**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ**

**РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Казанский государственный энергетический»**

**Отчет по лабораторной работе 2**

**Последовательный доступ**

Выполнил:

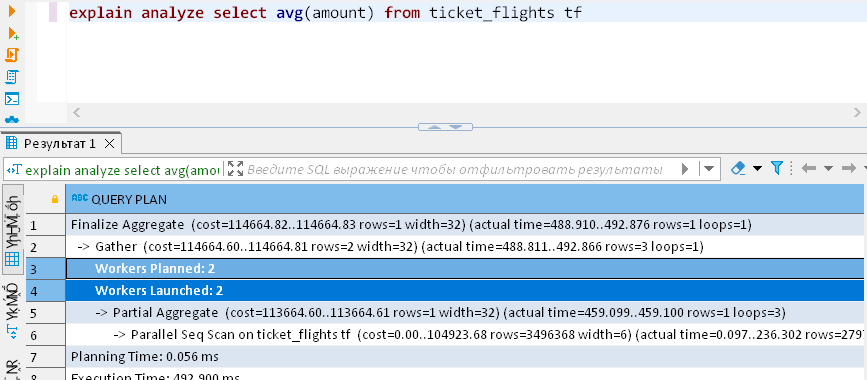
студент группы ПИ-1-22

Соловьёв Л.А.

Проверил:

Хамитов Р. М.

Казань 2024

**1) Убедитесь, что запрос, вычисляющий среднюю стоимость перелета, выполняется параллельно.  
explain** **analyze** **select** **avg**(amount) **from** ticket\_flights tf  


**Workers Planned: 2  
Workers Launched: 2**   
Это означает, что запрос выполняется параллельно

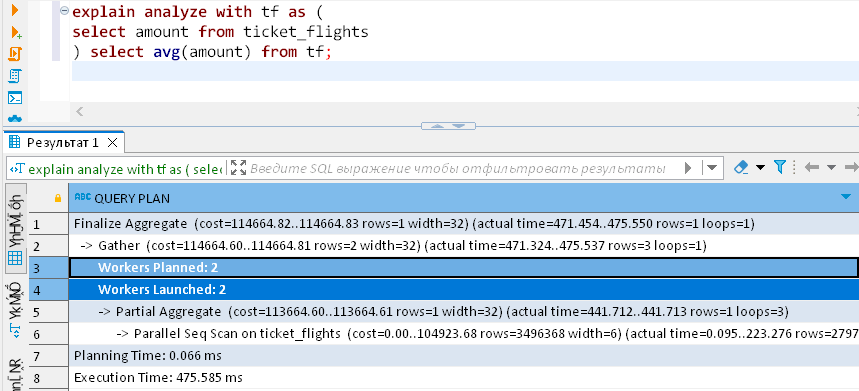
**2) Проверьте, как выполняется тот же запрос, если чтение из таблицы перелетов оформлено как общее табличное выражение. Используйте указание MATERIALIZE, чтобы планировщик не раскрывал общее табличное выражение.**

**Без использования materialized:**

**explain** **analyze** **with** tf **as** (

**select** amount **from** ticket\_flights

) **select** **avg**(amount) **from** tf;

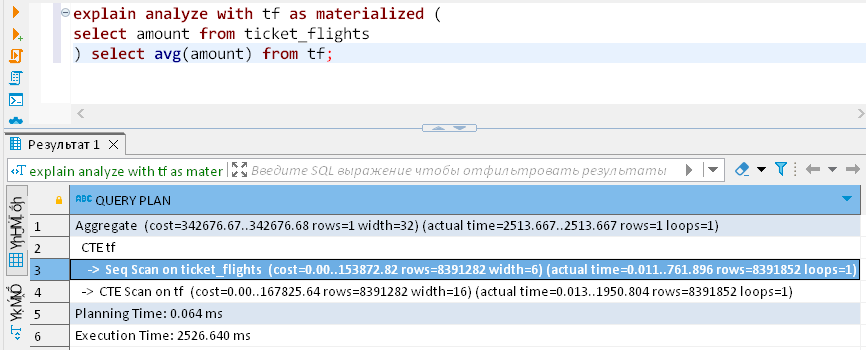


Без использования materialized параллельно выполняются два запроса: CTE и основной запрос.

