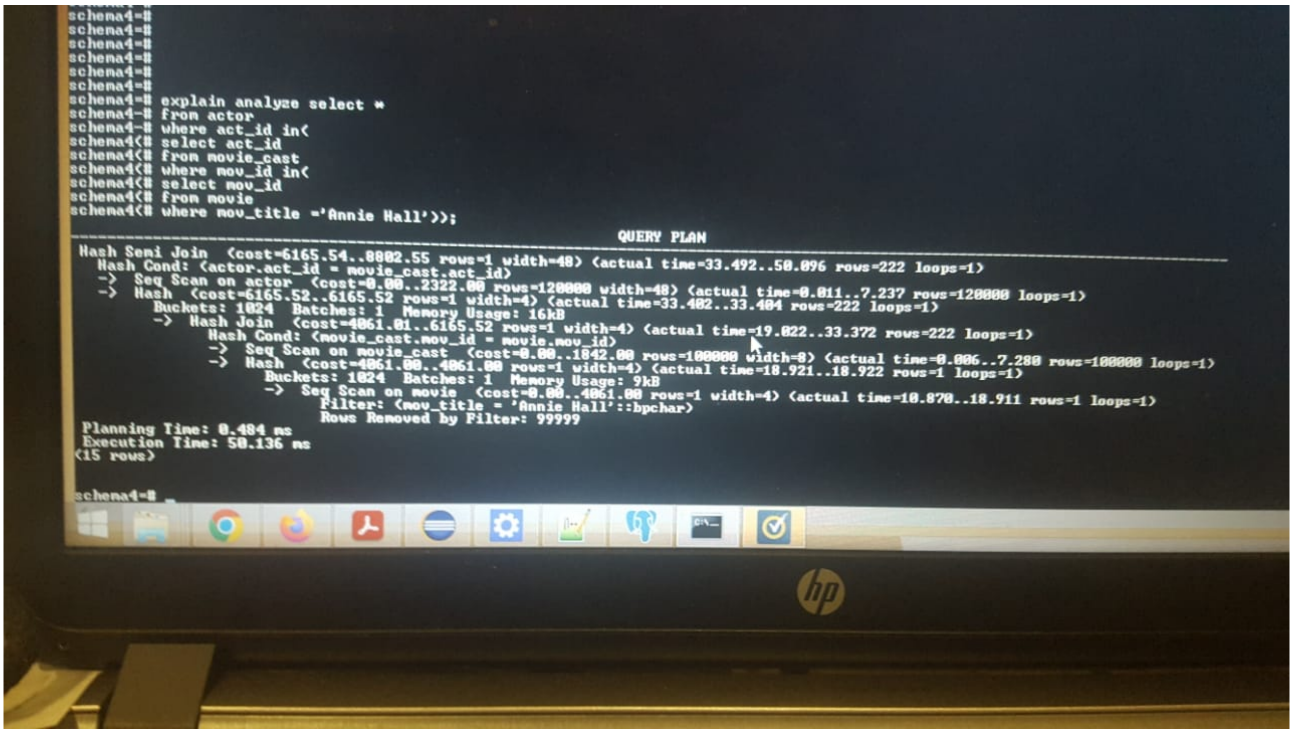
SCHEMA 4:

1)A)OLD QUERY 10 WITH NO INDEXES:

Flags:1- set enable_seqscan = on;
2- set enable_bitmapscan =off;
3-set enable_indexscan = off;
4- set enable_indexonlyscan = off;



```
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=# explain analyze select *
schema4-# from actor
schema4-# where act_id in(
schema4(# select act_id
schema4(# from movie_cast
schema4(# where mov_id in(
schema4(# select mov_id
schema4(# from movie
schema4(# where mov_title ='Annie Hall'));
                                                          QUERY PLAN
--------------------------------------------------------------------------------------------------------------------------------------
 Hash Semi Join  (cost=6165.54..8802.55 rows=1 width=48) (actual time=33.492..50.096 rows=222 loops=1)
   Hash Cond: (actor.act_id = movie_cast.act_id)
   ->  Seq Scan on actor  (cost=0.00..2322.00 rows=120000 width=48) (actual time=0.011..7.237 rows=120000 loops=1)
   ->  Hash  (cost=6165.52..6165.52 rows=1 width=4) (actual time=33.402..33.404 rows=222 loops=1)
         Buckets: 1024  Batches: 1  Memory Usage: 16kB
         ->  Hash Join  (cost=4061.01..6165.52 rows=1 width=4) (actual time=19.022..33.372 rows=222 loops=1)
               Hash Cond: (movie_cast.mov_id = movie.mov_id)
               ->  Seq Scan on movie_cast  (cost=0.00..1842.00 rows=100000 width=8) (actual time=0.006..7.280 rows=100000 loops=1)
               ->  Hash  (cost=4061.00..4061.00 rows=1 width=4) (actual time=18.921..18.922 rows=1 loops=1)
                     Buckets: 1024  Batches: 1  Memory Usage: 9kB
                     ->  Seq Scan on movie  (cost=0.00..4061.00 rows=1 width=4) (actual time=10.870..18.911 rows=1 loops=1)
                           Filter: (mov_title = 'Annie Hall'::bpchar)
                           Rows Removed by Filter: 99999
 Planning Time: 0.484 ms
 Execution Time: 50.136 ms
(15 rows)


schema4=# _
```

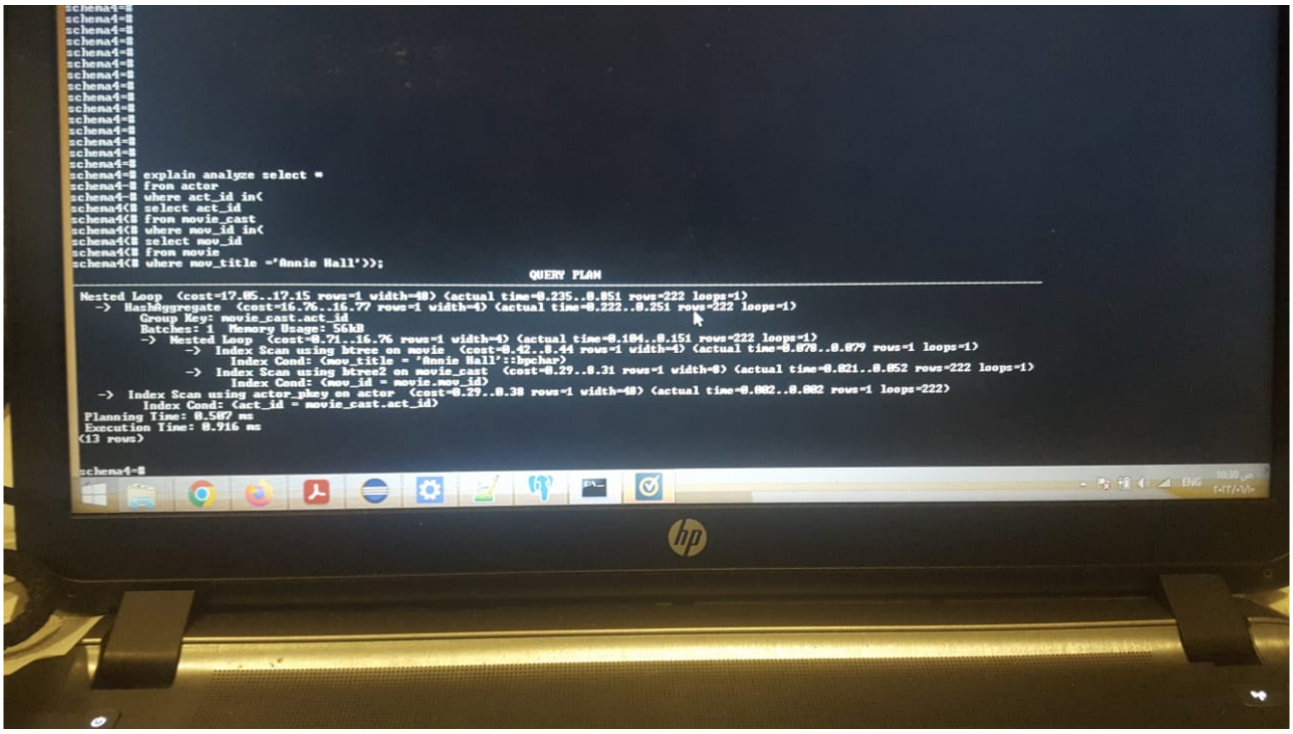
******************CONCLUSION OLD QUERY 10 NO INDEXES:*******************

1-OLD QUERY 10 NO INDEXES COST = 8802
2-OLD QUERY 10 NO INDEXES TIME = 50 MS

1)B)OLD QUERY 10 WITH B+ INDEXES:

B+Indexes:
 1- All default PKs indexes.
 2 - movie(mov_title)
 3 - movie_cast(mov_id).

Flags: 1- set enable_seqscan = on;
            2- set enable_bitmapscan =on;
            3-set enable_indexscan = on;
            4- set enable_indexonlyscan = on;



```
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=# explain analyze select *
schema4-# from actor
schema4-# where act_id in(
schema4(# select act_id
schema4(# from movie_cast
schema4(# where mov_id in(
schema4(# select mov_id
schema4(# from movie
schema4(# where mov_title ='Annie Hall'));
                                                          QUERY PLAN
------------------------------------------------------------------------------------------------------------------------------------
 Nested Loop  (cost=17.05..17.15 rows=1 width=48) (actual time=0.235..0.851 rows=222 loops=1)
   ->  HashAggregate  (cost=16.76..16.77 rows=1 width=4) (actual time=0.222..0.251 rows=222 loops=1)
         Group Key: movie_cast.act_id
         Batches: 1  Memory Usage: 56kB
         ->  Nested Loop  (cost=0.71..16.76 rows=1 width=4) (actual time=0.104..0.151 rows=222 loops=1)
               ->  Index Scan using btree on movie  (cost=0.42..8.44 rows=1 width=4) (actual time=0.078..0.079 rows=1 loops=1)
                     Index Cond: (mov_title = 'Annie Hall'::bpchar)
               ->  Index Scan using btree2 on movie_cast  (cost=0.29..8.31 rows=1 width=8) (actual time=0.021..0.052 rows=222 loops=1)
                     Index Cond: (mov_id = movie.mov_id)
   ->  Index Scan using actor_pkey on actor  (cost=0.29..0.38 rows=1 width=48) (actual time=0.002..0.002 rows=1 loops=222)
         Index Cond: (act_id = movie_cast.act_id)
 Planning Time: 0.507 ms
 Execution Time: 0.916 ms
(13 rows)

schema4=#
```

*************CONCLUSION QUERY 10 B+ INDEXES:****************

1-OLD QUERY 10 NO INDEXES COST = 8802
2-OLD QUERY 10 NO INDEXES TIME = 50 MS

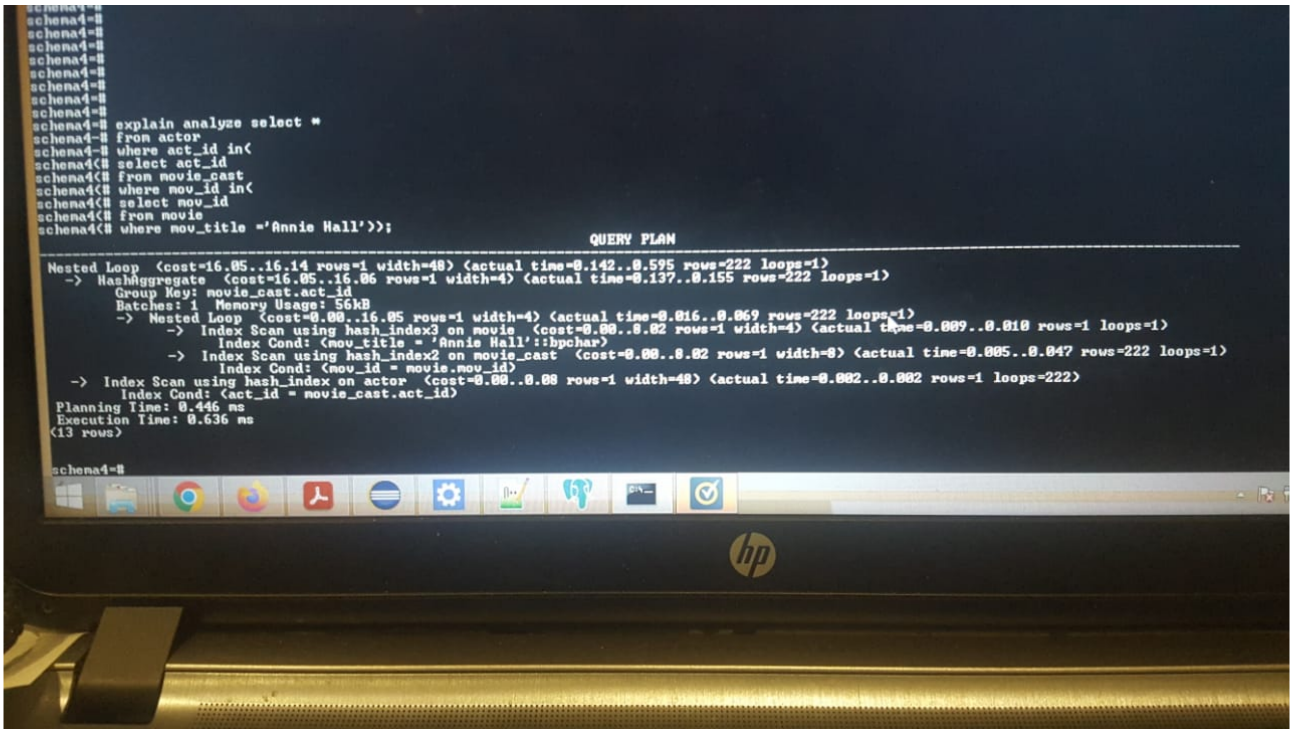
———————————————————————————

1-OLD QUERY 10 B+ INDEXES COST = 17
2-OLD QUERY 10 B+ INDEXES TIME = 0.9 MS
=> 99.8% more cost efficient & 98.2% more time efficient.

