

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

**ОТЧЕТ
О ЛАБОРАТОРНОЙ РАБОТЕ № 2**

по теме:

*«Запросы на выборку и модификацию данных, представления и
индексы в PostgreSQL»*

по дисциплине: Проектирование и реализация баз данных

Специальность:

45.03.04 Интеллектуальные системы в гуманитарной сфере

Проверила:

Говорова М.М.

Дата: «29» мая 2023 г.

Оценка _____

Выполнила:

студентка группы К32422

Королева Е. М.

Санкт-Петербург 2022/2023

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 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.

Выполнение работы:

Предметная область – Вариант 3. БД «Библиотека»

Наименование БД – library

Схема логической модели базы данных

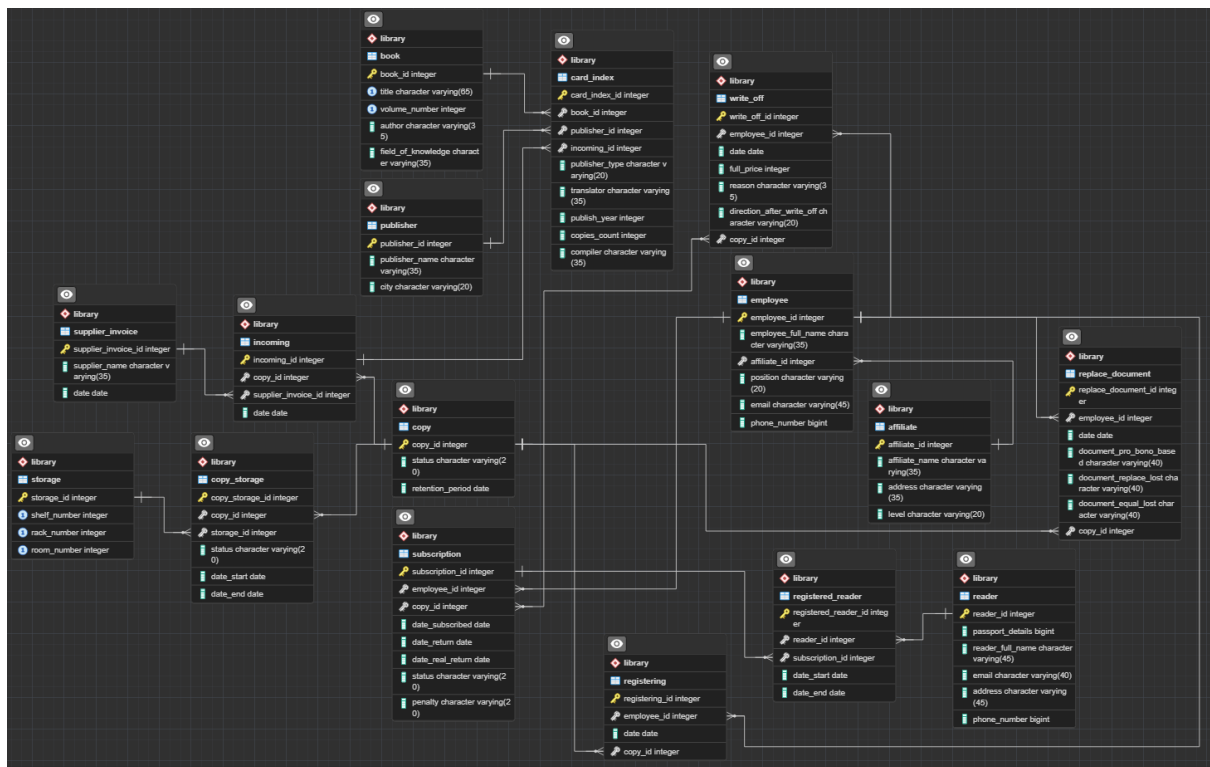


Рисунок 1 – ERD базы данных

Запросы на выборку

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

Запрос:

```
SELECT DISTINCT reader_full_name
FROM reader
JOIN registered_reader rr on reader.reader_id = rr.reader_id
JOIN subscription s on rr.subscription_id = s.subscription_id
JOIN copy c on s.copy_id = c.copy_id
JOIN incoming i on c.incoming_id = i.incoming_id
JOIN card_index ci on i.card_index_id = ci.card_index_id
JOIN book b on ci.book_id = b.book_id
WHERE language = 'Английский'
AND publish_year > 2000;
```

