Query 9

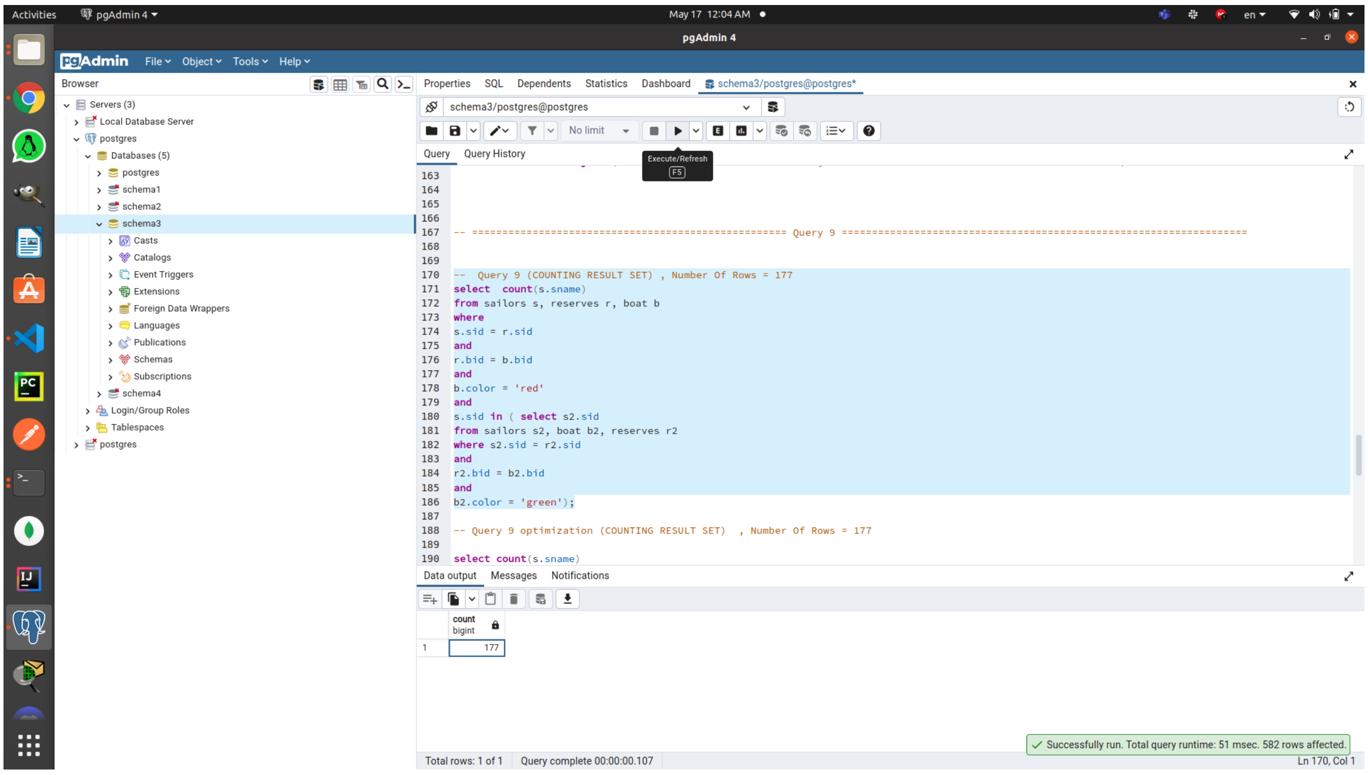
- Find the names of sailors who have reserved both a red and a green boat.

Original Query

```
select s.sname
from sailors s, reserves r, boat b
where
s.sid = r.sid
and
r.bid = b.bid
and
b.color = 'red'
and
s.sid in ( select s2.sid
from sailors s2, boat b2, reserves r2
where s2.sid = r2.sid
and
r2.bid = b2.bid
and
b2.color = 'red');
```

Result Set

- 177 Rows



```
-- ================================================= Query 9 ===============================================================


--  Query 9 (COUNTING RESULT SET) , Number Of Rows = 177
select  count(s.sname)
from sailors s, reserves r, boat b
where
s.sid = r.sid
and
r.bid = b.bid
and
b.color = 'red'
and
s.sid in ( select s2.sid
from sailors s2, boat b2, reserves r2
where s2.sid = r2.sid
and
r2.bid = b2.bid
and
b2.color = 'green');

-- Query 9 optimization (COUNTING RESULT SET)  , Number Of Rows = 177

select count(s.sname)
```

