

Министерство науки и высшего образования Российской Федерации Федеральное  
государственное автономное образовательное учреждение высшего образования  
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО» Факультет  
инфокоммуникационных технологий

#### ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 4

по теме: Запросы на выборку и модификацию данных, представления и индексы в  
PostgreSQL

Специальность:

09.03.03 Мобильные и сетевые технологии

Проверил:

Говорова М.М. \_\_\_\_\_

Выполнил:

студент группы К3240 Ковалев В.М.

Санкт-Петербург 2022

## ЦЕЛЬ РАБОТЫ

Овладеть практическими навыками создания представлений и запросов на выборку данных к базе данных PostgreSQL, использования подзапросов при модификации данных и индексов.

**Оборудование:** компьютерный класс.

**Программное обеспечение:** СУБД PostgreSQL, pgadmin 4.

**Практическое задание:**

1. Создать запросы и представления на выборку данных к базе данных PostgreSQL (согласно индивидуальному заданию, 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.

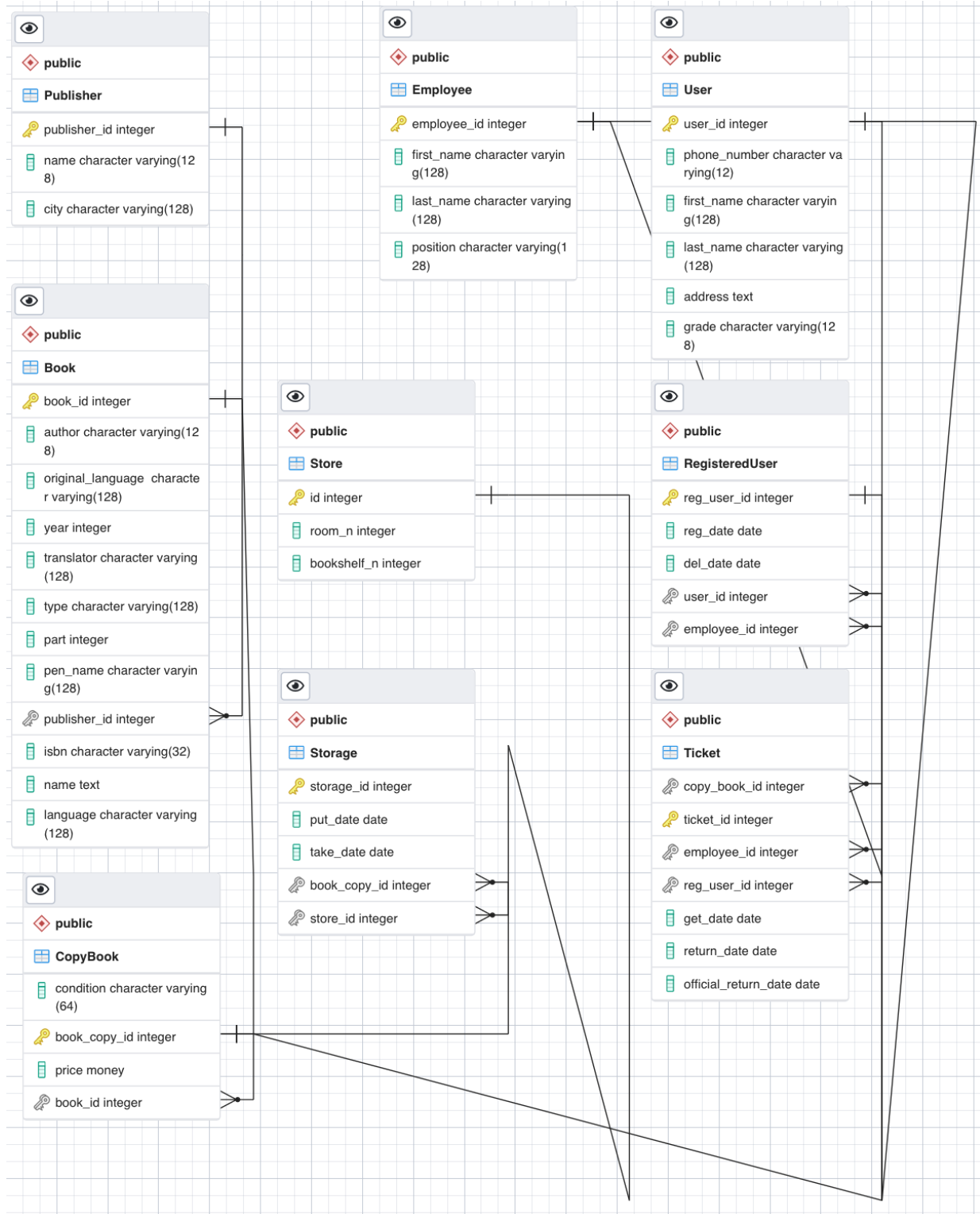
## Вариант 3

### 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. БД должна содержать следующий минимальный набор сведений: · Автор (фамилия и имя (инициалы) или псевдоним автора издания). · Название (заглавие) издания. · Номер тома (части, книги, выпуска). · Составитель (фамилия и имена (инициалы) каждого из составителей издания). · Язык, с которого выполнен перевод издания. · Вид издания (сборник, справочник, монография ...). · Область знания. · Переводчик (фамилия и инициалы переводчика). · Место издания (город). · Издательство (название издательства). · Год выпуска издания. · Библиотечный шифр (например, ББК 32.973). · Номер (инвентарный номер) экземпляра. · Номер комнаты (помещения для хранения экземпляров). · Номер стеллажа в комнате. · Номер полки на стеллаже. · Цена конкретного экземпляра. · Дата изъятия экземпляра с установленного места. · Номер читательского билета (формуляра). · Фамилия читателя. · Имя читателя. · Отчество читателя. · Адрес читателя. Телефон читателя.

Дополнить исходные данные информацией о читательском абонементе (выдаче книг).

Рисунок 1. ER-диаграмма



## Задание 1.

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

