# Министерство науки и высшего образования Российской Федерации ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

#### ОТЧЕТ

#### ПО ЛАБОРАТОРНОЙ РАБОТЕ №4

# «ЗАПРОСЫ НА ВЫБОРКУ И МОДИФИКАЦИЮ ДАННЫХ. ПРЕДСТАВЛЕНИЯ. РАБОТА С ИНДЕКСАМИ»

по дисциплине «Проектирование и реализация баз данных»

Обучающийся Христофоров Владислав Николаевич Факультет прикладной информатики Группа К3240 Направление подготовки 09.03.03 Прикладная информатика Образовательная программа Мобильные и сетевые технологии 2023 Преподаватель Говорова Марина Михайловна

# СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
СХЕМА БАЗЫ ДАННЫХ	6
1 ЗАПРОСЫ К БАЗЕ ДАННЫХ	7
2 ПРЕДСТАВЛЕНИЯ	14
3 ЗАПРОСЫ НА МОДИФИКАЦИЮ ДАННЫХ	18
3.1 INSERT	18
3.2 UPDATE	21
3.3 DELETE	24
4 СОЗДАНИЕ ИНДЕКСОВ	26
ЗАКЛЮЧЕНИЕ	33

#### **ВВЕДЕНИЕ**

**Цель работы:** овладеть практическими навыками создания представлений и запросов на выборку данных к базе данных PostgreSQL, использования подзапросов при модификации данных и индексов.

#### Практическое задание:

- 1. Создать запросы и представления на выборку данных к базе данных PostgreSQL (согласно индивидуальному заданию лабораторной работы №2, часть 2 и 3).
- 2. Составить 3 запроса на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов.
- 3. Изучить графическое представление запросов и просмотреть историю запросов.
- 4. Создать простой и составной индексы для двух произвольных запросов и сравнить время выполнения запросов без индексов и с индексами. Для получения плана запроса использовать команду EXPLAIN.

#### Индивидуальное задание: Вариант 3. БД «Библиотека»

Описание предметной области: Каждая книга может храниться в нескольких экземплярах. Для каждого экземпляра известно место его хранения (комната, стеллаж, полка). Читателю не может быть выдано более 3-х книг одновременно. Книги выдаются читателям на срок не более 10 дней. В случае просрочки читателю назначается денежный штраф.

Все издания, поступающие в библиотеку, ставятся на библиотечный учет, согласно существующим требованиям. Необходимо хранить информацию, кто из сотрудников поставил экземпляр на учет.

Книги принимаются к учету на основании первичных учетных документов (накладной от поставщика, акта о приеме документов). Если документы поступают на безвозмездной основе (в результате передачи обязательных экземпляров и т. п.), оформляется акт о приеме документов. Документы, поступающие от читателей взамен утерянных и признанные

равноценными утраченным, оформляются актом о приеме документов взамен утерянных.

Выбытие документов из библиотеки отражается в учете в связи с физической утратой либо утратой потребительских свойств (по причине ветхости, дефектности, устарелости по содержанию, непрофильности). Непрофильность профиля издания определяется на основании комплектования фонда или иного документа, утверждаемого руководителем библиотеки. При выбытии документов из библиотеки оформляется акт о списании исключенных объектов библиотечного фонда (далее – акт о которому прилагается список исключаемых К библиотечного фонда. В акте о списании отражаются сведения о количестве и общей стоимости исключаемых документов, а также причина списания и направление изданий после выбытия с учета. В прилагаемом к акту списке указываются:

- регистрационный номер и шифр хранения издания;
- краткое библиографическое описание;
- стоимость, зафиксированная в регистре индивидуального учета издания;
  - коэффициент переоценки, стоимость после переоценки;
  - общая стоимость исключаемых документов.

#### Задание 2. Создайте запросы:

- Вывести список читателей, имеющих на руках книги, переведенные с английского языка, изданные позднее 2000 года.
- Вывести список читателей, не вернувших в срок книги и имеющих на руках более десяти книг.
- Вывести список изданий, которые были наиболее популярными у читателей за предыдущий календарный год.
- Вывести список книг, которые находятся в библиотеке в единственном экземпляре.

- Подсчитать количество читателей, которые не обращались в библиотеку в течение года.
- Вывести список книг по программированию на С#, экземпляры которых отсутствуют в библиотеке, и которые должны быть возвращены не позднее, чем через 3 дня.

**Задание 3.** Создать представления для администрации библиотеки, содержащие:

- сведения о должниках;
- сведения о наиболее популярных книгах (все экземпляры находятся на руках у читателей).

