1. Hash Join and Merge Join Disabled.

```
set enable_hashjoin  = off;
set enable_mergejoin  = off;


select distinct p.x,q.x
from P p, Q q
where exists(select 1
          from R r
          where r.x = p.x and not exists (select
                                  from S s
                                  where s.x = q.x and s.z=r.y offset 0));
```

```
                                                                QUERY PLAN
-------------------------------------------------------------------------------------------------------------------------------------------
-
 Unique  (cost=497692743763.75..497692755955.94 rows=40000 width=8) (actual
time=7.616..8.238 rows=0 loops=1)
   ->  Sort  (cost=497692743763.75..497692747827.81 rows=1625625 width=8) (actual
time=7.612..7.617 rows=0 loops=1)
        Sort Key: p.x, q.x
        Sort Method: quicksort  Memory: 25kB
        ->  Nested Loop Semi Join  (cost=0.00..497692553831.20 rows=1625625 width=8) (actual
time=0.061..0.065 rows=0 loops=1)
             Join Filter: ((p.x = r.x) AND (NOT (SubPlan 1)))
             ->  Nested Loop  (cost=0.00..81358.62 rows=6502500 width=8) (actual
time=0.024..0.027 rows=0 loops=1)
                  ->  Seq Scan on p  (cost=0.00..35.50 rows=2550 width=4) (actual
time=0.023..0.024 rows=0 loops=1)
                  ->  Materialize  (cost=0.00..48.25 rows=2550 width=4) (never executed)
                       ->  Seq Scan on q  (cost=0.00..35.50 rows=2550 width=4) (never executed)
             ->  Materialize  (cost=0.00..43.90 rows=2260 width=8) (never executed)
                  ->  Seq Scan on r  (cost=0.00..32.60 rows=2260 width=8) (never executed)
             SubPlan 1
               ->  Seq Scan on s  (cost=0.00..43.90 rows=1 width=0) (never executed)
                    Filter: ((x = q.x) AND (z = r.y))
 Planning Time: 64.714 ms
 Execution Time: 28.581 ms
(17 rows)
```

Hash Join and Merge Join Enabled

set enable_hashjoin = on;
set enable_mergejoin = on;

```
                                        QUERY PLAN
------------------------------------------------------------------------------------------------------------------
 Unique  (cost=71708020.23..71720212.42 rows=40000 width=8) (actual time=0.077..0.107 rows=0 loops=1)
   ->  Sort  (cost=71708020.23..71712084.29 rows=1625625 width=8) (actual time=0.076..0.105 rows=0 loops=1)
         Sort Key: p.x, q.x
         Sort Method: quicksort  Memory: 25kB
         ->  Hash Semi Join  (cost=60.85..71518087.68 rows=1625625 width=8) (actual time=0.042..0.071 rows=0 loops=1)
               Hash Cond: (p.x = r.x)
               Join Filter: (NOT (SubPlan 1))
               ->  Nested Loop  (cost=0.00..81358.62 rows=6502500 width=8) (actual time=0.016..0.017 rows=0 loops=1)
                     ->  Seq Scan on p  (cost=0.00..35.50 rows=2550 width=4) (actual time=0.014..0.015 rows=0 loops=1)
                     ->  Materialize  (cost=0.00..48.25 rows=2550 width=4) (never executed)
                           ->  Seq Scan on q  (cost=0.00..35.50 rows=2550 width=4) (never executed)
               ->  Hash  (cost=32.60..32.60 rows=2260 width=8) (never executed)
                     ->  Seq Scan on r  (cost=0.00..32.60 rows=2260 width=8) (never executed)
               SubPlan 1
                 ->  Seq Scan on s  (cost=0.00..43.90 rows=1 width=4) (never executed)
                       Filter: ((x = q.x) AND (z = r.y))
 Planning Time: 5.237 ms
 Execution Time: 0.279 ms
(18 rows)
```

Hash Index

set enable_indexonlyscan = on;

