

Оптимизация запросов

В начало запроса поместите EXPLAIN [ANALYZE]

```
1 explain ANALYZE select * from aircrafts
```

Результат	План выполнения	Сообщения	Notifications
QUERY PLAN			
text			
1	Seq Scan on aircrafts (cost=0.00..20.20 rows=1020 width=52) (actual time=14.038..14.043 rows=12 loops=1)		
2	Planning Time: 23.645 ms		
3	Execution Time: 14.072 ms		

В ответе получен план выполнения запроса. Поскольку в этом запросе нет предложения WHERE, он должен просмотреть все строки таблицы, поэтому планировщик выбирает последовательный просмотр. С параметром ANALYZE оператор будет выполнен на самом деле, а не только запланирован. При этом в вывод добавляются фактические сведения о времени выполнения, включая общее время, затраченное на каждый узел плана (в миллисекундах) и общее число строк, выданных в результате.

Напишите запрос на вывод всей информации из любой имеющейся у вас таблицы добавив условие отбора. В начало запроса поместите EXPLAIN [ANALYZE]

```
1 explain ANALYZE select * from aircrafts where range>3000
```

Результат	План выполнения	Сообщения	Notifications
QUERY PLAN			
text			
1	Seq Scan on aircrafts (cost=0.00..22.75 rows=340 width=52) (actual time=0.029..0.033 rows=9 loops=1)		
2	Filter: (range > 3000)		
3	Rows Removed by Filter: 3		
4	Planning Time: 0.816 ms		
5	Execution Time: 0.056 ms		

Поскольку наложено дополнительное условие на строки, выбираемые из таблицы, то ниже узла плана, отвечающего за их последовательную выборку, добавляется еще один узел, описывающий критерий отбора строк.

```
1 SET enable_hashjoin = off;
```

Результат План выполнения Сообщения Notifications

SET

Запрос завершён успешно, время выполнения: 298 msec.

Query Editor История запросов

```
1 SET enable_mergejoin = off;  
2
```

Результат План выполнения Сообщения Notifications

SET

Запрос завершён успешно, время выполнения: 338 msec.

```
1 SET enable_nestloop = off;  
2
```

Результат План выполнения Сообщения Notifications

SET

Запрос завершён успешно, время выполнения: 314 msec.

```
1 SET enable_mergejoin = off;  
2
```

Результат План выполнения Сообщения Notifications

SET

Запрос завершён успешно, время выполнения: 214 msec.

Query EditorИстория запросов

1

EXPLAIN SELECT t.ticket_no,

2

t.passenger_name,

3

tf.flight_id,

4

tf.amount

5

FROM tickets t

6

JOIN ticket_flights tf ON t.ticket_no = tf.ticket_no

7

ORDER BY t.ticket_no;

РезультатПлан выполненияСообщенияNotifications

QUERY PLAN

text

1

Nested Loop (cost=10000000000.30..10000000172.30 rows=570 width=108)

2

-> Index Scan using tickets_pkey on tickets t (cost=0.15..49.25 rows=340 width=88)

3

-> Index Scan using ticket_flights_pkey on ticket_flights tf (cost=0.15..0.33 rows=3 width=76)

4

Index Cond: (ticket_no = t.ticket_no)

1

SET enable_mergejoin = on;

2

Найти в ЯндексеКопироватьВ заметки

РезультатПлан выполненияСообщенияNotifications

SET

Запрос завершён успешно, время выполнения: 322 msec.

postgres/postgres@PostgreSQL 12 (64bit)

Query EditorИстория запросов

1

EXPLAIN ANALYZE

2

SELECT t.ticket_no,

3

t.passenger_name,

4

tf.flight_id,

5

tf.amount

6

FROM tickets t

7

JOIN ticket_flights tf ON t.ticket_no = tf.ticket_no

8

ORDER BY t.ticket_no;

РезультатПлан выполненияСообщенияNotifications

QUERY PLAN

text

1

Merge Join (cost=69.49..79.74 rows=570 width=108) (actual time=0.040..0.041 rows=0 loops=1)

2

Merge Cond: (t.ticket_no = tf.ticket_no)

