

```
Q1: Query 1: Top 15 Buildings for Clean Bathrooms AND a Vending Machine
--> postgres post -d amethysts -u 'root'
EXPLAIN ANALYZE
SELECT
  B.Name AS BuildingName,
  A.Type AS AmenityType,
  R.Rating AS R.RatingRating AS R.Rating,
  A.Address
FROM Building B
JOIN Amenity A ON B.BuildingId = A.BuildingId
JOIN Review R ON A.AmenityId = A.AmenityId
JOIN Address A_2 ON B.AddressId = A_2.AddressId
WHERE A.Type = 'Bathroom'
AND EXISTS (
  SELECT 1
  FROM Amenity A2
  WHERE A2.BuildingId = B.BuildingId
  AND A2.Type = 'VendingMachine'
)
GROUP BY B.Name, A.Type, A_2.Address
ORDER BY Avg_Bathroom_Rating DESC
LIMIT 15;
50;

Q2: Query 1: Top 15 Buildings for Clean Bathrooms AND a Vending Machine
--> postgres post -d amethysts -u 'root'
EXPLAIN ANALYZE
SELECT
  B.Name AS BuildingName,
  A.Type AS AmenityType,
  R.Rating AS R.RatingRating AS R.Rating,
  A.Address
FROM Building B
JOIN Amenity A ON B.BuildingId = A.BuildingId
JOIN Review R ON A.AmenityId = A.AmenityId
JOIN Address A_2 ON B.AddressId = A_2.AddressId
WHERE A.Type = 'Bathroom'
AND EXISTS (
  SELECT 1
  FROM Amenity A2
  WHERE A2.BuildingId = B.BuildingId
  AND A2.Type = 'VendingMachine'
)
GROUP BY B.Name, A.Type, A_2.Address
ORDER BY Avg_Bathroom_Rating DESC
LIMIT 15;
50;

Q3: Query 1: Top 15 Buildings for Clean Bathrooms AND a Vending Machine
--> postgres post -d amethysts -u 'root'
EXPLAIN ANALYZE
SELECT
  B.Name AS BuildingName,
  A.Type AS AmenityType,
  R.Rating AS R.RatingRating AS R.Rating,
  A.Address
FROM Building B
JOIN Amenity A ON B.BuildingId = A.BuildingId
JOIN Review R ON A.AmenityId = A.AmenityId
JOIN Address A_2 ON B.AddressId = A_2.AddressId
WHERE A.Type = 'Bathroom'
AND EXISTS (
  SELECT 1
  FROM Amenity A2
  WHERE A2.BuildingId = B.BuildingId
  AND A2.Type = 'VendingMachine'
)
GROUP BY B.Name, A.Type, A_2.Address
ORDER BY Avg_Bathroom_Rating DESC
LIMIT 15;
50;
```

```
Q1: Query 1: Top 15 Buildings for Clean Bathrooms AND a Vending Machine
--> postgres post -d amethysts -u 'root'
EXPLAIN ANALYZE
SELECT
  B.Name AS BuildingName,
  A.Type AS AmenityType,
  R.Rating AS R.RatingRating AS R.Rating,
  A.Address
FROM Building B
JOIN Amenity A ON B.BuildingId = A.BuildingId
JOIN Review R ON A.AmenityId = A.AmenityId
JOIN Address A_2 ON B.AddressId = A_2.AddressId
WHERE A.Type = 'Bathroom'
AND EXISTS (
  SELECT 1
  FROM Amenity A2
  WHERE A2.BuildingId = B.BuildingId
  AND A2.Type = 'VendingMachine'
)
GROUP BY B.Name, A.Type, A_2.Address
ORDER BY Avg_Bathroom_Rating DESC
LIMIT 15;
50;

Q2: Query 1: Top 15 Buildings for Clean Bathrooms AND a Vending Machine
--> postgres post -d amethysts -u 'root'
EXPLAIN ANALYZE
SELECT
  B.Name AS BuildingName,
  A.Type AS AmenityType,
  R.Rating AS R.RatingRating AS R.Rating,
  A.Address
FROM Building B
JOIN Amenity A ON B.BuildingId = A.BuildingId
JOIN Review R ON A.AmenityId = A.AmenityId
JOIN Address A_2 ON B.AddressId = A_2.AddressId
WHERE A.Type = 'Bathroom'
AND EXISTS (
  SELECT 1
  FROM Amenity A2
  WHERE A2.BuildingId = B.BuildingId
  AND A2.Type = 'VendingMachine'
)
GROUP BY B.Name, A.Type, A_2.Address
ORDER BY Avg_Bathroom_Rating DESC
LIMIT 15;
50;

Q3: Query 1: Top 15 Buildings for Clean Bathrooms AND a Vending Machine
--> postgres post -d amethysts -u 'root'
EXPLAIN ANALYZE
SELECT
  B.Name AS BuildingName,
  A.Type AS AmenityType,
  R.Rating AS R.RatingRating AS R.Rating,
  A.Address
FROM Building B
JOIN Amenity A ON B.BuildingId = A.BuildingId
JOIN Review R ON A.AmenityId = A.AmenityId
JOIN Address A_2 ON B.AddressId = A_2.AddressId
WHERE A.Type = 'Bathroom'
AND EXISTS (
  SELECT 1
  FROM Amenity A2
  WHERE A2.BuildingId = B.BuildingId
  AND A2.Type = 'VendingMachine'
)
GROUP BY B.Name, A.Type, A_2.Address
ORDER BY Avg_Bathroom_Rating DESC
LIMIT 15;
50;
```

Query 1 returns the buildings with the top-rated buildings and vending machines. The original query uses hash joins and aggregation making it the fastest query, costing 594. The first alternate query relies on direct joins and filters by amenity type, it costs 611 and is slightly slower than the original. The second alternate query uses a GROUP BY ... HAVING, it costs 893 which is far too high as it uses nested loops and group aggregations. The third query uses pre-aggregated bathroom ratings then joins with vending machine data, this only costs 599 because it reduces intermediate join sizes. We will be using the original query as it is the fastest and most resource efficient.

[illegible][illegible]

```

echo "QUERY 3: Users who Reviewed 3+ Poorly Rated Vending Machine Buildings"
sudo -n postgres psql -d amenities <<"SQL"
EXPLAIN ANALYZE
WITH Ratings AS (
    SELECT
        U.userId,
        U.userName,
        A.buildingId,
        AVG(R.OverallRating) OVER (PARTITION BY U.userId AS UserAvg,
        AVG(R.OverallRating) OVER (1) AS CampusAvg)
    FROM "R" R
    JOIN Reviewer U ON R.userId = U.userId
    JOIN Amenity A ON R.buildingId = A.buildingId
    WHERE A.type = 'vendingMachine'
)
SELECT
    A.buildingId,
    A.userName,
    COUNT(DISTINCT R.buildingId) AS Poor_Vending_Building_Count
FROM Ratings R
WHERE A.UserAvg < A.CampusAvg
GROUP BY R.userId, R.userName
HAVING COUNT(DISTINCT R.buildingId) >= 3
ORDER BY Poor_Vending_Building_Count DESC
LIMIT 15;

QUERY 3: Users who Reviewed 3+ Poorly Rated Vending Machine Buildings
QUERY PLAN
Limit (cost=33.81..335.80 rows=22 width=22) (actual time=2.887..3.853 rows=1)
-> Sort (cost=33.81..335.11 rows=22 width=22) (actual time=2.886..3.838 rows=5)
    Sort Key: (count(DISTINCT R.buildingId)) DESC
    Sort Method: top-N heapsort  Memory: 256kB
    -> GroupAggregate (cost=26.63..334.13 rows=22 width=22) (actual time=2.568..2.757 rows=1)
        Group Key: R.userId, R.userName
        Filter: (count(DISTINCT R.buildingId) >= 3)
        Hashed by Filter: 15
        -> IncreaseSortKey (cost=20.29..311.95 rows=18 width=24) (actual time=2.548..2.671 rows=17)
            Sort Key: R.userId, R.userName
            Presorted Key: count
            PartialMode: quicksort  Average Memory: 274B  Peak Memory: 274B
            -> Subquery Scan #1 (cost=29.15..387.85 rows=18 width=1) (actual time=2.484..2.534 rows=17)
                Filter: (count(R.userId) < campusAvg)
                Hashed by Filter: 17
                -> WindowAgg (cost=29.15..387.85 rows=17 width=2) (actual time=2.478..2.534 rows=17)
                    -> WindowAgg (cost=29.15..387.85 rows=17 width=2) (actual time=2.468..2.534 rows=17)
                        Sort Key: R.userId
                        Sort Method: quicksort  Memory: 512kB
                        -> Hash Join (cost=3.75..279.49 rows=17 width=24) (actual time=425.1..955 rows=17)
                            Hash Cond: (r.userId = u.userId)
                            Hash Count: 152 users < u.userName
                            -> Hash Join (cost=48.56..275.35 rows=17 width=14) (actual time=357..1.088 rows=17)
                                Hash Cond: (r.userId = u.userId)
                                -> Seq Scan on Amenity A (cost=0.00..213.78 rows=188 width=14) (actual time=484..1.122 rows=1)
                                    Filter: (type=text & "vendingMachine"=1)
                                    Hashed by Filter: 1653
                                    Buckets: 1824 Buckets: 1 Memory Usage: 256B
                                -> Hash (cost=28.86..213.78 rows=188 width=14) (actual time=334..8.335 rows=188)
                                    Buckets: 1824 Buckets: 1 Memory Usage: 512kB
                                    -> Hash (cost=0.00..213.78 rows=188 width=14) (actual time=852..8.652 rows=188)
                                        Buckets: 1824 Buckets: 1 Memory Usage: 512kB
                                        -> Seq Scan on "User" U (cost=0.00..2.68 rows=188 width=14) (actual time=878..8.822 rows=188)
                                            Filter: (type=text & "vendingMachine"=1)
                                            Hashed by Filter: 1653
                                            -> Hash (cost=28.86..213.78 rows=188 width=14) (actual time=356..6.251 rows=188)
                                                Buckets: 1824 Buckets: 1 Memory Usage: 512kB
                                                -> Seq Scan on Review R (cost=0.00..22.80 rows=188 width=14) (actual time=484..8.172 rows=188)
                                                    Filter: (type=text & "vendingMachine"=1)
                                                    Hashed by Filter: 1653
                                                    -> Hash (cost=0.00..22.80 rows=188 width=14) (actual time=356..6.251 rows=188)
                                                        Buckets: 1824 Buckets: 1 Memory Usage: 512kB
                                                        -> Seq Scan on Review R (cost=0.00..22.80 rows=188 width=14) (actual time=484..8.172 rows=188)
                                                            Filter: (type=text & "vendingMachine"=1)
                                                            Hashed by Filter: 1653
                                                            -> Hash (cost=0.00..22.80 rows=188 width=14) (actual time=356..6.251 rows=188)
                                                                Buckets: 1824 Buckets: 1 Memory Usage: 512kB
                                                                -> Seq Scan on Review R (cost=0.00..22.80 rows=188 width=14) (actual time=484..8.172 rows=188)
                                                                    Filter: (type=text & "vendingMachine"=1)
                                                                    Hashed by Filter: 1653
                                                                    -> Hash (cost=0.00..22.80 rows=188 width=14) (actual time=356..6.251 rows=188)
                                                                        Buckets: 1824 Buckets: 1 Memory Usage: 512kB
                                                                        -> Seq Scan on Review R (cost=0.00..22.80 rows=188 width=14) (actual time=484..8.172 rows=188)
                                                                            Filter: (type=text & "vendingMachine"=1)
                                                                            Hashed by Filter: 1653
                                                                            -> Hash (cost=0.00..22.80 rows=188 width=14) (actual time=356..6.251 rows=188)
                                                                                Buckets: 1824 Buckets: 1 Memory Usage: 512kB
                                                                                -> Seq Scan on Review R (cost=0.00..22.80 rows=188 width=14) (actual time=484..8.172 rows=188)
                                                                                    Filter: (type=text & "vendingMachine"=1)
                                                                                    Hashed by Filter: 1653
                                                                                    -> Hash (cost=0.00..22.80 rows=188 width=14) (actual time=356..6.251 rows=188)
                                                                                        Buckets: 1824 Buckets: 1 Memory Usage: 512kB
                                                                                        -> Seq Scan on Review R (cost=0.00..22.80 rows=188 width=14) (actual time=484..8.172 rows=188)
                                                                                            Filter: (type=text & "vendingMachine"=1)
                                                                                            Hashed by Filter: 1653
                                                                                            -> Hash (cost=0.00..22.80 rows=188 width=14) (actual time=356..6.251 rows=188)
                                                                                                Buckets: 1824 Buckets: 1 Memory Usage: 512kB
                                                                                                -> Seq Scan on Review R (cost=0.00..22.80 rows=188 width=14) (actual time=484..8.172 rows=188)
                                                                                                    Filter: (type=text & "vendingMachine"=1)
                                                                                                    Hashed by Filter: 1653
                                                                                                    -> Hash (cost=0.00..22.80 rows=188 width=14) (actual time=356..6.251 rows=188)
                                                                                                        Buckets: 1824 Buckets: 1 Memory Usage: 512kB
                                                                                                        -> Seq Scan on Review R (cost=0.00..22.80 rows=188 width=14) (actual time=484..8.172 rows=188)
                                                                                                            Filter: (type=text & "vendingMachine"=1)
                                                                                                            Hashed by Filter: 1653
                                                                                                            -> Hash (cost=0.00..22.80 rows=188 width=14) (actual time=356..6.251 rows=188)
                                                                                                                Buckets: 1824 Buckets: 1 Memory Usage: 512kB
                                                                                                                -> Seq Scan on Review R (cost=0.00..22.80 rows=188 width=14) (actual time=484..8.172 rows=188)
                                                                                                                    Filter: (type=text & "vendingMachine"=1)
                                                                                                                    Hashed by Filter: 1653
                                                                                                                    -> Hash (cost=0.00..22.80 rows=188 width=14) (actual time=356..6.251 rows=188)
                                                                                                                        Buckets: 1824 Buckets: 1 Memory Usage: 512kB
                                                                                                                        -> Seq Scan on Review R (cost=0.00..22.80 rows=188 width=14) (actual time=484..8.172 rows=188)
                                                                                                                            Filter: (type=text & "vendingMachine"=1)
                                                                                                                            Hashed by Filter: 1653
                                                                                                                            -> Hash (cost=0.00..22.80 rows=188 width=14) (actual time=356..6.251 rows=188)
                                                                                                                                Buckets: 1824 Buckets: 1 Memory Usage: 512kB
                                                                                                                                -> Seq Scan on Review R (cost=0.00..22.80 rows=188 width=14) (actual time=484..8.172 rows=188)
                                                                                                                                    Filter: (type=text & "vendingMachine"=1)
                                                                                                                                    Hashed by Filter: 1653
                                                                                                                                    -> Hash (cost=0.00..22.80 rows=188 width=14) (actual time=356..6.251 rows=188)
                                                                                                                                        Buckets: 1824 Buckets: 1 Memory Usage: 512kB
                                                                                                                                        -> Seq Scan on Review R (cost=0.00..22.80 rows=188 width=14) (actual time=484..8.172 rows=188)
                                                                                                                                            Filter: (type=text & "vendingMachine"=1)
                                                                                                                                            Hashed by Filter: 1653
                                                                                                                                            -> Hash (cost=0.00..22.80 rows=188 width=14) (actual time=356..6.251 rows=188)
                                                                                                                                                Buckets: 1824 Buckets: 1 Memory Usage: 512kB
                                                                                                                                                -> Seq Scan on Review R (cost=0.00..22.80 rows=188 width=14) (actual time=484..8.172 rows=188)
                                                                                                                                                    Filter: (type=text & "vendingMachine"=1)
                                                                                                                                                    Hashed by Filter: 1653
                                                                                                                                                    -> Hash (cost=0.00..22.80 rows=188 width=14) (actual time=356..6.251 rows=188)
                                                                                                                                                        Buckets: 1824 Buckets: 1 Memory Usage: 512kB
                                                                                                                                                        -> Seq Scan on Review R (cost=0.00..22.80 rows=188 width=14) (actual time=484..8.172 rows=188)
                                                                                                                                                            Filter: (type=text & "vendingMachine"=1)
                                                                                                                                                            Hashed by Filter: 1653
                                                                                                                                                            -> Hash (cost=0.00..22.80 rows=188 width=14) (actual time=356..6.251 rows=188)
                                                                                                                                                                Buckets: 1824 Buckets: 1 Memory Usage: 512kB
                                                                                                                                                                -> Seq Scan on Review R (cost=0.00..22.80 rows=188 width=14) (actual time=484..8.172 rows=188)
                                                                                                                                                                    Filter: (type=text & "vendingMachine"=1)
                                                                                                                                                                    Hashed by Filter: 1653
                                                                                                                                                                    -> Hash (cost=0.00..22.80 rows=188 width=14) (actual time=356..6.251 rows=188)
                                                                                                                                                                        Buckets: 1824 Buckets: 1 Memory Usage: 512kB
                                                                                                                                                                        -> Seq Scan on Review R (cost=0.00..22.80 rows=188 width=14) (actual time=484..8.172 rows=188)
                                                                                                                                                                            Filter: (type=text & "vendingMachine"=1)
                                                                                                                                                                            Hashed by Filter: 1653
                                                                                                                                                                            -> Hash (cost=0.00..22.80 rows=188 width=14) (actual time=356..6.251 rows=188)
                                                                                                                                                                                Buckets: 1824 Buckets: 1 Memory Usage: 512kB
                                                                                                                                                                                -> Seq Scan on Review R (cost=0.00..22.80 rows=188 width=14) (actual time=484..8.172 rows=188)
                                                                                                                                                                                    Filter: (type=text & "vendingMachine"=1)
                                                                                                                                                                                    Hashed by Filter: 1653
                                                                                                                                                                                    -> Hash (cost=0.00..22.80 rows=188 width=14) (actual time=356..6.251 rows=188)
                                                                                                                                                                                        Buckets: 1824 Buckets: 1 Memory Usage: 512kB
                                                                                                                                                                                        -> Seq Scan on Review R (cost=0.00..22.80 rows=188 width=14) (actual time=484..8.172
```

Query 3 returns all users who reviewed at least 3 poorly rated vending machine buildings. The original code uses a subquery to find these users by comparing their vending machine ratings to the university average and has a cost of 886. The first alternate subquery uses a common table expression to join the items; it costs 888 and is more modular but slower and less resource efficient. The second alternate subquery uses a derived table instead of the subquery, it also replaces IN with a join to optimize. It costs 870, which is slightly faster than 886 but the code is much more complex. The third alternate function uses a window function instead of a subquery. It only costs 315 as it can find the campus average and summation of a user's bad reviews in a single pass, this is by far the most efficient version and will be used going forward

[illegible]

```

-- Query 4 (Single): Overall Amenity Ranking by Rating (Top 15)
-- Code --
--> query4 <- dbGetQuery(con, "
  SELECT
    B.Name AS Building_Name,
    A.Type,
    A.Floor,
    ROUND(CAST(AVG(R.OverallRating) AS NUMERIC), 2) AS Avg_Rating,
    COUNT(R.ReviewId) AS Review_Count
  FROM Building B
  JOIN Amenity A ON B.BuildingId = A.BuildingId
  JOIN Review R ON A.AmenityId = R.AmenityId
  GROUP BY B.Name, A.Type, A.Floor
  ORDER BY Avg_Rating DESC, Review_Count DESC
  LIMIT 15;
")
-- SQL --

-- Query 4 (Single): Overall Amenity Ranking by Rating (Top 15)
-- QUERY PLAN
--
-- List (cost=388.63..388.67 rows=15 width=86) (actual time=4.911..4.916 rows=15 loops=1)
--> Sort (cost=388.63..391.13 rows=1880 width=86) (actual time=4.818..4.833 rows=15 loops=1)
--> Sort Key: (round(avg(overallrating), 2)) DESC, (count(reviewid)) DESC
-- Sort Method: topological sorting  (actual time=0.284..0.285 rows=1880)
--> HashAggregate (cost=348.19..364.18 rows=1880 width=86) (actual time=3.938..4.578 rows=927 loops=1)
--> Group Key: B.name, A.type, A.floor
--> HashJoin 1: Memory Usage: 3736
--> Hash Join (cost=77.56..335.68 rows=1880 width=56) (actual time=0.818..3.888 rows=1880 loops=1)
--> Hash Cond: (A.buildingid = B.buildingid)
--> Hash Cond: (A.amenityid = R.amenityid)
--> Hash Join (cost=149.56..296.93 rows=1880 width=28) (actual time=0.396..2.431 rows=1880 loops=1)
--> Sort Key on amenity_a (cost=0.00..192.40 rows=782 width=24) (actual time=0.402..0.721 rows=782 loops=1)
--> Hash (cost=0.00..0.40 rows=1880 width=16) (actual time=0.276..0.377 rows=1880 loops=1)
--> Buckets: 1824 Batches: 1 Memory Usage: 5548
--> Hash Join on review_r (cost=0.00..20.48 rows=1880 width=16) (actual time=0.004..0.163 rows=1880 loops=1)
--> Hash (cost=22.22..22.58 rows=1156 width=36) (actual time=0.406..0.488 rows=1156 loops=1)
--> Buckets: 2448 Batches: 1 Memory Usage: 8168
--> Sort Key on building_b (cost=0.00..22.58 rows=1156 width=36) (actual time=0.008..0.183 rows=1156 loops=1)
--> Hash (cost=0.00..0.40 rows=1880 width=16) (actual time=0.276..0.377 rows=1880 loops=1)
--> Buckets: 1824 Batches: 1 Memory Usage: 5548
--> Sort Key on amenity_a (cost=0.00..192.40 rows=782 width=24) (actual time=0.402..0.721 rows=782 loops=1)
--> Hash (cost=0.00..0.40 rows=1880 width=16) (actual time=0.276..0.377 rows=1880 loops=1)
--> Buckets: 1824 Batches: 1 Memory Usage: 5548
--> Hash Join on review_r (cost=0.00..20.48 rows=1880 width=16) (actual time=0.004..0.163 rows=1880 loops=1)
--> Hash (cost=22.22..22.58 rows=1156 width=36) (actual time=0.406..0.488 rows=1156 loops=1)
--> Buckets: 2448 Batches: 1 Memory Usage: 8168
--> Sort Key on building_b (cost=0.00..22.58 rows=1156 width=36) (actual time=0.008..0.183 rows=1156 loops=1)
--> Hash (cost=0.00..0.40 rows=1880 width=16) (actual time=0.276..0.377 rows=1880 loops=1)
--> Buckets: 1824 Batches: 1 Memory Usage: 5548
--> Sort Key on amenity_a (cost=0.00..192.40 rows=782 width=24) (actual time=0.402..0.721 rows=782 loops=1)
--> Hash (cost=0.00..0.40 rows=1880 width=16) (actual time=0.276..0.377 rows=1880 loops=1)
--> Buckets: 1824 Batches: 1 Memory Usage: 5548
--> Hash Join on review_r (cost=0.00..20.48 rows=1880 width=16) (actual time=0.004..0.163 rows=1880 loops=1)
--> Hash (cost=22.22..22.58 rows=1156 width=36) (actual time=0.406..0.488 rows=1156 loops=1)
--> Buckets: 2448 Batches: 1 Memory Usage: 8168
--> Sort Key on building_b (cost=0.00..22.58 rows=1156 width=36) (actual time=0.008..0.183 rows=1156 loops=1)
--> Hash (cost=0.00..0.40 rows=1880 width=16) (actual time=0.276..0.377 rows=1880 loops=1)
--> Buckets: 1824 Batches: 1 Memory Usage: 5548
--> Sort Key on amenity_a (cost=0.00..192.40 rows=782 width=24) (actual time=0.402..0.721 rows=782 loops=1)
--> Hash (cost=0.00..0.40 rows=1880 width=16) (actual time=0.276..0.377 rows=1880 loops=1)
--> Buckets: 1824 Batches: 1 Memory Usage: 5548
--> Hash Join on review_r (cost=0.00..20.48 rows=1880 width=16) (actual time=0.004..0.163 rows=1880 loops=1)
--> Hash (cost=22.22..22.58 rows=1156 width=36) (actual time=0.406..0.488 rows=1156 loops=1)
--> Buckets: 2448 Batches: 1 Memory Usage: 8168
--> Sort Key on building_b (cost=0.00..22.58 rows=1156 width=36) (actual time=0.008..0.183 rows=1156 loops=1)
--> Hash (cost=0.00..0.40 rows=1880 width=16) (actual time=0.276..0.377 rows=1880 loops=1)
--> Buckets: 1824 Batches: 1 Memory Usage: 5548
--> Sort Key on amenity_a (cost=0.00..192.40 rows=782 width=24) (actual time=0.402..0.721 rows=782 loops=1)
--> Hash (cost=0.00..0.40 rows=1880 width=16) (actual time=0.276..0.377 rows=1880 loops=1)
--> Buckets: 1824 Batches: 1 Memory Usage: 5548
--> Hash Join on review_r (cost=0.00..20.48 rows=1880 width=16) (actual time=0.004..0.163 rows=1880 loops=1)
--> Hash (cost=22.22..22.58 rows=1156 width=36) (actual time=0.406..0.488 rows=1156 loops=1)
--> Buckets: 2448 Batches: 1 Memory Usage: 8168
--> Sort Key on building_b (cost=0.00..22.58 rows=1156 width=36) (actual time=0.008..0.183 rows=1156 loops=1)
--> Hash (cost=0.00..0.40 rows=1880 width=16) (actual time=0.276..0.377 rows=1880 loops=1)
--> Buckets: 1824 Batches: 1 Memory Usage: 5548
--> Sort Key on amenity_a (cost=0.00..192.40 rows=782 width=24) (actual time=0.402..0.721 rows=782 loops=1)
--> Hash (cost=0.00..0.40 rows=1880 width=16) (actual time=0.276..0.377 rows=1880 loops=1)
--> Buckets: 1824 Batches: 1 Memory Usage: 5548
--> Hash Join on review_r (cost=0.00..20.48 rows=1880 width=16) (actual time=0.004..0.163 rows=1880 loops=1)
--> Hash (cost=22.22..22.58 rows=1156 width=36) (actual time=0.406..0.488 rows=1156 loops=1)
--> Buckets: 2448 Batches: 1 Memory Usage: 8168
--> Sort Key on building_b (cost=0.00..22.58 rows=1156 width=36) (actual time=0.008..0.183 rows=1156 loops=1)
--> Hash (cost=0.00..0.40 rows=1880 width=16) (actual time=0.276..0.377 rows=1880 loops=1)
--> Buckets: 1824 Batches: 1 Memory Usage: 5548
--> Sort Key on amenity_a (cost=0.00..192.40 rows=782 width=24) (actual time=0.402..0.721 rows=782 loops=1)
--> Hash (cost=0.00..0.40 rows=1880 width=16) (actual time=0.276..0.377 rows=1880 loops=1)
--> Buckets: 1824 Batches: 1 Memory Usage: 5548
--> Hash Join on review_r (cost=0.00..20.48 rows=1880 width=16) (actual time=0.004..0.163 rows=1880 loops=1)
--> Hash (cost=22.22..22.58 rows=1156 width=36) (actual time=0.406..0.488 rows=1156 loops=1)
--> Buckets: 2448 Batches: 1 Memory Usage: 8168
--> Sort Key on building_b (cost=0.00..22.58 rows=1156 width=36) (actual time=0.008..0.183 rows=1156 loops=1)
--> Hash (cost=0.00..0.40 rows=1880 width=16) (actual time=0.276..0.377 rows=1880 loops=1)
--> Buckets: 1824 Batches: 1 Memory Usage: 5548
--> Sort Key on amenity_a (cost=0.00..192.40 rows=782 width=24) (actual time=0.402..0.721 rows=782 loops=1)
--> Hash (cost=0.00..0.40 rows=1880 width=16) (actual time=0.276..0.377 rows=1880 loops=1)
--> Buckets: 1824 Batches: 1 Memory Usage: 5548
--> Hash Join on review_r (cost=0.00..20.48 rows=1880 width=16) (actual time=0.004..0.163 rows=1880 loops=1)
--> Hash (cost=22.22..22.58 rows=1156 width=36) (actual time=0.406..0.488 rows=1156 loops=1)
--> Buckets: 2448 Batches: 1 Memory Usage: 8168
--> Sort Key on building_b (cost=0.00..22.58 rows=1156 width=36) (actual time=0.008..0.183 rows=1156 loops=1)
--> Hash (cost=0.00..0.40 rows=1880 width=16) (actual time=0.276..0.377 rows=1880 loops=1)
--> Buckets: 1824 Batches: 1 Memory Usage: 5548
--> Sort Key on amenity_a (cost=0.00..192.40 rows=782 width=24) (actual time=0.402..0.721 rows=782 loops=1)
--> Hash (cost=0.00..0.40 rows=1880 width=16) (actual time=0.276..0.377 rows=1880 loops=1)
--> Buckets: 1824 Batches: 
```

Query 4 returns the overall amenity ranking. It is an important query as it is a key feature for our project. The original query uses only groups and joins and costs 388. The first alternate query uses common table expressions. Because the original query is so simple, it actually slows down the query a lot to use a CTE, as this function costs 621. The second alternate query uses a derived table and joins on AmenityID. It is also way slower and more costly, costing 621, most likely due to the extra overhead of making the derived table. The third alternate query uses a window function, this function is also much slower than the original, costing 463. Because the original function is the simplest and costs the least, it will be our implementation going forward.