CMPT 454 Assignment 4: Query Optimization 2

This assignment is worth approximately 7% of your final grade. Marks are shown in []s.

Question 1

Answer the questions that follow about the SQL query shown below. [10]

```
SELECT p.lastname, p.income, p.birthdate, op.sin, op.opdate
FROM Patient p, Operation op, Doctor doc
WHERE (p.city = 'Vancouver' OR p.lastname = 'Smith')
AND (p.city = 'Vancouver' OR p.firstname = 'bob')
AND p.msp = op.msp AND op.sin = doc.sin
AND (op.description = 'appendectomy' OR op.description = 'amputation')
AND doc.doctorname <> 'Smith' AND gender = 'F')
```

Sketch a relational algebra expression tree that represents an equivalent logical plan to your answer to the SQL query shown below. Your query should be derived by using the following heuristics:

- Replace Cartesian products with joins
- Push selections as far down the tree as possible
- Push projections as far down the tree as possible, except after selections, add attributes to the projections where necessary to achieve this
- Group join operators

Question 2

You are to estimate the cost of the relational algebra query shown below.

$\pi_{\text{cname,cemail,vname,vcity,ceo,pname,ptype,description,sdate}}($
$\pi_{\text{vname,vcity,ceo}}(\sigma_{\text{vcountry='Canada'} \wedge \text{employees} > 500}(\text{Vendor})) \bowtie$
$\pi_{\text{pid,pname,description,ptype}}(\text{Product}) \bowtie$
$\pi_{\text{cid,cname,cemail}}(\sigma_{\text{ccity = 'Vancouver'} \vee \text{ccity = 'Edmonton'}}(\text{Customer})) \bowtie \text{Sales})$

Indexes

- Secondary B+ tree on {*vcountry*, *vcity*}, of height 3 in Vendor
- Primary B+ tree on {*ccity*, *cstreet*, *cnumber*}, of height 4 in Customer
- Linear hash index on cid in Sales, no bucket consists of more than one block

Relation Statistics

	Customer	Product	Vendor	Sales
T(R)	200,000	100,000	50,000	10,000,000
B(R)	25,000	10,000	5,000	500,000
V(R, cid)	200,000			200,000
V(C, ccity)	100			
V(R, pid)		100,000		100,000
V(R, vname)		50,000	50,000	
V(V, vcountry)			5	
V(V, employees)			5,000	

- Primary B+ tree index of on pid in Sales
- For B+ tree indexes assume that the root node of the index is retained in main memory
- Records of all tables, including intermediate relations. should not span more than one block

Attribute, Block and Main Memory Sizes

- cid, pid, employees – 4 bytes
- ptype, sdate – 8 bytes
- cname, ccity, pname, vname, vcity, ceo, vcountry – 16 bytes
- cemail – 32 bytes
- description – 64 bytes
- Block (and main memory frame) – 4,096 bytes
- There are 600 main memory frames available for the query

a) Estimate the cost for performing the selection and projection on Vendor, do not include the cost for writing out the result (if any): $\pi_{vname,vcity,ceo}(\sigma_{vcountry='Canada' \wedge employees > 500}(\text{Vendor}))$. The minimum and maximum values for the *employees* attribute are 0 and 5,000. [2]

b) State the size of the Vendor selection and projection result in records and bytes. [2]

c) Estimate the cost for performing the selection and projection on Customer, do not include the cost for writing out the result (if any): $\pi_{cid,cname,cemail}(\sigma_{ccity = 'Vancouver' \vee ccity = 'Edmonton'}(\text{Customer}))$. Histograms kept by the DB indicate that 25% of customers live in Vancouver and 15% live in Edmonton. [2]

d) State the size of the Customer selection and projection result in records and bytes. [2]

e) State the size of the Product projection result in records and bytes. [2]

f) Determine the most efficient join order using the dynamic programming approach described in class*. The metric you use to determine join order should be the least number of *records*. You should not consider the actual cost of performing the join in your calculation, or the cost of producing input for the join. You should only consider left-deep join trees. [10]

*You can do this informally, and do not have to submit the tables shown in the class example (and Exercise 9). Don't forget that the size in records of some of the tables is affected by previous operations.

Your answer should consist of the following:

- The order in which relations are to be joined. *e.g.* $(((A \bowtie B) \bowtie C) \bowtie D)$
- The total cost estimate (the sum of the intermediate relation sizes) for your chosen join order, broken down by join. *e.g. Total cost: AB cost + ABC cost = total cost*
- The size in *records* of each of the three joined relations. *e.g. AB, ABC and ABCD size*
- The size in *blocks* of each of the three joined relations. *e.g. AB, ABC and ABCD size*

g) For the first of the three joins specified in part (f) you are to calculate the cost in reads and writes of performing the join. [10 divided between g, h and i]

Note that:

- Your cost estimate should only include the cost of performing the join and should not include the cost of previous operations or the cost of writing out the result.
- Your cost estimate should not be the same as the result of part (f) – the value computed in (f) is used only to determine the join order.
- You should only consider three join algorithms: *block nested loop join*, *hash join* and *index nested loop join* ***and must state the cost for each of these algorithms***. If an index nested loop join is not possible (because there is no relevant index) state this.

h) For the second of the three joins specified in part (f) you are to calculate the cost in reads and writes of performing the join. Read the description in part (g) for more information.

i) For the third of the three joins specified in part (f) you are to calculate the cost in reads and writes of performing the join. Read the description in part (g) for more information.

j) Estimate the total cost of performing the query. This is mostly adding up the costs from previous parts of the question, but you should note whether or not operations can be pipelined. If not, you should include the cost of materializing the results of intermediate operations. [6]

Assessment

The assignment is out of 46. Marks are assigned as follows:

- Question 1 – 10
- Question 2 – 36

Submission

You should submit your assignment online as a single .pdf file. The assignment is due by 11:59pm on Friday July 24th.