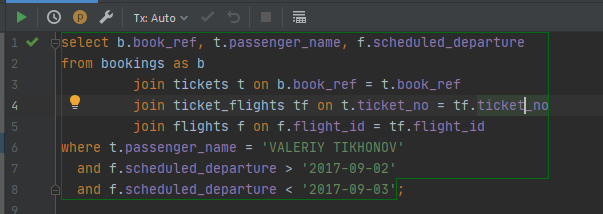
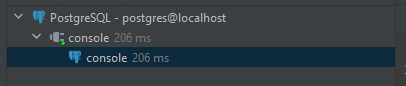
12.7. Упражнения

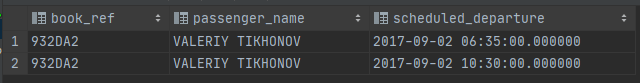
Упражнение 12.1. В демонстрационной базе данных постройте индексы для поиска бронирований по имени пассажира и интервалу дат вылета.

Запрос до создания индексов

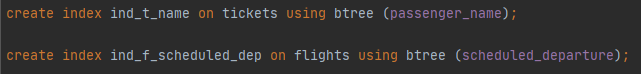




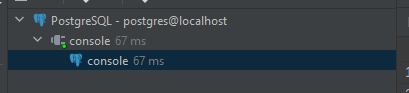
Найденные записи в базе данных



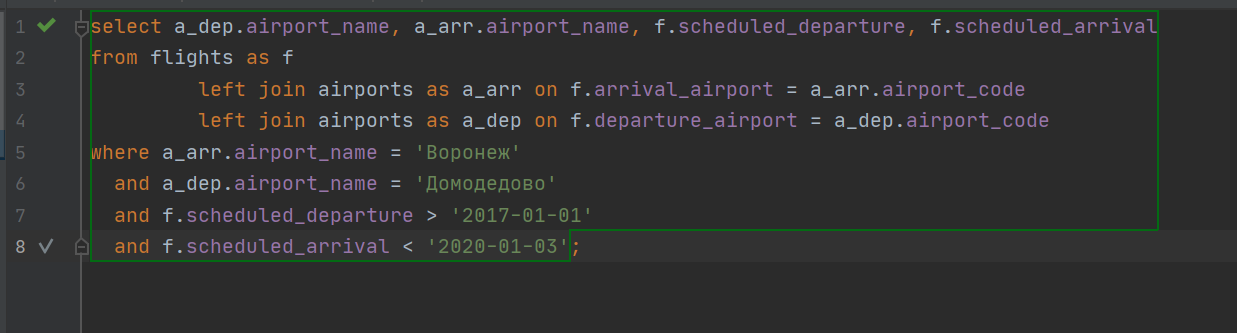
Создание индексов

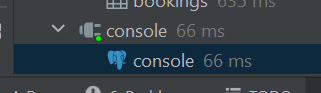


Выполнение запроса после создания индексов

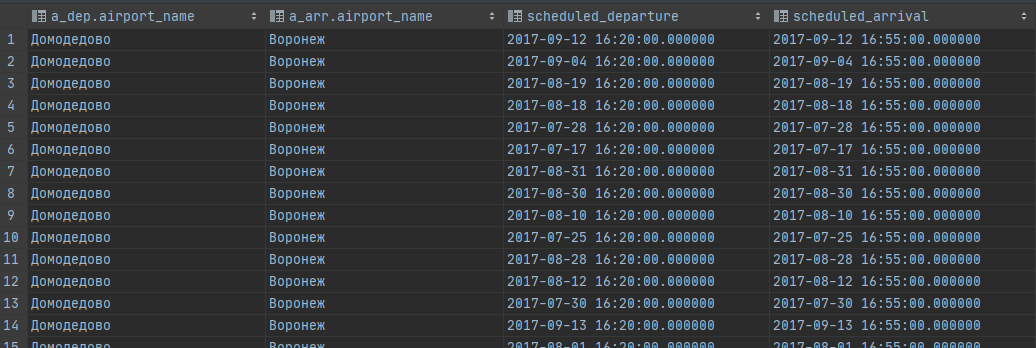


Упражнение 12.2. В демонстрационной базе данных постройте индексы для поиска вариантов перелета по названию пунктов отправления и прибытия и дате вылета.

