School of Computer Science, McGill University

COMP-421 Database Systems, Winter 2022

Written Assignment 2: Query Evaluation

Due Date March 24, 12:00 noon EST

This is an individual assignment. You are required to work on your own to create the solution.

This assignment is worth 15% of your course grade. The total points in this assignment is 45.

Please read the complete assignment description, including the **Hints and Guidelines** section before starting your work. There are important instructions contained in them.

SUBMISSION FORMAT:
Submit your solution as **assignment2.pdf**. Unless stated elsewhere in the question description, no handwritten components are allowed in your solution. Most of your solutions involve writing equations and simple math computations that can be easily typed into a document. Handwritten equations and numbers are often difficult for the TAs to evaluate accurately for which points can be lost. There will be a 25% penalty if this is not followed. Where handwritten components are allowed (relational algbera expression and execution tree), you are allowed to (if you do not want to type or draw) hand write / draw and include the image into the document (or upload as a separate file and include the name in the document). However, all such handwritten submissions must be very legible. This is not an exam, there is sufficient time and therefore, you shouldn't be submitting works that are scrached and smeared all over, which are difficult for the TAs to evaluate. You maybe familiar with your hand writing, others are not. If TAs have a hard time to read your "script", you will not get points.

In this assignment we evaluate some queries used by an application of a business reviews website. We look at the three relations whose schema and a sample record is given below:

Business (bid CHAR(10), btype VARCHAR(30), bname VARCHAR(60), bcity VARCHAR(30)
,baddr VARCHAR(60), bprovince CHAR(2), bstartdate DATE)
'R821213412', 'restaurant', 'Quick Bytes Cafe', 'Montreal', '150 Rue Raven', 'PQ', '2014-05-20'
Users (uemail VARCHAR(45), uname VARCHAR(45), udateofbirth DATE, ujoindate DATE, uprovince CHAR(2))
, 'caffineaddict@rbemid.com', 'Ashley Richards', '1989-04-25', '2019-06-30', 'ON'
Reviews (reviewid INT, bid CHAR(10), uemail VARCHAR(45), rstars SHORT, rtext VARCHAR(600), rdate DATE)
, bid references Business, uemail references Users.
, 51233, 'C89456729', 'marion23@rfemid.com', 4, 'The croissants are good.', '2020-06-21'

INT occupies 8 bytes and each CHAR is 1 byte (i.e., a CHAR(10) attribute would occupy 10 bytes). Unless mentioned otherwise, assume for all VARCHAR attributes that the average size is 2/3 the total capacity allocated for that attribute (e.g `btype` is on an average 20 bytes). DATE occupies 10 bytes. SHORT occupies 1 byte.

There are 20,000 businesses and 90,000 users. There are 100 different types of businesses (`btype`). Assume that each user writes on an average 20 reviews. The database has 4 full years of reviews (years 2018 - 2021, 365 $\times$ 4 days - we will pretend 2020 is not a leap year for simplicity). 2% of the reviews are the same user reviewing the same business a second time (we will assume that there is no third review, etc.). The value of `rstars` range 1-5. Their distribution is as follows.

- 15% reviews have 5 stars
- 10% reviews have 1 stars
- Remaining reviews are equally distributed across 2,3,4 stars.

There are 1825 users whose date of birth falls in the year 1990. Provinces have 10 possible values.

In general, all database data pages are 4000 bytes with on an average 75% fill factor. For indexes, a single data entry may not be spread over more than one leaf page. Each rid has 10 bytes and each internal pointer has 8 bytes. Leaf/intermediate pages are filled on an average of 75%. Index page are also 4000 bytes. The root may have any fill factor. Assume that the root and all intermediate nodes of an index are in memory. For all questions, assume 100 free buffer frames in memory. If it is convenient, you may assume a couple more for simplifying your calculations. Minor rounding errors are acceptable. **However, keep in mind that it is important you write down the correct steps. If you got a close enough solution by chance eventhough your steps and equations are wrong, you will still lose points**.

