Оптимизация запросов

В начало запроса поместите ехргаім [амагуzе]



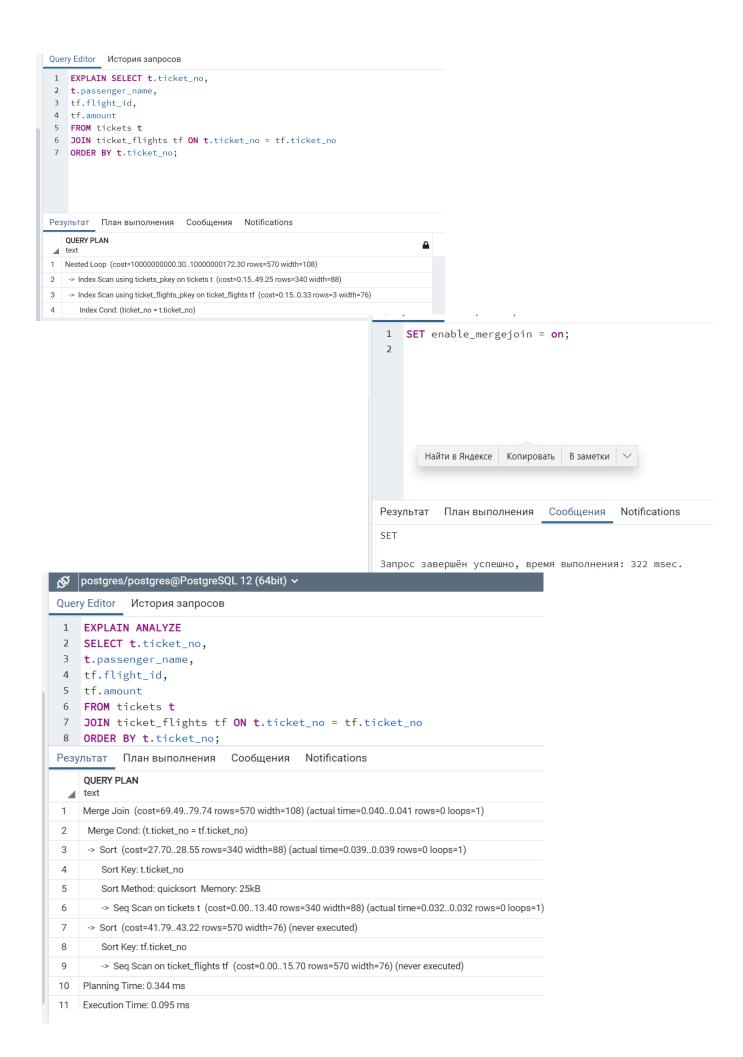
В ответе получен план выполнения запроса. Поскольку в этом запросе нет предложения WHERE, он должен просмотреть все строки таблицы, поэтому планировщик выбирает последовательный просмотр. С параметром ANALYZE оператор будет выполнен на самом деле, а не только запланирован. При этом в вывод добавляются фактические сведения о времени выполнения, включая общее время, затраченное на каждый узел плана (в миллисекундах) и общее число строк, выданных в результате.

Напишите запрос на вывод всей информации из любой имеющейся у вас таблицы добавив условие отбора. В начало запроса поместите EXPLAIN [ANALYZE]

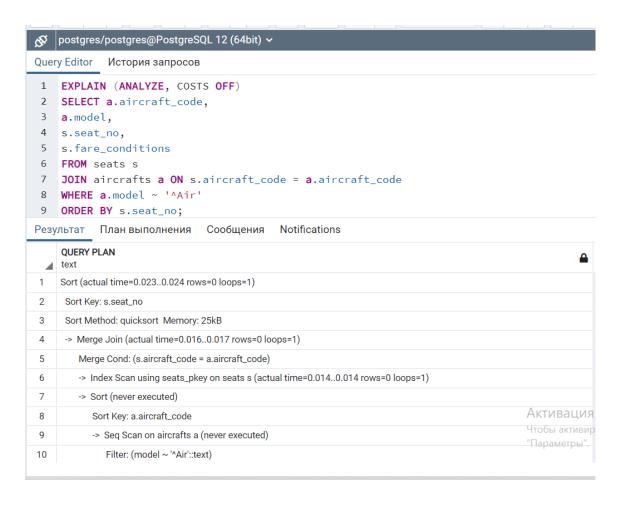


Поскольку наложено дополнительное условие на строки, выбираемые из таблицы, то ниже узла плана, отвечающего за их последовательную выборку, добавляется еще один узел, описывающий критерий отбора строк.

```
SET enable_hashjoin = off;
 Результат
           План выполнения Сообщения
                                         Notifications
 SET
 Запрос завершён успешно, время выполнения: 298 msec.
                                               Query Editor
                                                           История запросов
                                                   SET enable_mergejoin = off;
                                               Результат
                                                         План выполнения
                                                                           Сообщения
                                                                                       Notifications
                                               SET
                                               Запрос завершён успешно, время выполнения: 338 msec.
    SET enable_nestloop = off;
 1
 2
Результат
          План выполнения Сообщения
                                          Notifications
SET
Запрос завершён успешно, время выполнения: 314 msec.
                                              1
                                                 SET enable_mergejoin = off;
                                              2
                                             Результат
                                                        План выполнения
                                                                                       Notifications
                                                                          Сообщения
                                             SET
                                             Запрос завершён успешно, время выполнения: 214 msec.
```



```
Query Editor История запросов
2 SELECT t.ticket_no,
3 t.passenger_name,
 4 tf.flight_id,
 5 tf.amount
 6 FROM tickets t
 7 JOIN ticket_flights tf ON t.ticket_no = tf.ticket_no
 8 WHERE amount > 50000
 9 ORDER BY t.ticket_no;
Результат План выполнения Сообщения Notifications
    QUERY PLAN
    Merge Join (cost=52.01..56.56 rows=190 width=108) (actual time=0.063..0.064 rows=0 loops=1)
1
2
     Merge Cond: (t.ticket_no = tf.ticket_no)
     -> Sort (cost=27.70..28.55 rows=340 width=88) (actual time=0.061..0.062 rows=0 loops=1)
3
4
        Sort Key: t.ticket_no
        Sort Method: quicksort Memory: 25kB
5
6
        -> Seq Scan on tickets t (cost=0.00..13.40 rows=340 width=88) (actual time=0.040..0.040 rows=0 loops=1)
7
     -> Sort (cost=24.32..24.79 rows=190 width=76) (never executed)
8
        Sort Key: tf.ticket_no
                                                                                                  Активация Windows
9
        -> Seq Scan on ticket_flights tf (cost=0.00..17.13 rows=190 width=76) (never executed)
10
           Filter: (amount > '50000'::numeric)
11 Planning Time: 1.033 ms
```



```
SQL Shell (psql)

екущая кодовая страница: 1251
laponь пользователя postgres:
sql (12.5)
ведите "help", чтобы получить справку.

costgres=# BEGIN;
leGIN
costgres=# EXPLAIN (ANALYZE, COSTS OFF)
costgres=# UPDATE aircrafts
costgres=# SET range = range + 100
costgres=# WHERE model ~ '^Air';
QUERY PLAN

Update on aircrafts (actual time=7.397..7.398 rows=0 loops=1)
-> Seq Scan on aircrafts (actual time=0.170..0.206 rows=3 loops=1)
Filter: (model ~ '^Air':text)
Rows Removed by Filter: 9

Planning Time: 2.178 ms
Execution Time: 14.978 ms
6 строк)

costgres=# ROLLBACK;

costgres=# ROLLBACK;
```