#### СХЕМА БАЗЫ ДАННЫХ

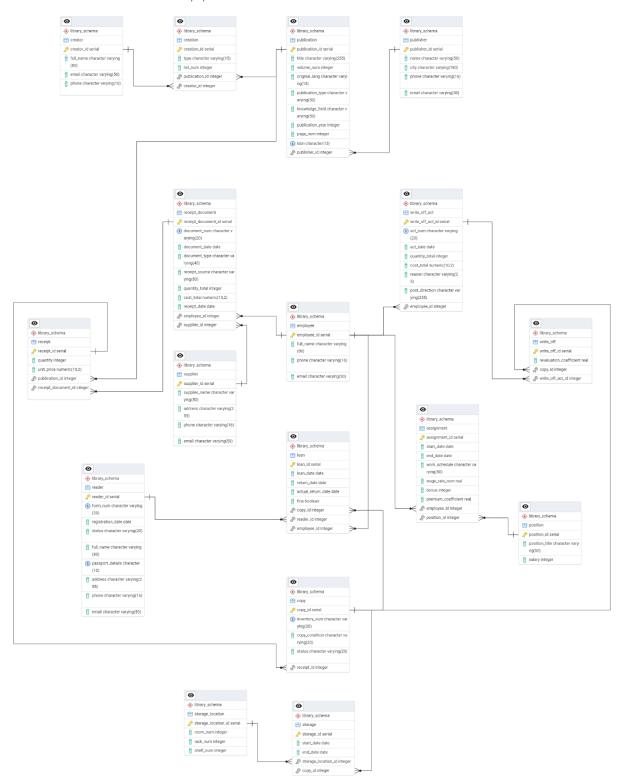


Рисунок 1 – Схема логической модели базы данных ERD

#### 1 ЗАПРОСЫ К БАЗЕ ДАННЫХ

#### Задание:

Вывести список читателей, имеющих на руках книги, переведенные с английского языка, изданные позднее 2000 года.

#### SQL-запрос:

```
SELECT r.reader_id, r.full_name

FROM library_schema.loan AS l

JOIN library_schema.reader AS r ON l.reader_id = r.reader_id

JOIN library_schema.copy AS c ON l.copy_id = c.copy_id

JOIN library_schema.receipt AS rec ON c.receipt_id = rec.receipt_id

JOIN library_schema.publication AS p ON rec.publication_id = p.publication_id

JOIN library_schema.creation AS cr ON p.publication_id = cr.publication_id

WHERE
```

```
p.original_lang = 'английский'

AND p.publication_year > 2000

AND cr.type = 'nepeвod'

AND l.actual_return_date IS NULL
```

	•					
Data (	Data Output Messages Notifications					
<b>=</b> +		<b>1 1 1 2 1 2 3 3 3 4 3 3 3 3 3 3 3 3 3 3</b>				
	reader_id [PK] integer	full_name character varying (80)				
1	3254	Филатова Герман Филипповна				
2	3383	Гурьев Степан Юльевич				
3	3260 Дементьев Федосий Герасимович					
4	3376	3376 Панов Наталья Фадеевич				
5	3272 Панфилов Макар Устинович					
6	3326 Носков Афиноген Филимонович					
Total	rows: 6 Qu	ery complete 00:00:00.171				

#### Задание:

Вывести список читателей, не вернувших в срок книги и имеющих на руках более десяти книг.

#### SQL-запрос:

```
WITH loan_stats AS (
      SELECT
            reader_id,
            COUNT(*) FILTER (WHERE actual_return_date IS NULL) AS active_loans,
            COUNT(*) FILTER (WHERE actual_return_date IS NULL AND return_date <
CURRENT_DATE) AS overdue_loans
      FROM library_schema.loan
      GROUP BY reader_id
SELECT
      r.reader_id,
      r.full_name,
      ls.active_loans,
      ls.overdue_loans
FROM loan_stats AS ls
      JOIN library_schema.reader AS r ON r.reader_id = ls.reader_id
WHERE ls.active_loans > 10 AND ls.overdue_loans > 0
```

Data Output Messages Notifications					
=+	<u> </u>	<b>■ ■ * SQL</b>			
	reader_id [PK] integer	full_name character varying (80)	active_loans bigint	overdue_loans bigint	
1	3190	Гусева Ярополк Эдгарович	12	12	
2	3379	Гусев Фирс Чеславович	14	14	
3	3234	Афанасьев Пимен Кирилловна	18	18	
4	3254	Филатова Герман Филипповна	14	14	
5	3260	3260 Дементьев Федосий Герасимович		14	
6	3272	Панфилов Макар Устинович	13	13	
7	3285	Сергеева Таисия Гертрудович	11	11	
8	3326	Носков Афиноген Филимонович	15	15	
9	3376	Панов Наталья Фадеевич	15	15	
10	3383	Гурьев Степан Юльевич	15	15	
Total	rows: 10 Qu	uery complete 00:00:00.133			

#### Задание

Вывести список изданий, которые были наиболее популярными у читателей за предыдущий календарный год.

