

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение высшего образования
«Национальный исследовательский университет ИТМО»

Факультет инфокоммуникационных технологий

ЛАБОРАТОРНАЯ РАБОТА №2

ЗАПРОСЫ НА ВЫБОРКУ И МОДИФИКАЦИЮ ДАННЫХ, ПРЕДСТАВЛЕНИЯ И ИНДЕКСЫ В POSTGRESQL

по дисциплине:
«Проектирование и Реализация Баз Данных»

Выполнил:
студент II курса ИКТ
группы К3241
Селезнев Илья.

Санкт-Петербург
2022

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

Задачи:

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 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.

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. БД должна содержать следующий минимальный набор сведений:

- Название (заглавие) издания;
- Автор (фамилия и имя (инициалы) или псевдоним автора издания);
- Язык, с которого выполнен перевод издания;
- Номер тома (части, книги, выпуска);
- Область знания;
- Вид издания (сборник, справочник, монография ...);
- Переводчик (фамилия и инициалы переводчика);
- Место издания (город);
- Год выпуска издания;
- Издательство (название издательства);
- Библиотечный шифр (например, ББК 32.973);
- Номер стеллажа в комнате;
- Номер (инвентарный номер) экземпляра;

- Номер комнаты (помещения для хранения экземпляров);
- Номер полки на стеллаже;
- Дата изъятия экземпляра с установленного места;
- Цена конкретного экземпляра;
- Номер читательского билета (формуляра);
- Фамилия читателя;
- Имя читателя;
- Отчество читателя;
- Адрес читателя;
- Телефон читателя.

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

Задание 2. Создать запросы:

- Вывести загрузку преподавателей в понедельник (в часах).
- Найти недельную нагрузку студентов каждой группы (в часах).
- Вывести список свободных лекционных аудиторий в заданное время.
- Вывести количество аудиторий каждого типа.
- Вывести еженедельное количество часов занятий для каждой группы.
- Найти номера аудиторий каждого типа, имеющих максимальное количество мест. - Вывести фамилии преподавателей, которые всегда проводят практические занятия в одной и той же аудитории.

Задание 3. Создать представление:

- содержащее данные о расписании заданной группы на каждый день; -средняя недельная аудиторная нагрузка по группам по каждому направлению.

Схема базы данных представлена на рисунке 1.

