Database Systems, CSCI 4380-01
Homework # 11 Answers

Question 1. What is the cost of following operations given below:
$PAGES(R) = 8,000$ $PAGES(S) = 100,000$.

(a) Sort R using $M = 100$ blocks.

Answer. Step 1: Read/write once, create 8,000/100= 80 sorted groups. Cost = 16K pages.

Step 2: 80 groups can be merged using 100 blocks, read once, merge and output. Cost = 8K pages.

Total cost = 24K pages

(b) Sort S using $M = 100$ blocks.

Answer. Step 1: Read/write once, create 100,000/100= 1000 sorted groups. Cost = 200K pages.

Step 2: 1,000 groups cannot be merged using 100 blocks, read/write once, merge 1,000 groups to get 1000/100=10 sorted groups. Cost = 200K pages.

Step 2 (repeated): 10 groups can be merged using 100 blocks, read once, merge and output. Cost = 100K pages.

Total cost = 500K pages

(c) Sort merge join $R \bowtie S$ using $M = 100$ blocks, assuming that if the second stage of the sort requires less than M blocks, then it can be combined with the merge-join step.

Answer.

Naive answer: Cost of sorting R + Cost of sorting S + reading R and S once more = 24K + 500K + 108K = 632K

However, R has 80 sorted groups after step 1 and S has 10 sorted groups after Step 2. We can combine the last sort+merge steps with a join using 90 blocks (less than 100 allocated).

Cost: R (read/write/read) + S (read/write/read/write/read) = 3*8K + 5*100K = 524K pages

(d) Block-nested loop join $R \bowtie S$ using $M = 101$ blocks.

Answer. Read R once into 100 blocks and S : 8000/100=80 times.

Cost = 8K + 80 * 100K = 8008K pages.

(e) Hash based join $R \bowtie S$ using $M = 100$ blocks. Assume both R and S are hashed using a hash function that distributes the tuples to hash buckets uniformly. Each hash bucket is joined using block-nested loop join.

Answer.

Hash R: Cost = 16K, each bucket has 8000/100=80 pages.

Hash S: Cost = 200K, each bucket has 100,000/100=1,000 pages.

Join each bucket of R with each bucket of S: Since 80 pages fit in memory, we can read a full bucket of R into memory into 80 blocks and read the correspoing S bucket using the remaining memory (one bucket a time but only once). Hence, the join requires only one read of R and S.

Total cost = 16K + 200K + 8K + 100K= 324K

Question 2. What is the cost of following index scan operations given the information below:

```
PAGES(S)=100,000  TUPLES(S)=4,000,000
Index I1 on S(B) with 5,000 leaf nodes and 3 levels (root/internal/leaf)
Index I2 on S(B,C) with 20,000 leaf nodes and 3 levels (root/internal/leaf)
Index I3 on S(C) with 5,000 leaf nodes and 3 levels (root/internal/leaf)
Index I4 on S(C,A,B) with 30,000 leaf nodes and 3 levels (root/internal/leaf)

Q1: SELECT A,B,C FROM S WHERE B>21 AND C=12180;
Q2: SELECT B FROM S WHERE C=12180;
Q3: SELECT A,B FROM S WHERE A LIKE 'A%' AND B>21 AND C=12180;
Q4: SELECT D FROM S WHERE A LIKE 'A%' AND C=12180;
Q5: SELECT A FROM S WHERE B>21;
```

Condition	Number of tuples expected
`A LIKE 'A%'`	8,000
`B>21`	200,000
`C=12180`	80,000
`A LIKE 'A%' AND B>21`	400
`B>21 AND C=12180`	4,000
`A LIKE 'A%' AND C=12180`	160
`A LIKE 'A%' AND B>21 AND C=12180`	8

Fill out the following table to estimate the cost of different query plans. If a certain index does not match a query (no conditions on the indexed attributes), simply write `N/A` in that cell.

