

Artefato E

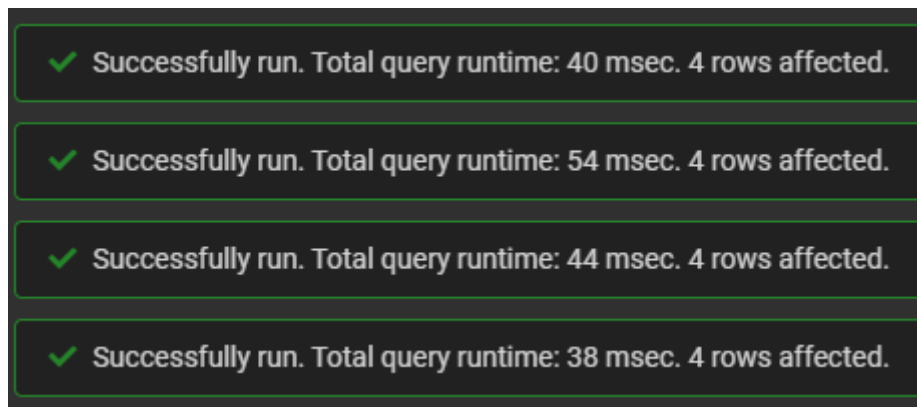
Para este, foi montada a seguinte pesquisa que se utiliza de subquery:

```
1  SELECT nome_ator
2  FROM ator
3  WHERE EXISTS(
4      SELECT obra.nome
5      FROM obra
6      INNER JOIN esta_em
7      ON esta_em.obra_id = obra.id
8      WHERE ator.id = esta_em.ator_id
9      AND obra.nome = 'Brocolis 99'
10 )
```

1.	→ Nested Loop Inner Join (cost=76.01..77.49 rows=5 width=13)
2.	→ Aggregate (cost=75.87..75.92 rows=5 width=4)
3.	→ Hash Inner Join (cost=32.39..75.85 rows=5 width=4) Hash Cond: (esta_em.obra_id = obra.id)
4.	→ Seq Scan on esta_em as esta_em (cost=0..36.88 rows=2488 width=8)
5.	→ Hash (cost=32.38..32.38 rows=1 width=4)
6.	→ Seq Scan on obra as obra (cost=0..32.38 rows=1 width=4) Filter: (nome = 'Brocolis 99'::text)
7.	→ Index Scan using ator_pkey on ator as ator (cost=0.14..0.31 rows=1 width=17) Index Cond: (id = esta_em.ator_id)

Podemos ver que primeiro, a subquery é executada, depois seu resultado é passado para o SELECT exterior, que é executado.

Em questão de tempo no caso temos:

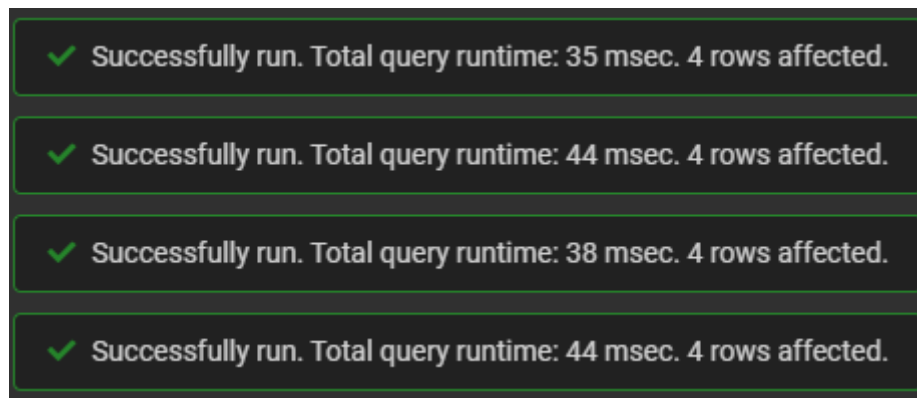


Agora, para que não necessitemos de subqueries, precisamos nos utilizar de vários INNER JOINS, portanto:

```
1  SELECT nome_ator FROM obra
2  INNER JOIN esta_em
3  ON obra.id = esta_em.obra_id
4  INNER JOIN ator
5  ON esta_em.ator_id = ator.id
6  WHERE obra.nome = 'Brocolis 99'
```