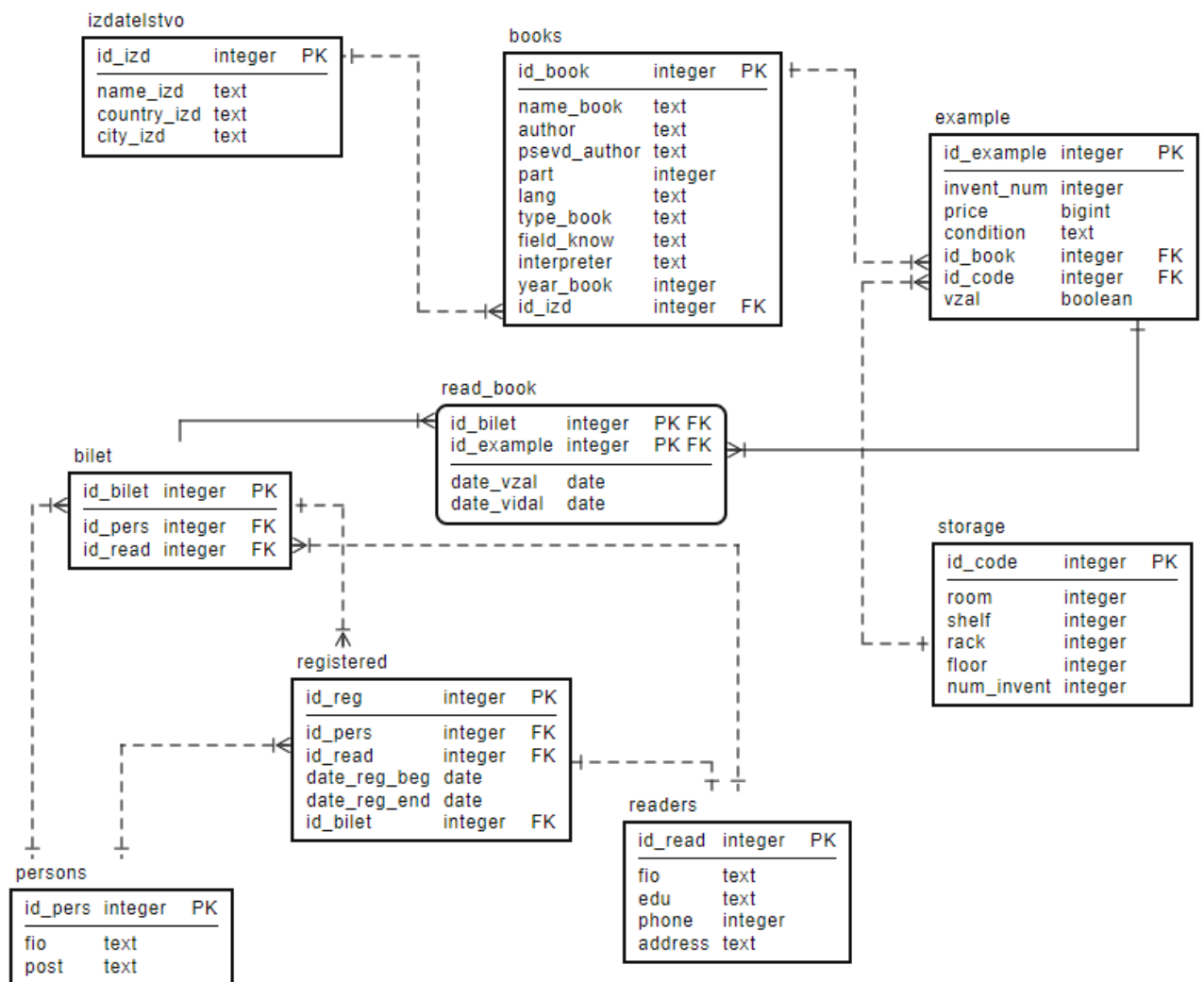


Рисунок 1 - схема базы данных

Выполнение

Задание 2. Создайте запросы:

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

```

SELECT r.fio
FROM books b
join example e on e.id_book = b.id_book
join read_book rb on rb.id_example = e.id_example
join billet bi on bi.id_bilet = rb.id_bilet
join readers r on r.id_read = bi.id_read
where rb.date_vozvrat < now()
and lang = 'Английский'
and year_book > 2000
and e.vzal = true
  
```

	fio text
1	ФИО 1
2	ФИО 2
3	ФИО 3
4	ФИО 4
5	ФИО 5
6	ФИО 6

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

```
SELECT r.fio,
       count(rb.id_example) as cnt
FROM example e
join read_book rb on rb.id_example = e.id_example
join билет b on b.id_bilet = rb.id_bilet
join readers r on r.id_read = b.id_read
where rb.date_vozvrat < now()
and e.vzal = true
group by r.id_read
having count(rb.id_example) > 10
```

	fio text	cnt bigint
1	ФИО 1	11

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

```
with cnt_book as (
SELECT r.id_read,
       count(rb.id_example) as cnt
FROM example e
join read_book rb on rb.id_example = e.id_example
join билет b on b.id_bilet = rb.id_bilet
join readers r on r.id_read = b.id_read
where rb.date_vozvrat < now()
and e.vzal = true
group by r.id_read
having count(rb.id_example) > 10)
select count(c.id_read)
from readers r
join cnt_book c on c.id_read = r.id_read
```

	count bigint
1	1

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

```

select b.id_book,
       b.name_book,
       count(e.id_example)
from books b
join example e on e.id_book = b.id_book
group by b.id_book, b.name_book
having count(e.id_example) = 1

```

	id_book [PK] integer	name_book text	count bigint
1	15	Книга 15	1
2	19	Книга 19	1
3	14	Книга 14	1
4	13	Книга 13	1
5	20	Книга 20	1
6	18	Книга 18	1
7	11	Книга 11	1
8	17	Книга 17	1
9	16	Книга 16	1
10	12	Книга 12	1

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

```

select count(r.id_read)
from readers r
join билет b on b.id_read = r.id_read
join read_book rb on rb.id_bilet = b.id_bilet
where rb.date_vzal < current_date-365

```

	count bigint
1	1

- Подсчитать количество читателей библиотеки по уровню образования.

```

select edu, count(r.id_read)
from readers r
group by edu

```

	edu text	count bigint
1	Среднее	3
2	Начальное	3
3	Высшее	4

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

```

SELECT b.name_book,type_book
FROM public.books b
join example e on e.id_book = b.id_book
join read_book rb on rb.id_example = e.id_example
where type_book = 'Программирование на C#'
and vzial = true
and rb.date_vozvrat between current_date and current_date+3

```

	name_book text	type_book text
1	Книга 16	Программирование на C#

Задание 3. Создать представления для администрации библиотеки, содержащие:

- сведения о должниках;

```

CREATE OR REPLACE VIEW public.dolg
AS
SELECT r.fio,
       b.name_book,
       e.invent_num,
       rb.date_vzial,
       rb.date_vozvrat
FROM books b
JOIN example e ON e.id_book = b.id_book
JOIN read_book rb ON rb.id_example = e.id_example
JOIN билет bi ON bi.id_bilet = rb.id_bilet
JOIN readers r ON r.id_read = bi.id_read
WHERE rb.date_vozvrat < now() AND e.vzial = true;

ALTER TABLE public.dolg
OWNER TO postgres;

