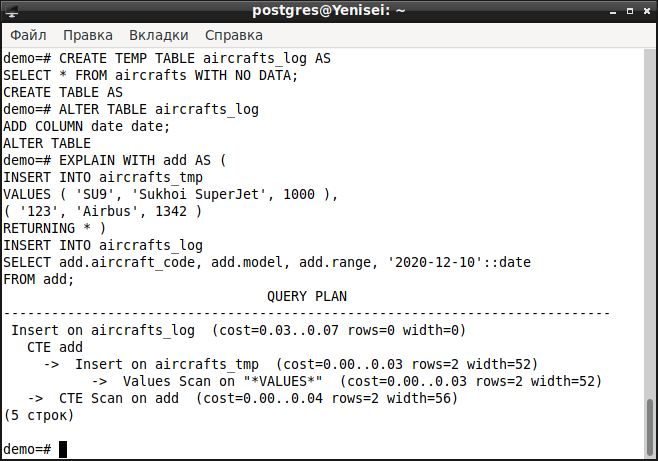
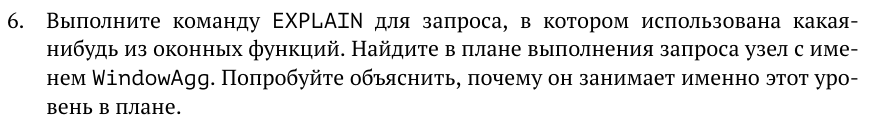
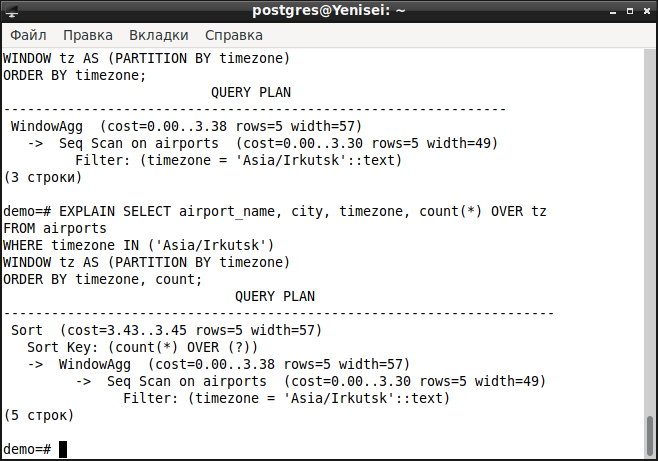


*Предварительно создадим необходимые таблицы и выполним команду EXPLAIN для CTE-запроса.*

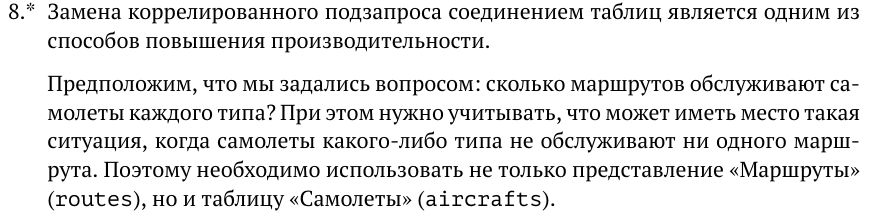
**

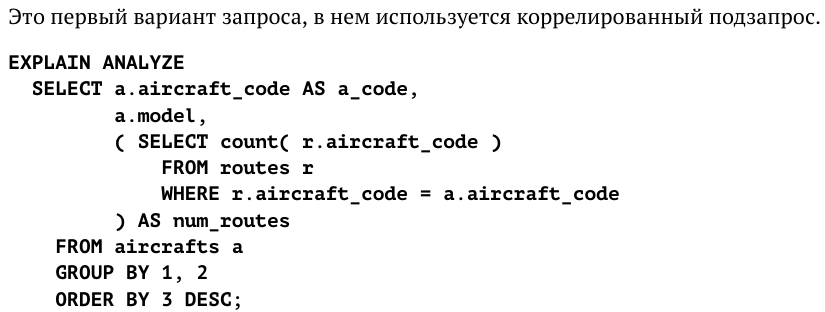
*CTE-запрос хранится на внешнем узле (CTE add). При этом сначала выполняется сканирование добавляемых значений VALUES, после эти данные добавляются в таблицу aircrafts\_tmp и материализуется общее табличное выражение с названием add. Далее на самом верхнем уровне данные из табличного выражения используются для добавления данных в таблицу aircrafts\_log (сканируются (CTE Scan) и добавляются в таблицу).*