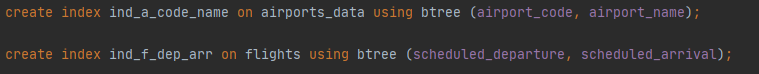
Выполнение запроса без индексов 



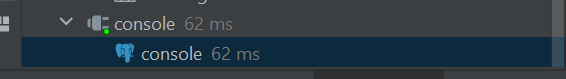
Найденные записи в базе данных



Добавление индексов

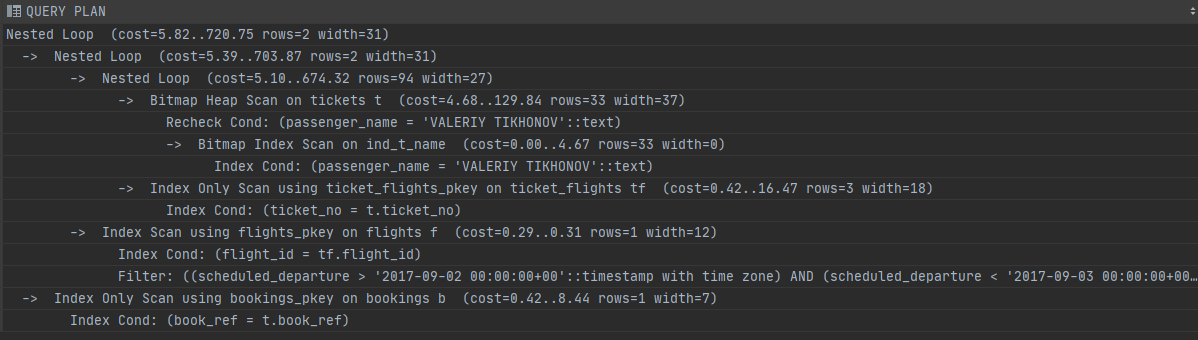


Выполнение запросов после добавления индексов

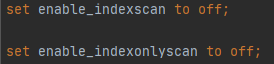


Упражнение 12.3. В контексте предыдущих упражнений отключите использование индексов и сравните планы выполнения запросов с индексами и без них.

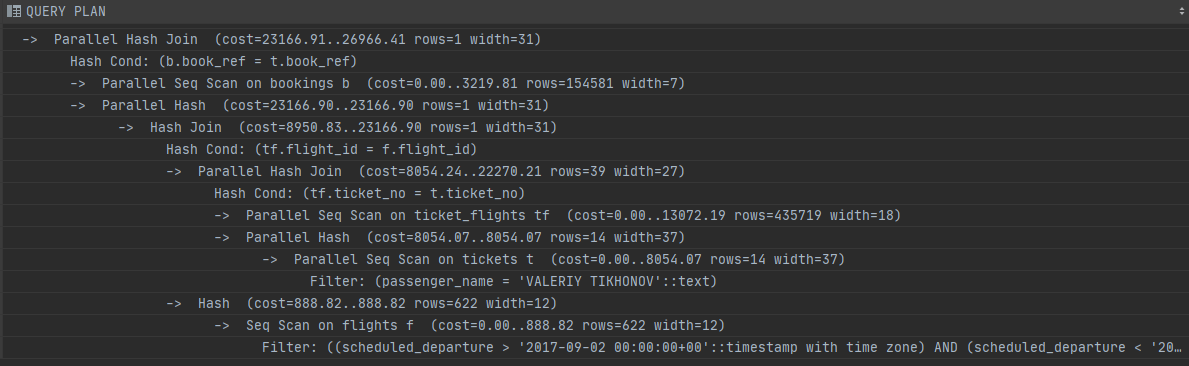
Логический план выполнения для первого запроса



