Assignment 2

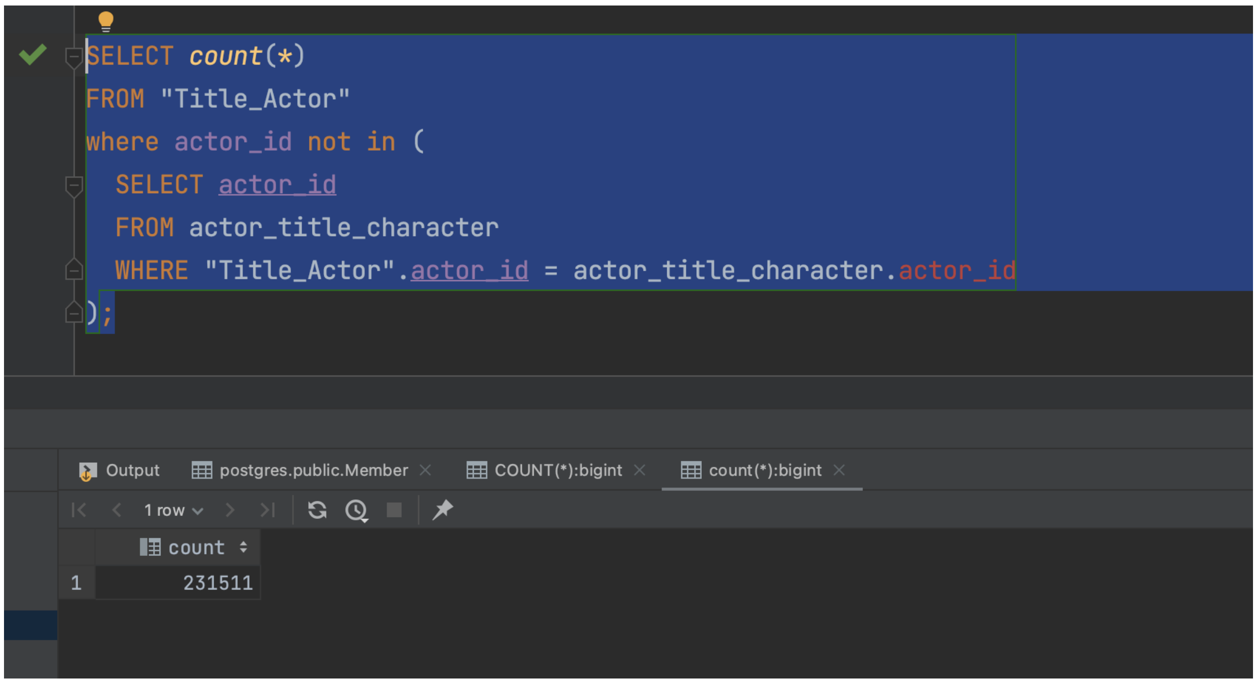
Anikhet Mulky
am9559@g.rit.edu

Q1)

Using datagrip, the imdb tsv files were dragged and dropped within 15 minutes.
The attached files contain the code in tables.sql .

Q2) All the code is attached in questions2.sql

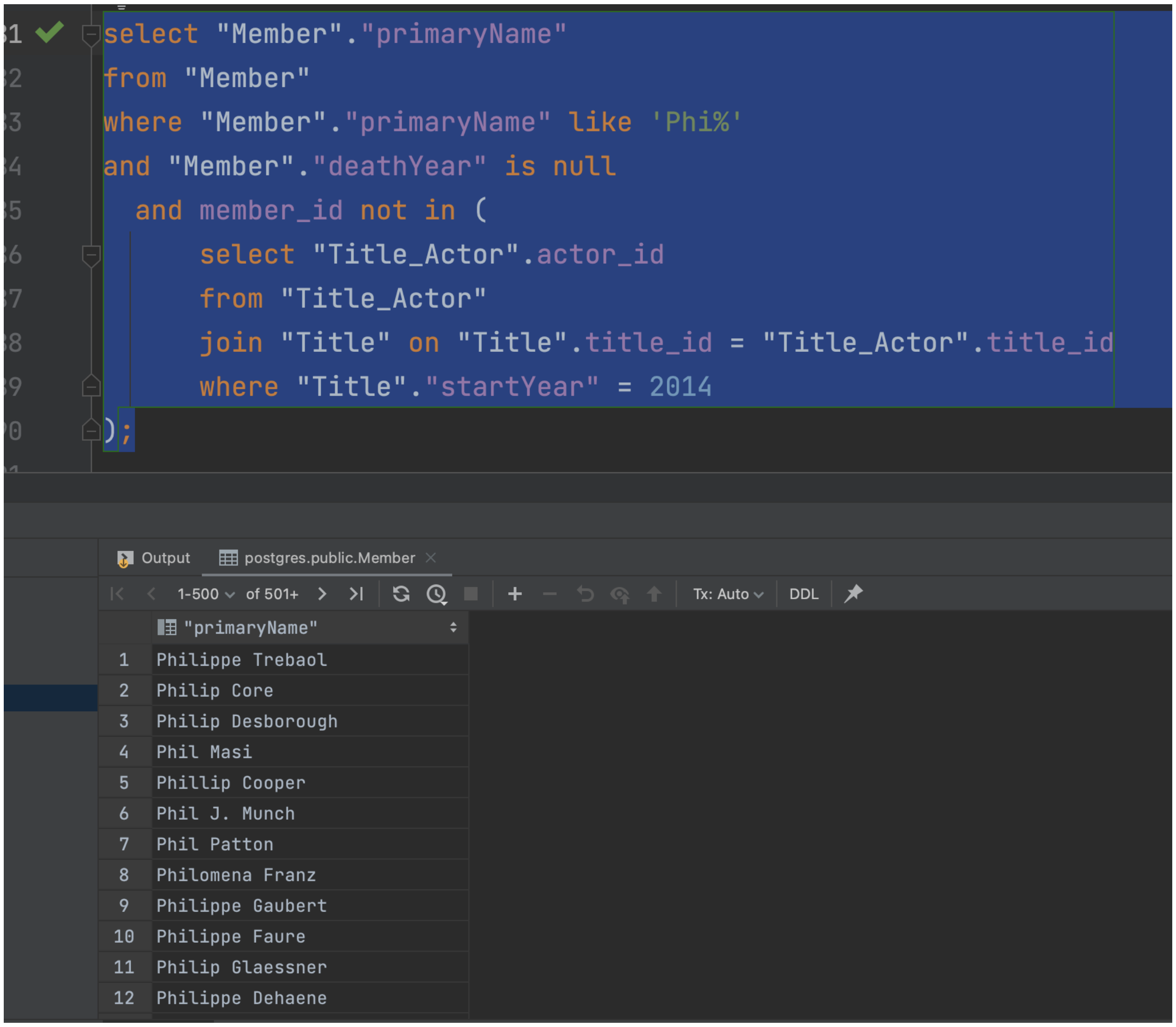
1) **3.7 seconds**



```sql
SELECT count(*)
FROM "Title_Actor"
where actor_id not in (
  SELECT actor_id
  FROM actor_title_character
  WHERE "Title_Actor".actor_id = actor_title_character.actor_id
);
```

```
   count
1  231511
```

