

# Домашняя работа №9

Попов Матвей, М8О-114СВ-24

## Задание 3

Самостоятельно выполните команду EXPLAIN для запроса, содержащего общее табличное выражение (CTE). Посмотрите, на каком уровне находится узел плана, отвечающий за это выражение, как он оформляется. Учтите, что общие табличные выражения всегда материализуются, т. е. вычисляются однократно и результат их вычисления сохраняется в памяти, а затем все последующие обращения в рамках запроса направляются уже к этому материализованному результату.

## Запрос

```
EXPLAIN WITH cities AS (  
    SELECT DISTINCT city FROM airports  
)  
SELECT count(*)  
FROM cities a1  
JOIN cities a2 ON a1.city <> a2.city;
```

## Результат

```
Aggregate (cost=292.11..292.12 rows=1 width=8)  
  CTE cities  
    -> Unique (cost=33.78..34.30 rows=101 width=32)  
        -> Sort (cost=33.78..34.04 rows=104 width=32)  
            Sort Key: (ml.city ->> lang())  
            -> Seq Scan on airports_data ml  
(cost=0.00..30.30 rows=104 width=32)  
    -> Nested Loop (cost=0.00..232.55 rows=10100 width=0)  
        Join Filter: (a1.city <> a2.city)  
        -> CTE Scan on cities a1 (cost=0.00..2.02 rows=101  
width=32)  
        -> CTE Scan on cities a2 (cost=0.00..2.02 rows=101  
width=32)
```

Узел, отвечающий за CTE – CTE cities. Он состоит из последовательного сканирования всей таблицы и сортировки, чтобы отобрать уникальные значения.

## Задание 6

Выполните команду EXPLAIN для запроса, в котором использована какая-нибудь из оконных функций. Найдите в плане выполнения запроса узел с именем WindowAgg. Попробуйте объяснить, почему он занимает именно этот уровень в плане.

### Запрос

```
explain select city, row_number() over (partition by city order
by city) as city_rank from airports;
```

### Результат

```
WindowAgg  (cost=33.78..62.12 rows=104 width=40)
->  Sort   (cost=33.78..34.04 rows=104 width=32)
      Sort Key: ((ml.city ->> lang()))
      ->  Seq Scan on airports_data ml  (cost=0.00..30.30
rows=104 width=32)
```

WindowAgg занимает этот уровень, так как оконные функции для их корректной отработки должны выполняться после сортировки данных.

## Задание 8

Исследуйте планы выполнения обоих запросов. Попробуйте найти объяснение различиям в эффективности их выполнения. Чтобы получить усредненную картину, выполните каждый запрос несколько раз. Поскольку таблицы, участвующие в запросах, небольшие, то различие по абсолютным затратам времени выполнения будет незначительным. Но если бы число строк в таблицах было большим, то экономия ресурсов сервера могла оказаться заметной.

Предложите аналогичную пару запросов к базе данных «Авиаперевозки». Проведите необходимые эксперименты с вашими запросами.

### Запрос

```
EXPLAIN ANALYZE
SELECT a.aircraft_code AS a_code,
       a.model,
       (SELECT count(r.aircraft_code)
        FROM routes r
```

```
        WHERE r.aircraft_code = a.aircraft_code) AS num_routes
FROM aircrafts a
GROUP BY 1, 2
ORDER BY 3 DESC;
```

```
EXPLAIN ANALYZE
SELECT a.aircraft_code AS a_code,
       a.model,
       count(r.aircraft_code) AS num_routes
FROM aircrafts a
     LEFT OUTER JOIN routes r ON r.aircraft_code =
a.aircraft_code
GROUP BY 1, 2
ORDER BY 3 DESC;
```

## Результат

Запрос с коррелированным подзапросом – 35ms

Запрос с внешним соединением – 80ms

Различие в производительности двух запросов может заключаться в том, что в запросе с коррелированным подзапросом подзапрос выполняется для каждой строки таблицы, в то время как в запросе с внешним соединением соединение выполняется всего один раз.