#### SQL-запрос:

```
WITH yearly AS (
      SELECT
             p.publication_id,
             p.title,
             COUNT(*) AS loan_count
      FROM library_schema.loan AS l
             JOIN library_schema.copy AS c ON l.copy_id = c.copy_id
             JOIN library_schema.receipt AS rec ON c.receipt_id = rec.receipt_id
             JOIN library_schema.publication AS p ON rec.publication_id = p.publication_id
      WHERE
             l.loan_date >= DATE_TRUNC('year', CURRENT_DATE) - INTERVAL '1 year'
             AND l.loan_date < DATE_TRUNC('year', CURRENT_DATE)
      GROUP BY
             p.publication_id,
             p.title
),
ranked AS (
      SELECT
             publication_id,
             title,
             loan_count,
             RANK() OVER (ORDER BY loan_count DESC) AS rk
      FROM yearly
SELECT
      publication_id,
      title,
      loan_count
FROM ranked
WHERE rk = 1
```

Результат выполнения запроса

Data (	Output Messag	es Notifications	
<b>=</b> +		<b>1</b>	
	publication_id [PK] integer	title character varying (255)	loan_count bigint
1	2723	Python для профессионалов	20
2	2722	Углубленный SQL	20
3	2730	Angular: Полное руководство	20
4	2728	Node.js в действии	20
5	2731	Vue.js: Путь к мастерству	20
6	2729	React: Быстрый старт	20
7	2726	Ruby on Rails	20
8	2724	Современный Java	20
9	2725	РНР для веб-разработчиков	20
10	2727	HTML и CSS: Разработка и дизайн веб-приложений	20
Total	rows: 10 Que	ry complete 00:00:00.129	

#### Задание:

Вывести список книг, которые находятся в библиотеке в единственном экземпляре.

#### SQL-запрос:

```
SELECT p.publication_id, p.title

FROM library_schema.publication AS p

JOIN library_schema.receipt AS rec ON p.publication_id = rec.publication_id

JOIN library_schema.copy AS c ON rec.receipt_id = c.receipt_id

WHERE c.status <> 'cnucah'

GROUP BY p.publication_id, p.title

HAVING COUNT(*) = 1

;
```

Data	Data Output Messages Notifications			
=+		<u>'</u>	<b>1 1 1 2 1 2 3 3 3 3 3 3 3 3 3 3</b>	
	publicatio [PK] integ	A**	title character varying (255)	
1		2732	Хоббихорсинг для чайников	
2		2739	Основы С#	
3		2733	Путешествие по миру квадробер	
4		2720	Классическая проза	
5		2737	Практика С#	
6		2740	С# для профессионалов	
7		2738	Современный С#	
8		2703	Домашний что.	
9		2721	Современная проза	
10		2735	Тайны древнего хоббихорсинга	
11		2741	С# для начинающих	
12		2717	Современные рассказы	
13		2734	Как вырастить единорога	
14		2716	Английская классика	
15		2718	Мировая драма	
16		2575	Прежний песенка число.	
17		2736	В поисках идеального квадробера	
18		2719	Театр абсурда	
Total	rows: 18	Que	ery complete 00:00:00.122	

#### Задание:

Подсчитать количество читателей, которые не обращались в библиотеку в течение года.

#### SQL-запрос:

```
SELECT COUNT(*) AS inactive_readers

FROM library_schema.reader AS r

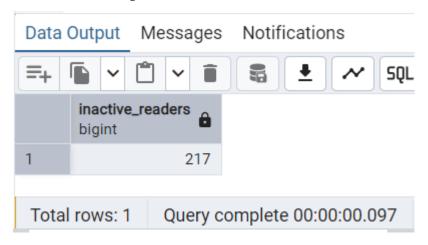
WHERE NOT EXISTS (

SELECT 1

FROM library_schema.loan AS l

WHERE l.reader_id = r.reader_id

AND l.loan_date >= CURRENT_DATE - INTERVAL '1 year'
)
```



#### Задание:

Вывести список книг по программированию на С#, экземпляры которых отсутствуют в библиотеке, и которые должны быть возвращены не позднее, чем через 3 дня.

#### SQL-запрос:

```
SELECT

p.publication_id,

p.title,
l.return_date

FROM library_schema.publication AS p

JOIN library_schema.receipt AS rec ON p.publication_id = rec.publication_id

JOIN library_schema.copy AS c ON rec.receipt_id = c.receipt_id

JOIN library_schema.loan AS l ON c.copy_id = l.copy_id

WHERE p.knowledge_field = 'программирование'

AND p.title LIKE '%C#%'

AND c.status = 'выдан'

AND l.actual_return_date IS NULL

AND l.return_date >= CURRENT_DATE - INTERVAL '3 days'
```

Data	Output M	essag	es Notifications	
=+		~	<b>1</b>	)L
	publication, integer	_id	title character varying (255)	return_date date
1		2737	Практика С#	2025-05-26
2		2741	С# для начинающих	2025-05-27
3		2738	Современный С#	2025-05-28
4		2739	Основы С#	2025-05-29
Total	rows: 4	Query	y complete 00:00:00.101	

#### 2 ПРЕДСТАВЛЕНИЯ

#### Задание:

Сведения о должниках.

