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

ОТЧЕТ
О ЛАБОРАТОРНОЙ РАБОТЕ № 4

по теме: Запросы на выборку данных к БД PostgreSQL. по дисциплине: Проектирование и реализация баз данных

Специальность:
09.03.03 Мобильные и сетевые технологии

Проверил:
Говорова М.М. _____
Дата: «__»_____2021 г.
Оценка_____

Выполнил:
студентка группы К3240

Вали Насибулла

Санкт-Петербург 2022 г

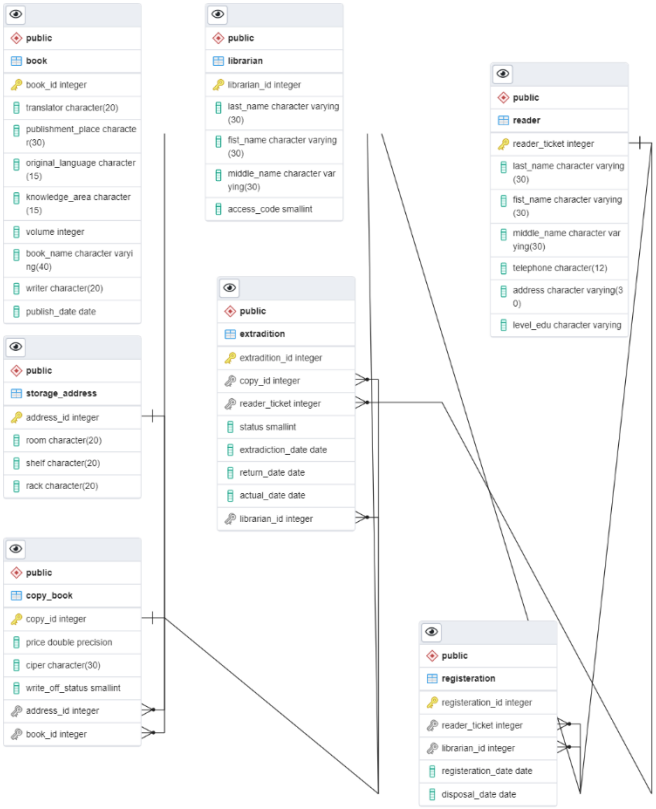
ЦЕЛЬ РАБОТЫ

Овладеть практическими навыками создания представлений и запросовна выборку данных к базе данных PostgreSQL и использования подзапросов при модификации данных.

ПРАКТИЧЕСКОЕ ЗАДАНИЕ

1. Создать запросы и представления на выборку данных к базе данныхPostgreSQL (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 посмотреть 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.

СХЕМА БАЗЫ ДАННЫХ



Задание 2. Запросы

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

SELECT reader.fist_name, reader.middle_name,reader.last_name,reader.reader_ticket,book.original_language,book.publish_date, extradition.copy_id From reader, book,extradition,copy_book Where extradition.reader_ticket= reader.reader_ticket And extradition.copy_id = copy_book.copy_id And book.book_id= copy_book.book_id Andbook.publish_date >='2001-01-01' AND book.original_language='English';

Query Editor Query History

```
SELECT reader.fist_name, reader.middle_name,reader.last_name,reader.reader_ticket,book.original_language,
reader, book,extradition,copy_book
Where extradition.reader_ticket= reader.reader_ticket And
extradition.copy_id = copy_book.copy_id And
book.book_id= copy_book.book_id
And
book.publish_date >'2000-01-01' AND  book.original_language='English';
```

Data Output Explain Messages Notifications

	fist_name character varying (30)	middle_name character varying (30)	last_name character varying (30)	reader_ticket integer	original_language character (15)	publish_date date	copy_id integer
1	Potar	Crazy	Harry	2	English	2008-11-11	1
2	John	freaken	Kerry	1	English	2010-11-11	2

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

SELECT t1.reader_ticket, COUNT(*)

From extradition AS t1

LEFT JOIN "reader" AS t2 ON t1.reader_ticket = t2.reader_ticket

Where ("return_date" < "actual_date" OR (current_date > "return_date" And "actual_date" is null))

Group BY t1.reader_ticket

Having COUNT(*) >2 ;