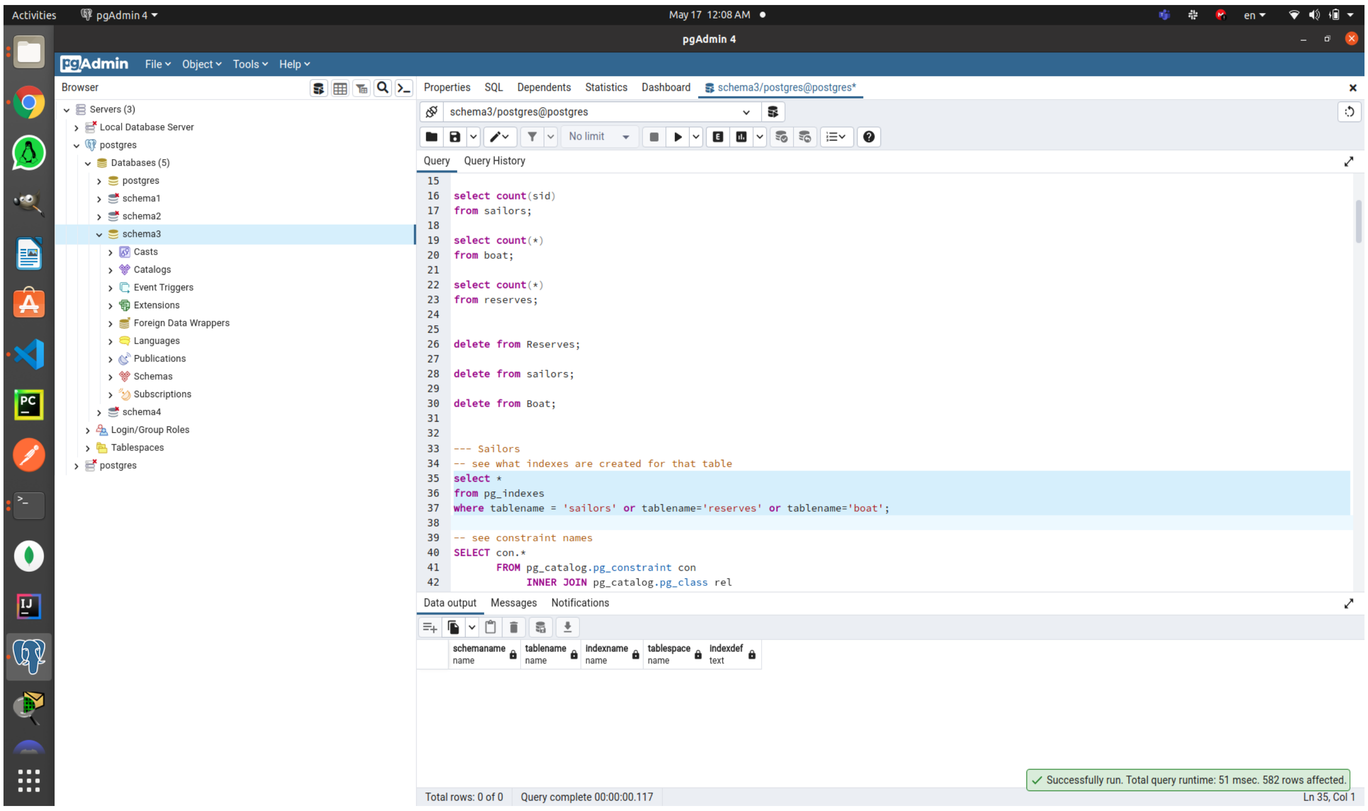
```
count
bigint
1   177
```

Successfully run. Total query runtime: 51 msec. 582 rows affected.

