Schema 1:

Inserted 15 departments , 180 instructor where each department is linked with 12 instructors , 12000 classrooms , 10000 time slot , 9000 student where each department is linked with 600 students and each instructor is linked with 50 students , 225 courses where each department is linked with 15 courses , 225 prerequisites , 12000 sections with years between 2019 and 2021 , semester between 1 and 2 , instructor between 1 and 20 , courses between 1 and 20 , populated table takes with 45000 tuples with each 20 student is linked with 5 courses and 10000 section time

And after insertion removed all primary key constraints.

Query 1 :

Scenario #1 : without an index
All flags are on

```
"Hash Full Join  (cost=480.00..1450.88 rows=7500 width=60) (actual time=7.469..17.209 rows=8000 loops=1)"
"  Hash Cond: (t.student_id = student.student_id)"
"  ->  Hash Join  (cost=294.00..1231.75 rows=7500 width=40) (actual time=3.908..11.580 rows=8000 loops=1)"
"        Hash Cond: (t.section_id = s.section_id)"
"        ->  Seq Scan on takes t  (cost=0.00..694.00 rows=45000 width=12) (actual time=0.161..2.693 rows=45000 loops=1)"
"        ->  Hash  (cost=269.00..269.00 rows=2000 width=28) (actual time=3.590..3.590 rows=2000 loops=1)"
"              Buckets: 2048  Batches: 1  Memory Usage: 134kB"
"              ->  Seq Scan on section s  (cost=0.00..269.00 rows=2000 width=28) (actual time=0.064..2.873 rows=2000 loops=1)"
"                    Filter: ((semester = 1) AND (year = 2019))"
"                    Rows Removed by Filter: 10000"
"  ->  Hash  (cost=178.50..178.50 rows=600 width=24) (actual time=3.537..3.537 rows=600 loops=1)"
"        Buckets: 1024  Batches: 1  Memory Usage: 41kB"
"        ->  Seq Scan on student  (cost=0.00..178.50 rows=600 width=24) (actual time=0.044..3.300 rows=600 loops=1)"
"              Filter: ((department)::text = 'CS1'::text)"
"              Rows Removed by Filter: 8400"
"Planning time: 0.331 ms"
```

```
"Execution time: 17.530 ms"
```

Scenario #2: with a mixed bitmap index on columns year,semester on table section
All flags are on

```
"Hash Full Join  (cost=366.50..1337.38 rows=7500 width=60) (actual time=5.040..20.405
rows=8000 loops=1)"
"  Hash Cond: (t.student_id = student.student_id)"
"  ->  Hash Join  (cost=180.50..1118.25 rows=7500 width=40) (actual time=1.835..13.213
rows=8000 loops=1)"
"        Hash Cond: (t.section_id = s.section_id)"
"        ->  Seq Scan on takes t  (cost=0.00..694.00 rows=45000 width=12) (actual
time=0.035..3.389 rows=45000 loops=1)"
"        ->  Hash  (cost=155.50..155.50 rows=2000 width=28) (actual time=1.765..1.765
rows=2000 loops=1)"
"              Buckets: 2048  Batches: 1  Memory Usage: 134kB"
"              ->  Bitmap Heap Scan on section s  (cost=36.50..155.50 rows=2000 width=28) (actual
time=0.735..1.121 rows=2000 loops=1)"
"                    Recheck Cond: ((semester = 1) AND (year = 2019))"
"                    Heap Blocks: exact=19"
"                    ->  Bitmap Index Scan on bitmaponmixed  (cost=0.00..36.00 rows=2000 width=0)
(actual time=0.711..0.711 rows=2000 loops=1)"
"                          Index Cond: ((semester = 1) AND (year = 2019))"
"  ->  Hash  (cost=178.50..178.50 rows=600 width=24) (actual time=3.192..3.192 rows=600
loops=1)"
"        Buckets: 1024  Batches: 1  Memory Usage: 41kB"
"        ->  Seq Scan on student  (cost=0.00..178.50 rows=600 width=24) (actual
time=0.031..2.968 rows=600 loops=1)"
"              Filter: ((department)::text = 'CS1'::text)"
"              Rows Removed by Filter: 8400"
"Planning time: 2.435 ms"
"Execution time: 20.995 ms"
```

Scenario #3: with a hash index on column department on table student
All flags are on

```
"Hash Full Join  (cost=395.65..1366.53 rows=7500 width=60) (actual time=4.281..22.318
rows=8000 loops=1)"
"  Hash Cond: (t.student_id = student.student_id)"
"  ->  Hash Join  (cost=294.00..1231.75 rows=7500 width=40) (actual time=3.477..17.758
rows=8000 loops=1)"
"        Hash Cond: (t.section_id = s.section_id)"
"        ->  Seq Scan on takes t  (cost=0.00..694.00 rows=45000 width=12) (actual
time=0.091..4.746 rows=45000 loops=1)"
"        ->  Hash  (cost=269.00..269.00 rows=2000 width=28) (actual time=3.303..3.303
rows=2000 loops=1)"
"              Buckets: 2048  Batches: 1  Memory Usage: 134kB"
"              ->  Seq Scan on section s  (cost=0.00..269.00 rows=2000 width=28) (actual
time=0.045..2.595 rows=2000 loops=1)"
"                    Filter: ((semester = 1) AND (year = 2019))"
"                    Rows Removed by Filter: 10000"
"  ->  Hash  (cost=94.15..94.15 rows=600 width=24) (actual time=0.784..0.784 rows=600
loops=1)"
"        Buckets: 1024  Batches: 1  Memory Usage: 41kB"
"        ->  Bitmap Heap Scan on student  (cost=20.65..94.15 rows=600 width=24) (actual
time=0.149..0.506 rows=600 loops=1)"
"              Recheck Cond: ((department)::text = 'CS1'::text)"
"              Heap Blocks: exact=4"
"              ->  Bitmap Index Scan on hash  (cost=0.00..20.50 rows=600 width=0) (actual
time=0.125..0.125 rows=600 loops=1)"
"                    Index Cond: ((department)::text = 'CS1'::text)"
"Planning time: 0.621 ms"
"Execution time: 22.939 ms"
```

Scenario #4: with a bitmap index on column department on table student
All flags are on

```
"Hash Full Join  (cost=387.65..1358.53 rows=7500 width=60) (actual time=1.370..17.757
rows=8000 loops=1)"
"  Hash Cond: (t.student_id = student.student_id)"
"  ->  Hash Join  (cost=294.00..1231.75 rows=7500 width=40) (actual time=1.194..14.059
rows=8000 loops=1)"
"        Hash Cond: (t.section_id = s.section_id)"
"        ->  Seq Scan on takes t  (cost=0.00..694.00 rows=45000 width=12) (actual
time=0.011..4.022 rows=45000 loops=1)"
```

```
"        ->  Hash  (cost=269.00..269.00 rows=2000 width=28) (actual time=1.173..1.173
rows=2000 loops=1)"
"              Buckets: 2048  Batches: 1  Memory Usage: 134kB"
"              ->  Seq Scan on section s  (cost=0.00..269.00 rows=2000 width=28) (actual
time=0.009..0.970 rows=2000 loops=1)"
"                    Filter: ((semester = 1) AND (year = 2019))"
"                    Rows Removed by Filter: 10000"
"  ->  Hash  (cost=86.15..86.15 rows=600 width=24) (actual time=0.170..0.170 rows=600
loops=1)"
"        Buckets: 1024  Batches: 1  Memory Usage: 41kB"
"        ->  Bitmap Heap Scan on student  (cost=12.65..86.15 rows=600 width=24) (actual
time=0.060..0.095 rows=600 loops=1)"
"              Recheck Cond: ((department)::text = 'CS1'::text)"
"              Heap Blocks: exact=4"
"              ->  Bitmap Index Scan on bitmap  (cost=0.00..12.50 rows=600 width=0) (actual
time=0.053..0.053 rows=600 loops=1)"
"                    Index Cond: ((department)::text = 'CS1'::text)"
"Planning time: 1.244 ms"
"Execution time: 18.165 ms"
```