Query Editor | Query History

```
1  SELECT t1.reader_ticket, COUNT(*)
2  From extradition AS t1
3  LEFT JOIN "reader" AS t2 ON t1.reader_ticket = t2.reader_ticket
4  Where
5  ("return_date" < "actual_date" OR (current_date > "return_date"  And "actual_date" is null ))
6
7
8  Group BY t1.reader_ticket
9  Having COUNT(*) >2 ;
10
11
```

Data Output | Explain | Messages | Notifications

	reader_ticket integer	count bigint
1	2	3

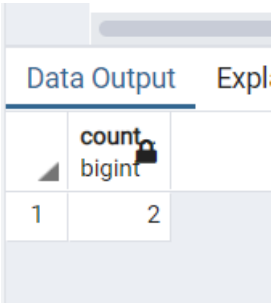
3: **Найти количество читателей, не вернувших в срок книги и имеющих на руках более десяти книг.**

SELECT COUNT(reader_ticket) FROM (SELECT t1.reader_ticket, COUNT(t1.return_date) FROM

"extradition" AS t1

LEFT JOIN "reader" AS t2 ON t1.reader_ticket = t2.reader_ticket

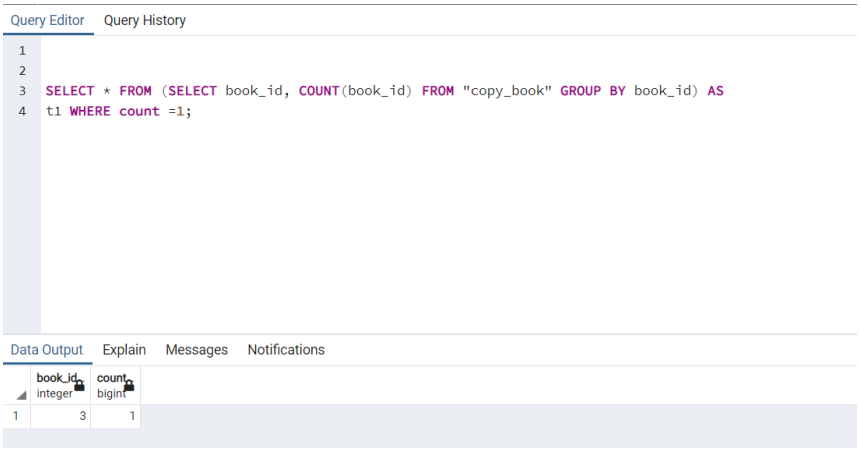
WHERE "return_date"<"actual_date" GROUP BY t1.reader_ticket) AS t;



4: **вывести список книг, которые находятся в библиотеке в единственном экземпляре.**

SELECT * FROM (SELECT book_id, COUNT(book_id) FROM "copy_book" GROUP BY book_id) AS

t1 WHERE count =1;



6 **Подсчитать количество читателей библиотеки по уровню образования.**

SELECT COUNT(level_edu) FROM reader WHERE reader.level_edu='Secondry'

UNION SELECT COUNT(level_edu) FROM reader

WHERE reader.level_edu='Hight'

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

```
SELECT COUNT(*) FROM (
 SELECT DISTINCT reader_ticket FROM "extradition"
 WHERE extradiction_date < (NOW()-INTERVAL '1 year')
 GROUP BY reader_Ticket) AS T1;
```

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 book.book_id, book.book_name, book.original_language,
book.publish_date,extradition.extradiction_date, extradition.return_date
FROM book, extradition,copy_book
Where extradition.copy_id= copy_book.copy_id And
copy_book.book_id = book.book_id And
extradition.status = 1 AND
extradition.return_date < NOW()-3
;
```

Query Editor | Query History

