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 K3240 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

**A picture containing diagram

Description automatically generated**

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

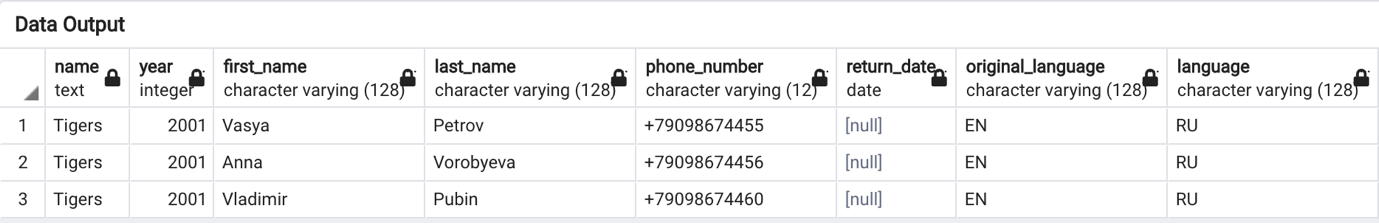
LEFT JOIN "User" AS u

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;



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

SELECT \* FROM (SELECT t1.reg\_user\_id, COUNT(t1.return\_date) as \_count FROM "Ticket" AS t1

LEFT JOIN "RegisteredUser" AS t2 ON t1.reg\_user\_id = t2.reg\_user\_id

WHERE "official\_return\_date"<"return\_date" GROUP BY t1.reg\_user\_id) AS t3 WHERE \_count > 10;

Table

Description automatically generated

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

SELECT COUNT(\*) FROM (SELECT \* FROM (SELECT t1.reg\_user\_id, COUNT(t1.return\_date) as \_count FROM "Ticket" AS t1

LEFT JOIN "RegisteredUser" AS t2 ON t1.reg\_user\_id = t2.reg\_user\_id

WHERE "official\_return\_date"<"return\_date" GROUP BY t1.reg\_user\_id) AS t3 WHERE \_count > 10) AS t5 GROUP BY reg\_user\_id;

Table

Description automatically generated

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

SELECT \* FROM (SELECT book\_id, COUNT(book\_id) FROM "CopyBook" GROUP BY book\_id) AS t1 WHERE count =1;

Table

Description automatically generated

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

SELECT COUNT(\*) FROM (

SELECT DISTINCT reg\_user\_id FROM "Ticket"

WHERE get\_date < '2022-01-01'

OR return\_date < '2022-01-01'

GROUP BY reg\_user\_id) AS T1;

Table

Description automatically generated

1. *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;

Graphical user interface, application, table

Description automatically generated

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

**Сегодня 2022-01-31**

SELECT name, official\_return\_date, return\_date FROM "Book" as t4 RIGHT JOIN (

SELECT \* FROM "CopyBook" AS t2 RIGHT JOIN (

SELECT \* FROM "Ticket"

WHERE return\_date IS NULL

AND official\_return\_date-'2022-01-31'<=3

AND official\_return\_date-'2022-01-31'>0) AS t1 ON t1.copy\_book\_id=t2.book\_copy\_id) AS t3 ON t4.book\_id=t3.book\_id

WHERE t4.name LIKE '%C#%';

**Table

Description automatically generated**

Задание 2.

Создать представления для администрации библиотеки, содержащие:

1. сведения о должниках

CREATE VIEW CREDITS\_USER\_LIST AS

SELECT t4.\*, t3.credits FROM "User" t4 RIGHT JOIN

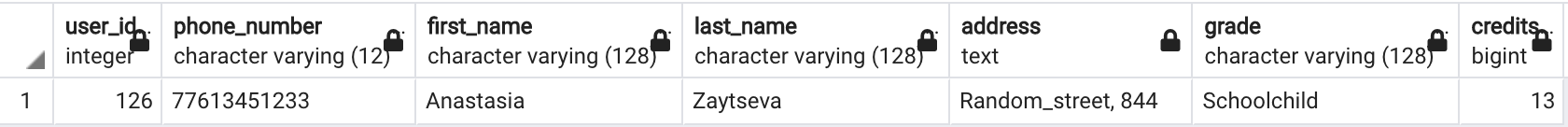
(SELECT t2.reg\_user\_id, t2.user\_id, COUNT(t1.return\_date) AS credits FROM "Ticket" AS t1