1)C)OLD QUERY 10 WITH Hash INDEXES:

Hash indexes:1-actor(act_id)
2-movie_cast(mov_id)
3-movie(mov_title)

Flags: 1- set enable_seqscan = on;
2- set enable_bitmapscan =on;
3-set enable_indexscan = on;
4- set enable_indexonlyscan = on;



```
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=# explain analyze select *
schema4-# from actor
schema4-# where act_id in(
schema4(# select act_id
schema4(# from movie_cast
schema4(# where mov_id in(
schema4(# select mov_id
schema4(# from movie
schema4(# where mov_title ='Annie Hall'));
                                                    QUERY PLAN
--------------------------------------------------------------------------------------------------------------------------------
 Nested Loop  (cost=16.05..16.14 rows=1 width=48) (actual time=0.142..0.595 rows=222 loops=1)
   ->  HashAggregate  (cost=16.05..16.06 rows=1 width=4) (actual time=0.137..0.155 rows=222 loops=1)
         Group Key: movie_cast.act_id
         Batches: 1  Memory Usage: 56kB
         ->  Nested Loop  (cost=0.00..16.05 rows=1 width=4) (actual time=0.016..0.069 rows=222 loops=1)
               ->  Index Scan using hash_index3 on movie  (cost=0.00..8.02 rows=1 width=4) (actual time=0.009..0.010 rows=1 loops=1)
                     Index Cond: (mov_title = 'Annie Hall'::bpchar)
               ->  Index Scan using hash_index2 on movie_cast  (cost=0.00..8.02 rows=1 width=8) (actual time=0.005..0.047 rows=222 loops=1)
                     Index Cond: (mov_id = movie.mov_id)
   ->  Index Scan using hash_index on actor  (cost=0.00..0.08 rows=1 width=48) (actual time=0.002..0.002 rows=1 loops=222)
         Index Cond: (act_id = movie_cast.act_id)
 Planning Time: 0.446 ms
 Execution Time: 0.636 ms
(13 rows)

schema4=#
```

*************CONCLUSION QUERY 10 HASH INDEXES:****************

1-OLD QUERY 10 NO INDEXES COST =8802
2-OLD QUERY 10 NO INDEXES TIME = 50 MS

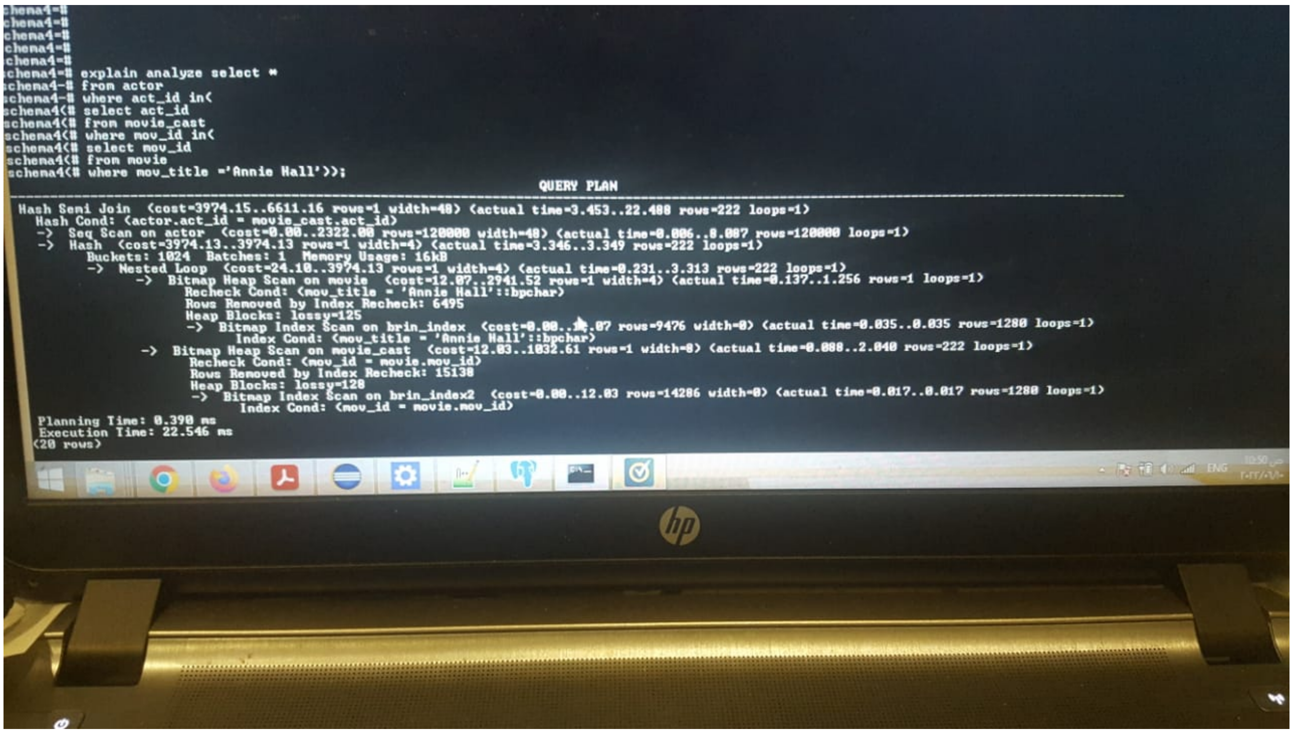
————————————————————————

1-OLD QUERY 10 HASH INDEXES COST = 16
2-OLD QUERY 10 HASH INDEXES TIME = 0.6 MS
=>99.8% more cost efficient &98.8% more time efficient.

1)D)OLD QUERY 10 WITH BRIN INDEX:

BRIN INDEXES: 1-movie(mov_title)
2- movie_cast(mov_id)

Flags: 1- set enable_seqscan = on;
2- set enable_bitmapscan =on;
3-set enable_indexscan = on;
4- set enable_indexonlyscan = on;
5- update pg_index set indisvalid = false where indexrelid = 'actor_pkey'::regclass ;



```
hema4=#
chema4=#
chema4=#
chema4=#
chema4=#
chema4=# explain analyze select *
chema4-# from actor
chema4-# where act_id in(
schema4(# select act_id
schema4(# from movie_cast
schema4(# where mov_id in(
schema4(# select mov_id
schema4(# from movie
schema4(# where mov_title ='Annie Hall'));
                                                        QUERY PLAN
------------------------------------------------------------------------------------------------------------------------------------------
 Hash Semi Join  (cost=3974.15..6611.16 rows=1 width=48) (actual time=3.453..22.488 rows=222 loops=1)
   Hash Cond: (actor.act_id = movie_cast.act_id)
   ->  Seq Scan on actor  (cost=0.00..2322.00 rows=120000 width=48) (actual time=0.006..8.087 rows=120000 loops=1)
   ->  Hash  (cost=3974.13..3974.13 rows=1 width=4) (actual time=3.346..3.349 rows=222 loops=1)
         Buckets: 1024  Batches: 1  Memory Usage: 16kB
         ->  Nested Loop  (cost=24.10..3974.13 rows=1 width=4) (actual time=0.231..3.313 rows=222 loops=1)
               ->  Bitmap Heap Scan on movie  (cost=12.07..2941.52 rows=1 width=4) (actual time=0.137..1.256 rows=1 loops=1)
                     Recheck Cond: (mov_title = 'Annie Hall'::bpchar)
                     Rows Removed by Index Recheck: 6495
                     Heap Blocks: lossy=125
                     ->  Bitmap Index Scan on brin_index  (cost=0.00..1[illegible].07 rows=9476 width=0) (actual time=0.035..0.035 rows=1280 loops=1)
                           Index Cond: (mov_title = 'Annie Hall'::bpchar)
               ->  Bitmap Heap Scan on movie_cast  (cost=12.03..1032.61 rows=1 width=8) (actual time=0.088..2.040 rows=222 loops=1)
                     Recheck Cond: (mov_id = movie.mov_id)
                     Rows Removed by Index Recheck: 15138
                     Heap Blocks: lossy=128
                     ->  Bitmap Index Scan on brin_index2  (cost=0.00..12.03 rows=14286 width=0) (actual time=0.017..0.017 rows=1280 loops=1)
                           Index Cond: (mov_id = movie.mov_id)
 Planning Time: 0.390 ms
 Execution Time: 22.546 ms
(20 rows)
```

*************CONCLUSION QUERY 10 BRIN INDEX:****************

1-OLD QUERY 10 NO INDEXES COST = 8802
2-OLD QUERY 10 NO INDEXES TIME = 50 MS

—————————————————————————

1-OLD QUERY 10 BRIN INDEX COST = 6611
2-OLD QUERY 10 BRIN INDEX TIME = 22 MS
=>25% more cost efficient &56% more time efficient.

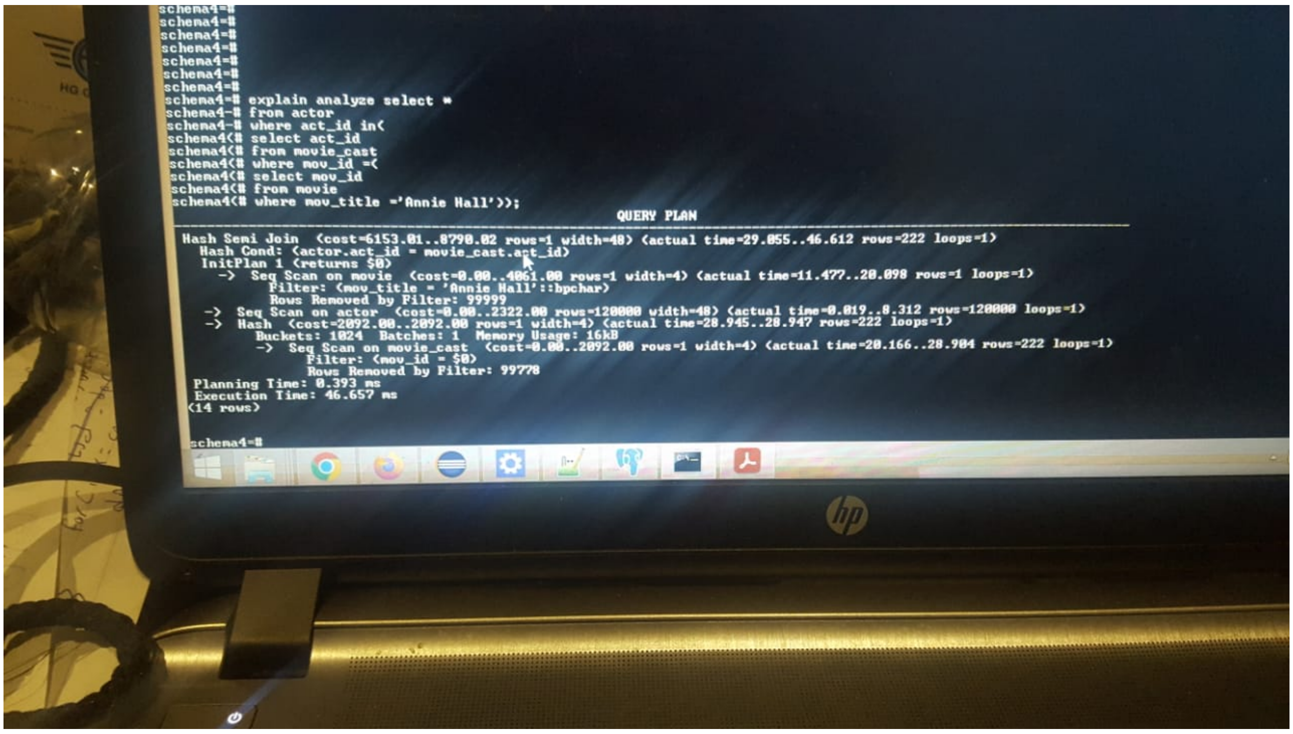
1)E)OLD QUERY 10 WITH MY CHOICE FOR INDEXES:

Full hash indexes same as 1)c): as all the conditions are equality conditions
Hash indexes:1-actor(act_id)
                    2-movie_cast(mov_id)
                    3-movie(mov_title)
Flags: 1- set enable_seqscan = on;
           2- set enable_bitmapscan =on;
           3-set enable_indexscan = on;
           4- set enable_indexonlyscan = on;

*************CONCLUSION QUERY 10 HASH INDEXES:****************
     1-OLD QUERY 10 NO INDEXES COST =8802
     2-OLD QUERY 10 NO INDEXES TIME = 50 MS
— —
        1-OLD QUERY 10 HASH INDEXES COST = 16
        2-OLD QUERY 10 HASH INDEXES TIME = 0.6 MS
        =>99.8% more cost efficient &98.8% more time efficient.

2)A)OPTIMIZED QUERY 10 WITH NO INDEXES:

Flags:1- set enable_seqscan = on;
2- set enable_bitmapscan =off;
3-set enable_indexscan = off;
4- set enable_indexonlyscan = off;



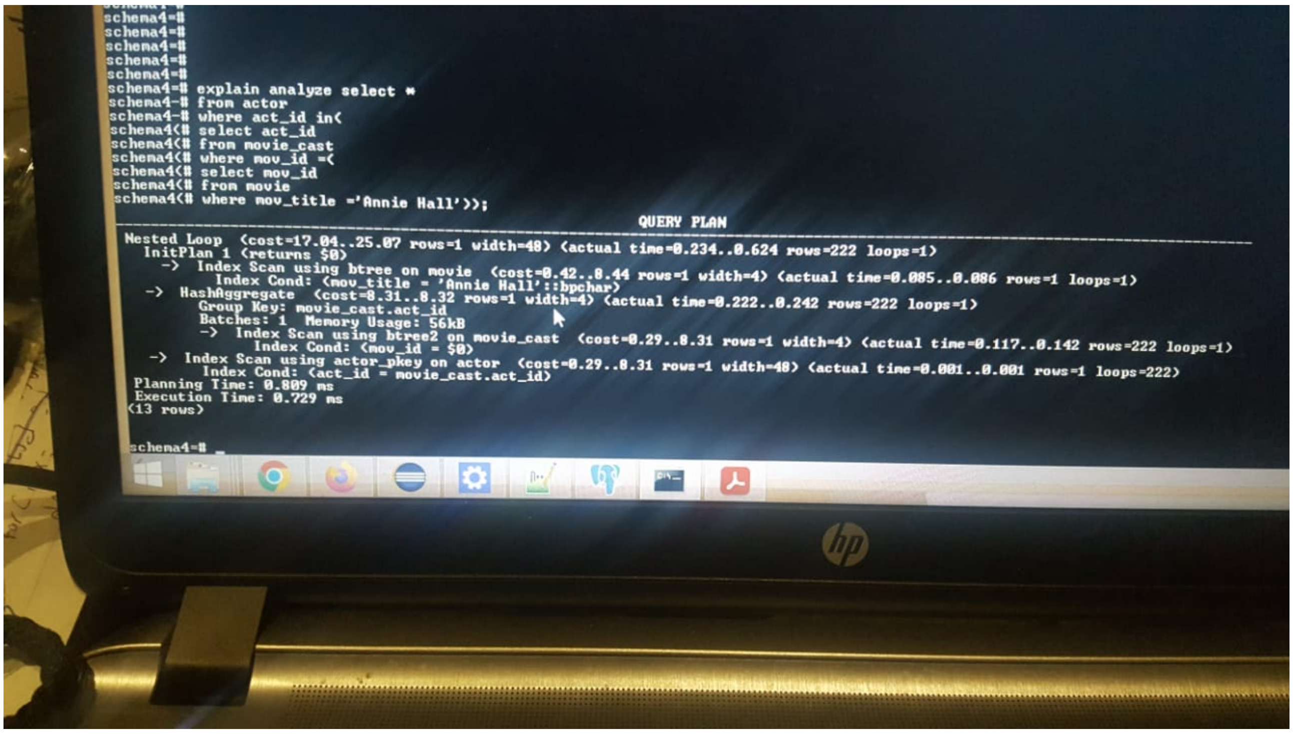
```
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=# explain analyze select *
schema4-# from actor
schema4-# where act_id in(
schema4(# select act_id
schema4(# from movie_cast
schema4(# where mov_id =(
schema4(# select mov_id
schema4(# from movie
schema4(# where mov_title ='Annie Hall'));
                                                     QUERY PLAN
------------------------------------------------------------------------------------------------------------------------------------
 Hash Semi Join  (cost=6153.01..8790.02 rows=1 width=48) (actual time=29.055..46.612 rows=222 loops=1)
   Hash Cond: (actor.act_id = movie_cast.act_id)
   InitPlan 1 (returns $0)
     ->  Seq Scan on movie  (cost=0.00..4061.00 rows=1 width=4) (actual time=11.477..20.098 rows=1 loops=1)
           Filter: (mov_title = 'Annie Hall'::bpchar)
           Rows Removed by Filter: 99999
   ->  Seq Scan on actor  (cost=0.00..2322.00 rows=120000 width=48) (actual time=0.019..8.312 rows=120000 loops=1)
   ->  Hash  (cost=2092.00..2092.00 rows=1 width=4) (actual time=28.945..28.947 rows=222 loops=1)
         Buckets: 1024  Batches: 1  Memory Usage: 16kB
         ->  Seq Scan on movie_cast  (cost=0.00..2092.00 rows=1 width=4) (actual time=20.166..28.904 rows=222 loops=1)
               Filter: (mov_id = $0)
               Rows Removed by Filter: 99778
 Planning Time: 0.393 ms
 Execution Time: 46.657 ms
(14 rows)

schema4=#
```

Optimised query 10 no index cost = 6153
Optimised query 10 no index time = 46.6

2)B)OPTIMIZED QUERY 10 WITH B+ INDEXES:

B+Indexes:
 1- All default PKs indexes.
 2 - movie(mov_title)
 3 - movie_cast(mov_id).