```
SELECT book.book_id, book.book_name, book.original_language,
book.publish_date,extradition.extradiction_date, extradition.return_date
FROM book, extradition,copy_book
Where extradition.copy_id= copy_book.copy_id And
copy_book.book_id = book.book_id And
extradition.status = 1 AND
extradition.return_date < (NOW()-INTERVAL '1 month')
;
```

Data Output | Explain | Messages | Notifications

	book_id integer	book_name character varying (40)	original_language character (15)	publish_date date	extradiction_date date	return_date date
1	1	SkyBook	English	2008-11-11	2022-01-22	2022-03-22
2	2	EarthBook	English	2010-11-11	2021-02-22	2021-03-22
3	2	EarthBook	English	2010-11-11	2022-01-20	2022-03-20
4	2	EarthBook	English	2010-11-11	2022-01-20	2022-03-20
5	3	SunBook	Italian	2020-11-11	2022-01-20	2022-03-20

Задание 3. Представления

Создать представления для администрации библиотеки, содержащие:

- сведения о должниках;

View

```
CREATE VIEW readers_view1 AS

SELECT Distinct reader.reader_ticket , reader.fist_name, reader.last_name

FROM reader,extradition

WHERE

reader.reader_ticket = extradition.reader_ticket

AND extradition.return_date < extradition.actual_date;
```

Query Editor | Query History

```
1
2  SELECT Distinct reader.reader_ticket , reader.fist_name, reader.last_name
3  FROM reader,extradition
4  WHERE
5  reader.reader_ticket = extradition.reader_ticket
6  AND extradition.return_date < extradition.actual_date;
7
8
9
```

Data Output | Explain | Messages | Notifications

	reader_ticket [PK] integer	fist_name character varying (30)	last_name character varying (30)
1	1	John	Kerry
2	2	Potar	Harry

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

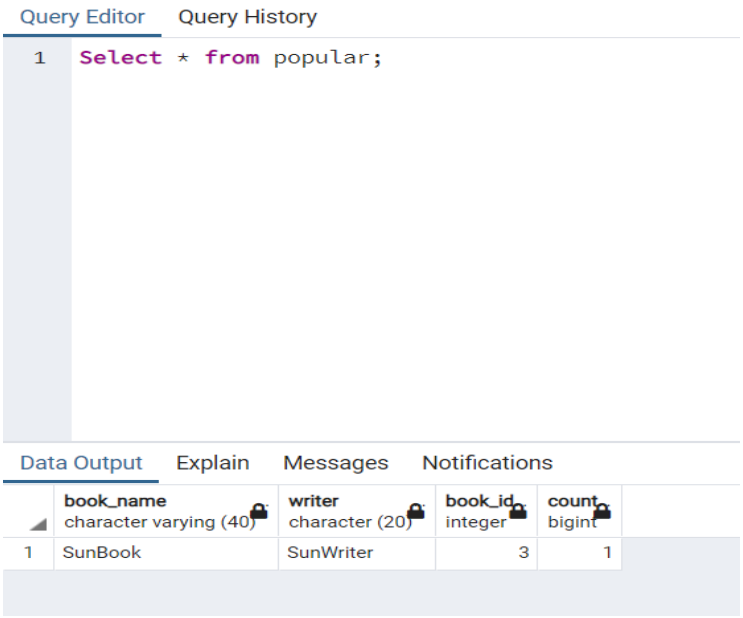
CREATE VIEW popular AS

SELECT t4.book_name, t4.writer, t3.book_id, COUNT(t3.book_id) FROM "book" AS t4 RIGHT JOIN (SELECT * FROM "copy_book" AS t2 RIGHT JOIN (

SELECT * FROM "extradition" WHERE return_date IS NULL AND extradition.status=1) AS t1

ON t1.copy_id=t2.copy_id) AS t3

ON t3.book_id=t4.book_id GROUP BY (t3.book_id, t4.writer, t4.book_name) ORDER BY count DESC;



Задание 4. Модификация данных

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

INSERT INTO public.book(

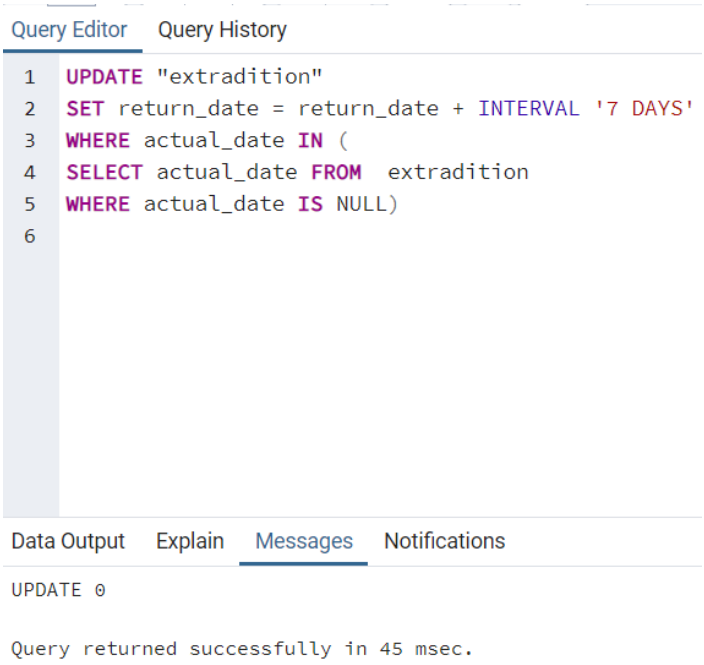
book_id, translator, publishment_place, original_language, knowledge_area, volume, book_name, writer, publish_date)

VALUES (322, 'translator322', 'address322', 'English', 'Presentation',

(SELECT volume FROM book WHERE book.book_name ='SunBook'), 'Moonbook', 'writer322', '1955-09-09');

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

UPDATE "extradition" SET return_date = return_date + INTERVAL '7 DAYS' WHERE actual_date IN (SELECT actual_date FROM extradition WHERE actual_date IS NULL)



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

Query Editor | Query History