Do not assume any indexes other than the ones mentioned for the specific questions. (Index mentioned in one question does not automatically carry over to the next one). You can also assume that the root and intermediate nodes of an index is already available in memory (and not counted towards our free memory pool of 100 buffers).

Tip:- Most of the questions below will require the number of records for the tables in our model. The information needed to compute these is given in the description above.

Ex. 1 — (11 Points)

A typical query would be the one below, which searches the `Reviews` table for all the reviews pertaining to a business `B`, with a stars rating of at the least `N`. Here `B` and `N` represent (valid) parameters that might differ for each execution.

```
SELECT *
FROM Reviews
WHERE bid = B AND rstars >= N
```

(a)(3 Points) Find the number of data pages to store each of the 3 tables.

(b)What is the I/O cost of the above query when:

(i)(1 Points) You use an unclustered type I index on bid?

(ii)(4 Points) You use a clustered type II index on bid?

(iii)(3 Points) You use an unclustered type II multi-attribute index on (bid, rstars)?

Turn in:- Your solution (typed) in **assignment2.pdf**, make sure to write down each sub question number, etc. In some cases you maybe able to compute a "concrete" number as the solution, in other cases it will be an equation that depends on the actual values used in the query.

Ex. 2 — (0 Points)

Note:- This is a warmup problem, please do not turn in the solution to this or cross reference solution steps in this problem. This will not be graded.

Given an unclustered type II index on `bid` of `Reviews`, calculate the estimate I/O and give an estimate of the number of output tuples for:

(a)index nested loop join between `Reviews` and `Business`.

(b)block nested loop join between `Reviews` and `Business` and `Business` is the outer relation.

(c)block nested loop join between `Reviews` and `Business` and `Reviews` is the outer relation.

(d)sort merge join between `Reviews` and `Business` (assume relations are not already sorted on the join attribute). Hint:- Remember we used a technique in class to reduce the cost of merge-join so that it is less than the sum of the individual costs of merge sort and join.

Ex. 3 — (19 Points)

Consider the following query:

```
SELECT B.bname, B.bprovince, U.uemail, R.rstars, R.rdate
FROM Users U, Reviews R, Business B
WHERE U.uemail = R.uemail AND R.bid = B.bid
  AND U.udateofbirth BETWEEN '1990-01-01' AND '1990-12-31' AND B.btype = 'restaurants'
```

Referring to the `Business` table as `B`, the `Reviews` table as `R`, and the `Users` table as `U`, a non-optimized relational expression for this query is:

$\rho(T_1, \sigma_{U.udateofbirth \geq '1990-01-01' \wedge U.udateofbirth \leq '1990-12-31'}(\sigma_{B.btype='restaurants'}(B \times U \times R)))$
$\rho(T_2, \sigma_{U.uemail=R.uemail \wedge R.bid=B.bid}(T_1))$
$\Pi_{B.bname,B.bprovince,U.uemail,R.rstars,R.rdate}(T_2)$

(a)(3 Points) Optimise this expression according to the rules of **algebraic optimisation** discussed in class. Your answer need not take into account data cardinalities for this sub-question, and you need not draw a tree.

Turn in:- You may type this solution / include this as a screen shot, or a scan/photo of handwritten relational algebra expression.

(b)(16 Points)

Given an unclustered Type I index on `bid` on the `Business` table, give an execution plan for your expression above and indicate how to execute each operator. Draw the execution plan tree. You need not include the number of rows and bytes passing through the operators in the tree. Give a rough estimation of the best I/O cost for your execution plan. Do not assume any other indexes on the tables. While you do not have to re-do the relational algebra expression that you wrote in the previous step to come up with the most optimal execution plan, remember to pick the most optimal approach for each individual operation themselves in the tree. I.e., if the tree has a join operator for t_1 being joined with t_2, pick the best approach to do that specific join that will reduce the cost of that step. Also take into account if adjacent steps can be combined/pipelined.

Turn in:- The steps, discussion and any computations must be typed in the **assignment2.pdf**. For the execution plan tree, you may draw / include as a screen shot, or a scan/photo of hand-drawn execution plan tree.

