МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ  
 РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Чувашский государственный университет имени И.Н. Ульянова»

Факультет информатики и вычислительной техники

Кафедра математического и аппаратного обеспечения   
 информационных систем

Лабораторная работа №6 по дисциплине “Безопасность систем баз данных”

Тема: **Индексы**

Выполнил: студент гр. ИВТ 21-22

Иванов Н.А. \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Проверил Старший преподаватель

Марков А.В. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Цель работы:

Изучить способы создания индексов БД PostgreSQL, создать индексы согласно заданию.

Содержание:

CREATE INDEX создаёт индексы по указанному столбцу(ам) заданного отношения, которым может быть таблица или материализованное представление. Индексы применяются в первую очередь для оптимизации производительности базы данных (хотя при неправильном использовании возможен и противоположный эффект).

Ключевое поле для индекса задаётся как имя столбца или выражение, заключённое в скобки. Если метод индекса поддерживает составные индексы, допускается указание нескольких полей.

Пример — индекс по полю dt в таблице payments:

CREATE INDEX payments\_dt\_idx ON payments(dt);

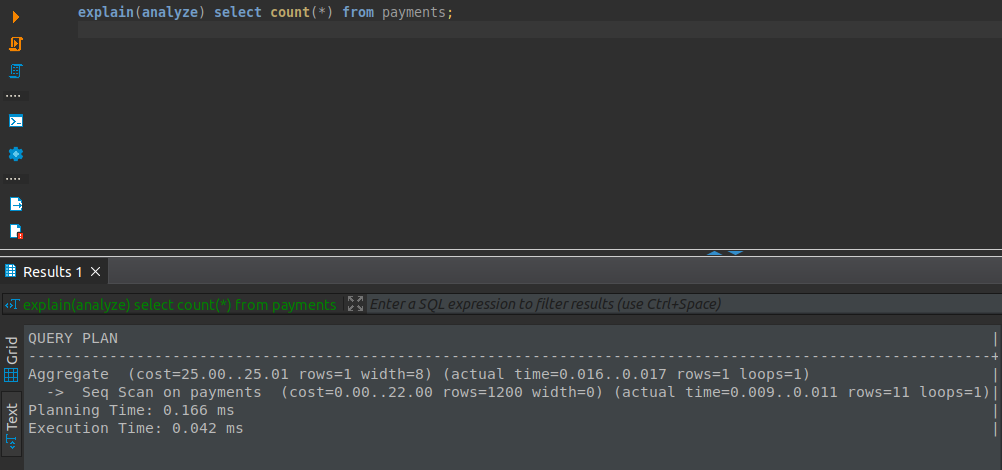
Частичный индекс:

CREATE INDEX accounts\_id\_status\_idx ON accounts(id\_account\_from) WHERE dt >= '2022-01-01';

DROP TRIGGER удаляет индекс.

Дополнительная информация по созданию индексов доступна на  
<https://postgrespro.ru/docs/postgresql/15/indexes>, <https://postgrespro.ru/docs/postgresql/15/sql-createindex>.

Задание:

1. Создать скрипт fill\_payments.sql, который создаёт 100000 записей в таблице payments (использовать функции из предыдущих лабораторных работ).
2. Создать скрипт find\_payments.sql, который содержит запрос, который выводит счёт, на который поступило больше всего денег по платежам в 2022 году.
3. Добавить в запрос в скрипте find\_payments.sql (перед текстом запроса, как указано на рисунке) explain(analyze), запустить скрипт и посмотреть план запроса (для удобства нажать переключатель Text в левом нижнем углу), а затем скопировать текст плана запросса в сервис https://explain.tensor.ru/ и проанализировать результат.
4. Создать и запустить скрипт create\_indexes.sql, содержащий следующие индексы:  
   - индекс по id статуса в таблице accounts;  
   - частичный индекс по label в таблице pages, позволяющий ускорить поиск среди записей, у которых нет родительских записей  
   - индексы, позволяющие ускорить запрос из п.3.
5. Запустить и проверить результат работы скрипта fill\_payments.sql после создания индексов из п.4 — посмотреть, как изменился план запроса, скорость запроса и аналитика на <https://explain.tensor.ru/>.
6. Создать и запустить скрипт удаления индексов drop\_indexes.sql.

Результат:

<https://github.com/bwl-andromeda/db/tree/lab-6>

Вывод:

В ходе лабораторной работы были изучены способы создания и использования индексов в PostgreSQL. Были рассмотрены основные виды индексов, такие как стандартные индексы и частичные индексы, которые позволяют улучшить производительность запросов, особенно в случае работы с большим объёмом данных. Также были исследованы составные индексы и их применение для оптимизации сложных запросов.

В рамках задания были созданы несколько индексов на таблицы базы данных. В частности:

Индекс по полю статуса в таблице accounts.

Частичный индекс по полю label в таблице pages для ускорения поиска записей без родительских элементов.

Индексы, ускоряющие выполнение запроса на вывод счёта, который получил наибольшие суммы по платежам за 2022 год.

Для проверки эффективности индексов был использован скрипт fill\_payments.sql, который добавил 100000 записей в таблицу payments. Запрос был проанализирован до и после создания индексов с использованием утилиты EXPLAIN ANALYZE, а результаты анализа были загружены и проанализированы на сервисе https://explain.tensor.ru/. Анализ показал, что после создания индексов скорость выполнения запросов значительно улучшилась, что подтверждает важность правильного использования индексов для оптимизации работы базы данных.

Также были выполнены шаги по удалению созданных индексов с помощью скрипта drop\_indexes.sql.

Таким образом, цель работы была достигнута — были получены практические навыки создания индексов в PostgreSQL и их использования для повышения производительности запросов. Эксперименты продемонстрировали, что правильное применение индексов может значительно ускорить выполнение сложных SQL-запросов в условиях работы с большими объёмами данных.