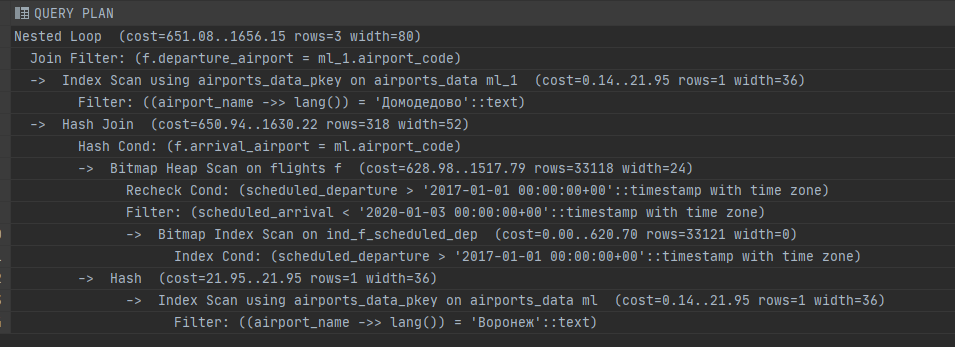
Отключение индексов



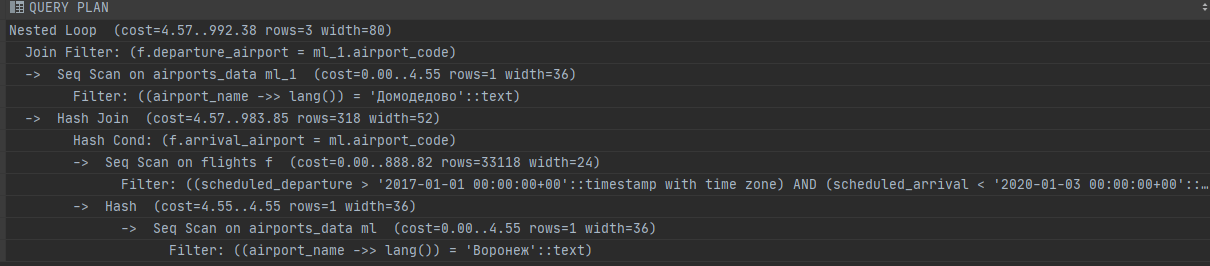
Логический план выполнения для первого запроса после отключения индексов



Логический план выполнения для второго запроса



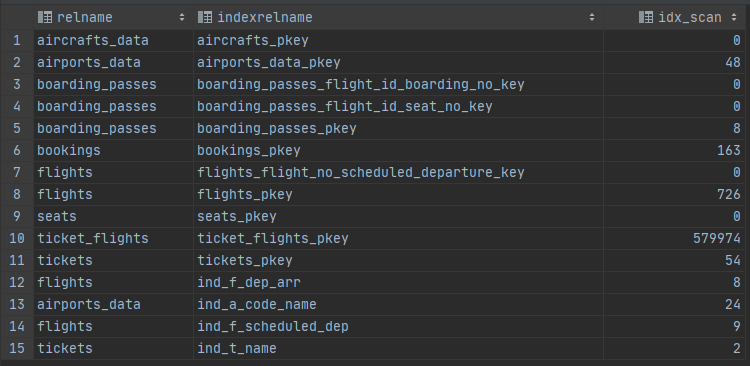
Логический план выполнения для первого запроса после отключения индексов



Упражнение 12.4. Найдите в демонстрационной базе данных статистическую информацию.