Вывод данных:

| | reader_full_name |
|---|-------------------------------------|
| 1 | Сидоров Сидор Сидорович |
| 2 | Смирнова Анна Владимировна |
| 3 | Сергеев Сергей Сергеевич |
| 4 | Иванов Иван Иванович |
| 5 | Александров Александр Александрович |
| 6 | Кузнецова Анастасия Ивановна |
| 7 | Иванова Екатерина Игоревна |
| 8 | Попова Ольга Александровна |

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

Запрос:

```
SELECT reader_full_name
FROM reader r
JOIN registered_reader rr on r.reader_id = rr.reader_id
JOIN subscription s on rr.subscription_id = s.subscription_id
WHERE status = 'активный'
AND date_real_return IS NULL
GROUP BY r.reader_id
HAVING COUNT(s.subscription_id) > 10
```

Вывод данных:

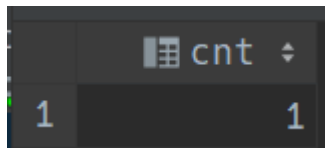
| | reader_full_name |
|---|-------------------------------------|
| 1 | Александров Александр Александрович |

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

Запрос:

```
SELECT COUNT(reader_full_name) cnt
FROM (
    SELECT reader_full_name
    FROM reader r
        JOIN registered_reader rr on r.reader_id = rr.reader_id
        JOIN subscription s on rr.subscription_id = s.subscription_id
    WHERE status = 'активный'
        AND date_real_return IS NULL
    GROUP BY r.reader_id
    HAVING COUNT(s.subscription_id) > 10
) t
```

Вывод данных:



| | cnt |
|---|-----|
| 1 | 1 |

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

Запрос:

```
SELECT title, author
FROM book b
    JOIN card_index ci on b.book_id = ci.book_id
    JOIN incoming i on ci.card_index_id = i.card_index_id
    JOIN copy c on i.incoming_id = c.incoming_id
GROUP BY b.book_id
HAVING COUNT(c.copy_id) = 1
ORDER BY 1, 2;
```

Вывод данных:

| | title | author |
|----|------------------------------------|-------------------|
| 1 | C# для начинающих | Джон Смит |
| 2 | C# и .NET: Подробное руководство | Марк Майкл |
| 3 | Властелин колец | Дж. Р. Р. Толкиен |
| 4 | Властелин колец | Дж. Р. Р. Толкиен |
| 5 | Властелин колец | Дж. Р. Р. Толкиен |
| 6 | Война и мир | Лев Толстой |
| 7 | Война и мир | Лев Толстой |
| 8 | Вокруг света за 80 дней | Жюль Верн |
| 9 | Гарри Поттер и Тайная комната | Дж. К. Роулинг |
| 10 | Гарри Поттер и узник Азкабана | Дж. К. Роулинг |
| 11 | Гарри Поттер и философский камень | Дж. К. Роулинг |
| 12 | Основные концепции математики | Р. Д. Кумар |
| 13 | Основы C# | Питер Джонсон |
| 14 | Преступление и наказание | Федор Достоевский |
| 15 | Путешествие в центр Земли | Жюль Верн |
| 16 | Скотный двор | Джордж Оруэлл |
| 17 | Теория относительности | Альберт Эйнштейн |
| 18 | Эффективное программирование на C# | Стив Мартин |

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

Запрос:

```
SELECT reader_id, reader_full_name
FROM (
    SELECT
        r.reader_id AS reader_id,
        reader_full_name,
        MAX(s.date_subscribed::timestamp) AS date_subscribed
    FROM reader r
        JOIN registered_reader rr ON r.reader_id = rr.reader_id
        JOIN subscription s ON rr.subscription_id = s.subscription_id
    GROUP BY r.reader_id
) t
WHERE NOW() - date_subscribed > INTERVAL '360 days'
ORDER BY 1, 2;
```

Вывод данных:

| | reader_id | reader_full_name |
|---|-----------|----------------------------|
| 1 | 1 | Иванов Иван Иванович |
| 2 | 6 | Попова Ольга Александровна |

6. Подсчитать количество читателей библиотеки по уровню образования.

