In this assignment, you should have EITHER problems 1 and 2 (worth 20 points each) OR problem 3 (worth 40 points). If you do both, then we will take the max(your problem 1 plus problem 2 scores, problem 3 score). In addition, you should do the supplemental homework 2 where each problem is worth 10 points. You may work with one partner and sign both of your names to your paper.

1. Design a program in your favorite programming language that can generate tuples where field values can be unique, have a fixed set of distinct values that are allocated to tuples according to a uniform probability distribution, and a fixed set allocated to the tuples according to a fractal probability distribution (70-30 rule). An example function is pseudocode to do this is as follows:

```pseudocode
gen(frac, N)
begin
p := random permutation of numbers from 1 to N
outvec := p // so outvec is of length N
while(|p| > 1
  p := first frac*|p| elements of p
  concatenate p to outvec
end while
return random permutation of outvec
```

Execute this program with gen(0.3, x) where x is 70,000 or so (might be more). The idea is to generate an array of 100,000 fractally distributed numbers.

Populate table sales(id, itemid, quantity) such that there are 100,000 ids, 20,000 itemids distributed with uniform probability among the sale ids. Then quantities are to be distributed in a fractal manner by using the above-generated fractal array.

Next write one or more queries that find the ten items having the greatest total quantities (sum of the quantities grouped by item) and the ten items having the least total quantities. Use any relational database on any operating system you like. Show us the queries and then a terminal history session showing execution and the results.

2. Choose two rules of thumb from the book (e.g., rules having to do with chopping, indexes, commits, checkpoints, etc.). For each one, find (and demonstrate using mysql or some other database system) data distributions/access patterns where it is satisfied and data distributions/access patterns where it is not. A data distribution refers to the frequency of occurrences of different values of a given variable across the rows - e.g. uniform and fractal are two different (and convenient since you already generated data with them) distributions. Thus, you need to show four different performance numbers for each rule of thumb derived by executing real queries in a real database: results with and without the rule of thumb applied to columns
with two different data distributions. Comment on whether the case (i.e. the data distribution) where the rule of thumb is satisfied is more likely than when it isn’t. Also explore how you would modify the statement of the rule of thumb to make it more precise given the performance results.

3. Download vistrails and Philippe Bonnet’s workflow and run his experiment. Vary the index choice. Show an image of the vistrails flow, the queries, and the timing results. Instructions are at http://effdas.itu.dk/repeatability/experiment.html For supplemental material, please see the general guidelines for repeatability at the Sigmod conference here: http://www.sigmod2011.org/calls_papers_sigmod_research_repeatability.shtml and a tuning case study here: http://www.vistrails.org/index.php/WikiQuery