

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 2 курса ФИКТ
группы K3241
Тимофеев Н.А.

Проверил:
Говорова М.В.

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 команд у EXPLAIN.

Индивидуальное задание. Вариант 5. БД «Издательство компьютерной литературы»

Описание предметной области: Издательство занимается выпуском литературы по различным областям информатики. Покупатели книг приобретают книги на базе издательства. Когда на базе заканчиваются книги, издается дополнительный тираж. В каждом заказе заказчик может заказать разную литературу.

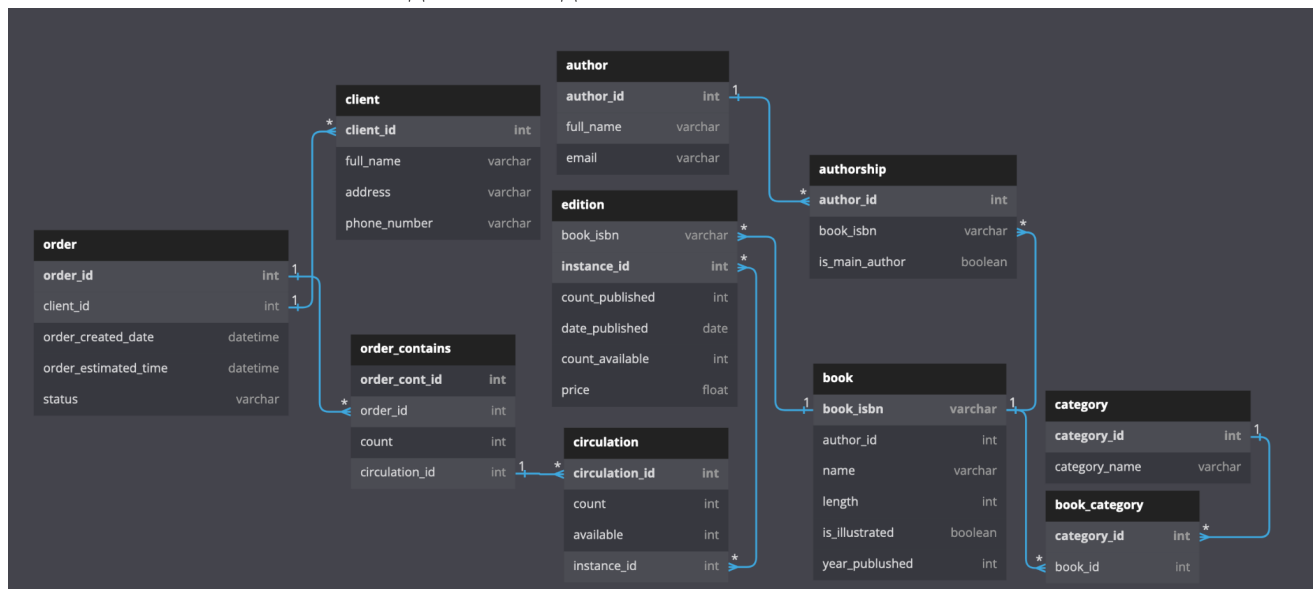
БД должна содержать следующий минимальный набор сведений: Фамилия автора. Имя автора. Отчество автора. Код автора. E-mail автора. Код ISBN. Название книги. Количество страниц. Наличие иллюстраций. Код категории книги. Категория книги. Количество страниц. Год начала издания. Розничная цена книги. Тираж. Дата тиража. Количество экземпляров на базе издательства. Код заказчика. Фамилия заказчика. Имя заказчика. Отчество заказчика. Адрес заказчика. Телефон заказчика. Код заказа. Дата заказа. Срок заказа. Количество экземпляров книги в заказе. Статус заказа.

1. БД «Издательство компьютерной литературы»

2. Сущности:

- a. Писатель (Код автора (PK), ФИО автора, Электронная почта автора)
- b. Книга (Код ISBN (PK), название книги, количество страниц, наличие иллюстраций, год начала издания)
- c. Категория (Код категории книги (PK), название категории)
- d. Издание (Номер издания (PK), дата тиража, Код ISBN, тираж, кол-во экземпляров на базе издательства, код заказа, розничная цена книги)
- e. Заказчик (Код заказчика (PK), ФИО заказчика, адрес заказчика, телефон заказчика)
- f. Заказ (Код заказа (PK), код заказчика, дата заказа, срок заказа, статус заказа)
- g. Состав заказа (Номер состава заказа (PK), номер заказа, номер издания, кол-во экземпляров книги в заказе)

3. Схема логической модели базы данных



4. Запросы и представления:

- Список книг, изданных в текущем году и относящихся к категории «Programming».

```
SELECT * FROM book WHERE book_isbn IN (SELECT book_category.book_id
FROM category, book_category WHERE category.category_id =
book_category.category_id AND category.category_name =
'Programming' )
```

| book_isbn | name | length | is_illustrated | year_published |
|-----------|--------------------|--------|----------------|----------------|
| 1 | Python programming | 1120 | false | 2022 |

- Список покупателей, заказавших книг на сумму, превышающую среднюю сумму заказа.

```
SELECT c.full_name FROM "order"
JOIN client c on "order".client_id = c.client_id
JOIN public.order_contains oc on "order".order_id = oc.order_id
JOIN circulation c2 on oc.circulation_id = c2.circulation_id
JOIN edition e on c2.instance_id = e.instance_id
WHERE oc.count * e.price > (SELECT
round(avg(order_contains.count * e.price)) FROM order_contains JOIN
"order" o on order_contains.order_id = o.order_id JOIN circulation
c on order_contains.circulation_id = c.circulation_id JOIN edition
e on c.instance_id = e.instance_id)
GROUP BY c.full_name;
```

| Output | |
|------------------------------|------------------|
| GROUP BY "order".order_id, X | |
| 4 rows | |
| full_name | |
| 1 | Ivan Semenov |
| 2 | Peter Zyablow |
| 3 | Inna Maina |
| 4 | Nikolas Timofeev |

- Список книг, которые не заказывались в течение последних двух кварталов.

```
SELECT book.* FROM book
JOIN edition e on book.book_isbn = e.book_isbn
JOIN circulation c on e.instance_id = c.instance_id
JOIN order_contains oc on c.circulation_id = oc.circulation_id
JOIN "order" o on oc.order_id = o.order_id
WHERE o.order_created_date < (CURRENT_DATE::date - INTERVAL '2
months')::date;
```

| book_isbn | name | length | is_illustrated | year_published |
|-----------|--------------------|--------|----------------|----------------|
| 1 | Python programming | 1120 | false | 2022 |

- Список авторов, не написавших ни одной книги, относящейся к категории “Языки программирования”.

```
SELECT author.full_name, c.category_name FROM author
JOIN authorship a on author.author_id = a.author_id
JOIN book b on a.book_isbn = b.book_isbn
JOIN book_category bc on b.book_isbn = bc.book_id
JOIN category c on bc.category_id = c.category_id
WHERE c.category_name <> 'Programming'
```

| | full_name | category_name |
|---|--------------------|---------------|
| 1 | Zarina Zemljanchka | Physics |
| 2 | Some author | Tech |
| 3 | Zarina Zemljanchka | Tech |
| 4 | Mikolas Timofeev | Tech |

- Список книг, в названиях которых содержится слово “Python” и которые присутствуют на базе в количестве, превышающем 50 экземпляров.

```
SELECT book.name, c.available FROM book JOIN edition e on
book.book_isbn = e.book_isbn JOIN circulation c on e.instance_id =
c.instance_id WHERE book.name LIKE '%Python%' AND c.available >=
50;
```

| name | available |
|----------------------|-----------|
| 1 Python programming | 91 |

- Покупателя, сделавшего заказ на максимальную сумму за последний месяц.

```

SELECT client.*
FROM "order" o
    JOIN client on client.client_id = o.client_id
    JOIN order_contains oc on o.order_id = oc.order_id
    JOIN circulation c on oc.circulation_id = c.circulation_id
    JOIN edition e on c.instance_id = e.instance_id
WHERE (SELECT max(e.price * oc.count)
FROM circulation c
    JOIN "order" o
    JOIN client on client.client_id = o.client_id
    JOIN order_contains oc on o.order_id = oc.order_id
        on oc.circulation_id = c.circulation_id
    JOIN edition e on e.instance_id = c.instance_id
WHERE o.order_created_date >= (CURRENT_DATE - INTERVAL '1 month')
and o.order_created_date <= CURRENT_DATE) =
e.price * oc.count;

```

Output aboba_db.public.client

| | client_id | full_name | address | phone_number |
|---|-----------|------------|-----------------------------|--------------|
| 1 | 3 | Inna Maina | lanskoye shosse 21k1 kv 122 | +79535555281 |

- Список книг, не попавших ни в один из заказов в течение последнего года.

```

SELECT *
FROM book
WHERE book_isbn NOT IN (SELECT distinct e.book_isbn
FROM order_contains oc
    JOIN book
    JOIN edition e on book.book_isbn = e.book_isbn
    JOIN circulation c on e.instance_id = c.instance_id
        on c.circulation_id = oc.circulation_id)

```

Output aboba_db.public.book

| | book_isbn | name | length | is_illustrated | year_published |
|---|-----------|-------------------------|--------|----------------|----------------|
| 1 | 5 | PostgreSQL v13 Handbook | 1200 | true | 2019 |