Note:- Remember to apply the techniques we saw in class such as pipelining and "combining" operations (projection with join, etc) when useful, to reduce the overall I/O cost.

Ex. 4 — (15 Points)

Consider the following query that computes some review statistics for each business over the reviews written since 2021 December.

```
SELECT R.bid, U.uprovince, AVG(R.rstars), COUNT(DISTINCT R.uemail)
FROM Reviews R, Users U
WHERE R.uemail = U.uemail
  AND R.rdate >= DATE '2021-12-01'
GROUP BY R.bid, U.uprovince
ORDER BY U.uprovince, R.bid
```

Find the optimal execution plan and its cost. Also draw the execution plan tree.
Turn in:- The steps, discussion and any computations must be typed in the **assignment2.pdf**. For the execution plan tree, you may draw / include as a screen shot, or a scan/photo of hand-drawn execution plan tree.

Hints & Guidelines

- You do not have to "budget" for slot directory, sibling pointers, row header, etc., when doing your page size calculations - only data entries, index entries and record sizes need to be considered.
- Assume uniform distribution of values, unless the information given indicates otherwise.
- You can use the information (record length, pages, etc.) computed in one Exercise (except Ex.2, which is not graded), in the following exercises.
- Use pipelining when possible to save I/O cost between multiple steps (operators) in the execution plan. Not doing this could result in points lost (depending on how straightforward it would have been to integrate).

- When calculating I/O, take into account the I/O cost savings due to pipelining in your equations.
- You can leave fractions as it is, if you would like to (as in number of data entries per page, records per page, etc.). However, keep in mind that there are some steps that cannot produce fractions. For example, if you have a specific block nested join step in your execution plan, its cost cannot be 24.6 pages, it has to be 27 pages - because that is the unit in which we do I/O (same with memory buffers).
- Minor rounding errors due to approximating data entries per page, records per page, etc., are acceptable. In general look at the magnitudes of the numbers you are rounding and how big a number you are multiplying it with later to see how significant the impact is. (for example, rounding 1.5 to 2 and rounding 1000.5 to 1001 introduces very different magnitude shifts to the numbers with which they are multiplied later). At the end of the day, remember that our objective is to have a reasonably good computational methodology to help us choose between multiple possible execution plans based on their costs.
- When making execution plans, remember that any temporary intermediate pages, etc., that you create as part of query execution can be packed near 100% to maximally use the memory/disk and reduce I/O cost. Do not leave them at the fill factor of regular database pages (of tables and indexes) that are usually below 100%.
- Partial points are given to steps and methodology even if you making errors doing math. But try to use a calculator to do your math to avoid making mistakes.

What to turn in

assignment2.pdf that contains the answers to all the questions in the assignment description, numbered appropriately. This is the only acceptable file format. There will be no accommodation for submitting incorrect files.

Questions ?

Please use **Ed** for any clarifications you need (`assignments` → `a2`). Do not email the instructor or TAs as this leads to a lot of duplicate questions and responses (not an efficient system). Such emails will not receive any replies.

Please check the pinned post "A2 general clarifications" in Ed before you post a new question. It might have been already addressed there, in which case we will not address it again.

Questions about general clarifications must be marked public (as other students will also benefit from this and may even have a valid response). TAs and Instructors upon their discretion may toggle any private posts into public mode for the benefit of the student population at large.

There will be specific office hours for the assignment that will be announced closer to the due date.

Extensions and Late submissions

- Remember, your submission is **due on March 24th 12:00 noon**. There is no place for excuses.
- A maximum of 1 day of late submission is allowed with a penalty of 20% of the achieved grade per day (rounded up, even for a minute).
- Penalty waivers are granted only for medically documented emergencies and under any circumstances **will not be granted unless requested 24 hours before the due**. I expect you to be better organized and get the job done ahead and leave the last 24 hours only for one last final check.
- There is no "partial late penalty" concept, it is applied to your entire submission, and not just specific questions that you submitted late.