LEFT JOIN "RegisteredUser" AS t2 ON t1.reg\_user\_id = t2.reg\_user\_id

WHERE "official\_return\_date"<"return\_date" GROUP BY t2.reg\_user\_id) AS t3 ON t3.user\_id=t4.user\_id;

Запрос:

SELECT \* FROM credits\_user\_list;



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

CREATE VIEW BOOK\_TOP\_LIST AS

SELECT t4.name, t4.part, t3.book\_id, COUNT(t3.book\_copy\_id) FROM "Book" AS t4 RIGHT JOIN (

SELECT \* FROM "CopyBook" AS t2 RIGHT JOIN (

SELECT \* FROM "Ticket" WHERE return\_date IS NULL) AS t1

ON t1.copy\_book\_id=t2.book\_copy\_id) AS t3

ON t3.book\_id=t4.book\_id GROUP BY (t3.book\_id, t4.name, t4.part) ORDER BY count DESC;

Запрос:

SELECT \* FROM book\_top\_list;

Table

Description automatically generated

Задание 3.

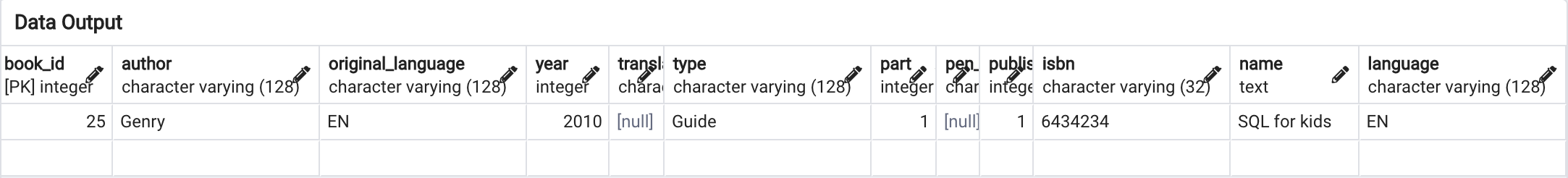
1. Запрос на модификацию данных с INSERT (добавление новой книги):

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';



1. 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):

Table

Description automatically generated

1. 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:

Graphical user interface, application

Description automatically generated

Теперь их нет:

Graphical user interface, application

Description automatically generated

Задание 4.

Графическое представление запроса SELECT \* FROM book\_top\_list;:

Diagram

Description automatically generated

История запросов:

Text

Description automatically generated with medium confidence

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.

Graphical user interface, table

Description automatically generated with medium confidence

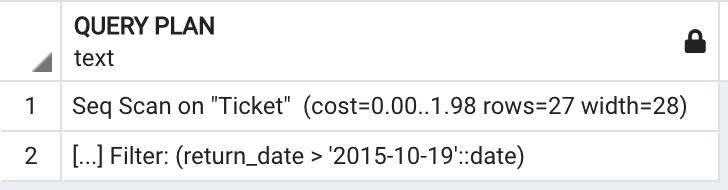
CREATE INDEX date\_of\_return ON "Ticket"(return\_date);

После:

EXPLAIN SELECT \* FROM "Ticket" WHERE return\_date > '2015-10-19';

Successfully run. Total query runtime: 70 msec.

2 rows affected.



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.

Graphical user interface

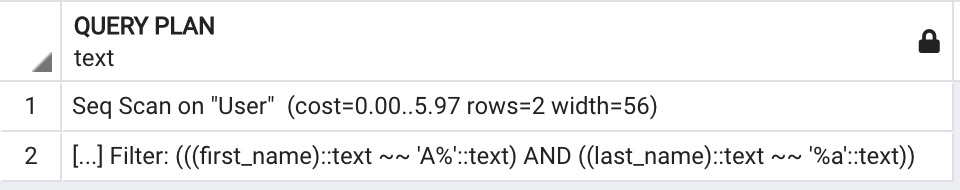
Description automatically generated with medium confidence

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.



Удаление индексов:

Table

Description automatically generated

DROP INDEX full\_name;

Table

Description automatically generated

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