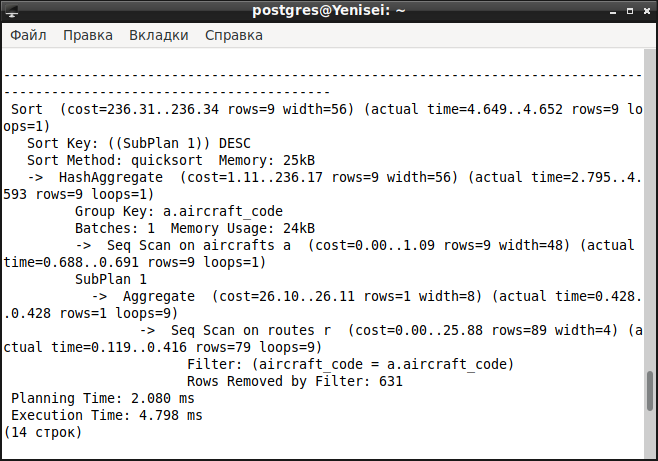
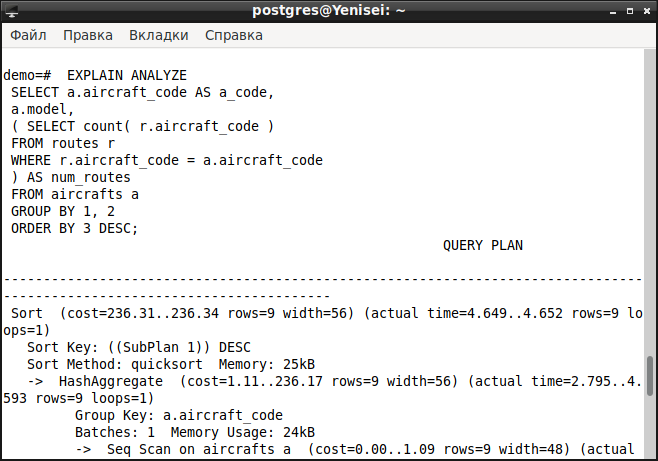
**