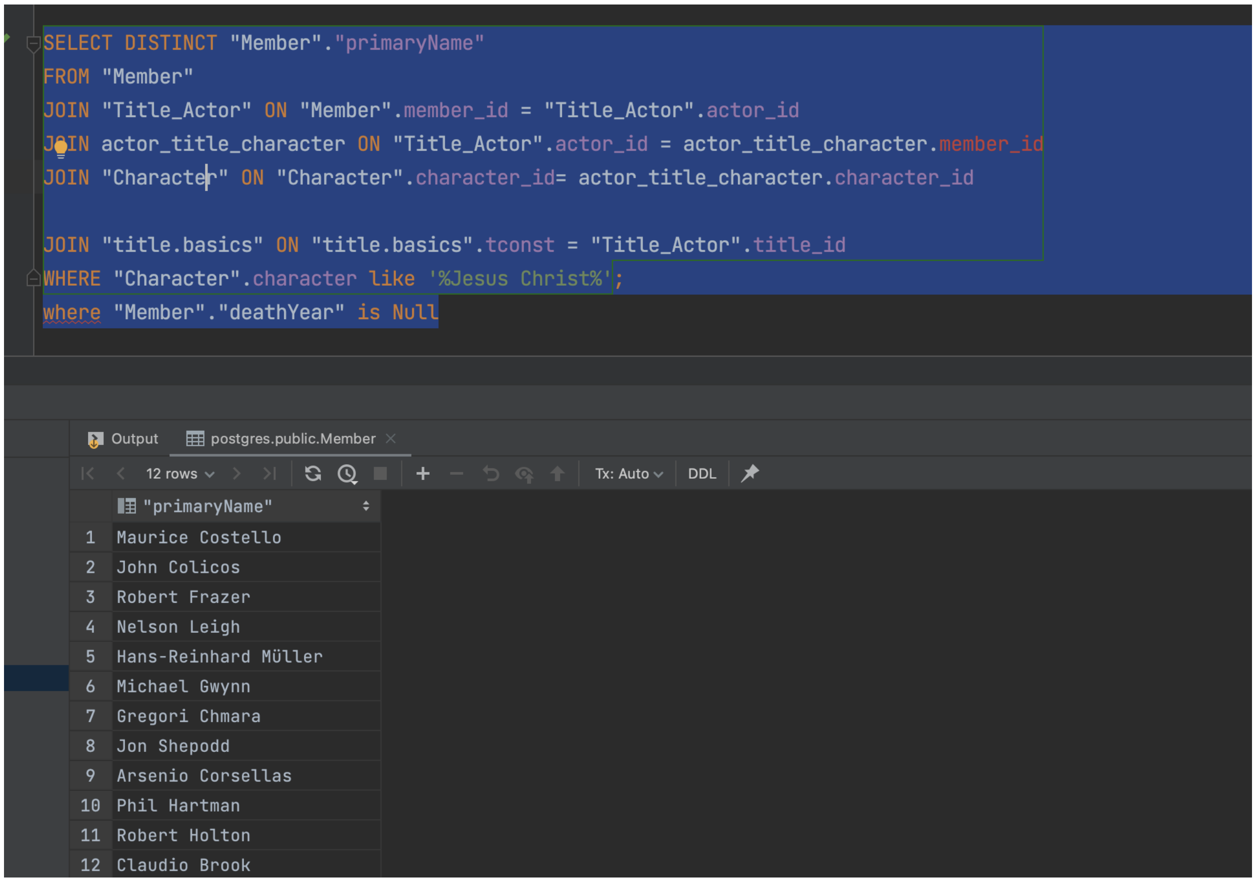
2) 5.3 seconds



```
select "Member"."primaryName"
from "Member"
where "Member"."primaryName" like 'Phi%'
and "Member"."deathYear" is null
  and member_id not in (
      select "Title_Actor".actor_id
      from "Title_Actor"
      join "Title" on "Title".title_id = "Title_Actor".title_id
      where "Title"."startYear" = 2014
);
```

```
    "primaryName"
1   Philippe Trebaol
2   Philip Core
3   Philip Desborough
4   Phil Masi
5   Phillip Cooper
6   Phil J. Munch
7   Phil Patton
8   Philomena Franz
9   Philippe Gaubert
10  Philippe Faure
11  Philip Glaessner
12  Philippe Dehaene
```

