DB Assignment 6

Giannina Flamiano

December 10, 2024

Results Summary for Manual Execution in Tabular Format

Query Type	Description	Dataset Size	Index Type	Microseconds

Point Query 1	`SELECT COUNT(*)`
	`FROM accounts`
	`WHERE branch_name = 'Redwood' AND balance = 25000;`
		50,000 records	Without indexes	1) 33502
				2) 31681
				3) 32686
				4) 32750
				5) 32323
				6) 34938
				7) 33713
				8) 37417
				9) 34140
				10) 33176

			With indexes (branch_name & balance)	1) 21973 2) 30047 3) 25397 4) 23967 5) 22263 6) 24503 7) 23814 8) 25648 9) 22399 10) 25228
		100,000 records	Without indexes	1) 42192 2) 46944 3) 65590 4) 62456 5) 41856 6) 45126 7) 44998 8) 43812

				9) 41847 10) 43162
			With indexes (branch_name & balance)	1) 26421 2) 23674 3) 30256 4) 27422 5) 23243 6) 34239 7) 23765 8) 26285 9) 24951 10) 25028
		150,000 records	Without indexes	1) 53810 2) 56385 3) 53819 4) 53819 5) 58413 6) 58546 7) 58935 8) 54984 9) 54136 10) 56593
			With indexes (branch_name & balance)	1) 22676 2) 26514 3) 29766 4) 27474 5) 24003 6) 23774 7) 25923 8) 29961 9) 24500 10) 26109

Point Query 2	`SELECT COUNT(*)`
	`FROM accounts`
	`WHERE account_type = 'Checking' AND balance = 50000;`
		50,000 records	Without indexes	1) 34748
				2) 37667
				3) 34256
				4) 35736
				5) 34918
				6) 37065
				7) 34391
				8) 34143
				9) 32515
				10) 34206

			With indexes (account_type & balance)	1) 26151 2) 22113 3) 24701 4) 25886 5) 23184 6) 25398 7) 24237 8) 23706 9) 24684 10) 41053
		100,000 records	Without indexes	1) 46305 2) 54909 3) 52531 4) 51003 5) 56992 6) 47887 7) 51294 8) 53275 9) 51230 10) 49912

			With indexes (account_type & balance)	1) 22627 2) 29709 3) 26290 4) 27596 5) 27507 6) 31284 7) 26755 8) 29989 9) 27182 10) 25567
		150,000 records	Without indexes	1) 63915 2) 66056 3) 56793 4) 58206 5) 70299 6) 62478 7) 57380 8) 60317 9) 59853 10) 63808
			With indexes (account_type & balance)	1) 24125 2) 27454 3) 26849 4) 28805 5) 25484 6) 27670 7) 25783 8) 24381 9) 24381 10) 27621

<table>
<tr><td rowspan="3">Range Query 1</td><td rowspan="3">SELECT COUNT(*)
FROM accounts
WHERE branch_name = 'Redwood' AND balance
BETWEEN 10000 AND 25000;</td><td rowspan="2">50,000 records</td><td>Without indexes</td><td>1) 34424
2) 34919
3) 33618
4) 33589
5) 34351
6) 38585
7) 35176
8) 35419
9) 34203
10) 47239</td></tr>
<tr><td>With indexes (branch_name & balance)</td><td>1) 24029
2) 26418
3) 26226
4) 35880
5) 28109
6) 32830
7) 24090
8) 26956
9) 27600
10) 26334</td></tr>
<tr><td>100,000 records</td><td>Without indexes</td><td>1) 47098
2) 44954
3) 45897
4) 43809
5) 44208
6) 42645
7) 52167
8) 43200
9) 49572
10) 45799</td></tr>
</table>

			With indexes (branch_name & balance)	1) 27754 2) 23973 3) 24541 4) 26743 5) 24145 6) 30554 7) 24311 8) 25503 9) 24278 10) 24910
		150,000 records	Without indexes	1) 55665 2) 56527 3) 55559 4) 57244 5) 59177 6) 59160 7) 66539 8) 56539 9) 61197 10) 62384
			With indexes (branch_name & balance)	1) 33153 2) 29925 3) 29779 4) 28948 5) 30868 6) 29771 7) 25214 8) 37739 9) 29846 10) 27288

