Schema 3

Inserted 9000 sailors, 3000 boats, and 15000 reserves

The sailors have ids from 1 up to 9000.

The boats have ids from 1 up to 3000.There are four colors distributed equally on them.

The reserves are as follows each boat is assigned to (3:5) sailors.

Query 7:

Scenario 1: No index is used:

QUERY PLAN

Hash Semi Join (cost=269.56..458.26 rows=5 width=21) (actual time=3.805..7.400 rows=4 loops=1)

Hash Cond: (s.sid = r.sid)

-> Seq Scan on sailors s (cost=0.00..165.00 rows=9000 width=25) (actual time=0.033..1.778 rows=9000 loops=1)

-> Hash (cost=269.50..269.50 rows=5 width=4) (actual time=3.699..3.699 rows=4 loops=1)

Buckets: 1024 Batches: 1 Memory Usage: 9kB

-> Seq Scan on reserves r (cost=0.00..269.50 rows=5 width=4) (actual time=0.046..3.675 rows=4 loops=1)

Filter: (bid = 103)

Rows Removed by Filter: 14996

Planning time: 0.546 ms

Execution time: 7.463 ms

(10 rows)

Scenario 2: BTree index on reserves.bid

QUERY PLAN

Hash Semi Join (cost=20.74..209.44 rows=5 width=21) (actual time=0.082..1.101 rows=4 loops=1)

Hash Cond: (s.sid = r.sid)

-> Seq Scan on sailors s (cost=0.00..165.00 rows=9000 width=25) (actual time=0.012..0.550 rows=9000 loops=1)

-> Hash (cost=20.68..20.68 rows=5 width=4) (actual time=0.050..0.050 rows=4 loops=1)

Buckets: 1024 Batches: 1 Memory Usage: 9kB

-> Bitmap Heap Scan on reserves r (cost=4.32..20.68 rows=5 width=4) (actual time=0.041..0.044 rows=4 loops=1) Recheck Cond: (bid = 103)

Heap Blocks: exact=4

-> Bitmap Index Scan on reserves_bid_idx (cost=0.00..4.32 rows=5 width=0) (actual time=0.037..0.038 rows=4 loops=1)

Index Cond: (bid = 103)

Planning time: 1.661 ms

Execution time: 1.147 ms

(12 rows)

Scenario 3: BTree index on reserves.bid and hash index on sailors.sid

QUERY PLAN

Nested Loop (cost=20.69..60.88 rows=5 width=21) (actual time=0.030..0.037 rows=4 loops=1)

-> HashAggregate (cost=20.69..20.74 rows=5 width=4) (actual time=0.019..0.019 rows=4 loops=1)

Group Key: r.sid

-> Bitmap Heap Scan on reserves r (cost=4.32..20.68 rows=5 width=4) (actual time=0.010..0.014 rows=4 loops=1)

Recheck Cond: (bid = 103)

Heap Blocks: exact=4

-> Bitmap Index Scan on reserves_bid_idx (cost=0.00..4.32 rows=5 width=0) (actual time=0.007..0.007 rows=4 loops=1)

Index Cond: (bid = 103)

-> Index Scan using sailors_sid_idx on sailors s (cost=0.00..8.02 rows=1 width=25) (actual time=0.004..0.004 rows=1 loops=4)

Index Cond: (sid = r.sid)

Planning time: 0.854 ms

Execution time: 0.078 ms

(12 rows)

Scenario 4: hash index on reserves.bid and hash index on sailors.sid

QUERY PLAN

Nested Loop (cost=20.41..60.60 rows=5 width=21) (actual time=0.023..0.031 rows=4 loops=1)

-> HashAggregate (cost=20.41..20.46 rows=5 width=4) (actual time=0.017..0.018 rows=4 loops=1)

Group Key: r.sid

-> Bitmap Heap Scan on reserves r (cost=4.04..20.40 rows=5 width=4) (actual time=0.012..0.015 rows=4 loops=1)