3) 2.6 seconds

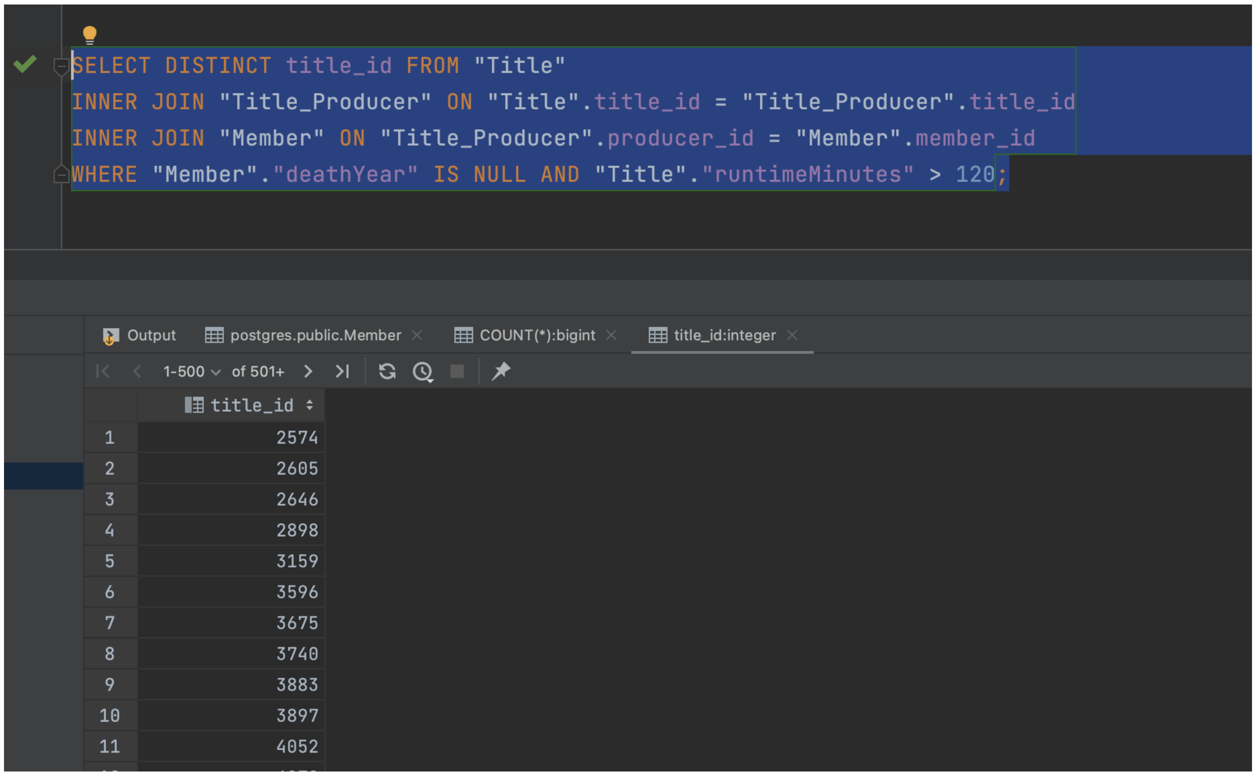


```
SELECT DISTINCT "Member"."primaryName"
FROM "Member"
JOIN "Title_Actor" ON "Member".member_id = "Title_Actor".actor_id
JOIN actor_title_character ON "Title_Actor".actor_id = actor_title_character.member_id
JOIN "Character" ON "Character".character_id= actor_title_character.character_id

JOIN "title.basics" ON "title.basics".tconst = "Title_Actor".title_id
WHERE "Character".character like '%Jesus Christ%';
where "Member"."deathYear" is Null
```

```
   "primaryName"
1  Maurice Costello
2  John Colicos
3  Robert Frazer
4  Nelson Leigh
5  Hans-Reinhard Müller
6  Michael Gwynn
7  Gregori Chmara
8  Jon Shepodd
9  Arsenio Corsellas
10 Phil Hartman
11 Robert Holton
12 Claudio Brook
```

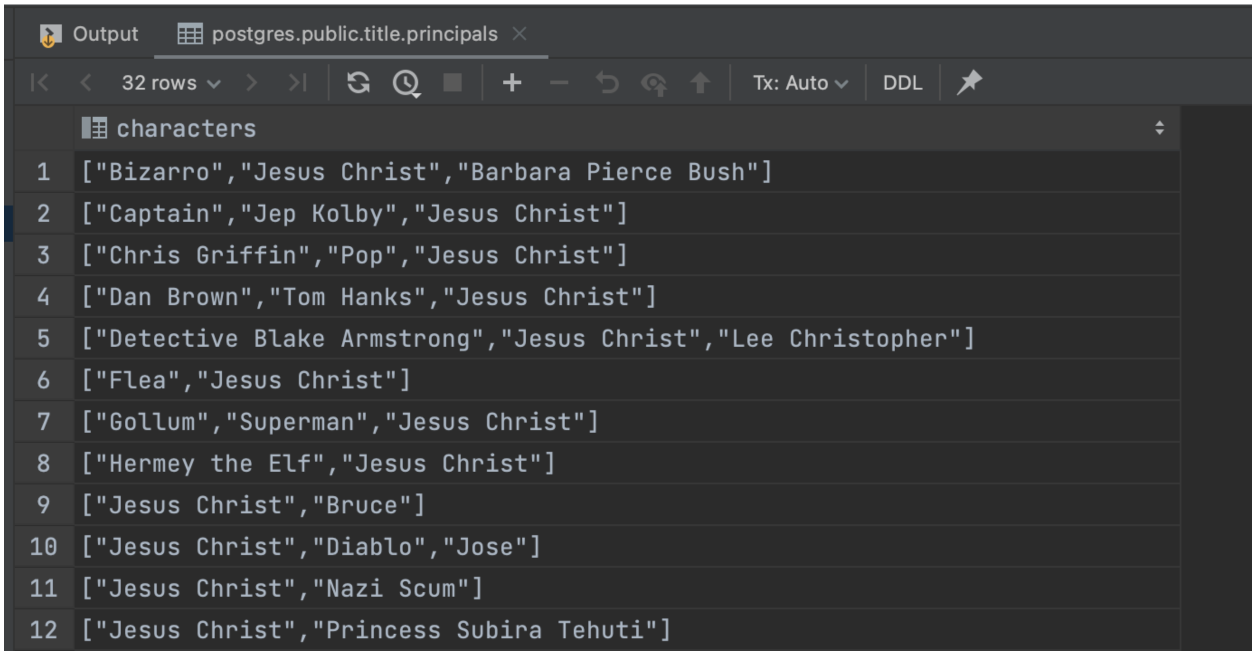
4) 5.7 seconds



```
SELECT DISTINCT title_id FROM "Title"
INNER JOIN "Title_Producer" ON "Title".title_id = "Title_Producer".title_id
INNER JOIN "Member" ON "Title_Producer".producer_id = "Member".member_id
WHERE "Member"."deathYear" IS NULL AND "Title"."runtimeMinutes" > 120;
```

```
    title_id
1       2574
2       2605
3       2646
4       2898
5       3159
6       3596
7       3675
8       3740
9       3883
10      3897
11      4052
```

