

Министерство науки высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
ИТМО»

О Т Ч Е Т

по лабораторной работе «Запросы на выборку и модификацию данных,
представления и индексы в PostgreSQL»

по дисциплине «**Проектирование и реализация баз данных**»

Автор: Бакшилова Анастасия Денисовна

Факультет: ИКТ

Группа: K33391

Преподаватель: Говорова М. М.

Дата: 23.10.2023



Санкт-Петербург

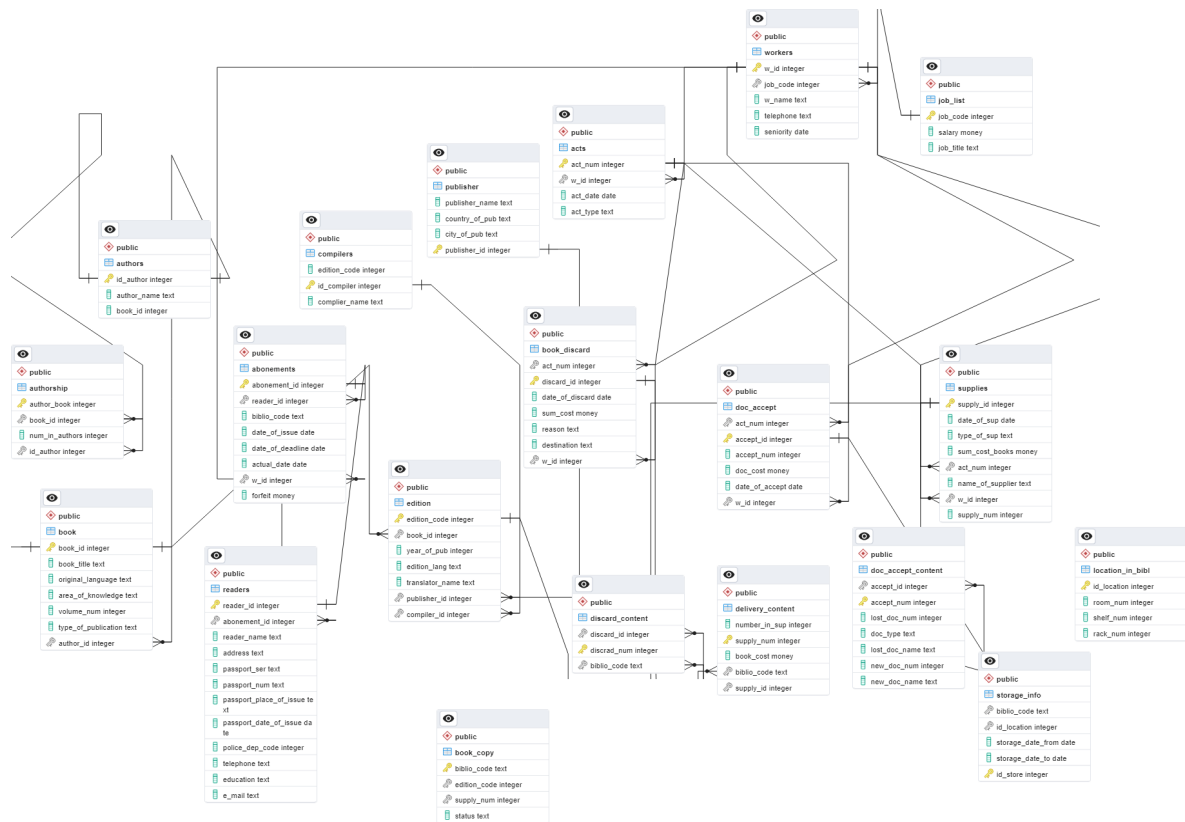
2023

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

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.

Схема БД в pgAdmin:



Задание 1 запросы:

Создайте запросы:

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

Запрос:

```
SELECT DISTINCT a.abonement_id
FROM abonements AS a
INNER JOIN book_copy AS bc ON a.biblio_code = bc.biblio_code
INNER JOIN edition AS e ON bc.edition_code = e.edition_code
INNER JOIN book AS b ON e.book_id = b.book_id
WHERE e.year_of_pub > 2000
      AND b.original_language = 'Английский'
      AND e.edition_lang != 'Английский'
      AND a.date_of_deadline > CURRENT_DATE;
```

Ответ:

	abonement_id integer
1	8965
2	9877

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

Запрос:

```
SELECT ab.reader_id, COUNT(ab.abonement_id) AS num_of_books
FROM abonements ab
WHERE ab.actual_date > ab.date_of_deadline
GROUP BY ab.reader_id
HAVING COUNT(ab.abonement_id) > 10;
```

