Original query

```
-> Limit: 50 row(s)  (actual time=89139..89139 rows=50 loops=1)
    -> Sort: avg_rating_amazon DESC  (actual time=89139..89139 rows=50
loops=1)
        -> Table scan on <temporary>  (cost=2.5..2.5 rows=0) (actual
time=89139..89139 rows=538 loops=1)
            -> Temporary table  (cost=0..0 rows=0) (actual time=89139..89139
rows=538 loops=1)
                -> Window aggregate: rank() OVER (PARTITION BY pb.genre ORDER
BY `avg(perfect_books.total_ratings)` desc )   (actual time=89136..89138
rows=538 loops=1)
                    -> Filter: (`count(distinct amazon_books.title)` >= 2)
(actual time=89136..89137 rows=538 loops=1)
                        -> Sort: pb.genre, `avg(perfect_books.total_ratings)`
DESC  (actual time=89136..89137 rows=2922 loops=1)
                            -> Stream results  (actual time=89069..89127
rows=2922 loops=1)
                                -> Group aggregate: count(distinct
amazon_books.title), avg(perfect_books.total_ratings), avg(tmp_field),
count(distinct amazon_books.title), avg(amazon_books.rating),
avg(perfect_books.rating)  (actual time=89069..89121 rows=2922 loops=1)
                                    -> Sort: pb.genre, pb.author  (actual
time=89069..89075 rows=10950 loops=1)
                                        -> Stream results  (cost=10666
rows=675) (actual time=38.2..89000 rows=10950 loops=1)
                                            -> Filter: ((pb.title = ab.title)
and (pb.author = ab.author))  (cost=10666 rows=675) (actual time=29.6..175
rows=10950 loops=1)
                                                -> Inner hash join
(<hash>(pb.title)=<hash>(ab.title)), (<hash>(pb.author)=<hash>(ab.author))
(cost=10666 rows=675) (actual time=29.6..147 rows=10950 loops=1)
                                                    -> Filter: (ab.price is
not null)  (cost=1.96 rows=68.7) (actual time=0.0327..20.7 rows=7928 loops=1)
                                                        -> Table scan on ab
(cost=1.96 rows=7630) (actual time=0.0313..18.1 rows=7928 loops=1)
                                                    -> Hash
                                                        -> Filter:
((pb.rating >= 4) and (pb.total_ratings > 100))  (cost=1039 rows=1092)
(actual time=0.0834..19.9 rows=7342 loops=1)
                                                            -> Table scan on
pb  (cost=1039 rows=9827) (actual time=0.0774..17.5 rows=10000 loops=1)
-> Select #2 (subquery in projection; dependent)
    -> Aggregate: count(0)  (cost=386 rows=1) (actual time=8.09..8.09 rows=1
loops=10950)
        -> Filter: ((apb.author = pb.author) and (apb.rating > 4.5))
(cost=370 rows=164) (actual time=6.66..8.08 rows=1.34 loops=10950)
            -> Table scan on apb  (cost=370 rows=4932) (actual
time=0.0125..6.9 rows=4846 loops=10950)
```

