COMP2411 Fall 2023 Class Exercise 8

Student Name: ______________________________

Student ID: ______________________________

Question 1. Consider the following relational schemas and instances:

```
Student (SID, Name, Gender, Major)

Student_Dir (SID, Address, Phone)
      FK: (SID) → Student (SID)

Course (Course_No, Name, Level)

Course_Taken (Course_No, Term, SID, Grade)
      FK: (Course_No) → Course (Course_No); (SID) → Student (SID)
```

Student

SID	Name	Gender	Major
123	John	M	CS
124	Mary	F	CS
126	Sam	M	CS
129	Julie	F	Math

Student_Dir

SID	Address	Phone
123	333 Library St	555-535-5263
124	219 Library St	555-963-9635
129	555 Library St	555-123-4567

Course

Course_No	Name	Level
CS1520	Web Programming	UGrad
CS1555	Database Management Systems	UGrad
CS1550	Operating Systems	UGrad
CS1655	Secure Data Management and Web Applications	Ugrad
CS2550	Database Management Systems	Grad

Course_Taken

Course_No	Term	SID	Grade
CS1520	Fall 2022	123	3.75
CS1520	Fall 2022	124	4
CS1520	Fall 2022	126	3
CS1555	Fall 2022	123	4
CS1555	Fall 2022	124	NULL
CS1550	Spring 2023	123	NULL
CS1550	Spring 2023	124	NULL
CS1550	Spring 2023	126	NULL
CS1550	Spring 2023	129	NULL
CS2550	Spring 2023	124	NULL
CS1520	Spring 2023	126	NULL

For each of the relational algebra expressions below, identify the expected arity (number of attributes), schema, and min/max cardinality (number of tuples) of the relation resulting from the query without actually evaluating the query and based only on the schemas and cardinalities of the four given relations.

A. $\sigma_{\texttt{Term = 'Spring 2023'}}$`(Course_Taken)`

B. `Course_Taken * Course` ('*' corresponds to the natural join operator on the common attribute, i.e., attribute Course_No)

Question 2. Consider a relation R(A,B,C) containing 5,000,000 records, where each data page (i.e., block) of the relation holds 10 records. R is organized as an ordered file that is sorted on R.A. Assume that R.A is a unique key for R, with values lying in the range 0 to 4,999,999. For each of the following relational algebra queries, state which of the following two approaches is most likely to be more efficient (i.e., reads fewer number of blocks) and justify your answer.

Approaches:

- Access the sorted file R directly.
- Use a B+ tree index on attribute R.A.

Relational algebra queries:

A. $\sigma_{\texttt{A≤50,000}}$`(R)`

B. $\sigma_{\texttt{A≥50,000 and A<50,010}}$`(R)`

Question 3. Consider the relations r1(<u>A</u>,B,C), r2(<u>C</u>,D,E), and r3(<u>E</u>,F), with primary keys A, C, and E, respectively. Assume that r1 has 1000 tuples, r2 has 1500 tuples, and r3 has 750 tuples. Given the relational algebra query r1 * r2 * r3 (where '*' denotes natural join),

A. We have two ways to do the natural joins:

i. r1 with r2 first and then with r3

ii. r2 with r3 first and then with r1

Which one is more efficient in terms of comparisons?

B. Assume that every primary key has a B+ tree index built already. Give the most efficient strategy for computing the join.

Answers for Question 1:

A. Arity = Arity of Course_Taken = 4.

Schema = Schema of Course_Taken = (Course_No, Term, SID, Grade).

Cardinality = Cardinality of Course_Taken * Selectivity of $\sigma_{\texttt{Term = 'Spring 2023'}}$

- Cardinality of Course_Taken = 11;
- Selectivity is in the range of 0 to 1;

Hence, Min Cardinality = 0 and Max Cardinality = 11.

B. Arity = Arity of Course_Taken + Arity of Course - # common attributes = 4 + 3 - 1 = 6.

Schema = (Course_No, Term, SID, Grade, Name, Level).

Attribute Course_No is a foreign key of Course_Taken that refers to Course, which means there is exactly one matching Course tuple for every Course_Taken tuple. Therefore, Cardinality = Cardinality of Course_Taken = 11.

Answers for Question 2:

A. Access the sorted file: Read all the blocks from the beginning of the file until the tuple with A=50,000.

Use a B+ tree index: Search the B+ tree for the address of the block containing the record with A=0; then read all the blocks from that one until the tuple with A=50,000.

The choice of accessing the sorted file is slightly superior to using the B+ tree index simply because of the lookup cost (i.e., search) required on the B+ tree.

B. Access the sorted file: Do a binary search on the file to find the block containing the record with A=50,000.

Use a B+ tree index: Search the B+ tree for the address of the block containing the record with A=50,000; then read that block.

The choice of using the B+ tree index should be superior since searching a B+ tree typically requires reading fewer blocks than performing a binary search on the sorted file.

Answers for Question 3:

A. For (i), in the worst case, we need 1000*1500+1000*750 comparisons. This is because C is the primary key in r2, thus we know that at most one tuple in r2 will match a specific tuple in r1. Therefore, there are at most 1000 tuples in the result of r1 * r2.

   For (ii), in the worst case, we need 1500*750+1500*1000 comparisons.

   Therefore, (i) is more efficient in terms of comparisons.

B. For any tuple from r1, we want to find the matching tuples from r2 and r3. By using the B+ tree index, we can find the matching tuples in the following way.

   1. Take out one tuple from r1 and assume that this tuple has value c for attribute C.

   2. Use the index on C in table r2 to find the tuple in r2 whose value for C is c (suppose this tuple has value e for E).

   3. Use the index on E in table r3 to find the tuple in r3 whose value for E is e.

   The combination of these three tuples will be one tuple in the result of r1 * r2 * r3. Because finding tuples by using the B+ tree index is usually very fast, Step 2 and Step 3 consume little time.