HW 6
CS 6083 sections A and INET, Fall 2024, Prof Frankl

Consider this relational database schema:

Course(courseID, title, …)
Section(sem, year, courseID, secID, building, room,...)
Takes(year, sem, courseID, secID, sID, grade)

Assume records and attributes have the following sizes:
Each Course record is 1KB, of which 8 bytes is the courseID
Each Section record is 512 bytes, of which the courseID,secID is 8 bytes
Each Takes record is 64 bytes, of which the sID, secID and courseID are each 8 bytes

Assume the following statistics about the data:
(You may approximate 1KB = as 1000 bytes or may round the powers of 10 off to nearby powers of 2 if you prefer, using 1000 is about 2^{10})

About 10,000 (=10^4) *course* records, each 1 KB ⇒ 10 MB
About 1,000,000 (=10^6) *section* records, each 0.5 KB ⇒ 500 MB
About 40,000,000 (= 4* 10^7) *takes* records, each 64 B ⇒ 2560 MB

Assume that the Block size = 4K **and consider** B+ trees where each tree node fits into a block RIDs (pointers to tree nodes and to records on disk) are 8 bytes.

Assume that there is a little more than 100 MB of main memory available (enough for 100 MB of the outer relation in nested joins plus at least one block for the inner relation, at least one block of output buffer space, and a few blocks for additional temporary storage (e.g for small temporary relations).

Assume that the seek time is 10ms and the transfer rate is 50MB/s

As in HW 5, problem 2, **assume** that student 12345 has taken 10 courses.
Also **assume** that half of the courses are in the JAB building

Problem 1

1. Write an SQL query to find the title of each course the student with ID 12345 took in the “JAB” building and their grades in those courses.

2. Draw an expression tree illustrating the following relational algebra query, with the operations done in the order indicated by the parentheses:

$\Pi_{title,grade}(\sigma_{sID=12345 \wedge building="JAB"}((section \bowtie takes) \bowtie course)$

[You may omit year and sem in the join and just join using courseID for simplification.]

3. Draw an expression tree illustrating transformation of the relational algebra query to push the selects and projects down (as early as possible) in the evaluation
4. First we'll consider a query plan that does not use any indexes. Analyze the cost of the following query plan noting the following for each step

    a. The part of the expression being evaluated
    b. How joins are being performed (e.g. whether either of the relations fit into memory; if not which is the outer relation in block-nested loop join and how many passes are needed)

    c. the number of seeks and total time for these seeks
    d. the number of block transfers or total amount of data transferred and time for this data transfer
    e. The size of results that are either final results or are passed onto the next phase
    f. Whether these intermediate results can stay in memory or need to be written out
    g. Summary of time required

Query Plan:

1. Scan Takes to find courses taken by student 12345 and their grades
2. Join result with Section, selecting sections in JAB building while doing the join
3. Join result with Course. (Projections can be done along with the joins; No duplicate elimination is needed.)

Problem 2

Now consider the secondary (unclustered) B+ tree index on Takes from Problem 2 of HW 5.

1. How is this index potentially helpful?
2. Analyze the cost of using the index and note which step(s) above are affected.
3. Is using the index beneficial for this query?

Problem 3 and 4 [Recommended but not Required]

Repeat Problems 1 and 2 for the query
SELECT * FROM Takes NATURAL JOIN Section WHERE year = 2024
assuming that 10% of the Takes records are from 2024