**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ**

**РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Казанский государственный энергетический»**

**Отчет по лабораторной работе 4**

**Сканирование по битовой карте**

Выполнил:

студент группы ПИ-1-22

Соловьёв Л.А.

Проверил:

Хамитов Р. М.

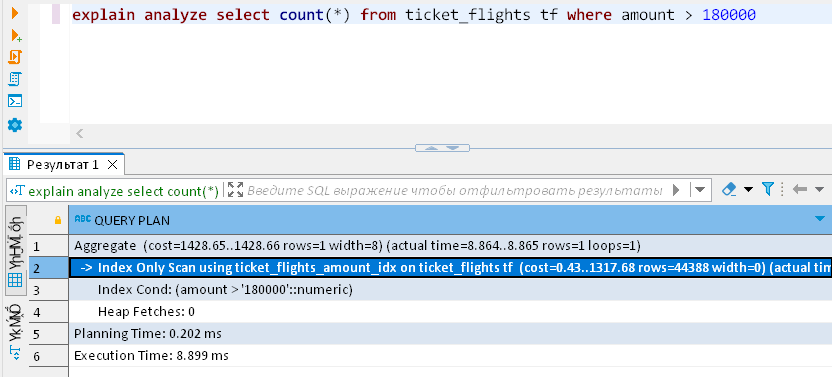
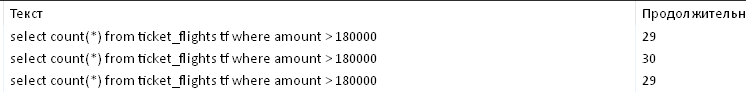
Казань 2024

**1. Создайте индекс по столбцу amount таблицы перелетов (ticket\_flights).**

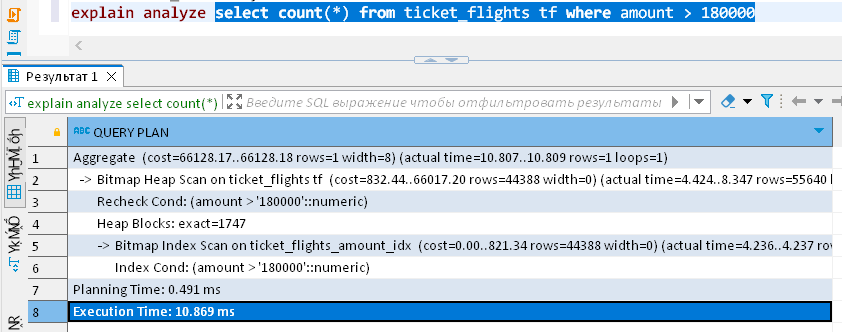
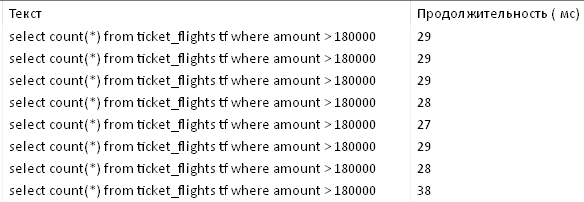
**create** **index** **on** ticket\_flights(amount);

**2. Напишите запрос, находящий количество перелетов стоимостью более 180 000 руб. (менее 1% строк).**

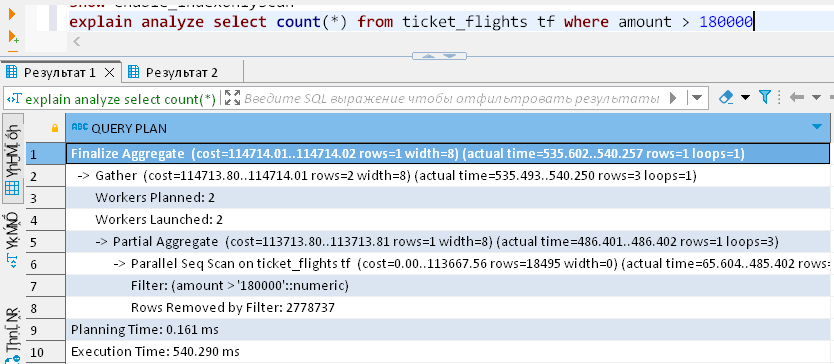
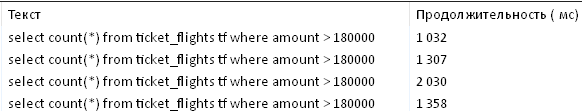
**Какой метод доступа был выбран? Сколько времени выполняется запрос?**