Flags: 1- set enable_seqscan = on;
 2- set enable_bitmapscan =on;
 3-set enable_indexscan = on;
 4- set enable_indexonlyscan = on;



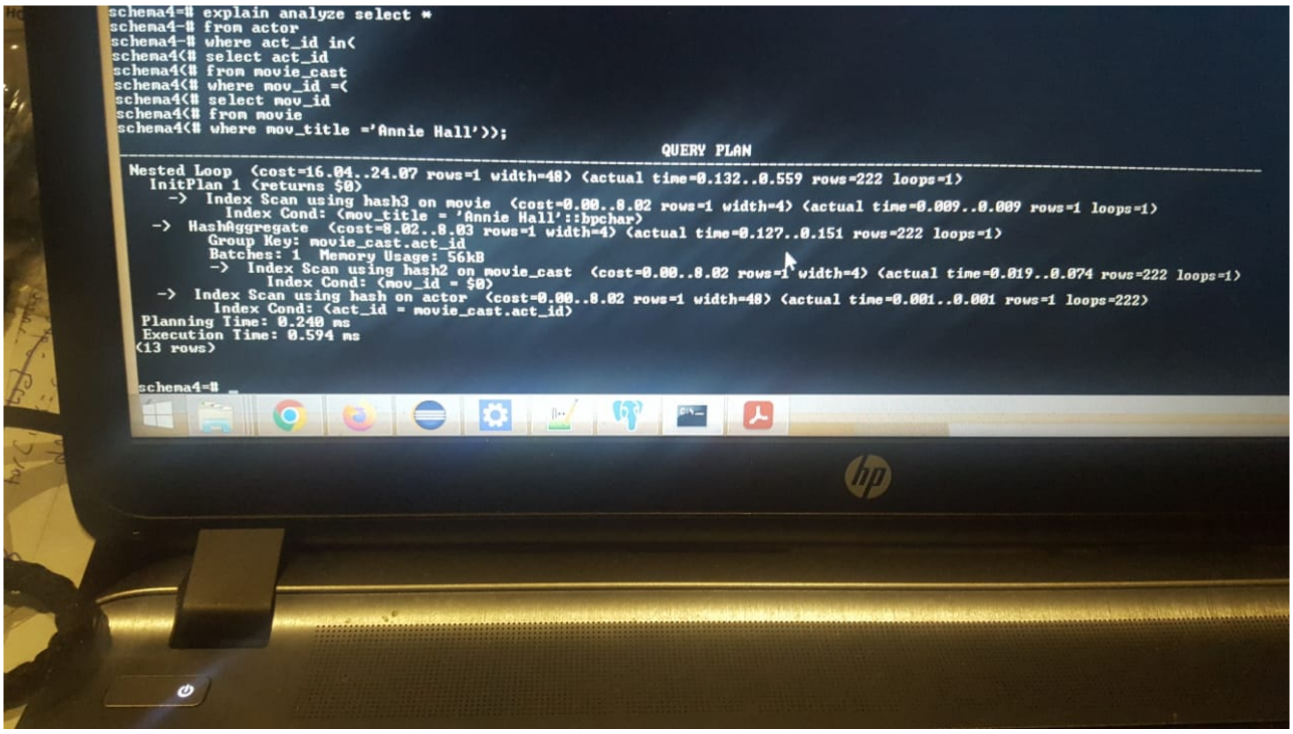
```
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=# explain analyze select *
schema4-# from actor
schema4-# where act_id in(
schema4(# select act_id
schema4(# from movie_cast
schema4(# where mov_id =(
schema4(# select mov_id
schema4(# from movie
schema4(# where mov_title ='Annie Hall'));
                                                   QUERY PLAN
--------------------------------------------------------------------------------------------------------------------------------
 Nested Loop  (cost=17.04..25.07 rows=1 width=48) (actual time=0.234..0.624 rows=222 loops=1)
   InitPlan 1 (returns $0)
     ->  Index Scan using btree on movie  (cost=0.42..8.44 rows=1 width=4) (actual time=0.085..0.086 rows=1 loops=1)
           Index Cond: (mov_title = 'Annie Hall'::bpchar)
   ->  HashAggregate  (cost=8.31..8.32 rows=1 width=4) (actual time=0.222..0.242 rows=222 loops=1)
         Group Key: movie_cast.act_id
         Batches: 1  Memory Usage: 56kB
         ->  Index Scan using btree2 on movie_cast  (cost=0.29..8.31 rows=1 width=4) (actual time=0.117..0.142 rows=222 loops=1)
               Index Cond: (mov_id = $0)
   ->  Index Scan using actor_pkey on actor  (cost=0.29..8.31 rows=1 width=48) (actual time=0.001..0.001 rows=1 loops=222)
         Index Cond: (act_id = movie_cast.act_id)
 Planning Time: 0.809 ms
 Execution Time: 0.729 ms
(13 rows)


schema4=# _
```

Cost = 25
Time =0.7ms

1)C)OPTIMIZED QUERY 10 WITH Hash INDEXES:

Hash indexes:1-actor(act_id)
2-movie_cast(mov_id)
3-movie(mov_title)

Flags: 1- set enable_seqscan = on;
2- set enable_bitmapscan =on;
3-set enable_indexscan = on;
4- set enable_indexonlyscan = on;

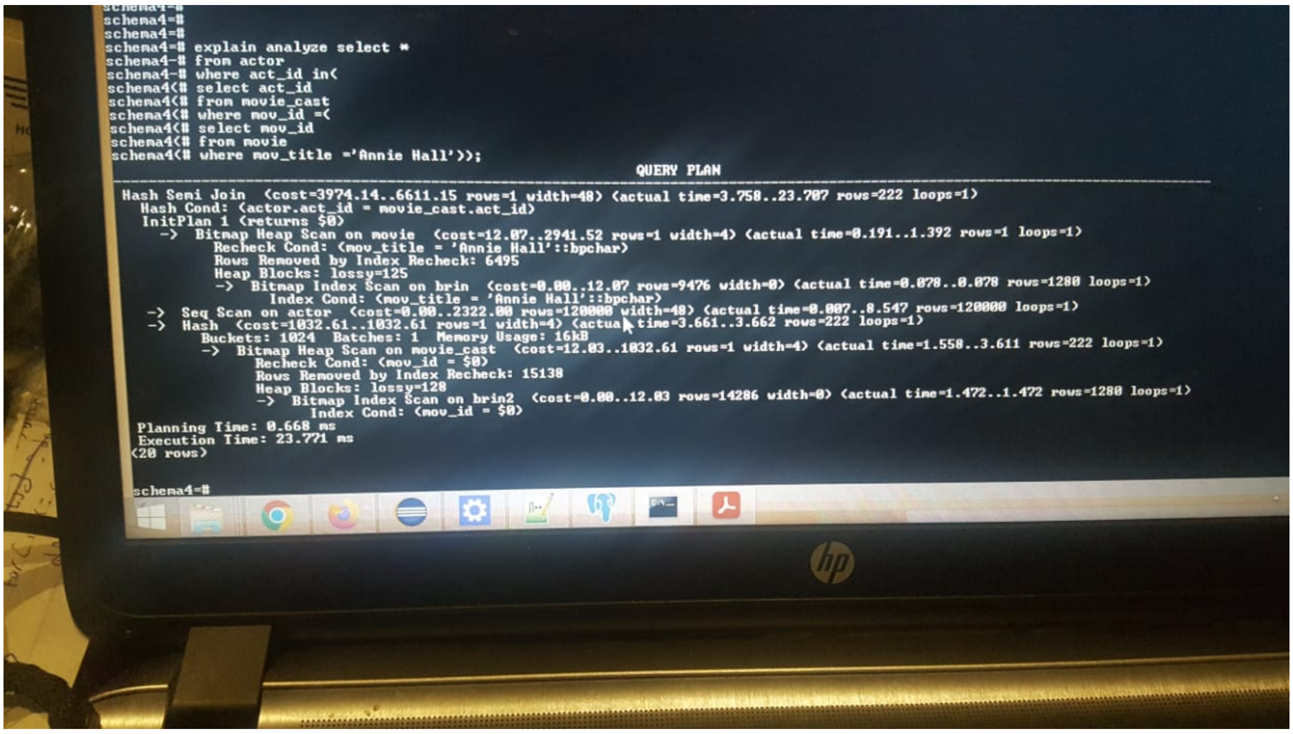


```
schema4=# explain analyze select *
schema4-# from actor
schema4-# where act_id in(
schema4(# select act_id
schema4(# from movie_cast
schema4(# where mov_id =(
schema4(# select mov_id
schema4(# from movie
schema4(# where mov_title ='Annie Hall'));
                                                    QUERY PLAN
-------------------------------------------------------------------------------------------------------------------------------
 Nested Loop  (cost=16.04..24.07 rows=1 width=48) (actual time=0.132..0.559 rows=222 loops=1)
   InitPlan 1 (returns $0)
     ->  Index Scan using hash3 on movie  (cost=0.00..8.02 rows=1 width=4) (actual time=0.009..0.009 rows=1 loops=1)
           Index Cond: (mov_title = 'Annie Hall'::bpchar)
   ->  HashAggregate  (cost=8.02..8.03 rows=1 width=4) (actual time=0.127..0.151 rows=222 loops=1)
         Group Key: movie_cast.act_id
         Batches: 1  Memory Usage: 56kB
         ->  Index Scan using hash2 on movie_cast  (cost=0.00..8.02 rows=1 width=4) (actual time=0.019..0.074 rows=222 loops=1)
               Index Cond: (mov_id = $0)
   ->  Index Scan using hash on actor  (cost=0.00..8.02 rows=1 width=48) (actual time=0.001..0.001 rows=1 loops=222)
         Index Cond: (act_id = movie_cast.act_id)
 Planning Time: 0.240 ms
 Execution Time: 0.594 ms
(13 rows)

schema4=# _
```

New cost hash = 24
New time hash = .2MS

1)D)optimized QUERY 10 WITH BRIN INDEX:

BRIN INDEXES: 1-movie(mov_title)
                        2- movie_cast(mov_id)
Flags: 1- set enable_seqscan = on;
            2- set enable_bitmapscan =on;
            3-set enable_indexscan = on;
            4- set enable_indexonlyscan = on;
            5- update pg_index set indisvalid = false where indexrelid = 'actor_pkey'::regclass ;



```
schema4=#
schema4=#
schema4=#
schema4=# explain analyze select *
schema4-# from actor
schema4-# where act_id in(
schema4(# select act_id
schema4(# from movie_cast
schema4(# where mov_id =(
schema4(# select mov_id
schema4(# from movie
schema4(# where mov_title ='Annie Hall'));
                                                    QUERY PLAN
--------------------------------------------------------------------------------------------------------------------------
 Hash Semi Join  (cost=3974.14..6611.15 rows=1 width=48) (actual time=3.758..23.707 rows=222 loops=1)
   Hash Cond: (actor.act_id = movie_cast.act_id)
   InitPlan 1 (returns $0)
     ->  Bitmap Heap Scan on movie  (cost=12.07..2941.52 rows=1 width=4) (actual time=0.191..1.392 rows=1 loops=1)
           Recheck Cond: (mov_title = 'Annie Hall'::bpchar)
           Rows Removed by Index Recheck: 6495
           Heap Blocks: lossy=125
           ->  Bitmap Index Scan on brin  (cost=0.00..12.07 rows=9476 width=0) (actual time=0.078..0.078 rows=1280 loops=1)
                 Index Cond: (mov_title = 'Annie Hall'::bpchar)
   ->  Seq Scan on actor  (cost=0.00..2322.00 rows=120000 width=48) (actual time=0.007..8.547 rows=120000 loops=1)
   ->  Hash  (cost=1032.61..1032.61 rows=1 width=4) (actual time=3.661..3.662 rows=222 loops=1)
         Buckets: 1024  Batches: 1  Memory Usage: 16kB
         ->  Bitmap Heap Scan on movie_cast  (cost=12.03..1032.61 rows=1 width=4) (actual time=1.558..3.611 rows=222 loops=1)
               Recheck Cond: (mov_id = $0)
               Rows Removed by Index Recheck: 15138
               Heap Blocks: lossy=128
               ->  Bitmap Index Scan on brin2  (cost=0.00..12.03 rows=14286 width=0) (actual time=1.472..1.472 rows=1280 loops=1)
                     Index Cond: (mov_id = $0)
 Planning Time: 0.668 ms
 Execution Time: 23.771 ms
(20 rows)

schema4=#
```

Cost = 6611
Time =23.7 ms

1)E)OPTIMIZED QUERY 10 WITH MY CHOICE FOR INDEXES:

Full hash indexes same as 1)c): as all the conditions are equality conditions
Hash indexes:1-actor(act_id)
                    2-movie_cast(mov_id)
                    3-movie(mov_title)
Flags: 1- set enable_seqscan = on;
           2- set enable_bitmapscan =on;
           3-set enable_indexscan = on;
           4- set enable_indexonlyscan = on;

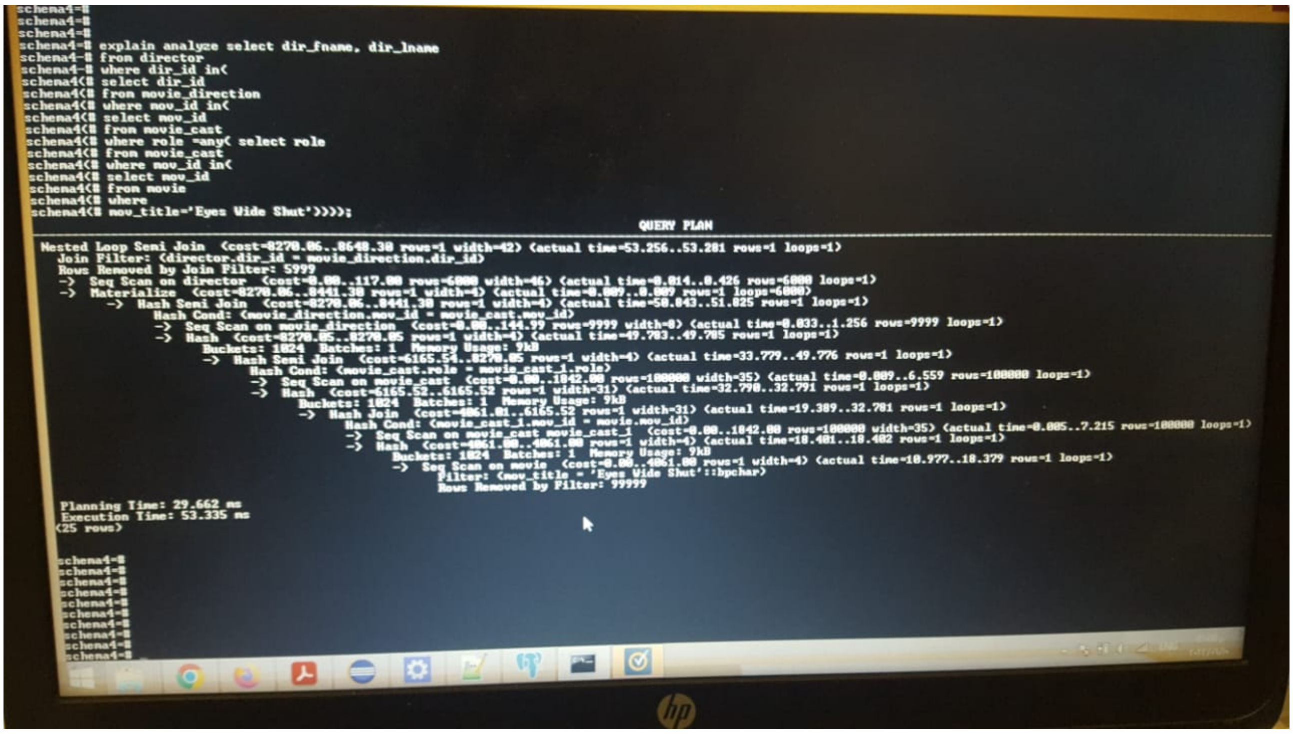
*************CONCLUSION QUERY 10 HASH INDEXES:****************