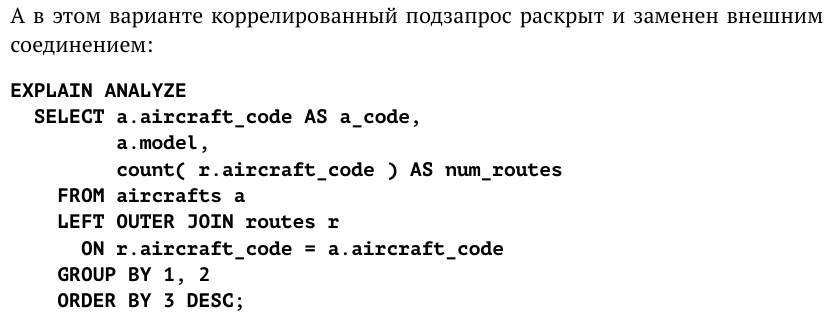
**

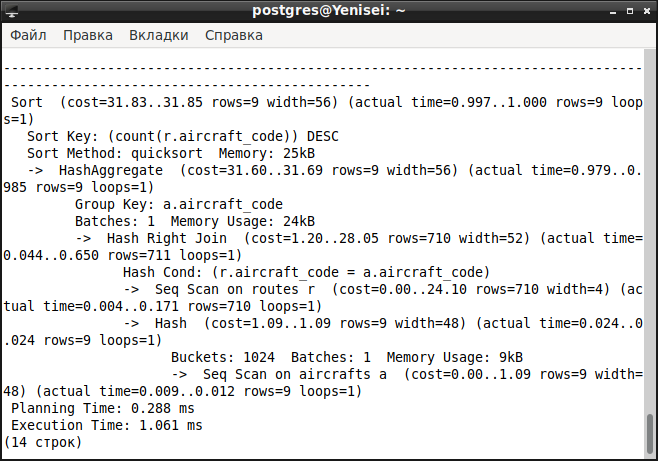
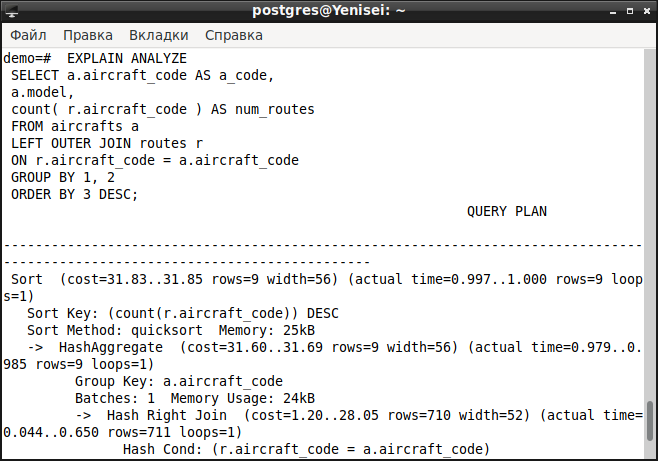
*Оконная функция выполняется (то есть выполняется “разбиение строк на разделы”, к которым дальше применяются агрегатные функции) после выборки строк, удовлетворяющих условиям заданным в WHERE (а также GROUP BY и HAVING, если они присутствуют в запросе), после чего кадры сортируются по timezone и count. Согласно этой логике WindowAgg и вызывается на 3 уровне (снизу).*