Total rows: 1 of 1   Query complete 00:00:00.107   Ln 170, Col 1

Report

1. given query without an index,

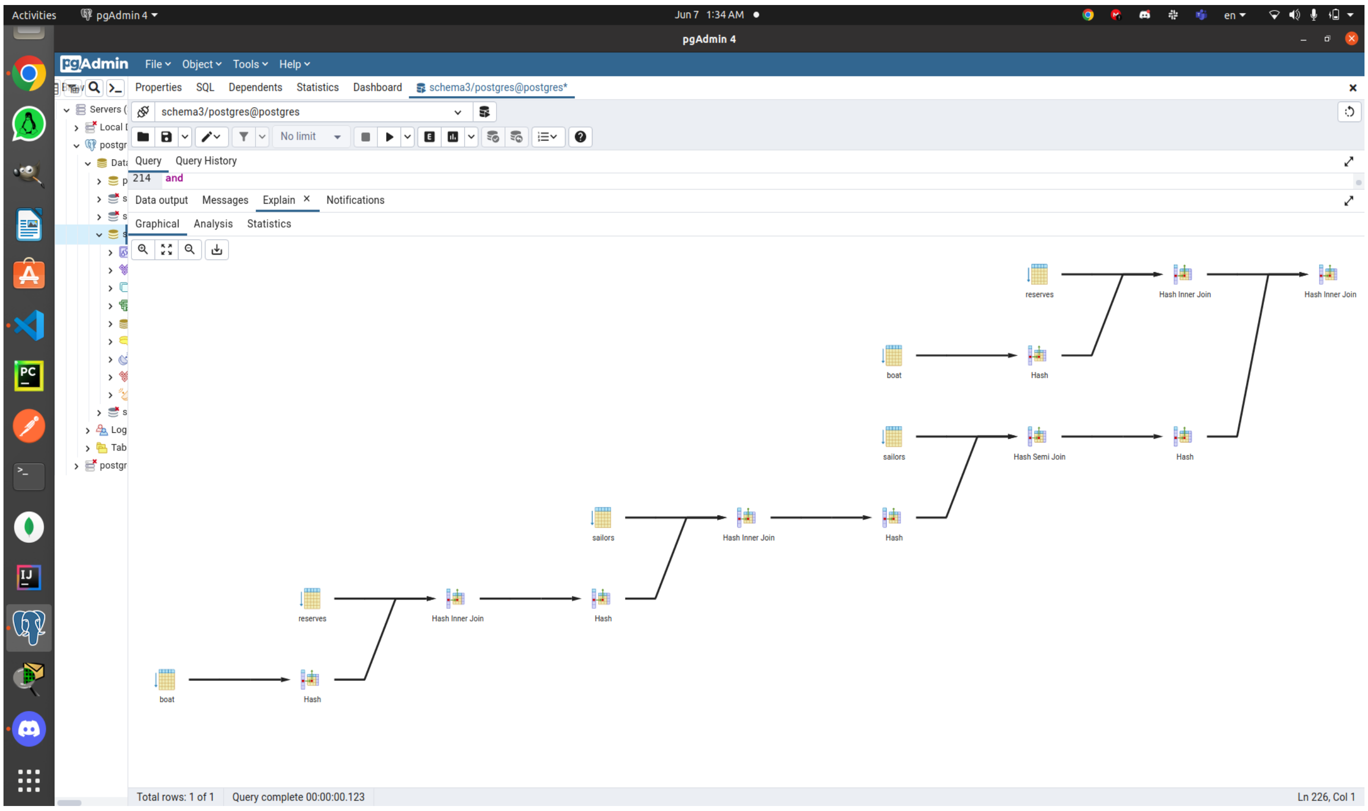


```
select count(sid)
from sailors;

select count(*)
from boat;

select count(*)
from reserves;


delete from Reserves;

delete from sailors;

delete from Boat;


--- Sailors
-- see what indexes are created for that table
select *
from pg_indexes
where tablename = 'sailors' or tablename='reserves' or tablename='boat';

-- see constraint names
SELECT con.*
       FROM pg_catalog.pg_constraint con
            INNER JOIN pg_catalog.pg_class rel
```

