

HAVING





Правила вебинара

<https://aristov.tech>

Задаем вопрос в чат

Вопросы вижу, отвечу в момент логической паузы

Если есть вопрос голосом - поставьте знак ? в чат

Если остались вопросы, можно их задать на следующем занятии

<https://aristov.tech>

Маршрут вебинара

<https://aristov.tech>

1. **HAVING**
2. **Общая структура SELECT**

HAVING

HAVING

<https://aristov.tech>

HAVING

<https://www.db-fiddle.com/f/kVSgXux6384wV7LUZJwnQn/10>

<https://aristov.tech>

Общая структура SELECT

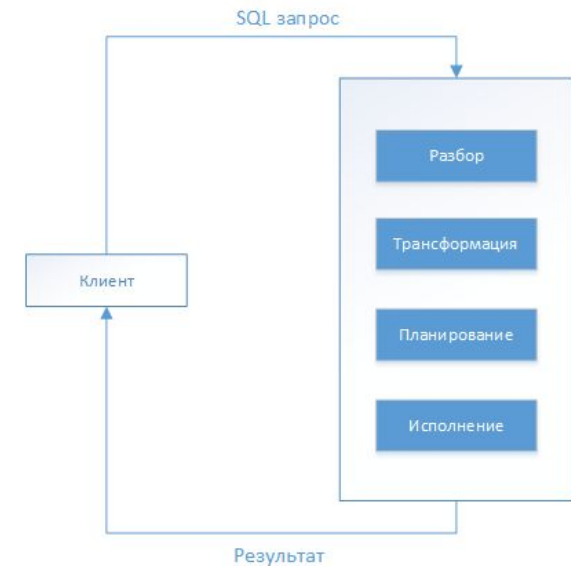
SELECT

Теперь, пройдя основные этапы, можно рассмотреть блоксхему оператора SELECT.

[Запросы в PostgreSQL: 1. Этапы выполнения](#), где рассмотрены:

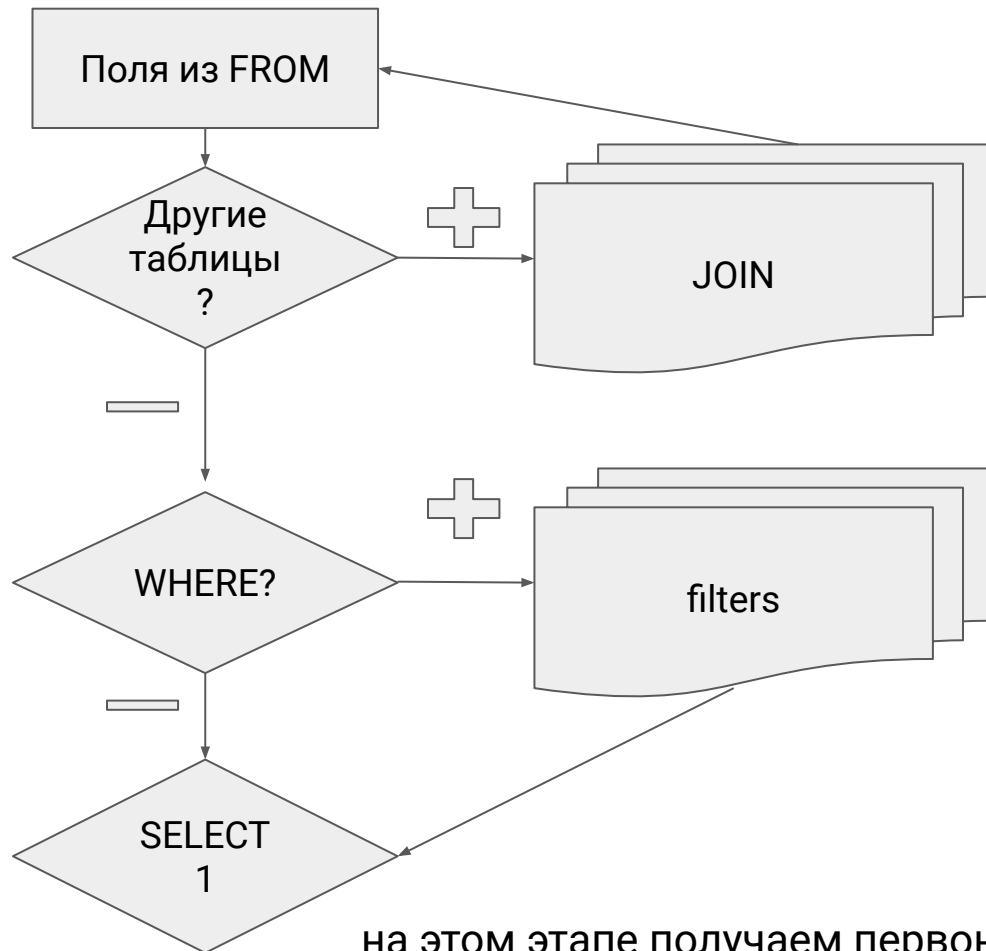
- ❖ Лексический и синтаксический разбор, семантический разбор
- ❖ Трансформация
- ❖ Планирование - > Дерево плана
- ❖ Перебор планов
- ❖ Управление порядком соединений -> Выбор лучшего плана
- ❖ Общая схема вычисления оценки, в т.ч. кардинальности
- ❖ И уже в самом конце выполнение