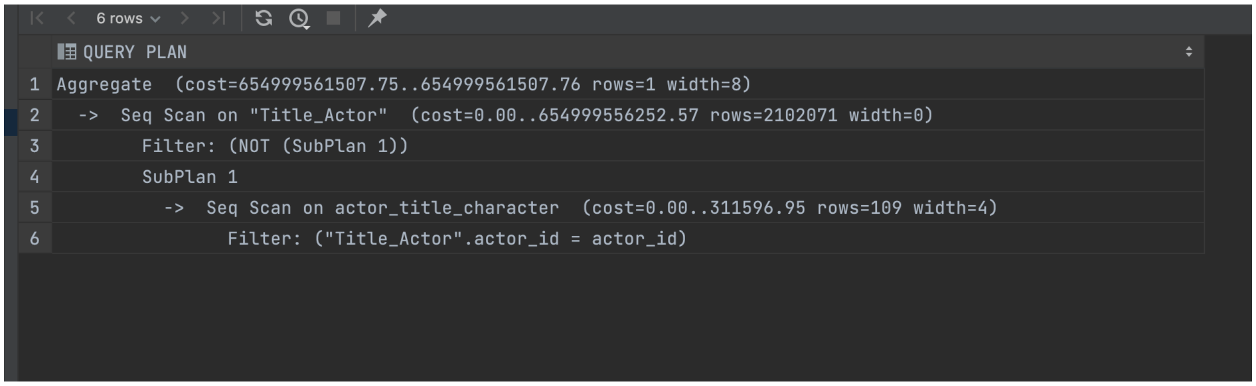
5) 4.7 seconds



```
    characters
1   ["Bizarro","Jesus Christ","Barbara Pierce Bush"]
2   ["Captain","Jep Kolby","Jesus Christ"]
3   ["Chris Griffin","Pop","Jesus Christ"]
4   ["Dan Brown","Tom Hanks","Jesus Christ"]
5   ["Detective Blake Armstrong","Jesus Christ","Lee Christopher"]
6   ["Flea","Jesus Christ"]
7   ["Gollum","Superman","Jesus Christ"]
8   ["Hermey the Elf","Jesus Christ"]
9   ["Jesus Christ","Bruce"]
10  ["Jesus Christ","Diablo","Jose"]
11  ["Jesus Christ","Nazi Scum"]
12  ["Jesus Christ","Princess Subira Tehuti"]
```

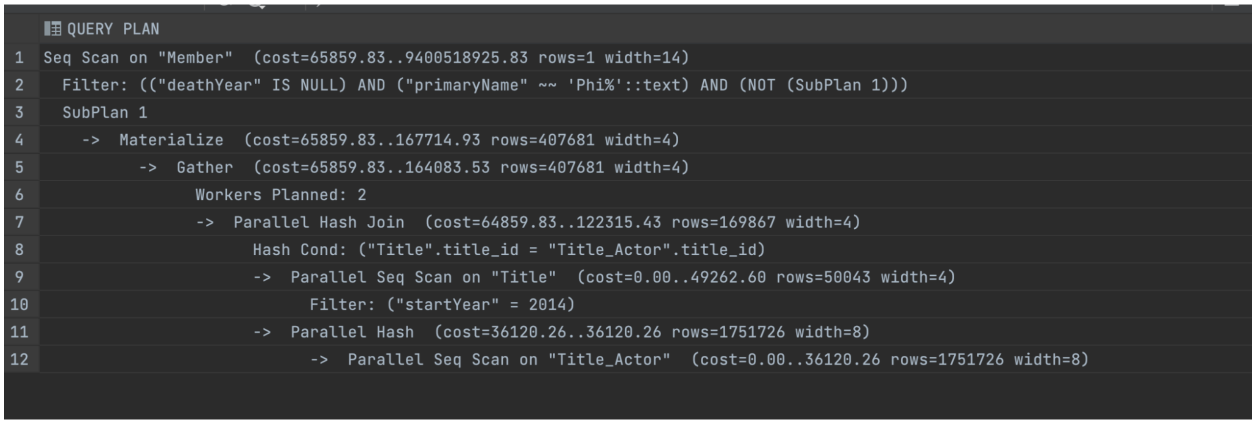
Q3)

1) **Here the subquery is executed first and the main query then uses the result to filter Title_actor.**