Recheck Cond: (bid = 103)

Heap Blocks: exact=4

-> Bitmap Index Scan on reserves_bid_idx (cost=0.00..4.04 rows=5 width=0) (actual time=0.008..0.008 rows=4 loops=1)

Index Cond: (bid = 103)

-> Index Scan using sailors_sid_idx on sailors s (cost=0.00..8.02 rows=1 width=25) (actual time=0.002..0.003 rows=1 loops=4)

Index Cond: (sid = r.sid)

Planning time: 0.910 ms

Execution time: 0.077 ms

(12 rows)

Scenario 5: bitmap index on reserves.bid and hash index on sailors.sid

QUERY PLAN

Nested Loop (cost=28.41..68.60 rows=5 width=21) (actual time=0.030..0.039 rows=4 loops=1)

-> HashAggregate (cost=28.41..28.46 rows=5 width=4) (actual time=0.024..0.024 rows=4 loops=1)

Group Key: r.sid

-> Bitmap Heap Scan on reserves r (cost=12.04..28.40 rows=5 width=4) (actual time=0.017..0.020 rows=4 loops=1)

Recheck Cond: (bid = 103)

Heap Blocks: exact=4

-> Bitmap Index Scan on reserves_bid_idx (cost=0.00..12.04 rows=5 width=0) (actual time=0.015..0.015 rows=4 loops=1)

Index Cond: (bid = 103)

-> Index Scan using sailors_sid_idx on sailors s (cost=0.00..8.02 rows=1 width=25) (actual time=0.003..0.003 rows=1 loops=4)

Index Cond: (sid = r.sid)

Planning time: 0.922 ms

Execution time: 0.082 ms

(12 rows)

Conclusion of query 7:

The best performance was for Scenario 4: hash index on reserves.bid and hash index on sailors.sid.

That's because the hash index is the best option for exact queries, in our case,

```
(select r.sid
from reserves r
where r.bid = 103)
```

Moreover, the bitmap index is not the best option in our case reserves.bid column does not have a significant number of duplicates (At most five insertions for each bid from 1:3000 is not significant).In addition to the long sequence of bits 15000.

The B-tree is good for exact queries, but it is O (log n) whereas the hash index is O (1).

Since the output from the bitmap index scan is 5 rows, it can also be treated as an exact query and a hash index can be used on sailors.sid.

Query 8:

Scenario 1: No index is used:

QUERY PLAN

Hash Semi Join (cost=429.84..664.88 rows=3750 width=21) (actual time=3.501..5.276 rows=3346 loops=1)

Hash Cond: (s.sid = r.sid)

-> Seq Scan on sailors s (cost=0.00..165.00 rows=9000 width=25) (actual time=0.044..0.822 rows=9000 loops=1)

-> Hash (cost=382.97..382.97 rows=3750 width=4) (actual time=3.436..3.436 rows=3730 loops=1)

Buckets: 4096 Batches: 1 Memory Usage: 164kB

-> Hash Semi Join (cost=69.88..382.97 rows=3750 width=4) (actual time=0.496..3.053 rows=3730 loops=1)

Hash Cond: (r.bid = b.bid)

-> Seq Scan on reserves r (cost=0.00..232.00 rows=15000 width=8) (actual time=0.017..1.190 rows=15000 loops=1)

-> Hash (cost=60.50..60.50 rows=750 width=4) (actual time=0.471..0.471 rows=750 loops=1)

Buckets: 1024 Batches: 1 Memory Usage: 35kB

-> Seq Scan on boat b (cost=0.00..60.50 rows=750 width=4) (actual time=0.024..0.397 rows=750 loops=1)

Filter: ((color)::text = 'red'::text)

Rows Removed by Filter: 2250

Planning time: 0.314 ms

Execution time: 5.445 ms

(15 rows)

Scenario 2: Bitmap index on boat.color

QUERY PLAN

Hash Semi Join (cost=415.53..650.56 rows=3750 width=21) (actual time=2.809..4.322 rows=3346 loops=1)

