

МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ  
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

Институт №8 «Компьютерные науки и прикладная математика»  
Кафедра 806 «Вычислительная математика и программирование»

**Лабораторная работа №9**  
**по курсу «Проектирование баз данных»**

Выполнила: Прудникова А. А.  
Группа: М8О-114СВ-24  
Преподаватель: Моргунов Е. П.

Москва, 2024

## 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 cte_aircrafts AS (  
    SELECT aircraft_code, range  
    FROM aircrafts_tmp  
    WHERE range > 5000  
)  
SELECT aircraft_code  
FROM cte_aircrafts  
WHERE range < 8000;
```

### Результат

1	Seq Scan on aircrafts_tmp (cost=0.00..1.14 rows=1 width=16)
2	Filter: ((range > 5000) AND (range < 8000))

### Решение

Несмотря на использование CTE, оптимизатор решил не материализовать его, а выполнить объединение условий фильтрации и выполнить их за один проход по таблице. Это делает запрос более эффективным, так как исключает дополнительную операцию по материализации данных.

## 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 aircraft_code, model, range,
```

```

        AVG(range) OVER (PARTITION BY model ORDER BY range) AS
avg_range
FROM aircrafts_data;

```

## Результат

1	WindowAgg (cost=1.23..1.41 rows=9 width=84)
2	-> Sort (cost=1.23..1.26 rows=9 width=52)
3	Sort Key: model, range
4	-> Seq Scan on aircrafts_data (cost=0.00..1.09 rows=9 width=52)

## Решение

WindowAgg находится на завершающем уровне, так как для работы оконной функции требуются уже отсортированные данные.

## Задание 8

Замена коррелированного подзапроса соединением таблиц является одним из способов повышения производительности.

Предположим, что мы задались вопросом: сколько маршрутов обслуживают самолеты каждого типа? При этом нужно учитывать, что может иметь место такая ситуация, когда самолеты какого-либо типа не обслуживают ни одного маршрута. Поэтому необходимо использовать не только представление «Маршруты» (routes), но и таблицу «Самолеты» (aircrafts).

Это первый вариант запроса, в нем используется коррелированный подзапрос.

### 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;

```

А в этом варианте коррелированный подзапрос раскрыт и заменен внешним

соединением:

**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;
```

Причина использования внешнего соединения в том, что может найтись модель самолета, не обслуживающая ни одного маршрута, и если не использовать внешнее соединение, она вообще не попадет в результирующую выборку.

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;
```

## Результат

Planning Time: 6.597 ms	Planning Time: 0.658 ms	Planning Time: 0.343 ms
Execution Time: 96.685 ms	Execution Time: 87.206 ms	Execution Time: 85.084 ms

## Запрос

**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

Planning Time: 0.338 ms
Execution Time: 32.481 ms

Planning Time: 0.279 ms
Execution Time: 31.517 ms

Planning Time: 0.426 ms
Execution Time: 32.254 ms

## Решение

Очевидно, что во втором случае время исполнения меньше. Так происходит из-за того, что при коррелированном подзапросе операция выполняется для каждой строки, а после замены на внешнее соединение операция используется однократно.