Вывод

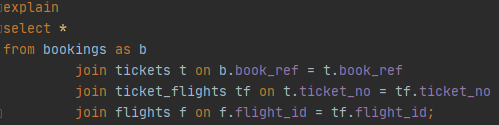


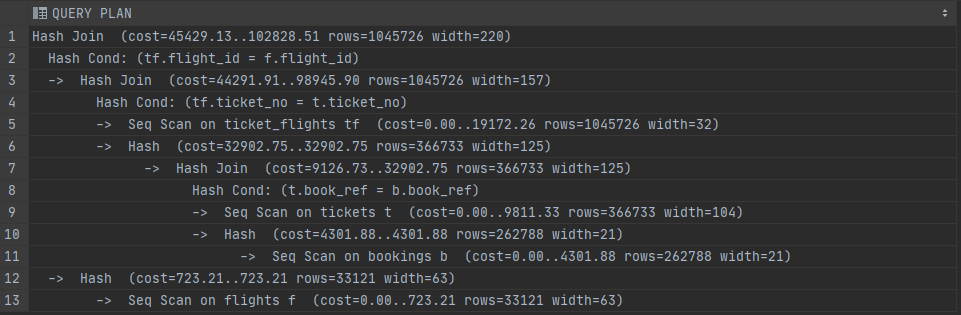
Упражнение 12.5. Установите параметры оптимизатора таким образом, чтобы изменение порядка операций соединения никогда не производилось. Получите разные планы выполнения запросов, содержащих 3, 5, 8 и 13операций соединения, изменяя порядок соединений вручную. Изучите и сравните оценки стоимости для различных порядков операций соединения каждого запроса.

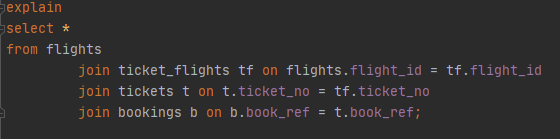
Изменение порядка запросов не будет производиться после выполнения следующей команды