Hash Cond: (s.sid = r.sid)

-> Seq Scan on sailors s (cost=0.00..165.00 rows=9000 width=25) (actual time=0.010..0.581 rows=9000 loops=1)

-> Hash (cost=368.66..368.66 rows=3750 width=4) (actual time=2.783..2.783 rows=3730 loops=1)

Buckets: 4096 Batches: 1 Memory Usage: 164kB

-> Hash Semi Join (cost=55.56..368.66 rows=3750 width=4) (actual time=0.237..2.419 rows=3730 loops=1)

Hash Cond: (r.bid = b.bid)

-> Seq Scan on reserves r (cost=0.00..232.00 rows=15000 width=8) (actual time=0.009..0.825 rows=15000 loops=1)

-> Hash (cost=46.19..46.19 rows=750 width=4) (actual time=0.220..0.220 rows=750 loops=1)

Buckets: 1024 Batches: 1 Memory Usage: 35kB

-> Bitmap Heap Scan on boat b (cost=13.81..46.19 rows=750 width=4) (actual time=0.059..0.160 rows=750 loops=1)

Recheck Cond: ((color)::text = 'red'::text)

Heap Blocks: exact=23

-> Bitmap Index Scan on boat_color_idx (cost=0.00..13.63 rows=750 width=0) (actual time=0.053..0.053 rows=750 loops=1)

Index Cond: ((color)::text = 'red'::text)

Planning time: 1.173 ms

Execution time: 4.470 ms

(17 rows)

Scenario 3: Hash index on boat.color

QUERY PLAN

Hash Semi Join (cost=429.84..664.88 rows=3750 width=21) (actual time=2.958..4.476 rows=3346 loops=1)

Hash Cond: (s.sid = r.sid)

-> Seq Scan on sailors s (cost=0.00..165.00 rows=9000 width=25) (actual time=0.011..0.578 rows=9000 loops=1)

-> Hash (cost=382.97..382.97 rows=3750 width=4) (actual time=2.927..2.927 rows=3730 loops=1)

Buckets: 4096 Batches: 1 Memory Usage: 164kB

-> Hash Semi Join (cost=69.88..382.97 rows=3750 width=4) (actual time=0.378..2.551 rows=3730 loops=1)

Hash Cond: (r.bid = b.bid)

-> Seq Scan on reserves r (cost=0.00..232.00 rows=15000 width=8) (actual time=0.009..0.815 rows=15000 loops=1)

-> Hash (cost=60.50..60.50 rows=750 width=4) (actual time=0.363..0.363 rows=750 loops=1)

Buckets: 1024 Batches: 1 Memory Usage: 35kB

-> Seq Scan on boat b (cost=0.00..60.50 rows=750 width=4) (actual time=0.009..0.293 rows=750 loops=1)

Filter: ((color)::text = 'red'::text)

Rows Removed by Filter: 2250

Planning time: 0.985 ms

Execution time: 4.607 ms

(15 rows)

Scenario 4: B-tree index on boat.color

QUERY PLAN

Hash Semi Join (cost=419.81..654.84 rows=3750 width=21) (actual time=2.800..4.269 rows=3346 loops=1)

Hash Cond: (s.sid = r.sid)

-> Seq Scan on sailors s (cost=0.00..165.00 rows=9000 width=25) (actual time=0.010..0.537 rows=9000 loops=1)

-> Hash (cost=372.94..372.94 rows=3750 width=4) (actual time=2.774..2.774 rows=3730 loops=1)

Buckets: 4096 Batches: 1 Memory Usage: 164kB

-> Hash Semi Join (cost=59.84..372.94 rows=3750 width=4) (actual time=0.266..2.389 rows=3730 loops=1)

Hash Cond: (r.bid = b.bid)

-> Seq Scan on reserves r (cost=0.00..232.00 rows=15000 width=8) (actual time=0.009..0.802 rows=15000 loops=1)