**select** **count**(\*) **from** ticket\_flights tf **where** amount > 180000; ****Планировщиком было выбрано сканирование только по индексу, запрос выполняется в среднем ~30 миллисекунд

**3. Запретите выбранный метод доступа, снова выполните запрос и сравните время выполнения.  Прав ли был оптимизатор?  
set** enable\_indexonlyscan = **off;**

****В этот раз планировщик выбрал сканирование по битовой карте. Среднее время почти не изменилось.  


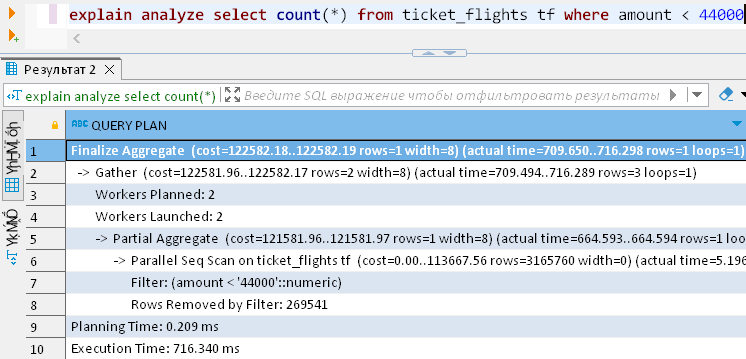
**set** enable\_bitmapscan = **off**

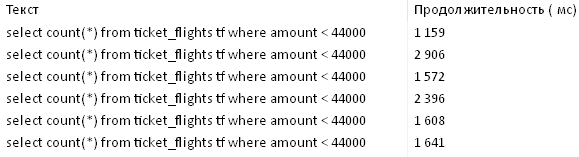
  
После блокировки сканирования по битовой карте, планировщик выбрал параллельное последовательное сканирование. Время выполнения запроса существенно изменилось.  


Планировщик выбирает наиболее оптимальное решение, следовательно для маленькой выборки сканирование по битовой карте и индексное сканировании наиболее эффективны.

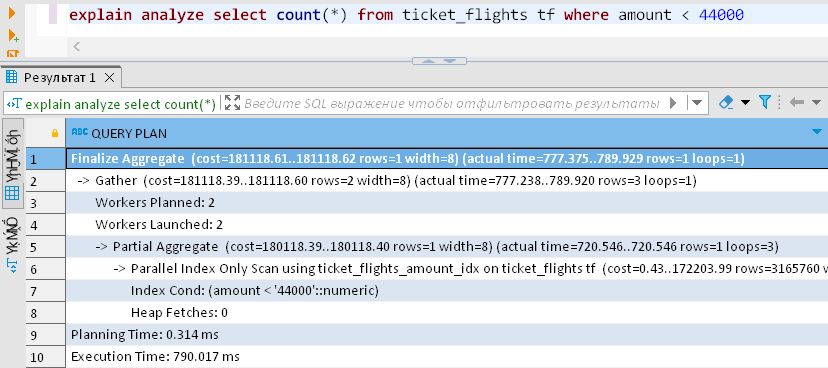
**4. Повторите пункт 2 для стоимости менее 44 000 руб. (чуть более 90% строк).  Какое сканирование выгоднее?**

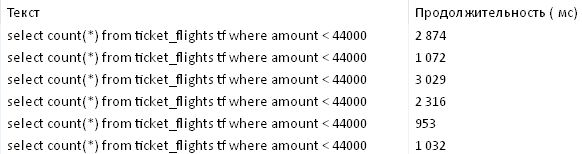
**explain** **analyze** **select** **count**(\*) **from** ticket\_flights tf **where** amount < 44000