Запрос:

```
SELECT level, COUNT(reader_id) readers_count
FROM registered_reader
    JOIN subscription s on registered_reader.subscription_id = s.subscription_id
    JOIN employee e on s.employee_id = e.employee_id
    JOIN affiliate a on e.affiliate_id = a.affiliate_id
GROUP BY level;
```

Вывод данных:

| | level | readers_count |
|---|---------|---------------|
| 1 | Высокий | 9 |
| 2 | Низкий | 8 |
| 3 | Средний | 11 |

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

Запрос:

```
SELECT title, author
FROM book
JOIN card_index ci ON book.book_id = ci.book_id
JOIN incoming i ON ci.card_index_id = i.card_index_id
JOIN copy c ON i.incoming_id = c.incoming_id
JOIN subscription s ON c.copy_id = s.copy_id
WHERE field_of_knowledge = 'Программирование на С#'
AND s.status = 'активный'
AND date_return < NOW() + INTERVAL '3 days'
```

Вывод данных:

| | title | author |
|---|------------------------------------|-------------|
| 1 | С# для начинающих | Джон Смит |
| 2 | С# и .NET: Подробное руководство | Марк Майкл |
| 3 | Эффективное программирование на С# | Стив Мартин |

Создание представлений

1. Сведения о должниках

Запрос:

```
CREATE VIEW debtors AS (  
  SELECT  
    r.reader_id,  
    r.reader_full_name,  
    r.address,  
    r.phone_number,  
    COUNT(s.subscription_id) overdue_subscriptions  
  FROM reader r  
    JOIN registered_reader rr on r.reader_id = rr.reader_id  
    JOIN subscription s on s.subscription_id = rr.subscription_id  
 WHERE status = 'активный'  
    AND date_real_return IS NULL  
 GROUP BY r.reader_id  
);
```

Вывод:

```
SELECT * FROM debtors;
```

| | reader_id | reader_full_name | address | phone_number | overdue_subscriptions |
|----|-----------|-----------------------------------|--------------------------------------|--------------|-----------------------|
| 1 | 9 | Александр Александр Александрович | Пермь, ул. Комсомольская, д.90 | 654321987 | 2 |
| 2 | 3 | Сидоров Сидор Сидорович | Екатеринбург, ул. Ленина, д.30 | 456789123 | 2 |
| 3 | 5 | Иванова Екатерина Игоревна | Казань, ул. Кирова, д.50 | 321654987 | 1 |
| 4 | 4 | Смирнова Анна Владимировна | Новосибирск, ул. Гоголя, д.40 | 789123456 | 2 |
| 5 | 10 | Николаева Елена Андреевна | Воронеж, ул. Ленинградская, д.100 | 987654321 | 2 |
| 6 | 6 | Попова Ольга Александровна | Ростов-на-Дону, ул. Лермонтова, д.60 | 654987321 | 2 |
| 7 | 2 | Петров Петр Петрович | Санкт-Петербург, пр. Невский, д.20 | 123456789 | 2 |
| 8 | 7 | Сергеев Сергей Сергеевич | Уфа, ул. Мира, д.70 | 987321654 | 2 |
| 9 | 1 | Иванов Иван Иванович | Москва, ул. Пушкина, д.10 | 987654321 | 2 |
| 10 | 8 | Кузнецова Анастасия Ивановна | Волгоград, ул. Победы, д.80 | 321987654 | 2 |

2. Сведения о наиболее популярных книгах (все экземпляры находятся на руках у читателей).

Запрос:

```
CREATE VIEW most_popular_books AS (  
  WITH t AS (  
    SELECT  
      b.book_id,  
      b.title,  
      b.author,  
      b.volume_number,  
      b.field_of_knowledge,  
      COUNT(subscription_id) count_copies_used  
    FROM book b  
    JOIN card_index ci ON b.book_id = ci.book_id  
    JOIN incoming i ON ci.card_index_id = i.card_index_id  
    JOIN copy c ON c.incoming_id = i.incoming_id  
    JOIN subscription s ON c.copy_id = s.copy_id  
    GROUP BY b.book_id  
    ORDER BY count_copies_used DESC, b.book_id  
  )  
  
  SELECT *  
  FROM t  
  WHERE count_copies_used = (SELECT MAX(count_copies_used) FROM t)  
);
```

Вывод:

```
SELECT * FROM most_popular_books;
```

| | book_id | title | author | volume_number | field_of_knowledge | count_copies_used |
|---|---------|--------------------------|-------------------|---------------|--------------------|-------------------|
| 1 | 1 | Война и мир | Лев Толстой | 1 | Литература | 11 |
| 2 | 2 | Война и мир | Лев Толстой | 2 | Литература | 11 |
| 3 | 3 | Преступление и наказание | Федор Достоевский | 1 | Литература | 11 |