schemaname name	tablename name	indexname name	tablespace name	indexdef text

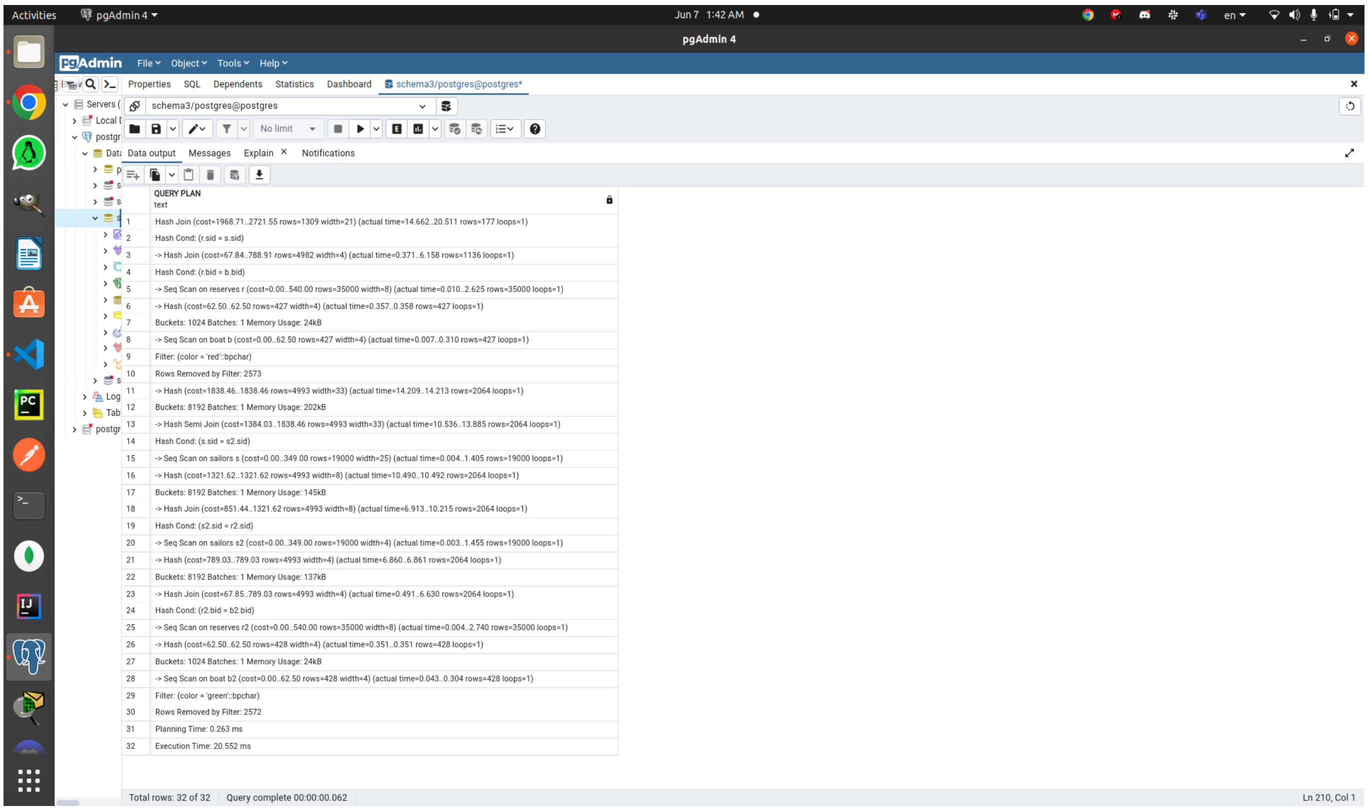
Successfully run. Total query runtime: 51 msec. 582 rows affected.

Total rows: 0 of 0 Query complete 00:00:00.117 Ln 35, Col 1



```
214 and
```

Total rows: 1 of 1 Query complete 00:00:00.123 Ln 226, Col 1



```
QUERY PLAN
text
Hash Join (cost=1968.71..2721.55 rows=1309 width=21) (actual time=14.662..20.511 rows=177 loops=1)
Hash Cond: (r.sid = s.sid)
-> Hash Join (cost=67.84..788.91 rows=4982 width=4) (actual time=0.371..6.158 rows=1136 loops=1)
Hash Cond: (r.bid = b.bid)
-> Seq Scan on reserves r (cost=0.00..540.00 rows=35000 width=8) (actual time=0.010..2.625 rows=35000 loops=1)
-> Hash (cost=62.50..62.50 rows=427 width=4) (actual time=0.357..0.358 rows=427 loops=1)
Buckets: 1024 Batches: 1 Memory Usage: 24kB
-> Seq Scan on boat b (cost=0.00..62.50 rows=427 width=4) (actual time=0.007..0.310 rows=427 loops=1)
Filter: (color = 'red'::bpchar)
Rows Removed by Filter: 2573
-> Hash (cost=1838.46..1838.46 rows=4993 width=33) (actual time=14.209..14.213 rows=2064 loops=1)
Buckets: 8192 Batches: 1 Memory Usage: 202kB
-> Hash Semi Join (cost=1384.03..1838.46 rows=4993 width=33) (actual time=10.536..13.885 rows=2064 loops=1)
Hash Cond: (s.sid = s2.sid)
-> Seq Scan on sailors s (cost=0.00..349.00 rows=19000 width=25) (actual time=0.004..1.405 rows=19000 loops=1)
-> Hash (cost=1321.62..1321.62 rows=4993 width=8) (actual time=10.490..10.492 rows=2064 loops=1)
Buckets: 8192 Batches: 1 Memory Usage: 145kB
-> Hash Join (cost=851.44..1321.62 rows=4993 width=8) (actual time=6.913..10.215 rows=2064 loops=1)
Hash Cond: (s2.sid = r2.sid)
-> Seq Scan on sailors s2 (cost=0.00..349.00 rows=19000 width=4) (actual time=0.003..1.455 rows=19000 loops=1)
-> Hash (cost=789.03..789.03 rows=4993 width=4) (actual time=6.860..6.861 rows=2064 loops=1)
Buckets: 8192 Batches: 1 Memory Usage: 137kB
-> Hash Join (cost=67.85..789.03 rows=4993 width=4) (actual time=0.491..6.630 rows=2064 loops=1)
Hash Cond: (r2.bid = b2.bid)
-> Seq Scan on reserves r2 (cost=0.00..540.00 rows=35000 width=8) (actual time=0.004..2.740 rows=35000 loops=1)
-> Hash (cost=62.50..62.50 rows=428 width=4) (actual time=0.351..0.351 rows=428 loops=1)
Buckets: 1024 Batches: 1 Memory Usage: 24kB
-> Seq Scan on boat b2 (cost=0.00..62.50 rows=428 width=4) (actual time=0.043..0.304 rows=428 loops=1)
Filter: (color = 'green'::bpchar)
Rows Removed by Filter: 2572
Planning Time: 0.263 ms
Execution Time: 20.552 ms
Total rows: 32 of 32    Query complete 00:00:00.062
```