#### SQL-запрос:

CREATE OR REPLACE VIEW library\_schema.view\_debtors AS SELECT

> r.reader\_id, r.full\_name,

COUNT(\*) AS overdue\_count

FROM library\_schema.loan AS l

JOIN library\_schema.reader AS r ON l.reader\_id = r.reader\_id

WHERE (l.actual\_return\_date > l.return\_date)

 $OR\ (l.actual\_return\_date\ IS\ NULL\ AND\ l.return\_date < CURRENT\_DATE)$ 

GROUP BY r.reader\_id, r.full\_name

#### Результат выполнения запроса

Data Output Messages Notifications

CREATE VIEW

Query returned successfully in 381 msec.

Total rows: Query complete 00:00:00.381

# Содержимое представления

 $SELECT*FROM\ library\_schema.view\_debtors;$ 

Data	Data Output Messages Notifications				
=+	<b>~ °</b>				
	reader_id integer	full_name character varying (80)	overdue_count bigint		
1	3243	Котова Мартьян Якубович	4		
2	3272	Панфилов Макар Устинович	13		
3	3255	Никифоров Ким Ермолаевич	4		
4	3190	Гусева Ярополк Эдгарович	12		
5	3285	Сергеева Таисия Гертрудович	11		
6	3234	Афанасьев Пимен Кирилловна	18		
7	3182	Сергеев Натан Зиновьевич 4			
8	3383	Гурьев Степан Юльевич 15			
9	3379	Гусев Фирс Чеславович	14		
10	3326	Носков Афиноген Филимонович	15		
11	3254	Филатова Герман Филипповна	14		
12	3260	Дементьев Федосий Герасимович	14		
13	3376	Панов Наталья Фадеевич	15		
14	3329	Александров Фаина Захарьевич	4		
Total	rows: 14	Query complete 00:00:00.124			

#### Задание:

Сведения о наиболее популярных книгах (все экземпляры находятся на руках у читателей).

#### SQL-запрос:

```
CREATE OR REPLACE VIEW library_schema.view_popular_books AS
SELECT
      p.publication_id,
      p.title,
      COUNT(l.loan_id) AS total_loans
FROM library_schema.publication AS p
      JOIN library_schema.receipt AS rec ON p.publication_id = rec.publication_id
      JOIN library_schema.copy AS c ON rec.receipt_id = c.receipt_id
      LEFT JOIN library_schema.loan AS l ON c.copy_id = l.copy_id
WHERE p.publication_id IN (
      SELECT p2.publication_id
      FROM library_schema.publication AS p2
             JOIN library_schema.receipt AS r2 ON p2.publication_id = r2.publication_id
             JOIN library_schema.copy AS c2 ON r2.receipt_id = c2.receipt_id
      GROUP BY p2.publication_id
      HAVING\ BOOL\ AND(c2.status = 'выдан')
GROUP BY p.publication_id, p.title
ORDER BY total_loans DESC
```

```
Data Output Messages Notifications

CREATE VIEW

Query returned successfully in 71 msec.

Total rows: Query complete 00:00:00.071
```

# Содержимое представления

 $SELECT*FROM\ library\_schema.view\_popular\_books$ 

=+ 🖺 🗸 🖹 🗸 🖹 🕹 💉 SQL					
	publication_id integer	title character varying (255)	total_loans bigint		
1	2730	Angular: Полное руководство	20		
2	2728	Node.js в действии	20		
3	2729	React: Быстрый старт	20		
4	2726	Ruby on Rails	20		
5	2724	Современный Java	20		
6	2731	Vue.js: Путь к мастерству	20		
7	2725	РНР для веб-разработчиков	20		
8	2722	Углубленный SQL	20		
9	2727	HTML и CSS: Разработка и дизайн веб-приложений	20		
10	2723	Python для профессионалов	20		
11	2719	Театр абсурда	1		
12	2739	Основы С#	1		
13	2720	Классическая проза	1		
14	2737	Практика С#	1		
15	2740	С# для профессионалов	1		
16	2738	Современный С#	1		
17	2721	Современная проза	1		
18	2741	С# для начинающих	1		
19	2717	Современные рассказы	1		
20	2716	Английская классика	1		
21	2718	Мировая драма	1		

## 3 ЗАПРОСЫ НА МОДИФИКАЦИЮ ДАННЫХ 3.1 INSERT

#### Задание:

Богданова Валентина Сергеевна хочет взять в библиотеке доступный и самый свежий по году выпуска экземпляр книги по истории, который поступил самым последним и не был ранее взят кем-либо. Ей выдает тот же сотрудник, который оформлял ей последнюю выдачу.