-> Hash (cost=50.47..50.47 rows=750 width=4) (actual time=0.250..0.250 rows=750 loops=1)

Buckets: 1024 Batches: 1 Memory Usage: 35kB

-> Bitmap Heap Scan on boat b (cost=18.09..50.47 rows=750 width=4) (actual time=0.091..0.179 rows=750 loops=1)

Recheck Cond: ((color)::text = 'red'::text)

Heap Blocks: exact=23

-> Bitmap Index Scan on boat_color_idx (cost=0.00..17.91 rows=750 width=0) (actual time=0.085..0.085 rows=750 loops=1)

Index Cond: ((color)::text = 'red'::text)

Planning time: 1.527 ms

Execution time: 4.421 ms

(17 rows)

Conclusion of query 8:

The best performance was for Scenario 2: Bitmap index on boat.color. That's because there is only four values in the column boat.color. The bitmap index scan is the best option when there is a significant number of duplicates in the column, which is the case here.

Query 9:

Scenario 1: No index is used

QUERY PLAN

Hash Join (cost=1083.97..1439.40 rows=1562 width=21) (actual time=7.383..9.925 rows=739 loops=1)

Hash Cond: (r.sid = s.sid)

-> Hash Join (cost=69.88..395.63 rows=3750 width=4) (actual time=0.382..2.559 rows=3730 loops=1)

Hash Cond: (r.bid = b.bid)

-> Seq Scan on reserves r (cost=0.00..232.00 rows=15000 width=8) (actual time=0.013..0.781 rows=15000 loops=1)

-> Hash (cost=60.50..60.50 rows=750 width=4) (actual time=0.360..0.360 rows=750 loops=1)

Buckets: 1024 Batches: 1 Memory Usage: 35kB

-> Seq Scan on boat b (cost=0.00..60.50 rows=750 width=4) (actual time=0.009..0.285 rows=750 loops=1)

Filter: ((color)::text = 'red'::text)

Rows Removed by Filter: 2250

-> Hash (cost=967.22..967.22 rows=3750 width=33) (actual time=6.981..6.981 rows=3395 loops=1)

Buckets: 4096 Batches: 1 Memory Usage: 258kB

-> Hash Semi Join (cost=736.88..967.22 rows=3750 width=33) (actual time=4.898..6.490 rows=3395 loops=1)

Hash Cond: (s.sid = s2.sid)

-> Seq Scan on sailors s (cost=0.00..165.00 rows=9000 width=25) (actual time=0.008..0.522 rows=9000 loops=1)

-> Hash (cost=690.00..690.00 rows=3750 width=8) (actual time=4.872..4.872 rows=3749 loops=1)

Buckets: 4096 Batches: 1 Memory Usage: 179kB

-> Hash Join (cost=442.50..690.00 rows=3750 width=8) (actual time=2.892..4.465 rows=3749 loops=1)

Hash Cond: (s2.sid = r2.sid)

-> Seq Scan on sailors s2 (cost=0.00..165.00 rows=9000 width=4) (actual time=0.006..0.517 rows=9000 loops=1)

-> Hash (cost=395.63..395.63 rows=3750 width=4) (actual time=2.872..2.872 rows=3749 loops=1)

Buckets: 4096 Batches: 1 Memory Usage: 164kB

-> Hash Join (cost=69.88..395.63 rows=3750 width=4) (actual time=0.335..2.502 rows=3749 loops=1)

Hash Cond: (r2.bid = b2.bid)

-> Seq Scan on reserves r2 (cost=0.00..232.00 rows=15000 width=8) (actual time=0.006..0.763 rows=15000 loops=1)

-> Hash (cost=60.50..60.50 rows=750 width=4) (actual time=0.323..0.323 rows=750 loops=1)

Buckets: 1024 Batches: 1 Memory Usage: 35kB

-> Seq Scan on boat b2 (cost=0.00..60.50 rows=750 width=4) (actual time=0.006..0.249 rows=750 loops=1)