```
SELECT name, year, first_name, last_name, phone_number, return_date,
"original_language ", language FROM "Book" AS book RIGHT JOIN
(SELECT book_id, first_name, last_name, phone_number, return_date FROM "CopyBook"
as copy_book RIGHT JOIN
(SELECT copy_book_id, first_name, last_name, phone_number, return_date FROM
"Ticket" AS ticket LEFT JOIN
(SELECT reg_user_id, first_name, last_name, phone_number FROM "RegisteredUser" AS
rg
LEFT JOIN "User" AS u
ON rg.user_id = u.user_id) AS reg_users
ON reg_users.reg_user_id = ticket.reg_user_id) AS tickets
ON copy_book.book_copy_id = tickets.copy_book_id) AS copy_books
ON book.book_id = copy_books.book_id WHERE "original_language " = 'EN' AND
"language" != 'EN' AND "return_date" IS NULL AND "year" >2000;
```

Data Output

	name text	year integer	first_name character varying (128)	last_name character varying (128)	phone_number character varying (12)	return_date date	original_language character varying (128)	language character varying (128)
1	Tigers	2001	Vasya	Petrov	+79098674455	[null]	EN	RU
2	Tigers	2001	Anna	Vorobyeva	+79098674456	[null]	EN	RU
3	Tigers	2001	Vladimir	Pubin	+79098674460	[null]	EN	RU

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

```
SELECT first_name, last_name, "User".user_id, COUNT(copy_book_id) FROM "User",
"RegisteredUser", "Ticket"
WHERE ("User".user_id = "RegisteredUser".user_id AND "RegisteredUser".reg_user_id =
"Ticket".reg_user_id)
AND (return_date > official_return_date OR (return_date IS NULL AND
official_return_date < NOW()))
GROUP BY ("User".user_id) HAVING COUNT(copy_book_id) > 10;
```

	first_name character varying (128)	last_name character varying (128)	user_id [PK] integer	count bigint
1	Anastasia	Zaytseva	126	13

3. Найти количество читателей, не вернувших в срок книги и имеющих на руках более десяти книг

```
SELECT COUNT(*) FROM (SELECT first_name, last_name, "User".user_id,
COUNT(copy_book_id) FROM "User", "RegisteredUser", "Ticket"
WHERE ("User".user_id = "RegisteredUser".user_id AND "RegisteredUser".reg_user_id =
"Ticket".reg_user_id)
AND (return_date > official_return_date OR (return_date IS NULL AND
official_return_date < NOW()))
GROUP BY ("User".user_id) HAVING COUNT(copy_book_id) >10) as t1;
```

