CS54100 - Homework 3
FALL 2016

Due: Friday, December 2, 2016 in class, before class starts.
(There will be a 10% penalty for each late day. After 5 late days, the homework will not be accepted.)

Part 1: Query Evaluation (30 Points)

1. For each of the following SQL queries, for each relation involved, list the attributes that must be examined to compute the answer. All queries refer to the following relations:
   Emp(eid: integer, did: integer, sal: integer, hobby: char(20))
   Dept(did: integer, dname: char(20), floor: integer, budget: real)

   a. SELECT E.eid, FROM Emp E, Dept D WHERE E.did = D.did GROUP BY E.did HAVING AVG(E.sal) > 5000
   b. SELECT E.eid, D.floor FROM Emp E, Dept D WHERE E.did = D.did

2. Consider the join R $\bowtie_{R.a = S.b}$ S given the following information about the relations to be joined. The cost metric is the number of page I/Os unless otherwise noted, and the cost of writing out the result should be uniformly ignored. Show your work.
   Relation R contains 10,000 tuples and has 10 tuples per page.
   Relation S contains 2000 tuples and also has 10 tuples per page.
   Attribute b of relation S is the primary key for S.
   Both relations are stored as simple heap files.
   Neither relation has any indexes built on it.
   52 buffer pages are available.

   a) What is the cost of joining R and S using a page-oriented simple nested loops join? What is the minimum number of buffer pages required for this cost to remain unchanged?
   b) What is the cost of joining R and S using a sort-merge join? What is the minimum number of buffer pages required for this cost to remain unchanged?
   c) What is the cost of joining R and S using a hash join? What is the minimum number of buffer pages required for this cost to remain unchanged?
   d) How many tuples does the join of R and S produce, at most, and how many pages are required to store the result of the join back on disk?

Part 2: Query Optimization (30 Points)

Consider a relation with this schema:

Employees(eid: integer, ename: string, sal: integer, title: string, age: integer)

Suppose that the following indexes, all using Alternative (2) for data entries, exist: a hash index on eid, a B+ tree index on sal, a hash index on age, and a clustered B+ tree index on (age, sal). Each Employees record is 100 bytes long, and you can assume that each index data entry is 20 bytes long. The Employees relation contains 10,000 pages.

1. Consider each of the following selection conditions and, assuming that the reduction factor (RF) for each term that matches an index is 0.1, compute the cost of the most selective access path for retrieving all Employees tuples that satisfy the condition:
    a. sal > 100
    b. age = 25
    c. age > 20
    d. eid = 1000
    e. sal > 200 ^ age > 30
    f. sal > 200 ^ age = 20
    g. sal > 200 ^ title = ‘CFO’
    h. sal > 200 ^ age > 30 ^ title = ‘CFO’