## Задание 3

Узел, который отвечает за материализацию, будет находиться в начале плана выполнения, до узла, который выполняет основной запрос. Это демонстрирует, что CTE выполняется и сохраняется перед тем, как остальные части запроса будут обработаны.

explain

WITH cte AS MATERIALIZED (

SELECT model, range

FROM aircrafts

WHERE range > 100

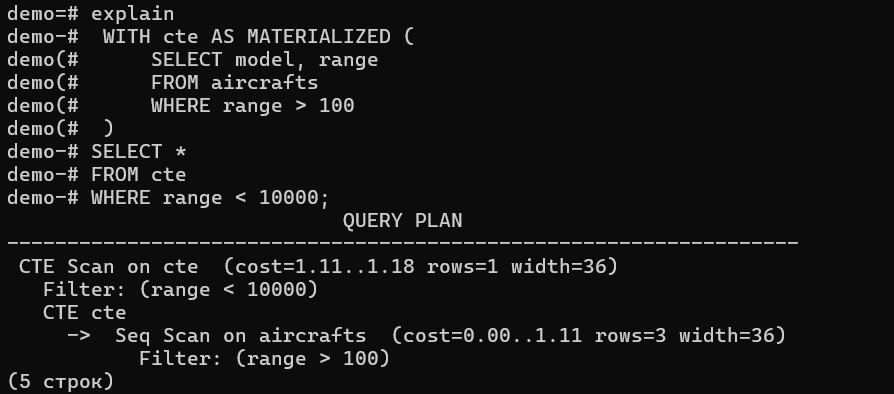
)

SELECT \*

FROM cte

WHERE range < 10000;

**(**было использована materialized чтобы запрос не свернулся в один**)**

****

## Задание 6

EXPLAIN

SELECT

model,

AVG(range) OVER (ORDER BY model ROWS BETWEEN 2 PRECEDING AND CURRENT ROW) AS moving\_average