Range Query 2	`SELECT COUNT(*)`
	`FROM accounts`
	`WHERE account_type = 'Checking' AND balance BETWEEN 50000 AND 100000;`
		50,000 records	Without indexes	1) 32004
				2) 35938
				3) 36102
				4) 35036
				5) 34702
				6) 36583
				7) 34662
				8) 38963
				9) 36911
				10) 36461

			With indexes (account_type & balance)	1) 38247 2) 29376 3) 28604 4) 29374 5) 29722 6) 35997 7) 28078 8) 28110 9) 31041 10) 30839
		100,000 records	Without indexes	1) 44666 2) 52825 3) 55014 4) 52513 5) 48784 6) 50510 7) 51633 8) 49998 9) 51435 10) 53303

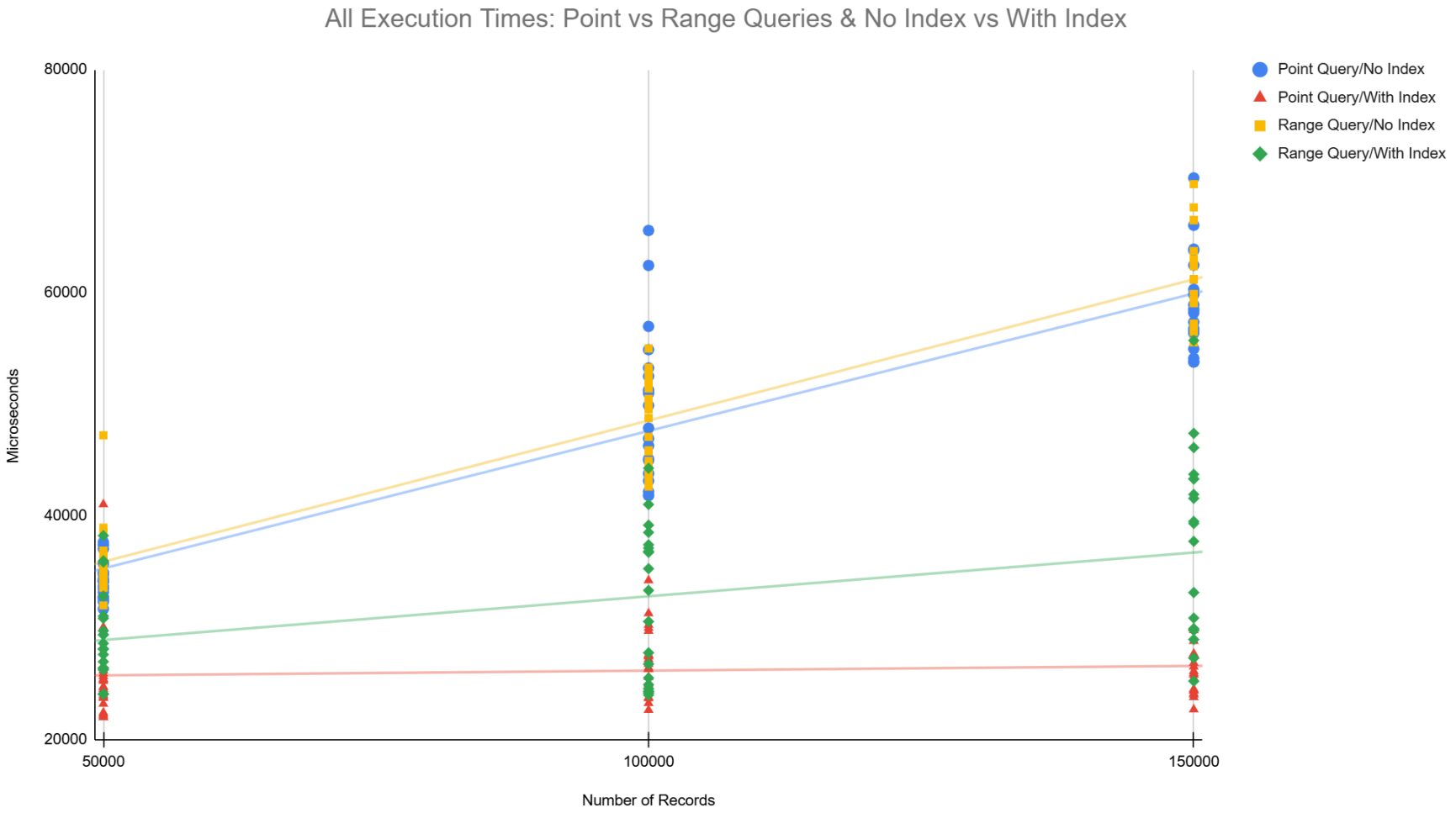
			With indexes (account_type & balance)	1) 44289 2) 33345 3) 38538 4) 36752 5) 39187 6) 35291 7) 41044 8) 37429 9) 36845 10) 37140
		150,000 records	Without indexes	1) 59902 2) 63749 3) 63010 4) 61178 5) 61244 6) 62802 7) 67655 8) 66494 9) 59070 10) 69745
			With indexes (account_type & balance)	1) 55734 2) 39337 3) 47413 4) 41590 5) 43327 6) 39523 7) 43337 8) 41944 9) 46136 10) 43743