select distinct p.x,q.x
from P p, Q q
where exists(select r.y

```
        from R r
        where r.x = p.x and not exists (select s.z
                                from S s
                                where s.x = q.x and s.z=r.y));
```

```
                                QUERY PLAN
-----------------------------------------------------------------------------------------------------------------------
 Unique  (cost=71708020.23..71720212.42 rows=40000 width=8) (actual time=0.032..0.038
rows=0 loops=1)
   ->  Sort  (cost=71708020.23..71712084.29 rows=1625625 width=8) (actual time=0.031..0.036
rows=0 loops=1)
        Sort Key: p.x, q.x
        Sort Method: quicksort  Memory: 25kB
        ->  Hash Semi Join  (cost=60.85..71518087.68 rows=1625625 width=8) (actual
time=0.015..0.019 rows=0 loops=1)
              Hash Cond: (p.x = r.x)
              Join Filter: (NOT (alternatives: SubPlan 1 or hashed SubPlan 2))
              ->  Nested Loop  (cost=0.00..81358.62 rows=6502500 width=8) (actual
time=0.013..0.015 rows=0 loops=1)
                    ->  Seq Scan on p  (cost=0.00..35.50 rows=2550 width=4) (actual
time=0.011..0.012 rows=0 loops=1)
                    ->  Materialize  (cost=0.00..48.25 rows=2550 width=4) (never executed)
                          ->  Seq Scan on q  (cost=0.00..35.50 rows=2550 width=4) (never executed)
              ->  Hash  (cost=32.60..32.60 rows=2260 width=8) (never executed)
                    ->  Seq Scan on r  (cost=0.00..32.60 rows=2260 width=8) (never executed)
              SubPlan 1
                ->  Seq Scan on s  (cost=0.00..43.90 rows=1 width=0) (never executed)
                      Filter: ((x = q.x) AND (z = r.y))
              SubPlan 2
                ->  Seq Scan on s s_1  (cost=0.00..32.60 rows=2260 width=8) (never executed)
 Planning Time: 0.738 ms
 Execution Time: 0.261 ms
(20 rows)
```

NOT IN

```
select distinct p.x,q.x
from P p, Q q
where exists(select r.y
        from R r
        where r.x = p.x and (q.x,r.y) not in (select s.x,s.z
                                from S s ));
```

```
                                     QUERY PLAN
-------------------------------------------------------------------------------------------------------------
 HashAggregate  (cost=169444.68..169844.68 rows=40000 width=8) (actual time=0.442..0.446
rows=0 loops=1)
   Group Key: p.x, q.x
   Batches: 1  Memory Usage: 1561kB
   ->  Hash Semi Join  (cost=99.10..161316.55 rows=1625625 width=8) (actual
time=0.012..0.016 rows=0 loops=1)
         Hash Cond: (p.x = r.x)
         Join Filter: (NOT (hashed SubPlan 1))
         ->  Nested Loop  (cost=0.00..81358.62 rows=6502500 width=8) (actual time=0.010..0.011
rows=0 loops=1)
               ->  Seq Scan on p  (cost=0.00..35.50 rows=2550 width=4) (actual time=0.009..0.009
rows=0 loops=1)
               ->  Materialize  (cost=0.00..48.25 rows=2550 width=4) (never executed)
                     ->  Seq Scan on q  (cost=0.00..35.50 rows=2550 width=4) (never executed)
         ->  Hash  (cost=32.60..32.60 rows=2260 width=8) (never executed)
               ->  Seq Scan on r  (cost=0.00..32.60 rows=2260 width=8) (never executed)
         SubPlan 1
           ->  Seq Scan on s  (cost=0.00..32.60 rows=2260 width=8) (never executed)
 Planning Time: 0.306 ms
 Execution Time: 1.083 ms
(16 rows)
```