Filter: ((color)::text = 'green'::text)

Rows Removed by Filter: 2250

Planning time: 0.336 ms

Execution time: 10.175 ms

(32 rows)

Scenario 2: Bitmap index on boat.color

QUERY PLAN

Hash Join (cost=1055.34..1410.78 rows=1562 width=21) (actual time=7.253..9.921 rows=739 loops=1)

Hash Cond: (r.sid = s.sid)

-> Hash Join (cost=55.56..381.31 rows=3750 width=4) (actual time=0.246..2.550 rows=3730 loops=1)

Hash Cond: (r.bid = b.bid)

-> Seq Scan on reserves r (cost=0.00..232.00 rows=15000 width=8) (actual time=0.009..0.888 rows=15000 loops=1)

-> Hash (cost=46.19..46.19 rows=750 width=4) (actual time=0.233..0.233 rows=750 loops=1)

Buckets: 1024 Batches: 1 Memory Usage: 35kB

-> Bitmap Heap Scan on boat b (cost=13.81..46.19 rows=750 width=4) (actual time=0.060..0.162 rows=750 loops=1)

Recheck Cond: ((color)::text = 'red'::text)

Heap Blocks: exact=23

-> Bitmap Index Scan on boat_color_idx (cost=0.00..13.63 rows=750 width=0) (actual time=0.053..0.053 rows=750 loops=1)

Index Cond: ((color)::text = 'red'::text)

-> Hash (cost=952.91..952.91 rows=3750 width=33) (actual time=6.994..6.994 rows=3395 loops=1)

Buckets: 4096 Batches: 1 Memory Usage: 258kB

-> Hash Semi Join (cost=722.56..952.91 rows=3750 width=33) (actual time=4.921..6.501 rows=3395 loops=1)

Hash Cond: (s.sid = s2.sid)

-> Seq Scan on sailors s (cost=0.00..165.00 rows=9000 width=25) (actual time=0.008..0.519 rows=9000 loops=1)

-> Hash (cost=675.69..675.69 rows=3750 width=8) (actual time=4.900..4.900 rows=3749 loops=1)

Buckets: 4096 Batches: 1 Memory Usage: 179kB

-> Hash Join (cost=428.19..675.69 rows=3750 width=8) (actual time=2.866..4.500 rows=3749 loops=1)

Hash Cond: (s2.sid = r2.sid)

-> Seq Scan on sailors s2 (cost=0.00..165.00 rows=9000 width=4) (actual time=0.005..0.573 rows=9000 loops=1)

-> Hash (cost=381.31..381.31 rows=3750 width=4) (actual time=2.847..2.847 rows=3749 loops=1)

Buckets: 4096 Batches: 1 Memory Usage: 164kB

-> Hash Join (cost=55.56..381.31 rows=3750 width=4) (actual time=0.224..2.468 rows=3749 loops=1)

Hash Cond: (r2.bid = b2.bid)

-> Seq Scan on reserves r2 (cost=0.00..232.00 rows=15000 width=8) (actual time=0.006..0.880 rows=15000 loops=1)

-> Hash (cost=46.19..46.19 rows=750 width=4) (actual time=0.210..0.210 rows=750 loops=1)

Buckets: 1024 Batches: 1 Memory Usage: 35kB

-> Bitmap Heap Scan on boat b2 (cost=13.81..46.19 rows=750 width=4) (actual time=0.057..0.139 rows=750 loops=1)

Recheck Cond: ((color)::text = 'green'::text)

Heap Blocks: exact=23

-> Bitmap Index Scan on boat_color_idx (cost=0.00..13.63 rows=750 width=0) (actual time=0.054..0.054 rows=750 loops=1)

Index Cond: ((color)::text = 'green'::text)

Planning time: 1.206 ms

Execution time: 10.155 ms

(36 rows)

Scenario 3:

(Hash index on reserves.bid, Hash index on reserves.sid,

Hash index on boat.bid, Hash index on sailors.sid, Bitmap index on boat.color)

QUERY PLAN

Nested Loop (cost=742.75..1186.60 rows=1562 width=21) (actual time=7.564..10.718 rows=739 loops=1)

-> Hash Join (cost=742.75..1098.18 rows=1563 width=12) (actual time=7.562..10.078 rows=739 loops=1)

Hash Cond: (r.sid = s2.sid)

-> Hash Join (cost=55.56..381.31 rows=3750 width=4) (actual time=0.298..2.460 rows=3730 loops=1)

Hash Cond: (r.bid = b.bid)

-> Seq Scan on reserves r (cost=0.00..232.00 rows=15000 width=8) (actual time=0.010..0.759 rows=15000 loops=1)

-> Hash (cost=46.19..46.19 rows=750 width=4) (actual time=0.276..0.276 rows=750 loops=1)

Buckets: 1024 Batches: 1 Memory Usage: 35kB

-> Bitmap Heap Scan on boat b (cost=13.81..46.19 rows=750 width=4) (actual time=0.067..0.198 rows=750 loops=1)

Recheck Cond: ((color)::text = 'red'::text)

Heap Blocks: exact=23

-> Bitmap Index Scan on boat_color_idx (cost=0.00..13.63 rows=750 width=0) (actual time=0.059..0.059 rows=750 loops=1)

Index Cond: ((color)::text = 'red'::text)

-> Hash (cost=640.31..640.31 rows=3750 width=8) (actual time=7.243..7.243 rows=3395 loops=1)

Buckets: 4096 Batches: 1 Memory Usage: 165kB

-> HashAggregate (cost=602.81..640.31 rows=3750 width=8) (actual time=6.551..6.895 rows=3395 loops=1)

Group Key: s2.sid

-> Nested Loop (cost=55.56..593.44 rows=3750 width=8) (actual time=0.249..5.873 rows=3749 loops=1)

-> Hash Join (cost=55.56..381.31 rows=3750 width=4) (actual time=0.238..2.516 rows=3749 loops=1)

Hash Cond: (r2.bid = b2.bid)

-> Seq Scan on reserves r2 (cost=0.00..232.00 rows=15000 width=8) (actual time=0.008..0.788 rows=15000 loops=1)

-> Hash (cost=46.19..46.19 rows=750 width=4) (actual time=0.225..0.225 rows=750 loops=1)

Buckets: 1024 Batches: 1 Memory Usage: 35kB

-> Bitmap Heap Scan on boat b2 (cost=13.81..46.19 rows=750 width=4) (actual time=0.062..0.152 rows=750 loops=1)

Recheck Cond: ((color)::text = 'green'::text)

Heap Blocks: exact=23

-> Bitmap Index Scan on boat_color_idx (cost=0.00..13.63 rows=750 width=0) (actual time=0.058..0.058 rows=750 loops=1)

Index Cond: ((color)::text = 'green'::text)

-> Index Scan using sailors_sid_idx on sailors s2 (cost=0.00..0.05 rows=1 width=4) (actual time=0.001..0.001 rows=1 loops=3749)

Index Cond: (sid = r2.sid)

-> Index Scan using sailors_sid_idx on sailors s (cost=0.00..0.05 rows=1 width=25) (actual time=0.001..0.001 rows=1 loops=739)

Index Cond: (sid = r.sid)

Planning time: 2.073 ms

Execution time: 11.034 ms

(34 rows)

Scenario 4:

Btree index on reserves.bid

Btree index on sailors.sid

Bitmap index on boat.color

QUERY PLAN

Hash Join (cost=1055.34..1410.78 rows=1562 width=21) (actual time=7.772..10.473 rows=739 loops=1)

Hash Cond: (r.sid = s.sid)

-> Hash Join (cost=55.56..381.31 rows=3750 width=4) (actual time=0.263..2.501 rows=3730 loops=1)