Scenario #5: with a mixed bitmap index on columns year,semester on table section , a bitmap index on column department on table student
All flags are on

```
"Hash Full Join  (cost=274.15..1245.03 rows=7500 width=60) (actual time=1.439..26.000
rows=8000 loops=1)"
"  Hash Cond: (t.student_id = student.student_id)"
"  ->  Hash Join  (cost=180.50..1118.25 rows=7500 width=40) (actual time=1.152..21.096
rows=8000 loops=1)"
"        Hash Cond: (t.section_id = s.section_id)"
"        ->  Seq Scan on takes t  (cost=0.00..694.00 rows=45000 width=12) (actual
time=0.016..6.081 rows=45000 loops=1)"
"        ->  Hash  (cost=155.50..155.50 rows=2000 width=28) (actual time=1.118..1.118
rows=2000 loops=1)"
"              Buckets: 2048  Batches: 1  Memory Usage: 134kB"
"              ->  Bitmap Heap Scan on section s  (cost=36.50..155.50 rows=2000 width=28) (actual
time=0.574..0.784 rows=2000 loops=1)"
"                    Recheck Cond: ((semester = 1) AND (year = 2019))"
"                    Heap Blocks: exact=19"
"                    ->  Bitmap Index Scan on bitmaponmixed  (cost=0.00..36.00 rows=2000 width=0)
(actual time=0.563..0.563 rows=2000 loops=1)"
"                          Index Cond: ((semester = 1) AND (year = 2019))"
"  ->  Hash  (cost=86.15..86.15 rows=600 width=24) (actual time=0.280..0.280 rows=600
loops=1)"
"        Buckets: 1024  Batches: 1  Memory Usage: 41kB"
"        ->  Bitmap Heap Scan on student  (cost=12.65..86.15 rows=600 width=24) (actual
time=0.081..0.154 rows=600 loops=1)"
"              Recheck Cond: ((department)::text = 'CS1'::text)"
"              Heap Blocks: exact=4"
"              ->  Bitmap Index Scan on bitmap  (cost=0.00..12.50 rows=600 width=0) (actual
time=0.070..0.070 rows=600 loops=1)"
"                    Index Cond: ((department)::text = 'CS1'::text)"
"Planning time: 0.299 ms"
"Execution time: 26.566 ms"
```

Conclusion of Query 1:

The best performance was in Scenario #5 , with a bitmap index on column department on table student was a good choice because it is an exact sub-query , also using mixed bitmap index on table section on columns year and semester was a good choice because it is an exact sub-query and also when filtering the table section with columns year and semester to be equal 2019 and 1 , this helped the planner to scan over all the tuples that is needed to be joined with table takes , so the index helped to enhance the join also .

Schema 4:

Inserted 1000 movies , 10000 reviewer , 10000 genres , 12000 actors half of them males and others are females , 2000 director , connecting first 10000 directors each with 10 movies , each movie with genre , each reviewer with movie , last actor with 10 movies with a role exists in movie2 (for query 11).
And after insertion removed all primary key constraints.

Query 10 :

Scenario #1 : without an index
All flags are on

```
"Hash Semi Join  (cost=243.00..507.61 rows=10 width=48) (actual time=3.158..6.187 rows=10 loops=1)"
"  Hash Cond: (actor.act_id = movie_cast.act_id)"
"  ->  Seq Scan on actor  (cost=0.00..233.00 rows=12000 width=48) (actual time=0.017..1.432 rows=12000 loops=1)"
"  ->  Hash  (cost=242.87..242.87 rows=10 width=4) (actual time=3.127..3.127 rows=10 loops=1)"
"        Buckets: 1024  Batches: 1  Memory Usage: 9kB"
"        ->  Hash Semi Join  (cost=32.51..242.87 rows=10 width=4) (actual time=0.168..3.120 rows=10 loops=1)"
"              Hash Cond: (movie_cast.mov_id = movie.mov_id)"
"              ->  Seq Scan on movie_cast  (cost=0.00..184.00 rows=10000 width=8) (actual time=0.010..1.670 rows=10011 loops=1)"
"              ->  Hash  (cost=32.50..32.50 rows=1 width=4) (actual time=0.150..0.150 rows=1 loops=1)"
"                    Buckets: 1024  Batches: 1  Memory Usage: 9kB"
"                    ->  Seq Scan on movie  (cost=0.00..32.50 rows=1 width=4) (actual time=0.010..0.147 rows=1 loops=1)"
"                          Filter: (mov_title = 'movie1'::bpchar)"
"                          Rows Removed by Filter: 999"
"Planning time: 0.152 ms"
"Execution time: 6.243 ms"
```

Scenario #2 : with B_tree on column (movie_title) on table (movie)
All flags are on

```
"Hash Semi Join  (cost=218.79..483.40 rows=10 width=48) (actual time=2.670..5.676 rows=10
loops=1)"
"  Hash Cond: (actor.act_id = movie_cast.act_id)"
"  ->  Seq Scan on actor  (cost=0.00..233.00 rows=12000 width=48) (actual time=0.025..1.583
rows=12000 loops=1)"
"  ->  Hash  (cost=218.67..218.67 rows=10 width=4) (actual time=2.628..2.628 rows=10
loops=1)"
"        Buckets: 1024  Batches: 1  Memory Usage: 9kB"
"        ->  Hash Semi Join  (cost=8.30..218.67 rows=10 width=4) (actual time=0.075..2.618
rows=10 loops=1)"
"              Hash Cond: (movie_cast.mov_id = movie.mov_id)"
"              ->  Seq Scan on movie_cast  (cost=0.00..184.00 rows=10000 width=8) (actual
time=0.017..1.306 rows=10011 loops=1)"
"              ->  Hash  (cost=8.29..8.29 rows=1 width=4) (actual time=0.043..0.043 rows=1
loops=1)"
"                    Buckets: 1024  Batches: 1  Memory Usage: 9kB"
"                    ->  Index Scan using btreeonmov_title on movie  (cost=0.28..8.29 rows=1 width=4)
(actual time=0.037..0.038 rows=1 loops=1)"
"                          Index Cond: (mov_title = 'movie1'::bpchar)"
"Planning time: 0.282 ms"
"Execution time: 5.771 ms"
```