- Представление содержащее сведения о количестве заказанных экземпляров каждой книги, изданной в текущем году

```

CREATE VIEW book_best sellers AS
SELECT book.name, sum(oc.count)
FROM book
    JOIN edition on book.book_isbn = edition.book_isbn
    JOIN circulation c on edition.instance_id = c.instance_id
    JOIN order_contains oc on c.circulation_id = oc.circulation_id
WHERE book.year_published = date_part('year', CURRENT_DATE)
GROUP BY book.name

```

| Output Result 391 | |
|--------------------------|-----|
| 2 rows | |
| name | sum |
| 1 Guide on ORM in Python | 3 |
| 2 Python programming | 13 |

- Представление количества заказов по покупателям за последний год

```

CREATE VIEW customers_yearly_rating AS
SELECT client.full_name, count(*) orc_cnt FROM client
    JOIN "order" o on client.client_id = o.client_id
    JOIN order_contains oc on o.order_id = oc.order_id
WHERE date_part('year', o.order_created_date) = date_part('year', CURRENT_DATE)
GROUP BY client.full_name
ORDER BY orc_cnt DESC

```

| 4 rows | |
|--------------------|---------|
| full_name | orc_cnt |
| 1 Inna Maina | 6 |
| 2 Peter Zyablow | 5 |
| 3 Ivan Semenov | 2 |
| 4 Nikolas Timofeev | 2 |

5. Хранимые процедуры

- Для снижения цен на книги, которые находятся на базе в количестве, превышающем 1000 штук.

```

UPDATE edition e
SET price = price * 0.90
WHERE instance_id IN (SELECT instance_id FROM circulation WHERE available > 100)

```

Цены до обновления:

| price |
|-------|
| 1000 |
| 1400 |
| 990 |
| 9990 |
| 200 |
| 300 |
| 2900 |

Цены после обновления:

| date_published | price |
|----------------|-------|
| 2002-10-02 | 1000 |
| 2023-10-02 | 1400 |
| 2019-11-02 | 990 |
| 2022-03-02 | 9990 |
| 2019-10-09 | 2900 |
| 2015-10-02 | 180 |
| 2017-10-02 | 270 |

- Для ввода новой книги.

```
INSERT INTO book
VALUES ((select max(book_isbn)::int + 1 FROM book), 'Clean Architecture', 670, true, 2022)
```

| book_isbn | name | length | is_illustrated |
|-----------|--------------------------|--------|----------------|
| 2 | Art and Science | 122 | true |
| 3 | Popular Mechanics | 320 | true |
| 4 | Guide on ORM in Python | 220 | true |
| 5 | PostgreSQL v13 Handbo... | 1200 | true |
| 6 | Clean Architecture | 670 | true |

- Для ввода нового заказа.

```
INSERT INTO "order"
VALUES ((SELECT max(order_id) + 1 FROM "order"),
(SELECT client_id FROM client WHERE full_name = 'Nikolas Timofeev'),
CURRENT_DATE::date, CURRENT_DATE + INTERVAL '10 days', 'process')
```

| orde... | client_id | order_crea... | order_esti... | status |
|---------|-----------|---------------|---------------|----------|
| 5 | 2 | 2022-09-15 | <null> | canceled |
| 6 | 2 | 2022-09-17 | 2022-09-27 | finished |
| 7 | 1 | 2022-09-21 | 2022-09-29 | shipping |
| 8 | 2 | 2022-02-21 | 2022-02-28 | finished |
| 9 | 4 | 2022-10-02 | 2022-10-12 | process |