#### SQL-запрос на вставку:

```
INSERT INTO library_schema.loan (
          loan_date,
          return_date,
          actual_return_date,
          fine,
          copy_id,
          reader id,
          employee_id)
VALUES (
          CURRENT DATE,
          CURRENT_DATE + INTERVAL '10 days',
          NULL,
          FALSE,
             SELECT c.copy_id
             FROM library_schema.copy AS c
                   JOIN library_schema.receipt AS rec ON c.receipt_id = rec.receipt_id
                   JOIN
                              library_schema.receipt_document
                                                                  AS
                                                                           rd
                                                                                   ON
rec.receipt_document_id = rd.receipt_document_id
                   JOIN library_schema.publication AS p ON rec.publication_id =
p.publication_id
             WHERE p.knowledge field = 'история'
                   AND c.status = 'доступен'
                   AND NOT EXISTS (
                          SELECT 1
                          FROM library_schema.loan AS 12
```

```
)
ORDER BY p.publication_year DESC, rd.document_date DESC
LIMIT 1
),
(
SELECT r.reader_id
FROM library_schema.reader AS r
WHERE r.full_name = 'Богданова Валентина Сергеевна'
LIMIT 1
),
(
SELECT l.employee_id
FROM library_schema.loan AS l
JOIN library_schema.reader AS r USING(reader_id)
WHERE r.full_name = 'Богданова Валентина Сергеевна'
ORDER BY l.loan_date DESC
LIMIT 1
)
)
```

 $WHERE\ l2.copy\_id = c.copy\_id$ 

#### SQL-запрос для просмотра До и После:

#### **SELECT**

loan\_id,
loan\_date,
return\_date,
actual\_return\_date,
fine,
copy\_id,
reader\_id,

FROM library\_schema.loan AS l

employee\_id

JOIN library\_schema.reader AS r USING(reader\_id)

WHERE r.full name = 'Богданова Валентина Сергеевна'

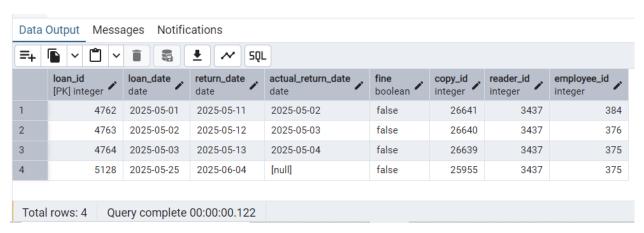
ORDER BY loan\_date

,

#### до:

Data	Data Output Messages Notifications							
<b>=</b> +	<b>□</b> ∨ □ ∨		<u>♣</u>	•				
	loan_id [PK] integer	loan_date date	return_date date	actual_return_date date	fine boolean	copy_id integer	reader_id integer	employee_id integer
1	4762	2025-05-01	2025-05-11	2025-05-02	false	26641	3437	384
2	4763	2025-05-02	2025-05-12	2025-05-03	false	26640	3437	376
3	4764	2025-05-03	2025-05-13	2025-05-04	false	26639	3437	375
Total	l rows: 3 Que	ery complete	00:00:00.174					

#### после:



### **3.2 UPDATE**

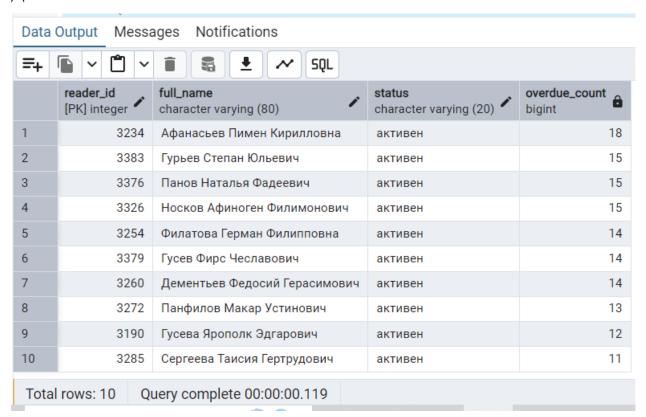
#### Задание:

Заблокировать всех читателей, у которых более 10 просрочек со штрафом.

