

## Queries

Query 1:

```
SELECT *  
FROM testRelation0  
WHERE testRelation0.A>210 AND testRelation0.A<=600;
```

Query 2:

```
SELECT testRelation0.A, testRelation0.Cs  
FROM testRelation0  
WHERE testRelation0.A > 500 OR testRelation0.B < 100;
```

Query 3:

```
SELECT *  
FROM testRelation0 t0, testRelation1 t1  
WHERE t0.A > 200 AND t1.E != 400 AND t0.A <= t1.F;
```

## Data Description

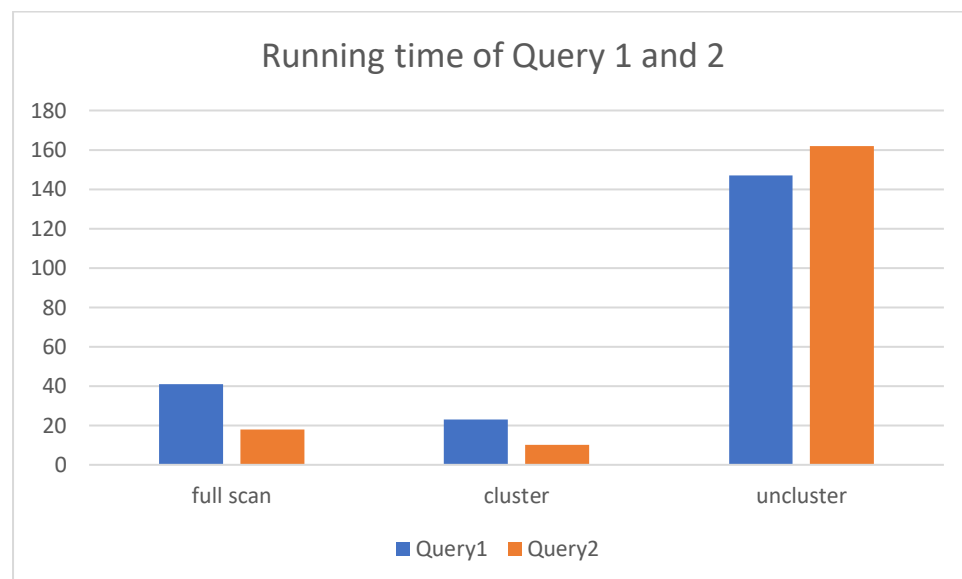
testRelation0 contains 5000 tuples with the following schema: (A, B, C). Each attribute value was chosen uniformly at random in the range 0 to 1000.

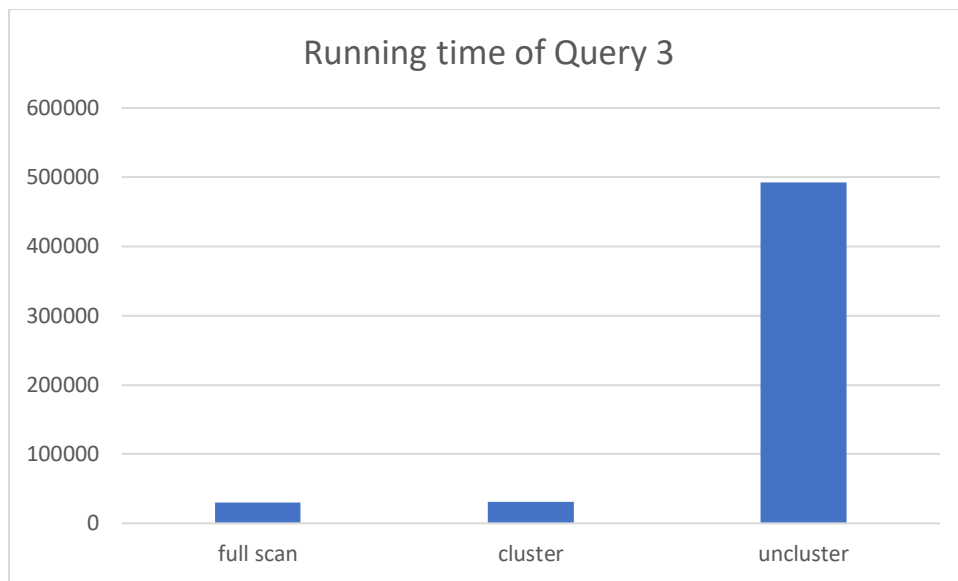
testRelation1 contains 5000 tuples with the following schema: (D, E, F, G). Each attribute value was chosen uniformly at random in the range 0 to 2000.

## Indexes used

For Query 1 and 2, we built an index file for testRelation0. For query 3, we built an index file for testRelation0 and testRelation1

## Results





### Conclusion

Query 1 has a higher selectivity than Query 2. In both of these queries, clustered index scan ran faster than full scan. However, when the query contains a join operation (Query 3), it is not clear whether indexing would speed the query evaluations.