Explanation :

- First , a Sequential Scan occured on Reserves Table to filter the reserved boats with bid 103 and it costs 0..627 rows,and read through 35000 Rows and 582 Rows are left after the filtration of the where condition.
- Second , a Hash Table was built on the run on r.bid of the 582 Rows and it costed 627.50...627.50 rows,this resulted with 1024 Buckets created ,and Memory Usage of 29 KB.
- Thirdly , a Sequential Scan occured on Sailors Table to be able to hash each row's sid to the Reserves's buckets to full inner join on the condition and it costed 00..349 rows, and it read through 19000 rows.
- Fourthly, a Hash Semi Join occurred to produce the result set of the condition of s.sid = r.sid , and the reason it is Hash Semi Join In the first query, only the r.sid needs to be saved from the reserves into the hash table, because that is the only data needed to implement the semi-join , it costs 634.77 ... 1084.60 rows.
- Execution Time : 10.462 ms

2. given query with B+ trees indices only,

-

3. given query with hash indices only,

-

4. given query with BRIN indices only,

-

5. given query with mixed indices (any mix of your choice).

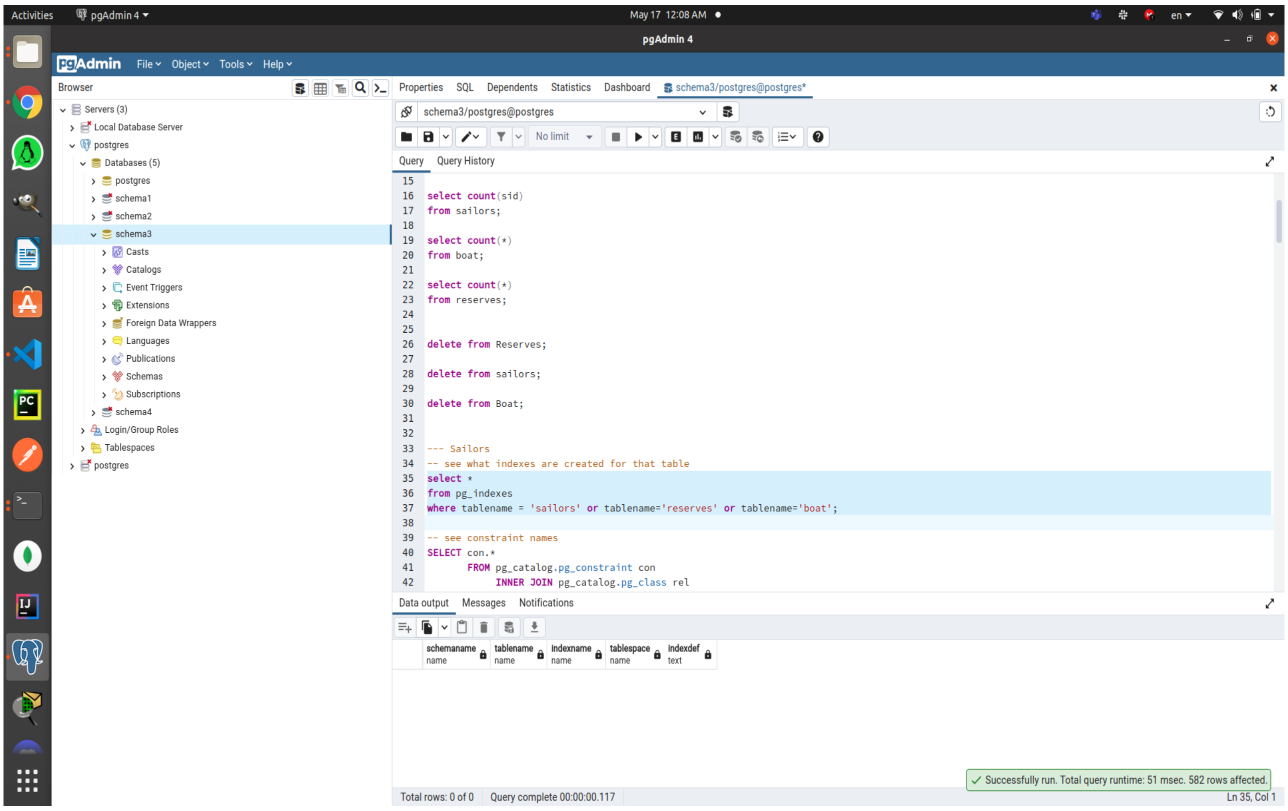
-

Optimized Query

```
select s.sname