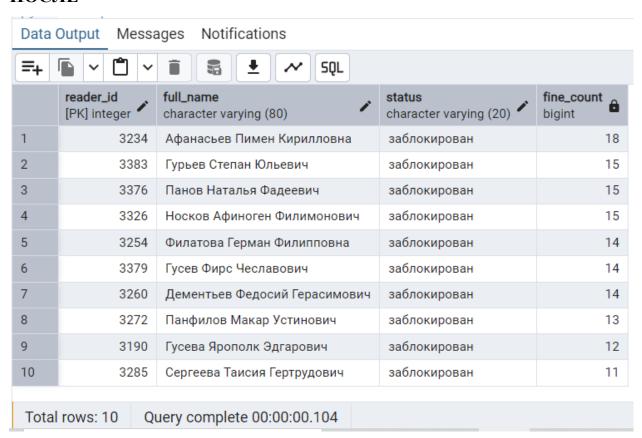
#### SQL-запрос на изменение:

#### SQL-запрос для просмотра до и после

```
SELECT
          r.reader_id,
          r.full_name,
          r.status,
             SELECT COUNT(*)
            FROM library_schema.loan AS l2
             WHERE\ l2.reader\_id = r.reader\_id
                   AND \ l2.fine = TRUE
          ) AS fine_count
FROM library_schema.reader AS r
WHERE
             SELECT COUNT(*)
             FROM library_schema.loan AS 12
             WHERE l2.reader_id = r.reader_id
                   AND\ l2.fine = TRUE
          ) > 10
          AND r.status <> 'заблокирован' -- AND r.status = 'заблокирован' - для ПОСЛЕ
ORDER BY fine_count DESC
```



#### ПОСЛЕ



#### 3.3 DELETE

#### Задание:

Удалить из таблицы выдач все записи со списанными экземплярами.

#### SQL-запрос на удаление

Data Output Messages Notifications

DELETE 12

Query returned successfully in 338 msec.

Total rows: Query complete 00:00:00.345

#### SQL-запрос для просмотра до и после

#### **SELECT**

l.loan\_id,

r.full\_name,

l.loan\_date,

l.return\_date,

c.status AS copy\_status

FROM library\_schema.loan AS l

JOIN library\_schema.reader AS r ON l.reader\_id = r.reader\_id

JOIN library\_schema.copy AS c ON l.copy\_id = c.copy\_id

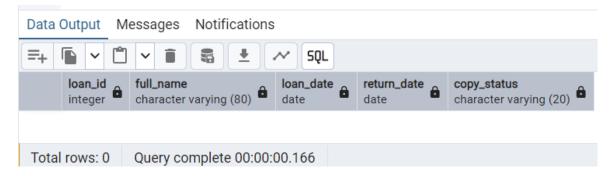
WHERE c.status = 'cnucah'

ORDER BY l.loan\_date

ДО

Data Output Messages Notifications					
=+		▼ SQL			
	loan_id integer	full_name character varying (80)	loan_date date	return_date date	copy_status character varying (20)
1	4555	Цветков Трифон Максимовна	2025-05-01	2025-05-11	списан
2	4564	Афанасьев Пимен Кирилловна	2025-05-01	2025-05-11	списан
3	4618	Красильникова Сократ Аркадьевна	2025-05-01	2025-05-11	списан
4	4629	Орлов Милица Ефимович	2025-05-01	2025-05-11	списан
5	4660	Калашников Лавр Эдгарович	2025-05-01	2025-05-11	списан
6	4673	Александров Фаина Захарьевич	2025-05-01	2025-05-11	списан
7	4676	Константинова Вероника Давидович	2025-05-01	2025-05-11	списан
8	4687	Козлова Дарья Архипович	2025-05-01	2025-05-11	списан
9	4694	Лаврентьева Тамара Антипович	2025-05-01	2025-05-11	списан
10	4720	Гурьев Степан Юльевич	2025-05-01	2025-05-11	списан
11	4728	Григорьева Софрон Максимовна	2025-05-01	2025-05-11	списан
12	4761	Лаврентьева Александра Фёдорович	2025-05-01	2025-05-11	списан
Tota	l rows: 12	Query complete 00:00:00.110			

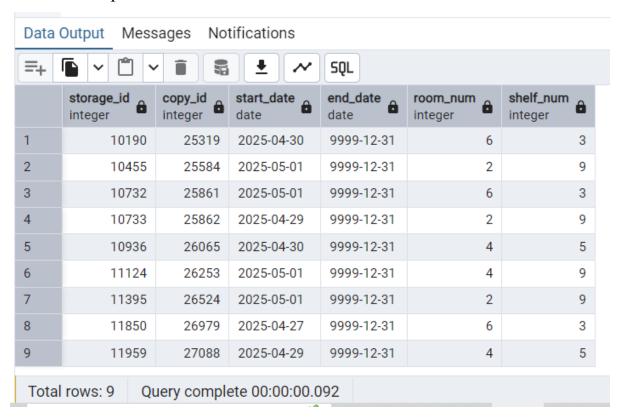
#### ПОСЛЕ



#### 4 СОЗДАНИЕ ИНДЕКСОВ

#### Задание:

Найти все экземпляры, хранящиеся в комнатах с номерами 1 по 10, на полках с номерами с 1 по 10 менее месяца.



