Queries

We ran the following three queries:

```
SELECT * FROM Sailors, Boats WHERE Boats.E = Sailors.A AND Boats.E > 100 AND Boats.E < 1000
AND Sailors.A > 100 AND Sailors.A < 1000

SELECT * FROM Sailors, Boats WHERE Boats.E = Sailors.A AND Boats.E > 100 AND Boats.E < 1000

SELECT * FROM Sailors, Boats WHERE Boats.E = Sailors.A AND Boats.E = 100 AND Boats.F =
Sailors.B

SELECT * FROM Sailors, Boats WHERE Boats.E = Sailors.A AND Boats.E = 100 AND Sailors.A > 100
AND Sailors.A < 1000

SELECT * FROM Sailors, Boats WHERE Boats.E = Sailors.A
```

Description

We used the same schema as in the sample queries provided for project 4, namely:

- Sailors table with columns A, B, C
- Boats table with columns D, E, F

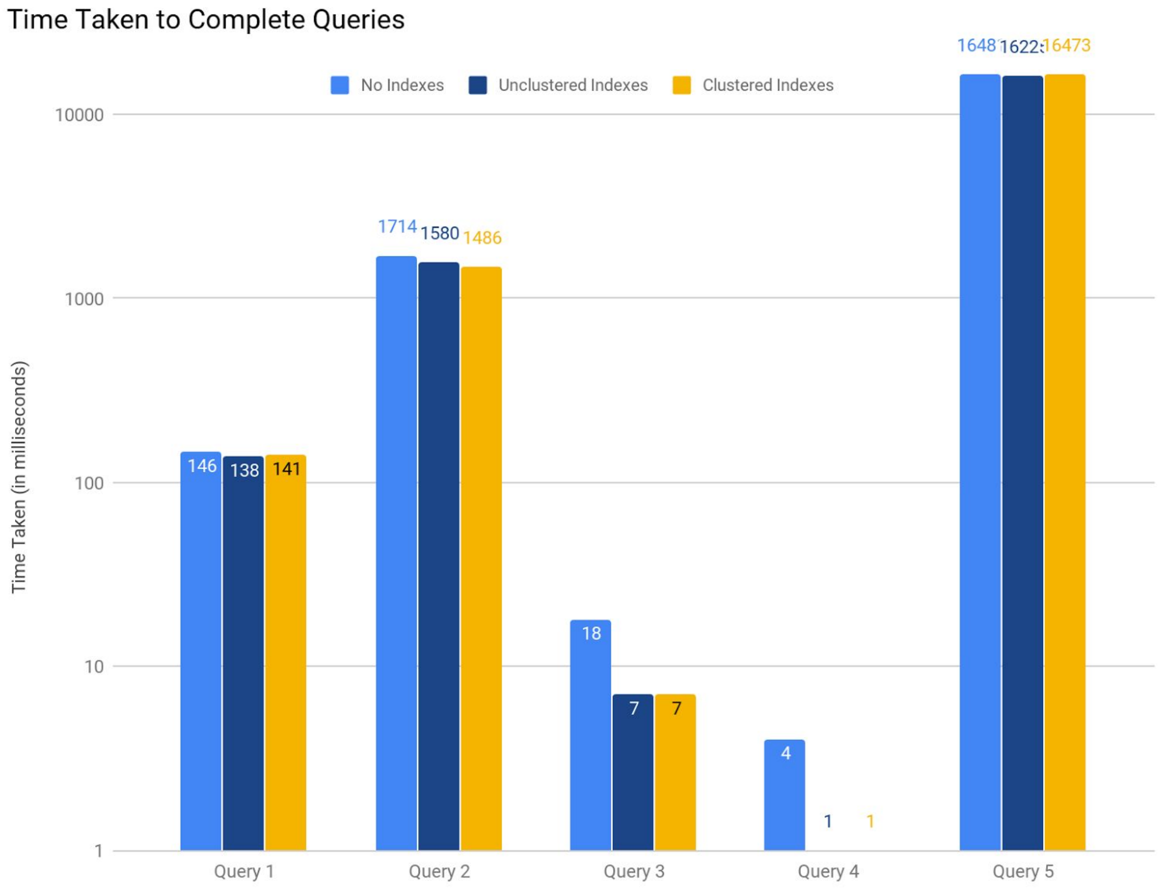
We used the sample data that was provided with project 4, so we did not generate the data ourselves.

Misc.

Since our queries involve joins, we decided to use Block-Nested Loop Join for these join operations and used 5 pages as the buffer size.

Benchmark Graph

These are the results of our performance benchmarking.



Since query 5 took so much longer than the others (understandable since it is the only query that doesn’t have any selection besides the join condition), I have plotted this graph on a log scale. As seen, performing the graphs with indexes always gave better results, and clustered indexes almost always gave better results than unclustered indexes.