3

-> Sort (cost=27.70..28.55 rows=340 width=88) (actual time=0.039..0.039 rows=0 loops=1)

4

Sort Key: t.ticket_no

5

Sort Method: quicksort Memory: 25kB

6

-> Seq Scan on tickets t (cost=0.00..13.40 rows=340 width=88) (actual time=0.032..0.032 rows=0 loops=1)

7

-> Sort (cost=41.79..43.22 rows=570 width=76) (never executed)

8

Sort Key: tf.ticket_no

9

-> Seq Scan on ticket_flights tf (cost=0.00..15.70 rows=570 width=76) (never executed)

10

Planning Time: 0.344 ms

11

Execution Time: 0.095 ms

Query EditorИстория запросов

```
2 SELECT t.ticket_no,
3 t.passenger_name,
4 tf.flight_id,
5 tf.amount
6 FROM tickets t
7 JOIN ticket_flights tf ON t.ticket_no = tf.ticket_no
8 WHERE amount > 50000
9 ORDER BY t.ticket_no;
```

РезультатПлан выполненияСообщенияNotifications

QUERY PLAN
text

1	Merge Join (cost=52.01..56.56 rows=190 width=108) (actual time=0.063..0.064 rows=0 loops=1)
2	Merge Cond: (t.ticket_no = tf.ticket_no)
3	-> Sort (cost=27.70..28.55 rows=340 width=88) (actual time=0.061..0.062 rows=0 loops=1)
4	Sort Key: t.ticket_no
5	Sort Method: quicksort Memory: 25kB
6	-> Seq Scan on tickets t (cost=0.00..13.40 rows=340 width=88) (actual time=0.040..0.040 rows=0 loops=1)
7	-> Sort (cost=24.32..24.79 rows=190 width=76) (never executed)
8	Sort Key: tf.ticket_no
9	-> Seq Scan on ticket_flights tf (cost=0.00..17.13 rows=190 width=76) (never executed)
10	Filter: (amount > '50000'::numeric)
11	Planning Time: 1.033 ms

Активация Windows
Чтобы активировать Windows, перейдите на сайт "Параметры".

postgres/postgres@PostgreSQL 12 (64bit) ▾

Query EditorИстория запросов

```
1 EXPLAIN (ANALYZE, COSTS OFF)
2 SELECT a.aircraft_code,
3 a.model,
4 s.seat_no,
5 s.fare_conditions
6 FROM seats s
7 JOIN aircrafts a ON s.aircraft_code = a.aircraft_code
8 WHERE a.model ~ '^Air'
9 ORDER BY s.seat_no;
```

РезультатПлан выполненияСообщенияNotifications

QUERY PLAN
text

1	Sort (actual time=0.023..0.024 rows=0 loops=1)
2	Sort Key: s.seat_no
3	Sort Method: quicksort Memory: 25kB
4	-> Merge Join (actual time=0.016..0.017 rows=0 loops=1)
5	Merge Cond: (s.aircraft_code = a.aircraft_code)
6	-> Index Scan using seats_pkey on seats s (actual time=0.014..0.014 rows=0 loops=1)
7	-> Sort (never executed)
8	Sort Key: a.aircraft_code
9	-> Seq Scan on aircrafts a (never executed)
10	Filter: (model ~ '^Air'::text)

Активация Windows
Чтобы активировать Windows, перейдите на сайт "Параметры".

```
SQL Shell (psql)
Текущая кодовая страница: 1251
Пароль пользователя postgres:
psql (12.5)
Введите "help", чтобы получить справку.

postgres=# BEGIN;
BEGIN
postgres=# EXPLAIN (ANALYZE, COSTS OFF)
postgres=# UPDATE aircrafts
postgres=# SET range = range + 100
postgres=# WHERE model ~ '^Air';
               QUERY PLAN
-----
Update on aircrafts (actual time=7.397..7.398 rows=0 loops=1)
  -> Seq Scan on aircrafts (actual time=0.170..0.206 rows=3 loops=1)
        Filter: (model ~ '^Air'::text)
        Rows Removed by Filter: 9
Planning Time: 2.178 ms
Execution Time: 14.978 ms
(6 строк)

postgres=# ROLLBACK;
```