Ferhat Elmas 214805

Advanced Databases Hw-3

1. If the table has *n* records and *k* buckets, an equi-depth histogram divides the set of values on that column into *k* ranges such that each range has same number of values, e.g **n/k.** Since optimizer will do interpolation according to the histogram, equi-depth histogram provides equal probable distribution of buckets which is better for the interpolation than what equi-width provides because equi-width just divides the values into k buckets according to max and min, and in skewed data set, much of tuples can go into one bucket, which makes the interpolation very wrong.

2. a) They aren't equivalent because in the second case, first we applied projection and get name and occupation but then there is no age column to apply age constraint.

   b) They are equivalent because this time we don't look for the age.

   c) They are equivalent because joins are associative.

   d) Not equivalent. Let's take this table:

ali	21	carpenter
veli	21	student
deli	19	carpenter

   First part returns only "ali 21 carpenter" but second returns all tuples.

   e) Not equivalent. If we take the same table with d.

   First part returns all tuples where second part returns only "ali 21 carpenter"

3.

	tpsize	#tps/pgs	#tps	#pgs	#I/O	#seeks
T	8+8+4	51	384000	7530	7530	1
σ(T)	20	51	57600	1130	0	0
π(T)	16	64	57600	900	0	0
C	8+56	16	1600	100	-	-
T⋈C	80	12	57600	4800	10	10
σ(TC)	80	12	36	3	0	0
π(TC)	8	128	36	1	0	0

S	8+56	16	24000	1500	-	-
TC⋈S	72	14	24000	1715	1500	1
$\pi(TCS)$	56	18	36	2	0	0
				Total	9040	12

4.

    a) No combiner:

- First mapper generates :
  - <advanced, 1>
  - <database, 1>
  - <advanced, 1>
  - <algorithms, 1>
- Second mapper generates:
  - <theory, 1>
  - <and, 1>
  - <practise, 1>
  - <advanced, 1>
  - <theory, 1>
- Third mapper generates:
  - <database, 1>
  - <theory, 1>
  - <and, 1>
  - <advanced, 1>
  - <modeling, 1>

Therefore, inputs to the reducers:

- First reducer (a-d):
  - <advanced, 1>
  - <database, 1>
  - <advanced, 1>
  - <algorithms, 1>
  - <and, 1>
  - <advanced, 1>
  - <database, 1>
  - <and, 1>
  - <advanced, 1>
- Second reducer (e-p):
  - <practise, 1>
  - <modeling, 1>
- Third reducer (r-z):
  - <theory, 1>

- ○ <theory, 1>
- ○ <theory, 1>

And output of reducers:
- ● First reducer:
    - ○ <advanced, 4>
    - ○ <algorithms, 1>
    - ○ <and, 2>
    - ○ <database, 2>
- ● Second reducer:
    - ○ <modeling, 1>
    - ○ <practise, 1>
- ● Third reducer:
    - ○ <theory, 3>

b) With combiner:

Output of mappers:
- ● First mapper:
    - ○ <advanced, 2>
    - ○ <database, 1>
    - ○ <algorithms, 1>
- ● Second mapper:
    - ○ <theory, 2>
    - ○ <and, 1>
    - ○ <practise, 1>
    - ○ <advanced, 1>
- ● Third mapper:
    - ○ <database, 1>
    - ○ <theory, 1>
    - ○ <and, 1>
    - ○ <advanced, 1>
    - ○ <modeling, 1>

Inputs to the reducers:
- ● First reducer (a-d):
    - ○ <advanced, 2>
    - ○ <database, 1>
    - ○ <algorithms, 1>
    - ○ <and, 1>
    - ○ <advanced, 1>
    - ○ <database, 1>
    - ○ <and, 1>
    - ○ <advanced, 1>
- ● Second reducer (e-p):
    - ○ <practise, 1>
    - ○ <modeling, 1>
- ● Third reducer (r-z):
    - ○ <theory, 2>
    - ○ <theory, 1>

Their results will be same with no combiner result because combiner doesn't change the overall results, it just fasten the execution by decreasing network traffic.