#### SQL-запрос с командой EXPLAIN

```
EXPLAIN ANALYZE
```

#### **SELECT**

s.storage\_id,

s.copy\_id,

s.start\_date,

s.end\_date,

loc.room\_num,

loc.shelf\_num

FROM library\_schema.storage AS s

JOIN library\_schema.storage\_location AS loc

ON s.storage\_location\_id = loc.storage\_location\_id

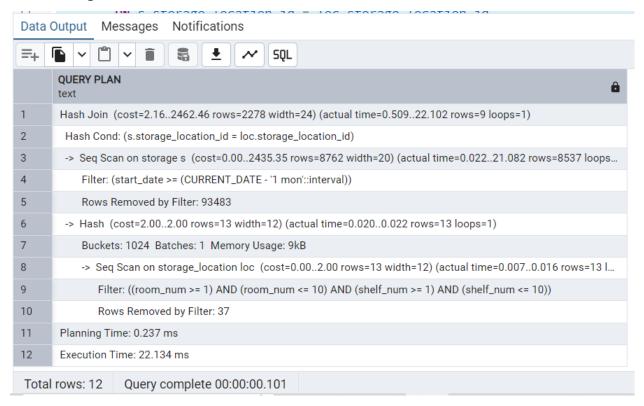
#### **WHERE**

loc.room\_num BETWEEN 1 AND 10

AND loc.shelf\_num BETWEEN 1 AND 10

AND s.start\_date >= CURRENT\_DATE - INTERVAL '1 month'

#### План запроса без индекса



#### Создание простого индекса и план запроса

CREATE INDEX idx\_storage\_start\_date

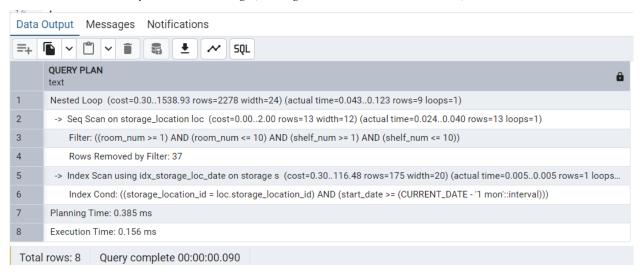
ON library\_schema.storage(start\_date);

Data	Output Messages Notifications
=+	
	QUERY PLAN text
1	Hash Join (cost=102.37930.65 rows=2278 width=24) (actual time=0.5253.511 rows=9 loops=1)
2	Hash Cond: (s.storage_location_id = loc.storage_location_id)
3	-> Bitmap Heap Scan on storage s (cost=100.20903.54 rows=8762 width=20) (actual time=0.4272.462 rows=8537 loops=1)
4	Recheck Cond: (start_date >= (CURRENT_DATE - '1 mon'::interval))
5	Heap Blocks: exact=650
6	-> Bitmap Index Scan on idx_storage_start_date (cost=0.0098.01 rows=8762 width=0) (actual time=0.3470.347 rows=8537 l
7	Index Cond: (start_date >= (CURRENT_DATE - '1 mon'::interval))
8	-> Hash (cost=2.002.00 rows=13 width=12) (actual time=0.0320.033 rows=13 loops=1)
9	Buckets: 1024 Batches: 1 Memory Usage: 9kB
10	-> Seq Scan on storage_location loc (cost=0.002.00 rows=13 width=12) (actual time=0.0180.027 rows=13 loops=1)
11	Filter: ((room_num >= 1) AND (room_num <= 10) AND (shelf_num >= 1) AND (shelf_num <= 10))
12	Rows Removed by Filter: 37
13	Planning Time: 0.325 ms
14	Execution Time: 3.551 ms
Tota	rows: 14 Query complete 00:00:00.089

#### Создание составного индекса и план запроса

CREATE INDEX idx\_storage\_loc\_date

ON library\_schema.storage(storage\_location\_id, start\_date);