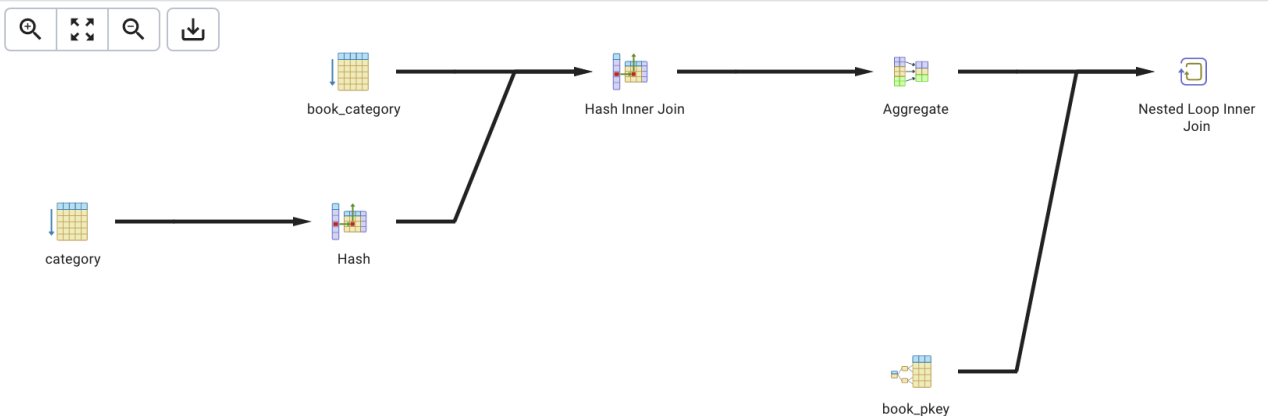
6. Добавление индексов

Запрос 1 без индексов:

```
1 SELECT *
2 FROM book
3 WHERE book_isbn IN (SELECT book_category.book_id
4                      FROM category, book_category
5                      WHERE category.category_id = book_category.category_id
6                      AND category.category_name = 'Programming');
```

Data output Messages Explain X Notifications

Graphical Analysis Statistics



Data output Messages Explain X Notifications

Successfully run. Total query runtime: 337 msec.
1 rows affected.

QUERY PLAN

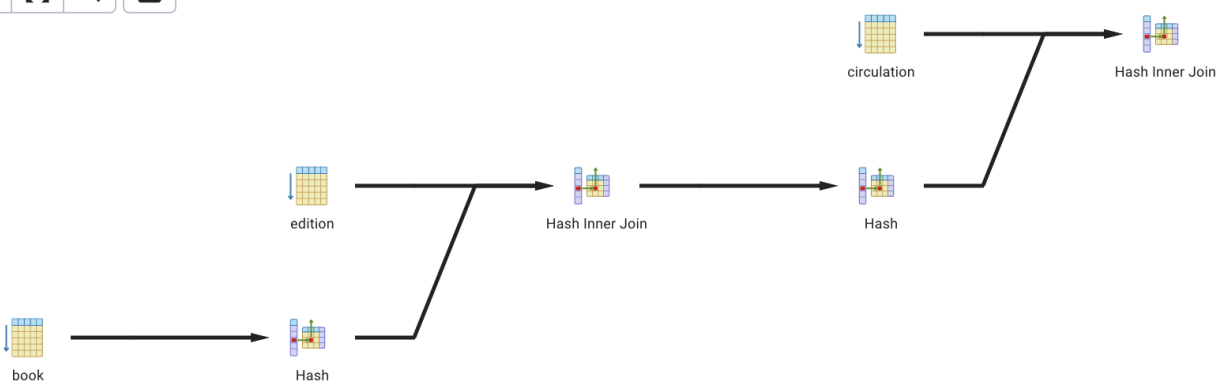
```
13 -> Index scan using book_pkey on book (...
14      Index Cond: ((book_isbn)::text = (b...
15 Planning Time: 3.431 ms
16 Execution Time: 3.632 ms
```


Запрос 4 без индексов:

```
1 SELECT book.name, c.available
2 FROM book
3     JOIN edition e on book.book_isbn = e.book_isbn
4     JOIN circulation c on e.instance_id = c.instance_id
5 WHERE book.name LIKE '%Python%' AND c.available >= 50;
```

Data output Messages Explain X Notifications

Graphical Analysis Statistics



Data output Messages Explain X Notifications

Successfully run. Total query runtime: 67 msec.
1 rows affected.

Создадим простой индекс для запроса 1:

```
CREATE INDEX category_name ON category(category_name);
```

Время выполнения после создания индекса:

```
-> Index Scan using category_name on category (cost=0.13..8.15 rows=
      Index Cond: ((category_name)::text = 'Programming'::text)
-> Index Scan using book_category_pkey on book_category (cost=0.15..
      Index Cond: (category_id = category.category_id)

Planning Time: 0.772 ms
Execution Time: 1.916 ms
```

Результат: Ускорение выполнения запроса в ~2 раза

Создадим составной индекс для запроса 4:

```
CREATE INDEX available_books ON circulation(circulation_id,
available);
```

Время выполнения после создания индекса:

| | |
|----|---|
| 3 | -> Index Scan using available_books on |
| 4 | Index Cond: (available >= 50) |
| 5 | -> Index Scan using edition_pkey o... |
| 6 | Index Cond: (instance_id = c... |
| 7 | -> Index Scan using book_pkey on book (...) |
| 8 | Index Cond: ((book_isbn)::text = (e... |
| 9 | Filter: ((name)::text ~~ '%Python%'... |
| 10 | Rows Removed by Filter: 1 |
| 11 | Planning Time: 0.802 ms |
| 12 | Execution Time: 0.396 ms |

Результат: ускорение выполнения запроса в ~4 раза.

Вывод: с помощью SQL запросов были получены, изменены, удалены и добавлены данные в базу (Проведены CRUD-операции). Время выполнения запросов с использованием индексов стало несколько быстрее, однако, использование ресурсов увеличилось.