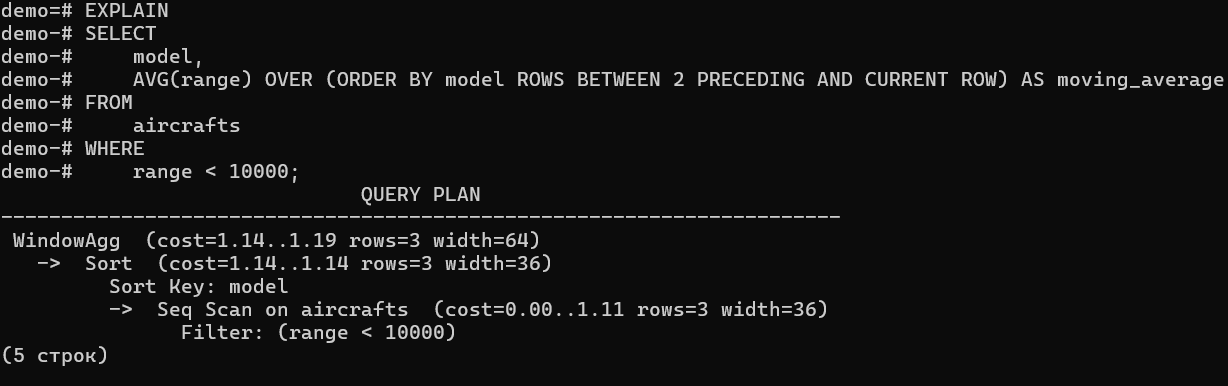
FROM

aircrafts

WHERE

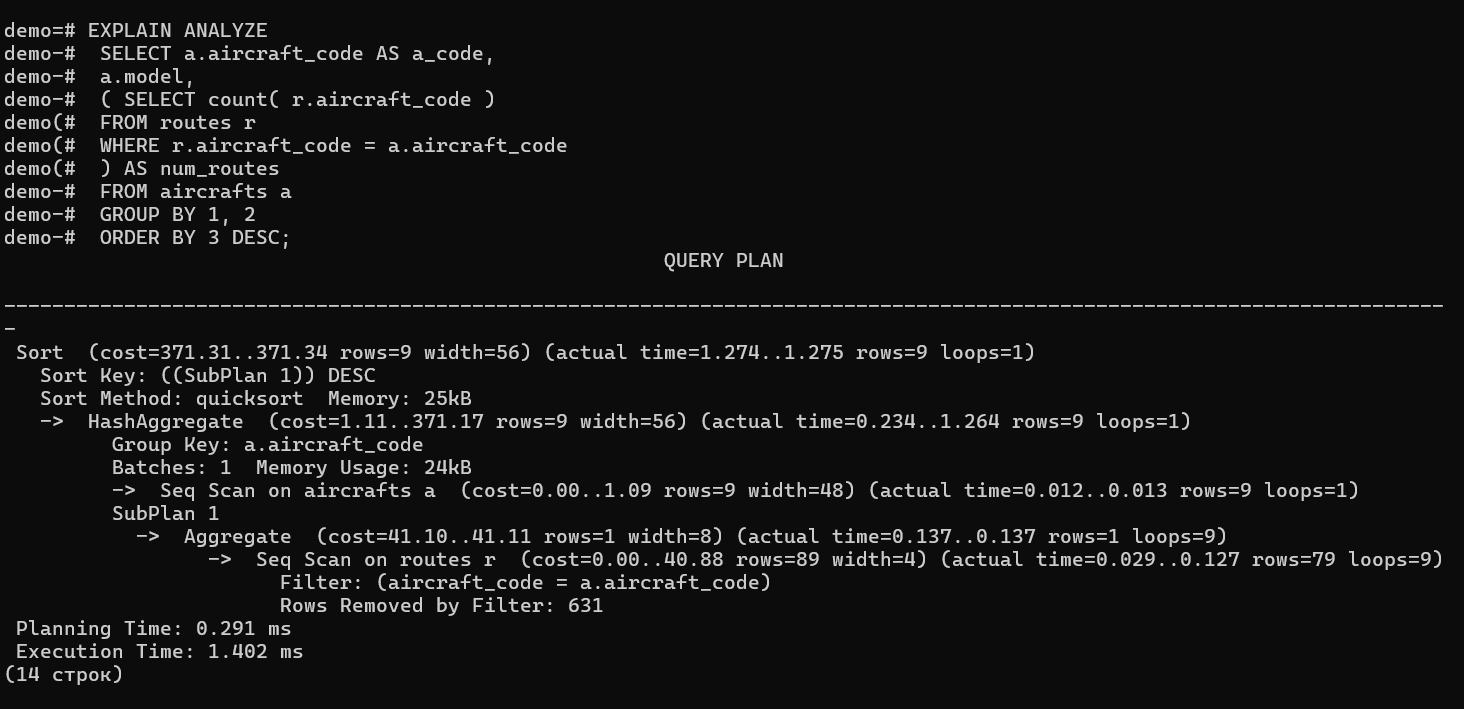
range < 10000;

windowagg находится в 1 строке



Сначала данные извлекаются из таблиц, после этого над ними выполняются преобразования, включая оконные функции. Поэтому узел windowagg появляется после извлечения данных, но до окончательного вывода результата(в данном случае это 1 строка. если бы мы добавили limit в конце запроса, то он был бы до windowagg).