Hash Cond: (r.bid = b.bid)

-> Seq Scan on reserves r (cost=0.00..232.00 rows=15000 width=8) (actual time=0.009..0.788 rows=15000 loops=1)

-> Hash (cost=46.19..46.19 rows=750 width=4) (actual time=0.244..0.244 rows=750 loops=1)

Buckets: 1024 Batches: 1 Memory Usage: 35kB

-> Bitmap Heap Scan on boat b (cost=13.81..46.19 rows=750 width=4) (actual time=0.073..0.175 rows=750 loops=1)

Recheck Cond: ((color)::text = 'red'::text)

Heap Blocks: exact=23

-> Bitmap Index Scan on boat_color_idx (cost=0.00..13.63 rows=750 width=0) (actual time=0.065..0.065 rows=750 loops=1)

Index Cond: ((color)::text = 'red'::text)

-> Hash (cost=952.91..952.91 rows=3750 width=33) (actual time=7.481..7.481 rows=3395 loops=1)

Buckets: 4096 Batches: 1 Memory Usage: 258kB

-> Hash Semi Join (cost=722.56..952.91 rows=3750 width=33) (actual time=5.114..6.925 rows=3395 loops=1)

Hash Cond: (s.sid = s2.sid)

-> Seq Scan on sailors s (cost=0.00..165.00 rows=9000 width=25) (actual time=0.008..0.581 rows=9000 loops=1)

-> Hash (cost=675.69..675.69 rows=3750 width=8) (actual time=5.088..5.088 rows=3749 loops=1)

Buckets: 4096 Batches: 1 Memory Usage: 179kB

-> Hash Join (cost=428.19..675.69 rows=3750 width=8) (actual time=2.955..4.668 rows=3749 loops=1)

Hash Cond: (s2.sid = r2.sid)

-> Seq Scan on sailors s2 (cost=0.00..165.00 rows=9000 width=4) (actual time=0.005..0.573 rows=9000 loops=1)

-> Hash (cost=381.31..381.31 rows=3750 width=4) (actual time=2.935..2.935 rows=3749 loops=1)

Buckets: 4096 Batches: 1 Memory Usage: 164kB

-> Hash Join (cost=55.56..381.31 rows=3750 width=4) (actual time=0.234..2.538 rows=3749 loops=1)

Hash Cond: (r2.bid = b2.bid)

-> Seq Scan on reserves r2 (cost=0.00..232.00 rows=15000 width=8) (actual time=0.008..0.858 rows=15000 loops=1)

-> Hash (cost=46.19..46.19 rows=750 width=4) (actual time=0.217..0.217 rows=750 loops=1)

Buckets: 1024 Batches: 1 Memory Usage: 35kB

-> Bitmap Heap Scan on boat b2 (cost=13.81..46.19 rows=750 width=4) (actual time=0.062..0.146 rows=750 loops=1)

Recheck Cond: ((color)::text = 'green'::text)

Heap Blocks: exact=23

-> Bitmap Index Scan on boat_color_idx (cost=0.00..13.63 rows=750 width=0) (actual time=0.058..0.058 rows=750 loops=1)

Index Cond: ((color)::text = 'green'::text)

Planning time: 2.021 ms

Execution time: 10.726 ms

(36 rows)

Scenario 5:

Bitmap index on reserves.bid

Bitmap index on sailors.sid

Bitmap index on boat.color

QUERY PLAN

Hash Join (cost=1055.34..1410.78 rows=1562 width=21) (actual time=7.401..9.947 rows=739 loops=1)

Hash Cond: (r.sid = s.sid)

-> Hash Join (cost=55.56..381.31 rows=3750 width=4) (actual time=0.264..2.462 rows=3730 loops=1)

Hash Cond: (r.bid = b.bid)

-> Seq Scan on reserves r (cost=0.00..232.00 rows=15000 width=8) (actual time=0.009..0.811 rows=15000 loops=1)