Запросы на модификацию данных

1. Новый читатель приходит в библиотеку и берет книгу

Запрос:

```
BEGIN;

WITH new_reader AS (
  INSERT INTO reader(passport_details, reader_full_name, email, address, phone_number)
  VALUES (
    1416932535,
    'Кузнецов Кирилл Евгеньевич',
    'ChuvakKrutoi98@gmail.com',
    'Ул. Пушкина д. Колотушкина 14к2',
    89339942245
  )
  RETURNING reader_id
),
new_subscription AS (
  INSERT INTO subscription(employee_id, copy_id, date_subscribed, date_return, status)
  SELECT
    employee_id,
    39,
    NOW()::date,
    NOW()::date + INTERVAL '1 month',
    'активный'
  FROM employee
  WHERE employee_full_name = 'Сидоров Сидр Сидорович'
  RETURNING subscription_id
),
new_registered_reader AS (
  INSERT INTO registered_reader(reader_id, subscription_id, date_start, date_end)
  VALUES (
    (SELECT reader_id FROM new_reader),
    (SELECT subscription_id FROM new_subscription),
    NOW()::date,
    NOW()::date + INTERVAL '1 month'
  )
  RETURNING registered_reader_id
)
SELECT * FROM new_registered_reader;

COMMIT;
```

Вывод:

| | registered_reader_id |
|---|----------------------|
| 1 | 29 |

2. Обновим статус тех экземпляров книг, которые не активны на данный момент

Запрос:

```
WITH updated_copies AS (  
  UPDATE copy  
  SET status = 'возвращенный'  
  WHERE copy_id IN (  
    SELECT copy_id  
    FROM subscription s  
    WHERE s.status = 'неактивный'  
    AND s.date_real_return IS NOT NULL  
  )  
  RETURNING *  
)  
SELECT copy_id, status, retention_period FROM updated_copies;
```

Вывод:

| | copy_id | status | retention_period |
|----|---------|--------------|------------------|
| 1 | 2 | возвращенный | 2023-04-18 |
| 2 | 5 | возвращенный | 2023-01-10 |
| 3 | 8 | возвращенный | 2022-10-20 |
| 4 | 11 | возвращенный | 2022-07-05 |
| 5 | 14 | возвращенный | 2022-04-15 |
| 6 | 17 | возвращенный | 2022-01-01 |
| 7 | 20 | возвращенный | 2021-10-10 |
| 8 | 23 | возвращенный | 2021-07-20 |
| 9 | 26 | возвращенный | 2021-04-05 |
| 10 | 29 | возвращенный | 2021-01-15 |

3. Удалим последний документ списания

Запрос:

```
WITH deleted_docs AS (  
  DELETE FROM document  
  WHERE document_id = (  
    SELECT MAX(document_id)  
    FROM document  
    WHERE document_type = 'Списание'  
  )  
  RETURNING *  
)
```

```
SELECT * FROM deleted_docs;
```

Вывод:

| | document_id | employee_id | copy_id | document_type |
|---|-------------|-------------|---------|---------------|
| 1 | 1 | 1 | 1 | Списание |

Сравнение запросов с использованием индексирования

1. Создание простого индекса

Запрос:

```
EXPLAIN ANALYZE SELECT * FROM employee WHERE employee_full_name = 'Титов Тимур Тимурович';
```

Создание индекса:

```
CREATE INDEX idx_employee_employee_full_name ON employee(employee_full_name);
```

Время выполнения без индекса:

```
Planning Time: 0.060 ms
Execution Time: 0.026 ms
```

С индексом:

```
Planning Time: 0.057 ms
Execution Time: 0.021 ms
```

2. Сравнение составного индекса

Запрос:

```
EXPLAIN ANALYZE
SELECT *
FROM subscription
WHERE status = 'неактивный'
AND date_real_return = '2022-08-31';
```

Создание индекса:

```
CREATE INDEX idx_subscription_status_date ON subscription(status, date_real_return);
```

Время выполнения без индекса:

```
Planning Time: 0.814 ms  
Execution Time: 0.025 ms
```

С индексом:

```
Planning Time: 0.056 ms  
Execution Time: 0.021 ms
```

Заключение:

В ходе выполнения данной лабораторной работе были реализованы запросы на выборку данных и представления к базе данных на PostgreSQL согласно индивидуальному заданию. Более того были реализованы различные запросы на модификацию данных.

Также был проведен анализ графического представления всех запросов. Созданы простые и составные индексы и проанализированы их время выполнения запросов с их использованием.