- Загальний час виконання: 89 139 мс

- Кількість викликів підзапиту: 10 950

- Тимчасові таблиці: так

- Тип доступу: повне сканування

- Індекси не використовуються

Step 1 refactor

```
-> Limit: 50 row(s)  (actual time=319..319 rows=50 loops=1)
    -> Sort: avg_rating_amazon DESC  (actual time=319..319 rows=50 loops=1)
        -> Table scan on <temporary>  (cost=2.5..2.5 rows=0) (actual
time=318..318 rows=538 loops=1)
            -> Temporary table  (cost=0..0 rows=0) (actual time=318..318
rows=538 loops=1)
                -> Window aggregate: rank() OVER (PARTITION BY b.genre ORDER
BY `avg(perfect_books.total_ratings)` desc )   (actual time=314..318 rows=538
loops=1)
                    -> Filter: (`count(distinct amazon_books.title)` >= 2)
(actual time=314..315 rows=538 loops=1)
                        -> Sort: b.genre, `avg(perfect_books.total_ratings)`
DESC  (actual time=314..315 rows=2922 loops=1)
                            -> Stream results  (actual time=239..304
rows=2922 loops=1)
                                -> Group aggregate: count(distinct
amazon_books.title), avg(perfect_books.total_ratings), avg(price),
count(distinct amazon_books.title), avg(amazon_books.amazon_rating),
avg(perfect_books.perfect_rating), max(tmp_field)  (actual time=239..298
rows=2922 loops=1)
                                    -> Sort: b.genre, b.author  (actual
time=239..246 rows=10950 loops=1)
                                        -> Stream results  (cost=198421
rows=0) (actual time=53.5..205 rows=10950 loops=1)
                                            -> Nested loop left join
(cost=198421 rows=0) (actual time=53.4..175 rows=10950 loops=1)
                                                -> Filter: ((pb.title =
ab.title) and (pb.author = ab.author))  (cost=10666 rows=675) (actual
time=30.8..116 rows=10950 loops=1)
                                                    -> Inner hash join
(<hash>(pb.title)=<hash>(ab.title)), (<hash>(pb.author)=<hash>(ab.author))
(cost=10666 rows=675) (actual time=30.8..96.7 rows=10950 loops=1)
                                                        -> Filter: (ab.price
is not null)  (cost=1.96 rows=68.7) (actual time=0.0269..17.8 rows=7928
loops=1)
                                                            -> Table scan on
ab  (cost=1.96 rows=7630) (actual time=0.0256..16.7 rows=7928 loops=1)
                                                        -> Hash
                                                            -> Filter:
((pb.rating >= 4) and (pb.total_ratings > 100))  (cost=1039 rows=1092)
(actual time=0.0836..20.9 rows=7342 loops=1)
                                                                -> Table scan
on pb  (cost=1039 rows=9827) (actual time=0.0781..18.2 rows=10000 loops=1)
                                                -> Index lookup on ps using
<auto_key0> (author=pb.author)  (cost=0.25..2.5 rows=10) (actual
time=0.0049..0.00503 rows=0.23 loops=10950)
                                    -> Materialize CTE
popular_stats  (cost=0..0 rows=0) (actual time=22.7..22.7 rows=2267 loops=1)
                                        -> Table scan on
<temporary>  (actual time=17..17.6 rows=2267 loops=1)
                                            -> Aggregate
using temporary table  (actual time=17..17 rows=2267 loops=1)
                                                -> Filter:
(amazon_popular_books.rating > 4.5)  (cost=517 rows=1644) (actual
time=0.015..8.64 rows=3745 loops=1)
                                                    -> Table
scan on amazon_popular_books  (cost=517 rows=4932) (actual time=0.014..7.9
rows=4846 loops=1)
```

- Загальний час виконання: 319 мс

- Кількість викликів підзапиту: 1 (використовується CTE один раз)

- Тимчасові таблиці: так

- Тип доступу: повне сканування

- Індекси не використовуються

Step 2 indexing

```
-> Limit: 50 row(s)  (actual time=416..416 rows=50 loops=1)
    -> Sort: avg_rating_amazon DESC  (actual time=416..416 rows=50 loops=1)
        -> Table scan on <temporary>  (cost=2.5..2.5 rows=0) (actual
time=415..416 rows=538 loops=1)
            -> Temporary table  (cost=0..0 rows=0) (actual time=415..415
rows=538 loops=1)
                -> Window aggregate: rank() OVER (PARTITION BY b.genre ORDER
BY `avg(perfect_books.total_ratings)` desc )   (actual time=413..415 rows=538
loops=1)
                    -> Filter: (`count(distinct amazon_books.title)` >= 2)
(actual time=413..414 rows=538 loops=1)
                        -> Sort: b.genre, `avg(perfect_books.total_ratings)`
DESC  (actual time=413..414 rows=2922 loops=1)
                            -> Stream results  (actual time=335..401
rows=2922 loops=1)
                                -> Group aggregate: count(distinct
amazon_books.title), avg(perfect_books.total_ratings), avg(price),
count(distinct amazon_books.title), avg(amazon_books.amazon_rating),
avg(perfect_books.perfect_rating), max(tmp_field)  (actual time=335..394
rows=2922 loops=1)
                                    -> Sort: b.author, b.genre  (actual
time=335..343 rows=10950 loops=1)
                                        -> Stream results  (cost=748596
rows=7.35e+6) (actual time=21.9..300 rows=10950 loops=1)
                                            -> Nested loop left join
(cost=748596 rows=7.35e+6) (actual time=21.9..264 rows=10950 loops=1)
                                                -> Nested loop inner join
(cost=2777 rows=4469) (actual time=0.132..198 rows=10950 loops=1)
                                                    -> Filter: ((pb.rating >=
4) and (pb.total_ratings > 100))  (cost=1039 rows=3725) (actual
time=0.0643..27 rows=7342 loops=1)
                                                        -> Table scan on pb
(cost=1039 rows=9827) (actual time=0.0605..23 rows=10000 loops=1)
                                                    -> Filter: ((ab.price is
not null) and (pb.title = ab.title) and (pb.author = ab.author))  (cost=0.333
rows=1.2) (actual time=0.0148..0.0229 rows=1.49 loops=7342)
                                                        -> Index lookup on ab
using idx_amazon_books_title_author (title=pb.title, author=pb.author)
(cost=0.333 rows=1.33) (actual time=0.013..0.0194 rows=1.5 loops=7342)
                                                -> Index lookup on ps using
<auto_key0> (author=pb.author)  (cost=846..849 rows=10) (actual
time=0.00543..0.00558 rows=0.23 loops=10950)
                                            -> Materialize CTE
popular_stats  (cost=846..846 rows=1644) (actual time=21.7..21.7 rows=2267
loops=1)
                                                -> Group aggregate:
count(0)  (cost=682 rows=1644) (actual time=0.0351..13.8 rows=2267 loops=1)
```

```
                                                                -> Filter:
(amazon_popular_books.rating > 4.5)  (cost=517 rows=1644) (actual
time=0.021..10.6 rows=3745 loops=1)
                                                                    -> Covering
index scan on amazon_popular_books using idx_popular_authot_rating  (cost=517
rows=4932) (actual time=0.0198..9.51 rows=4846 loops=1)
```

