CS532 Homework #4
Due at the Beginning of Class on April 23, 2018

1. [20%] Consider the following SQL query for a banking database system. For simplicity, assume that there is only one account type. Also, the status of a customer can be bronze, silver, or gold member.

   ```
   select account_id, customer_name
   from accounts
   where status = 'gold member' and customer_city = 'Vestal';
   ```

   Suppose that (a) each account tuple occupies 200 bytes; (b) all pages are of size 4KB (i.e., 4000 bytes); (c) there are 50 customers (out of total 10,000 customers) with gold member status; and (d) 1000 customers live in Vestal. Discuss how to evaluate the query for the following cases (Hint: If there are different options for a case, try to minimize the number of page I/Os. Also, don't forget to differentiate sequential I/Os from random I/Os):

   (a) [5%] Case 1: There is no index on either status or customer_city.

   (b) [5%] Case 2: There is an index on customer_city but no index on status.

   (c) [5%] Case 3: There is a secondary index on status and a secondary index on customer_city.

   (d) [5%] Case 4: There is a primary index on customer_city and a secondary index on status.

2. [30%] Consider the join $R \infty_{R.A = S.B} S$, where R and S are two relations. Three join methods, i.e., nested loop, sort merge and hash join, are discussed in class. Nested loop and sort merge may benefit from the existence of indexes. Identify three different situations (i.e., with given sizes of R and S, and the index status on R.A and/or S.B) such that each of the following claims is true for one situation.

   (a) [10%] Nested loop outperforms sort merge and hash join.
   (b) [10%] Sort merge outperforms nested loop and hash join.
   (c) [10%] Hash join outperforms sort merge.

   Here the performance is compared based on the number of I/O pages. Suppose N and M are sizes of R and S in pages, respectively. Without loss of generality, assume that $N > M$. Also, assume that the memory buffer for the join is not large enough to hold the entire R. Justify your answer.

3. [25%] Consider the following query execution plans:

   Plan A: σ GPA $\geq$ 2(Students ∞ $_{\text{Students.SSN = Faculty.SSN}}$ Faculty)
   Plan B: (σ GPA $\geq$ 2 (Students)) ∞ $_{\text{Students.SSN = Faculty.SSN}}$ Faculty

   Which plan will be selected if the four rules for query optimization in Chapter 11 are applied? Is the selected plan more efficient than the other one? Justify your answers.

4. [25%] Consider the following three relations:

   Supplier(Supp#, Name, City, Specialty)
   Project(Proj#, Name, City, Budget)
   Order(Supp#, Proj#, Part_name, Quantity, Cost)

   Apply the four heuristic optimization rules discussed in class to find an efficient execution plan for the following SQL query. It is assumed that there are much more suppliers in New York City than there are projects with budget over 10 million dollars. Draw the corresponding query tree after each rule is applied.

   ```
   select Supplier.Name, Project.Name
   from Supplier, Order, Project
   where Supplier.City = 'New York City' and Project.Budget > 10000000
   and Supplier.Supp# = Order.Supp# and Order.Proj# = Project.Proj#
   ```

   Show the query tree after each optimization rule is applied. Also, write the relational algebra expression corresponding to the fully optimized query tree.