

postgres/postgres@PostgreSQL 13 (64bit) ▾

Query Editor История запросов

```

1 SET enable_hashjoin = off;
2 SET enable_mergejoin = off;
3 SET enable_nestloop = off;

```

Результат План выполнения Сообщения Notifications

SET

Запрос завершён успешно, время выполнения: 57 мсес.

postgres/postgres@PostgreSQL 13 (64bit) ▾

Query Editor История запросов

```

1 EXPLAIN
2 SELECT a.model, count( * )
3 FROM aircrafts a, seats s
4 WHERE a.aircraft_code = s.aircraft_code
5 GROUP BY a.aircraft_code;

```

Результат План выполнения Сообщения Notifications

QUERY PLAN

text

1	GroupAggregate (cost=10000000000.30..10000000192.61 rows=800 width=56)
2	Group Key: a.aircraft_code
3	-> Nested Loop (cost=10000000000.30..10000000180.61 rows=800 width=48)
4	-> Index Only Scan using seats_pkey on seats s (cost=0.15..28.65 rows=800 width=16)
5	-> Index Scan using aircrafts_pkey on aircrafts a (cost=0.15..0.19 rows=1 width=48)
6	Index Cond: (aircraft_code = s.aircraft_code)

postgres/postgres@PostgreSQL 13 (64bit) ▾

Query Editor История запросов

```

1 EXPLAIN ANALYZE
2 SELECT a.model, count( * )
3 FROM aircrafts a, seats s
4 WHERE a.aircraft_code = s.aircraft_code
5 GROUP BY a.aircraft_code;

```

Результат План выполнения Сообщения Notifications

QUERY PLAN

text

1	GroupAggregate (cost=10000000000.30..10000000192.61 rows=800 width=56) (actual time=0.007..0.008 rows=0 loops=1)
2	Group Key: a.aircraft_code
3	-> Nested Loop (cost=10000000000.30..10000000180.61 rows=800 width=48) (actual time=0.006..0.007 rows=0 loops=1)
4	-> Index Only Scan using seats_pkey on seats s (cost=0.15..28.65 rows=800 width=16) (actual time=0.005..0.006 rows=0 loops=1)
5	Heap Fetches: 0
6	-> Index Scan using aircrafts_pkey on aircrafts a (cost=0.15..0.19 rows=1 width=48) (never executed)
7	Index Cond: (aircraft_code = s.aircraft_code)
8	Planning Time: 0.142 ms
9	Execution Time: 0.042 ms

postgres/postgres@PostgreSQL 13 (64bit) ▾	
Query Editor История запросов	
<pre> 1 set enable_nestloop = on; 2 EXPLAIN ANALYZE 3 SELECT a.model, count(*) 4 FROM aircrafts a, seats s 5 WHERE a.aircraft_code = s.aircraft_code 6 GROUP BY a.aircraft_code; </pre>	
Результат План выполнения Сообщения Notifications	
QUERY PLAN	text
1	HashAggregate (cost=173.96..181.96 rows=800 width=56) (actual time=0.022..0.023 rows=0 loops=1)
2	Group Key: a.aircraft_code
3	Batches: 1 Memory Usage: 49kB
4	-> Nested Loop (cost=0.15..169.96 rows=800 width=48) (actual time=0.019..0.019 rows=0 loops=1)
5	-> Seq Scan on seats s (cost=0.00..18.00 rows=800 width=16) (actual time=0.018..0.018 rows=0 loops=1)
6	-> Index Scan using aircrafts_pkey on aircrafts a (cost=0.15..0.19 rows=1 width=48) (never executed)
7	Index Cond: (aircraft_code = s.aircraft_code)
8	Planning Time: 0.235 ms
9	Execution Time: 0.095 ms

При включенном enable_nestloop оценка стоимости запроса при использовании вложенного запроса значительно уменьшается, но общее время выполнения запроса увеличилось.

postgres/postgres@PostgreSQL 13 (64bit) ▾	
Query Editor История запросов	
<pre> 1 set enable_mergejoin = on; 2 EXPLAIN ANALYZE 3 SELECT a.model, count(*) 4 FROM aircrafts a, seats s 5 WHERE a.aircraft_code = s.aircraft_code 6 GROUP BY a.aircraft_code; </pre>	
Результат План выполнения Сообщения Notifications	
QUERY PLAN	text
1	GroupAggregate (cost=0.30..86.65 rows=800 width=56) (actual time=0.008..0.009 rows=0 loops=1)
2	Group Key: a.aircraft_code
3	-> Merge Join (cost=0.30..74.65 rows=800 width=48) (actual time=0.008..0.008 rows=0 loops=1)
4	Merge Cond: (a.aircraft_code = s.aircraft_code)
5	-> Index Scan using aircrafts_pkey on aircrafts a (cost=0.15..33.45 rows=1020 width=48) (actual time=0.005..0.005 rows=1 loops=1)
6	-> Index Only Scan using seats_pkey on seats s (cost=0.15..28.65 rows=800 width=16) (actual time=0.002..0.002 rows=0 loops=1)
7	Heap Fetches: 0
8	Planning Time: 0.118 ms
9	Execution Time: 0.032 ms

При включенном mergejoin оценка стоимости запроса общее время выполнения запроса при использовании вложенного запроса значительно уменьшается.

postgres/postgres@PostgreSQL 13 (64bit) ▾	
Query Editor История запросов	
<pre> 1 set enable_nestloop = off; 2 EXPLAIN ANALYZE 3 SELECT a.model, count(*) 4 FROM aircrafts a left join seats s 5 on (a.aircraft_code = s.aircraft_code) 6 GROUP BY a.aircraft_code; </pre>	
Результат План выполнения Сообщения Notifications	
QUERY PLAN	text
1	GroupAggregate (cost=10000000000.30..10000000330.45 rows=1020 width=56) (actual time=0.016..0.037 rows=8 loops=1)
2	Group Key: a.aircraft_code
3	-> Nested Loop Left Join (cost=10000000000.30..10000000315.15 rows=1020 width=48) (actual time=0.009..0.029 rows=8 loops=1)
4	-> Index Scan using aircrafts_pkey on aircrafts a (cost=0.15..33.45 rows=1020 width=48) (actual time=0.005..0.007 rows=8 loops=1)
5	-> Index Only Scan using seats_pkey on seats s (cost=0.15..0.24 rows=4 width=16) (actual time=0.001..0.001 rows=0 loops=8)
6	Index Cond: (aircraft_code = a.aircraft_code)
7	Heap Fetches: 0
8	Planning Time: 0.194 ms
9	Execution Time: 0.079 ms

При использовании левого соединения и отключенном nestloop оценка стоимости запроса значительно увеличивается, а общее время выполнения запроса принимает среднее значение.