CS386D
4/4/17
**V1.0**

# Homework 10 (Lab 3)

**Assigned:** 4/4/17
**Due**: 4/17/17

The primary goal of this lab is to observe that query optimizers consider the statistical properties of the data. The earlier lab on real world query optimizer behavior targeted the relationship between select predicates, secondary indexes and disk locality. This homework targets the join operator and the optimizer's cost function wrt the size of intermediate join results. An ancillary goal of the lab is the introduction of classic methodology used in the benchmarking of database systems.

**Deliverable:**

The deliverable for this homework are the physical plans determined by the query optimizer for each of the queries. This can be obtained using the SQL explain command, but graphical presentation of the plans is preferred

**Important:** You do not have to execute the queries, just get the optimizer to "explain" its plan. You will have to run an explicit database command for the database to examine itself and gather statistics for the optimizer. Be sure you do this after loading the database AND building the indexes, and before generating the query plans. Per the last assignment budget time for the initial database load.

**Test data:** The test data will comprise the same data table replicated 6 times. Let that table be defined as *TestData*(pk, ht, tt, ot, filler) Let there be 1,000,000 rows with a unique primary key. Column ht should contain uniformly distributed values from 0 to 99,999 (i.e. **h**undred **t**housand), Column tt, should contain uniformly distributed values from 0 to 9,999 (i.e. **t**en **t**housand), Column ot, should contain uniformly distributed values from 0 to 999 (i.e. **o**ne **t**housand), Column filler – same as before Note this schema is modeled after the schema in a famous database benchmark, "The Wisconsin Benchmark", The original paper: "Benchmarking Database Systems , A Systematic Approach"
http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.54.7764

A retrospective paper is assigned reading: "The Wisconsin Benchmark: Past, Present, and Future"
http://firebird.sourceforge.net/download/test/wisconsin_benchmark_chapter4.pdf

**Physical Schema:** Create three copies of the test data table. Call the tables A, B and C. The contents should be identical. Only the names of the tables are different. (Hint: You may run a data generator similar to the last lab, once for each of the three tables. But there is a much faster way.)

Create three more copies of the test data table. Call them A', B', C', (or Aprime, Bprime, Cprime). Build secondary indices on each of the three columns, ht, tt, ot, for each of the three additional tables, for a total of nine indexes.

**Queries:** Determine physical query plans for SFW queries with the following where clauses You need only turn in certain plans and question answers. Do detail your hardware/software platform. i.e. what RDBMS and OS you used, CPU, RAM and Disk etc.

**Section 1:**
Determine if and when range size and index impact choice of join order and algorithm.

1. A.pk = B.pk
2. A.ht = B.ht
3. A'.ht = B'.ht
4. A.ht = B.ht = C.ht
5. A.ot = B.ot = C.ot
6. If 4 and 5 produce different plans, then A.tt = B.tt = C.tt, else simply report " 4 and 5 produce the same plan"
7. A.pk = B.ht = C.ht
8. A.pk = B.ot = C.ot
9. Same as 6, as applied to 7 and 8
10. A'.ht = B'.ht = C'.ht
11. repeat 10 for the ot column, (i.e. 10 is to 11, as 4 is to 5)
12. Same as 6, as applied to 10 and 11.
13. A.pk = B'.ht = C'.ht
14. Repeat 13 for the ot column, use A.pk for the "A" argument
15. Same as 6, as applied to 13 and 14.

Turn in results for 5 and 6, 8 and 9, 11 and 12, and 14 and 15. An objective here is to reveal if and where the change in size of the data distribution results in a change in the query plan.

**Section 2:**
Investigate join reordering.

16. A.ht = B.th = Cot
17. C.ot = B.th = A.ht
18. A.pk = B'.th = C'ot
19. C'.ot = B'.th = A.pk

Turn in screenshots that show the same query plan was produced, (or not), even though the order in the where clause is different.

Test to see if the order of the tables in the from clause impacts the final plan. If not, say so. If it does turn in the example.