Indexing

First Query

```
TRACING ON;

CAPTURE '/root/tracing.txt';

SELECT * FROM fact_data WHERE  age_group_desc = '>50' ALLOW FILTERING;

CREATE INDEX IF NOT EXISTS age_group_index ON fact_data (age_group_desc);

SELECT * FROM fact_data WHERE  age_group_desc = '>50' ALLOW FILTERING;

CAPTURE OFF;

TRACING OFF;

docker cp cassandra1:/root/tracing.txt "C:\Users\milek\OneDrive -
Technological University Dublin\Year-4\Advanced
Databases\CA2\Cassandra\tracing.txt"
```

Before Index

Execute CQL3 query | 2023-12-14 00:54:31.962000
Request complete | 2023-12-14 00:54:31.990624

0.028624 seconds

Execute CQL3 query | 2023-12-14 00:54:32.083000
Request complete | 2023-12-14 00:54:32.086988

0.003988 seconds

After index

Execute CQL3 query | 2023-12-14 00:54:47.015000
Request complete | 2023-12-14 00:54:47.022598

0.007598 seconds

Execute CQL3 query | 2023-12-14 00:54:47.045000
Request complete | 2023-12-14 00:54:47.054051

0.009051 seconds

Result

Avg Speed increased by approximately 67.29%.

Second Query

```
22  Query 2 Index
23
24  TRACING ON;
25
26  CAPTURE '/root/tracing2.txt';
27
28  SELECT * FROM fact_data WHERE  votemode = 'Facebook' ALLOW FILTERING;
29
30  CREATE INDEX IF NOT EXISTS vote_mode_index ON fact_data (votemode);
31
32  SELECT * FROM fact_data WHERE  votemode = 'Facebook' ALLOW FILTERING;
33
34  CAPTURE OFF;
35
36  TRACING OFF;
37
38  docker cp cassandra1:/root/tracing.txt "C:\Users\milek\OneDrive - Technological University Dublin\Year-4\Advanced
    Databases\CA2\Cassandra\tracing2.txt"
```

Before Index

Execute CQL3 query | 2023-12-14 01:25:27.080000
Request complete | 2023-12-14 01:25:27.084706

0.004706 seconds

Execute CQL3 query | 2023-12-14 01:25:27.117000
Request complete | 2023-12-14 01:25:27.119020

0.002020 seconds

After index

Execute CQL3 query | 2023-12-14 01:25:37.531000
Request complete | 2023-12-14 01:25:37.537920

0.00692 seconds

Execute CQL3 query | 2023-12-14 01:25:37.563000
Request complete | 2023-12-14 01:25:37.570094

Execution time ≈ 0.007094 seconds

Result

Avg speed increase is approximately 150.08%

Conclusion

By placing index on age group desc we automatically order it and have reference to the pointer so once we find one we don't need to look for more since its ordered increasing speed until all are done. The same is done with votemode Facebook. Similar to that of the previous CA but just simpler queries.