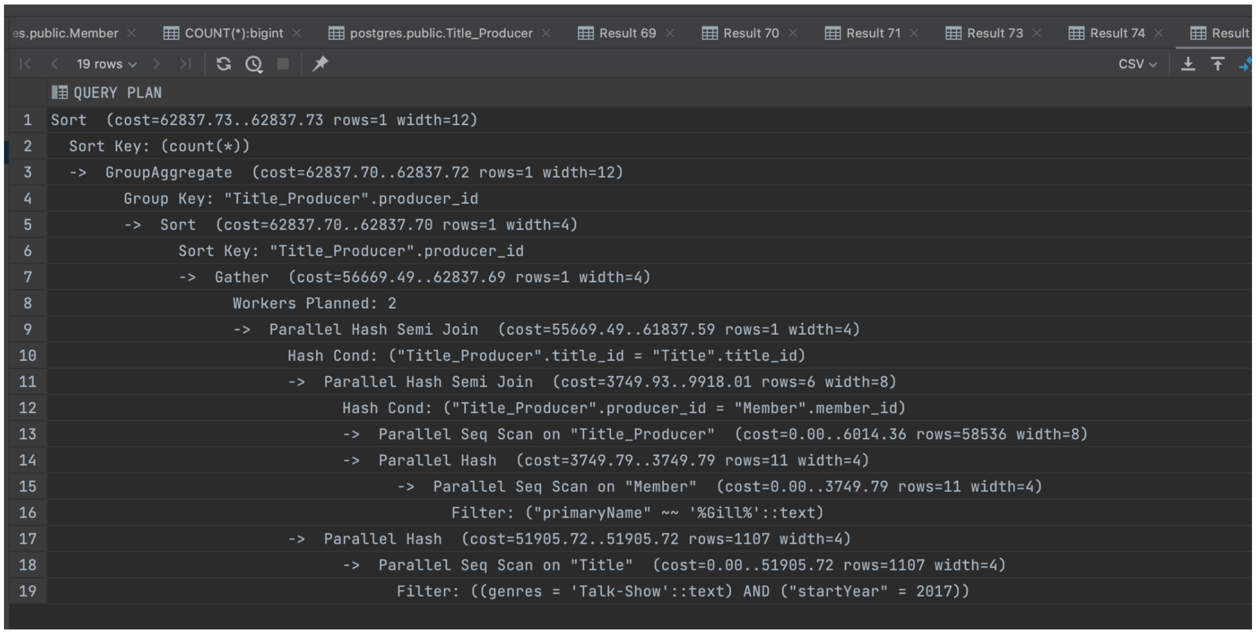
```
6 rows
QUERY PLAN
Aggregate  (cost=654999561507.75..654999561507.76 rows=1 width=8)
  ->  Seq Scan on "Title_Actor"  (cost=0.00..654999556252.57 rows=2102071 width=0)
        Filter: (NOT (SubPlan 1))
        SubPlan 1
          ->  Seq Scan on actor_title_character  (cost=0.00..311596.95 rows=109 width=4)
                Filter: ("Title_Actor".actor_id = actor_id)
```

2)



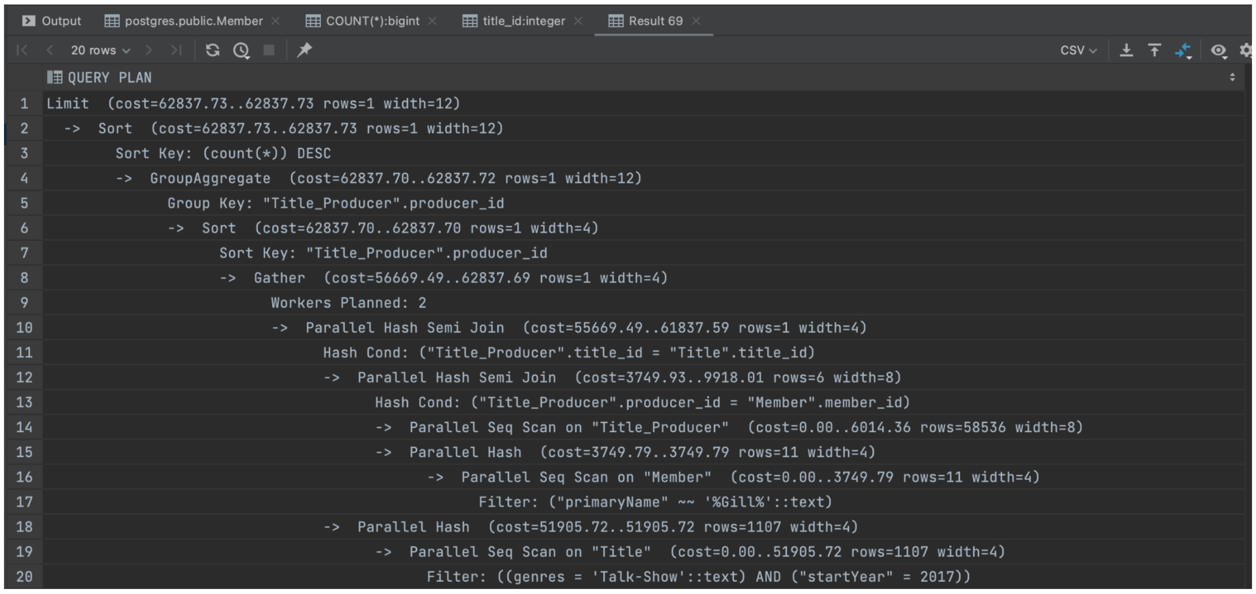
```
QUERY PLAN
Seq Scan on "Member"  (cost=65859.83..9400518925.83 rows=1 width=14)
  Filter: (("deathYear" IS NULL) AND ("primaryName" ~~ 'Phi%'::text) AND (NOT (SubPlan 1)))
  SubPlan 1
    ->  Materialize  (cost=65859.83..167714.93 rows=407681 width=4)
          ->  Gather  (cost=65859.83..164083.53 rows=407681 width=4)
                Workers Planned: 2
                ->  Parallel Hash Join  (cost=64859.83..122315.43 rows=169867 width=4)
                      Hash Cond: ("Title".title_id = "Title_Actor".title_id)
                      ->  Parallel Seq Scan on "Title"  (cost=0.00..49262.60 rows=50043 width=4)
                            Filter: ("startYear" = 2014)
                      ->  Parallel Hash  (cost=36120.26..36120.26 rows=1751726 width=8)
                            ->  Parallel Seq Scan on "Title_Actor"  (cost=0.00..36120.26 rows=1751726 width=8)
```

3)