[Выполнение запросов в PostgreSQL | Статья | Сообщество Directum](#)



SELECT

<https://aristov.tech>



на этом этапе получаем первоначальную выборку, которую и будем группировать и фильтровать группы и тд

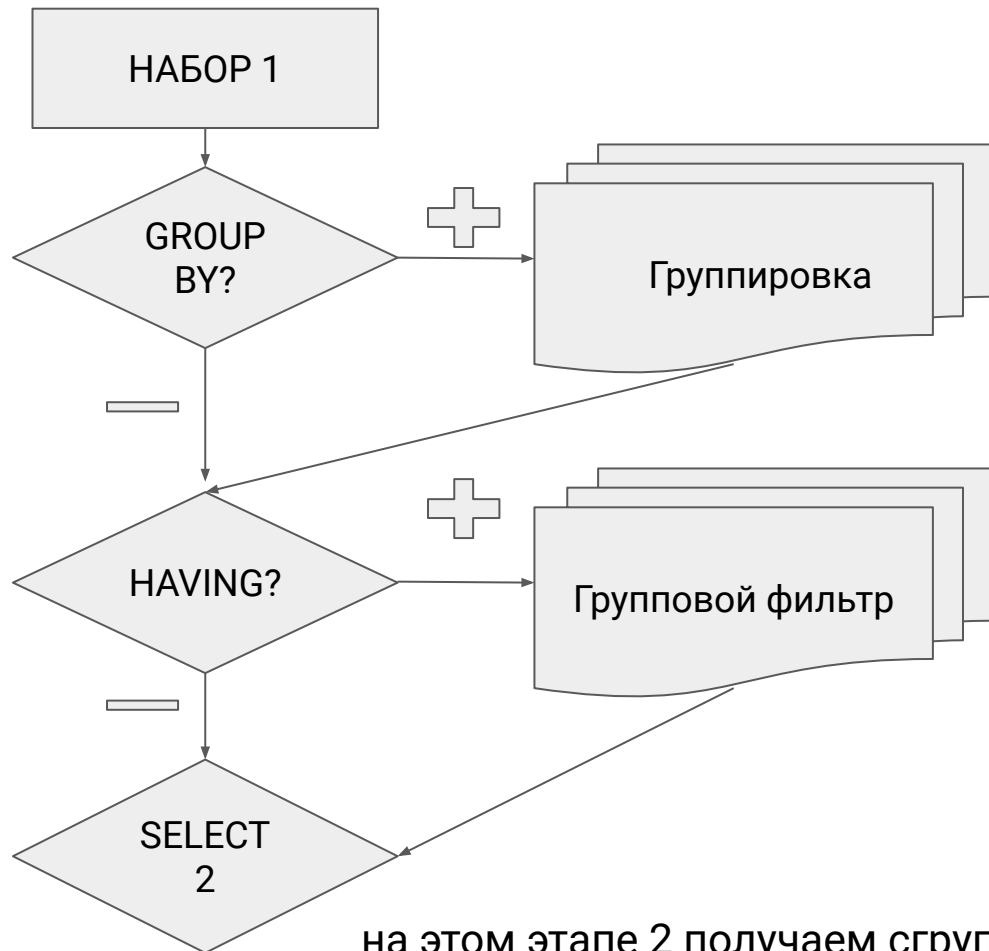
```
SELECT
customers.name,
COUNT(order_id) as Total_orders,
SUM(order_amount) as total_spent
FROM customers
JOIN orders ON customers.id = orders.customer_id
WHERE order_date >= '2023-01-01'
GROUP BY customers.name
HAVING total_spent >= 1000
ORDER BY customers.name
LIMIT 100;
```