```
1  DELETE FROM "storage_address" WHERE address_id IN
2  (SELECT storage_address.address_id FROM "storage_address" AS storage
3  RIGHT JOIN "storage_address" AS store
4  ON storage.address_id = storage_address.address_id
5  WHERE storage_address.address_id IS NULL);
```

Data Output | Explain | Messages | Notifications

```
DELETE 0

Query returned successfully in 76 msec.
```

Задание . Графическое представление запроса SELECT * FROM popular:

Query Editor | Query History

```
1  Explain Select * from popular;
```

Data Output | Explain | Messages | Notifications

	QUERY PLAN text
1	Sort (cost=35.99..35.99 rows=1 width=194)
2	[...] Sort Key: (count(t2.book_id)) DESC
3	[...] -> GroupAggregate (cost=35.95..35.98 rows=1 width=194)
4	[...] Group Key: t2.book_id, t4.writer, t4.book_name
5	[...] -> Sort (cost=35.95..35.96 rows=1 width=186)
6	[...] Sort Key: t2.book_id, t4.writer, t4.book_name
7	[...] -> Nested Loop Left Join (cost=0.29..35.94 rows=1 width=186)
8	[...] -> Seq Scan on extradition (cost=0.00..27.50 rows=1 width=4)
9	[...] Filter: ((return_date IS NULL) AND (status = 1))
10	[...] -> Nested Loop Left Join (cost=0.29..8.43 rows=1 width=190)

Задание 5. До создания простого индекса:

Query Editor | Query History

```
1  EXPLAIN SELECT * FROM "extradition" WHERE actual_date > '2015-10-19';
```

Data Output | Explain | Messages | Notifications

	QUERY PLAN text
1	Seq Scan on extradition (cost=0.00..27.50 rows=467 width=30)
2	[...] Filter: (actual_date > '2015-10-19'::date)

Query Editor | Query History

```
1  CREATE INDEX date_of_return ON "extradition"(actual_date);
```

Data Output | Explain | Messages | Notifications

```
CREATE INDEX

Query returned successfully in 40 msec.
```

После:

Query Editor Query History

```
1  EXPLAIN SELECT * FROM "extradition" WHERE actual_date > '2015-10-19';
```

Data Output Explain Messages Notifications

	QUERY PLAN text
1	Seq Scan on extradition (cost=0.00..1.10 rows=3 width=30)
2	[...] Filter: (actual_date > '2015-10-19'::date)

До создания составного индекса:

Query Editor Query History

```
1  EXPLAIN SELECT * FROM "reader" WHERE fist_name LIKE 'A%' AND last_name LIKE '%a';
2
```

Data Output Explain Messages Notifications

	QUERY PLAN text
1	Seq Scan on reader (cost=0.00..1.04 rows=1 width=400)
2	[...] Filter: (((fist_name)::text ~~ 'A%'::text) AND ((last_name)::text ~~ '%a'::text))

Query Editor | Query History