-> Hash (cost=46.19..46.19 rows=750 width=4) (actual time=0.246..0.246 rows=750 loops=1)

Buckets: 1024 Batches: 1 Memory Usage: 35kB

-> Bitmap Heap Scan on boat b (cost=13.81..46.19 rows=750 width=4) (actual time=0.067..0.175 rows=750 loops=1)

Recheck Cond: ((color)::text = 'red'::text)

Heap Blocks: exact=23

-> Bitmap Index Scan on boat_color_idx (cost=0.00..13.63 rows=750 width=0) (actual time=0.058..0.058 rows=750 loops=1)

Index Cond: ((color)::text = 'red'::text)

-> Hash (cost=952.91..952.91 rows=3750 width=33) (actual time=7.117..7.117 rows=3395 loops=1)

Buckets: 4096 Batches: 1 Memory Usage: 258kB

-> Hash Semi Join (cost=722.56..952.91 rows=3750 width=33) (actual time=4.965..6.630 rows=3395 loops=1)

Hash Cond: (s.sid = s2.sid)

-> Seq Scan on sailors s (cost=0.00..165.00 rows=9000 width=25) (actual time=0.009..0.567 rows=9000 loops=1)

-> Hash (cost=675.69..675.69 rows=3750 width=8) (actual time=4.939..4.939 rows=3749 loops=1)

Buckets: 4096 Batches: 1 Memory Usage: 179kB

-> Hash Join (cost=428.19..675.69 rows=3750 width=8) (actual time=2.888..4.539 rows=3749 loops=1)

Hash Cond: (s2.sid = r2.sid)

-> Seq Scan on sailors s2 (cost=0.00..165.00 rows=9000 width=4) (actual time=0.005..0.591 rows=9000 loops=1)

-> Hash (cost=381.31..381.31 rows=3750 width=4) (actual time=2.869..2.869 rows=3749 loops=1)

Buckets: 4096 Batches: 1 Memory Usage: 164kB

-> Hash Join (cost=55.56..381.31 rows=3750 width=4) (actual time=0.239..2.482 rows=3749 loops=1)

Hash Cond: (r2.bid = b2.bid)

-> Seq Scan on reserves r2 (cost=0.00..232.00 rows=15000 width=8) (actual time=0.007..0.863 rows=15000 loops=1)

-> Hash (cost=46.19..46.19 rows=750 width=4) (actual time=0.226..0.226 rows=750 loops=1)

Buckets: 1024 Batches: 1 Memory Usage: 35kB

-> Bitmap Heap Scan on boat b2 (cost=13.81..46.19 rows=750 width=4) (actual time=0.060..0.156 rows=750 loops=1)

Recheck Cond: ((color)::text = 'green'::text)

Heap Blocks: exact=23

-> Bitmap Index Scan on boat_color_idx (cost=0.00..13.63 rows=750 width=0) (actual time=0.056..0.056 rows=750 loops=1)

Index Cond: ((color)::text = 'green'::text)

Planning time: 2.175 ms

Execution time: 10.235 ms

(36 rows)

Conclusion of query 9:

The best performance was for Scenario 3: (Hash index on reserves.bid, Hash index on reserves.sid,

Hash index on boat.bid, Hash index on sailors.sid, Bitmap index on boat.color). As for the bitmap index on boat.color is better since the column has only four colors. It is the same reason in query 8.

The planner uses the Hash index on sailors.sid to filter the rows on both the main query and the sub-query and since there is no duplicates in sailors.sid, the hash index is the best option. Also, it is O (1).

The planner does not use the other indexes as the cost of a Nested Loop + Bitmap Heap scan is greater than the cost of Hash join + seq_scan. The sequential scan performs sequential I/O whereas Bitmap Heap scan perform random I/O and we are reading a large portion of the table (more than 40% of the data in the table). So, it is enough to use Hash index on sailors.sid and Bitmap index on boat.color.