<https://www.dev-notes.ru/articles/devops/understand-the-sql-execution-order/>

<https://aristov.tech>

SELECT

<https://aristov.tech>



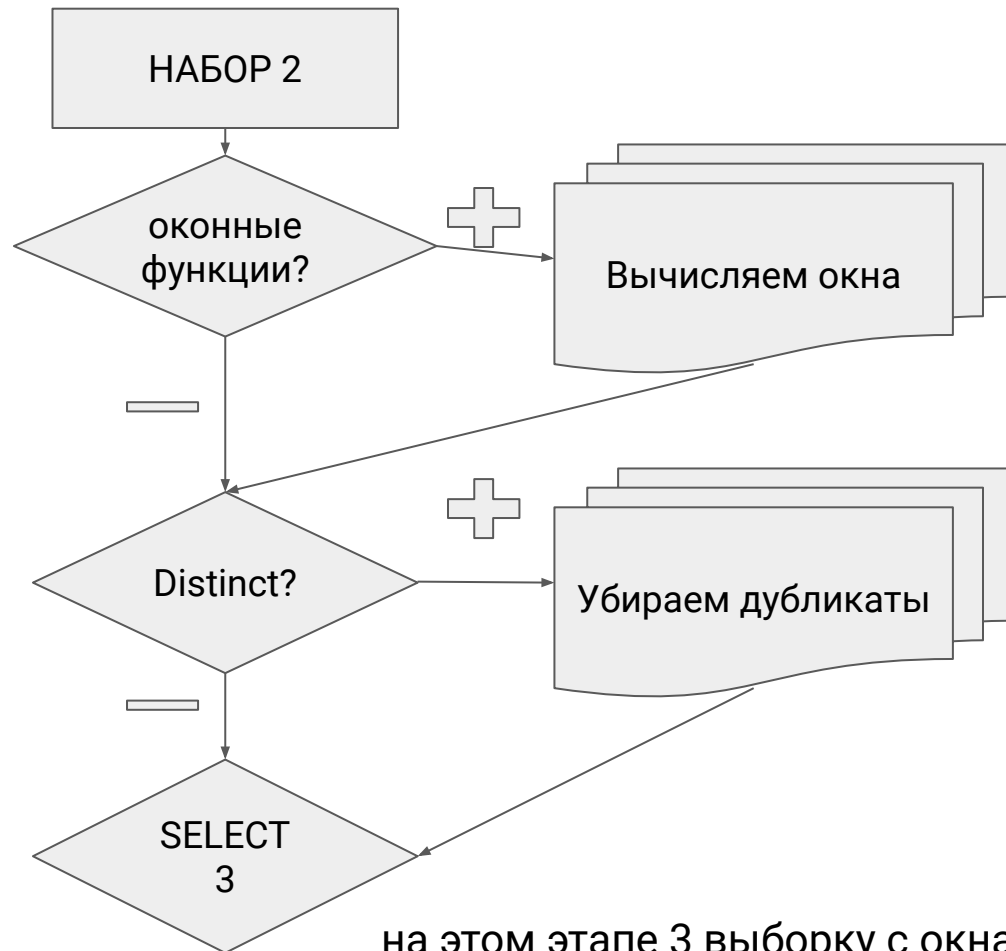
на этом этапе 2 получаем сгруппированную и отфильтрованную выборку

```
SELECT
customers.name,
COUNT(order_id) as Total_orders,
SUM(order_amount) as total_spent
FROM customers
JOIN orders ON customers.id = orders.customer_id
WHERE order_date >= '2023-01-01'
GROUP BY customers.name
HAVING total_spent >= 1000
ORDER BY customers.name
LIMIT 100;
```

