

Query 7

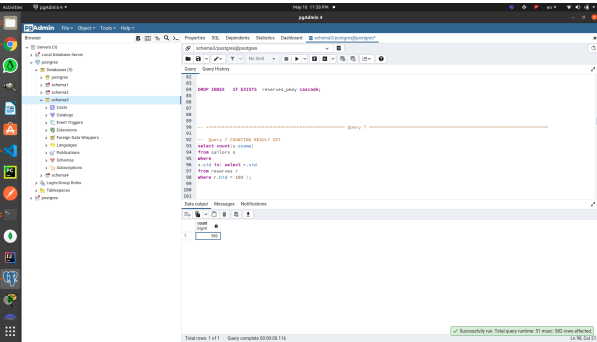
- Find the names of sailors who have reserved boat 103.

Original Query

```
select s.sname
from sailors s
where
s.sid in( select r.sid
from reserves r
where r.bid = 103 );
```

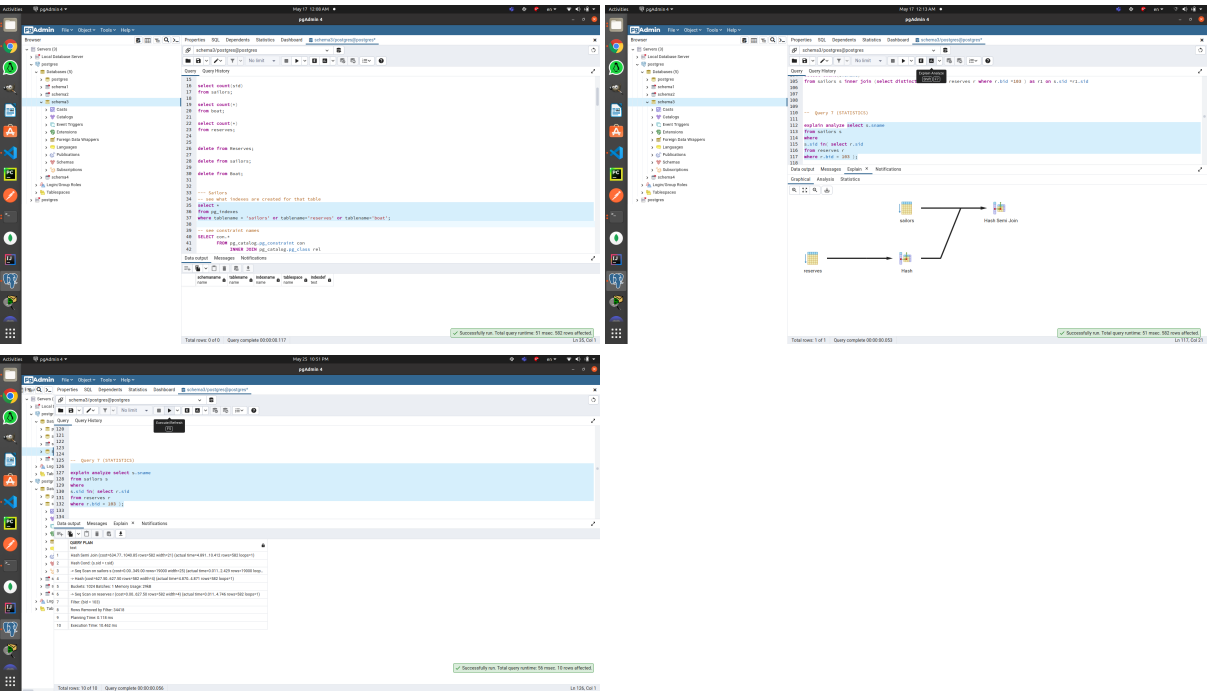
Result Set

- 582 Rows



Report

- given query without an index,



Explanation :

- First , a Sequential Scan occurred 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 occurred 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 ... 1040.85 rows.
- Execution Time : 10.462 ms

1. 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).

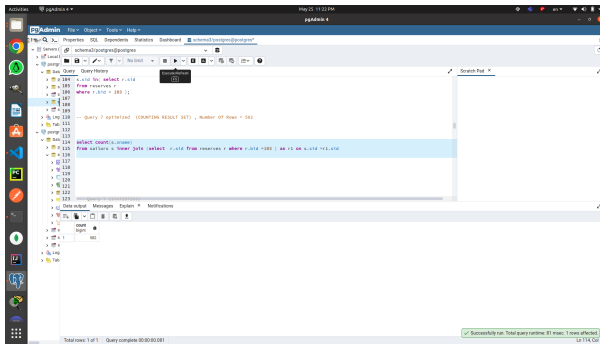
•

Optimized Query

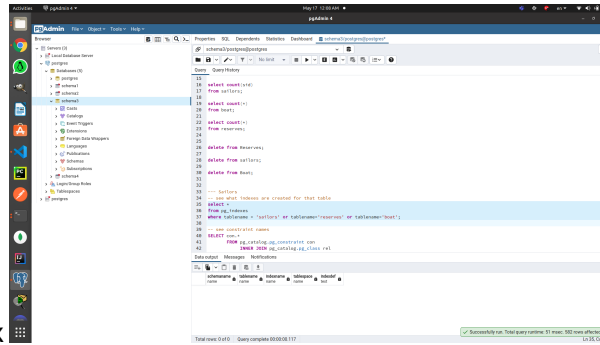
```
select s.sname
from sailors s
where
s.sid in( select r.sid
from reserves r
where r.bid = 103 );
```

Result Set

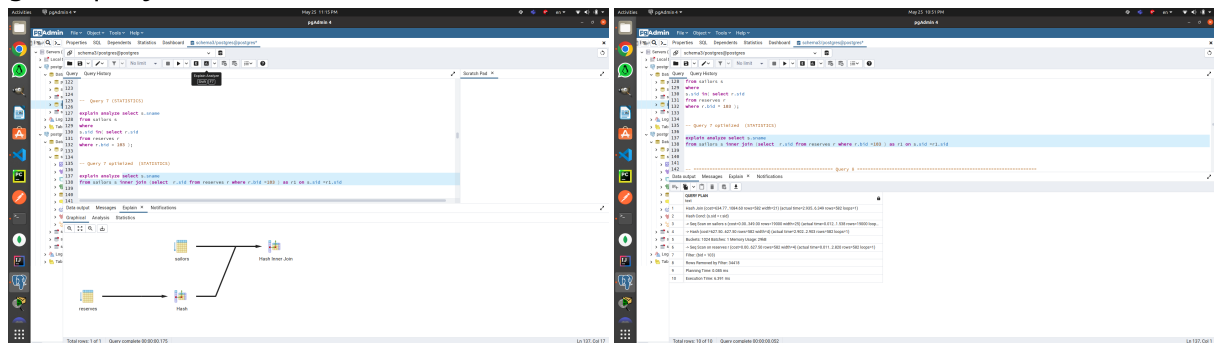
- 582 Rows



Report



1. given query without an index



Explanation :

- First, a Sequential Scan occurred 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 occurred 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).

-