**C использованием materialized:**

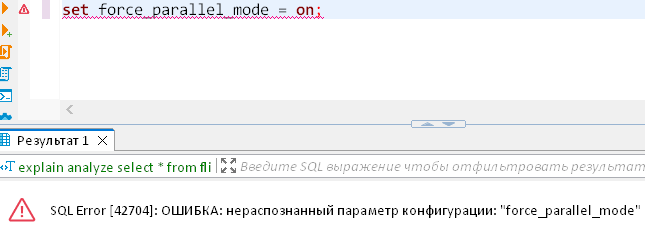
**explain** **analyze** **with** tf **as** **materialized** (

**select** amount **from** ticket\_flights

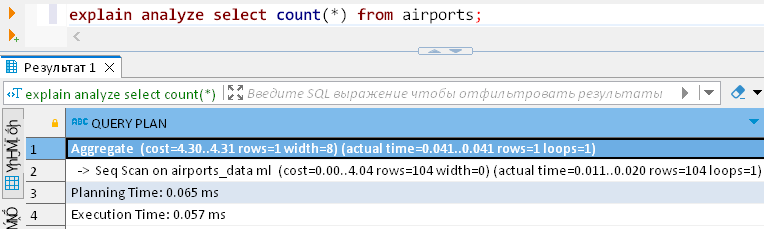
) **select** **avg**(amount) **from** tf;

С использованием materialized подзапрос выполняется после создания временной таблицы, т.е. последовательно.

Без использования materialized запрос выполняется быстрее, чем с materialized, потому что запрос выполняется параллельно.

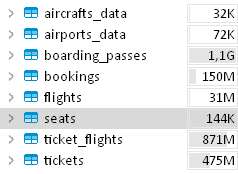
**3) В демонстрации запрос всех строк из таблицы рейсов выполнялся последовательно. Почему? Может ли такой запрос в принципе быть распараллелен или он запрещает параллельное выполнение? Воспользуйтесь параметром force\_parallel\_mode, чтобы ответить на этот вопрос.**   
Воспользоваться параметром не могу, поэтому буду использовать материалы урока.  
  


Запрос может быть распараллелен, однако планировщик будет признавать использование параллелизации на этом запросе неэффективным и использоваться будет только один процесс.  
  
 **4) Как будет выполнен запрос определяющий количество аэропортов: параллельно или последовательно и почему?**



Запрос выполняется последовательно, потому что для подсчёта количества достаточно 1 раз пройти все строки airport и получить их общее количество(rows=104). В этом запросе нет ничего, что при параллелизации может быть оптимизировано.

**5) Для каких таблиц в базе данных при подсчете количества строк таблицы будет выбран последовательный план и почему?**

Для любой таблицы меньше 8MB не будет рассматриваться параллельный план, потому что существует параметр min\_parallel\_table\_scan\_size, который по умолчанию равен 8MB.  
  
Т.е. таблицы aircrafts\_data(32K), airports\_data(72K), seats(144K) при подсчёте строк 100% будут выбирать последовательный план, что подтверждается при использовании команды EXPLAIN. Однако, даже если размер таблицы больше значения по умолчанию, не обязательно будет использован параллельный план, например, в таблице flights(31MB) он использоваться не будет(Workers Planned: 1, Workers Launched: 1), потому что размер таблицы не такой большой и используется простая команда COUNT, которую неэффективно разделять на несколько процессов. При подсчёте строк в bookings(150M) уже используется параллелизация, следовательно при бОльших объёмах(tickets(475M), ticket\_flights(871M), boarding\_passes(1.1G)) так же будет использовано распараллеливание. Ниже приведены результаты команды EXPLAIN.  