<https://aristov.tech>

SELECT

<https://aristov.tech>

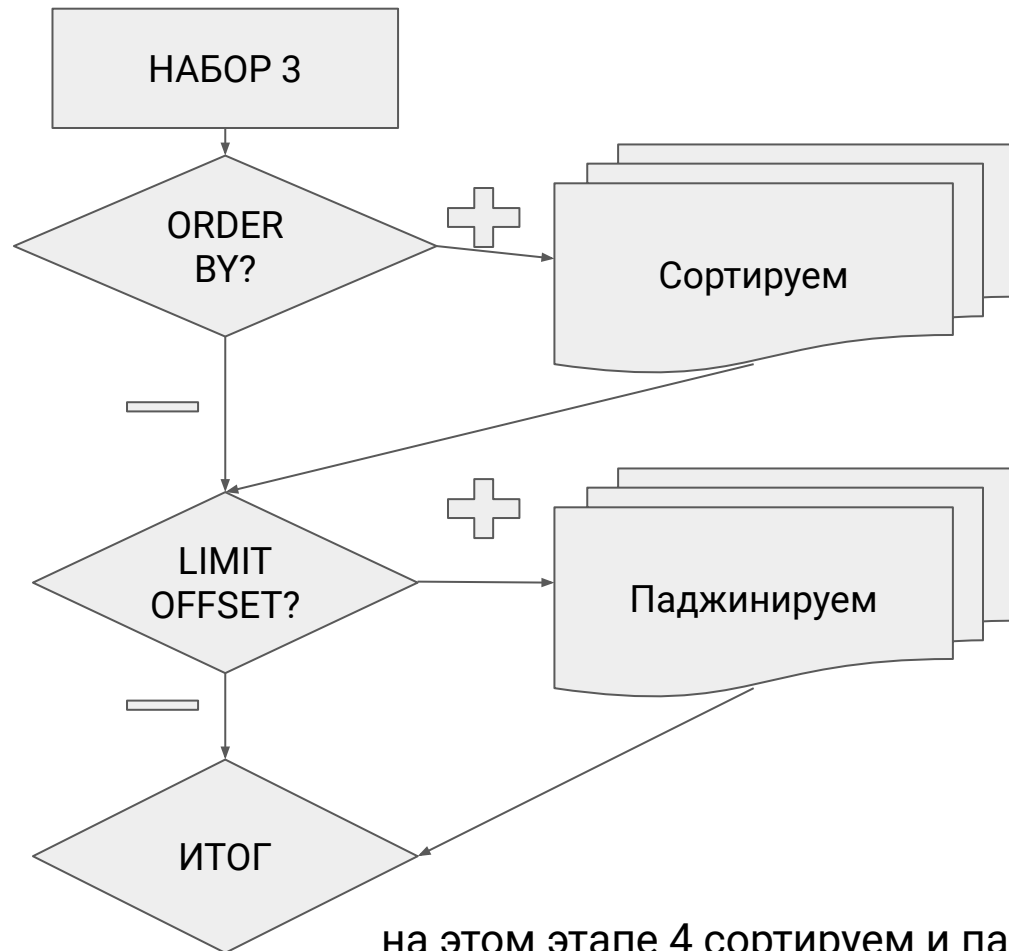


```
SELECT
customers.name,
COUNT(order_id) as Total_orders,
SUM(order_amount) as total_spent
FROM customers
JOIN orders ON customers.id = orders.customer_id
WHERE order_date >= '2023-01-01'
GROUP BY customers.name
HAVING total_spent >= 1000
ORDER BY customers.name
LIMIT 100;
```

<https://aristov.tech>

SELECT

<https://aristov.tech>



на этом этапе 4 сортируем и паджинируем

```
SELECT
customers.name,
COUNT(order_id) as Total_orders,
SUM(order_amount) as total_spent
FROM customers
JOIN orders ON customers.id = orders.customer_id
WHERE order_date >= '2023-01-01'
GROUP BY customers.name
HAVING total_spent >= 1000
ORDER BY customers.name
LIMIT 100;
```

<https://aristov.tech>

Итоги

Итоги

Остались ли вопросы?

Увидимся на следующем занятии

Спасибо за внимание!

Когда дальше и куда?
В чате напишу
материалы для бесплатного доступа будут появляться на ютубе

Аристов Евгений