Scenario #3 : with B_tree on column (act_id) on table (actor)
All flags are on

```
"Nested Loop  (cost=243.18..246.72 rows=10 width=48) (actual time=3.263..3.289 rows=10
loops=1)"
"  ->  HashAggregate  (cost=242.90..243.00 rows=10 width=4) (actual time=3.239..3.242
rows=10 loops=1)"
"        Group Key: movie_cast.act_id"
"        ->  Hash Semi Join  (cost=32.51..242.87 rows=10 width=4) (actual time=0.366..3.228
rows=10 loops=1)"
"              Hash Cond: (movie_cast.mov_id = movie.mov_id)"
"              ->  Seq Scan on movie_cast  (cost=0.00..184.00 rows=10000 width=8) (actual
time=0.018..1.257 rows=10011 loops=1)"
"              ->  Hash  (cost=32.50..32.50 rows=1 width=4) (actual time=0.340..0.340 rows=1
loops=1)"
"                    Buckets: 1024  Batches: 1  Memory Usage: 9kB"
"                  ->  Seq Scan on movie  (cost=0.00..32.50 rows=1 width=4) (actual
time=0.013..0.335 rows=1 loops=1)"
"                        Filter: (mov_title = 'movie1'::bpchar)"
"                        Rows Removed by Filter: 999"
"  ->  Index Scan using btreeindex on actor  (cost=0.29..0.36 rows=1 width=48) (actual
time=0.003..0.003 rows=1 loops=10)"
"        Index Cond: (act_id = movie_cast.act_id)"
"Planning time: 0.369 ms"
"Execution time: 3.345 ms"
```

Scenario #4 : with hash index on column (act_id) on table (actor)
All flags are on

```
"Nested Loop  (cost=242.90..243.99 rows=10 width=48) (actual time=4.997..5.033 rows=10
loops=1)"
"  ->  HashAggregate  (cost=242.90..243.00 rows=10 width=4) (actual time=4.978..4.981
rows=10 loops=1)"
"        Group Key: movie_cast.act_id"
"        ->  Hash Semi Join  (cost=32.51..242.87 rows=10 width=4) (actual time=0.764..4.964
rows=10 loops=1)"
"              Hash Cond: (movie_cast.mov_id = movie.mov_id)"
"              ->  Seq Scan on movie_cast  (cost=0.00..184.00 rows=10000 width=8) (actual
time=0.045..1.900 rows=10011 loops=1)"
"              ->  Hash  (cost=32.50..32.50 rows=1 width=4) (actual time=0.703..0.703 rows=1
loops=1)"
"                    Buckets: 1024  Batches: 1  Memory Usage: 9kB"
"                    ->  Seq Scan on movie  (cost=0.00..32.50 rows=1 width=4) (actual
time=0.027..0.697 rows=1 loops=1)"
"                          Filter: (mov_title = 'movie1'::bpchar)"
"                          Rows Removed by Filter: 999"
"  ->  Index Scan using hash on actor  (cost=0.00..0.09 rows=1 width=48) (actual
time=0.003..0.004 rows=1 loops=10)"
"        Index Cond: (act_id = movie_cast.act_id)"
"Planning time: 0.556 ms"
"Execution time: 5.130 ms"
```

Scenario #5 with hash index on column (mov_id) on table (movie_cast)
All flags are on

```
"Hash Semi Join  (cost=66.59..331.20 rows=10 width=48) (actual time=1.197..8.344 rows=10
loops=1)"
```

```
"  Hash Cond: (actor.act_id = movie_cast.act_id)"
"  ->  Seq Scan on actor  (cost=0.00..233.00 rows=12000 width=48) (actual time=0.047..3.217
rows=12000 loops=1)"
"  ->  Hash  (cost=66.46..66.46 rows=10 width=4) (actual time=1.127..1.127 rows=10 loops=1)"
"        Buckets: 1024  Batches: 1  Memory Usage: 9kB"
"        ->  Nested Loop  (cost=36.58..66.46 rows=10 width=4) (actual time=1.071..1.096 rows=10
loops=1)"
"              ->  HashAggregate  (cost=32.50..32.51 rows=1 width=4) (actual time=0.960..0.962
rows=1 loops=1)"
"                    Group Key: movie.mov_id"
"                    ->  Seq Scan on movie  (cost=0.00..32.50 rows=1 width=4) (actual
time=0.037..0.946 rows=1 loops=1)"
"                          Filter: (mov_title = 'movie1'::bpchar)"
"                          Rows Removed by Filter: 999"
"              ->  Bitmap Heap Scan on movie_cast  (cost=4.08..33.85 rows=10 width=8) (actual
time=0.044..0.059 rows=10 loops=1)"
"                    Recheck Cond: (mov_id = movie.mov_id)"
"                    Heap Blocks: exact=1"
"                    ->  Bitmap Index Scan on hash  (cost=0.00..4.08 rows=10 width=0) (actual
time=0.025..0.025 rows=10 loops=1)"
"                          Index Cond: (mov_id = movie.mov_id)"
"Planning time: 0.815 ms"
"Execution time: 8.522 ms"
```

Scenario #6 :with bitmap index on column (movie_title) on table (movie)
All flags are on

```
"Hash Semi Join  (cost=243.14..507.75 rows=10 width=48) (actual time=5.648..9.049 rows=10
loops=1)"
"  Hash Cond: (actor.act_id = movie_cast.act_id)"
"  ->  Seq Scan on actor  (cost=0.00..233.00 rows=12000 width=48) (actual time=0.041..1.692
rows=12000 loops=1)"
"  ->  Hash  (cost=243.01..243.01 rows=10 width=4) (actual time=5.574..5.574 rows=10
loops=1)"
"        Buckets: 1024  Batches: 1  Memory Usage: 9kB"
"        ->  Hash Semi Join  (cost=32.51..243.01 rows=10 width=4) (actual time=0.751..5.548
rows=10 loops=1)"
"              Hash Cond: (movie_cast.mov_id = movie.mov_id)"
"              ->  Seq Scan on movie_cast  (cost=0.00..184.11 rows=10011 width=8) (actual
time=0.031..2.126 rows=10011 loops=1)"
```

```
"            ->  Hash  (cost=32.50..32.50 rows=1 width=4) (actual time=0.694..0.694 rows=1 loops=1)"
"                  Buckets: 1024  Batches: 1  Memory Usage: 9kB"
"                  ->  Seq Scan on movie  (cost=0.00..32.50 rows=1 width=4) (actual time=0.027..0.685 rows=1 loops=1)"
"                        Filter: (mov_title = 'movie1'::bpchar)"
"                        Rows Removed by Filter: 999"
"Planning time: 0.385 ms"
"Execution time: 9.182 ms"
```

