HW12

- **13.4**

The relation resulting from the join of r_1, r_2, r_3 will be the same no matter which way we join them, due to the associative and commutative properties of the joins. So we will consider the size based on the strategy of $((r_1 \bowtie r_2) \bowtie r_3)$. Joining r_1 with r_2 will yield a relation of at most 1000 tuples, since C is a key for r_2. Likewise, joining that result with r_3 will yield a relation of at most 1000 tuples because E is a key for r_3. Therefore the final relation will have at most 1000 tuples.

An efficient strategy of computing this join would be to create an index on attribute C for relation r_2 and on E for r_3. Then for each tuple in r_1, we do the following:

a.

Use the created index for r_2 to look up at most one tuple which matches the C value of r_1.

b.

Use the created index on E to look up in r_3 at most one tuple which matches the unique value for E in r_2.

- **13.5**

The estimated size of the relation can be determined by calculating the average number of tuples which would be joined with each tuple of the second relation. In this case, for each tuple in r1, 1500/V(C, r2) = 15/11 tuples (on the average) of r2 would join with it. The intermediate relation would have 15000/11 tuples. This relation is joined with r3 to yield a result of approximately 10,227 tuples (15000/11 × 750/100 = 10227).

A good strategy should join r1 and r2 first, since the intermediate relation is about the same size as r1 or r2. Then r3 is joined to this result.

- **13.15**

Using the index on (dept_name, building), we locate the first tuple having (building "Watson" and dept_name"Music"). We then follow the pointers retrieving successive tuples as long as building is less than "Watson". From the tuples retrieved, the ones not satisfying the condition (budget < 55000) are rejected.

- **13.19**

Suppose the histogram H storing the distribution of values in r is divided into ranges $r_1, \ldots, r_n$. For each range r_i with low value $r_{i:low}$ and high value $r_{i:high}$, if $r_{i:high}$ is less than v, we add the number of tuples given by

$H(r_i)$

to the estimated total. If $v < r_{i:high}$ and $v >= r_{i:low}$, we assume that values within r_i are uniformly distributed and we add

$$H(r_i) \times \frac{v - r_{i:low}}{r_{i:high} - r_{i:low}}$$

to the estimated total.