from sailors s
where exists
        (
         select rTotal.sid
            from (select r1.sid
             from
                (select r.sid
                from reserves r
                 where  exists
                    (select bid
                     from boat b
                     where color = 'green' and r.bid =b.bid )
                )as r1
             inner join
                (select r.sid
                from reserves r
                 where  exists
                    (select bid
                     from boat b
                     where color = 'red' and r.bid =b.bid )
                ) as r2
             on r2.sid = r1.sid  ) as rTotal
          where rTotal.sid=s.sid
)
```

Report

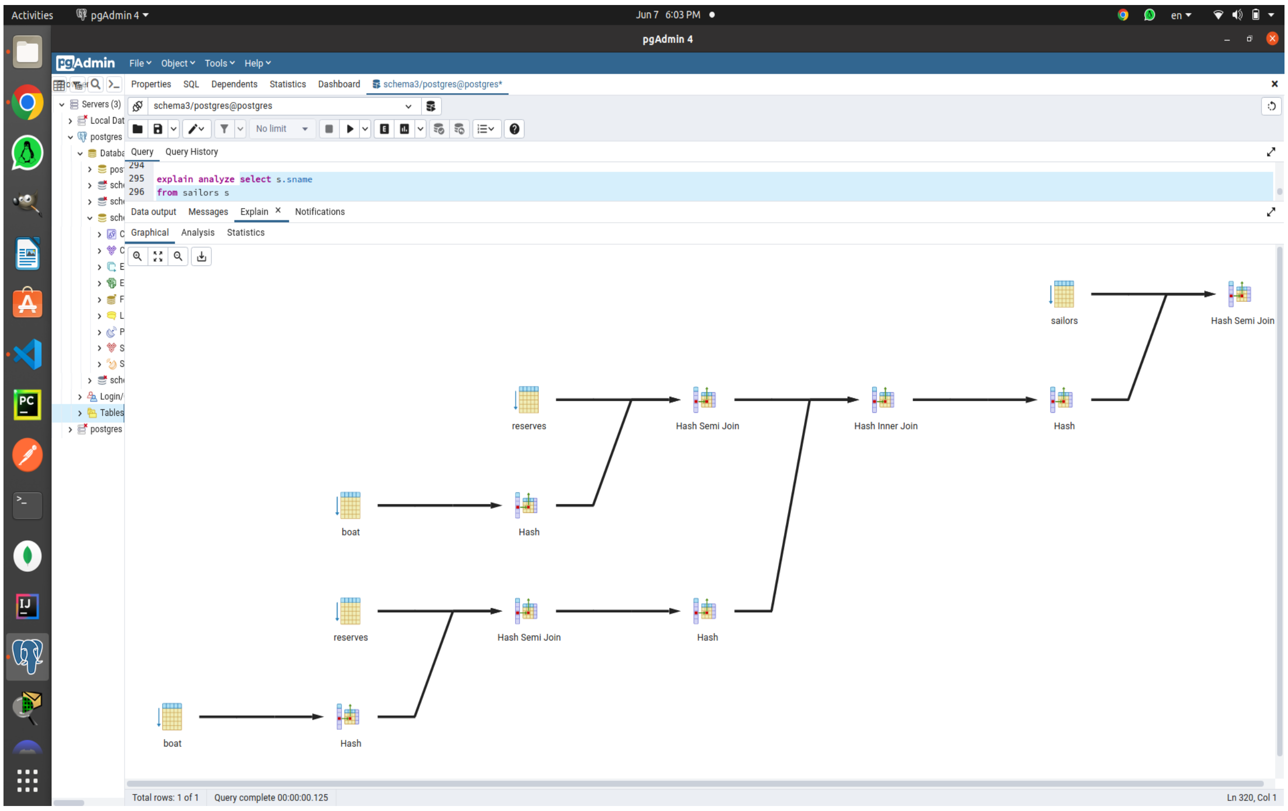
1. given query without an index,



```

select count(sid)
from sailors;

select count(*)
from boat;

select count(*)
from reserves;


delete from Reserves;

delete from sailors;

delete from Boat;


--- Sailors
-- see what indexes are created for that table
select *
from pg_indexes
where tablename = 'sailors' or tablename='reserves' or tablename='boat';

-- see constraint names
SELECT con.*
       FROM pg_catalog.pg_constraint con
            INNER JOIN pg_catalog.pg_class rel
```