Ответ:

	reader_id integer	num_of_books bigint
1	1940819	11

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

Запрос:

```
SELECT COUNT(*) AS num_of_readers
FROM (
    SELECT reader_id
    FROM abonements
```

```

WHERE actual_date > date_of_deadline
GROUP BY reader_id
HAVING COUNT(*) > 10
) AS subquery;

```

Ответ:

	num_of_readers bigint
1	1

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

Запрос:

```

SELECT b.book_title
FROM book AS b
WHERE b.book_id IN (
    SELECT e.book_id
    FROM edition AS e
    WHERE e.edition_code IN (
        SELECT bc.edition_code
        FROM book_copy AS bc
        WHERE bc.biblio_code NOT IN (
            SELECT dc.biblio_code
            FROM discard_content AS dc
        )
    )
)
GROUP BY b.book_title
HAVING COUNT(*) = 1;

```

Ответ:

	book_title text
1	Замок
2	Королёк - птичка певчая

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

Запрос:

```
WITH LatestReader AS (  
    SELECT MAX(r.abonement_id) AS latest_abonement_id, r.passport_ser,  
    r.passport_num  
    FROM readers r  
    GROUP BY r.passport_ser, r.passport_num  
)  
SELECT COUNT(*) AS num_of_readers  
FROM LatestReader AS lr  
JOIN abonements AS a ON lr.latest_abonement_id = a.abonement_id  
WHERE date_part('year', CURRENT_DATE) - date_part('year', a.date_of_issue) >=  
1;
```

Ответ:

	num_of_readers bigint
1	2

- Подсчитать количество читателей библиотеки по уровню образования.

Запрос:

```
SELECT education, COUNT(*) AS num_of_readers
FROM readers
GROUP BY education;
```

Ответ:


	education text	num_of_readers bigint
1	аспирантура	2
2	магистратура	3
3	основное общее	1
4	среднее профессиональное	2
5	бакалавриат	5

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

Запрос:

```
SELECT DISTINCT b.book_title
FROM book AS b
INNER JOIN edition AS e ON b.book_id = e.book_id
LEFT JOIN book_copy AS bc ON e.edition_code = bc.edition_code
LEFT JOIN abonements AS a ON bc.biblio_code = a.biblio_code
WHERE b.book_title LIKE '%C#%'
AND a.date_of_deadline <= CURRENT_DATE + INTERVAL '3 days'
AND a.abonement_id IS NULL;
```

Ответ:

	book_title text 
1	С# 10 и .NET 6. Современная кроссплатформенная разработ...
2	Изучаем С# через разработку игр на Unity
3	Программирование на С# для начинающих

Задание 1 представления:

Создать представления для администрации библиотеки, содержащие:

- сведения о должниках;

Код:

```
CREATE VIEW Debtors AS
SELECT r.reader_id, r.reader_name
FROM readers r
WHERE r.reader_id IN (
    SELECT a.reader_id
    FROM abonements a
    WHERE a.actual_date > a.date_of_deadline
);
```

Проверка:

```
SELECT * FROM Debtors;
```


Ответ:

	reader_id integer	reader_name text
1	3471789	Рожкова Айша Захаровна
2	3481784	Иванов Владимир Тимурович
3	1940819	Горшков Максим Владиславович
4	2384910	Васильев Алексей Михайлович
5	3224568	Нефедова Василиса Богдановна
6	7654389	Соколова Ульяна Владимировна
7	9765460	Воробьева Карина Ярославовна
8	988776	Павлова Мария Андреевна
9	2447970	Кузнецова Валерия Александровна

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

Код:

```
CREATE OR REPLACE VIEW popular_books AS
SELECT b.book_title, COUNT(bc.biblio_code) AS num_copies
FROM book b
JOIN edition e ON b.book_id = e.book_id
JOIN book_copy bc ON e.edition_code = bc.edition_code
LEFT JOIN abonements a ON bc.biblio_code = a.biblio_code
WHERE a.biblio_code IS NULL
GROUP BY b.book_title
HAVING COUNT(bc.biblio_code) > 1;
```

