Uvin Abeysinghe – Database Systems – 789931 – Assignment 3

Question 1

a) Page - oriented Nested L oops Join. Consider A as the outer relation.

Cost = NPages(A) + Npages(A) * Npages(B)
Cost= 10,000+10,000 * 1,000
Cost=10,010,000

b) Block - oriented Nested L oops Join. Consider A as the outer relation.

Cost = NPages(A) + ceil(NPages(A)/(BLOCK_SIZE-2))* NPages(B)
Cost= 10,000 + ceil (10,000/(502-2)) * 1000
Cost = 30,000

c) Sort-Merge Join

Cost= NPages(A) + NPages(B) + 2* NPages(A)* ceil(1+logBLOCKSIZE-1ceil(NPages(A)/BLOCKSIZE)) + 2* NPages(B)* ceil(1+logBLOCKSIZE-1ceil(NPages(B)/BLOCKSIZE)) +
Cost = 10,000 + 1,000 +2 * 10,000 * ceil(1+log 501 ceil(10,000/502)) +2 * 1,000 * ceil(1+log 501 ceil(1,000/502))
Cost = 11000 +4 *10,000 +4 * 1,000 = 55,000

d) Hash Join

Cost = 3* (NPages(A) + NPages(B))
Cost = 3*(10,000 + 1,000)
Cost= 33,000

e) What would the lowest possible I/O cost be for joining A and B using any join algorithm and how much buffer space would be needed to achieve this cost?? Explain briefly

Cost = NPages(B) + ceil(NPages(B)/(BLOCK_SIZE-2))* NPages(A)
Cost = 1,000 + ceil(1,000/ 1,002) * 10,000
Cost= 11,000 Buffer size = 1,002 pages

The lowest possible I/O costs given by the block oriented nested loop join with B (smallest relation) as the outer loop and all the pages of B in the buffer. However, having A in the buffer and in the outer loop would give the same cost but will need a larger buffer size.

Question 2

a) Compute the estimated result size and the reduction factor (selectivity)) of this query

RF(title= "CEO")=1/(Number of distant values in column) = 1/10
RF(level>15)= (Highest value – 15) / (Highest value- Lowest value)=(20-15)/(20-0)=5/20=1/4
Result Size = NTuples(R) * product of RF = 100,000 * 1/10 * ¼ = 2,500

b) Compute the estimated cost of the best plan assuming that a *clustered B+ tree* index on *(title, level)* is (the only index)) available. Suppose there are 200 index pages, and the index uses Alternative 2. Discuss and calculate alternative plans.

Cost = (NPages(INDEX) + NPages(R)) * product of RF(title, level)
Cost=(200+1,000) * 1/10 * ¼= 30

The only alternative left to do is use a heap scan (since no other indexes) on the relation, which will cost the number of pages in the relation, so the cost will be 1000 (since 1000 pages)

c) Compute the estimated cost of the best plan assuming that an *unclustered B+ tree* index on *(level)* is (the only index)) available. Suppose there are 200 index pages, and the index uses Alternative 2. Discuss and calculate alternative plans.

Cost = (NPages(INDEX) + NTuples(R)) * product of RF(level)
Cost = (200 + 100,000) * ¼ = 25,050

The only alternative left to do is use a heap scan (since no other indexes) on the relation, which will cost the number of pages in the relation, so the cost will be 1000 (since 1000 pages)

d) Compute the estimated cost of the best plan assuming that an *unclustered Hash* index on *(title)* is (the only index)) available. The index uses Alternative 2. Discuss and calculate alternative plans.

Cost = NTuples(R) * product of RF(title) * 2.2
Cost = 100,000 * 1/10 * 2.2 = 22,000

The only alternative left to do is use a heap scan (since no other indexes) on the relation, which will cost the number of pages in the relation, so the cost will be 1000 (since 1000 pages)

e) Compute the estimated cost of the best plan assuming that an *unclustered Hash* index on *(level)* is (the only index)) available. The index uses Alternative 2. Discuss and calculate alternative plans.

Has to do a heap scan since hash index cannot be used for ranges

Cost = NTuples(R)
Cost = 1,000

The only way is use a heap scan (since no other indexes) on the relation, which will cost the number of pages in the relation, so the cost will be 1000 (since 1000 pages) so no alternatives.

Question 3

a) Compute the estimated result size and the reduction factors (selectivity) of this query

E.did = D.did
RF=1/(MAX(Distinct values in E.did, Distinct values in D.did))= 1/5,000
D.did=F.did
RF=1/(MAX(Distinct values in D.did, Distinct values in F.did))= 1/5,000
E.sal>=59,000
RF(level>59,000)= (Highest value – 59,000) / (Highest value- Lowest value)
RF(level>59,000)=(60,000-59,000)/(60,000-10,000)=1/50
E.hobby = 'yodeling'
RF=1/ Distinct values in E.hobby = 1/200
Result SIze= 50,000 * 5,000 * 5,000 *1/5,000 *1/5,000 * 1/50 * 1/200 = 5

b)

1)Result Size of Dept + Finance = 5,000 * 5,000 * 1/5,000 =5,000 tuples = 50 Pages
Cost of joining Dept & Finance = NPages(Dept) + Npages(Dept) * Npages(Finance)
= 50 + 50 * 50
= 2550
Cost of joining with Emp = Npages(Result) * Npages(Emp) = 50 * 500 = 25,000
Therefore total cost = 25,000 + 2550 = 27,550

2) Result Size of Dept + Finance = 5,000 * 5,000 * 1/5,000 =5,000 tuples = 50 Pages

Cost of joining Dept & Finance = (NPages(Dept) + Npages(Finance)) *3
=(50 + 50)* 3
= 300
Cost of joining with Emp = Npages(Result) + Npages(Emp) + (4-1) *Npages(Result) +4* Npages(Emp)
/*4-1 since its already read before*/
= 50 + 500 + 3(50) + 4(500)= 4(50) + 5(500)= 2,700
Therefore total cost = 2,700 + 300 = 3,000

3) Result Size of Emp + Dept = 50,000 * 5,000 * 1/5,000 =50,000 tuples = 500 Pages
/*no need to sort Emp since its already sorted*/
Cost of joining Emp & Dept = 50+500 + 50 + 4 * 50 = 800
Cost of joining with Finance = Npages(Finance) + 2*(Npages(Result) + Npages(Finance))
= 50 + 2 *(500 +50) = 1150
Therefore total cost = 800 + 1,150= 1,950

4) Result Size of Emp + Dept = 50,000 * 5,000 * 1/5,000 =50,000 tuples = 500 Pages
Cost of joining Emp & Dept = 3*(Npages(Emp) + Npages(Dept))= 3*(500+50)= 1650
Cost of joining with Finance = Npages(Finance) + Npages(Result) = 50* 500 = 25000
Therefore total cost = 1650 + 25000 = 26650