```
1  CREATE INDEX full_name ON "reader"(fist_name, last_name);
2
```

Data Output | Explain | Messages | Notifications

```
CREATE INDEX

Query returned successfully in 53 msec.
```

После:

Query Editor | Query History

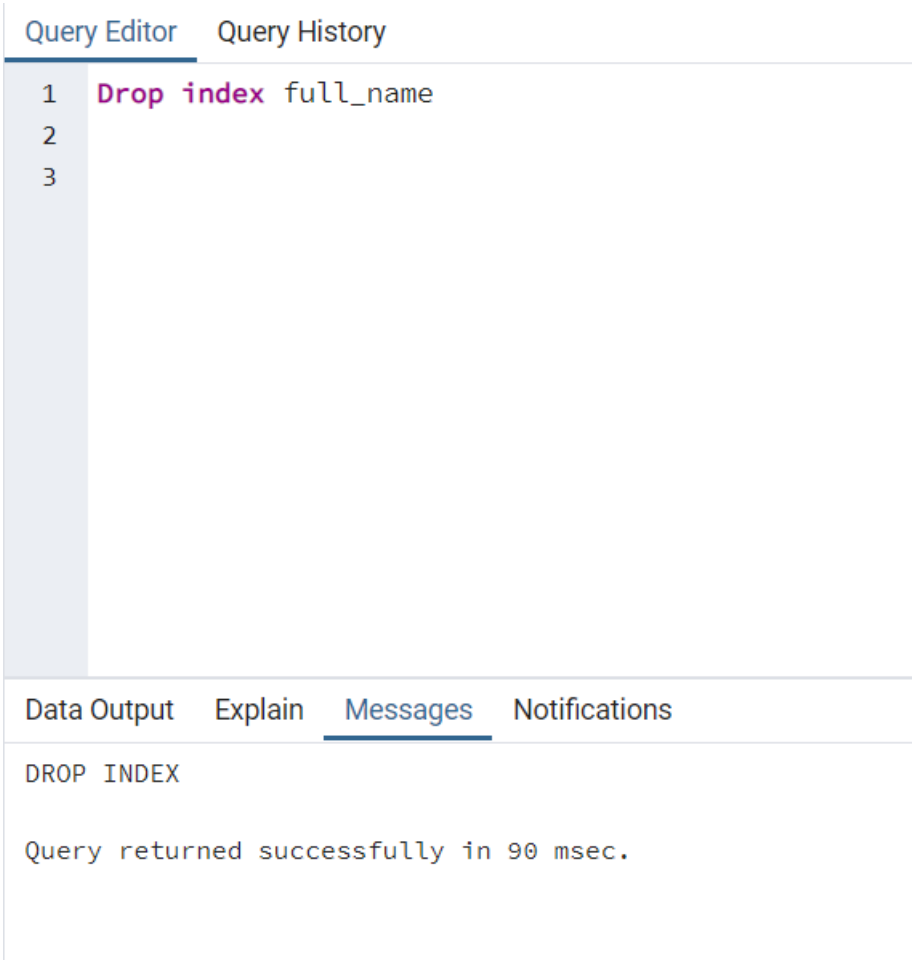
```
1  EXPLAIN SELECT * FROM "reader" WHERE fist_name LIKE 'A%' AND last_name LIKE '%a';
2
3
```

Data Output | Explain | Messages | Notifications

	QUERY PLAN text
1	Seq Scan on reader (cost=0.00..1.04 rows=1 width=400)
2	[...] Filter: (((fist_name)::text ~~ 'A%'::text) AND ((last_name)::text ~~ '%a'::text))

Удаление индексов:

DROP INDEX full_name;



ВЫВОДЫ

SQL запросы позволяют изменять, добавлять или удалять данные, атакже составлять различные выборки, подсчитывать числовые характеристики.

Сравнив время выполнения запросов с индексами и без, можно сделать вывод, что с индексами запросы выполнялись немного медленнее. Это связано с небольшим количеством данных в таблице.