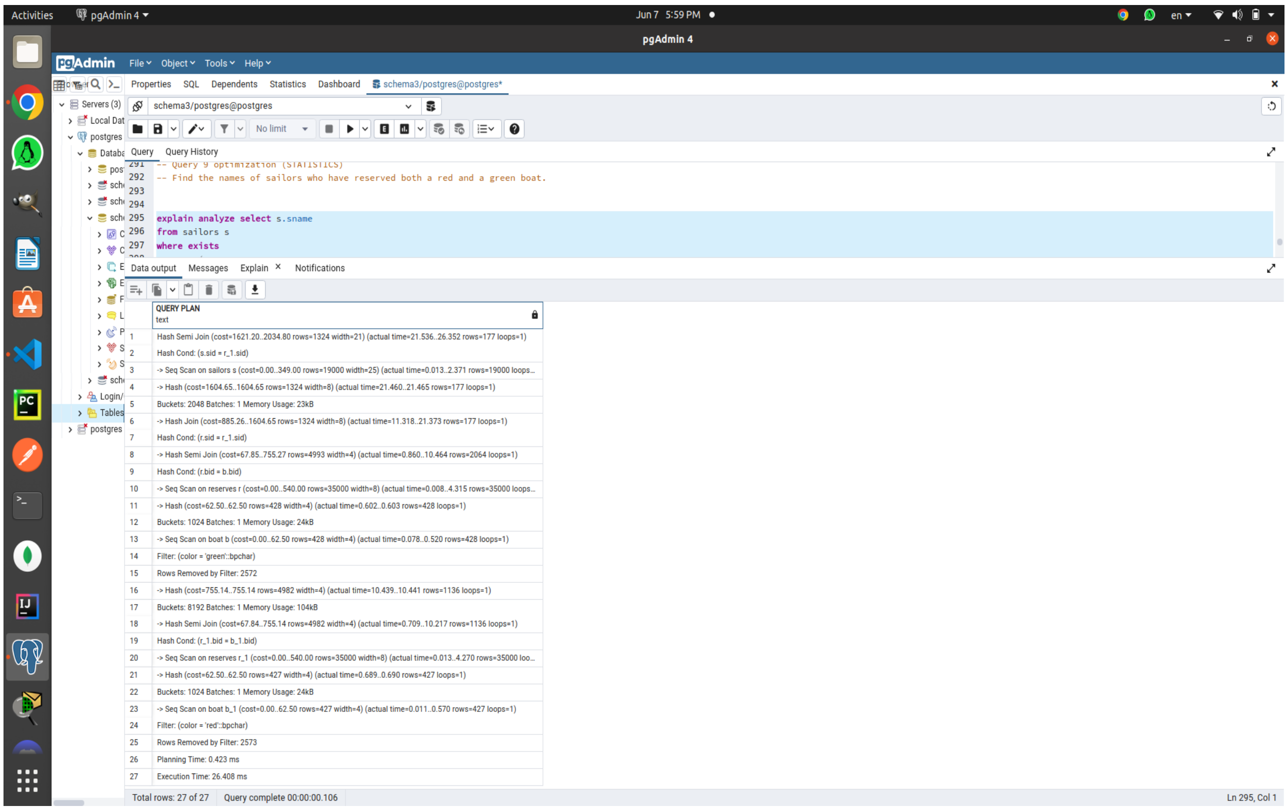
schemaname name	tablename name	indexname name	tablespace name	indexdef text

```
Successfully run. Total query runtime: 51 msec. 582 rows affected.
Total rows: 0 of 0    Query complete 00:00:00.117    Ln 35, Col 1
```



```
explain analyze select s.sname
from sailors s
```

