Name: Xiang Gu
EID: xg2847
HW6a

1. Performance Result:

	Load Time		Query 1		Query 2		Query 3	
Data Gen Physical Config	I (sorted)	II (random)	I	II	I	II	I	II
1 (no index)	446.27	377.37	0.404	0.330	0.379	0.325	0.383	0.317
2 (index on ColA)	476.56	369.15	0.001	0.001	0.408	0.314	0.001	0.001
3 (index on ColB)	488.33	376.24	0.430	0.316	0.001	0.001	0.001	0.001
4 (index on both Cols)	419.24	415.49	0.001	0.001	0.001	0.001	0.0003	0.0002

2. Relative Speedup

	Load Time		Query 1		Query 2		Query 3	
Data Gen Physical Config	I (sorted)	II (random)	I	II	I	II	I	II
1 (no index)	1	0.846	1	0.817	1	0.858	1	0.828
2 (index on ColA)	1.068	0.827	0.002	0.002	1.077	0.829	0.003	0.003
3 (index on ColB)	1.094	0.843	1.064	0.782	0.003	0.003	0.003	0.003
4 (index on both Cols)	0.939	0.931	0.002	0.002	0.003	0.003	0.001	0.001

3. Findings:

As for load time, the result shows loading random data is faster than loading sorted data, which is surprising. I thought loading random data will be more time-consuming. In addition, there isn't much change in the loading time between having no index and having index, which is another surprising fact. I thought inserting with index will be more time-consuming because we don’t only need to insert the data but also maintain the index structure as we insert.

As for the query execution time, first of all, we notice that if no index is used, query on randomly inserted database is faster than query on sortedly inserted database, which is again surprising because I thought without index, we need to scan the whole table anyway but maybe it will be faster if the data was sortedly inserted. Aside from this, we observed that having an index will significantly speed up the query time if the query variable is the one on which we built an index. Furthermore, even with only one index (on either colA or colB), it will significantly help speed up type 3 queries as well.