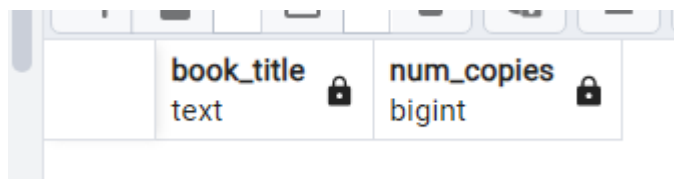
Ответ:

	book_title text	num_copies bigint
1	Война и Мир	4
2	Золотой ключик, или Приключения Буратино	2
3	Капитанская дочка	3
4	Королёк - птичка певчая	2
5	Программирование на C# для начинающих	3

Если учитывать, что у самых популярных книг в библиотеке не должно остаться экземпляров вообще, то можно переделать код вот так:

```
CREATE OR REPLACE VIEW popular_books AS
SELECT b.book_title, COUNT(bc.biblio_code) AS num_copies
FROM book b
JOIN edition e ON b.book_id = e.book_id
JOIN book_copy bc ON e.edition_code = bc.edition_code
LEFT JOIN abonements a ON bc.biblio_code = a.biblio_code
WHERE a.biblio_code IS NULL
GROUP BY b.book_title
HAVING COUNT(bc.biblio_code) = 0;
```

Ответ:



book_title	num_copies
text	bigint

Пояснение: пока что таких книг в библиотеке нет, но мы можем ориентироваться по ответу выше и найти самое маленькое кол-во экземпляров по столбику num_copies.

Задание 2 запросы на модификацию данных:

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

- INSERT INTO book (book_title, original_language, area_of_knowledge, volume_num, type_of_publication, author_id)
VALUES ('Отрочество', 'Русский', 'Художественная литература', 1, 'Книга',
(SELECT id_author FROM authors WHERE author_name = 'Толстой Лев Николаевич'));

b) UPDATE workers

SET seniority = seniority + INTERVAL '1 year'

WHERE w_id = (SELECT w_id FROM workers WHERE w_name = 'Лапина
Людмила Вадимовна');


c) DELETE FROM authors

WHERE id_author = (SELECT author_id FROM book WHERE book_title =
'Детство');

Простой запрос без использования индекса:

EXPLAIN ANALYZE SELECT * FROM book WHERE book_title = 'Война и Мир';


Ответ:

	QUERY PLAN	
	text	
1	Seq Scan on book (cost=0.00..2.55 rows=4 width=128) (actual time=0.021..0.025 rows=4 loops...	
2	Filter: (book_title = 'Война и Мир'::text)	
3	Rows Removed by Filter: 40	
4	Planning Time: 3.654 ms	
5	Execution Time: 0.041 ms	

Составной запрос без использования индекса:

EXPLAIN ANALYZE SELECT * FROM book WHERE book_title = 'Война и Мир'
AND author_id = '1';

Ответ:

	QUERY PLAN	
	text	
1	Seq Scan on book (cost=0.00..2.66 rows=1 width=128) (actual time=0.040..0.045 rows=4 loops...	
2	Filter: ((book_title = 'Война и Мир'::text) AND (author_id = 1))	
3	Rows Removed by Filter: 40	
4	Planning Time: 0.179 ms	
5	Execution Time: 0.066 ms	

Простой запрос с использованием индекса:

```
(CREATE INDEX idx_b_tit ON book (book_title);)
```

```
EXPLAIN ANALYZE SELECT * FROM book WHERE book_title = 'Война и Мир';
```

Ответ:

	QUERY PLAN text	
1	Seq Scan on book (cost=0.00..2.55 rows=4 width=128) (actual time=0.022..0.025 rows=4 loops...	
2	Filter: (book_title = 'Война и Мир'::text)	
3	Rows Removed by Filter: 40	
4	Planning Time: 0.743 ms	
5	Execution Time: 0.043 ms	

Составной запрос с использованием индекса:

```
(CREATE INDEX idx_b_tit_au_id ON book (book_title, author_id);)
```

```
EXPLAIN ANALYZE SELECT * FROM book WHERE book_title = 'Война и Мир'  
AND author_id = '1';
```

Ответ:

	QUERY PLAN text	
1	Seq Scan on book (cost=0.00..2.66 rows=1 width=128) (actual time=0.037..0.043 rows=4 loops...	
2	Filter: ((book_title = 'Война и Мир'::text) AND (author_id = 1))	
3	Rows Removed by Filter: 40	
4	Planning Time: 0.229 ms	
5	Execution Time: 0.064 ms	

Вывод: использование составного индекса уменьшает время поиска, так как он ускоряет запрос на поиск и доступ к данным, однако, простой запрос без индекса выполнялся быстрее, так как данные в столбце, по которому создан индекс, не были уникальными и индекс не принёс существенной пользы.