Scenario #7 :with bitmap index on column (act_id) on table (actor)
All flags are on

```
"Nested Loop  (cost=243.08..283.77 rows=10 width=48) (actual time=6.039..6.622 rows=10 loops=1)"
"  ->  HashAggregate  (cost=243.04..243.14 rows=10 width=4) (actual time=5.965..5.979 rows=10 loops=1)"
"        Group Key: movie_cast.act_id"
"        ->  Hash Semi Join  (cost=32.51..243.01 rows=10 width=4) (actual time=0.702..5.949 rows=10 loops=1)"
"              Hash Cond: (movie_cast.mov_id = movie.mov_id)"
"              ->  Seq Scan on movie_cast  (cost=0.00..184.11 rows=10011 width=8) (actual time=0.055..2.290 rows=10011 loops=1)"
"              ->  Hash  (cost=32.50..32.50 rows=1 width=4) (actual time=0.629..0.629 rows=1 loops=1)"
"                    Buckets: 1024  Batches: 1  Memory Usage: 9kB"
"                    ->  Seq Scan on movie  (cost=0.00..32.50 rows=1 width=4) (actual time=0.029..0.613 rows=1 loops=1)"
"                          Filter: (mov_title = 'movie1'::bpchar)"
"                          Rows Removed by Filter: 999"
"  ->  Bitmap Heap Scan on actor  (cost=0.04..4.05 rows=1 width=48) (actual time=0.017..0.017 rows=1 loops=10)"
"        Recheck Cond: (act_id = movie_cast.act_id)"
"        Heap Blocks: exact=10"
"        ->  Bitmap Index Scan on bitmapindex  (cost=0.00..0.04 rows=1 width=0) (actual time=0.011..0.011 rows=1 loops=10)"
"              Index Cond: (act_id = movie_cast.act_id)"
"Planning time: 0.652 ms"
"Execution time: 6.802 ms"
```

Scenario #8 :with hash index on table (movie) on column (mov_title) , hash index on table (movie_cast) on column (mov_id) and hash index on table (actor) on column (act_id)

All flags are on

```
"Nested Loop  (cost=42.01..43.10 rows=10 width=48) (actual time=0.115..0.191 rows=10
loops=1)"
"  ->  HashAggregate  (cost=42.01..42.11 rows=10 width=4) (actual time=0.103..0.110 rows=10
loops=1)"
"        Group Key: movie_cast.act_id"
"        ->  Nested Loop  (cost=12.10..41.98 rows=10 width=4) (actual time=0.082..0.093 rows=10
loops=1)"
"              ->  HashAggregate  (cost=8.02..8.03 rows=1 width=4) (actual time=0.033..0.034
rows=1 loops=1)"
"                    Group Key: movie.mov_id"
"                    ->  Index Scan using hash1 on movie  (cost=0.00..8.02 rows=1 width=4) (actual
time=0.022..0.025 rows=1 loops=1)"
"                          Index Cond: (mov_title = 'movie1'::bpchar)"
"              ->  Bitmap Heap Scan on movie_cast  (cost=4.08..33.85 rows=10 width=8) (actual
time=0.018..0.025 rows=10 loops=1)"
"                    Recheck Cond: (mov_id = movie.mov_id)"
"                    Heap Blocks: exact=1"
"                    ->  Bitmap Index Scan on hash2  (cost=0.00..4.08 rows=10 width=0) (actual
time=0.009..0.009 rows=10 loops=1)"
"                          Index Cond: (mov_id = movie.mov_id)"
"  ->  Index Scan using hash3 on actor  (cost=0.00..0.09 rows=1 width=48) (actual
time=0.005..0.006 rows=1 loops=10)"
"        Index Cond: (act_id = movie_cast.act_id)"
"Planning time: 0.795 ms"
"Execution time: 0.370 ms"
```

Conclusion of query 10:
The best performance was for Scenario #8 , because in this query we search for actors who acted in a specific movie , so the number of tuples wanted from table movie and table movie_cast is only one tuple , so hash index on these 2 columns would get this tuple in O(1) complexity , also we want some portion of table actors which contains many tuples that are likely not to act on one movie so we do want all of them so making a hash index on column act_id on table actor enhances the complexity , so all the sub queries is relatively exact queries so hash index was the best to get the tuples with exact value in O(1).

Query 11 :

Scenario #1 : without an index
All flags are on

```
"Hash Semi Join  (cost=471.62..515.99 rows=10 width=42) (actual time=0.432..0.432 rows=0
loops=1)"
"  Hash Cond: (director.dir_id = movie_direction.dir_id)"
"  ->  Seq Scan on director  (cost=0.00..39.00 rows=2000 width=46) (actual time=0.035..0.035
rows=1 loops=1)"
"  ->  Hash  (cost=471.50..471.50 rows=10 width=4) (actual time=0.386..0.386 rows=0 loops=1)"
"        Buckets: 1024  Batches: 1  Memory Usage: 8kB"
"        ->  Hash Semi Join  (cost=453.76..471.50 rows=10 width=4) (actual time=0.386..0.386
rows=0 loops=1)"
"              Hash Cond: (movie_direction.mov_id = movie_cast.mov_id)"
"              ->  Seq Scan on movie_direction  (cost=0.00..15.00 rows=1000 width=8) (actual
time=0.029..0.029 rows=1 loops=1)"
"              ->  Hash  (cost=453.64..453.64 rows=10 width=4) (actual time=0.349..0.349 rows=0
loops=1)"
"                    Buckets: 1024  Batches: 1  Memory Usage: 8kB"
"                    ->  Hash Semi Join  (cost=243.14..453.64 rows=10 width=4) (actual
time=0.349..0.349 rows=0 loops=1)"
"                          Hash Cond: (movie_cast.role = movie_cast_1.role)"
"                          ->  Seq Scan on movie_cast  (cost=0.00..184.11 rows=10011 width=35)
(actual time=0.029..0.029 rows=1 loops=1)"
"                          ->  Hash  (cost=243.01..243.01 rows=10 width=31) (actual time=0.310..0.310
rows=0 loops=1)"
"                                Buckets: 1024  Batches: 1  Memory Usage: 8kB"
"                                ->  Hash Semi Join  (cost=32.51..243.01 rows=10 width=31) (actual
time=0.310..0.310 rows=0 loops=1)"
"                                      Hash Cond: (movie_cast_1.mov_id = movie.mov_id)"
"                                      ->  Seq Scan on movie_cast movie_cast_1  (cost=0.00..184.11
rows=10011 width=35) (actual time=0.026..0.026 rows=1 loops=1)"
"                                      ->  Hash  (cost=32.50..32.50 rows=1 width=4) (actual
time=0.276..0.276 rows=0 loops=1)"
```

```
"                                Buckets: 1024  Batches: 1  Memory Usage: 8kB"
"                                ->  Seq Scan on movie  (cost=0.00..32.50 rows=1 width=4) (actual
time=0.275..0.275 rows=0 loops=1)"
"                                    Filter: (mov_title = 'movie"
"2'::bpchar)"
"                                    Rows Removed by Filter: 1000"
"Planning time: 0.718 ms"
"Execution time: 0.506 ms"
```

