**Лабораторная работа на тему «Оптимизация запросов».**

Для выполнения лабораторной работы понадобится демонстрационная БД : <https://edu.postgrespro.ru/demo-medium.zip>

Структура бд: https://postgrespro.ru/docs/postgrespro/15/demodb-schema-diagram

**Подсказки / рекомендации:**

**Именование индексов**

В процессе выполнения практики нужно будет создавать индексы. Индексам можно давать любые название, однако есть правила хорошего тона в именовании индексов.

Лучше начинать индексы с префикса ix\_ или idx\_, чтобы отличать их от таблиц. После префикса пишется название таблицы, а затем столбец или краткое название индекса.

Например, индекс на поле user\_id в таблице order можно назвать ix\_order\_user.

**Выполнение задания**

В ответе на задание должны содержаться первоначальный план запроса, запрос на создание индекса, а также план запроса после оптимизации. Если требуется обоснование или предположение, то оно может быть написано в свободной форме.

**Параллельное выполнение**

Если в процессе анализа плана запроса PostgreSQL начинает использовать параллельное выполнение (появляются строчки Gather, Workers Planned: 2, Workers Launched: 2), можно отключить параллельное выполнение командой:

SET max\_parallel\_workers\_per\_gather = 0;

Для просмотра количества воркеров для параллельного выполнения используется команда:

SHOW max\_parallel\_workers\_per\_gather;

Если параллельное выполнение не мешает анализу плана запроса, то можно его не отключать (отключить можно для упрощения понимания плана запроса).

**Задания:**

1. Выбрать любой рейс (flight\_id из таблицы flights). Дан запрос для получения всех данных по перелётам (ticket\_flights) по выбранному рейсу:

SELECT \* FROM ticket\_flights WHERE flight\_id=выбранный\_рейс;

Проанализировать план запроса и оптимизировать его.

2. Для отображения данных в табло задержанных рейсов в каждом аэропорту прибытия (arrival\_airport) используются данные из таблицы полётов (flights):

SELECT \* FROM flights WHERE status='Delayed' AND arrival\_airport=аэропорт;

Проанализировать запрос для любого из аэропортов и оптимизировать его (подсказка: нужно использовать частичные индексы). Проанализировать запрос после добавления индекса. Проанализировать запросы для отображения всех рейсов данного аэропорта, а также для отображения всех задержанных рейсов по всем аэропортам. Использует ли планировщик созданный индекс в этих запросах? Предположить причины такого поведения.

3. В системе продажи билетов необходимо часто просматривать рейсы за диапазон дат отправления (scheduled\_departure) из аэропорта отправления (departure\_airport) в аэропорт прибытия (arrival\_airport).

Написать запрос для получения этих данных.

Рассмотреть несколько вариантов оптимизации: с одним индексом, с составным индексом (из нескольких столбцов) и с несколькими индексами. Проверить на нескольких диапазонах дат и разными аэропортами. Какой из вариантов оказался самым оптимальным? Предположить, почему.

4. В системе продажи билетов иногда нужно найти билеты пассажира (tickets) по номеру телефона. Телефон хранится в json-поле contact\_data. Написать запрос для получения всей информации по номеру телефона. Оптимизировать запрос.

5. Посмотреть размер ранее созданных индексов. Предположить, какие следует оставить, а какие удалить.

Размер индексов можно посмотреть с помощью запроса Index size/usage statistics (второй запрос по ссылке):

https://wiki.postgresql.org/wiki/Index\_Maintenance

6. Придумать любой запрос, использующий несколько таблиц. Посмотреть план запроса. Предположить, можно ли его оптимизировать и как (саму оптимизацию можно не выполнять).