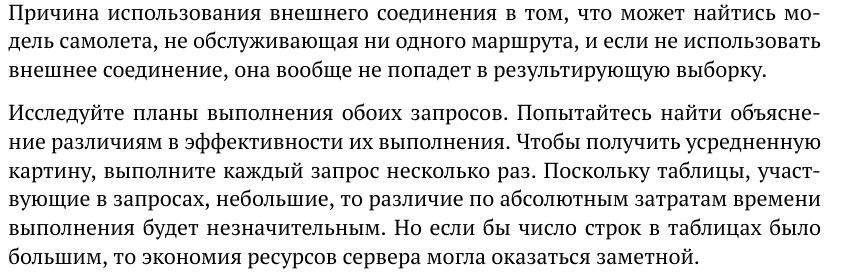
**

**

**

**

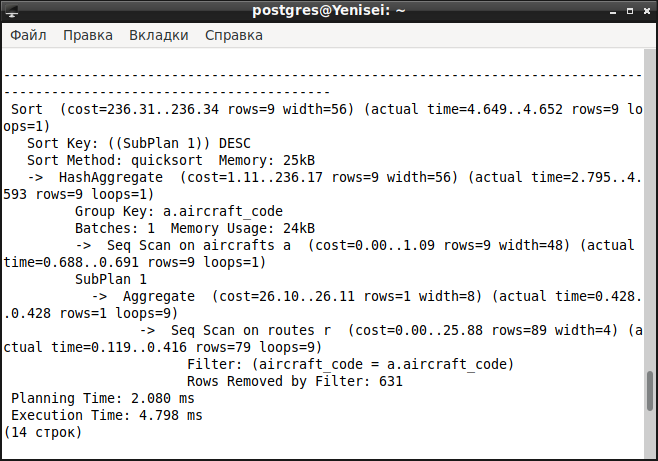
**

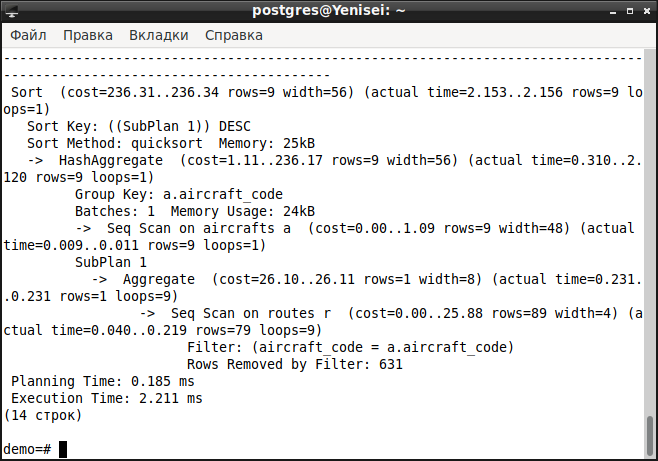
**

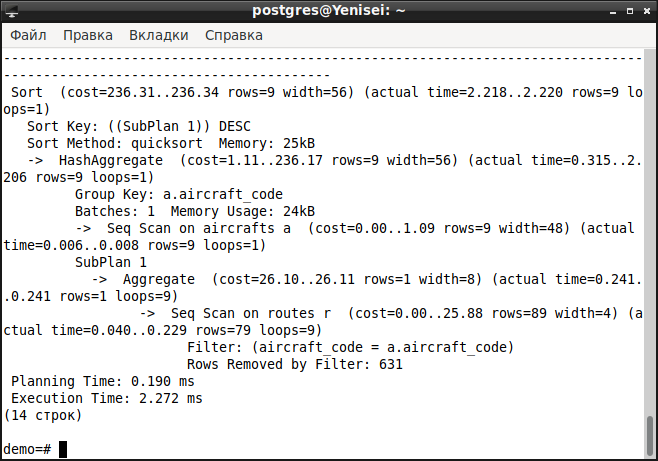
*В первом случае для каждого значения из таблицы aircrafts планировщик проходится по всем элементам таблицы routes, пытаясь найти соответствующие aircraft\_code.*

*Во втором случае создается хеш-таблица с ключами – aircraft\_code, значениями – строками таблицы aircrafts. Сопоставление строк из таблицы routes строкам из таблицы aircrafts происходит по этому ключу (за константное время).*

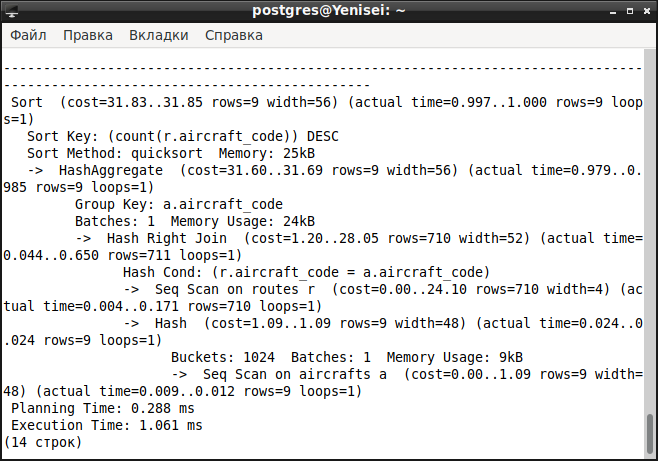
*Таким образом, первый запрос менее эффективен. То же самое заключение можно сделать из среднего времени работы обоих запросов.*