Scenario #2 : with Btree index on column role on movie_cast table
All flags are on

```
"Hash Semi Join  (cost=264.84..309.20 rows=10 width=42) (actual time=0.400..0.400 rows=0
loops=1)"
"  Hash Cond: (director.dir_id = movie_direction.dir_id)"
"  ->  Seq Scan on director  (cost=0.00..39.00 rows=2000 width=46) (actual time=0.022..0.022
rows=1 loops=1)"
"  ->  Hash  (cost=264.71..264.71 rows=10 width=4) (actual time=0.366..0.366 rows=0 loops=1)"
"        Buckets: 1024  Batches: 1  Memory Usage: 8kB"
"        ->  Hash Semi Join  (cost=246.97..264.71 rows=10 width=4) (actual time=0.365..0.366
rows=0 loops=1)"
"              Hash Cond: (movie_direction.mov_id = movie_cast.mov_id)"
"              ->  Seq Scan on movie_direction  (cost=0.00..15.00 rows=1000 width=8) (actual
time=0.015..0.015 rows=1 loops=1)"
"              ->  Hash  (cost=246.85..246.85 rows=10 width=4) (actual time=0.337..0.337 rows=0
loops=1)"
"                    Buckets: 1024  Batches: 1  Memory Usage: 8kB"
"                    ->  Nested Loop  (cost=243.32..246.85 rows=10 width=4) (actual
time=0.337..0.337 rows=0 loops=1)"
"                          ->  HashAggregate  (cost=243.04..243.14 rows=10 width=31) (actual
time=0.337..0.337 rows=0 loops=1)"
"                                Group Key: movie_cast_1.role"
"                                ->  Hash Semi Join  (cost=32.51..243.01 rows=10 width=31) (actual
time=0.336..0.336 rows=0 loops=1)"
"                                      Hash Cond: (movie_cast_1.mov_id = movie.mov_id)"
"                                      ->  Seq Scan on movie_cast movie_cast_1  (cost=0.00..184.11
rows=10011 width=35) (actual time=0.014..0.014 rows=1 loops=1)"
"                                      ->  Hash  (cost=32.50..32.50 rows=1 width=4) (actual
time=0.312..0.312 rows=0 loops=1)"
"                                            Buckets: 1024  Batches: 1  Memory Usage: 8kB"
"                                            ->  Seq Scan on movie  (cost=0.00..32.50 rows=1 width=4) (actual
time=0.312..0.312 rows=0 loops=1)"
"                                                  Filter: (mov_title = 'movie"
"2'::bpchar)"
```

```
"                                             Rows Removed by Filter: 1000"
"                       ->  Index Scan using btreeindex on movie_cast  (cost=0.29..0.36 rows=1
width=35) (never executed)"
"                             Index Cond: (role = movie_cast_1.role)"
"Planning time: 0.604 ms"
"Execution time: 0.511 ms"
```

Scenario #3 : with Btree index on column mov_id on movie_direction table
All flags are on

```
"Hash Semi Join  (cost=457.31..501.67 rows=10 width=42) (actual time=1.026..1.026 rows=0
loops=1)"
"  Hash Cond: (director.dir_id = movie_direction.dir_id)"
"  ->  Seq Scan on director  (cost=0.00..39.00 rows=2000 width=46) (actual time=0.256..0.256
rows=1 loops=1)"
"  ->  Hash  (cost=457.19..457.19 rows=10 width=4) (actual time=0.697..0.697 rows=0 loops=1)"
"        Buckets: 1024  Batches: 1  Memory Usage: 8kB"
"        ->  Nested Loop  (cost=453.94..457.19 rows=10 width=4) (actual time=0.696..0.696
rows=0 loops=1)"
"              ->  HashAggregate  (cost=453.66..453.76 rows=10 width=4) (actual time=0.693..0.693
rows=0 loops=1)"
"                    Group Key: movie_cast.mov_id"
"                    ->  Hash Semi Join  (cost=243.14..453.64 rows=10 width=4) (actual
time=0.688..0.688 rows=0 loops=1)"
"                          Hash Cond: (movie_cast.role = movie_cast_1.role)"
"                          ->  Seq Scan on movie_cast  (cost=0.00..184.11 rows=10011 width=35)
(actual time=0.079..0.079 rows=1 loops=1)"
"                          ->  Hash  (cost=243.01..243.01 rows=10 width=31) (actual time=0.571..0.571
rows=0 loops=1)"
"                                Buckets: 1024  Batches: 1  Memory Usage: 8kB"
"                                ->  Hash Semi Join  (cost=32.51..243.01 rows=10 width=31) (actual
time=0.567..0.567 rows=0 loops=1)"
"                                      Hash Cond: (movie_cast_1.mov_id = movie.mov_id)"
"                                      ->  Seq Scan on movie_cast movie_cast_1  (cost=0.00..184.11
rows=10011 width=35) (actual time=0.052..0.052 rows=1 loops=1)"
"                                      ->  Hash  (cost=32.50..32.50 rows=1 width=4) (actual
time=0.368..0.368 rows=0 loops=1)"
"                                            Buckets: 1024  Batches: 1  Memory Usage: 8kB"
"                                            ->  Seq Scan on movie  (cost=0.00..32.50 rows=1 width=4) (actual
time=0.367..0.367 rows=0 loops=1)"
"                                                  Filter: (mov_title = 'movie"
"2'::bpchar)"
```

```
"                                    Rows Removed by Filter: 1000"
"          ->  Index Scan using btreeindex on movie_direction  (cost=0.28..0.33 rows=1 width=8)
(never executed)"
"                Index Cond: (mov_id = movie_cast.mov_id)"
"Planning time: 3.305 ms"
"Execution time: 1.434 ms"
```

Scenario #4 : with hash index on column mov_id on movie_cast table
All flags are on

```
"Hash Semi Join  (cost=295.08..339.44 rows=10 width=42) (actual time=1.066..1.066 rows=0
loops=1)"
"  Hash Cond: (director.dir_id = movie_direction.dir_id)"
"  ->  Seq Scan on director  (cost=0.00..39.00 rows=2000 width=46) (actual time=0.061..0.062
rows=1 loops=1)"
"  ->  Hash  (cost=294.95..294.95 rows=10 width=4) (actual time=0.948..0.949 rows=0 loops=1)"
"        Buckets: 1024  Batches: 1  Memory Usage: 8kB"
"        ->  Hash Semi Join  (cost=277.21..294.95 rows=10 width=4) (actual time=0.948..0.948
rows=0 loops=1)"
"              Hash Cond: (movie_direction.mov_id = movie_cast.mov_id)"
"              ->  Seq Scan on movie_direction  (cost=0.00..15.00 rows=1000 width=8) (actual
time=0.061..0.062 rows=1 loops=1)"
"              ->  Hash  (cost=277.09..277.09 rows=10 width=4) (actual time=0.819..0.819 rows=0
loops=1)"
"                    Buckets: 1024  Batches: 1  Memory Usage: 8kB"
"                    ->  Hash Semi Join  (cost=66.59..277.09 rows=10 width=4) (actual
time=0.817..0.817 rows=0 loops=1)"
"                          Hash Cond: (movie_cast.role = movie_cast_1.role)"
"                          ->  Seq Scan on movie_cast  (cost=0.00..184.11 rows=10011 width=35)
(actual time=0.052..0.052 rows=1 loops=1)"
"                          ->  Hash  (cost=66.46..66.46 rows=10 width=31) (actual time=0.719..0.719
rows=0 loops=1)"
"                                Buckets: 1024  Batches: 1  Memory Usage: 8kB"
"                                ->  Nested Loop  (cost=36.58..66.46 rows=10 width=31) (actual
time=0.718..0.718 rows=0 loops=1)"
"                                      ->  HashAggregate  (cost=32.50..32.51 rows=1 width=4) (actual
time=0.717..0.717 rows=0 loops=1)"
"                                            Group Key: movie.mov_id"
"                                            ->  Seq Scan on movie  (cost=0.00..32.50 rows=1 width=4) (actual
time=0.711..0.711 rows=0 loops=1)"
"                                                  Filter: (mov_title = 'movie"
"2'::bpchar)"
```

```
"                                        Rows Removed by Filter: 1000"
"                              ->  Bitmap Heap Scan on movie_cast movie_cast_1  (cost=4.08..33.85
rows=10 width=35) (never executed)"
"                                   Recheck Cond: (mov_id = movie.mov_id)"
"                                   ->  Bitmap Index Scan on hash  (cost=0.00..4.08 rows=10 width=0)
(never executed)"
"                                        Index Cond: (mov_id = movie.mov_id)"
"Planning time: 1.035 ms"
"Execution time: 1.441 ms"
```