Data Output		
	count bigint	
1		1

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

```
SELECT name, count FROM (SELECT book_id, COUNT(book_id) FROM "CopyBook" GROUP BY
book_id) AS t1, "Book" WHERE count =1 AND t1.book_id="Book".book_id;
```

	name text	count bigint
1	History	1
2	Egypt's history	1

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

```
SELECT COUNT(*) FROM (
SELECT DISTINCT reg_user_id FROM "Ticket"
WHERE get_date < (NOW()-INTERVAL'1 year')
GROUP BY reg_user_id) AS T1;
```

	count bigint
1	31

6. Подсчитать количество читателей библиотеки по уровню образования

```
SELECT grade, COUNT(*) FROM "RegisteredUser" AS rg LEFT JOIN "User" AS u ON  
rg.user_id=u.user_id GROUP BY grade;
```

Data Output

	grade character varying (128)	count bigint
1	Bachelor	16
2	Master	9
3	Schoolchild	11
4	Student	11

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

```
SELECT name, return_date, official_return_date, copy_book_id FROM "Book",  
"CopyBook", "Ticket" WHERE "Book".book_id = "CopyBook".book_id  
AND "Ticket".copy_book_id = "CopyBook".book_copy_id AND name LIKE '%C#%' AND  
return_date IS NULL AND (official_return_date<NOW()+INTERVAL '3 day');
```

	name text	return_date date	official_return_date date	copy_book_id integer
1	C# Guide	[null]	2022-05-11	1
2	C# Guide	[null]	2022-05-11	2
3	C# Guide	[null]	2022-05-11	3

## Задание 2.

Создать представления для администрации библиотеки, содержащие:

### 1. сведения о должниках

```
CREATE VIEW CREDITS_USER_LIST AS
SELECT first_name, last_name, "User".user_id, COUNT(copy_book_id) FROM "User",
"RegisteredUser", "Ticket"
WHERE ("User".user_id = "RegisteredUser".user_id AND "RegisteredUser".reg_user_id =
"Ticket".reg_user_id)
AND (return_date IS NULL AND official_return_date < NOW())
GROUP BY ("User".user_id);
```

Запрос:

```
SELECT * FROM credits_user_list;
```




	first_name character varying (128)	last_name character varying (128)	user_id integer	count bigint
1	Fyodor	Magomedov	87	1
2	Petr	Popov	51	1
3	Valery	Zaytseva	41	1
4	Anastasia	Magomedov	43	1
5	Dinara	Volkov	157	1
6	Paver	Popov	63	1
7	Mikhail	Popov	10	1
8	Vladimir	Pubin	7	1
9	Fyodor	Petrov	132	1
10	Mikhail	Petrov	45	1
11	Matvey	Petrov	15	1
12	Nikolai	Lebedev	140	1

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

```
CREATE VIEW BOOK_TOP_LIST AS
SELECT name, part, COUNT(*) FROM "Storage", "Ticket", "CopyBook", "Book"
WHERE "Storage".book_copy_id="Ticket".copy_book_id
AND "Ticket".copy_book_id="CopyBook".book_copy_id
AND "Book".book_id="CopyBook".book_id
AND take_date>put_date
AND return_date IS NULL GROUP BY(name, part) ORDER BY count DESC;
```

Запрос:

```
SELECT * FROM book_top_list;
```

	 name text	 part integer	 count bigint
1	Who I Am?	2	5
2	Spider-Man	1	4
3	Monkeys	2	3
4	Fresh watermelons	1	3
5	Math	2	3
6	C# Guide	1	2
7	Economics	2	2
8	Along	2	2
9	Along	1	2
10	Cockroaches. Bees. Ladybugs.	1	1
11	Math	1	1
12	Economics	1	1
13	Spider-Man	2	1
14	Fresh watermelons	2	1
15	Venom	1	1
16	Cockroaches. Bees. Ladybugs.	2	1



### Задание 3.

1. Запрос на модификацию данных с INSERT (добавление новой книги):

```
INSERT INTO "Book" (book_id, "original_language ", year, type, part, publisher_id, isbn, name, language, author)
SELECT 25, 'EN', 2010, 'Guide', 1, 1, 6434234, 'SQL for kids', 'EN', 'Genry';
```