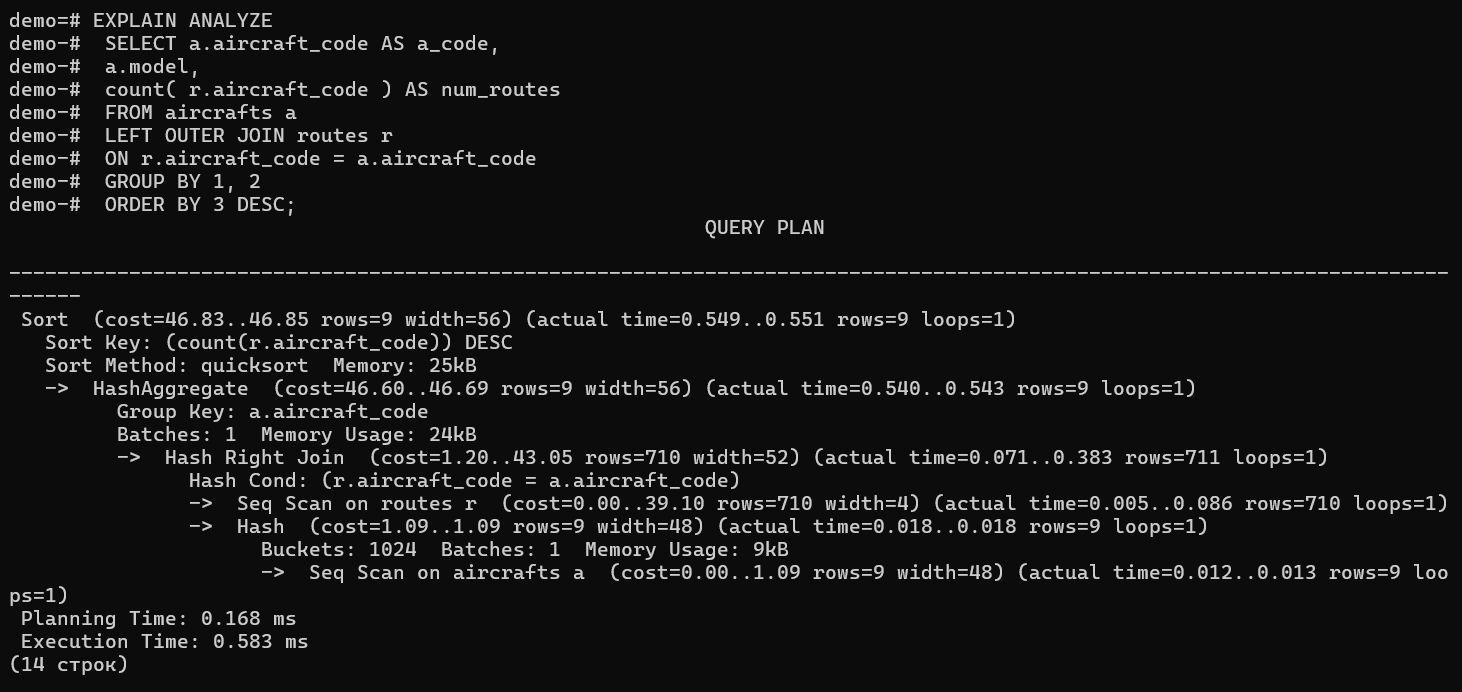
## Задание 8



запустим 3 раза

| 1 | 2 | 3 | среднее |
| --- | --- | --- | --- |
| 1.402 | 1.323 | 1.116 | 1.280 |

второй запрос



| 1 | 2 | 3 | среднее |
| --- | --- | --- | --- |
| 0.583 | 0.612 | 0.763 | 0.653 |

Объяснение результатов: в первом случае подзапросы выполняются для каждой строки, в то время как join выполняется за один проход -> нет многократного повторения для каждой строки.

