Lab 7

Xiaohui Chen
EID:xc2388
CS ID: xc2388

1) Performance results

a) Fill in the following table of absolute measurements

		Load Time		Query 1		Query 2		Query 3	
	Data Generator	I	II	I	II	I	II	I	II
Physical Organization									
1		1.7 hr	4.15576hr	4960.3ms	5886.3ms	3379.2ms	3639.5ms	3959.7ms	3792.0
2		2.26489hr	11.7108hr	12.9ms	14.3ms	3285.4ms	3348.8ms	15.3ms	11.2ms
3		2.57332hr	11.5733hr	3356.7ms	3524.0ms	13.3ms	13.0ms	8.6ms	9.1ms
4		9.33694hr	20.3260hr	10.2ms	12.1ms	11.1ms	12.9ms	6.3ms	7.9ms

b) Measurements with respect to speed-up

i) Let the load time of the database on physical organization 1 on sorted inserts form a baseline. What is the speed-up, (other-organization/baseline), for the remaining load times?

ii) Similarly, for each of the queries, let the physical organization 1 on sorted inserts form a baseline. What is the speed-up, for the remaining executions?

Report your results by completing the following table.

		Load Time		Query 1		Query 2		Query 3	
	Data Generator	I	II	I	II	I	II	I	II
Physical Organization									
1		1	2.4446	1	1.1867	1	1.0770	1	0.9576
2		1.3322	6.8887	0.0026	0.0029	0.9722	0.9910	0.0039	0.0028
3		1.5137	6.8078	0.6767	0.7104	0.0039	0.0038	0.0021	0.0022
4		5.4923	11.9565	0.0021	0.0024	0.0033	0.0038	0.0016	0.0020

Additional measurement:

The loading time for Variation 1, physical organization 1 with 1 index before insertion is 2.26489hr.

The loading time for Variation 1 physical organization 1 without index before insertion is 1.7 hr and the time for adding index is 3.066 minutes.

2) Discuss your findings. (4-10 sentences) (Hint: Use the SQL explain command, or its GUI equivalent, and see if/how the query plan changes for each physical organization.)

When no index keys are added, insertion is faster. This is mainly because the complier doesn’t have to build index trees while insertion. Of course, when the relation has 2 secondary indexes, the load time will be much slower. Queries which examine attributes with indexes will much faster than those whose attributes don’t have indexes. This is because, searching in the index search tree is much faster than iterating the whole table. When the relation has 2 secondary indexes, the queries involve those 2 attributes will be fastest because the complier searches through to the first index search tree and then match with the second condition. In all, adding indexes speeds up queries by slows down insertions.