Queries:

1. SELECT \* FROM Sailors, Reserves WHERE Sailors.A <= 500 AND Sailors.A >= 100;

* Index used Sailor.A Order 15

1. SELECT \* FROM Sailors, Reserves

WHERE Sailors.A > 100 AND Reserves.H >=100 AND Sailors.A = Reserves.H;

* Indexes used Sailor.A Order 15, Reserves.H Order 15

1. SELECT \* FROM Reserves, Boats

WHERE Boats.E >= 100 AND Reserves.H >=700 AND Reserves.H > Boats.E;

* Indexes used Boats.E Order 10, Reserves.H Order 15.

1. SELECT \* FROM Reserves, Boats WHERE Boats.E < 5 AND Reserves.H >=700;

* Indexes used Boats.E Order 10, Reserves.H Order 15

The schema is:

Sailors A B C

Boats D E F

Reserves G H

Indexes information unclustered:

* Boats E 0 10
* Sailors A 0 15
* Reserves H 0 15

Indexes information clustered:

* Boats E 1 10
* Sailors A 1 15
* Reserves H 1 15

Data description:

Each attribute value was chosen uniformly at random in the range of 0 to 1000; 5000 tuples per relation.

\*Data in millisecond

|  |  |  |  |
| --- | --- | --- | --- |
|  | Full Scan | Uncluster | Cluster |
| Query 1 | 15253 | 15373 | 9533 |
| Query 2 | 18607 | 144 | 110 |
| Query3 | 11270 | 89 | 36 |
| Query4 | 2309 | 45 | 31 |

Conclusion: Using BPlusTree indexes, scanning for selection queries is significantly faster than full scan under the queries we have tested. The comparison used the same join method and reader but different scanning method: full scan, un-clustered indexes and clustered indexes.