1-OPTIMIZED QUERY 10 HASH INDEXES COST = 16
2-OPTIMIZED QUERY 10 HASH INDEXES TIME = 0.6 MS

3)A)OLD QUERY 11 WITH NO INDEXES:

```
Flags:1- set enable_seqscan = on;
      2- set enable_bitmapscan =off;
```

3-set enable_indexscan = off;
4- set enable_indexonlyscan = off;



```
schema4=#
schema4=#
schema4=#
schema4=# explain analyze select dir_fname, dir_lname
schema4-# from director
schema4-# where dir_id in(
schema4(# select dir_id
schema4(# from movie_direction
schema4(# where mov_id in(
schema4(# select mov_id
schema4(# from movie_cast
schema4(# where role =any( select role
schema4(# from movie_cast
schema4(# where mov_id in(
schema4(# select mov_id
schema4(# from movie
schema4(# where
schema4(# mov_title='Eyes Wide Shut')))));
                                                              QUERY PLAN
-------------------------------------------------------------------------------------------------------------------------------------------
 Nested Loop Semi Join  (cost=8270.06..8648.30 rows=1 width=42) (actual time=53.256..53.281 rows=1 loops=1)
   Join Filter: (director.dir_id = movie_direction.dir_id)
   Rows Removed by Join Filter: 5999
   ->  Seq Scan on director  (cost=0.00..117.00 rows=6000 width=46) (actual time=0.014..0.426 rows=6000 loops=1)
   ->  Materialize  (cost=8270.06..8441.30 rows=1 width=4) (actual time=0.009..0.009 rows=1 loops=6000)
         ->  Hash Semi Join  (cost=8270.06..8441.30 rows=1 width=4) (actual time=50.843..51.825 rows=1 loops=1)
               Hash Cond: (movie_direction.mov_id = movie_cast.mov_id)
               ->  Seq Scan on movie_direction  (cost=0.00..144.99 rows=9999 width=8) (actual time=0.033..1.256 rows=9999 loops=1)
               ->  Hash  (cost=8270.05..8270.05 rows=1 width=4) (actual time=49.783..49.785 rows=1 loops=1)
                     Buckets: 1024  Batches: 1  Memory Usage: 9kB
                     ->  Hash Semi Join  (cost=6165.54..8270.05 rows=1 width=4) (actual time=33.779..49.776 rows=1 loops=1)
                           Hash Cond: (movie_cast.role = movie_cast_1.role)
                           ->  Seq Scan on movie_cast  (cost=0.00..1842.00 rows=100000 width=35) (actual time=0.009..6.559 rows=100000 loops=1)
                           ->  Hash  (cost=6165.52..6165.52 rows=1 width=31) (actual time=32.790..32.791 rows=1 loops=1)
                                 Buckets: 1024  Batches: 1  Memory Usage: 9kB
                                 ->  Hash Join  (cost=4061.01..6165.52 rows=1 width=31) (actual time=19.389..32.781 rows=1 loops=1)
                                       Hash Cond: (movie_cast_1.mov_id = movie.mov_id)
                                       ->  Seq Scan on movie_cast movie_cast_1  (cost=0.00..1842.00 rows=100000 width=35) (actual time=0.005..7.215 rows=100000 loops=1)
                                       ->  Hash  (cost=4061.00..4061.00 rows=1 width=4) (actual time=18.401..18.402 rows=1 loops=1)
                                             Buckets: 1024  Batches: 1  Memory Usage: 9kB
                                             ->  Seq Scan on movie  (cost=0.00..4061.00 rows=1 width=4) (actual time=10.977..18.379 rows=1 loops=1)
                                                   Filter: (mov_title = 'Eyes Wide Shut'::bpchar)
                                                   Rows Removed by Filter: 99999
 Planning Time: 29.662 ms
 Execution Time: 53.335 ms
(25 rows)

schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
```

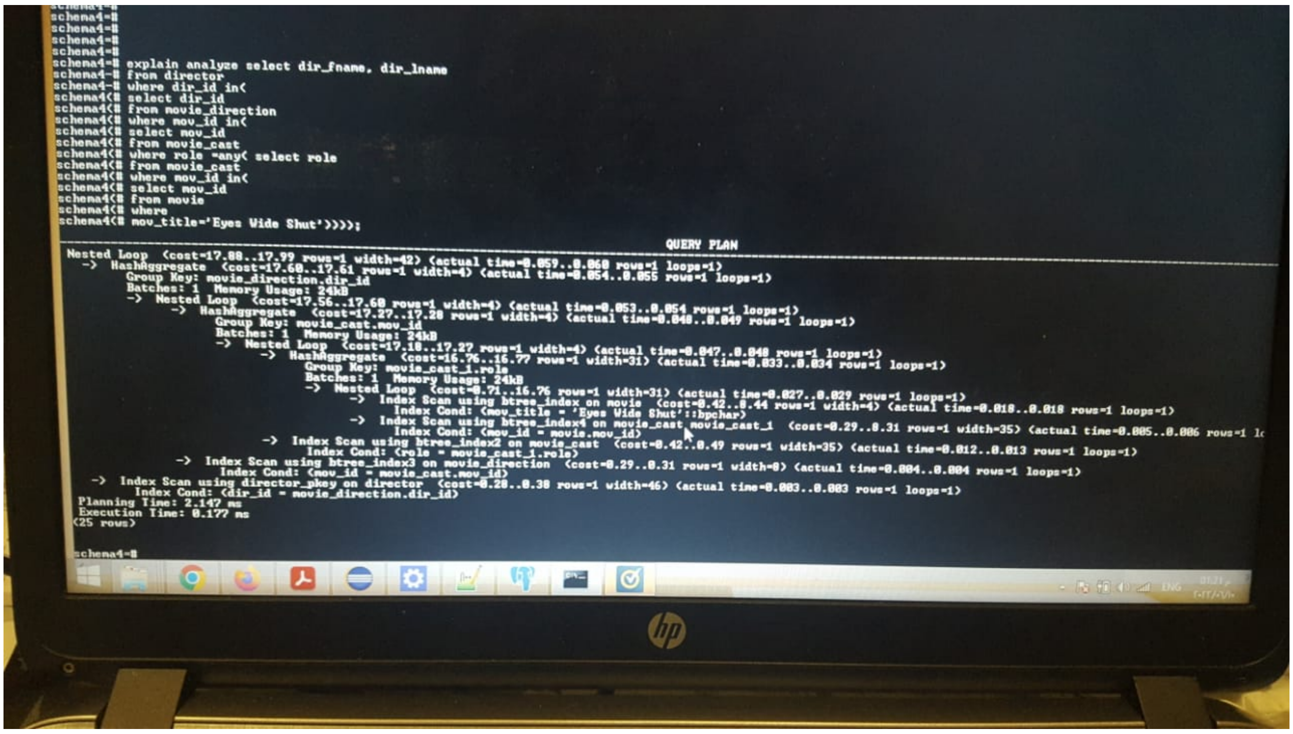
**********CONCLUSION OLD QUERY 11 NO INDEXES:**********

1-OLD QUERY 11 NO INDEXES COST = 8638
2-OLD QUERY 11 NO INDEXES TIME = 53 MS

3)B)OLD QUERY 11 WITH B+ INDEXES:

B+Indexes:
1- All default PKs indexes.
2 - movie(mov_title).
3 - movie_cast(mov_id).

4 - movie_cast(role).
5 -movie_direction(mov_id).
Flags: 1- set enable_seqscan = on;
2- set enable_bitmapscan =on;
3-set enable_indexscan = on;
4- set enable_indexonlyscan = on;



```
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=# explain analyze select dir_fname, dir_lname
schema4-# from director
schema4-# where dir_id in(
schema4(# select dir_id
schema4(# from movie_direction
schema4(# where mov_id in(
schema4(# select mov_id
schema4(# from movie_cast
schema4(# where role =any( select role
schema4(# from movie_cast
schema4(# where mov_id in(
schema4(# select mov_id
schema4(# from movie
schema4(# where
schema4(# mov_title='Eyes Wide Shut')))));
                                                                    QUERY PLAN
------------------------------------------------------------------------------------------------------------------------------------------------
 Nested Loop  (cost=17.88..17.99 rows=1 width=42) (actual time=0.059..0.060 rows=1 loops=1)
   ->  HashAggregate  (cost=17.60..17.61 rows=1 width=4) (actual time=0.054..0.055 rows=1 loops=1)
         Group Key: movie_direction.dir_id
         Batches: 1  Memory Usage: 24kB
         ->  Nested Loop  (cost=17.56..17.60 rows=1 width=4) (actual time=0.053..0.054 rows=1 loops=1)
               ->  HashAggregate  (cost=17.27..17.28 rows=1 width=4) (actual time=0.048..0.049 rows=1 loops=1)
                     Group Key: movie_cast.mov_id
                     Batches: 1  Memory Usage: 24kB
                     ->  Nested Loop  (cost=17.18..17.27 rows=1 width=4) (actual time=0.047..0.048 rows=1 loops=1)
                           ->  HashAggregate  (cost=16.76..16.77 rows=1 width=31) (actual time=0.033..0.034 rows=1 loops=1)
                                 Group Key: movie_cast_1.role
                                 Batches: 1  Memory Usage: 24kB
                                 ->  Nested Loop  (cost=0.71..16.76 rows=1 width=31) (actual time=0.027..0.029 rows=1 loops=1)
                                       ->  Index Scan using btree_index on movie  (cost=0.42..0.44 rows=1 width=4) (actual time=0.018..0.018 rows=1 loops=1)
                                             Index Cond: (mov_title = 'Eyes Wide Shut'::bpchar)
                                       ->  Index Scan using btree_index4 on movie_cast movie_cast_1  (cost=0.29..0.31 rows=1 width=35) (actual time=0.005..0.006 rows=1 lo
                                             Index Cond: (mov_id = movie.mov_id)
                           ->  Index Scan using btree_index2 on movie_cast  (cost=0.42..0.49 rows=1 width=35) (actual time=0.012..0.013 rows=1 loops=1)
                                 Index Cond: (role = movie_cast_1.role)
               ->  Index Scan using btree_index3 on movie_direction  (cost=0.29..0.31 rows=1 width=8) (actual time=0.004..0.004 rows=1 loops=1)
                     Index Cond: (mov_id = movie_cast.mov_id)
   ->  Index Scan using director_pkey on director  (cost=0.28..0.38 rows=1 width=46) (actual time=0.003..0.003 rows=1 loops=1)
         Index Cond: (dir_id = movie_direction.dir_id)
 Planning Time: 2.147 ms
 Execution Time: 0.177 ms
(25 rows)

schema4=#
```

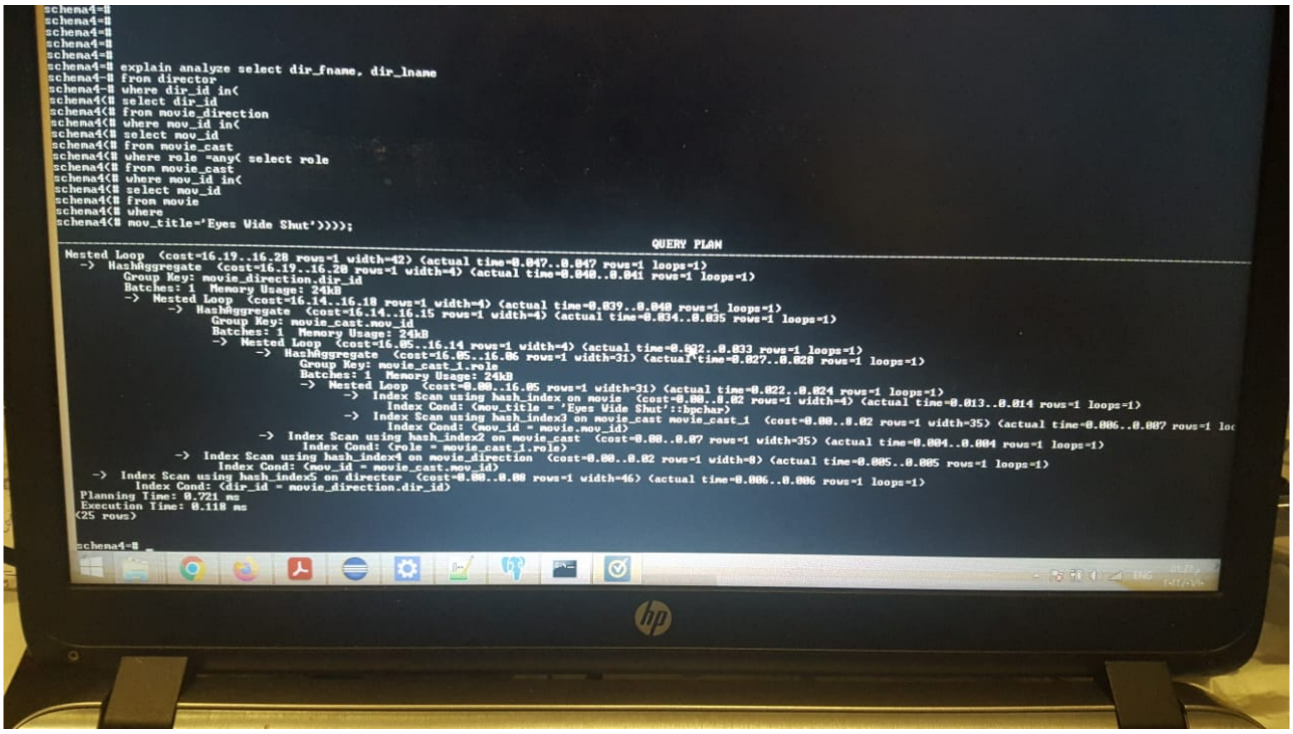
*************CONCLUSION OLD QUERY 11 B+ INDEXES:****************
1-OLD QUERY 11 NO INDEXES COST = 8638
2-OLD QUERY 11 NO INDEXES TIME = 53 MS
——————————————————————————
1-OLD QUERY 11 B+ INDEXES COST = 18
2-OLD QUERY 11 B+ INDEXES TIME = 0.1 MS
=> 99.8% more cost efficient & 99.8% more time efficient.