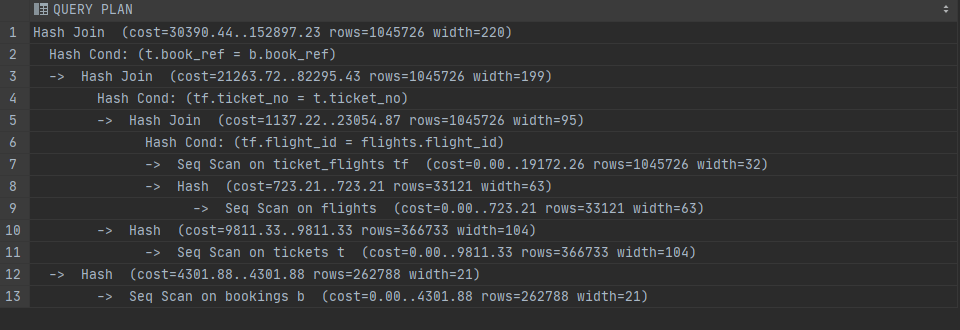


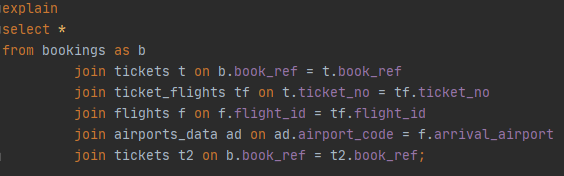
Запросы и планы их выполнения

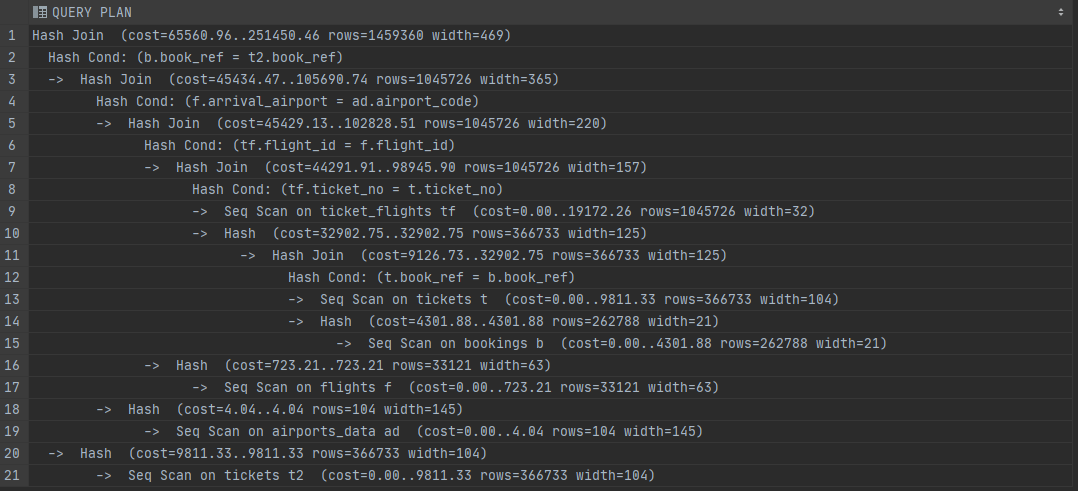


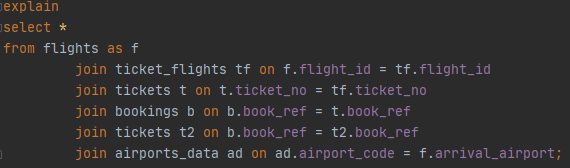


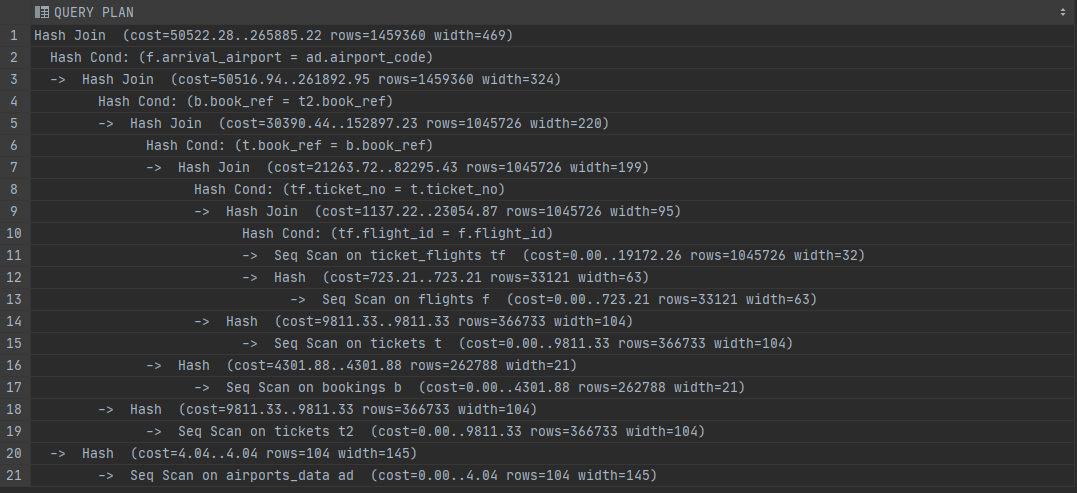






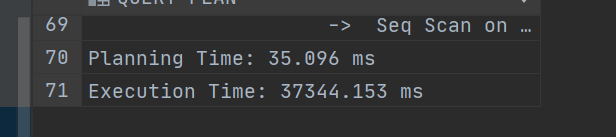




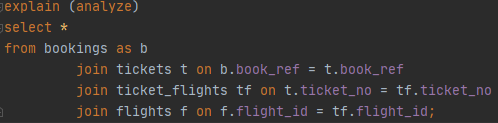


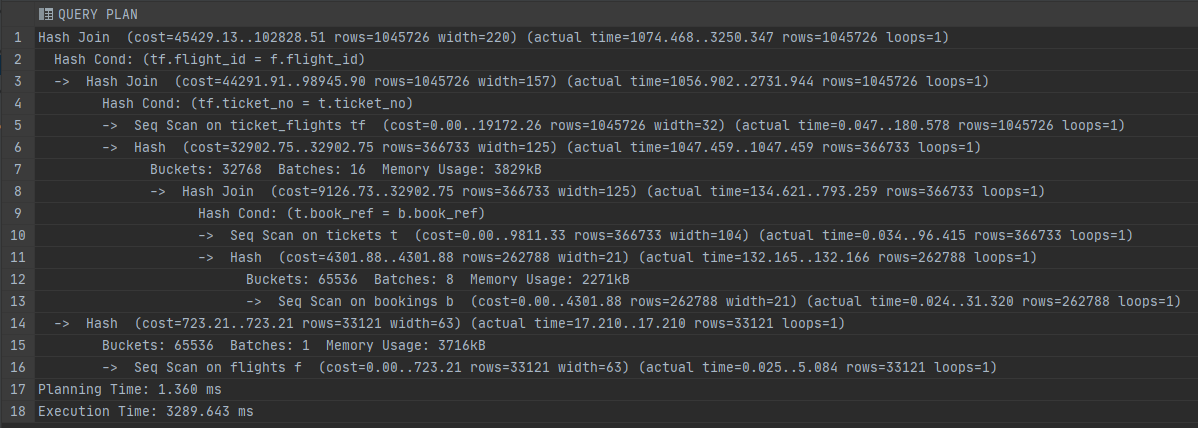