RA optimization

```
select q1.* from (select p.x,q.x
from P p, Q q ,R r
except
select p.x,q.x
from P p, Q q ,R r ,S s
where r.x = p.x and q.x=s.x and r.y=s.z) q1;
```

```
                                          QUERY PLAN
-------------------------------------------------------------------------------------------------------------------------
--------------------
 Subquery Scan on q1  (cost=0.00..477832458.01 rows=40000 width=8) (actual
time=0.331..0.336 rows=0 loops=1)
```

```
  ->  HashSetOp Except  (cost=0.00..477832058.01 rows=40000 width=12) (actual
time=0.330..0.335 rows=0 loops=1)
       ->  Append  (cost=0.00..404333050.37 rows=14699801528 width=12) (actual
time=0.057..0.062 rows=0 loops=1)
            ->  Subquery Scan on "*SELECT* 1"  (cost=0.00..330724278.12 rows=14695650000
width=12) (actual time=0.010..0.011 rows=0 loops=1)
                 ->  Nested Loop  (cost=0.00..183767778.12 rows=14695650000 width=8) (actual
time=0.010..0.010 rows=0 loops=1)
                      ->  Nested Loop  (cost=0.00..72111.25 rows=5763000 width=4) (actual
time=0.009..0.010 rows=0 loops=1)
                           ->  Seq Scan on p  (cost=0.00..35.50 rows=2550 width=4) (actual
time=0.009..0.009 rows=0 loops=1)
                           ->  Materialize  (cost=0.00..43.90 rows=2260 width=0) (never executed)
                                ->  Seq Scan on r  (cost=0.00..32.60 rows=2260 width=0) (never
executed)
                      ->  Materialize  (cost=0.00..48.25 rows=2550 width=4) (never executed)
                           ->  Seq Scan on q  (cost=0.00..35.50 rows=2550 width=4) (never
executed)
            ->  Subquery Scan on "*SELECT* 2"  (cost=5832.43..109764.60 rows=4151528
width=12) (actual time=0.047..0.049 rows=0 loops=1)
                 ->  Merge Join  (cost=5832.43..68249.32 rows=4151528 width=8) (actual
time=0.046..0.049 rows=0 loops=1)
                      Merge Cond: (r_1.y = s.z)
                      ->  Sort  (cost=2916.22..2988.25 rows=28815 width=8) (actual
time=0.046..0.047 rows=0 loops=1)
                           Sort Key: r_1.y
                           Sort Method: quicksort  Memory: 25kB
                           ->  Merge Join  (cost=338.29..781.81 rows=28815 width=8) (actual
time=0.025..0.026 rows=0 loops=1)
                                Merge Cond: (r_1.x = p_1.x)
                                ->  Sort  (cost=158.51..164.16 rows=2260 width=8) (actual
time=0.025..0.025 rows=0 loops=1)
                                     Sort Key: r_1.x
                                     Sort Method: quicksort  Memory: 25kB
                                     ->  Seq Scan on r r_1  (cost=0.00..32.60 rows=2260 width=8)
(actual time=0.011..0.011 rows=0 loops=1)
                                ->  Sort  (cost=179.78..186.16 rows=2550 width=4) (never executed)
                                     Sort Key: p_1.x
                                     ->  Seq Scan on p p_1  (cost=0.00..35.50 rows=2550 width=4)
(never executed)
                      ->  Sort  (cost=2916.22..2988.25 rows=28815 width=8) (never executed)
                           Sort Key: s.z
                           ->  Merge Join  (cost=338.29..781.81 rows=28815 width=8) (never
executed)
```

```
                        Merge Cond: (s.x = q_1.x)
                        ->  Sort  (cost=158.51..164.16 rows=2260 width=8) (never executed)
                              Sort Key: s.x
                              ->  Seq Scan on s  (cost=0.00..32.60 rows=2260 width=8) (never
executed)
                        ->  Sort  (cost=179.78..186.16 rows=2550 width=4) (never executed)
                              Sort Key: q_1.x
                              ->  Seq Scan on q q_1  (cost=0.00..35.50 rows=2550 width=4)
(never executed)
 Planning Time: 8.091 ms
 Execution Time: 0.982 ms
(38 rows)
```

2.

```
explain analyze SELECT DISTINCT p.a
FROM P p  natural join R r1 natural join R r2 natural join R r3 natural join S s;
```

b.

R	Q3	Q4
10^3	7.302 ms	4.581 ms
10^4	36.848 ms	32.380 ms
10^5	414.718 ms	304.368 ms

c.Explanation:

We can notice that for 10^3 Q4 takes shorter execution time, We can also notice that as the size increases to 10^5 the execution time observed will still be lesser for Q4 when compared with Q3

3. a.

```
explain analyze SELECT Q.a
FROM (SELECT r.a
                FROM P p JOIN R r ON r.a=p.a
                INTERSECT
                (SELECT r.a
                        FROM R r
                        EXCEPT
                        SELECT r.a
                        FROM R r JOIN S s ON r.b=s.b))Q;
```

b.

R	Q5	Q6	Q7
10^3	5.466 ms	7.539 ms	1.439 ms
10^4	28.382 ms	44.268 ms	29.803 ms
10^5	316.905 ms	424.756 ms	13685.946 ms

c.Explanation:

We can notice that for 10^1 we won't be able to differentiate as Q7 shows very less time but Q5 and Q6 will have similar or less of a difference in execution time. For larger values i.e 10^5 it was observed that Q7 takes significantly more time as compared to the other two. Hence Q7 must be not optimal.

4.

R	Q8	Q9	Q10
10^3	4.178 ms	595.230 ms	5.472 ms
10^4	19.654 ms	58220.585 ms	30.553 ms
10^5	186.120 ms	Took more than 30 minutes to execute	358.401 ms

Explanation:

We can notice that Q9 takes a lot of time initially for a small size which is 10^1 when compared with the other two. I waited for over 45 mins for Q9 for size 10^5 to execute but was still not able to get the execution time hence we can conclude that it takes a significant amount of time when compared with Q8 and Q10 for larger sizes as well.

5. In problem 3 the query Q7 performs the worst which is the object relational query, when compared to the other two. Q6 and Q5 show similar executional behaviour. In problem 4 Q9

performs the worst as we saw, where Q8 shows best execution time when compared to the other two which is Q10 and Q9. Q9 performed worsley as it did not complete the execution for 10^5 when left for more than 45 mins