3)C)OLD QUERY 11 WITH Hash INDEXES:

Hash indexes: 1- director(dir_id)

2- movie_cast(mov_id)
3- movie_cast(role)
4 - movie_direction(mov_id)
5-movie(mov_title)

Flags: 1- set enable_seqscan = on;
2- set enable_bitmapscan =on;
3-set enable_indexscan = on;
4- set enable_indexonlyscan = on;



```
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=# explain analyze select dir_fname, dir_lname
schema4-# from director
schema4-# where dir_id in(
schema4(# select dir_id
schema4(# from movie_direction
schema4(# where mov_id in(
schema4(# select mov_id
schema4(# from movie_cast
schema4(# where role =any( select role
schema4(# from movie_cast
schema4(# where mov_id in(
schema4(# select mov_id
schema4(# from movie
schema4(# where
schema4(# mov_title='Eyes Wide Shut')))));
                                                                QUERY PLAN
--------------------------------------------------------------------------------------------------------------------------------------------
 Nested Loop  (cost=16.19..16.28 rows=1 width=42) (actual time=0.047..0.047 rows=1 loops=1)
   ->  HashAggregate  (cost=16.19..16.20 rows=1 width=4) (actual time=0.040..0.041 rows=1 loops=1)
         Group Key: movie_direction.dir_id
         Batches: 1  Memory Usage: 24kB
         ->  Nested Loop  (cost=16.14..16.18 rows=1 width=4) (actual time=0.039..0.040 rows=1 loops=1)
               ->  HashAggregate  (cost=16.14..16.15 rows=1 width=4) (actual time=0.034..0.035 rows=1 loops=1)
                     Group Key: movie_cast.mov_id
                     Batches: 1  Memory Usage: 24kB
                     ->  Nested Loop  (cost=16.05..16.14 rows=1 width=4) (actual time=0.032..0.033 rows=1 loops=1)
                           ->  HashAggregate  (cost=16.05..16.06 rows=1 width=31) (actual time=0.027..0.028 rows=1 loops=1)
                                 Group Key: movie_cast_1.role
                                 Batches: 1  Memory Usage: 24kB
                                 ->  Nested Loop  (cost=0.00..16.05 rows=1 width=31) (actual time=0.022..0.024 rows=1 loops=1)
                                       ->  Index Scan using hash_index on movie  (cost=0.00..8.02 rows=1 width=4) (actual time=0.013..0.014 rows=1 loops=1)
                                             Index Cond: (mov_title = 'Eyes Wide Shut'::bpchar)
                                       ->  Index Scan using hash_index3 on movie_cast movie_cast_1  (cost=0.00..8.02 rows=1 width=35) (actual time=0.006..0.007 rows=1 loc
                                             Index Cond: (mov_id = movie.mov_id)
                           ->  Index Scan using hash_index2 on movie_cast  (cost=0.00..0.07 rows=1 width=35) (actual time=0.004..0.004 rows=1 loops=1)
                                 Index Cond: (role = movie_cast_1.role)
               ->  Index Scan using hash_index4 on movie_direction  (cost=0.00..0.02 rows=1 width=8) (actual time=0.005..0.005 rows=1 loops=1)
                     Index Cond: (mov_id = movie_cast.mov_id)
   ->  Index Scan using hash_index5 on director  (cost=0.00..0.08 rows=1 width=46) (actual time=0.006..0.006 rows=1 loops=1)
         Index Cond: (dir_id = movie_direction.dir_id)
 Planning Time: 0.721 ms
 Execution Time: 0.118 ms
(25 rows)

schema4=# _
```

*************CONCLUSION QUERY 11 HASH INDEXES:****************

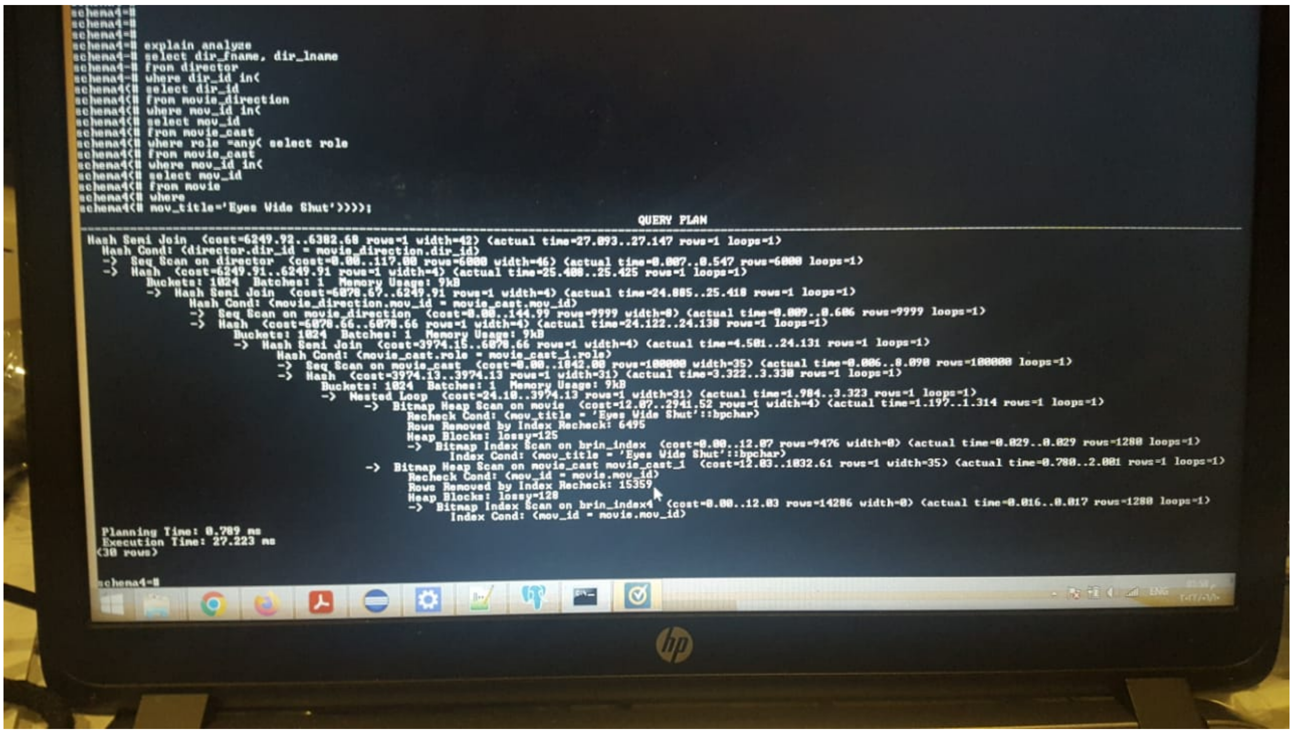
1-OLD QUERY 11 NO INDEXES COST = 8638
2-OLD QUERY 11 NO INDEXES TIME = 53 MS
3-OLD QUERY 11 HASH INDEXES COST = 16.2
4-OLD QUERY 11 HASH INDEXES TIME = 0.1 MS
=>99.8% more cost efficient &99.8% more time efficient.

3)D)OLD QUERY 11 WITH BRIN INDEX:

BRIN INDEXES: 1-movie(mov_title)
2- movie_cast(mov_id)

Flags: 1- set enable_seqscan = on;
2- set enable_bitmapscan =on;
3-set enable_indexscan = on;

4- set enable_indexonlyscan = on;
5- update pg_index set indisvalid = false where indexrelid = 'director_pkey'::regclass ;
6- update pg_index set indisvalid = false where indexrelid = ‘movie_direction_pkey'::regclass ;



```
schema4=#
schema4=#
schema4=#
schema4=# explain analyze
schema4-# select dir_fname, dir_lname
schema4-# from director
schema4-# where dir_id in(
schema4(# select dir_id
schema4(# from movie_direction
schema4(# where mov_id in(
schema4(# select mov_id
schema4(# from movie_cast
schema4(# where role =any( select role
schema4(# from movie_cast
schema4(# where mov_id in(
schema4(# select mov_id
schema4(# from movie
schema4(# where
schema4(# mov_title='Eyes Wide Shut'))));
                                                              QUERY PLAN
---------------------------------------------------------------------------------------------------------------------------------------
 Hash Semi Join  (cost=6249.92..6382.68 rows=1 width=42) (actual time=27.093..27.147 rows=1 loops=1)
   Hash Cond: (director.dir_id = movie_direction.dir_id)
   ->  Seq Scan on director  (cost=0.00..117.00 rows=6000 width=46) (actual time=0.007..0.547 rows=6000 loops=1)
   ->  Hash  (cost=6249.91..6249.91 rows=1 width=4) (actual time=25.408..25.425 rows=1 loops=1)
         Buckets: 1024  Batches: 1  Memory Usage: 9kB
         ->  Hash Semi Join  (cost=6078.67..6249.91 rows=1 width=4) (actual time=24.885..25.418 rows=1 loops=1)
               Hash Cond: (movie_direction.mov_id = movie_cast.mov_id)
               ->  Seq Scan on movie_direction  (cost=0.00..144.99 rows=9999 width=8) (actual time=0.009..0.606 rows=9999 loops=1)
               ->  Hash  (cost=6078.66..6078.66 rows=1 width=4) (actual time=24.122..24.138 rows=1 loops=1)
                     Buckets: 1024  Batches: 1  Memory Usage: 9kB
                     ->  Hash Semi Join  (cost=3974.15..6078.66 rows=1 width=4) (actual time=4.501..24.131 rows=1 loops=1)
                           Hash Cond: (movie_cast.role = movie_cast_1.role)
                           ->  Seq Scan on movie_cast  (cost=0.00..1842.00 rows=100000 width=35) (actual time=0.006..8.090 rows=100000 loops=1)
                           ->  Hash  (cost=3974.13..3974.13 rows=1 width=31) (actual time=3.322..3.330 rows=1 loops=1)
                                 Buckets: 1024  Batches: 1  Memory Usage: 9kB
                                 ->  Nested Loop  (cost=24.10..3974.13 rows=1 width=31) (actual time=1.984..3.323 rows=1 loops=1)
                                       ->  Bitmap Heap Scan on movie  (cost=12.07..2941.52 rows=1 width=4) (actual time=1.197..1.314 rows=1 loops=1)
                                             Recheck Cond: (mov_title = 'Eyes Wide Shut'::bpchar)
                                             Rows Removed by Index Recheck: 6495
                                             Heap Blocks: lossy=125
                                             ->  Bitmap Index Scan on brin_index  (cost=0.00..12.07 rows=9476 width=0) (actual time=0.029..0.029 rows=1280 loops=1)
                                                   Index Cond: (mov_title = 'Eyes Wide Shut'::bpchar)
                                       ->  Bitmap Heap Scan on movie_cast movie_cast_1  (cost=12.03..1032.61 rows=1 width=35) (actual time=0.780..2.001 rows=1 loops=1)
                                             Recheck Cond: (mov_id = movie.mov_id)
                                             Rows Removed by Index Recheck: 15359
                                             Heap Blocks: lossy=128
                                             ->  Bitmap Index Scan on brin_index4  (cost=0.00..12.03 rows=14286 width=0) (actual time=0.016..0.017 rows=1280 loops=1)
                                                   Index Cond: (mov_id = movie.mov_id)
 Planning Time: 0.789 ms
 Execution Time: 27.223 ms
(30 rows)

schema4=#
```

*************CONCLUSION OLD QUERY 11 BRIN INDEX:****************

1-OLD QUERY 11 NO INDEXES COST = 8638
2-OLD QUERY 11 NO INDEXES TIME = 53 MS.

———————————————————————————

1-OLD QUERY 11 BRIN INDEX COST = 6382
2-OLD QUERY 11 BRIN INDEX TIME = 27 MS
=>26% more cost efficient &49% more time efficient.

3)E)OLD QUERY 11 WITH MY CHOICE FOR INDEXES:

Same as 3)c) using only hash indexes as all the conditions are equality conditions.

Hash indexes: 1- director(dir_id)
2- movie_cast(mov_id)
3- movie_cast(role)
4 - movie_direction(mov_id)
5-movie(mov_title)

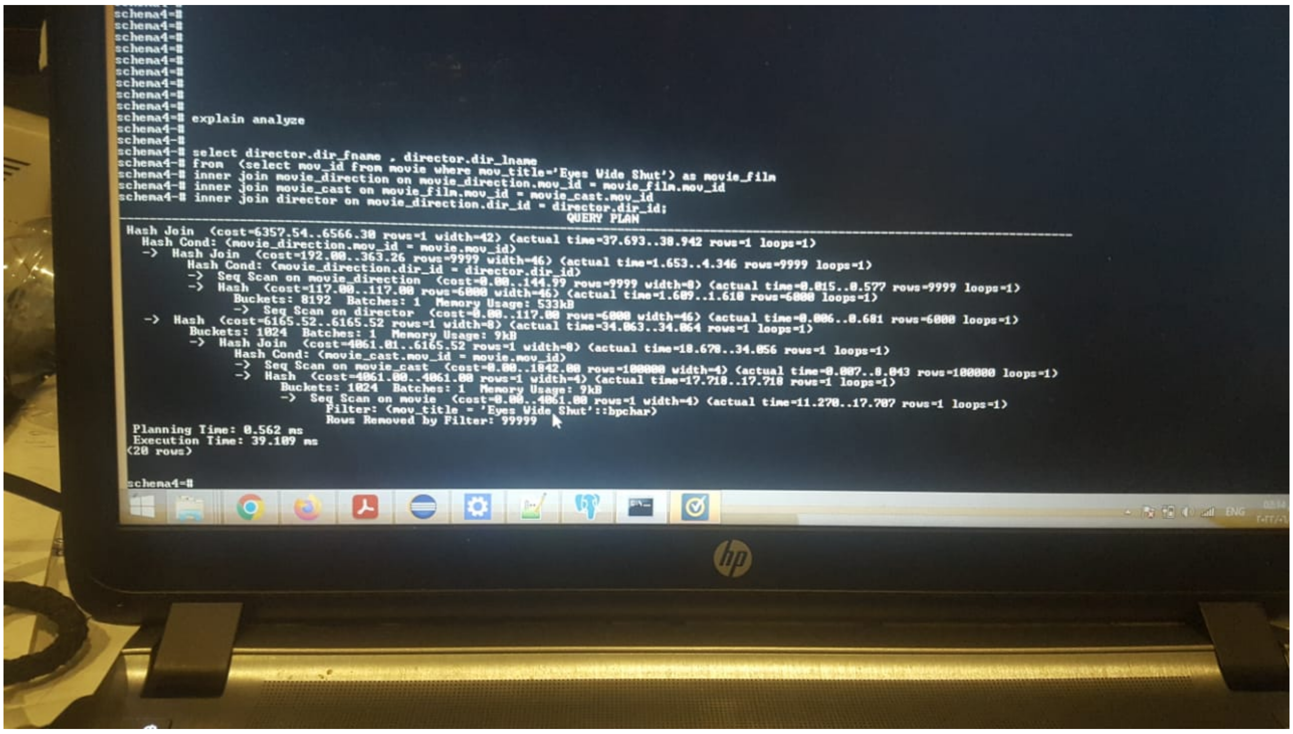
Flags: 1- set enable_seqscan = on;
2- set enable_bitmapscan =on;
3-set enable_indexscan = on;
4- set enable_indexonlyscan = on;

*************CONCLUSION OLD QUERY 11 HASH INDEXES:****************

1-OLD QUERY 11 NO INDEXES COST = 8638
2-OLD QUERY 11 NO INDEXES TIME = 53 MS
3-OLD QUERY 11 HASH INDEXES COST = 16.2
4-OLD QUERY 11 HASH INDEXES TIME = 0.1 MS
=>99.8% more cost efficient &99.8% more time efficient.

