1, 'CASH', 688);
34
```

Operation	Params	Rows	Actual Rows	Total Cost	Actual Total Time	Startup Cost	Actual Startup Time	Raw Desc
Insert								Planning Time = 0.022; Trigger...
Value (Result)		1	1	0.02	0.028	0.0	0.028	Parent Relationship = Outer; P...

## Update Payment before index

```
35 UPDATE payment
36 SET payment_method = 'CARD'
37 WHERE user_id = 688;
38
```

Operation	Params	Rows	Actual Rows	Total Cost	Actual Total Time	Startup Cost	Actual Startup Time	Raw Desc
Update								Planning Time = 0.063; Trig...
Bitmap Index Scan (Bitmap table: payment;		595	604	1835.1	0.192	9.04	0.076	Parent Relationship = Outer...
Bitmap Index Scan index: payment_user_id_index;		595	604	8.89	0.07	0.0	0.07	Parent Relationship = Outer...

5. По потреба, да се преуреди прашалникот и/или да се додадат индекси. Кој е планот на извршување по додадениот индекс? Кое е времето на извршување сега? Ако не е прифатливо се повторуваат чекор 4 и 5.

## Insert Payment after index

```
32 INSERT INTO payment(payment_date, membership_id, payment_method, user_id)
33 VALUES (now(), 1, 'CASH', 688);
34
```

Operation	Params	Rows	Actual Rows	Total Cost	Actual Total Time	Startup Cost	Actual Startup Time	Raw Desc
Insert								Planning Time = 0.024; Trigger...
Value (Result)		1	1	0.02	0.007	0.0	0.007	Parent Relationship = Outer; P...

## Update Payment after index

```
35 UPDATE payment
36 SET payment_method = 'CARD'
37 WHERE user_id = 688;
38
```

Operation	Params	Rows	Actual Rows	Total Cost	Actual Total Time	Startup Cost	Actual Startup Time	Raw Desc
Update								Planning Time = 0.497; Trigg...
Full Scan (Seq Scan) table: payment;		596	604	22369.94	110.184	0.0	109.947	Parent Relationship = Outer; ...

## Payment Index

```
CREATE INDEX payment_user_id_index on payment(user_id);
```

## User Info View after payment index



## Book title borrow count view

1. Со кои филтри најчесто ќе се извршува погледот?  
Најчесто ќе се извршува филтрирање по име на книга.
2. Кои се случаите на употреба на погледот? Дали ни се важни перформансите, или станува збор за аналитички поглед?  
Станува збор за аналитички поглед па ќе се употребува само кога ќе има потреба да се провери колку пати некој наслов бил позајмен.
3. Кое е иницијалното време за извршување на погледот (со сите најчесто користени филтри)? Дали ова време е прифатливо за апликацијата?  
Иницијалното време е околу 280 милисекунди што е прифатливо за апликацијата.

```
postgres.public> SELECT t.*
                  FROM public.book_title_borrow_count t
                  LIMIT 501
[2023-05-25 23:08:12] 10 rows retrieved starting from 1 in 279 ms (execution: 259 ms, fetching: 20 ms)
```

## Book copy borrow count view

1. Со кои филтри најчесто ќе се извршува погледот?  
Најчесто ќе се извршува филтрирање по примарен клуч на book copy.
2. Кои се случаите на употреба на погледот? Дали ни се важни перформансите, или станува збор за аналитички поглед?  
Станува збор за аналитички поглед па ќе се употребува само кога ќе има потреба да се провери колку пати некоја копија била позајмена.
3. Кое е иницијалното време за извршување на погледот (со сите најчесто користени филтри)? Дали ова време е прифатливо за апликацијата?  
Иницијалното време е околу 1 милисекунди што е прифатливо за апликацијата.

```
[2023-05-25 23:11:07] Connected
postgres.public> SELECT t.*
                  FROM public.copy_borrow_count t
                  LIMIT 501
[2023-05-25 23:11:07] 10 rows retrieved starting from 1 in 125 ms (execution: 104 ms, fetching: 21 ms)
```

## Event attendance view

1. Со кои филтри најчесто ќе се извршува погледот?  
Најчесто ќе се извршува филтрирање по име и датум на настанот.
2. Кои се случаите на употреба на погледот? Дали ни се важни перформансите, или станува збор за аналитички поглед?  
Перформансите се важни бидејќи со овој поглед ги прегледуваме одржаните настани и нивната посетеност.
3. Кое е иницијалното време за извршување на погледот (со сите најчесто користени филтри)? Дали ова време е прифатливо за апликацијата?  
Иницијалното време е околу 320 милисекунди што е прифатливо за апликацијата.

```
[2023-05-25 23:13:45] Connected
postgres.public> SELECT t.*
                  FROM public.event_attendance t
[2023-05-25 23:13:46] 3,999 rows retrieved starting from 1 in 322 ms (execution: 180 ms, fetching: 142 ms)
```

## Librarian users view

1. Со кои филтри најчесто ќе се извршува погледот?  
Најчесто ќе се извршува филтрирање по име, презиме и имејл на членот.
2. Кои се случаите на употреба на погледот? Дали ни се важни перформансите, или станува збор за аналитички поглед?  
Перформансите се важни бидејќи со овој поглед библиотекарите ги прегледуваат сите корисници.
3. Кое е иницијалното време за извршување на погледот (со сите најчесто користени филтри)? Дали ова време е прифатливо за апликацијата?  
Иницијалното време е околу 120 милисекунди што е прифатливо за апликацијата.

```
[2023-05-25 23:18:02] Connected
postgres@postgres:~$ SELECT t.*
FROM public.librarian_users t
[2023-05-25 23:18:02] 4,036 rows retrieved starting from 1 in 124 ms (execution: 16 ms, fetching: 108 ms)
```

## User profile view

1. Со кои филтри најчесто ќе се извршува погледот?  
Најчесто ќе се извршува филтрирање примарен клуч на корисникот.
2. Кои се случаите на употреба на погледот? Дали ни се важни перформансите, или станува збор за аналитички поглед?  
Перформансите се важни бидејќи со овој членовите си го прегледуваат својот профил.
3. Кое е иницијалното време за извршување на погледот (со сите најчесто користени филтри)? Дали ова време е прифатливо за апликацијата?  
Иницијалното време е околу 70 милисекунди што е прифатливо за апликацијата.

```
[2023-05-25 23:20:51] Connected
postgres@postgres:~$ SELECT t.*
FROM public.user_profile t
LIMIT 501
[2023-05-25 23:20:51] 500 rows retrieved starting from 1 in 67 ms (execution: 14 ms, fetching: 53 ms)
```

## Reading list view

4. Со кои филтри најчесто ќе се извршува погледот?  
Најчесто ќе се извршува филтрирање по број на картичка на корисникот.
5. Кои се случаите на употреба на погледот? Дали ни се важни перформансите, или станува збор за аналитички поглед?  
Перформансите се важни бидејќи со овој членовите си ја прегледуваат својата листа од книги за читање.
6. Кое е иницијалното време за извршување на погледот (со сите најчесто користени филтри)? Дали ова време е прифатливо за апликацијата?  
Иницијалното време е околу 80 милисекунди што е прифатливо за апликацијата.

```
postgres@postgres:~$ SELECT t.*
FROM public.view_reading_list t
LIMIT 501
[2023-05-25 23:23:22] 500 rows retrieved starting from 1 in 77 ms (execution: 14 ms, fetching: 63 ms)
```