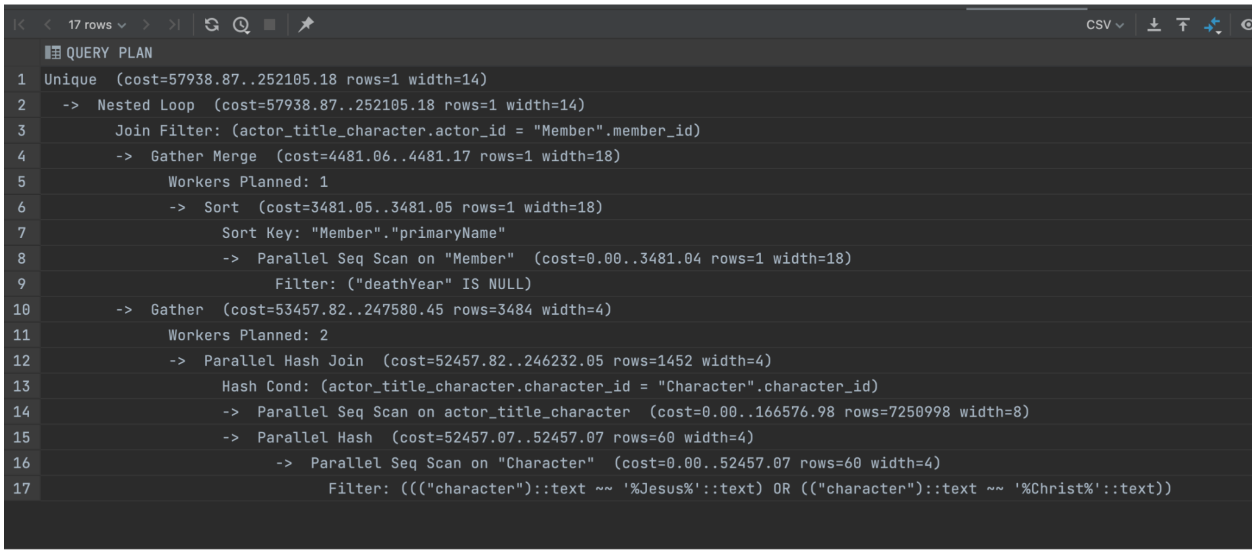
```
QUERY PLAN
Sort  (cost=62837.73..62837.73 rows=1 width=12)
  Sort Key: (count(*))
  ->  GroupAggregate  (cost=62837.70..62837.72 rows=1 width=12)
        Group Key: "Title_Producer".producer_id
        ->  Sort  (cost=62837.70..62837.70 rows=1 width=4)
              Sort Key: "Title_Producer".producer_id
              ->  Gather  (cost=56669.49..62837.69 rows=1 width=4)
                    Workers Planned: 2
                    ->  Parallel Hash Semi Join  (cost=55669.49..61837.59 rows=1 width=4)
                          Hash Cond: ("Title_Producer".title_id = "Title".title_id)
                          ->  Parallel Hash Semi Join  (cost=3749.93..9918.01 rows=6 width=8)
                                Hash Cond: ("Title_Producer".producer_id = "Member".member_id)
                                ->  Parallel Seq Scan on "Title_Producer"  (cost=0.00..6014.36 rows=58536 width=8)
                                ->  Parallel Hash  (cost=3749.79..3749.79 rows=11 width=4)
                                      ->  Parallel Seq Scan on "Member"  (cost=0.00..3749.79 rows=11 width=4)
                                            Filter: ("primaryName" ~~ '%Gill%'::text)
                          ->  Parallel Hash  (cost=51905.72..51905.72 rows=1107 width=4)
                                ->  Parallel Seq Scan on "Title"  (cost=0.00..51905.72 rows=1107 width=4)
                                      Filter: ((genres = 'Talk-Show'::text) AND ("startYear" = 2017))
```

4)

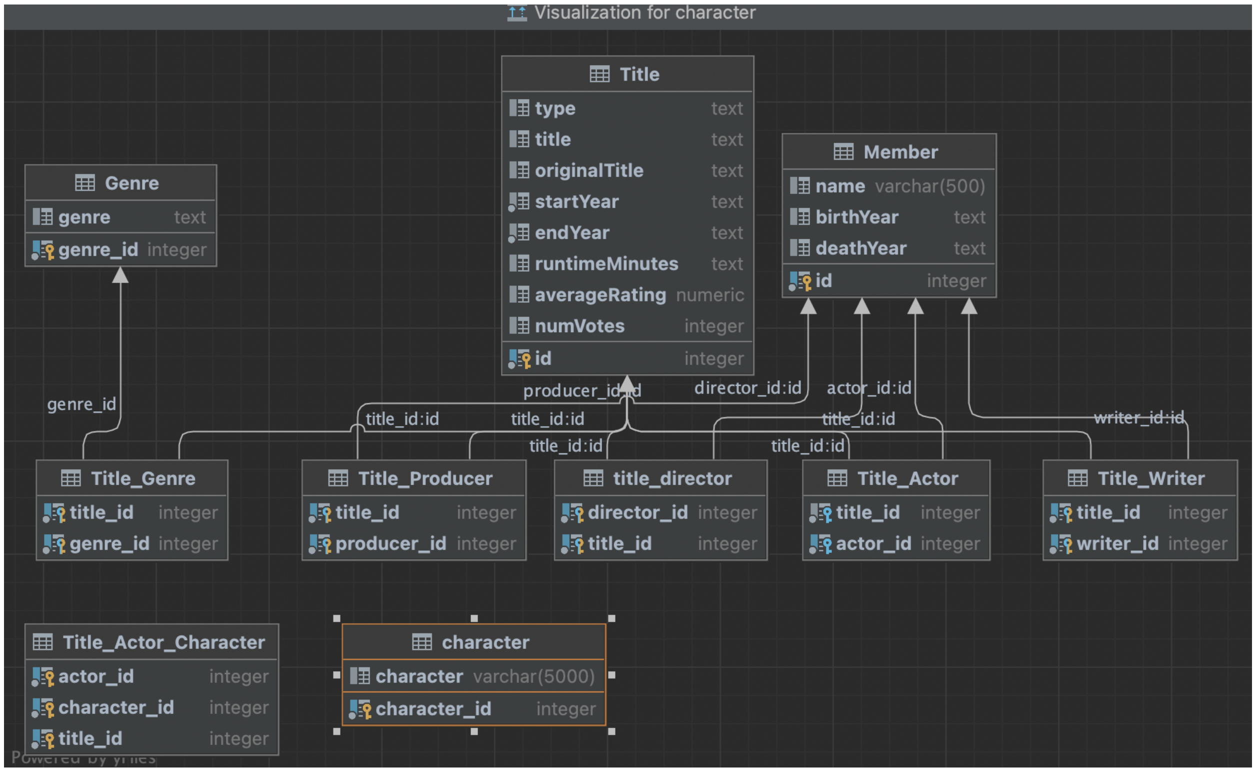


```
QUERY PLAN
Limit  (cost=62837.73..62837.73 rows=1 width=12)
  ->  Sort  (cost=62837.73..62837.73 rows=1 width=12)
        Sort Key: (count(*)) DESC
        ->  GroupAggregate  (cost=62837.70..62837.72 rows=1 width=12)
              Group Key: "Title_Producer".producer_id
              ->  Sort  (cost=62837.70..62837.70 rows=1 width=4)
                    Sort Key: "Title_Producer".producer_id
                    ->  Gather  (cost=56669.49..62837.69 rows=1 width=4)
                          Workers Planned: 2
                          ->  Parallel Hash Semi Join  (cost=55669.49..61837.59 rows=1 width=4)
                                Hash Cond: ("Title_Producer".title_id = "Title".title_id)
                                ->  Parallel Hash Semi Join  (cost=3749.93..9918.01 rows=6 width=8)
                                      Hash Cond: ("Title_Producer".producer_id = "Member".member_id)
                                      ->  Parallel Seq Scan on "Title_Producer"  (cost=0.00..6014.36 rows=58536 width=8)
                                      ->  Parallel Hash  (cost=3749.79..3749.79 rows=11 width=4)
                                            ->  Parallel Seq Scan on "Member"  (cost=0.00..3749.79 rows=11 width=4)
                                                  Filter: ("primaryName" ~~ '%Gill%'::text)
                                ->  Parallel Hash  (cost=51905.72..51905.72 rows=1107 width=4)
                                      ->  Parallel Seq Scan on "Title"  (cost=0.00..51905.72 rows=1107 width=4)
                                            Filter: ((genres = 'Talk-Show'::text) AND ("startYear" = 2017))
```