The cost of an index scan includes the cost of scanning the index (internal and leaf nodes) and the cost of reading the matching tuples from disk. For the matching tuples, assume each tuple is in a different disk page in the worst case.

Answer. Compute for each index, tuples per leaf node:

I1 and I3: 4,000,000/5,000 = 800

I2: 4,000,000/20,000 = 200

I4: 4,000,000/30,000 = 134

	Q1	Q2	Q3	Q4	Q5
Cost of sequential scan	100K	100K	100K	100K	100K
Condition for scanning I1	B>21	N/A	same as Q1	N/A	B>21
Cost of index scan using I1	2+250+100,000	N/A	same as Q1	N/A	same as Q1
Condition for scanning I2	B>21	True	same as Q1	True	B>21
Cost of index scan using I2	2+1000+4000	2+20000	same as Q1	2+20,000+80,000	2+1000+100,000
Condition for scanning I3	C=12180	C=12180	same as Q1	C=12180	N/A
Cost of index scan using I3	2+100+80,000	2+100+80,000	same as Q1	2+100+80,000	N/A
Condition for scanning I4	C=12180	C=12180	A LIKE 'A%' AND C=12180	same as Q3	True
Cost of index scan using I4	2+602+0	2+602+0	2+2+0	2+2+160	0+30,000

NOTE: When reading the matching 200,000 tuples for B>21, we know that the tuples cannot be in more than 100,000 pages. However, we will read pages as the index returns, which may lead to reading the same page more than once. Hence, I will accept 100K or 200K both as potential answers with proper explanation.

In all the index cost computations, you can add +1 to the cost as the a range may scan one extra leaf node. That is also an acceptable answer. Obviously as far as the optimizer is concerned, we only care about orders of magnitude difference, so plus 1 is not important.

Question 3. What is the estimated number of tuples expected in the output of the following queries given the following statistics (note that VALUES is DISTINCT values for that attribute):

```
TUPLES(T)=200,000     PAGES(T) = 10,000
TUPLES(V)=10,000,000  PAGES(V) = 20,000

Q1: SELECT * FROM T WHERE T.X=10 ;
Q2: SELECT * FROM T WHERE T.Y>100 AND T.Y<200 ;
Q3: SELECT * FROM T WHERE T.X<>10 OR T.Y=20;
Q4: SELECT * FROM V WHERE V.X=10 ;
Q5: SELECT * FROM V,T WHERE V.X=T.X ;
Q6: SELECT * FROM V,T WHERE V.Y=T.Y ;
Q7: SELECT * FROM V,T WHERE V.Y=T.Y AND T.Z='ABC123' ;
Q8: SELECT * FROM V,T WHERE V.Y=T.Y AND V.X=T.X ;
```

Attribute	VALUES	Minval	Maxval
T.X	20,000	1	20,000
T.Y	2,000	1	4,000
T.Z	5,000	'AAA111'	'ZZZ999'
V.X	15,000	1	20,000
V.Y	3,000	1	4,000

Answer.

Q1: 200,000 * 1/20,000 = 10

Q2: 200,000 * 100/4,000 = 5,000

Q3: 200,000 * (1 - (1-(1-(1/2000))) * (1-1/4,000))) = 199,900

Q4: 10,000,000 * 1/15,000 = 666

Q5: 10,000,000 * 200,000 * 1/20,000 = 100,000,000

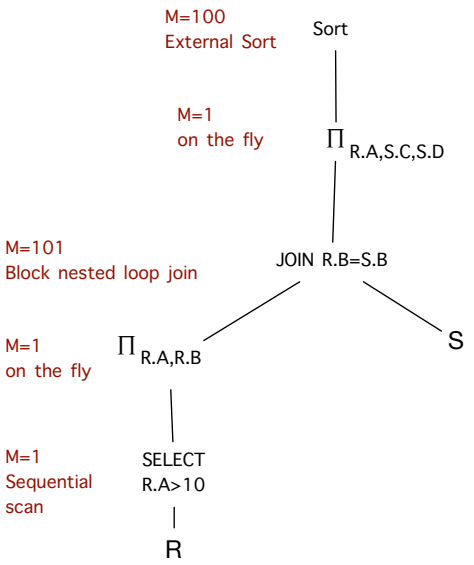
Q6: 10,000,000 * 200,000 * 1/3,000 = 666,666,666