#### Сравнение времени выполнения запроса

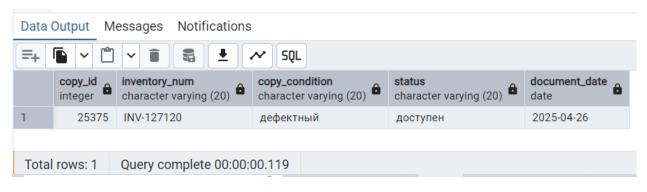
Без индекса – 22.134 мс

С простым индексом -3.551 мс

C составным индексом — 0.156 мс

#### Задание:

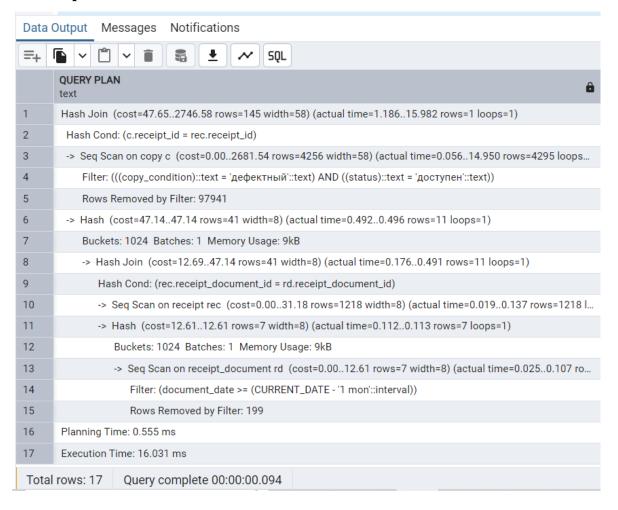
Найти все доступные дефектные экземпляры, поступившие в последний месяц.



#### SQL-запрос с командой EXPLAIN

```
EXPLAIN ANALYZE
SELECT
        c.copy_id,
        c.inventory_num,
        c.copy_condition,
        c.status,
        rd.document_date
FROM library_schema.copy AS c
        JOIN library_schema.receipt AS rec ON c.receipt_id = rec.receipt_id
        JOIN library_schema.receipt_document AS rd ON rec.receipt_document_id =
rd.receipt_document_id
WHERE c.copy_condition = 'дефектный'
        AND c.status = 'доступен'
        AND rd.document date >= CURRENT DATE - INTERVAL '1 month'
```

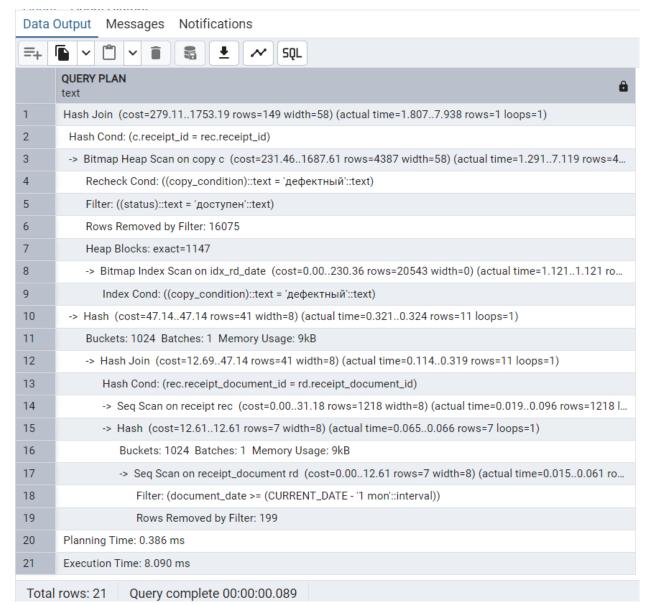
#### План запроса без индекса



#### Создание простого индекса и план запроса

CREATE INDEX idx\_copy\_cond

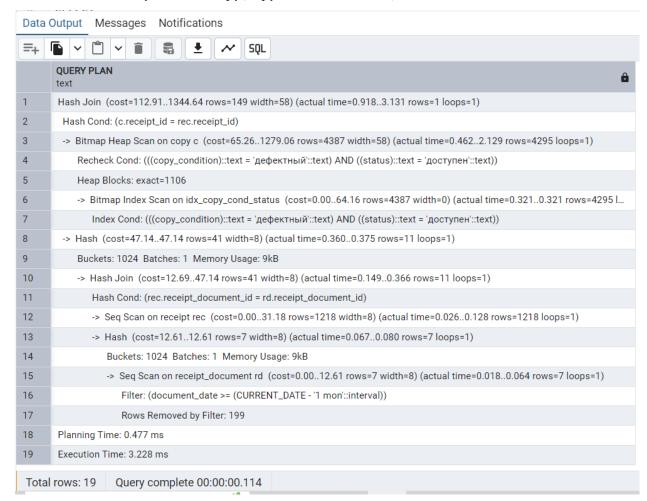
*ON library\_schema.copy(copy\_condition);* 



#### Создание составного индекса и план запроса

CREATE INDEX idx\_copy\_cond\_status

ON library\_schema.copy(copy\_condition, status);



#### Сравнение времени выполнения запроса

Без индекса – 16.031 мc

С простым индексом – 8.090 мс

С составным индексом – 3.228 мс

#### ЗАКЛЮЧЕНИЕ

В результате выполнения лабораторной работы написаны и выполнены запросы на выборку и модификацию данных к базе данных PostgreSQL. По индивидуальному заданию реализованы выборки и созданы представления, охватывающие различные аспекты работы библиотеки. Также составлены запросы на модификацию данных с использованием подзапросов. Для оценки производительности запросов созданы простые и составные индексы, проведено сравнение времени выполнения с помощью команды EXPLAIN, по результатам которого выявлено значительное сокращение времени при использовании индексов.