5)



```
QUERY PLAN
Unique  (cost=57938.87..252105.18 rows=1 width=14)
  ->  Nested Loop  (cost=57938.87..252105.18 rows=1 width=14)
        Join Filter: (actor_title_character.actor_id = "Member".member_id)
        ->  Gather Merge  (cost=4481.06..4481.17 rows=1 width=18)
              Workers Planned: 1
              ->  Sort  (cost=3481.05..3481.05 rows=1 width=18)
                    Sort Key: "Member"."primaryName"
                    ->  Parallel Seq Scan on "Member"  (cost=0.00..3481.04 rows=1 width=18)
                          Filter: ("deathYear" IS NULL)
        ->  Gather  (cost=53457.82..247580.45 rows=3484 width=4)
              Workers Planned: 2
              ->  Parallel Hash Join  (cost=52457.82..246232.05 rows=1452 width=4)
                    Hash Cond: (actor_title_character.character_id = "Character".character_id)
                    ->  Parallel Seq Scan on actor_title_character  (cost=0.00..166576.98 rows=7250998 width=8)
                    ->  Parallel Hash  (cost=52457.07..52457.07 rows=60 width=4)
                          ->  Parallel Seq Scan on "Character"  (cost=0.00..52457.07 rows=60 width=4)
                                Filter: ((("character")::text ~~ '%Jesus%'::text) OR (("character")::text ~~ '%Christ%'::text))
```



Q4)

1)
π actor_id (σ "Title_Actor".actor_id = actor_title_character.actor_id (Title_Actor x Actor_Title_Character))

2)
π name (σ deathYear IS NULL (σ primaryName LIKE 'Phi%' (Member)) - π actor_id (Title_Actor ⋈ σ startYear = 2014 (Title)))

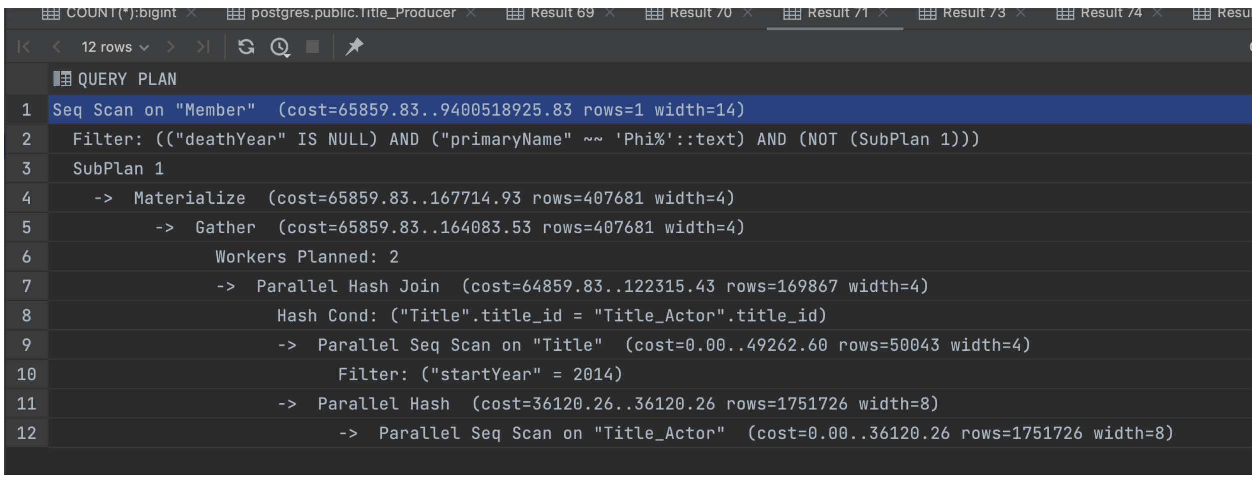
3)
π producer_id (σ COUNT(*) (γ producer_id (π producer_id (σ title_id ∈ π title_id (π title_id (σ genres = 'Talk-Show' ∧ startYear = 2017 (Title))) ∧ producer_id ∈ π member_id (σ name LIKE '%Gill%' (Member)) (Title_Producer ⋈ π title_id (π title_id (σ genres = 'Talk-Show' ∧ startYear = 2017 (Title))))))))

4)
π title_id (δ (π title_id (σ deathYear IS NULL ∧ runtimeMinutes > 120 (Title ⋈ Title_Producer ⋈ Member))))