4)A)OPTIMIZED QUERY 11 WITH NO INDEXES:

Flags:1- set enable_seqscan = on;
2- set enable_bitmapscan =off;
3-set enable_indexscan = off;
4- set enable_indexonlyscan = off;



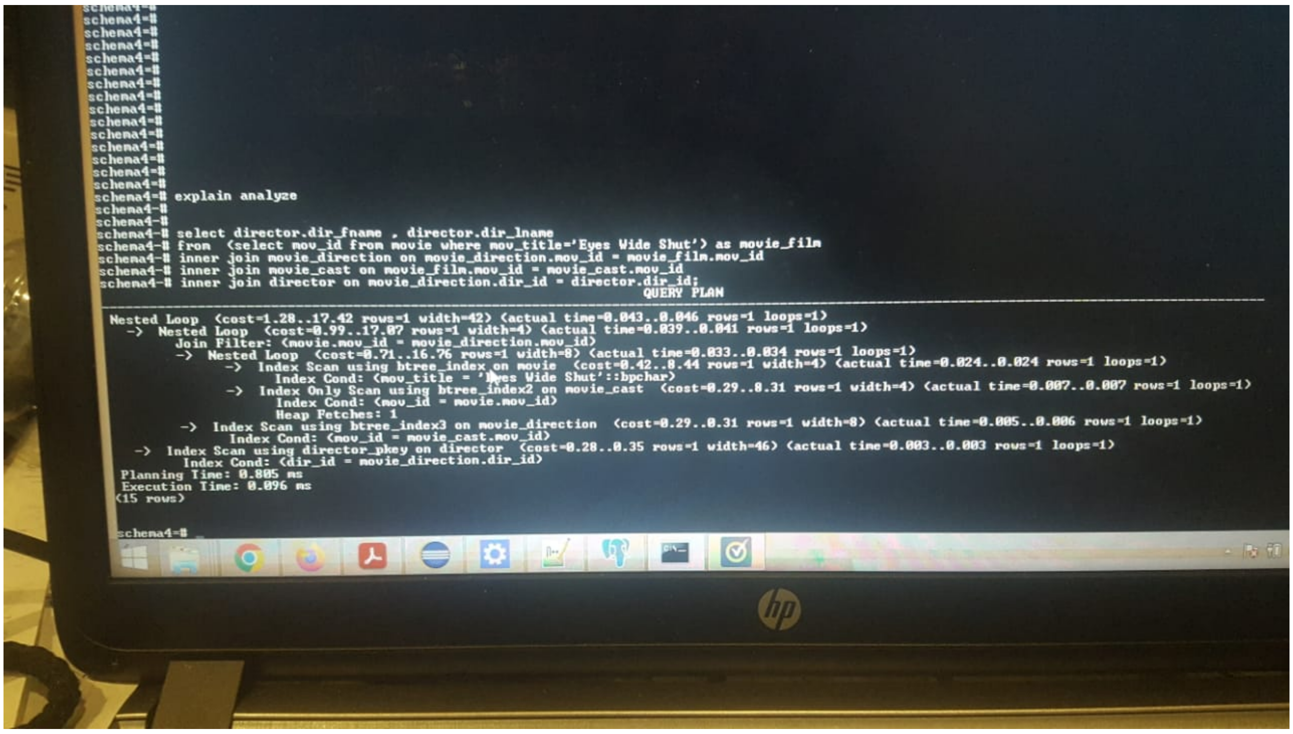
```
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=# explain analyze
schema4-#
schema4-#
schema4-# select director.dir_fname , director.dir_lname
schema4-# from  (select mov_id from movie where mov_title='Eyes Wide Shut') as movie_film
schema4-# inner join movie_direction on movie_direction.mov_id = movie_film.mov_id
schema4-# inner join movie_cast on movie_film.mov_id = movie_cast.mov_id
schema4-# inner join director on movie_direction.dir_id = director.dir_id;
                                                   QUERY PLAN
--------------------------------------------------------------------------------------------------------------------
 Hash Join  (cost=6357.54..6566.30 rows=1 width=42) (actual time=37.693..38.942 rows=1 loops=1)
   Hash Cond: (movie_direction.mov_id = movie.mov_id)
   ->  Hash Join  (cost=192.00..363.26 rows=9999 width=46) (actual time=1.653..4.346 rows=9999 loops=1)
         Hash Cond: (movie_direction.dir_id = director.dir_id)
         ->  Seq Scan on movie_direction  (cost=0.00..144.99 rows=9999 width=8) (actual time=0.015..0.577 rows=9999 loops=1)
         ->  Hash  (cost=117.00..117.00 rows=6000 width=46) (actual time=1.609..1.610 rows=6000 loops=1)
               Buckets: 8192  Batches: 1  Memory Usage: 533kB
               ->  Seq Scan on director  (cost=0.00..117.00 rows=6000 width=46) (actual time=0.006..0.681 rows=6000 loops=1)
   ->  Hash  (cost=6165.52..6165.52 rows=1 width=8) (actual time=34.063..34.064 rows=1 loops=1)
         Buckets: 1024  Batches: 1  Memory Usage: 9kB
         ->  Hash Join  (cost=4061.01..6165.52 rows=1 width=8) (actual time=18.670..34.056 rows=1 loops=1)
               Hash Cond: (movie_cast.mov_id = movie.mov_id)
               ->  Seq Scan on movie_cast  (cost=0.00..1842.00 rows=100000 width=4) (actual time=0.007..8.043 rows=100000 loops=1)
               ->  Hash  (cost=4061.00..4061.00 rows=1 width=4) (actual time=17.718..17.718 rows=1 loops=1)
                     Buckets: 1024  Batches: 1  Memory Usage: 9kB
                     ->  Seq Scan on movie  (cost=0.00..4061.00 rows=1 width=4) (actual time=11.270..17.707 rows=1 loops=1)
                           Filter: (mov_title = 'Eyes Wide Shut'::bpchar)
                           Rows Removed by Filter: 99999
 Planning Time: 0.562 ms
 Execution Time: 39.109 ms
(20 rows)


schema4=#
```

**********CONCLUSION OPTIMIZED QUERY 11 NO INDEXES:**********
1-OLD QUERY 11 NO INDEXES COST = 8638
2-OLD QUERY 11 NO INDEXES TIME = 53 MS
3-OPTIMIZED QUERY 11 NO INDEXES COST = 6566
4-OPTIMIZED QUERY 11 NO INDEXES TIME = 39 MS
=>24% more cost efficient &26% more time efficient.

4)B)OPTIMIZED QUERY 11 WITH B+ INDEXES:

B+Indexes:
1- All default PKs indexes.
2 - movie(mov_title).
3 - movie_cast(mov_id).
4 -movie_direction(mov_id).

Flags: 1- set enable_seqscan = on;
2- set enable_bitmapscan =on;
3-set enable_indexscan = on;
4- set enable_indexonlyscan = on;



```
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=# explain analyze
schema4-#
schema4-#
schema4-# select director.dir_fname , director.dir_lname
schema4-# from  (select mov_id from movie where mov_title='Eyes Wide Shut') as movie_film
schema4-# inner join movie_direction on movie_direction.mov_id = movie_film.mov_id
schema4-# inner join movie_cast on movie_film.mov_id = movie_cast.mov_id
schema4-# inner join director on movie_direction.dir_id = director.dir_id;
                                                                 QUERY PLAN
---------------------------------------------------------------------------------------------------------------------------------------
 Nested Loop  (cost=1.28..17.42 rows=1 width=42) (actual time=0.043..0.046 rows=1 loops=1)
   ->  Nested Loop  (cost=0.99..17.07 rows=1 width=4) (actual time=0.039..0.041 rows=1 loops=1)
         Join Filter: (movie.mov_id = movie_direction.mov_id)
         ->  Nested Loop  (cost=0.71..16.76 rows=1 width=8) (actual time=0.033..0.034 rows=1 loops=1)
               ->  Index Scan using btree_index on movie  (cost=0.42..8.44 rows=1 width=4) (actual time=0.024..0.024 rows=1 loops=1)
                     Index Cond: (mov_title = 'Eyes Wide Shut'::bpchar)
               ->  Index Only Scan using btree_index2 on movie_cast  (cost=0.29..8.31 rows=1 width=4) (actual time=0.007..0.007 rows=1 loops=1)
                     Index Cond: (mov_id = movie.mov_id)
                     Heap Fetches: 1
         ->  Index Scan using btree_index3 on movie_direction  (cost=0.29..0.31 rows=1 width=8) (actual time=0.005..0.006 rows=1 loops=1)
               Index Cond: (mov_id = movie_cast.mov_id)
   ->  Index Scan using director_pkey on director  (cost=0.28..0.35 rows=1 width=46) (actual time=0.003..0.003 rows=1 loops=1)
         Index Cond: (dir_id = movie_direction.dir_id)
 Planning Time: 0.805 ms
 Execution Time: 0.096 ms
(15 rows)


schema4=# _
```

*************CONCLUSION OPTIMIZED QUERY 11 B+ INDEXES:****************

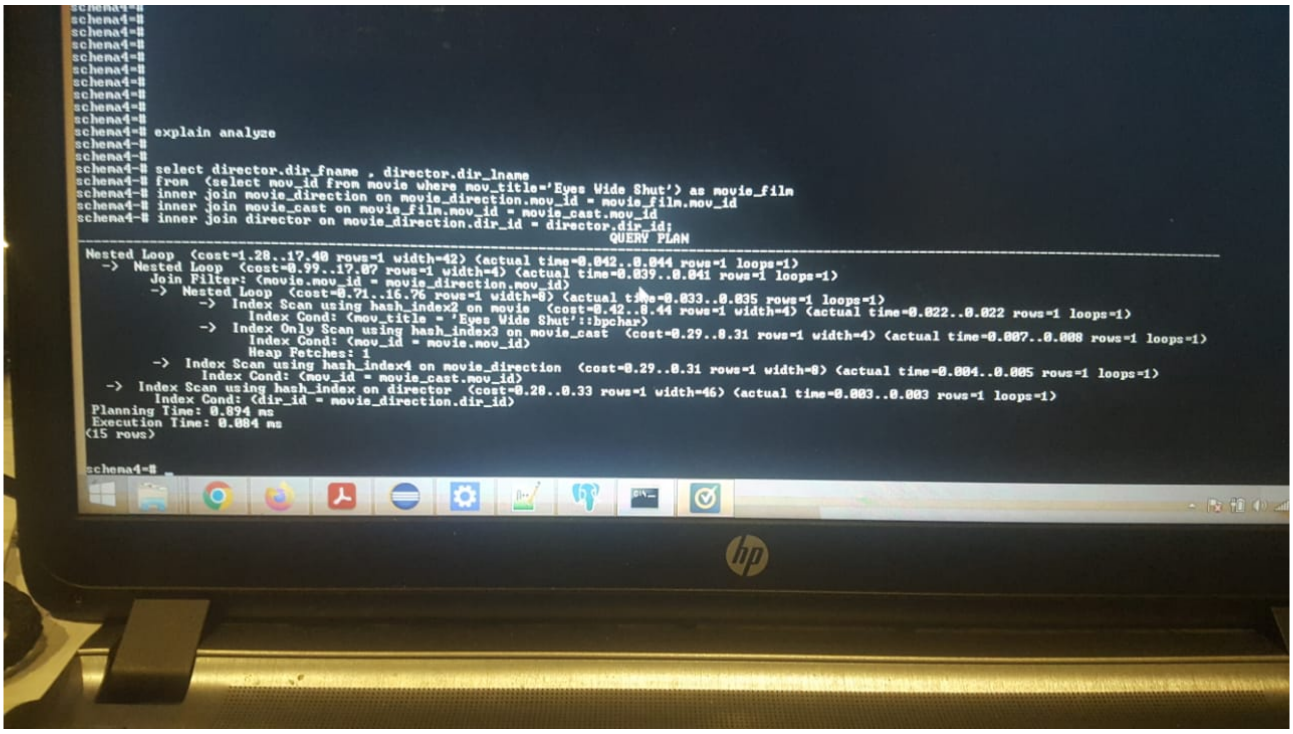
1-OLD QUERY 11 B+ INDEXES COST = 18
2-OLD QUERY 11 B+ INDEXES TIME = 0.1 MS
3-OPTMIZED QUERY 11 B+ INDEXES COST = 17
4-OPTIMIZED QUERY 11 B+ INDEXES TIME = 0.09 MS

=> 5.5% more cost efficient & 10% more time efficient.

4)C)OPTIMIZED QUERY 11 WITH Hash INDEXES:

Hash indexes: 1- director(dir_id)
2- movie_cast(mov_id)
3 - movie_direction(mov_id)
4-movie(mov_title)

Flags: 1- set enable_seqscan = on;
2- set enable_bitmapscan =on;
3-set enable_indexscan = on;
4- set enable_indexonlyscan = on;



```
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=# explain analyze
schema4-#
schema4-#
schema4-# select director.dir_fname , director.dir_lname
schema4-# from  (select mov_id from movie where mov_title='Eyes Wide Shut') as movie_film
schema4-# inner join movie_direction on movie_direction.mov_id = movie_film.mov_id
schema4-# inner join movie_cast on movie_film.mov_id = movie_cast.mov_id
schema4-# inner join director on movie_direction.dir_id = director.dir_id;
                                                   QUERY PLAN
---------------------------------------------------------------------------------------------------------------------------
 Nested Loop  (cost=1.28..17.40 rows=1 width=42) (actual time=0.042..0.044 rows=1 loops=1)
   ->  Nested Loop  (cost=0.99..17.07 rows=1 width=4) (actual time=0.039..0.041 rows=1 loops=1)
         Join Filter: (movie.mov_id = movie_direction.mov_id)
         ->  Nested Loop  (cost=0.71..16.76 rows=1 width=8) (actual time=0.033..0.035 rows=1 loops=1)
               ->  Index Scan using hash_index2 on movie  (cost=0.42..0.44 rows=1 width=4) (actual time=0.022..0.022 rows=1 loops=1)
                     Index Cond: (mov_title = 'Eyes Wide Shut'::bpchar)
               ->  Index Only Scan using hash_index3 on movie_cast  (cost=0.29..0.31 rows=1 width=4) (actual time=0.007..0.008 rows=1 loops=1)
                     Index Cond: (mov_id = movie.mov_id)
                     Heap Fetches: 1
         ->  Index Scan using hash_index4 on movie_direction  (cost=0.29..0.31 rows=1 width=8) (actual time=0.004..0.005 rows=1 loops=1)
               Index Cond: (mov_id = movie_cast.mov_id)
   ->  Index Scan using hash_index on director  (cost=0.28..0.33 rows=1 width=46) (actual time=0.003..0.003 rows=1 loops=1)
         Index Cond: (dir_id = movie_direction.dir_id)
 Planning Time: 0.894 ms
 Execution Time: 0.084 ms
(15 rows)

schema4=# _
```

*************CONCLUSION OPTIMZED QUERY 11 HASH INDEXES:************

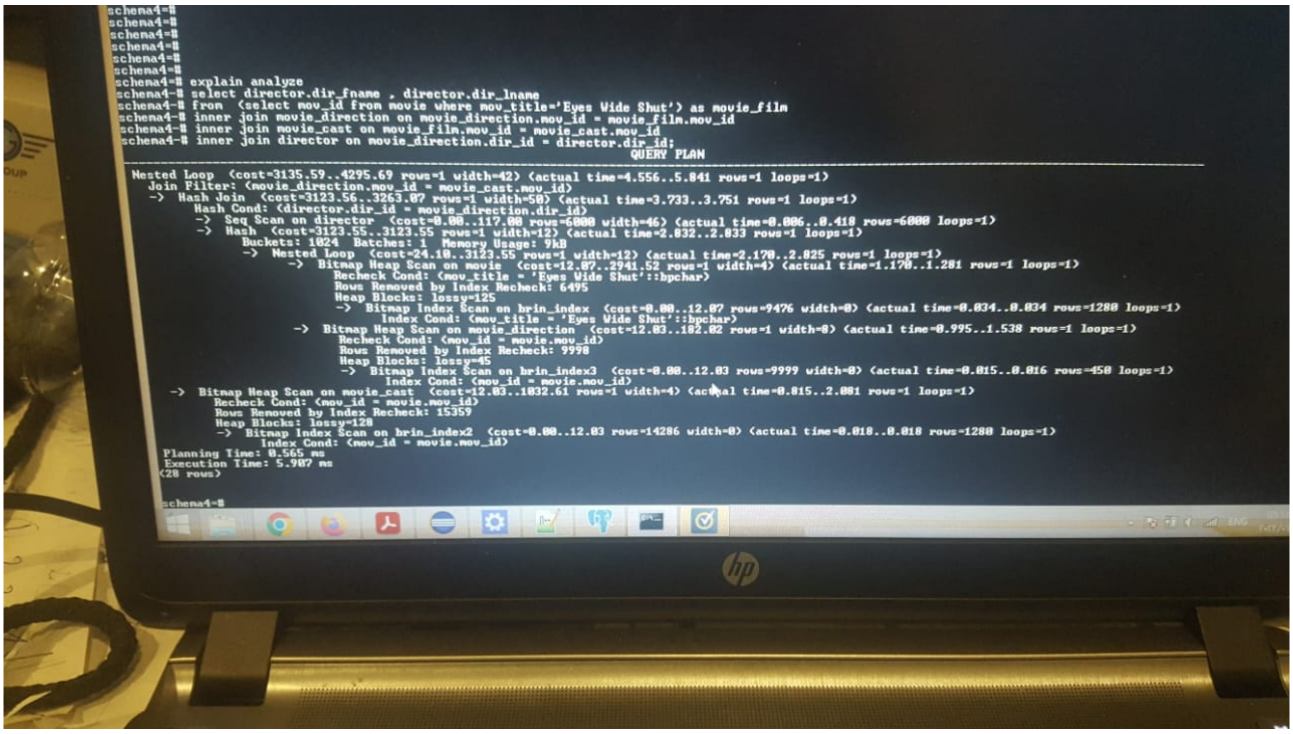
1-OLD QUERY 11 HASH INDEXES COST = 16.2
2-OLD QUERY 11 HASH INDEXES TIME = 0.1 MS
3-OPTIMIZED QUERY 11 HASH INDEXES COST = 17
4-OPTMIZED QUERY 11 HASH INDEXES TIME = 0.084 MS