Аналогично были выполнены построения планов с 8 и 13 операциями соединения, в отчет они не включены из-за большого объема и нечитаемости планов на скриншотах.

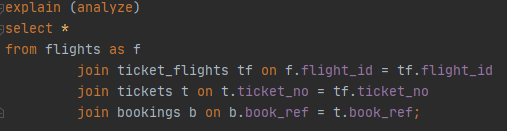
Статистика запроса с 13 операциями соединения:

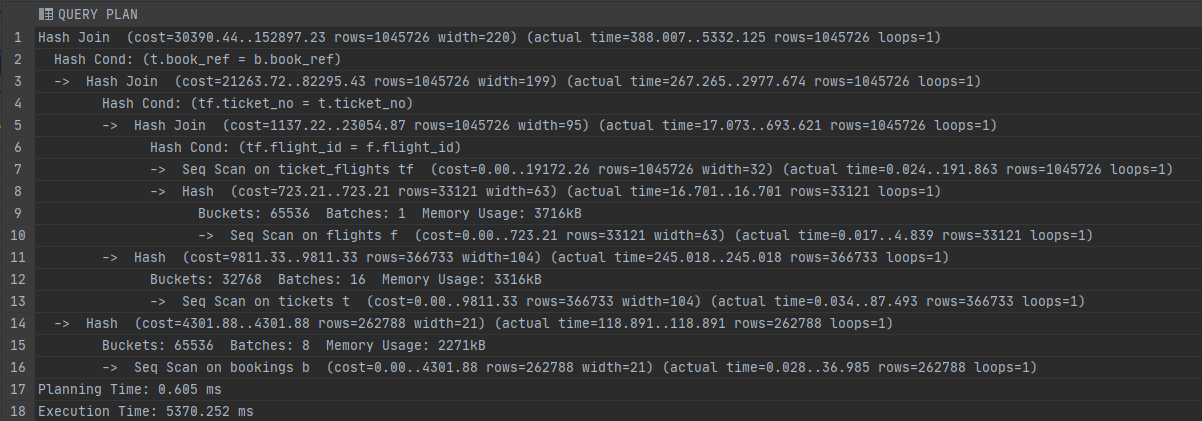


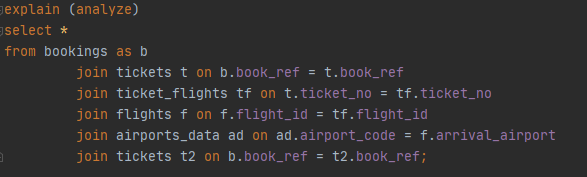
Упражнение 12.6. В контексте предыдущего упражнения получите фактические размеры промежуточных результатов и сравните их с оценками, используемыми оптимизатором.