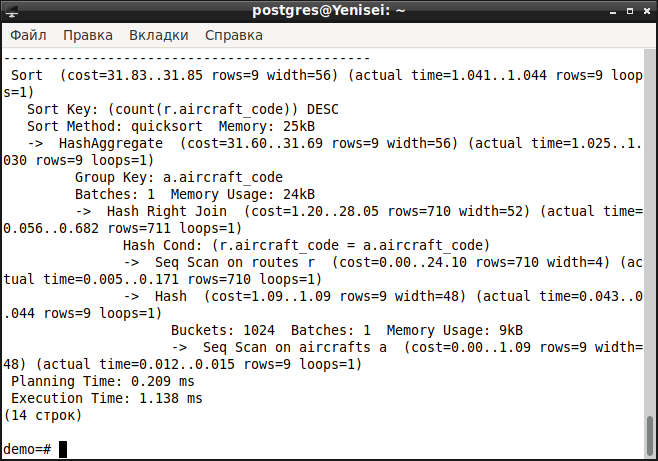
**

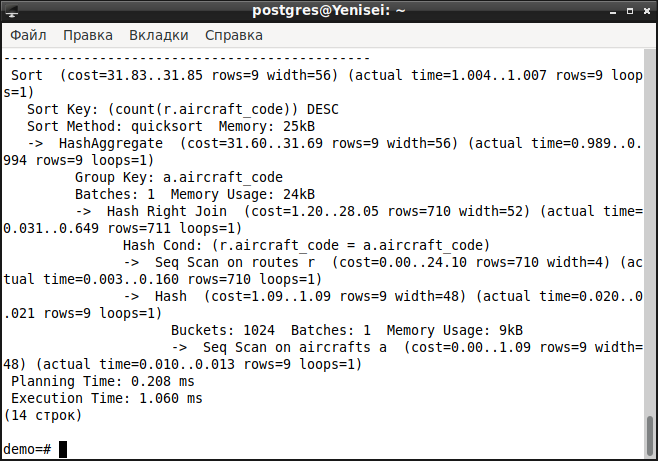
**

**

*Среднее время работы первого запроса: 3.091 ms*

**

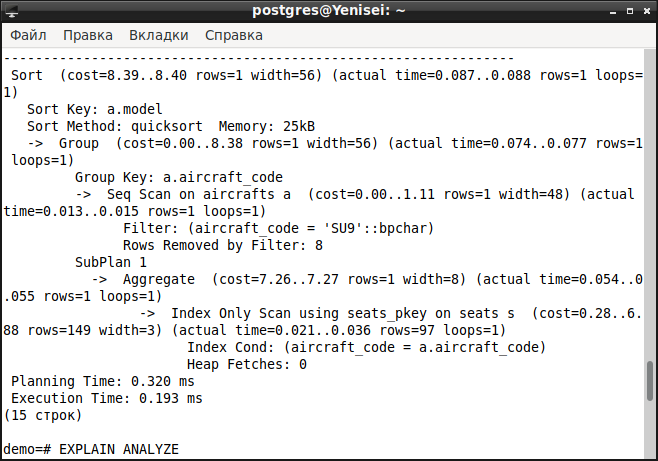
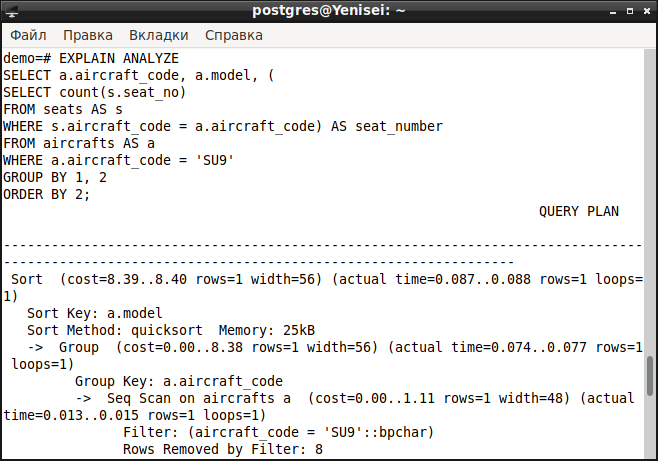
**