Scenario #5 : with hash index on column role on movie_cast table
All flags are on

```
"Hash Semi Join  (cost=262.00..306.36 rows=10 width=42) (actual time=0.254..0.254 rows=0
loops=1)"
"  Hash Cond: (director.dir_id = movie_direction.dir_id)"
"  ->  Seq Scan on director  (cost=0.00..39.00 rows=2000 width=46) (actual time=0.022..0.022
rows=1 loops=1)"
"  ->  Hash  (cost=261.87..261.87 rows=10 width=4) (actual time=0.221..0.221 rows=0 loops=1)"
"        Buckets: 1024  Batches: 1  Memory Usage: 8kB"
"        ->  Hash Semi Join  (cost=244.14..261.87 rows=10 width=4) (actual time=0.221..0.221
rows=0 loops=1)"
"              Hash Cond: (movie_direction.mov_id = movie_cast.mov_id)"
"              ->  Seq Scan on movie_direction  (cost=0.00..15.00 rows=1000 width=8) (actual
time=0.014..0.014 rows=1 loops=1)"
"              ->  Hash  (cost=244.01..244.01 rows=10 width=4) (actual time=0.194..0.194 rows=0
loops=1)"
"                    Buckets: 1024  Batches: 1  Memory Usage: 8kB"
"                    ->  Nested Loop  (cost=243.04..244.01 rows=10 width=4) (actual
time=0.194..0.194 rows=0 loops=1)"
"                          ->  HashAggregate  (cost=243.04..243.14 rows=10 width=31) (actual
time=0.194..0.194 rows=0 loops=1)"
"                                Group Key: movie_cast_1.role"
"                                ->  Hash Semi Join  (cost=32.51..243.01 rows=10 width=31) (actual
time=0.193..0.193 rows=0 loops=1)"
"                                      Hash Cond: (movie_cast_1.mov_id = movie.mov_id)"
"                                      ->  Seq Scan on movie_cast movie_cast_1  (cost=0.00..184.11
rows=10011 width=35) (actual time=0.012..0.012 rows=1 loops=1)"
```

```
"                                      ->  Hash  (cost=32.50..32.50 rows=1 width=4) (actual
time=0.172..0.172 rows=0 loops=1)"
"                                            Buckets: 1024  Batches: 1  Memory Usage: 8kB"
"                                            ->  Seq Scan on movie  (cost=0.00..32.50 rows=1 width=4) (actual
time=0.171..0.172 rows=0 loops=1)"
"                                                  Filter: (mov_title = 'movie"
"2'::bpchar)"
"                                                  Rows Removed by Filter: 1000"
"                          ->  Index Scan using hash on movie_cast  (cost=0.00..0.08 rows=1 width=35)
(never executed)"
"                                Index Cond: (role = movie_cast_1.role)"
"Planning time: 0.419 ms"
"Execution time: 0.327 ms"
```

Scenario #6 : with hash index on column role on movie_cast table
All flags are on

```
"Hash Semi Join  (cost=471.62..515.99 rows=10 width=42) (actual time=0.832..0.832 rows=0
loops=1)"
"  Hash Cond: (director.dir_id = movie_direction.dir_id)"
"  ->  Seq Scan on director  (cost=0.00..39.00 rows=2000 width=46) (actual time=0.039..0.039
rows=1 loops=1)"
"  ->  Hash  (cost=471.50..471.50 rows=10 width=4) (actual time=0.781..0.781 rows=0 loops=1)"
"        Buckets: 1024  Batches: 1  Memory Usage: 8kB"
"        ->  Hash Semi Join  (cost=453.76..471.50 rows=10 width=4) (actual time=0.780..0.781
rows=0 loops=1)"
"              Hash Cond: (movie_direction.mov_id = movie_cast.mov_id)"
"              ->  Seq Scan on movie_direction  (cost=0.00..15.00 rows=1000 width=8) (actual
time=0.030..0.030 rows=1 loops=1)"
"              ->  Hash  (cost=453.64..453.64 rows=10 width=4) (actual time=0.743..0.743 rows=0
loops=1)"
"                    Buckets: 1024  Batches: 1  Memory Usage: 8kB"
"                    ->  Hash Semi Join  (cost=243.14..453.64 rows=10 width=4) (actual
time=0.742..0.742 rows=0 loops=1)"
"                          Hash Cond: (movie_cast.role = movie_cast_1.role)"
"                          ->  Seq Scan on movie_cast  (cost=0.00..184.11 rows=10011 width=35)
(actual time=0.026..0.026 rows=1 loops=1)"
"                          ->  Hash  (cost=243.01..243.01 rows=10 width=31) (actual time=0.709..0.709
rows=0 loops=1)"
"                                Buckets: 1024  Batches: 1  Memory Usage: 8kB"
"                                ->  Hash Semi Join  (cost=32.51..243.01 rows=10 width=31) (actual
time=0.709..0.709 rows=0 loops=1)"
"                                      Hash Cond: (movie_cast_1.mov_id = movie.mov_id)"
"                          ->  Seq Scan on movie_cast movie_cast_1  (cost=0.00..184.11
rows=10011 width=35) (actual time=0.022..0.022 rows=1 loops=1)"
"                          ->  Hash  (cost=32.50..32.50 rows=1 width=4) (actual
time=0.680..0.680 rows=0 loops=1)"
"                              Buckets: 1024  Batches: 1  Memory Usage: 8kB"
"                              ->  Seq Scan on movie  (cost=0.00..32.50 rows=1 width=4) (actual
time=0.679..0.679 rows=0 loops=1)"
"                                  Filter: (mov_title = 'movie"
"2'::bpchar)"
"                                  Rows Removed by Filter: 1000"
"Planning time: 0.995 ms"
"Execution time: 0.926 ms"
```