Проверка:

```
SELECT * FROM "Book" WHERE name = 'SQL for kids';
```

Data Output											
book_id 	author 	original_language 	year 	transl 	type 	part 	publ 	publis 	isbn 	name 	language 
[PK] integer	character varying (128)	character varying (128)	integer	char	character varying (128)	integer	char	integer	character varying (32)	text	character varying (128)
25	Genry	EN	2010	[null]	Guide	1	[null]	1	6434234	SQL for kids	EN

2. Запрос на модификацию данных с UPDATE (продлить всем читателям, у которых на руках книга, дату обязательной ее сдачи на неделю):

```
UPDATE "Ticket"
SET official_return_date = official_return_date + INTERVAL '7 DAYS'
WHERE return_date IS NULL;
```

Проверка (изначально дата обязательной сдачи была через два месяца после получения книги):

59	16	68	2	56	2022-01-10	[null]	2022-03-17
60	17	69	2	57	2022-01-15	[null]	2022-03-22
61	149	70	2	58	2022-01-07	[null]	2022-03-14
62	150	71	2	60	2022-01-15	[null]	2022-03-22
63	24	72	2	61	2022-01-14	[null]	2022-03-21
64	25	73	2	62	2022-01-07	[null]	2022-03-14
65	26	74	2	63	2022-01-08	[null]	2022-03-15
66	155	75	2	64	2022-01-12	[null]	2022-03-19

3. Запрос на модификацию данных с DELETE (удалить неиспользуемые пустые книжные полки):




```
DELETE FROM "Store" WHERE id IN (SELECT id FROM "Storage" AS storage
RIGHT JOIN "Store" AS store
ON storage.store_id = store.id
WHERE storage_id IS NULL);
```

Проверка:

Полки, которыми никогда не пользовались:

```
SELECT storage_id, id AS store_id FROM "Storage" AS storage
RIGHT JOIN "Store" AS store
ON storage.store_id = store.id
WHERE storage_id IS NULL;
```

### Data Output

	 storage_id 	store_id 
	integer	integer
1	[null]	15
2	[null]	16
3	[null]	14

Теперь их нет:

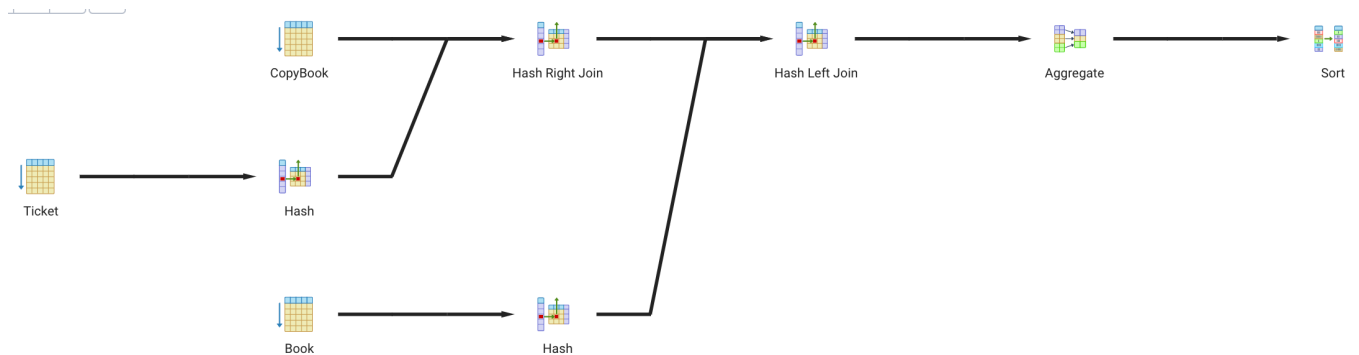
```
SELECT storage_id, id AS store_id FROM "Storage" AS storage
RIGHT JOIN "Store" AS store
ON storage.store_id = store.id
WHERE storage_id IS NULL;
```

## Data Output












	<b>storage_id</b> integer	<b>store_id</b> integer	
--	------------------------------	----------------------------	--