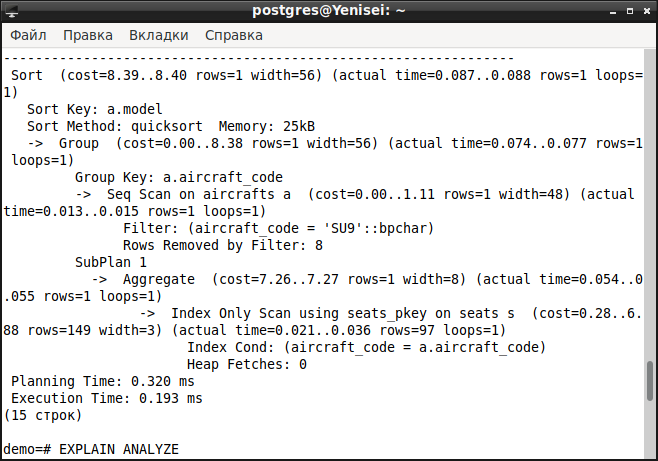
**

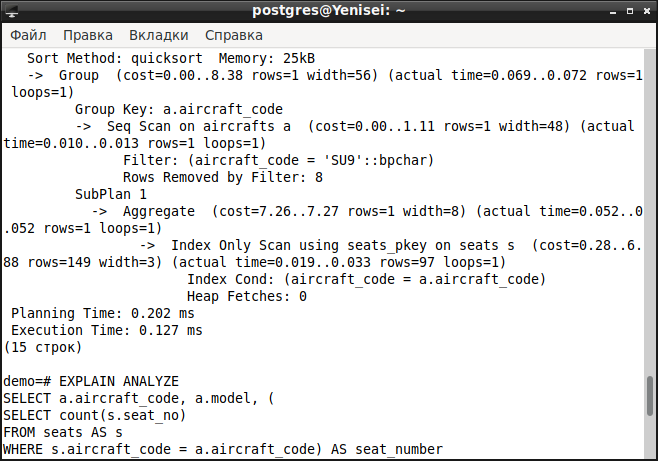
*Среднее время работы второго запроса: 1.086 ms*

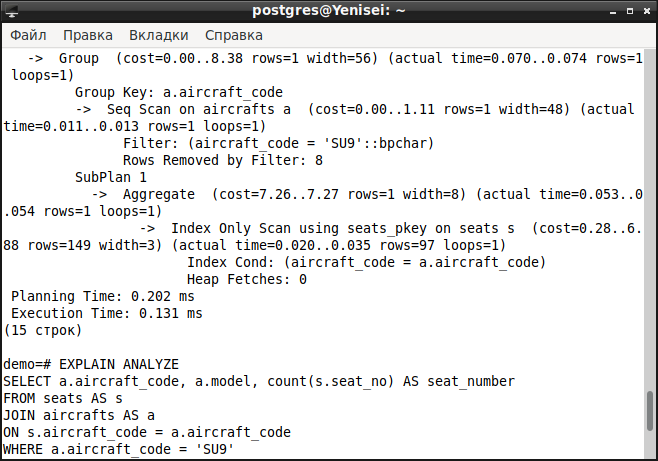
**

*Проведем пару аналогичных запросов, чтобы выяснить количество мест в самолете с кодом ‘SU-9’.*

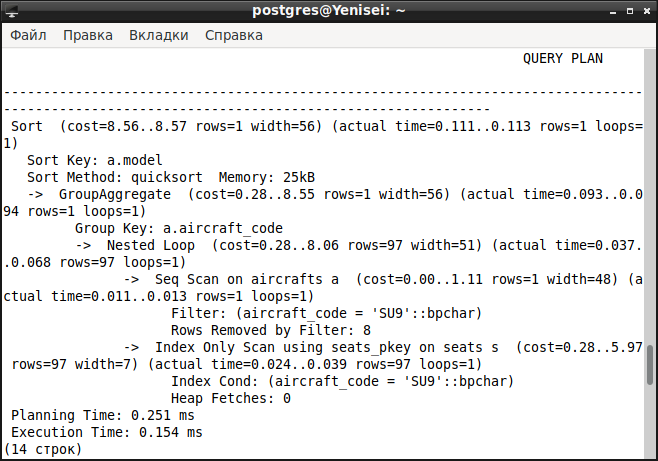
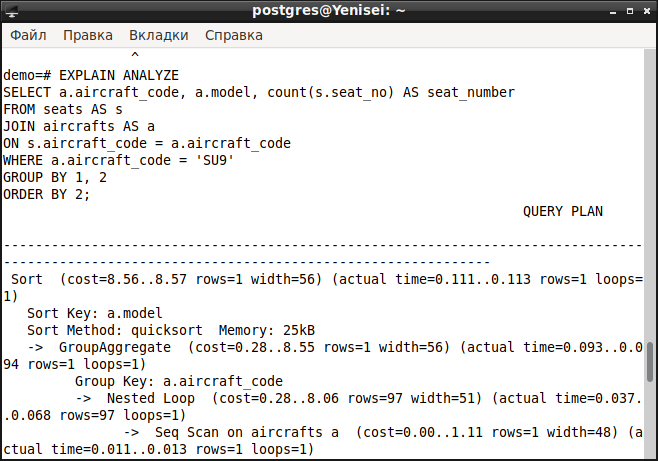
**