****

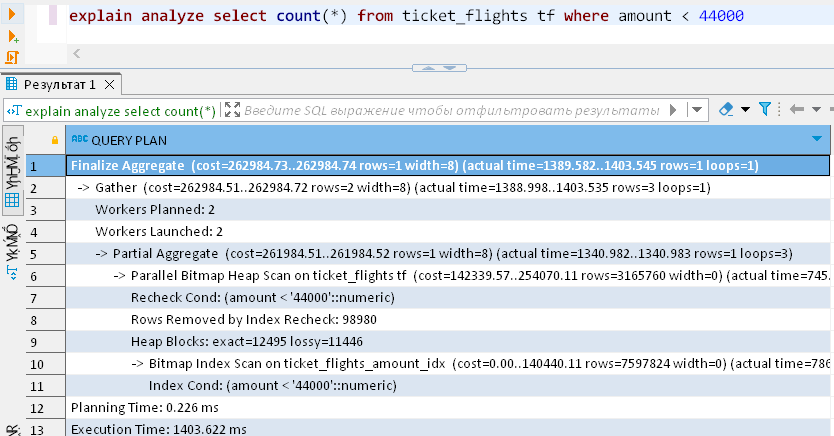
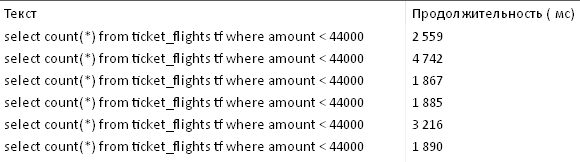
Среднее время выполнения запроса ~1.8 секунд  


Изначально планировщик выбрал параллельное последовательное сканирование.  
После запрета последовательного сканирования он выбрал параллельное сканирование по индексу.

****

Среднее время выполнения запроса ~1.8 секунд

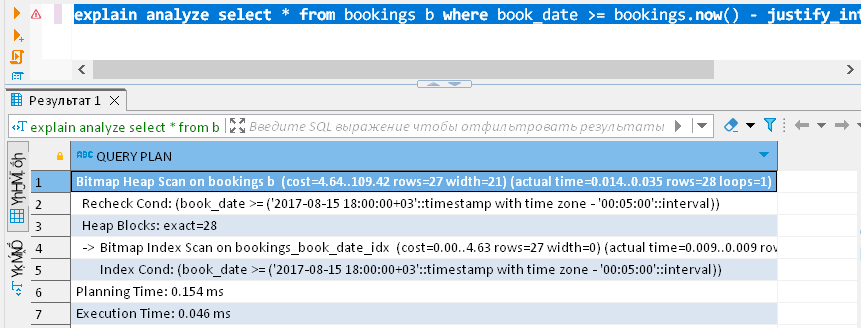
После запрета индексного сканирования планировщик выбирает сканирование по битовой карте.

  
Среднее время выполнения запроса ~2.7 секунд  


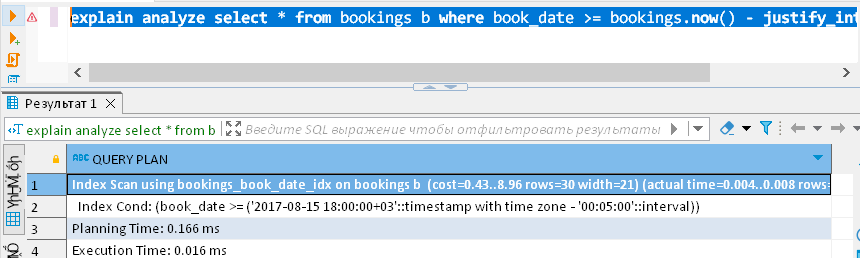
Для больших выборок наиболее эффективны последовательное и индексное сканирование.

**5. В таблице бронирований bookings  создайте индекс по дате бронирования. Найдите бронирования сделанные за последние 5 минут. Какой метод сканирования был выбран? Выполните кластеризацию по индексу bookings\_book\_date\_idx. Выполните сбор статистики о базе данных командой ANALYZE. Повторите запрос. Какой метод сканирования  используется теперь? Почему и надолго ли? Функция bookings.now()  возвращает текущий момент времени в демо базе.**

**create** **index** **on** bookings(book\_date);  
**explain** **analyze** **select** \* **from** bookings b **where** book\_date >= bookings.**now**() - **justify\_interval**(**'5 MINUTE'**);

Планировщик выбрал сканирование по битовой карте.  
  
**cluster** bookings **using** bookings\_book\_date\_idx;

**analyze** bookings;



Теперь планировщик выбрал индексное сканирование, потому что строки упорядочены по возрастанию даты, однако со временем в процессе работы кластеризация будет ухудшаться, потому что строки не поддерживаются в упорядоченном виде.  
Упорядоченные строки по book\_date:

