v3 Indexing Improvements
Cole Robinson, Geoff Blaylock

Below are all of our endpoints and their plans using “EXPLAIN ANALYZE” on them. After creating the indexes, we ran the “EXPLAIN ANALYZE” function on them to view the changes (highlighted in green).

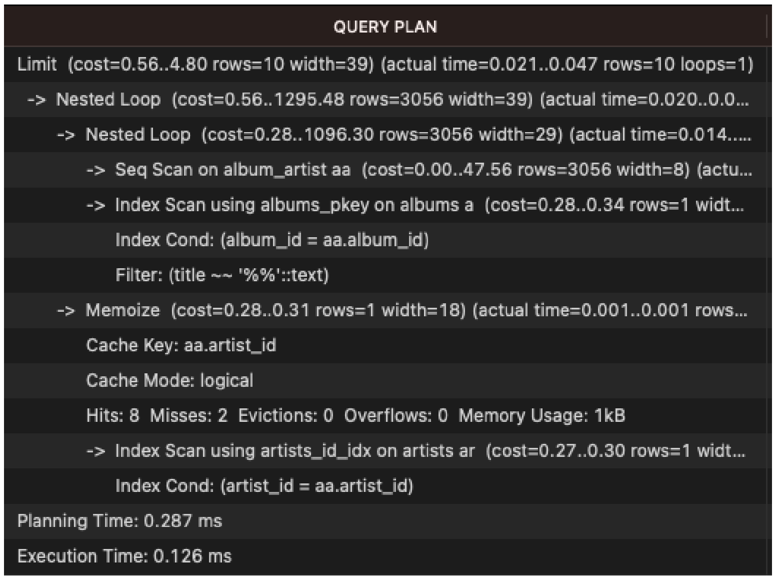
We chose these columns as indexes because they were most commonly found in the GROUP BY and WHERE clauses in our endpoint queries in an effort to maximize performance with minimum memory usage.

SQL Used to Generate Indexes:

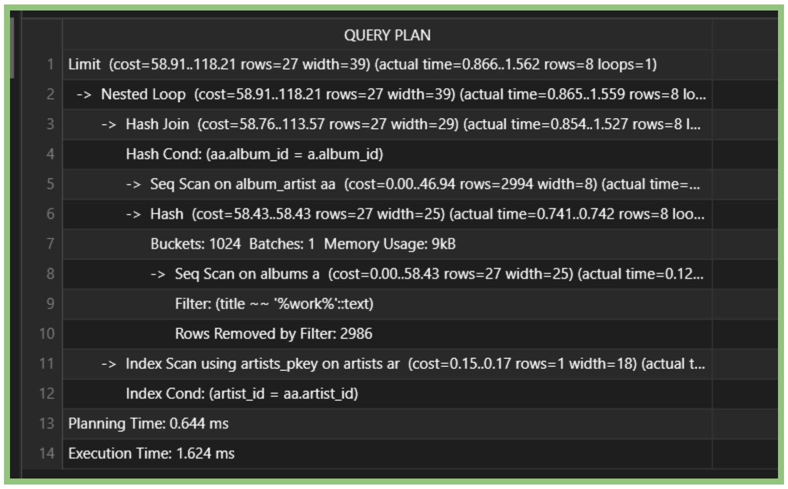
```
CREATE INDEX idx_tracks_title ON tracks(title);
CREATE INDEX idx_tracks_genre ON tracks(genre);
CREATE INDEX idx_tracks_album_id ON tracks(album_id);
CREATE INDEX idx_tracks_release_date ON tracks(release_date);
CREATE INDEX idx_album_artist_album_id ON album_artist(album_id);
CREATE INDEX idx_album_artist_artist_id ON album_artist(artist_id);
CREATE INDEX idx_playlists_user_id ON playlists(user_id);
CREATE INDEX idx_playlist_track_playlist_id ON playlist_track(playlist_id);
CREATE INDEX idx_playlist_track_track_id ON playlist_track(track_id);
CREATE INDEX idx_track_artist_track_id ON track_artist(track_id);
CREATE INDEX idx_track_artist_artist_id ON track_artist(artist_id);
```

Albums

/albums/

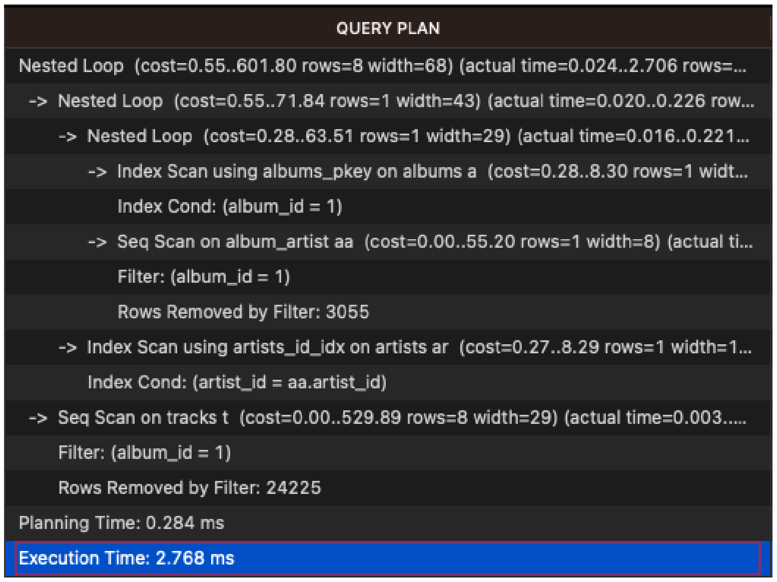


```
QUERY PLAN
Limit  (cost=0.56..4.80 rows=10 width=39) (actual time=0.021..0.047 rows=10 loops=1)
  -> Nested Loop  (cost=0.56..1295.48 rows=3056 width=39) (actual time=0.020..0.0...
      -> Nested Loop  (cost=0.28..1096.30 rows=3056 width=29) (actual time=0.014.....
          -> Seq Scan on album_artist aa  (cost=0.00..47.56 rows=3056 width=8) (actu...
          -> Index Scan using albums_pkey on albums a  (cost=0.28..0.34 rows=1 widt...
              Index Cond: (album_id = aa.album_id)
              Filter: (title ~~ '%%'::text)
      -> Memoize  (cost=0.28..0.31 rows=1 width=18) (actual time=0.001..0.001 rows...
          Cache Key: aa.artist_id
          Cache Mode: logical
          Hits: 8  Misses: 2  Evictions: 0  Overflows: 0  Memory Usage: 1kB
          -> Index Scan using artists_id_idx on artists ar  (cost=0.27..0.30 rows=1 widt...
              Index Cond: (artist_id = aa.artist_id)
Planning Time: 0.287 ms
Execution Time: 0.126 ms
```

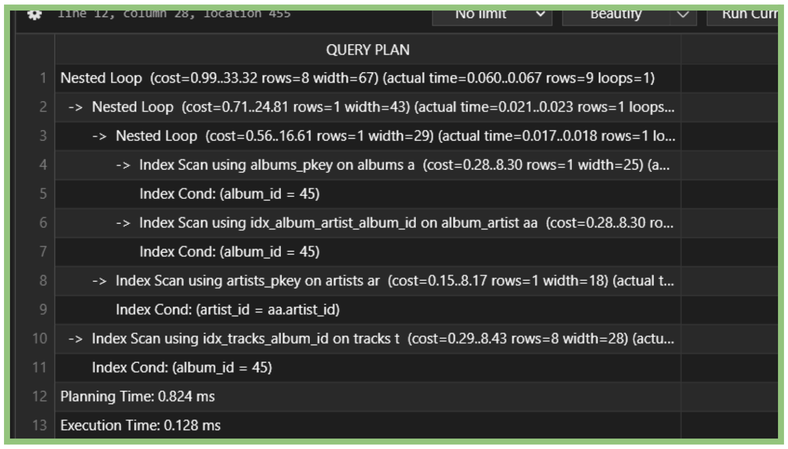


```
QUERY PLAN
Limit  (cost=58.91..118.21 rows=27 width=39) (actual time=0.866..1.562 rows=8 loops=1)
  ->  Nested Loop  (cost=58.91..118.21 rows=27 width=39) (actual time=0.865..1.559 rows=8 lo...
        ->  Hash Join  (cost=58.76..113.57 rows=27 width=29) (actual time=0.854..1.527 rows=8 l...
              Hash Cond: (aa.album_id = a.album_id)
              ->  Seq Scan on album_artist aa  (cost=0.00..46.94 rows=2994 width=8) (actual time=...
              ->  Hash  (cost=58.43..58.43 rows=27 width=25) (actual time=0.741..0.742 rows=8 loo...
                    Buckets: 1024  Batches: 1  Memory Usage: 9kB
                    ->  Seq Scan on albums a  (cost=0.00..58.43 rows=27 width=25) (actual time=0.12...
                          Filter: (title ~~ '%work%'::text)
                          Rows Removed by Filter: 2986
        ->  Index Scan using artists_pkey on artists ar  (cost=0.15..0.17 rows=1 width=18) (actual t...
              Index Cond: (artist_id = aa.artist_id)
Planning Time: 0.644 ms
Execution Time: 1.624 ms
```

/albums/{album_id}

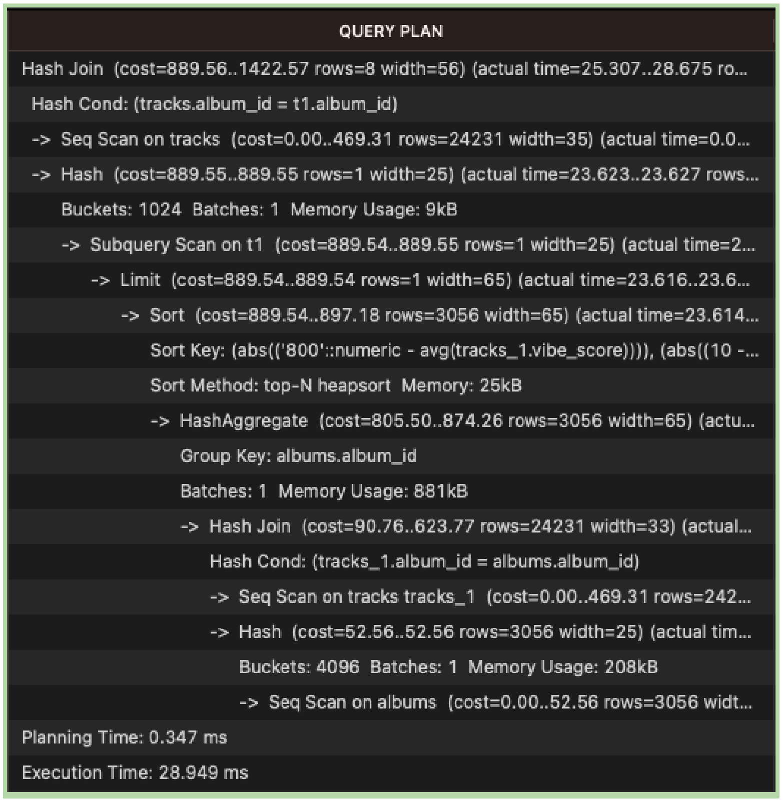


```
QUERY PLAN
Nested Loop  (cost=0.55..601.80 rows=8 width=68) (actual time=0.024..2.706 rows=...
  ->  Nested Loop  (cost=0.55..71.84 rows=1 width=43) (actual time=0.020..0.226 row...
        ->  Nested Loop  (cost=0.28..63.51 rows=1 width=29) (actual time=0.016..0.221...
              ->  Index Scan using albums_pkey on albums a  (cost=0.28..8.30 rows=1 widt...
                    Index Cond: (album_id = 1)
              ->  Seq Scan on album_artist aa  (cost=0.00..55.20 rows=1 width=8) (actual ti...
                    Filter: (album_id = 1)
                    Rows Removed by Filter: 3055
        ->  Index Scan using artists_id_idx on artists ar  (cost=0.27..8.29 rows=1 width=1...
              Index Cond: (artist_id = aa.artist_id)
  ->  Seq Scan on tracks t  (cost=0.00..529.89 rows=8 width=29) (actual time=0.003.....
        Filter: (album_id = 1)
        Rows Removed by Filter: 24225
Planning Time: 0.284 ms
Execution Time: 2.768 ms
```

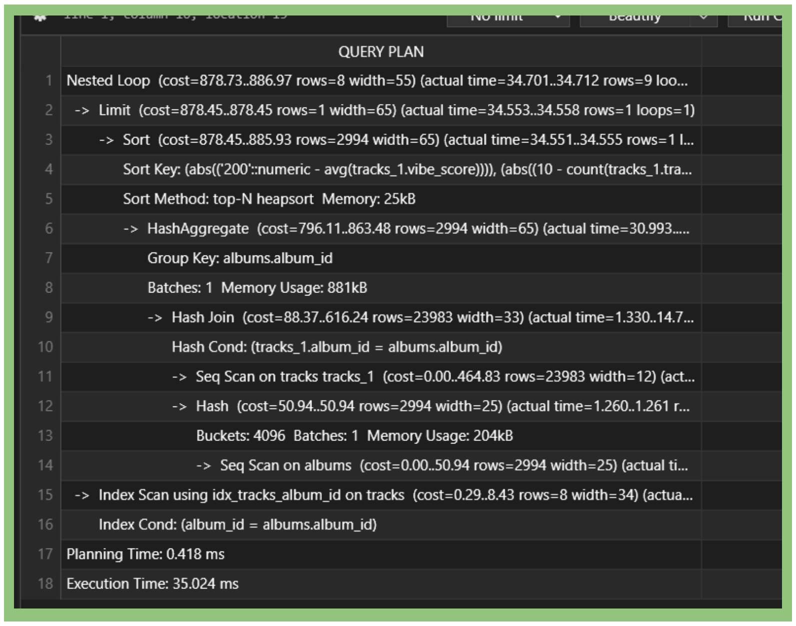


```
QUERY PLAN
Nested Loop  (cost=0.99..33.32 rows=8 width=67) (actual time=0.060..0.067 rows=9 loops=1)
  ->  Nested Loop  (cost=0.71..24.81 rows=1 width=43) (actual time=0.021..0.023 rows=1 loops...
        ->  Nested Loop  (cost=0.56..16.61 rows=1 width=29) (actual time=0.017..0.018 rows=1 lo...
              ->  Index Scan using albums_pkey on albums a  (cost=0.28..8.30 rows=1 width=25) (a...
                    Index Cond: (album_id = 45)
              ->  Index Scan using idx_album_artist_album_id on album_artist aa  (cost=0.28..8.30 ro...
                    Index Cond: (album_id = 45)
        ->  Index Scan using artists_pkey on artists ar  (cost=0.15..8.17 rows=1 width=18) (actual t...
              Index Cond: (artist_id = aa.artist_id)
  ->  Index Scan using idx_tracks_album_id on tracks t  (cost=0.29..8.43 rows=8 width=28) (actu...
        Index Cond: (album_id = 45)
Planning Time: 0.824 ms
Execution Time: 0.128 ms
```

/recommend/



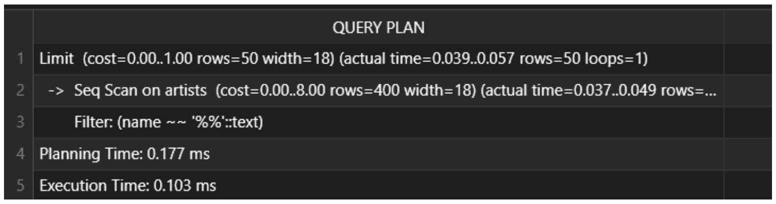
```
QUERY PLAN
Hash Join  (cost=889.56..1422.57 rows=8 width=56) (actual time=25.307..28.675 ro...
  Hash Cond: (tracks.album_id = t1.album_id)
  ->  Seq Scan on tracks  (cost=0.00..469.31 rows=24231 width=35) (actual time=0.0...
  ->  Hash  (cost=889.55..889.55 rows=1 width=25) (actual time=23.623..23.627 rows...
        Buckets: 1024  Batches: 1  Memory Usage: 9kB
        ->  Subquery Scan on t1  (cost=889.54..889.55 rows=1 width=25) (actual time=2...
              ->  Limit  (cost=889.54..889.54 rows=1 width=65) (actual time=23.616..23.6...
                    ->  Sort  (cost=889.54..897.18 rows=3056 width=65) (actual time=23.614...
                          Sort Key: (abs(('800'::numeric - avg(tracks_1.vibe_score)))), (abs((10 -...
                          Sort Method: top-N heapsort  Memory: 25kB
                          ->  HashAggregate  (cost=805.50..874.26 rows=3056 width=65) (actu...
                                Group Key: albums.album_id
                                Batches: 1  Memory Usage: 881kB
                                ->  Hash Join  (cost=90.76..623.77 rows=24231 width=33) (actual...
                                      Hash Cond: (tracks_1.album_id = albums.album_id)
                                      ->  Seq Scan on tracks tracks_1  (cost=0.00..469.31 rows=242...
                                      ->  Hash  (cost=52.56..52.56 rows=3056 width=25) (actual tim...
                                            Buckets: 4096  Batches: 1  Memory Usage: 208kB
                                            ->  Seq Scan on albums  (cost=0.00..52.56 rows=3056 widt...
Planning Time: 0.347 ms
Execution Time: 28.949 ms
```



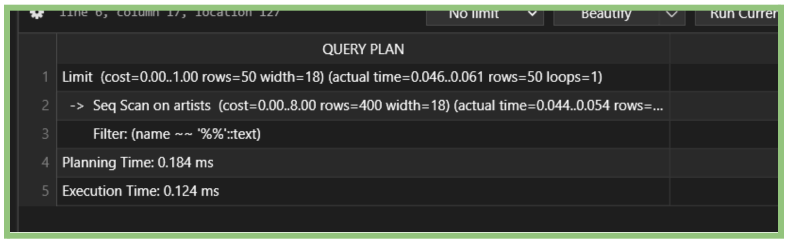
```
QUERY PLAN
Nested Loop  (cost=878.73..886.97 rows=8 width=55) (actual time=34.701..34.712 rows=9 loo...
  -> Limit  (cost=878.45..878.45 rows=1 width=65) (actual time=34.553..34.558 rows=1 loops=1)
        -> Sort  (cost=878.45..885.93 rows=2994 width=65) (actual time=34.551..34.555 rows=1 l...
              Sort Key: (abs(('200'::numeric - avg(tracks_1.vibe_score)))), (abs((10 - count(tracks_1.tra...
              Sort Method: top-N heapsort  Memory: 25kB
              -> HashAggregate  (cost=796.11..863.48 rows=2994 width=65) (actual time=30.993.....
                    Group Key: albums.album_id
                    Batches: 1  Memory Usage: 881kB
                    -> Hash Join  (cost=88.37..616.24 rows=23983 width=33) (actual time=1.330..14.7...
                          Hash Cond: (tracks_1.album_id = albums.album_id)
                          -> Seq Scan on tracks tracks_1  (cost=0.00..464.83 rows=23983 width=12) (act...
                          -> Hash  (cost=50.94..50.94 rows=2994 width=25) (actual time=1.260..1.261 r...
                                Buckets: 4096  Batches: 1  Memory Usage: 204kB
                                -> Seq Scan on albums  (cost=0.00..50.94 rows=2994 width=25) (actual ti...
  -> Index Scan using idx_tracks_album_id on tracks  (cost=0.29..8.43 rows=8 width=34) (actua...
        Index Cond: (album_id = albums.album_id)
Planning Time: 0.418 ms
Execution Time: 35.024 ms
```

Artists

/artists/

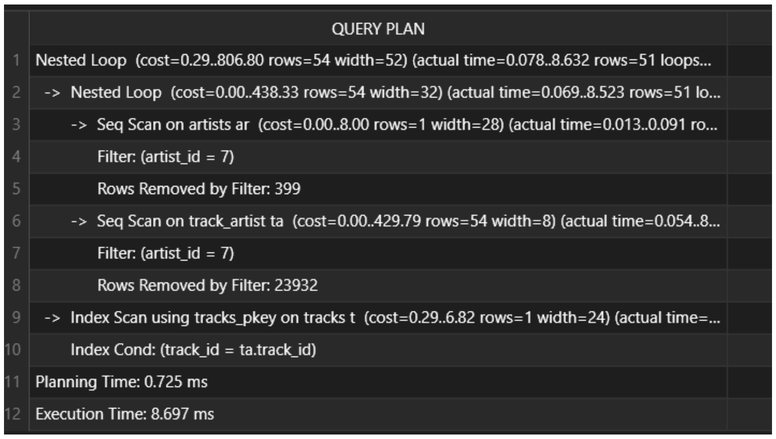


```
QUERY PLAN
Limit  (cost=0.00..1.00 rows=50 width=18) (actual time=0.039..0.057 rows=50 loops=1)
  -> Seq Scan on artists  (cost=0.00..8.00 rows=400 width=18) (actual time=0.037..0.049 rows=...
        Filter: (name ~~ '%%'::text)
Planning Time: 0.177 ms
Execution Time: 0.103 ms
```

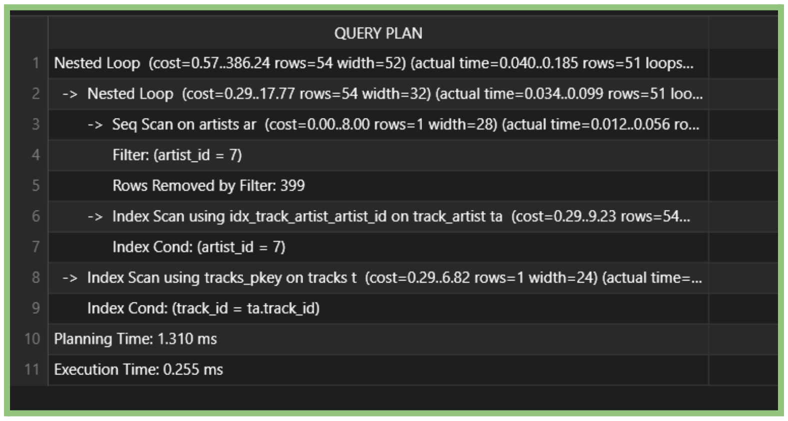


```
QUERY PLAN
Limit  (cost=0.00..1.00 rows=50 width=18) (actual time=0.046..0.061 rows=50 loops=1)
  -> Seq Scan on artists  (cost=0.00..8.00 rows=400 width=18) (actual time=0.044..0.054 rows=...
        Filter: (name ~~ '%%'::text)
Planning Time: 0.184 ms
Execution Time: 0.124 ms
```

/artists/{artist_id}



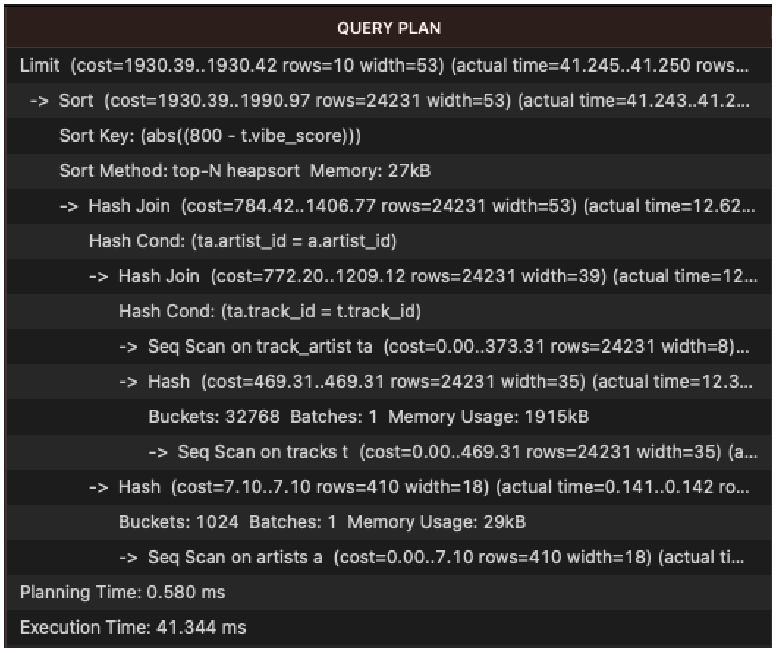
```
QUERY PLAN
Nested Loop  (cost=0.29..806.80 rows=54 width=52) (actual time=0.078..8.632 rows=51 loops...
  -> Nested Loop  (cost=0.00..438.33 rows=54 width=32) (actual time=0.069..8.523 rows=51 lo...
        -> Seq Scan on artists ar  (cost=0.00..8.00 rows=1 width=28) (actual time=0.013..0.091 ro...
              Filter: (artist_id = 7)
              Rows Removed by Filter: 399
        -> Seq Scan on track_artist ta  (cost=0.00..429.79 rows=54 width=8) (actual time=0.054..8...
              Filter: (artist_id = 7)
              Rows Removed by Filter: 23932
  -> Index Scan using tracks_pkey on tracks t  (cost=0.29..6.82 rows=1 width=24) (actual time=...
        Index Cond: (track_id = ta.track_id)
Planning Time: 0.725 ms
Execution Time: 8.697 ms
```



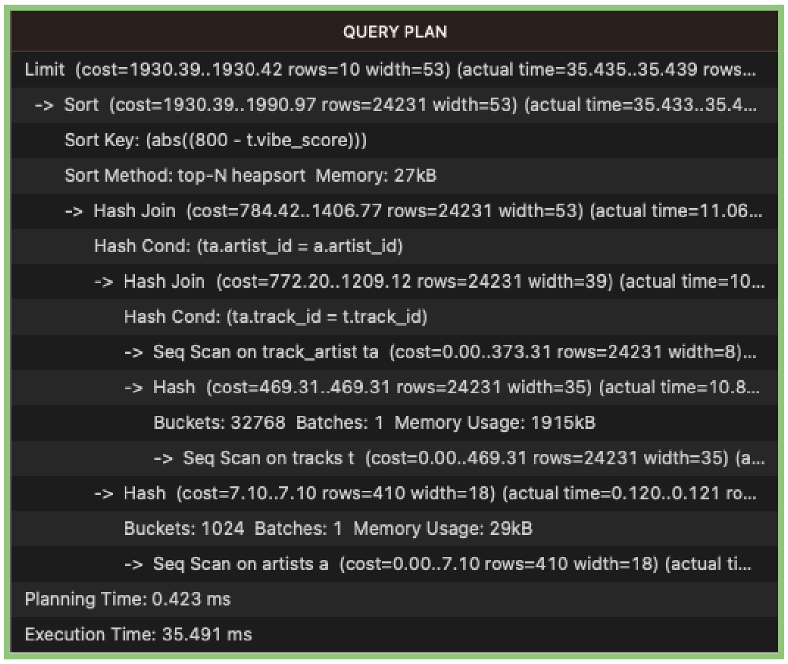
```
QUERY PLAN
Nested Loop  (cost=0.57..386.24 rows=54 width=52) (actual time=0.040..0.185 rows=51 loops...
  -> Nested Loop  (cost=0.29..17.77 rows=54 width=32) (actual time=0.034..0.099 rows=51 loo...
        -> Seq Scan on artists ar  (cost=0.00..8.00 rows=1 width=28) (actual time=0.012..0.056 ro...
              Filter: (artist_id = 7)
              Rows Removed by Filter: 399
        -> Index Scan using idx_track_artist_artist_id on track_artist ta  (cost=0.29..9.23 rows=54...
              Index Cond: (artist_id = 7)
  -> Index Scan using tracks_pkey on tracks t  (cost=0.29..6.82 rows=1 width=24) (actual time=...
        Index Cond: (track_id = ta.track_id)
Planning Time: 1.310 ms
Execution Time: 0.255 ms
```

Playlists

/playlists/generate

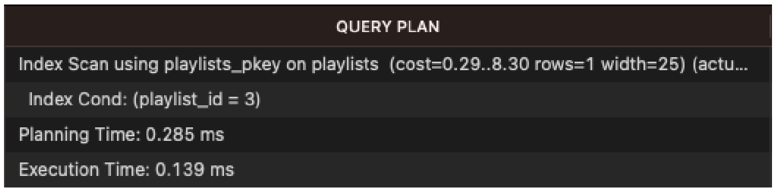


```
QUERY PLAN
Limit  (cost=1930.39..1930.42 rows=10 width=53) (actual time=41.245..41.250 rows...
  -> Sort  (cost=1930.39..1990.97 rows=24231 width=53) (actual time=41.243..41.2...
        Sort Key: (abs((800 - t.vibe_score)))
        Sort Method: top-N heapsort  Memory: 27kB
        -> Hash Join  (cost=784.42..1406.77 rows=24231 width=53) (actual time=12.62...
              Hash Cond: (ta.artist_id = a.artist_id)
              -> Hash Join  (cost=772.20..1209.12 rows=24231 width=39) (actual time=12...
                    Hash Cond: (ta.track_id = t.track_id)
                    -> Seq Scan on track_artist ta  (cost=0.00..373.31 rows=24231 width=8)...
                    -> Hash  (cost=469.31..469.31 rows=24231 width=35) (actual time=12.3...
                          Buckets: 32768  Batches: 1  Memory Usage: 1915kB
                          -> Seq Scan on tracks t  (cost=0.00..469.31 rows=24231 width=35) (a...
              -> Hash  (cost=7.10..7.10 rows=410 width=18) (actual time=0.141..0.142 ro...
                    Buckets: 1024  Batches: 1  Memory Usage: 29kB
                    -> Seq Scan on artists a  (cost=0.00..7.10 rows=410 width=18) (actual ti...
Planning Time: 0.580 ms
Execution Time: 41.344 ms
```

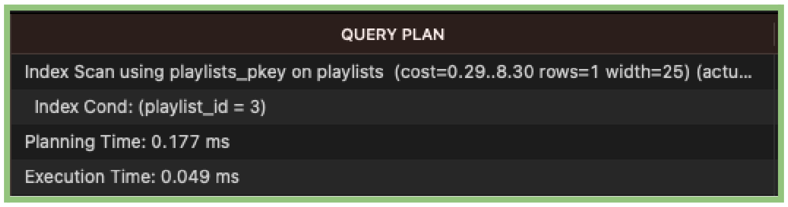


```
QUERY PLAN
Limit  (cost=1930.39..1930.42 rows=10 width=53) (actual time=35.435..35.439 rows...
  -> Sort  (cost=1930.39..1990.97 rows=24231 width=53) (actual time=35.433..35.4...
        Sort Key: (abs((800 - t.vibe_score)))
        Sort Method: top-N heapsort  Memory: 27kB
        -> Hash Join  (cost=784.42..1406.77 rows=24231 width=53) (actual time=11.06...
              Hash Cond: (ta.artist_id = a.artist_id)
              -> Hash Join  (cost=772.20..1209.12 rows=24231 width=39) (actual time=10...
                    Hash Cond: (ta.track_id = t.track_id)
                    -> Seq Scan on track_artist ta  (cost=0.00..373.31 rows=24231 width=8)...
                    -> Hash  (cost=469.31..469.31 rows=24231 width=35) (actual time=10.8...
                          Buckets: 32768  Batches: 1  Memory Usage: 1915kB
                          -> Seq Scan on tracks t  (cost=0.00..469.31 rows=24231 width=35) (a...
              -> Hash  (cost=7.10..7.10 rows=410 width=18) (actual time=0.120..0.121 ro...
                    Buckets: 1024  Batches: 1  Memory Usage: 29kB
                    -> Seq Scan on artists a  (cost=0.00..7.10 rows=410 width=18) (actual ti...
Planning Time: 0.423 ms
Execution Time: 35.491 ms
```

/playlists/{playlist_id}/track/{track_id}

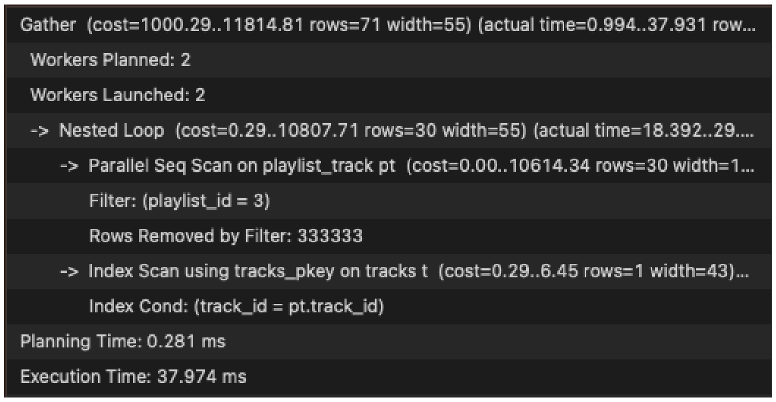


```
QUERY PLAN
Index Scan using playlists_pkey on playlists  (cost=0.29..8.30 rows=1 width=25) (actu...
  Index Cond: (playlist_id = 3)
Planning Time: 0.285 ms
Execution Time: 0.139 ms
```

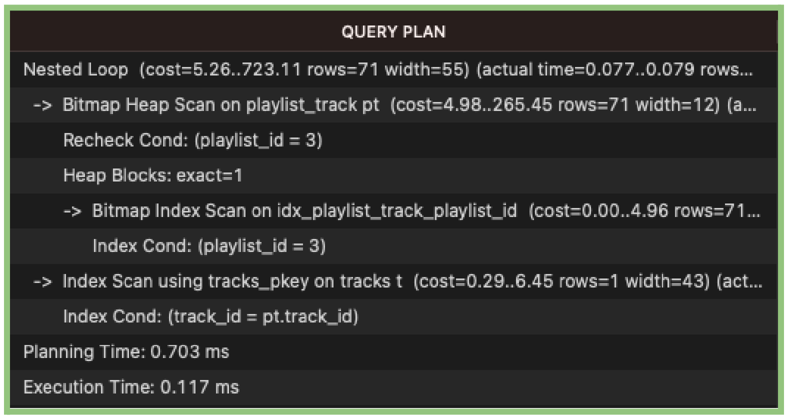


```
QUERY PLAN
Index Scan using playlists_pkey on playlists  (cost=0.29..8.30 rows=1 width=25) (actu...
  Index Cond: (playlist_id = 3)
Planning Time: 0.177 ms
Execution Time: 0.049 ms
```

/playlist/{playlist_id}



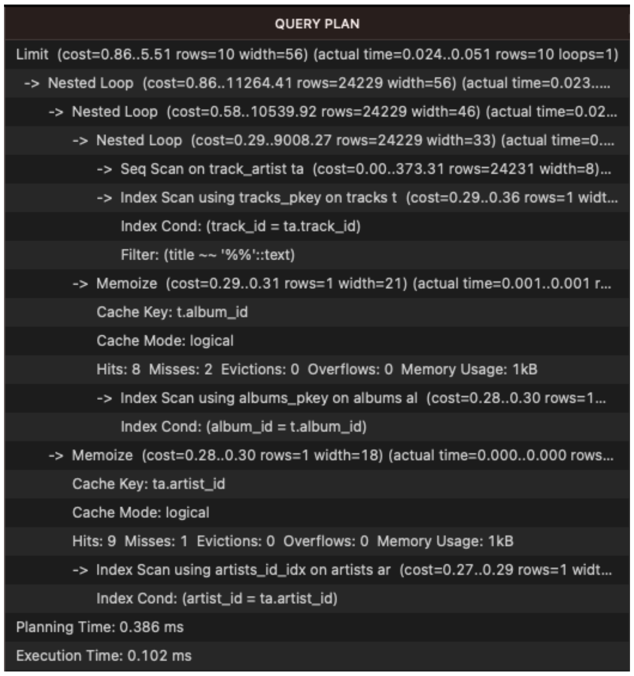
```
Gather  (cost=1000.29..11814.81 rows=71 width=55) (actual time=0.994..37.931 row...
  Workers Planned: 2
  Workers Launched: 2
  ->  Nested Loop  (cost=0.29..10807.71 rows=30 width=55) (actual time=18.392..29....
        ->  Parallel Seq Scan on playlist_track pt  (cost=0.00..10614.34 rows=30 width=1...
              Filter: (playlist_id = 3)
              Rows Removed by Filter: 333333
        ->  Index Scan using tracks_pkey on tracks t  (cost=0.29..6.45 rows=1 width=43)...
              Index Cond: (track_id = pt.track_id)
Planning Time: 0.281 ms
Execution Time: 37.974 ms
```



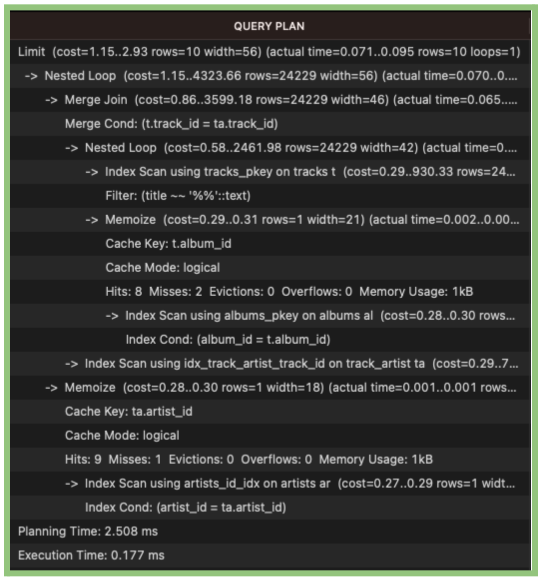
```
QUERY PLAN
Nested Loop  (cost=5.26..723.11 rows=71 width=55) (actual time=0.077..0.079 rows...
  ->  Bitmap Heap Scan on playlist_track pt  (cost=4.98..265.45 rows=71 width=12) (a...
        Recheck Cond: (playlist_id = 3)
        Heap Blocks: exact=1
        ->  Bitmap Index Scan on idx_playlist_track_playlist_id  (cost=0.00..4.96 rows=71...
              Index Cond: (playlist_id = 3)
  ->  Index Scan using tracks_pkey on tracks t  (cost=0.29..6.45 rows=1 width=43) (act...
        Index Cond: (track_id = pt.track_id)
Planning Time: 0.703 ms
Execution Time: 0.117 ms
```

Tracks

/tracks/

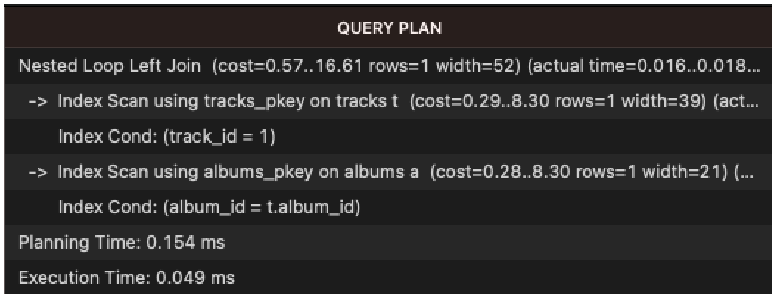


```
QUERY PLAN
Limit  (cost=0.86..5.51 rows=10 width=56) (actual time=0.024..0.051 rows=10 loops=1)
  ->  Nested Loop  (cost=0.86..11264.41 rows=24229 width=56) (actual time=0.023.....
        ->  Nested Loop  (cost=0.58..10539.92 rows=24229 width=46) (actual time=0.02...
              ->  Nested Loop  (cost=0.29..9008.27 rows=24229 width=33) (actual time=0....
                    ->  Seq Scan on track_artist ta  (cost=0.00..373.31 rows=24231 width=8)...
                    ->  Index Scan using tracks_pkey on tracks t  (cost=0.29..0.36 rows=1 widt...
                          Index Cond: (track_id = ta.track_id)
                          Filter: (title ~~ '%%'::text)
              ->  Memoize  (cost=0.29..0.31 rows=1 width=21) (actual time=0.001..0.001 r...
                    Cache Key: t.album_id
                    Cache Mode: logical
                    Hits: 8  Misses: 2  Evictions: 0  Overflows: 0  Memory Usage: 1kB
                    ->  Index Scan using albums_pkey on albums al  (cost=0.28..0.30 rows=1...
                          Index Cond: (album_id = t.album_id)
        ->  Memoize  (cost=0.28..0.30 rows=1 width=18) (actual time=0.000..0.000 rows...
              Cache Key: ta.artist_id
              Cache Mode: logical
              Hits: 9  Misses: 1  Evictions: 0  Overflows: 0  Memory Usage: 1kB
              ->  Index Scan using artists_id_idx on artists ar  (cost=0.27..0.29 rows=1 widt...
                    Index Cond: (artist_id = ta.artist_id)
Planning Time: 0.386 ms
Execution Time: 0.102 ms
```

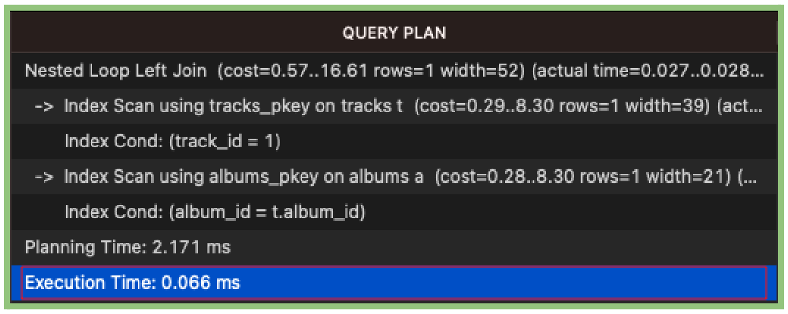


```
QUERY PLAN
Limit  (cost=1.15..2.93 rows=10 width=56) (actual time=0.071..0.095 rows=10 loops=1)
  ->  Nested Loop  (cost=1.15..4323.66 rows=24229 width=56) (actual time=0.070..0....
        ->  Merge Join  (cost=0.86..3599.18 rows=24229 width=46) (actual time=0.065.....
              Merge Cond: (t.track_id = ta.track_id)
              ->  Nested Loop  (cost=0.58..2461.98 rows=24229 width=42) (actual time=0....
                    ->  Index Scan using tracks_pkey on tracks t  (cost=0.29..930.33 rows=24...
                          Filter: (title ~~ '%%'::text)
                    ->  Memoize  (cost=0.29..0.31 rows=1 width=21) (actual time=0.002..0.00...
                          Cache Key: t.album_id
                          Cache Mode: logical
                          Hits: 8  Misses: 2  Evictions: 0  Overflows: 0  Memory Usage: 1kB
                          ->  Index Scan using albums_pkey on albums al  (cost=0.28..0.30 rows...
                                Index Cond: (album_id = t.album_id)
              ->  Index Scan using idx_track_artist_track_id on track_artist ta  (cost=0.29..7...
        ->  Memoize  (cost=0.28..0.30 rows=1 width=18) (actual time=0.001..0.001 rows...
              Cache Key: ta.artist_id
              Cache Mode: logical
              Hits: 9  Misses: 1  Evictions: 0  Overflows: 0  Memory Usage: 1kB
              ->  Index Scan using artists_id_idx on artists ar  (cost=0.27..0.29 rows=1 widt...
                    Index Cond: (artist_id = ta.artist_id)
Planning Time: 2.508 ms
Execution Time: 0.177 ms
```

/tracks/{track_id}



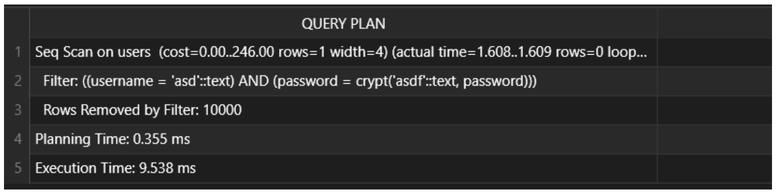
```
QUERY PLAN
Nested Loop Left Join  (cost=0.57..16.61 rows=1 width=52) (actual time=0.016..0.018...
  -> Index Scan using tracks_pkey on tracks t  (cost=0.29..8.30 rows=1 width=39) (act...
        Index Cond: (track_id = 1)
  -> Index Scan using albums_pkey on albums a  (cost=0.28..8.30 rows=1 width=21) (...
        Index Cond: (album_id = t.album_id)
Planning Time: 0.154 ms
Execution Time: 0.049 ms
```



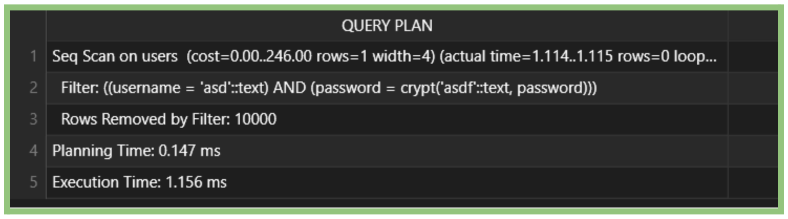
```
QUERY PLAN
Nested Loop Left Join  (cost=0.57..16.61 rows=1 width=52) (actual time=0.027..0.028...
  -> Index Scan using tracks_pkey on tracks t  (cost=0.29..8.30 rows=1 width=39) (act...
        Index Cond: (track_id = 1)
  -> Index Scan using albums_pkey on albums a  (cost=0.28..8.30 rows=1 width=21) (...
        Index Cond: (album_id = t.album_id)
Planning Time: 2.171 ms
Execution Time: 0.066 ms
```

Users

/users/validate



```
QUERY PLAN
Seq Scan on users  (cost=0.00..246.00 rows=1 width=4) (actual time=1.608..1.609 rows=0 loop...
  Filter: ((username = 'asd'::text) AND (password = crypt('asdf'::text, password)))
  Rows Removed by Filter: 10000
Planning Time: 0.355 ms
Execution Time: 9.538 ms
```



```
QUERY PLAN
Seq Scan on users  (cost=0.00..246.00 rows=1 width=4) (actual time=1.114..1.115 rows=0 loop...
  Filter: ((username = 'asd'::text) AND (password = crypt('asdf'::text, password)))
  Rows Removed by Filter: 10000
Planning Time: 0.147 ms
Execution Time: 1.156 ms
```