- Загальний час виконання: 416 мс

- Кількість викликів підзапиту: 1 (використовується CTE один раз)

- Тимчасові таблиці: так

- Тип доступу: індексне сканування

- Індекси використані: idx_popular_author_rating

Step 3 Joins

```
-> Limit: 50 row(s)  (actual time=343..343 rows=50 loops=1)
    -> Sort: avg_rating_amazon DESC  (actual time=343..343 rows=50 loops=1)
        -> Table scan on <temporary>  (cost=2.5..2.5 rows=0) (actual
time=342..342 rows=538 loops=1)
            -> Temporary table  (cost=0..0 rows=0) (actual time=342..342
rows=538 loops=1)
                -> Window aggregate: rank() OVER (PARTITION BY b.genre ORDER
BY `avg(perfect_books.total_ratings)` desc )   (actual time=339..342 rows=538
loops=1)
                    -> Filter: (`count(distinct amazon_books.title)` >= 2)
(actual time=339..341 rows=538 loops=1)
                        -> Sort: b.genre, `avg(perfect_books.total_ratings)`
DESC  (actual time=339..340 rows=2922 loops=1)
                            -> Stream results  (actual time=278..330
rows=2922 loops=1)
                                -> Group aggregate: count(distinct
amazon_books.title), avg(perfect_books.total_ratings),
avg(perfect_books.perfect_rating), avg(amazon_books.amazon_rating),
avg(price), count(distinct amazon_books.title), max(tmp_field)  (actual
time=278..325 rows=2922 loops=1)
                                    -> Sort: b.author, b.genre  (actual
time=278..283 rows=10950 loops=1)
                                        -> Stream results  (cost=748596
rows=7.35e+6) (actual time=16.9..251 rows=10950 loops=1)
                                            -> Nested loop left join
(cost=748596 rows=7.35e+6) (actual time=16.8..220 rows=10950 loops=1)
                                                -> Nested loop inner join
(cost=2777 rows=4469) (actual time=0.118..166 rows=10950 loops=1)
                                                    -> Filter: ((pb.rating >=
4) and (pb.total_ratings > 100) and (pb.title is not null) and (pb.author is
not null))  (cost=1039 rows=3725) (actual time=0.0582..22.8 rows=7342
loops=1)
                                                        -> Table scan on pb
(cost=1039 rows=9827) (actual time=0.0538..19.3 rows=10000 loops=1)
                                                    -> Filter: ((pb.title =
ab.title) and (pb.author = ab.author) and (ab.price is not null))
(cost=0.333 rows=1.2) (actual time=0.012..0.0191 rows=1.49 loops=7342)
                                                        -> Index lookup on ab
using idx_amazon_books_title_author (title=pb.title, author=pb.author)
(cost=0.333 rows=1.33) (actual time=0.0105..0.016 rows=1.5 loops=7342)
```

```
                                                    -> Index lookup on ps using
<auto_key0> (author=pb.author)  (cost=846..849 rows=10) (actual
time=0.00445..0.00457 rows=0.23 loops=10950)
                                                -> Materialize CTE
popular_stats  (cost=846..846 rows=1644) (actual time=16.7..16.7 rows=2267
loops=1)
                                                    -> Group aggregate:
count(0)  (cost=682 rows=1644) (actual time=0.0214..10.7 rows=2267 loops=1)
                                                        -> Filter:
(amazon_popular_books.rating > 4.5)  (cost=517 rows=1644) (actual
time=0.0121..8.32 rows=3745 loops=1)
                                                            -> Covering
index scan on amazon_popular_books using idx_popular_authot_rating  (cost=517
rows=4932) (actual time=0.0115..7.55 rows=4846 loops=1)
```

- Загальний час виконання: 343 мс

- Кількість викликів підзапиту: 1 (використовується CTE один раз)

- Тимчасові таблиці: так

- Тип доступу: індексне сканування

- Індекси використані: idx_popular_author_rating

Step 4 index hints

```
-> Limit: 50 row(s)  (actual time=401..401 rows=50 loops=1)
    -> Sort: avg_rating_amazon DESC  (actual time=401..401 rows=50 loops=1)
        -> Table scan on <temporary>  (cost=2.5..2.5 rows=0) (actual
time=401..401 rows=538 loops=1)
            -> Temporary table  (cost=0..0 rows=0) (actual time=401..401
rows=538 loops=1)
                -> Window aggregate: rank() OVER (PARTITION BY b.genre ORDER
BY `avg(perfect_books.total_ratings)` desc )   (actual time=398..400 rows=538
loops=1)
                    -> Filter: (`count(distinct amazon_books.title)` >= 2)
(actual time=398..399 rows=538 loops=1)
                        -> Sort: b.genre, `avg(perfect_books.total_ratings)`
DESC  (actual time=398..399 rows=2922 loops=1)
                            -> Stream results  (actual time=332..387
rows=2922 loops=1)
                                -> Group aggregate: count(distinct
amazon_books.title), avg(perfect_books.total_ratings),
avg(perfect_books.perfect_rating), avg(amazon_books.amazon_rating),
avg(price), count(distinct amazon_books.title), max(tmp_field)  (actual
time=332..382 rows=2922 loops=1)
                                    -> Sort: b.author, b.genre  (actual
time=332..338 rows=10950 loops=1)
                                        -> Stream results  (cost=749768
rows=7.35e+6) (actual time=15.7..302 rows=10950 loops=1)
                                            -> Nested loop left join
(cost=749768 rows=7.35e+6) (actual time=15.7..268 rows=10950 loops=1)
                                                -> Nested loop inner join
(cost=3949 rows=4469) (actual time=0.124..211 rows=10950 loops=1)
                                                    -> Filter:
((pb.total_ratings > 100) and (pb.title is not null) and (pb.author is not
null))  (cost=2211 rows=3725) (actual time=0.0786..42 rows=7342 loops=1)
                                                        -> Index range scan
on pb using idx_perfect_rating over (4 <= rating), with index condition:
```

```
(pb.rating >= 4)  (cost=2211 rows=4913) (actual time=0.0759..38.6 rows=9250
loops=1)
                                                    -> Filter: ((pb.title =
ab.title) and (pb.author = ab.author) and (ab.price is not null))
(cost=0.333 rows=1.2) (actual time=0.0149..0.0225 rows=1.49 loops=7342)
                                                      -> Index lookup on ab
using idx_amazon_books_title_author (title=pb.title, author=pb.author)
(cost=0.333 rows=1.33) (actual time=0.0132..0.0192 rows=1.5 loops=7342)
                                                  -> Index lookup on ps using
<auto_key0> (author=pb.author)  (cost=2055..2058 rows=10) (actual
time=0.00463..0.00476 rows=0.23 loops=10950)
                                                    -> Materialize CTE
popular_stats  (cost=2055..2055 rows=1644) (actual time=15.6..15.6 rows=2267
loops=1)
                                                      -> Group aggregate:
count(0)  (cost=1891 rows=1644) (actual time=0.0292..9.99 rows=2267 loops=1)
                                                        -> Filter:
(amazon_popular_books.rating > 4.5)  (cost=1726 rows=1644) (actual
time=0.017..7.73 rows=3745 loops=1)
                                                          -> Covering
index scan on amazon_popular_books using idx_popular_authot_rating
(cost=1726 rows=4932) (actual time=0.016..7.02 rows=4846 loops=1)
```

- Загальний час виконання: 401 мс

- Кількість викликів підзапиту: 1 (використовується CTE один раз)

- Тимчасові таблиці: так

- Тип доступу: індекси

- Індекси використані: idx_perfect_rating, idx_popular_author_rating, idx_amazon_books_title_author