Node	
1.	→ Nested Loop Inner Join (cost=32.53..76.68 rows=5 width=13)
2.	→ Hash Inner Join (cost=32.39..75.85 rows=5 width=4) Hash Cond: (esta_em.obra_id = obra.id)
3.	→ Seq Scan on esta_em as esta_em (cost=0..36.88 rows=2488 width=8)
4.	→ Hash (cost=32.38..32.38 rows=1 width=4)
5.	→ Seq Scan on obra as obra (cost=0..32.38 rows=1 width=4) Filter: (nome = 'Brocolis 99':text)
6.	→ Index Scan using ator_pkey on ator as ator (cost=0.14..0.16 rows=1 width=17) Index Cond: (id = esta_em.ator_id)

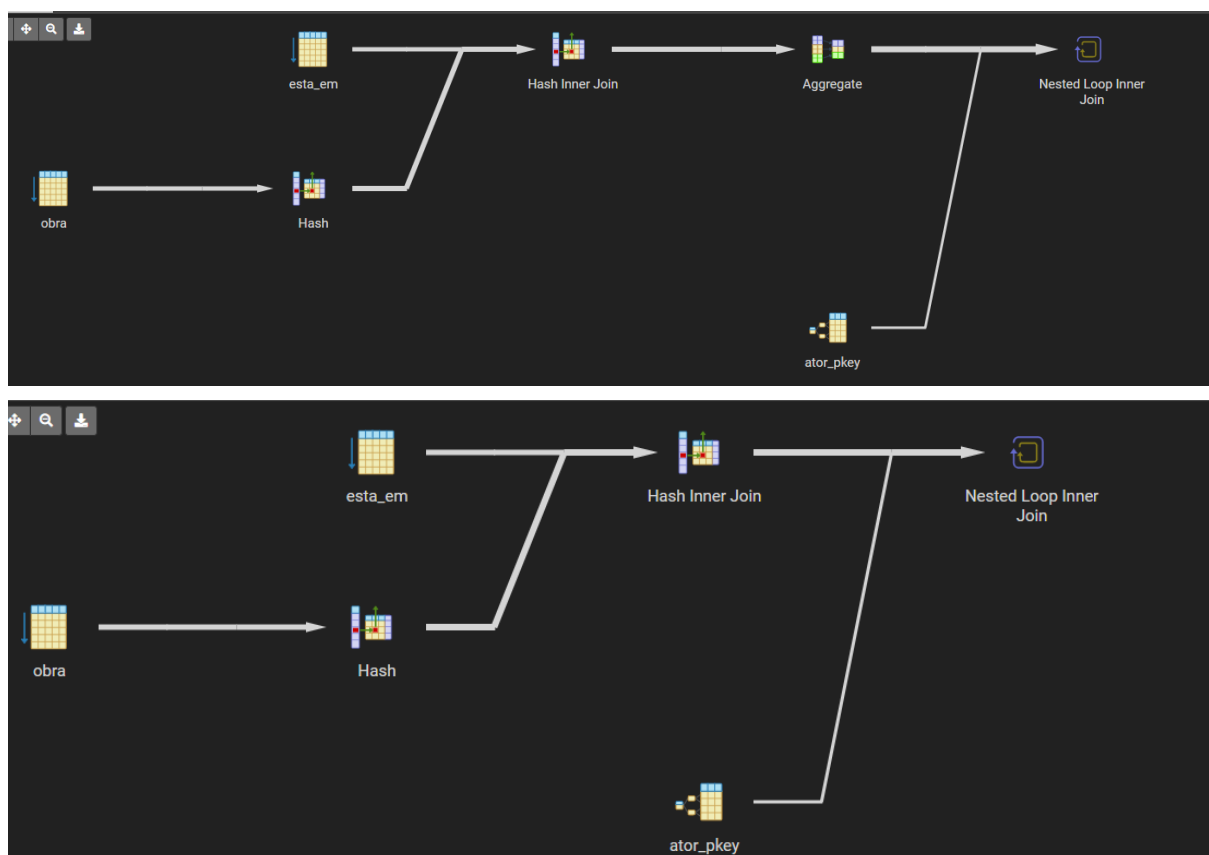
Que também possui um tempo similar (dada a pequena amostra de testes):



Comparando os dois, temos que (em cima o com subquery, em baixo o sem), vemos que ambas as consultas possuem planos e custos equivalentes, exceto pela necessidade de um aggregate, proporcionado por

```
ON esta_em.obra_id = obra.id  
WHERE ator.id = esta_em.ator_id
```

Que é feito na chave comum a ambos.



Portanto, a query que não se utiliza da estratégia de subqueries, neste caso, tem um custo menor e, conseqüentemente, uma performance maior (mesmo que imperceptível neste caso).