**Для авиаперевозок:**

EXPLAIN ANALYZE

SELECT f.flight\_id, (SELECT sum(tf.amount)

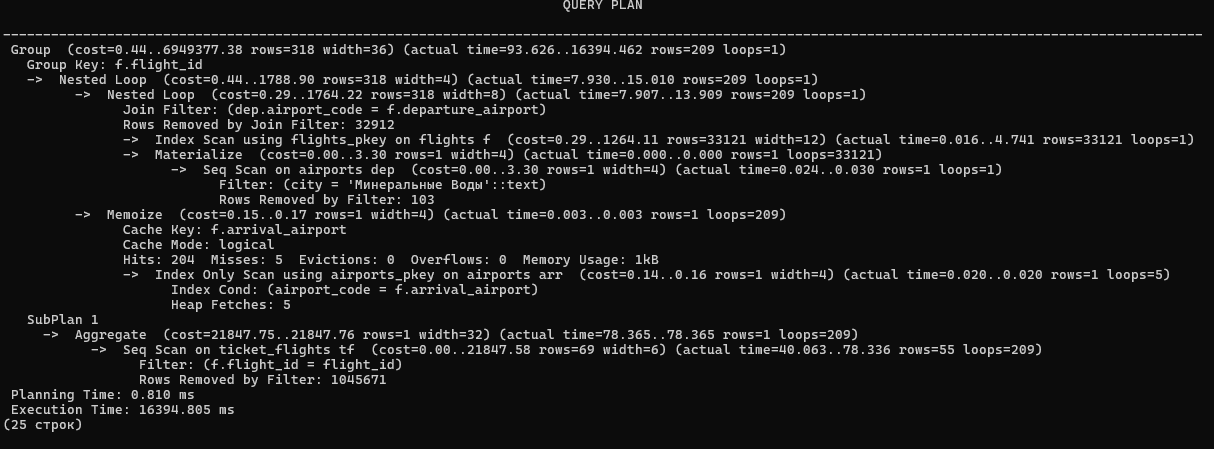
FROM ticket\_flights tf

WHERE f.flight\_id = tf.flight\_id)

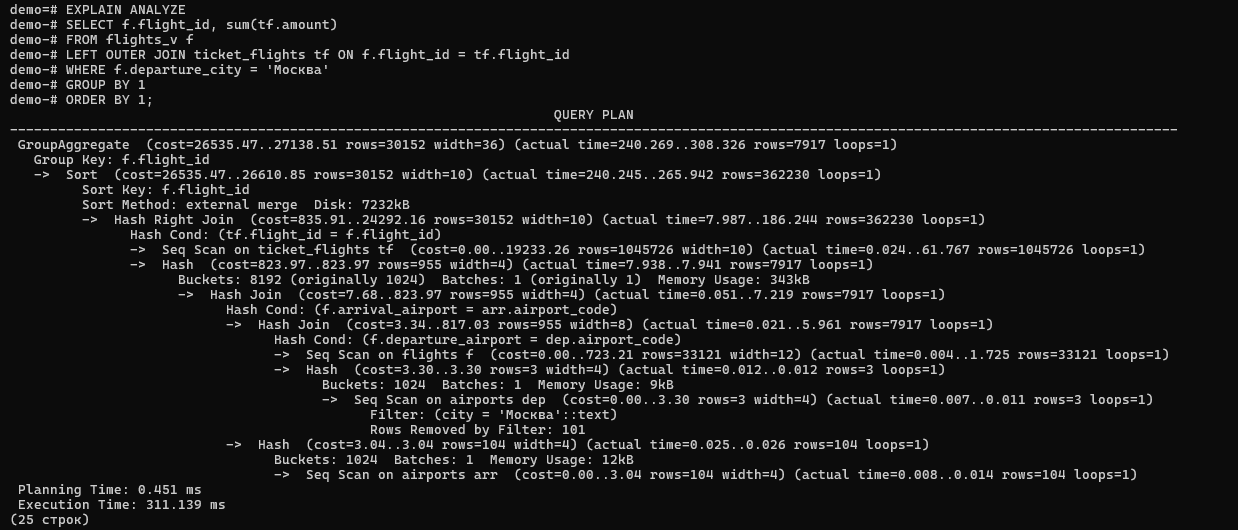
FROM flights\_v f WHERE f.departure\_city = 'Москва'

GROUP BY 1

ORDER BY 1;



среднее по 3 запросам - 18354



среднее по 3 запросам - 264