Total rows: 1 of 1 Query complete 00:00:00.125



```
-- Query 9 optimization (STATISTICS)
-- Find the names of sailors who have reserved both a red and a green boat.


explain analyze select s.sname
from sailors s
where exists
```

	QUERY PLAN text
1	Hash Semi Join (cost=1621.20..2034.80 rows=1324 width=21) (actual time=21.536..26.352 rows=177 loops=1)
2	Hash Cond: (s.sid = r_1.sid)
3	-> Seq Scan on sailors s (cost=0.00..349.00 rows=19000 width=25) (actual time=0.013..2.371 rows=19000 loops...
4	-> Hash (cost=1604.65..1604.65 rows=1324 width=8) (actual time=21.460..21.465 rows=177 loops=1)
5	Buckets: 2048 Batches: 1 Memory Usage: 23kB
6	-> Hash Join (cost=885.26..1604.65 rows=1324 width=8) (actual time=11.318..21.373 rows=177 loops=1)
7	Hash Cond: (r.sid = r_1.sid)
8	-> Hash Semi Join (cost=67.85..755.27 rows=4993 width=4) (actual time=0.860..10.464 rows=2064 loops=1)
9	Hash Cond: (r.bid = b.bid)
10	-> Seq Scan on reserves r (cost=0.00..540.00 rows=35000 width=8) (actual time=0.008..4.315 rows=35000 loops...
11	-> Hash (cost=62.50..62.50 rows=428 width=4) (actual time=0.602..0.603 rows=428 loops=1)
12	Buckets: 1024 Batches: 1 Memory Usage: 24kB
13	-> Seq Scan on boat b (cost=0.00..62.50 rows=428 width=4) (actual time=0.078..0.520 rows=428 loops=1)
14	Filter: (color = 'green'::bpchar)
15	Rows Removed by Filter: 2572
16	-> Hash (cost=755.14..755.14 rows=4982 width=4) (actual time=10.439..10.441 rows=1136 loops=1)
17	Buckets: 8192 Batches: 1 Memory Usage: 104kB
18	-> Hash Semi Join (cost=67.84..755.14 rows=4982 width=4) (actual time=0.709..10.217 rows=1136 loops=1)
19	Hash Cond: (r_1.bid = b_1.bid)
20	-> Seq Scan on reserves r_1 (cost=0.00..540.00 rows=35000 width=8) (actual time=0.013..4.270 rows=35000 loo...
21	-> Hash (cost=62.50..62.50 rows=427 width=4) (actual time=0.689..0.690 rows=427 loops=1)
22	Buckets: 1024 Batches: 1 Memory Usage: 24kB
23	-> Seq Scan on boat b_1 (cost=0.00..62.50 rows=427 width=4) (actual time=0.011..0.570 rows=427 loops=1)
24	Filter: (color = 'red'::bpchar)
25	Rows Removed by Filter: 2573
26	Planning Time: 0.423 ms
27	Execution Time: 26.408 ms

Total rows: 27 of 27 Query complete 00:00:00.106

Explanation :

- First , a Sequential Scan occured on Reserves Table to filter the reserved boats with bid 103 and it costs 0..627 rows,and read through 35000 Rows and 582 Rows are left after the filtration of the where condition.
- Second , a Hash Table was built on the run on r.bid of the 582 Rows and it costed 627.50...627.50 rows,this resulted with 1024 Buckets created ,and Memory Usage of 29 KB.
- Thirdly , a Sequential Scan occured on Sailors Table to be able to hash each row's sid to the Reserves's buckets to full inner join on the condition and it costed 00..349 rows, and it read through

  19000 rows.
- Fourthly, a Hash Semi Join occurred to produce the result set of the condition of s.sid = r.sid , and the reason it is Hash Semi Join In the first query, only the r.sid needs to be saved from the reserves into the hash table, because that is the only data needed to implement the semi-join , it costs 634.77 ... 1084.60 rows.
- Execution Time : 10.462 ms

2. given query with B+ trees indices only,

-

3. given query with hash indices only,

-

4. given query with BRIN indices only,

-

5. given query with mixed indices (any mix of your choice)

-