

Normal query is 90ms New faster query without index is 7ms.

Normal:

```

GroupAggregate (cost=193423.12..193451.71 rows=128 width=21)
  Group Key: actor.first_name, actor.last_name
  -> Sort (cost=193423.12..193429.95 rows=2731 width=15)
      Sort Key: actor.first_name, actor.last_name
      -> Nested Loop (cost=0.28..193267.25 rows=2731 width=15)
          -> Seq Scan on actor (cost=0.00..4.00 rows=200 width=17)
          -> Index Only Scan using film_actor_pkey on film_actor
              (cost=0.28..966.18 rows=14 width=4)
                  Index Cond: (actor_id = actor.actor_id)
                  Filter: (SubPlan 1)
                  SubPlan 1
                      -> Materialize (cost=0.00..70.16 rows=531
width=4)
                                  -> Seq Scan on film (cost=0.00..67.50
rows=531 width=4)
                                      Filter: (length < 120)

```

In the normal query there is a nested loop. First loop must wait for the second loop to finish.

```

faster:
Sort (cost=210.72..211.04 rows=128 width=21)
  Sort Key: actor.first_name, actor.last_name
  -> HashAggregate (cost=204.96..206.24 rows=128 width=21)
      Group Key: actor.first_name, actor.last_name
      -> Hash Join (cost=79.86..185.75 rows=2562 width=17)
          Hash Cond: (film_actor.actor_id = actor.actor_id)
          -> Hash Join (cost=73.36..172.38 rows=2562 width=6)
              Hash Cond: (film_actor.film_id = film.film_id)
              -> Seq Scan on film_actor (cost=0.00..84.62
rows=5462 width=4)
                  -> Hash (cost=67.50..67.50 rows=469 width=4)
                      -> Seq Scan on film (cost=0.00..67.50
rows=469 width=4)
                          Filter: (length >= 120)
          -> Hash (cost=4.00..4.00 rows=200 width=17)
              -> Seq Scan on actor (cost=0.00..4.00 rows=200
width=17)

```

In the new query there is a hash join. The hash join is faster because it can start the second loop before the first loop is finished.