```

	fio text	name_book text	invent_num integer	date_vzal date	date_vozvrat date
1	ФИО 1	Книга 2	10001	2022-06-01	2022-06-03
2	ФИО 2	Книга 6	10005	2022-06-01	2022-06-03
3	ФИО 3	Книга 11	10010	2022-06-01	2022-06-03
4	ФИО 4	Книга 19	10018	2022-06-01	2022-06-03
5	ФИО 5	Книга 2	10021	2022-06-01	2022-06-03
6	ФИО 6	Книга 6	10025	2022-06-01	2022-06-03
7	ФИО 1	Книга 1	10000	2022-06-01	2022-06-03
8	ФИО 1	Книга 3	10002	2022-06-01	2022-06-03
9	ФИО 1	Книга 4	10003	2022-06-01	2022-06-03
10	ФИО 1	Книга 5	10004	2022-06-01	2022-06-03
11	ФИО 1	Книга 6	10005	2022-06-01	2022-06-03
12	ФИО 1	Книга 7	10006	2022-06-01	2022-06-03
13	ФИО 1	Книга 8	10007	2022-06-01	2022-06-03
14	ФИО 1	Книга 9	10008	2022-06-01	2022-06-03
15	ФИО 1	Книга 10	10009	2022-06-01	2022-06-03
16	ФИО 1	Книга 12	10011	2022-06-01	2022-06-03
17	ФИО 9	Книга 10	10029	2020-06-01	2020-06-03

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

```

CREATE OR REPLACE VIEW public.top_read_book
AS
SELECT b.name_book,
       b.author,
       b.type_book,
       b.year_book
FROM books b
      JOIN example e ON e.id_book = b.id_book
WHERE e.vzal = true;

ALTER TABLE public.top_read_book
OWNER TO postgres;

```


	name_book text	author text	type_book text	year_book integer
1	Книга 1	Автор 1	Мировая	2001
2	Книга 2	Автор 2	Алгебра	2001
3	Книга 3	Автор 3	Программирование на Java	2001
4	Книга 4	Автор 4	Квантовая	2001
5	Книга 5	Автор 1	Италии	2001
6	Книга 6	Автор 3	Программирование на C#	2001
7	Книга 7	Автор 1	России	2001
8	Книга 8	Автор 2	Геометрия	2001
9	Книга 9	Автор 1	Америки	2001
10	Книга 10	Автор 4	Квантовая	2001
11	Книга 11	Автор 1	Мировая	2001
12	Книга 12	Автор 4	Квантовая	2001
13	Книга 13	Автор 2	Тригонометрия	2001
14	Книга 14	Автор 5	Здания	2001
15	Книга 15	Автор 1	Мировая	2001
16	Книга 16	Автор 3	Программирование на C#	2001

Составить 3 запроса на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов

```
INSERT INTO public.read_book(
id_bilet, id_example, date_vzal, date_vozvrat)
VALUES (
(select id_bilet from bilet b
join readers r on r.id_read = b.id_read
where fio = 'ФИО 7'),
(select id_example from example e
join books b on b.id_book = e.id_book
where name_book = 'Книга 7'),
'2022-06-06',
'2022-06-16'
);
```

```
UPDATE public.example
SET price=price+250
WHERE id_book in (
select id_book
from books
where author = 'Автор 4');
```

```
DELETE FROM public.persons
WHERE post = 'Подчиненная';
```

История запросов

Show queries generated internally by pgAdmin?

Да

16:13:18

▶ SELECT b.id_book, e.id_example, type_book FROM publi...

16:13:03

▶ INSERT INTO public.read_book(id_bilet, id_example, ...

16:12:57

▶ delete from read_book where id_bilet = 20

16:12:51

▶ INSERT INTO public.read_book(id_bilet, id_example, ...

16:10:30

▶ SELECT id_bilet, id_example, date_vzal, date_vozvrat...

16:10:10

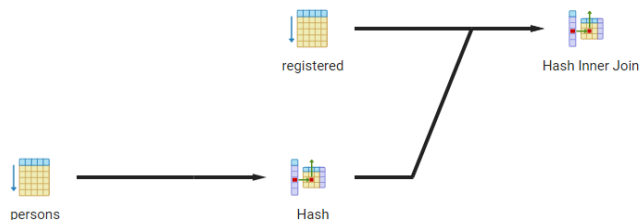
▶ SELECT b.id_book, e.id_example, type_book FROM publi...

Простой запрос

```
3 select p.*
4 from persons p
5 join registered r on r.id_pers = p.id_pers
```

Результат План выполнения Сообщения Notifications

Graphical Analysis Статистика



✓ Запрос выполнен успешно. Общее время выполнения: 137 мсек. обработано строк: 1.

Добавление составного индекса

```
CREATE INDEX test_pers ON persons (id_pers, fio);
```

Выполнение того же запроса

Выводы

Созданы запросы и представления на выборку данных к базе данных PostgreSQL, составлены запросы на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов, изучено графическое представление запросов, созданы

простой и составной индексы для двух произвольных запросов, проведено сравнение времени выполнения запросов без индексов и с индексами.