#### Задание 4.

Графическое представление запроса `SELECT * FROM book_top_list;`



## История запросов:

Query History	
Show queries generated internally by pgAdmin? <input type="button" value="No"/>	
Today - 20/04/2022	
	SELECT * FROM book_top_list; 17:14:01
	EXPLAIN SELECT * FROM book_top_list; 17:12:45
	SELECT * FROM book_top_list; 17:09:06
	EXPLAIN SELECT * FROM book_top_list; 17:08:56
	EXPLAIN SELECT * FROM book_top_list; 17:06:49
	EXPLAIN SELECT * FROM book_top_list; 17:06:42
	EXPLAIN SELECT * FROM book_top_list; 17:04:22
	SELECT storage_id, id AS store_id FROM "Storage" AS storage R... 17:00:08
	SELECT storage_id, id AS store_id FROM "Storage" AS storage R... 16:57:26
	SELECT storage_id, id FROM "Storage" AS storage RIGHT JOIN "S... 16:56:58
	SELECT id FROM "Storage" AS storage RIGHT JOIN "Store" AS sto... 16:56:44

## Задание 5.

До создания простого индекса:

```
EXPLAIN SELECT * FROM "Ticket" WHERE return_date > '2015-10-19';
```

Successfully run. Total query runtime: 53 msec.

26 rows affected.

QUERY PLAN	
	text 
1	Seq Scan on "Ticket" (cost=0.00..1.98 rows=27 width=28)
2	[...] Filter: (return_date > '2015-10-19'::date)

```
CREATE INDEX date_of_return ON "Ticket"(return_date);
```

После:

```
EXPLAIN SELECT * FROM "Ticket" WHERE return_date > '2015-10-19';
```

Successfully run. Total query runtime: 70 msec.

2 rows affected.

QUERY PLAN		
	text	
1	Seq Scan on "Ticket" (cost=0.00..1.98 rows=27 width=28)	
2	[...] Filter: (return_date > '2015-10-19'::date)	

До создания составного индекса:

```
EXPLAIN SELECT * FROM "User" WHERE first_name LIKE 'A%' AND last_name LIKE '%a';
```

Successfully run. Total query runtime: 45 msec. 2 rows affected.

QUERY PLAN		
	text	
1	Seq Scan on "User" (cost=0.00..5.97 rows=2 width=56)	
2	[...] Filter: (((first_name)::text ~~ 'A% '::text) AND ((last_name)::text ~~ '%a '::text))	

```
CREATE INDEX full_name ON "User"(first_name, last_name);
```

После:

```
EXPLAIN SELECT * FROM "User" WHERE first_name LIKE 'A%' AND last_name LIKE '%a';
```

Successfully run. Total query runtime: 48 msec. 2 rows affected.

QUERY PLAN		
	text	
1	Seq Scan on "User" (cost=0.00..5.97 rows=2 width=56)	
2	[...] Filter: (((first_name)::text ~~ 'A% '::text) AND ((last_name)::text ~~ '%a '::text))	

Удаление индексов:

Column	Type	Table "public.User"			Default
		Collation	Nullable		
user_id	integer		not null		generated by default as identity
phone_number	character varying(12)		not null		
first_name	character varying(128)		not null		
last_name	character varying(128)		not null		
address	text		not null		
grade	character varying(128)		not null		

Indexes:

"User\_pkey" PRIMARY KEY, btree (user\_id)

"full\_name" btree (first\_name, last\_name)

Referenced by:

TABLE ""RegisteredUser"" CONSTRAINT "User\_id" FOREIGN KEY (user\_id) REFERENCES "User"(user\_id)

```
DROP INDEX full_name;
```

```

-----, , , -----
      Column      |      Type      | Table "public.User"
                  |                 | | Collation | Nullable |
                  |                 | |-----|-----|
                  |                 | |         |         |
user_id           | integer         | |         | not null | generated by default as identity
phone_number      | character varying(12) | |         | not null |
first_name        | character varying(128) | |         | not null |
last_name         | character varying(128) | |         | not null |
address           | text            | |         | not null |
grade             | character varying(128) | |         | not null |
Indexes:
    "User_pkey" PRIMARY KEY, btree (user_id)
Referenced by:
    TABLE "RegisteredUser" CONSTRAINT "User_id" FOREIGN KEY (user_id) REFERENCES "User"(user_id)

```

## Вывод:

В ходе выполнения лабораторной работы мною были выполнены запросы из своей базы данных. Созданы представления на администрации библиотеки и сделаны запросы на модификацию данных. Также созданы индексы. После создания индекса время выполнения запроса увеличилось.