=> APPROX SAME COST & TIME

4)D)OPTIMIZED QUERY 11 WITH BRIN INDEX:

BRIN INDEXES: 1-movie(mov_title)
2- movie_cast(mov_id)
3- movie_direction(mov_id)

Flags: 1- set enable_seqscan = on
2- set enable_bitmapscan =on;
3-set enable_indexscan = on;
4- set enable_indexonlyscan = on;
5- update pg_index set indisvalid = false where indexrelid = ‘director_pkey'::regclass;
6- update pg_index set indisvalid = false where indexrelid = ‘movie_direction_pkey'::regclass ;
7- update pg_index set indisvalid = false where indexrelid = ‘movie_cast_pkey'::regclass ;



```
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=# explain analyze
schema4-# select director.dir_fname , director.dir_lname
schema4-# from  (select mov_id from movie where mov_title='Eyes Wide Shut') as movie_film
schema4-# inner join movie_direction on movie_direction.mov_id = movie_film.mov_id
schema4-# inner join movie_cast on movie_film.mov_id = movie_cast.mov_id
schema4-# inner join director on movie_direction.dir_id = director.dir_id;
                                                                  QUERY PLAN
----------------------------------------------------------------------------------------------------------------------------------
 Nested Loop  (cost=3135.59..4295.69 rows=1 width=42) (actual time=4.556..5.841 rows=1 loops=1)
   Join Filter: (movie_direction.mov_id = movie_cast.mov_id)
   ->  Hash Join  (cost=3123.56..3263.07 rows=1 width=50) (actual time=3.733..3.751 rows=1 loops=1)
         Hash Cond: (director.dir_id = movie_direction.dir_id)
         ->  Seq Scan on director  (cost=0.00..117.00 rows=6000 width=46) (actual time=0.006..0.418 rows=6000 loops=1)
         ->  Hash  (cost=3123.55..3123.55 rows=1 width=12) (actual time=2.832..2.833 rows=1 loops=1)
               Buckets: 1024  Batches: 1  Memory Usage: 9kB
               ->  Nested Loop  (cost=24.10..3123.55 rows=1 width=12) (actual time=2.170..2.825 rows=1 loops=1)
                     ->  Bitmap Heap Scan on movie  (cost=12.07..2941.52 rows=1 width=4) (actual time=1.170..1.281 rows=1 loops=1)
                           Recheck Cond: (mov_title = 'Eyes Wide Shut'::bpchar)
                           Rows Removed by Index Recheck: 6495
                           Heap Blocks: lossy=125
                           ->  Bitmap Index Scan on brin_index  (cost=0.00..12.07 rows=9476 width=0) (actual time=0.034..0.034 rows=1280 loops=1)
                                 Index Cond: (mov_title = 'Eyes Wide Shut'::bpchar)
                     ->  Bitmap Heap Scan on movie_direction  (cost=12.03..182.02 rows=1 width=8) (actual time=0.995..1.538 rows=1 loops=1)
                           Recheck Cond: (mov_id = movie.mov_id)
                           Rows Removed by Index Recheck: 9998
                           Heap Blocks: lossy=45
                           ->  Bitmap Index Scan on brin_index3  (cost=0.00..12.03 rows=9999 width=0) (actual time=0.015..0.016 rows=450 loops=1)
                                 Index Cond: (mov_id = movie.mov_id)
   ->  Bitmap Heap Scan on movie_cast  (cost=12.03..1032.61 rows=1 width=4) (actual time=0.015..2.081 rows=1 loops=1)
         Recheck Cond: (mov_id = movie.mov_id)
         Rows Removed by Index Recheck: 15359
         Heap Blocks: lossy=128
         ->  Bitmap Index Scan on brin_index2  (cost=0.00..12.03 rows=14286 width=0) (actual time=0.018..0.018 rows=1280 loops=1)
               Index Cond: (mov_id = movie.mov_id)
 Planning Time: 0.565 ms
 Execution Time: 5.907 ms
(28 rows)

schema4=#
```

*************CONCLUSION OPTIMIZED QUERY 11 BRIN INDEX:****************
1-OLD QUERY 11 BRIN INDEX COST = 6382
2-OLD QUERY 11 BRIN INDEX TIME = 27 MS
3-OPTIMIZED QUERY 11 BRIN INDEX COST = 4295
4-OPTIMIZED QUERY 11 BRIN INDEX TIME = 6 MS
=>32.7% more cost efficient &77.7% more time efficient.
****

4)E)OPTIMIZED QUERY 11 WITH MY CHOICE FOR INDEXES:

Same as 4)c) using only hash indexes as all the conditions are equality conditions.

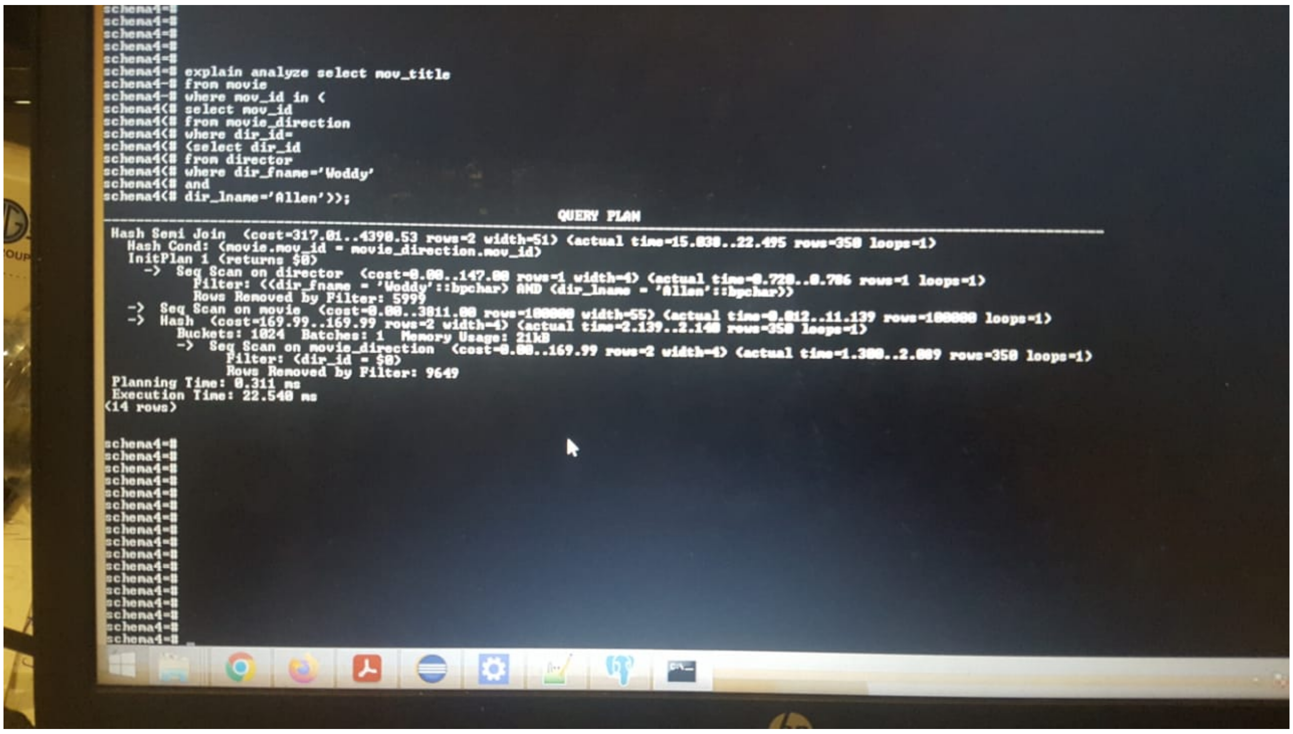
Hash indexes: 1- director(dir_id)
2- movie_cast(mov_id)
3 - movie_direction(mov_id)
4-movie(mov_title)

Flags: 1- set enable_seqscan = on;
2- set enable_bitmapscan =on;
3-set enable_indexscan = on;
4- set enable_indexonlyscan = on;

*************CONCLUSION OPTIMIZED QUERY 11 HASH INDEXES:*************
1-OLD QUERY 11 HASH INDEXES COST = 16.2
2-OLD QUERY 11 HASH INDEXES TIME = 0.1 MS
3-OPTIMIZED QUERY 11 HASH INDEXES COST = 17
4-OPTMIZED QUERY 11 HASH INDEXES TIME = 0.084 MS
=> APPROX SAME COST & TIME

5)A)OLD QUERY 12 WITH NO INDEXES:

Flags:1- set enable_seqscan = on;
2- set enable_bitmapscan =off;
3-set enable_indexscan = off;
4- set enable_indexonlyscan = off;



```
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=# explain analyze select mov_title
schema4-# from movie
schema4-# where mov_id in (
schema4(# select mov_id
schema4(# from movie_direction
schema4(# where dir_id=
schema4(# (select dir_id
schema4(# from director
schema4(# where dir_fname='Woddy'
schema4(# and
schema4(# dir_lname='Allen'));
                                                    QUERY PLAN
------------------------------------------------------------------------------------------------------------------------
 Hash Semi Join  (cost=317.01..4390.53 rows=2 width=51) (actual time=15.030..22.495 rows=350 loops=1)
   Hash Cond: (movie.mov_id = movie_direction.mov_id)
   InitPlan 1 (returns $0)
     ->  Seq Scan on director  (cost=0.00..147.00 rows=1 width=4) (actual time=0.720..0.786 rows=1 loops=1)
           Filter: ((dir_fname = 'Woddy'::bpchar) AND (dir_lname = 'Allen'::bpchar))
           Rows Removed by Filter: 5999
   ->  Seq Scan on movie  (cost=0.00..3011.00 rows=100000 width=55) (actual time=0.012..11.139 rows=100000 loops=1)
   ->  Hash  (cost=169.99..169.99 rows=2 width=4) (actual time=2.139..2.140 rows=350 loops=1)
         Buckets: 1024  Batches: 1  Memory Usage: 21kB
         ->  Seq Scan on movie_direction  (cost=0.00..169.99 rows=2 width=4) (actual time=1.300..2.009 rows=350 loops=1)
               Filter: (dir_id = $0)
               Rows Removed by Filter: 9649
 Planning Time: 0.311 ms
 Execution Time: 22.540 ms
(14 rows)


schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
```

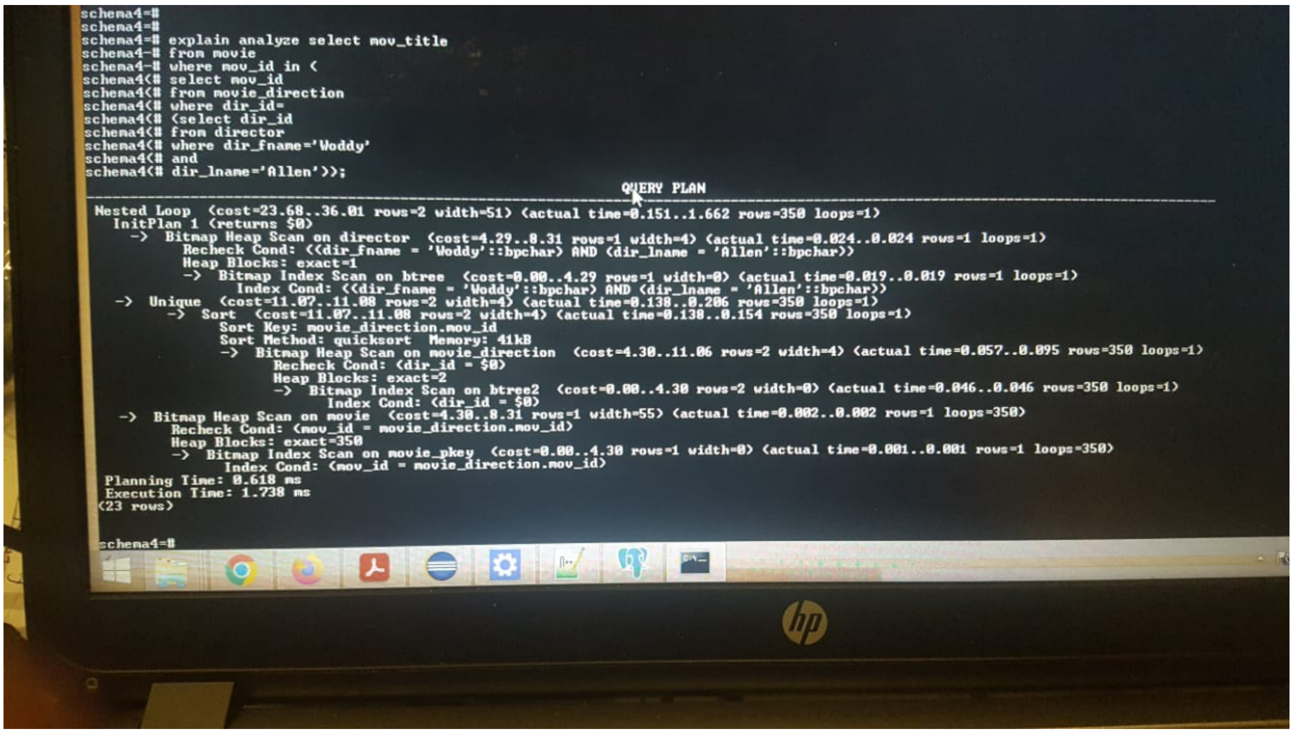
******************CONCLUSION OLD QUERY 12 NO INDEXES:*******************

1-OLD QUERY 12 NO INDEXES COST = 4390
2-OLD QUERY 12 NO INDEXES TIME = 22.5 MS

5)B)OLD QUERY 12 WITH B+ INDEXES:

Flags:1- set enable_seqscan = on;
2- set enable_bitmapscan =on;
3-set enable_indexscan = on;
4- set enable_indexonlyscan = on;

B+ Indexes: 1- movie_direction(dir_id)
2 - multi column B+ tree on director(dir_fname,dir_lname)
3- any default PKs B+ trees



```
schema4=#
schema4=#
schema4=# explain analyze select mov_title
schema4-# from movie
schema4-# where mov_id in (
schema4(# select mov_id
schema4(# from movie_direction
schema4(# where dir_id=
schema4(# (select dir_id
schema4(# from director
schema4(# where dir_fname='Woddy'
schema4(# and
schema4(# dir_lname='Allen'));
                                                QUERY PLAN
------------------------------------------------------------------------------------------------------------------------------------
 Nested Loop  (cost=23.68..36.01 rows=2 width=51) (actual time=0.151..1.662 rows=350 loops=1)
   InitPlan 1 (returns $0)
     ->  Bitmap Heap Scan on director  (cost=4.29..8.31 rows=1 width=4) (actual time=0.024..0.024 rows=1 loops=1)
           Recheck Cond: ((dir_fname = 'Woddy'::bpchar) AND (dir_lname = 'Allen'::bpchar))
           Heap Blocks: exact=1
           ->  Bitmap Index Scan on btree  (cost=0.00..4.29 rows=1 width=0) (actual time=0.019..0.019 rows=1 loops=1)
                 Index Cond: ((dir_fname = 'Woddy'::bpchar) AND (dir_lname = 'Allen'::bpchar))
   ->  Unique  (cost=11.07..11.08 rows=2 width=4) (actual time=0.138..0.206 rows=350 loops=1)
         ->  Sort  (cost=11.07..11.08 rows=2 width=4) (actual time=0.138..0.154 rows=350 loops=1)
               Sort Key: movie_direction.mov_id
               Sort Method: quicksort  Memory: 41kB
               ->  Bitmap Heap Scan on movie_direction  (cost=4.30..11.06 rows=2 width=4) (actual time=0.057..0.095 rows=350 loops=1)
                     Recheck Cond: (dir_id = $0)
                     Heap Blocks: exact=2
                     ->  Bitmap Index Scan on btree2  (cost=0.00..4.30 rows=2 width=0) (actual time=0.046..0.046 rows=350 loops=1)
                           Index Cond: (dir_id = $0)
   ->  Bitmap Heap Scan on movie  (cost=4.30..8.31 rows=1 width=55) (actual time=0.002..0.002 rows=1 loops=350)
         Recheck Cond: (mov_id = movie_direction.mov_id)
         Heap Blocks: exact=350
         ->  Bitmap Index Scan on movie_pkey  (cost=0.00..4.30 rows=1 width=0) (actual time=0.001..0.001 rows=1 loops=350)
               Index Cond: (mov_id = movie_direction.mov_id)
 Planning Time: 0.618 ms
 Execution Time: 1.738 ms
(23 rows)

schema4=#
```