Q7: 10,000,000 * 200,000 * 1/3,000 * 1/5,000 = 133,333

Q8: 10,000,000 * 200,000 * 1/3,000 * 1/20,000 = 33,333

Question 4. What is the total cost of the following query plan? The size of each relation after a specific operation is given below. Show the cost of each operation and explain with a single sentence.

Relation	Size in Pages
R	10,000
$\sigma_{R.A>10}(R)$	2,000
$\Pi_{R.A,R.B}(\sigma_{R.A>10}(R))$	80
S	20,000
$(\Pi_{R.A,R.B}(\sigma_{R.A>10}(R))) \bowtie_{R.B=S.B} S$	5,000
$\Pi_{R.A,S.C,S.D}(\Pi_{R.A,R.B}(\sigma_{R.A>10}(R))) \bowtie_{R.B=S.B} S$	400



Answer.

Read R once, do selection/projection to fill the 100 blocks of join.

Read S once (after selection/projection, R fits in 80 blocks).

Join = 10,000 (R) + 20,000 (S) + 0 (projection)

Sort:

Step 1: The join result after projection is 400 pages. It is already produced in memory, so there is no read cost to the first step. Sort will fill 100 pages at a time, sort and write to disk. Total = 400 pages written and 4 sorted groups are generated.

Step 2: Merge 4 groups and output, cost = 400 pages.

Total = 30,000 (join) + 800 (sort)

Question 5. Suppose you are given the following schedules. List all the conflicts and check whether they are serializable or not. Show your work.

(a) $S_1 : r_1(x)\ r_2(z)\ r_1(y)\ w_2(w)\ w_2(z)\ r_3(z)\ r_3(x)\ w_1(x)\ w_3(x)\ r_1(w)\ w_1(y)\ w_3(z)$

(b) $S_2 : r_1(x)\ r_1(y)\ r_2(z)\ r_3(z)\ r_3(x)\ w_1(x)\ w_3(x)\ w_2(w)\ w_2(z)\ r_1(w)\ w_1(y)\ w_3(z)$

Answer.

S1 conflicts:

$r_1(x)\ w_3(x)$
$r_2(z)\ w_3(z)$
$w_2(w)\ r_1(w)$
$w_2(z)\ r_3(z)$
$w_2(z)\ w_3(z)$
$r_3(x)\ w_1(x)$
$w_1(x)\ w_3(x)$

Conflict graph: $T1 -> T3, T3 -> T1, T2 -> T1, T2 -> T3$. There is a cycle between T1 and T3, hence this schedule is not serializable.

S2 conflicts:

$r_1(x)\ w_3(x)$
$r_2(z)\ w_3(z)$
$r_3(z)\ w_2(z)$
$r_3(x)\ w_1(x)$
$w_1(x)\ w_3(x)$
$w_2(w)\ r_1(w)$
$w_2(z)\ w_3(z)$

Conflict graph (with many cycles): $T1 -> T3, T3 -> T1, T2 -> T1, T2 -> T3, T3 -> T2$.

It is not serializable.

Question 6. You are given the following schedule:

$$S : r_1(x)\ r_1(y)\ r_2(y)\ r_3(x)\ r_3(z)\ w_3(z)\ commit_3\ w_1(x)\ commit_1\ w_2(y)\ commit_2$$

Is this schedule possible using Strict Two Phase Locking with shared and exclusive locks? If yes, show which transactions obtain which locks when. If no, explain why a necessary locks cannot be obtained for this schedule to occur.

Answer.

Yes, it is possible. T3 shares x locks and locks z alone, then releases all locks. T1 can now get X lock on x, and T2 can get a lock on y as no one else is locking.