5)
π primaryName (δ (π primaryName (σ deathYear IS NULL ∧ character LIKE '%Jesus Christ%' (Member ⋈ Title_Actor ⋈ actor_title_character))))

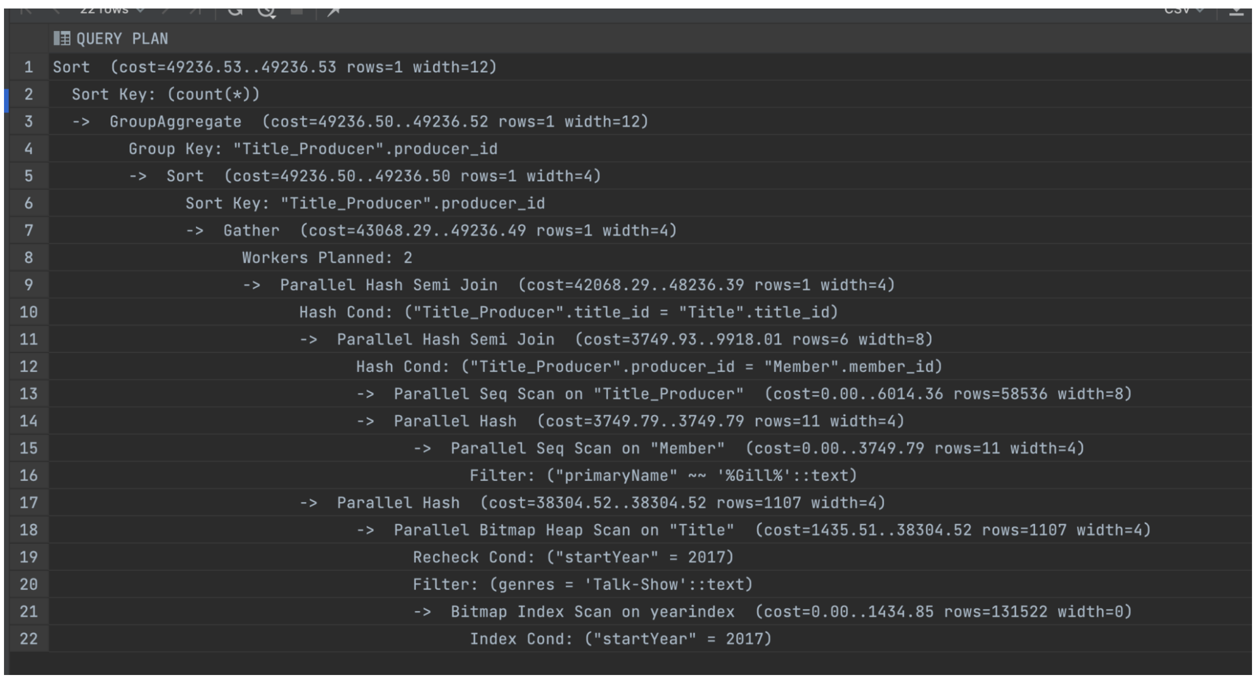
Q5)

1)We can create an index for startYear in the Title table. There was a time difference of 1 second from 3.7 to 2.697 seconds,the code is attached.



```
QUERY PLAN
Seq Scan on "Member"  (cost=65859.83..9400518925.83 rows=1 width=14)
  Filter: (("deathYear" IS NULL) AND ("primaryName" ~~ 'Phi%'::text) AND (NOT (SubPlan 1)))
  SubPlan 1
    ->  Materialize  (cost=65859.83..167714.93 rows=407681 width=4)
          ->  Gather  (cost=65859.83..164083.53 rows=407681 width=4)
                Workers Planned: 2
                ->  Parallel Hash Join  (cost=64859.83..122315.43 rows=169867 width=4)
                      Hash Cond: ("Title".title_id = "Title_Actor".title_id)
                      ->  Parallel Seq Scan on "Title"  (cost=0.00..49262.60 rows=50043 width=4)
                            Filter: ("startYear" = 2014)
                      ->  Parallel Hash  (cost=36120.26..36120.26 rows=1751726 width=8)
                            ->  Parallel Seq Scan on "Title_Actor"  (cost=0.00..36120.26 rows=1751726 width=8)
```

3)We can create an index for Member Table for the name column, since we frequently search actors or producers by name



```
QUERY PLAN
Sort  (cost=49236.53..49236.53 rows=1 width=12)
  Sort Key: (count(*))
  ->  GroupAggregate  (cost=49236.50..49236.52 rows=1 width=12)
        Group Key: "Title_Producer".producer_id
        ->  Sort  (cost=49236.50..49236.50 rows=1 width=4)
              Sort Key: "Title_Producer".producer_id
              ->  Gather  (cost=43068.29..49236.49 rows=1 width=4)
                    Workers Planned: 2
                    ->  Parallel Hash Semi Join  (cost=42068.29..48236.39 rows=1 width=4)
                          Hash Cond: ("Title_Producer".title_id = "Title".title_id)
                          ->  Parallel Hash Semi Join  (cost=3749.93..9918.01 rows=6 width=8)
                                Hash Cond: ("Title_Producer".producer_id = "Member".member_id)
                                ->  Parallel Seq Scan on "Title_Producer"  (cost=0.00..6014.36 rows=58536 width=8)
                                ->  Parallel Hash  (cost=3749.79..3749.79 rows=11 width=4)
                                      ->  Parallel Seq Scan on "Member"  (cost=0.00..3749.79 rows=11 width=4)
                                            Filter: ("primaryName" ~~ '%Gill%'::text)
                          ->  Parallel Hash  (cost=38304.52..38304.52 rows=1107 width=4)
                                ->  Parallel Bitmap Heap Scan on "Title"  (cost=1435.51..38304.52 rows=1107 width=4)
                                      Recheck Cond: ("startYear" = 2017)
                                      Filter: (genres = 'Talk-Show'::text)
                                      ->  Bitmap Index Scan on yearindex  (cost=0.00..1434.85 rows=131522 width=0)
                                            Index Cond: ("startYear" = 2017)
```