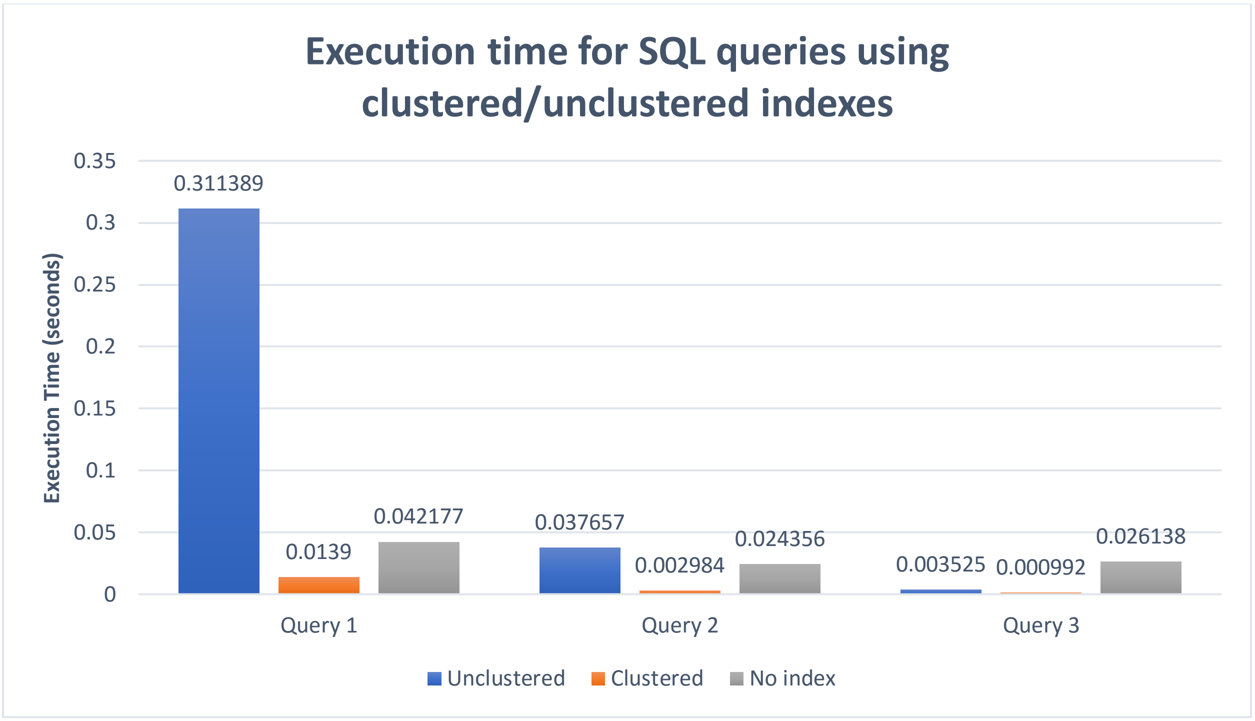
Queries ran:

Query 1: SELECT * FROM Boats WHERE Boats.E > 1000;
Query 2: SELECT * FROM Boats WHERE Boats.E > 100 AND Boats.E < 2000;
Query 3: SELECT * FROM Boats WHERE Boats.E > 100 AND Boats.E < 200 AND Boats.D < 1000;

The “Boats” relation was taken from the sample data provided with the assignment, and it appears each tuple attribute was generated uniformly at random from 0 to 10,000.

Each query was run using an unclustered index on Boats.E, a clustered index on Boats.E and then with no indexes. There was no index on Boats.D.



One interesting thing to note is that the full-scan operator outperformed the unclustered index on all but query 3. This is likely due to the fact that the index scan operator generated a small range of tuples where Boats.E ranged from 100-200, and then passed this small set to the select operator which was able to quickly move through the set.

As expected, the clustered index significantly outperformed the unclustered index and full-scan operators, with the biggest difference being in query 3 with the clustered index performing ~3.5 times faster than the unclustered index, and performing ~26.3 times faster than the full-scan.