CMPT 454 Assignment 3: Query Evaluation

This assignment is worth approximately 7% of your final grade.

Question 1

For each of the operations described below you are to calculate the number of disk reads (and writes) to perform the operation in the most efficient way. Do not include the cost to write out the result of the operation. Briefly explain the process of each operation and its components.

For example, *use the index on a, read x nodes of the index and y records of the file*.

Assume the root node of tree indexes and directory of extensible hash indexes are *not* held in main memory at the start of the operation. [2 marks each, unless noted otherwise]

Product = {<u>*pid*</u>, *pname*, *ptype*, *pnumber*, *description*, *manufacturer*, *mcountry*, *mcity*, *price*}

Table Statistics (note for all V the table is the Product table)

B(Product)	**T(Product)**	**V(*pid*)**	**V(*pname*)**	**V(*ptype*)**	**V(*pnumber*)**
25,000	250,000	250,000	50,000	500	1,000
V(*mcountry*)	**V(*mcity*)**	**V(*manufacturer*)**	**V(*price*)**	**V(*description*)**	
40	1,000	10,000	50,000	125,000	
V({*ptype*, *pnumber*})		**V({*mcountry*, *mcity*})**			
250,000		2,500			

Indexes

- Primary dense B+ tree index on {*ptype*, *pnumber*} where *ptype* is the prefix of the search key. The index is height 4 (includes leaf and root level) and interior and leaf nodes contain 50 search keys on average.
- B+ tree index on {*mcountry*, *mcity*} where *mcountry* is the prefix of the search key. The index is height 5 (includes leaf and root level) and interior and leaf nodes contain 30 search keys on average.
- B+ tree index on *pname*. The index is height 4 (includes leaf and root level) and interior and leaf nodes contain 60 search keys on average.
- Extensible hash index on *pid*. The directory resides on two disk blocks and each bucket contains 100 search keys on average.
- Linear hash index on *manufacturer*. Each bucket contains 40 search keys on average – there are no overflow blocks.

a) $\sigma_{(\text{ptype = 'ABC'} \wedge \text{pnumber = 13})}$ (Product)

b) $\sigma_{(\text{ptype = 'ABC'})}$ (Product)

c) $\sigma_{(\text{pid = 123456} \vee \text{pid = 678324})}$ (Product)

d) $\sigma_{\text{(mcountry = 'UK'} \wedge \text{mcity = 'Sheffield')}}$ (Product)

e) $\sigma_{\text{(mcity = 'Detroit')}}$ (Product)

f) $\sigma_{\text{(pname = 'foo345'} \vee \text{manufacturer = 'acme')}}$ (Product)

g) $\sigma_{\text{(mcountry = 'Canada'} \wedge \text{price > 25.00} \wedge \text{description = 'sweet widget)}}$ (Product)

h) $\sigma_{\text{(pname = 'bar111'} \vee \text{price = 73.80} \vee \text{ptype = 'TLR')}}$ (Product)

i) $\sigma_{\text{((pname = 'foo17'} \vee \text{price = 19.99)} \wedge \text{(ptype = 'HJK'} \vee \text{price = 19.99)} \wedge \text{(ptype = 'HJK'} \vee \text{pid = 432911))}}$ (Product)

j) Sort the Product table assuming there are 100 main memory frames available

k) $\pi_{\text{(ptype, description)}}$ (Product) – assume there are 50 main memory frames available, and that duplicates are *not* to be removed

l) $\pi_{\text{(ptype, description)}}$ (Product) – assume there are 50 main memory frames available, and that duplicates are to be removed using sort projection; also assume that duplicates are only encountered in the final stage of the process [4 marks]

Note that *ptype* is 4 bytes, *description* is 96 bytes and pages are 4,096 bytes

Question 2

Answer the following questions about performing a natural join between the *Patient* and *Visit* tables. The only attribute the two tables have in common is *msp*, which is the primary key of *Patient* and a foreign key in *Visit*. Relevant information is shown to the right. [15]

	Visit	Patient
T(R)	1,800,000	180,000
B(R)	180,000	18,000
V(R, msp)	180,000	180,000

a) How many records will the joined relation contain? [1]

b) Approximately how many records of the joined relation fit on a single block? [1]

c) What is the main memory requirement (in frames) to perform the join in two passes using the *sort-join* algorithm? Briefly explain your calculation. [2]

d) If *Patient* was already sorted on *msp* would your answer to part (**c**) change? Explain why or why not? [2]

e) What is the main memory requirement (in frames) to perform the join in two passes using the *hash-join* algorithm? Briefly explain your calculation. [2]

f) Assume that there are approximately 1,200 frames available for performing the join; what is the cost of performing the join using the *block nested loop join* algorithm? Briefly explain your calculation. [3]

g) Assume that there is a secondary extensible hash index on *msp* in *Patient*. If the directory page of the index is retained in main memory, what is the cost of performing the join using the *index nested loop join* algorithm? Briefly explain your calculation. [3]

h) Assume that there is a primary B+ tree index of height 3 on *msp* in *Visit*. If the root node of the index is retained in main memory, what is the cost of performing the join using the *index nested loop join* algorithm? Briefly explain your calculation. [3]

i) Assume that there are just over 4,000 frames available for performing the join; what is the cost of performing the join using the *hybrid hash join* algorithm where one partition of the outer relation is to be retained in main memory? Briefly explain your calculation. [3]

Assessment

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

- Question 1 – 26
- Question 2 – 20

Submission

You should submit your assignment online as a single .pdf file. The assignment is due by 11:59pm on Wednesday July 8th.

John Edgar (johnwill@sfu.ca)