*****************CONCLUSION OLD QUERY 12 B+ INDEXES:*******************

1-OLD QUERY 12 NO INDEXES COST = 4390
2-OLD QUERY 12 NO INDEXES TIME = 22.5 MS

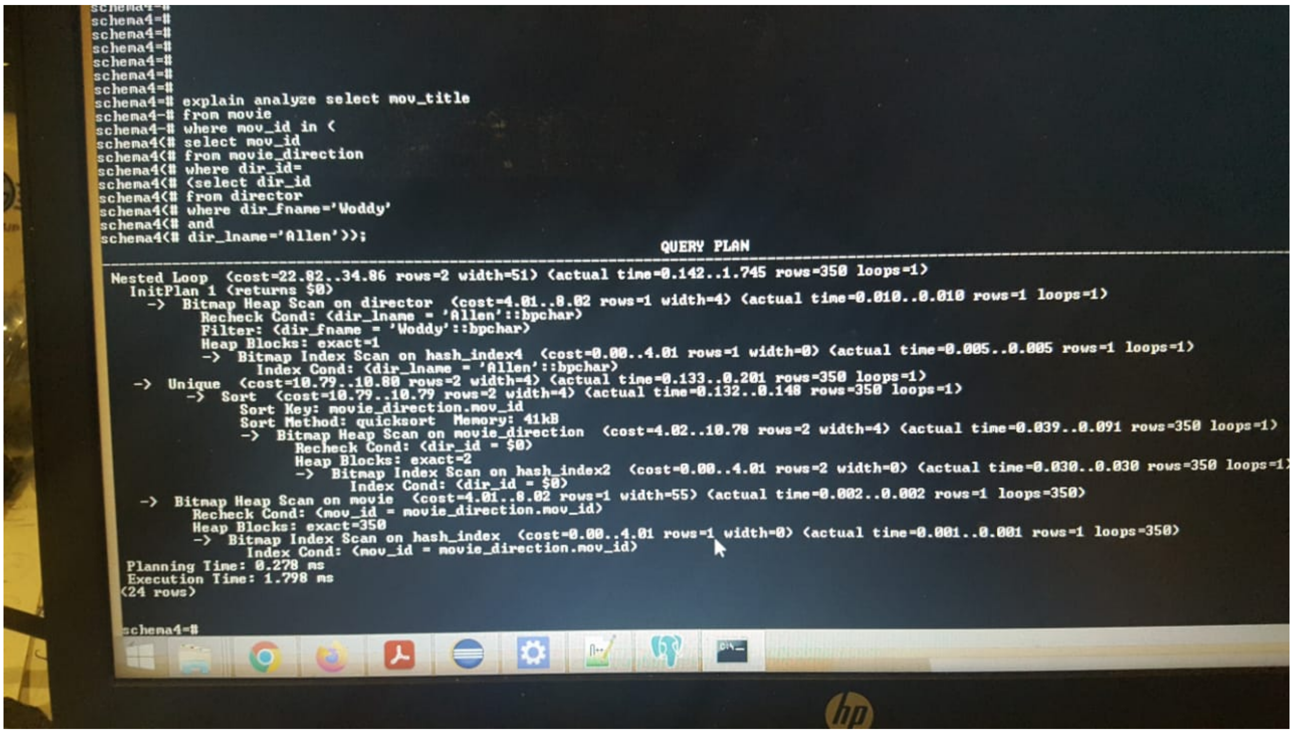
— —

3-OLD QUERY 12 B+ INDEXES COST = 36
4-OLD QUERY 12 B+ INDEXES TIME = 1.7 MS

=>99% more cost efficient & 92.4 more time efficient

5)C)OLD QUERY 12 WITH HASH INDEXES:

Flags:1- set enable_seqscan = on;
2- set enable_bitmapscan =on;
3-set enable_indexscan = on;
4- set enable_indexonlyscan = on;
Hash Indexes:1 - director(dir_lname)
2 - movie_direction(dir_id)
3 - movie(mov_id)
NOTE : Postgres doesn't support multi column indexes using Hash yet
So couldn't make a single hash on dir_lname & dir_fname
Also if a single hash table is created for dir_fname and another for dir_lname the query optimiser uses only one and fetches the other using filter but doesn't use both hash tables so I had to choose a single hash on dir_lname .



```
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=#
schema4=# explain analyze select mov_title
schema4-# from movie
schema4-# where mov_id in (
schema4(# select mov_id
schema4(# from movie_direction
schema4(# where dir_id=
schema4(# (select dir_id
schema4(# from director
schema4(# where dir_fname='Woddy'
schema4(# and
schema4(# dir_lname='Allen'));
                                                        QUERY PLAN
--------------------------------------------------------------------------------------------------------------------------------
 Nested Loop  (cost=22.82..34.86 rows=2 width=51) (actual time=0.142..1.745 rows=350 loops=1)
   InitPlan 1 (returns $0)
     ->  Bitmap Heap Scan on director  (cost=4.01..8.02 rows=1 width=4) (actual time=0.010..0.010 rows=1 loops=1)
           Recheck Cond: (dir_lname = 'Allen'::bpchar)
           Filter: (dir_fname = 'Woddy'::bpchar)
           Heap Blocks: exact=1
           ->  Bitmap Index Scan on hash_index4  (cost=0.00..4.01 rows=1 width=0) (actual time=0.005..0.005 rows=1 loops=1)
                 Index Cond: (dir_lname = 'Allen'::bpchar)
   ->  Unique  (cost=10.79..10.80 rows=2 width=4) (actual time=0.133..0.201 rows=350 loops=1)
         ->  Sort  (cost=10.79..10.79 rows=2 width=4) (actual time=0.132..0.148 rows=350 loops=1)
               Sort Key: movie_direction.mov_id
               Sort Method: quicksort  Memory: 41kB
               ->  Bitmap Heap Scan on movie_direction  (cost=4.02..10.78 rows=2 width=4) (actual time=0.039..0.091 rows=350 loops=1)
                     Recheck Cond: (dir_id = $0)
                     Heap Blocks: exact=2
                     ->  Bitmap Index Scan on hash_index2  (cost=0.00..4.01 rows=2 width=0) (actual time=0.030..0.030 rows=350 loops=1)
                           Index Cond: (dir_id = $0)
   ->  Bitmap Heap Scan on movie  (cost=4.01..8.02 rows=1 width=55) (actual time=0.002..0.002 rows=1 loops=350)
         Recheck Cond: (mov_id = movie_direction.mov_id)
         Heap Blocks: exact=350
         ->  Bitmap Index Scan on hash_index  (cost=0.00..4.01 rows=1 width=0) (actual time=0.001..0.001 rows=1 loops=350)
               Index Cond: (mov_id = movie_direction.mov_id)
 Planning Time: 0.278 ms
 Execution Time: 1.798 ms
(24 rows)

schema4=#
```

******************CONCLUSION OLD QUERY 12 Hash INDEXES:***************

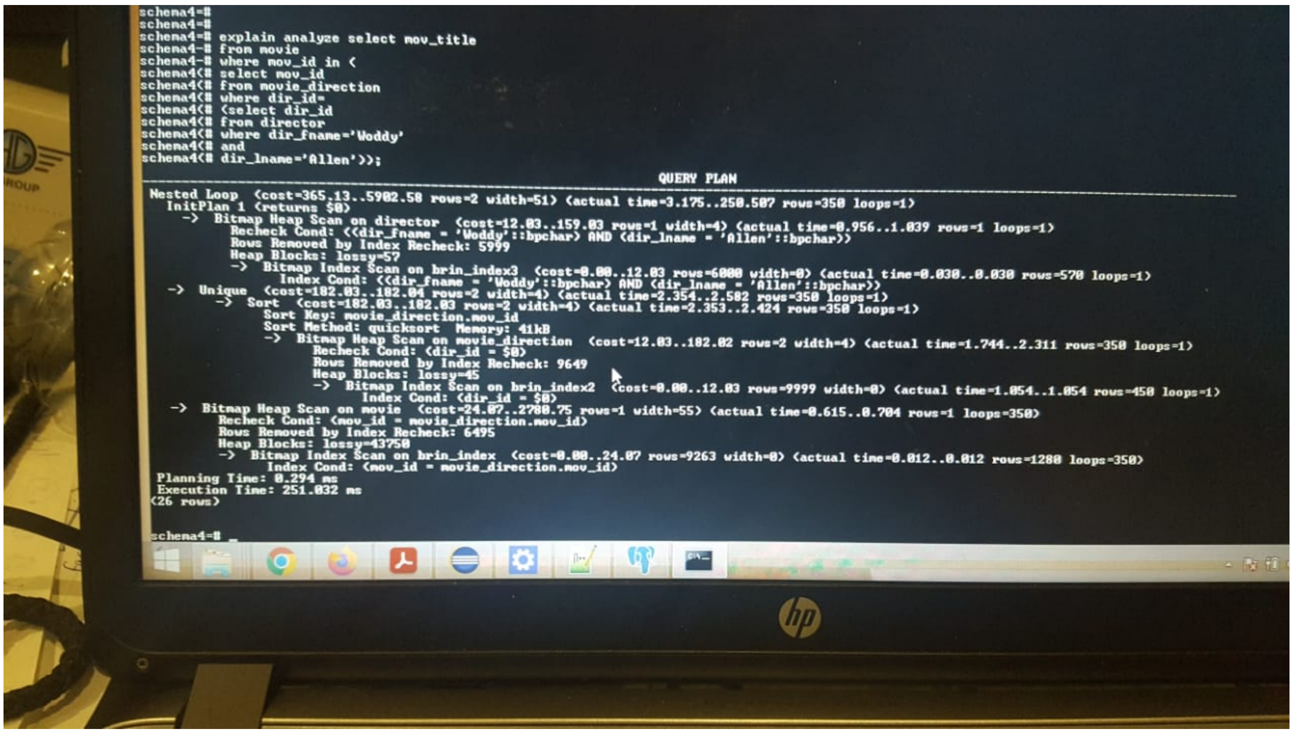
1-OLD QUERY 12 NO INDEXES COST = 4390
2-OLD QUERY 12 NO INDEXES TIME = 22.5 MS
————————————————————————

3-OLD QUERY 12 Hash INDEXES COST = 35
4-OLD QUERY 12 Hash INDEXES TIME = 1.8 MS
=>99% more cost efficient & 92.4 more time efficient
****

5)D)OLD QUERY 12 WITH BRIN INDEX:

Flags:1- set enable_seqscan = on;
2- set enable_bitmapscan =on;
3-set enable_indexscan = on;
4- set enable_indexonlyscan = on;
5- update pg_index set indisvalid = false where indexrelid = 'movie_pkey'::regclass;
6- update pg_index set indisvalid = false where indexrelid = 'movie_direction_pkey'::regclass;

BRIN Indexes: 1- movie(mov_id)
2 - movie_direction(dir_id)
3 - multi column BRIN director(dir_fname,dir_lname)



```
schema4=#
schema4=#
schema4=# explain analyze select mov_title
schema4-# from movie
schema4-# where mov_id in (
schema4(# select mov_id
schema4(# from movie_direction
schema4(# where dir_id=
schema4(# (select dir_id
schema4(# from director
schema4(# where dir_fname='Woddy'
schema4(# and
schema4(# dir_lname='Allen'));
                                        QUERY PLAN
----------------------------------------------------------------------------------------------------------
 Nested Loop  (cost=365.13..5902.58 rows=2 width=51) (actual time=3.175..250.507 rows=350 loops=1)
   InitPlan 1 (returns $0)
     ->  Bitmap Heap Scan on director  (cost=12.03..159.03 rows=1 width=4) (actual time=0.956..1.039 rows=1 loops=1)
           Recheck Cond: ((dir_fname = 'Woddy'::bpchar) AND (dir_lname = 'Allen'::bpchar))
           Rows Removed by Index Recheck: 5999
           Heap Blocks: lossy=57
           ->  Bitmap Index Scan on brin_index3  (cost=0.00..12.03 rows=6000 width=0) (actual time=0.030..0.030 rows=570 loops=1)
                 Index Cond: ((dir_fname = 'Woddy'::bpchar) AND (dir_lname = 'Allen'::bpchar))
   ->  Unique  (cost=182.03..182.04 rows=2 width=4) (actual time=2.354..2.582 rows=350 loops=1)
         ->  Sort  (cost=182.03..182.03 rows=2 width=4) (actual time=2.353..2.424 rows=350 loops=1)
               Sort Key: movie_direction.mov_id
               Sort Method: quicksort  Memory: 41kB
               ->  Bitmap Heap Scan on movie_direction  (cost=12.03..182.02 rows=2 width=4) (actual time=1.744..2.311 rows=350 loops=1)
                     Recheck Cond: (dir_id = $0)
                     Rows Removed by Index Recheck: 9649
                     Heap Blocks: lossy=45
                     ->  Bitmap Index Scan on brin_index2  (cost=0.00..12.03 rows=9999 width=0) (actual time=1.054..1.054 rows=450 loops=1)
                           Index Cond: (dir_id = $0)
   ->  Bitmap Heap Scan on movie  (cost=24.07..2780.75 rows=1 width=55) (actual time=0.615..0.704 rows=1 loops=350)
         Recheck Cond: (mov_id = movie_direction.mov_id)
         Rows Removed by Index Recheck: 6495
         Heap Blocks: lossy=43750
         ->  Bitmap Index Scan on brin_index  (cost=0.00..24.07 rows=9263 width=0) (actual time=0.012..0.012 rows=1280 loops=350)
               Index Cond: (mov_id = movie_direction.mov_id)
 Planning Time: 0.294 ms
 Execution Time: 251.032 ms
(26 rows)

schema4=# _
```

******************CONCLUSION OLD QUERY 12 BRIN INDEXES:****************

1-OLD QUERY 12 NO INDEXES COST = 4390
2-OLD QUERY 12 NO INDEXES TIME = 22.5 MS

———————————————————————

3-OLD QUERY 12 BRIN INDEXES COST = 5902
4-OLD QUERY 12 BRIN INDEXES TIME = 251 MS
=> Much less efficient cost and time wise compared to no indexes this is
This is mainly because BRINs should be on very large tables
(On the clustering columns) .

****

5)E)OPTIMIZED QUERY 12 WITH MY CHOICE FOR INDEXES:

I choose B+trees same as 5)b)
Flags:1- set enable_seqscan = on;
2- set enable_bitmapscan =on;
3-set enable_indexscan = on;
4- set enable_indexonlyscan = on;
B+ Indexes: 1- movie_direction(dir_id)
2 - multi column B+ tree on director(dir_fname,dir_lname)
3- any default PKs B+ trees
*OLD QUERY 12 B+ INDEXES COST = 36
*OLD QUERY 12 B+ INDEXES TIME = 1.7 MS

6)optimised query 12:

No better query than the old query as the join won’t give any additional benefits ,yes this is a nested query but it contains ZERO dependencies
 So it is a perfect nested query .