Scenario #7 :with hash index on table (movie_cast) on column (mov_id) , hash index on table (movie_cast) on column (role)
All flags are on

```
"Hash Semi Join  (cost=85.45..129.81 rows=10 width=42) (actual time=0.450..0.450 rows=0
loops=1)"
"  Hash Cond: (director.dir_id = movie_direction.dir_id)"
"  ->  Seq Scan on director  (cost=0.00..39.00 rows=2000 width=46) (actual time=0.024..0.024
rows=1 loops=1)"
"  ->  Hash  (cost=85.32..85.32 rows=10 width=4) (actual time=0.409..0.409 rows=0 loops=1)"
"      Buckets: 1024  Batches: 1  Memory Usage: 8kB"
"      ->  Hash Semi Join  (cost=67.59..85.32 rows=10 width=4) (actual time=0.409..0.409
rows=0 loops=1)"
"          Hash Cond: (movie_direction.mov_id = movie_cast.mov_id)"
"          ->  Seq Scan on movie_direction  (cost=0.00..15.00 rows=1000 width=8) (actual
time=0.021..0.022 rows=1 loops=1)"
"          ->  Hash  (cost=67.46..67.46 rows=10 width=4) (actual time=0.373..0.373 rows=0
loops=1)"
"              Buckets: 1024  Batches: 1  Memory Usage: 8kB"
"              ->  Nested Loop  (cost=66.49..67.46 rows=10 width=4) (actual time=0.373..0.373
rows=0 loops=1)"
"                  ->  HashAggregate  (cost=66.49..66.59 rows=10 width=31) (actual
time=0.373..0.373 rows=0 loops=1)"
"                      Group Key: movie_cast_1.role"
"                      ->  Nested Loop  (cost=36.58..66.46 rows=10 width=31) (actual
time=0.372..0.372 rows=0 loops=1)"
"                          ->  HashAggregate  (cost=32.50..32.51 rows=1 width=4) (actual
time=0.371..0.371 rows=0 loops=1)"
"                              Group Key: movie.mov_id"
"                                            ->  Seq Scan on movie  (cost=0.00..32.50 rows=1 width=4) (actual
time=0.370..0.370 rows=0 loops=1)"
"                                                  Filter: (mov_title = 'movie"
"2'::bpchar)"
"                                                  Rows Removed by Filter: 1000"
"                                      ->  Bitmap Heap Scan on movie_cast movie_cast_1  (cost=4.08..33.85
rows=10 width=35) (never executed)"
"                                            Recheck Cond: (mov_id = movie.mov_id)"
"                                            ->  Bitmap Index Scan on hash2  (cost=0.00..4.08 rows=10
width=0) (never executed)"
"                                                  Index Cond: (mov_id = movie.mov_id)"
"                          ->  Index Scan using hash1 on movie_cast  (cost=0.00..0.08 rows=1 width=35)
(never executed)"
"                                Index Cond: (role = movie_cast_1.role)"
"Planning time: 0.788 ms"
"Execution time: 0.626 ms"
```

Scenario #8 :with hash index on table (movie_cast) on column (mov_id) , hash index on table (movie_cast) on column (role) , with hash index on table (director) on column (dir_id) , hash index on table (movie_direction) on column (movie_id) , hash index on table (movie) on column (movie_title)
All flags are on

```
"Nested Loop  (cost=43.85..48.86 rows=10 width=42) (actual time=0.013..0.013 rows=0
loops=1)"
"  ->  HashAggregate  (cost=43.85..43.95 rows=10 width=4) (actual time=0.013..0.013 rows=0
loops=1)"
"        Group Key: movie_direction.dir_id"
"        ->  Nested Loop  (cost=43.01..43.82 rows=10 width=4) (actual time=0.012..0.012 rows=0
loops=1)"
"              ->  HashAggregate  (cost=43.01..43.11 rows=10 width=4) (actual time=0.012..0.012
rows=0 loops=1)"
"                    Group Key: movie_cast.mov_id"
"                    ->  Nested Loop  (cost=42.01..42.98 rows=10 width=4) (actual time=0.011..0.011
rows=0 loops=1)"
"                          ->  HashAggregate  (cost=42.01..42.11 rows=10 width=31) (actual
time=0.011..0.011 rows=0 loops=1)"
"                                Group Key: movie_cast_1.role"
"                                ->  Nested Loop  (cost=12.10..41.98 rows=10 width=31) (actual
time=0.011..0.011 rows=0 loops=1)"
"                                      ->  HashAggregate  (cost=8.02..8.03 rows=1 width=4) (actual
time=0.010..0.010 rows=0 loops=1)"
"                                            Group Key: movie.mov_id"
"                                    ->  Index Scan using hash5 on movie  (cost=0.00..8.02 rows=1
width=4) (actual time=0.009..0.009 rows=0 loops=1)"
"                                          Index Cond: (mov_title = 'movie"
"2'::bpchar)"
"                              ->  Bitmap Heap Scan on movie_cast movie_cast_1  (cost=4.08..33.85
rows=10 width=35) (never executed)"
"                                    Recheck Cond: (mov_id = movie.mov_id)"
"                                    ->  Bitmap Index Scan on hash2  (cost=0.00..4.08 rows=10
width=0) (never executed)"
"                                          Index Cond: (mov_id = movie.mov_id)"
"                    ->  Index Scan using hash1 on movie_cast  (cost=0.00..0.08 rows=1 width=35)
(never executed)"
"                          Index Cond: (role = movie_cast_1.role)"
"          ->  Index Scan using hash4 on movie_direction  (cost=0.00..0.06 rows=1 width=8)
(never executed)"
"                Index Cond: (mov_id = movie_cast.mov_id)"
"  ->  Index Scan using hash3 on director  (cost=0.00..0.48 rows=1 width=46) (never executed)"
"        Index Cond: (dir_id = movie_direction.dir_id)"
"Planning time: 0.740 ms"
"Execution time: 0.121 ms"
```

Conclusion of query 11:
The best performance was at Scenario #8 , because in this query we search for a specific tuple in table movie then collecting all its associated tuples in table movie cast that has same role , then we select all movies that has some roles specifically , then names directors associated with specific tuples in table movie_direction , so all the sub queries is relatively exact queries so hash index was the best to get the tuples with exact value in O(1).

Query 12 :

Scenario #1 : without an index
All flags are on

```
"Seq Scan on movie  (cost=66.50..99.00 rows=1 width=51) (actual time=1.376..1.768 rows=1
loops=1)"
"  Filter: (mov_id = $1)"
"  Rows Removed by Filter: 999"
"  InitPlan 2 (returns $1)"
"    ->  Seq Scan on movie_direction  (cost=49.00..66.50 rows=4 width=4) (actual
time=0.946..1.264 rows=1 loops=1)"
"          Filter: (dir_id = $0)"
"          Rows Removed by Filter: 996"
"        InitPlan 1 (returns $0)"
"          ->  Seq Scan on director  (cost=0.00..49.00 rows=1 width=4) (actual time=0.020..0.918
rows=1 loops=1)"
"                Filter: ((dir_fname = 'actor1'::bpchar) AND (dir_lname = 'actor1'::bpchar))"
"                Rows Removed by Filter: 1999"
"Planning time: 0.117 ms"
"Execution time: 1.837 ms"
```

Scenario #2 : with Btree on column dir_id on table movie_direction
All flags are on

```
"Seq Scan on movie  (cost=57.34..89.84 rows=1 width=51) (actual time=0.523..0.732 rows=1
loops=1)"
"  Filter: (mov_id = $1)"
"  Rows Removed by Filter: 999"
"  InitPlan 2 (returns $1)"
"    ->  Index Scan using btreeindex on movie_direction  (cost=49.27..57.34 rows=4 width=4)
(actual time=0.505..0.506 rows=1 loops=1)"
"        Index Cond: (dir_id = $0)"
"        InitPlan 1 (returns $0)"
"          ->  Seq Scan on director  (cost=0.00..49.00 rows=1 width=4) (actual time=0.015..0.490
rows=1 loops=1)"
"                Filter: ((dir_fname = 'actor1'::bpchar) AND (dir_lname = 'actor1'::bpchar))"
"                Rows Removed by Filter: 1999"
"Planning time: 0.196 ms"
"Execution time: 0.759 ms"
```