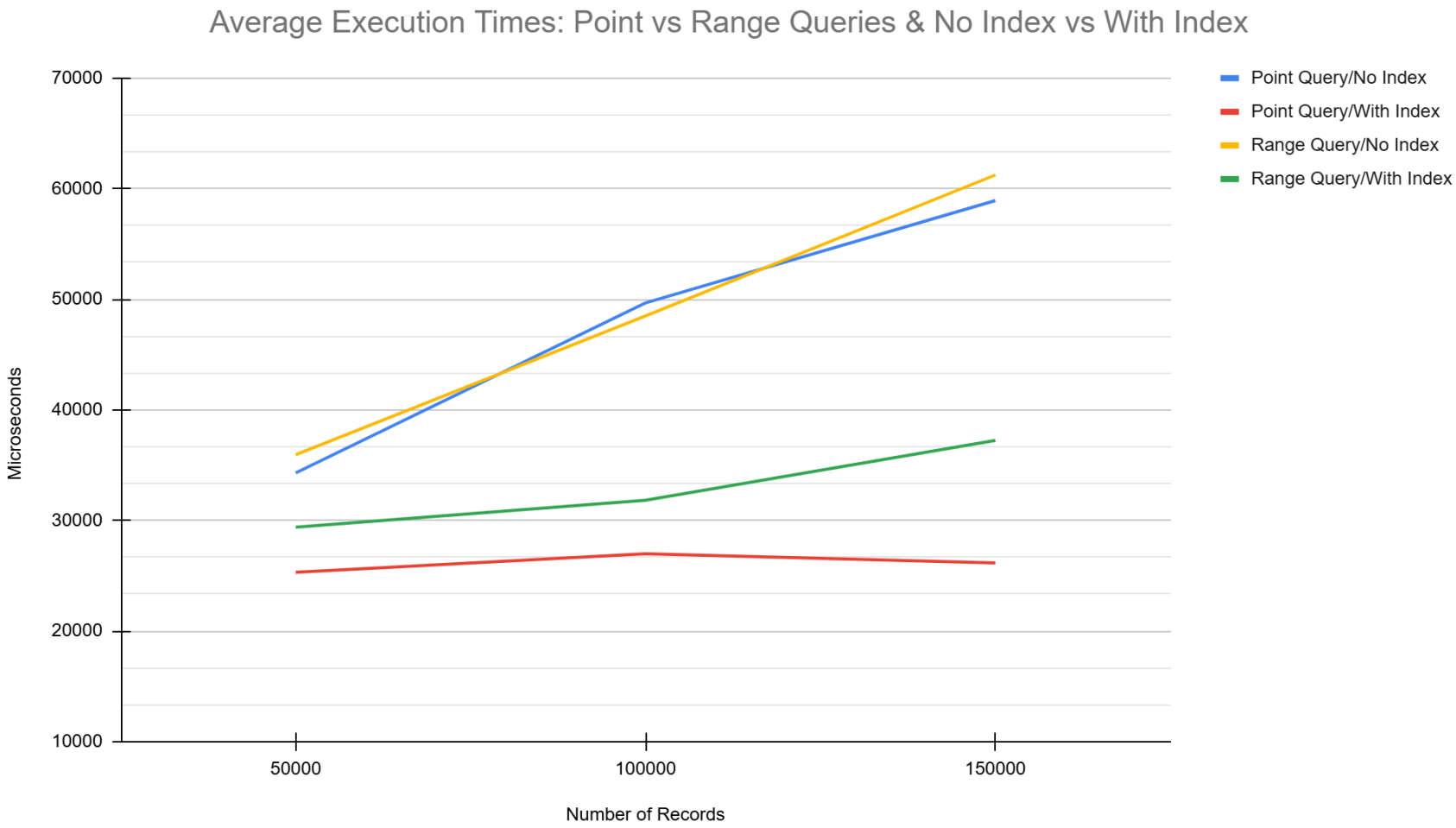
Results Summary for Stored Procedure that Measures Average Execution Time