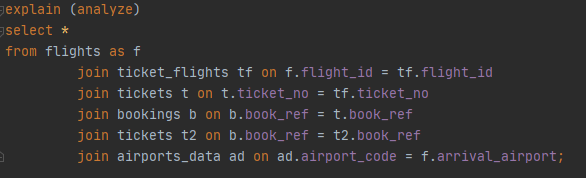














# Полный перечень всех использованных запросов

-- лр 3

-- 1

select b.book\_ref, t.passenger\_name, f.scheduled\_departure

from bookings as b

join tickets t on b.book\_ref = t.book\_ref

join ticket\_flights tf on t.ticket\_no = tf.ticket\_no

join flights f on f.flight\_id = tf.flight\_id

where t.passenger\_name = 'VALERIY TIKHONOV'

and f.scheduled\_departure > '2017-09-02'

and f.scheduled\_departure < '2017-09-03';

create index ind\_t\_name on tickets using btree (passenger\_name);

create index ind\_f\_scheduled\_dep on flights using btree (scheduled\_departure);

-- 2

select a\_dep.airport\_name, a\_arr.airport\_name, f.scheduled\_departure, f.scheduled\_arrival

from flights as f

left join airports as a\_arr on f.arrival\_airport = a\_arr.airport\_code

left join airports as a\_dep on f.departure\_airport = a\_dep.airport\_code

where a\_arr.airport\_name = 'Воронеж'

and a\_dep.airport\_name = 'Домодедово'

and f.scheduled\_departure > '2017-01-01'

and f.scheduled\_arrival < '2020-01-03';

create index ind\_a\_code\_name on airports\_data using btree (airport\_code, airport\_name);

create index ind\_f\_dep\_arr on flights using btree (scheduled\_departure, scheduled\_arrival);

-- 3

explain

select b.book\_ref, t.passenger\_name, f.scheduled\_departure

from bookings as b

join tickets t on b.book\_ref = t.book\_ref

join ticket\_flights tf on t.ticket\_no = tf.ticket\_no

join flights f on f.flight\_id = tf.flight\_id

where t.passenger\_name = 'VALERIY TIKHONOV'

and f.scheduled\_departure > '2017-09-02'

and f.scheduled\_departure < '2017-09-03';

set enable\_indexscan to off;

set enable\_indexonlyscan to off;

set enable\_bitmapscan to off;

set enable\_indexscan to on;

set enable\_indexonlyscan to on;

set enable\_bitmapscan to on;

explain

select a\_dep.airport\_name, a\_arr.airport\_name, f.scheduled\_departure, f.scheduled\_arrival

from flights as f

left join airports as a\_arr on f.arrival\_airport = a\_arr.airport\_code

left join airports as a\_dep on f.departure\_airport = a\_dep.airport\_code

where a\_arr.airport\_name = 'Воронеж'

and a\_dep.airport\_name = 'Домодедово'

and f.scheduled\_departure > '2017-01-01'

and f.scheduled\_arrival < '2020-01-03';

-- set enable\_seqscan to on;

-- 4

SELECT relname, indexrelname, idx\_scan

FROM pg\_catalog.pg\_stat\_user\_indexes

WHERE schemaname = 'bookings';

-- 5, 6

set join\_collapse\_limit to 1;

explain (analyze)

select \*

from bookings as b

join tickets t on b.book\_ref = t.book\_ref

join ticket\_flights tf on t.ticket\_no = tf.ticket\_no

join flights f on f.flight\_id = tf.flight\_id

join airports\_data ad on ad.airport\_code = f.arrival\_airport

join tickets t2 on b.book\_ref = t2.book\_ref;

explain (analyze)

select \*

from flights as f

join ticket\_flights tf on f.flight\_id = tf.flight\_id

join tickets t on t.ticket\_no = tf.ticket\_no

join bookings b on b.book\_ref = t.book\_ref

join tickets t2 on b.book\_ref = t2.book\_ref

join airports\_data ad on ad.airport\_code = f.arrival\_airport;

set enable\_bitmapscan to on;

set enable\_hashagg to on;

set enable\_hashjoin to on;

set enable\_indexscan to on;

set enable\_indexonlyscan to on;

set enable\_material to on;

set enable\_mergejoin to on;

set enable\_nestloop to on;

set enable\_seqscan to on;

set enable\_sort to on;

set enable\_tidscan to on;