Scenario #3 : with Btree mixed index on column dir_fname,dir_lname on table director
All flags are on

```
"Seq Scan on movie  (cost=25.76..58.26 rows=1 width=51) (actual time=0.146..0.222 rows=1
loops=1)"
"  Filter: (mov_id = $1)"
"  Rows Removed by Filter: 999"
"  InitPlan 2 (returns $1)"
"    ->  Seq Scan on movie_direction  (cost=8.30..25.76 rows=4 width=4) (actual
time=0.064..0.134 rows=1 loops=1)"
"        Filter: (dir_id = $0)"
"        Rows Removed by Filter: 996"
"        InitPlan 1 (returns $0)"
"          ->  Index Scan using btreeindex on director  (cost=0.28..8.30 rows=1 width=4) (actual
time=0.047..0.047 rows=1 loops=1)"
"                Index Cond: ((dir_fname = 'actor1'::bpchar) AND (dir_lname = 'actor1'::bpchar))"
```

```
"Planning time: 2.123 ms"
"Execution time: 0.246 ms"
```

Scenario #4 : with hash index on column mov_id on table movie , hash index on column dir_id on table movie_direction
All flags are on

```
"Index Scan using hash1 on movie  (cost=58.11..66.13 rows=1 width=51) (actual
time=0.209..0.210 rows=1 loops=1)"
"  Index Cond: (mov_id = $1)"
"  InitPlan 2 (returns $1)"
"    ->  Bitmap Heap Scan on movie_direction  (cost=53.03..58.11 rows=4 width=4) (actual
time=0.205..0.205 rows=1 loops=1)"
"          Recheck Cond: (dir_id = $0)"
"          Heap Blocks: exact=1"
"          InitPlan 1 (returns $0)"
"            ->  Seq Scan on director  (cost=0.00..49.00 rows=1 width=4) (actual time=0.013..0.197
rows=1 loops=1)"
"                  Filter: ((dir_fname = 'actor1'::bpchar) AND (dir_lname = 'actor1'::bpchar))"
"                  Rows Removed by Filter: 1999"
"          ->  Bitmap Index Scan on hash2  (cost=0.00..4.03 rows=4 width=0) (actual
time=0.202..0.202 rows=1 loops=1)"
"                Index Cond: (dir_id = $0)"
"Planning time: 0.108 ms"
"Execution time: 0.235 ms"
```

Scenario #5 : with bitmap mixed index on column dir_fname,dir_lname on table director
All flags are on

```
"Seq Scan on movie  (cost=41.48..73.98 rows=1 width=51) (actual time=0.136..0.229 rows=1
loops=1)"
"  Filter: (mov_id = $1)"
"  Rows Removed by Filter: 999"
"  InitPlan 2 (returns $1)"
"    ->  Seq Scan on movie_direction  (cost=24.02..41.48 rows=4 width=4) (actual
time=0.035..0.123 rows=1 loops=1)"
"          Filter: (dir_id = $0)"
"          Rows Removed by Filter: 996"
"        InitPlan 1 (returns $0)"
"          ->  Bitmap Heap Scan on director  (cost=20.00..24.02 rows=1 width=4) (actual
time=0.024..0.024 rows=1 loops=1)"
"                Recheck Cond: (((dir_fname)::text = 'actor1'::text) AND ((dir_lname)::text =
'actor1'::text))"
"                Heap Blocks: exact=1"
"                ->  Bitmap Index Scan on bitmapindex  (cost=0.00..20.00 rows=1 width=0) (actual
time=0.020..0.021 rows=1 loops=1)"
"                      Index Cond: (((dir_fname)::text = 'actor1'::text) AND ((dir_lname)::text =
'actor1'::text))"
"Planning time: 0.146 ms"
"Execution time: 0.265 ms"
```

Scenario #6 :with bitmap mixed index on column dir_fname,dir_lname on table director ,hash index on column mov_id on table movie , hash index on column dir_id on table movie_direction
All flags are on

```
"Index Scan using hash1 on movie  (cost=33.13..41.14 rows=1 width=51) (actual
time=0.047..0.048 rows=1 loops=1)"
"  Index Cond: (mov_id = $1)"
"  InitPlan 2 (returns $1)"
"    ->  Bitmap Heap Scan on movie_direction  (cost=28.05..33.13 rows=4 width=4) (actual
time=0.040..0.041 rows=1 loops=1)"
"        Recheck Cond: (dir_id = $0)"
"        Heap Blocks: exact=1"
"        InitPlan 1 (returns $0)"
"          ->  Bitmap Heap Scan on director  (cost=20.00..24.02 rows=1 width=4) (actual
time=0.031..0.032 rows=1 loops=1)"
"                Recheck Cond: (((dir_fname)::text = 'actor1'::text) AND ((dir_lname)::text =
'actor1'::text))"
"                Heap Blocks: exact=1"
"                ->  Bitmap Index Scan on bitmapindex  (cost=0.00..20.00 rows=1 width=0) (actual
time=0.027..0.027 rows=1 loops=1)"
"                      Index Cond: (((dir_fname)::text = 'actor1'::text) AND ((dir_lname)::text =
'actor1'::text))"
"        ->  Bitmap Index Scan on hash2  (cost=0.00..4.03 rows=4 width=0) (actual
time=0.038..0.038 rows=1 loops=1)"
"              Index Cond: (dir_id = $0)"
"Planning time: 0.227 ms"
"Execution time: 0.112 ms"
```

Scenario #7:with Btree mixed index on column dir_fname,dir_lname on table director ,hash index on column mov_id on table movie , hash index on column dir_id on table movie_direction All flags are on

```
"Index Scan using hash1 on movie  (cost=17.41..25.42 rows=1 width=51) (actual
time=0.048..0.049 rows=1 loops=1)"
"  Index Cond: (mov_id = $1)"
"  InitPlan 2 (returns $1)"
"    ->  Bitmap Heap Scan on movie_direction  (cost=12.33..17.41 rows=4 width=4) (actual
time=0.041..0.041 rows=1 loops=1)"
"          Recheck Cond: (dir_id = $0)"
"          Heap Blocks: exact=1"
"          InitPlan 1 (returns $0)"
"            ->  Index Scan using btreeindex on director  (cost=0.28..8.30 rows=1 width=4) (actual
time=0.029..0.030 rows=1 loops=1)"
"                  Index Cond: (((dir_fname)::text = 'actor1'::text) AND ((dir_lname)::text =
'actor1'::text))"
"          ->  Bitmap Index Scan on hash2  (cost=0.00..4.03 rows=4 width=0) (actual
time=0.037..0.037 rows=1 loops=1)"
"                Index Cond: (dir_id = $0)"
"Planning time: 0.244 ms"
"Execution time: 0.102 ms"
```

Conclusion of Query 12:
The best performance is for Scenario #7 and #6 , Scenario #6 is slightly better in time , while Scenario #7 cost in rows is better , in this query all sub queries are exact queries so using hash index for table movie and table movie_direction is the best choice , and for table director using mixed index is better since we need the exact tuple but with 2 parameters so mixed Btree or mixed bitmap was a good choics