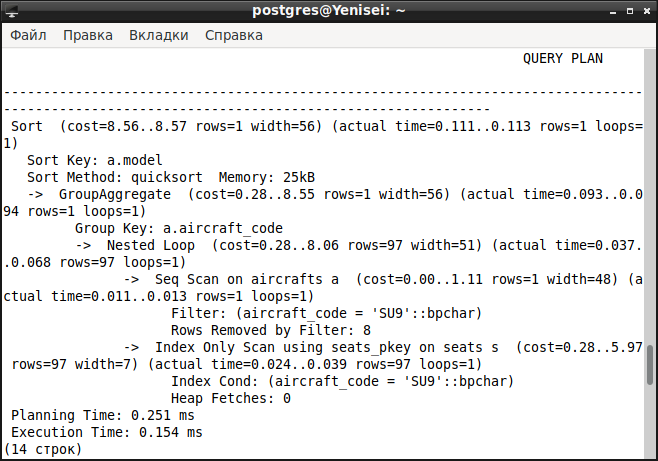
**

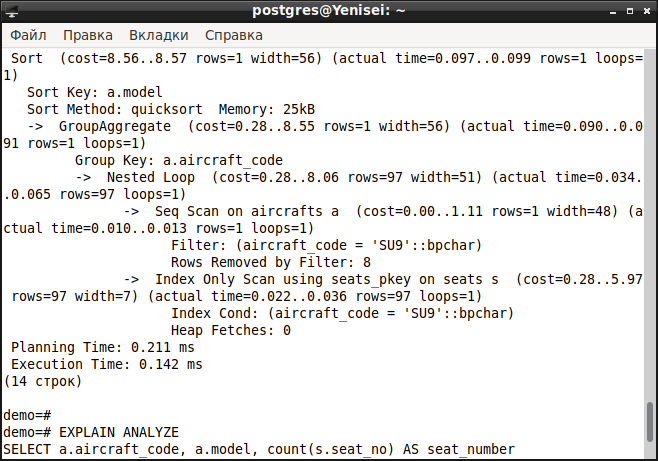
**

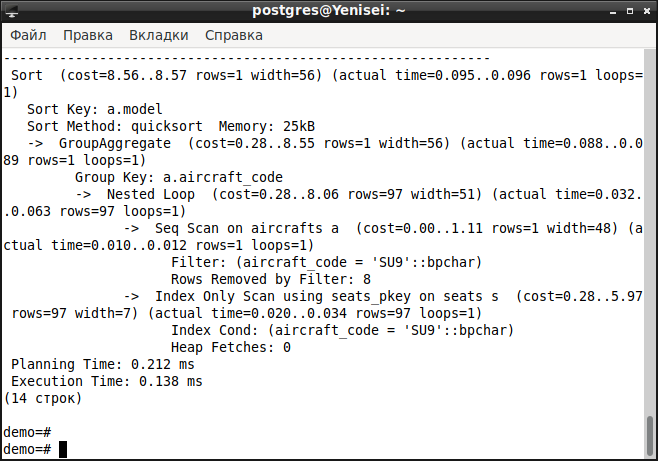
**

*Запрос с коррелированным подзапросом работает в среднем за 0.150 ms*

**

**

**

**

*Запрос с внешним соединением работает в среднем за 0.147 ms.*

*Причина того, что время выполнения запроса почти не отличается заключается в том, что во втором случае был выбран способ соединения таблиц nested loop, то есть планировщик для каждой строки из aircrafts перебирает все строки из seats.*