Query Type	Description	Dataset Size	Index Type	Average Execution Time Microseconds
Point Query 1	`SELECT COUNT(*) FROM accounts WHERE branch_name = 'Redwood' AND balance = 25000;`	50,000 records	Without indexes	9837.2
			With indexes (branch_name & balance)	155.7
		100,000 records	Without indexes	20003.6
			With indexes (branch_name & balance)	229
		150,000 records	Without indexes	29077.9
			With indexes (branch_name & balance)	213.2
Point Query 2	`SELECT COUNT(*) FROM accounts WHERE account_type = 'Checking' AND balance = 50000;`	50,000 records	Without indexes	12617.8
			With indexes (account_type & balance)	132.9
		100,000 records	Without indexes	25263.6
			With indexes (account_type & balance)	320.5
		150,000 records	Without indexes	35448.2
			With indexes (account_type & balance)	229.5

Range Query 1	SELECT COUNT(*) FROM accounts WHERE branch_name = 'Redwood' AND balance BETWEEN 10000 AND 25000;	50,000 records	Without indexes	10662.2
			With indexes (branch_name & balance)	721
		100,000 records	Without indexes	25320
			With indexes (branch_name & balance)	1213.4
		150,000 records	Without indexes	33150.5
			With indexes (branch_name & balance)	1617.7
Range Query 2	SELECT COUNT(*) FROM accounts WHERE account_type = 'Checking' AND balance BETWEEN 50000 AND 100000;	50,000 records	Without indexes	12341
			With indexes (account_type & balance)	5612.2
		100,000 records	Without indexes	26807.9
			With indexes (account_type & balance)	10542.8
		150,000 records	Without indexes	35448.2
			With indexes (account_type & balance)	20689.2

Results Summary for Manual Execution in Graphical Format




Average Execution Times: Point vs Range Queries & No Index vs With Index
70000
60000
50000
40000
30000
20000
10000
Microseconds
50000
100000
150000
Number of Records
Point Query/No Index
Point Query/With Index
Range Query/No Index
Range Query/With Index

Conclusion

In this assignment, we analyze the performance of Point and Range Queries with and without indexing. Based on the experiment conducted above, we see that the performance of Point Queries without indexing linearly deteriorated as the number of records increased, in other words, the execution time grew with the number of records. We see the same behavior with Range Queries without indexing. However, with both Point and Range Queries with indexing, we see a faster execution time overall, but we also see that the execution time remains almost stagnant as the number of records increases. In conclusion, based on the data found above, Range Queries tend to have longer execution time in comparison to Point Queries. Additionally, indexing drastically improves performance for both query types.