UNIVERSITY OF VICTORIA

EXAMINATION #4 (Physical Database Layer)

PRACTICE VERSION

CSC 370: Database Systems

5 December 2022

13.00 - 13.55 UTC-7

(0 hours, 55 minutes)

This examination consists of ten equally-weighted multiple choice questions. You should record your solutions in the provided bubble sheet. Each question has a single best solution though some responses may earn partial marks; if you record more than one solution for the same question, you will receive a score of zero on that question. If you answer x questions correctly, then your grade on the exam will be x/10, i.e., if you answer at least five questions correctly, you will pass. This exam is closed-book: you are welcome to bring with you empty pages, a calculator, and one single-sided A4/US letter note sheet, but you cannot bring other notes or electronic devices to your desk. Please confirm immediately after the exam starts that you have all five pages and ten questions. You are encouraged to record your responses in this booklet and take it with you when you leave, but you must turn in your note sheet.

Sections: A01, A02, A03
CRN's: 10874, 10875, 14303
Instructor: Dr. Sean Chester

Question 1

You have an empty B+-tree with a node capacity for three keys. What are the contents (keys, not pointers) of the root node after the following sequence of insertions?

`⟨1, 2, 13, 17, 27, 42, 76, 87, 50, 20, 99, 3⟩`

(a) The root is empty

(b) ⟨13, 27, 76⟩

(c) ⟨13, 42, 50⟩

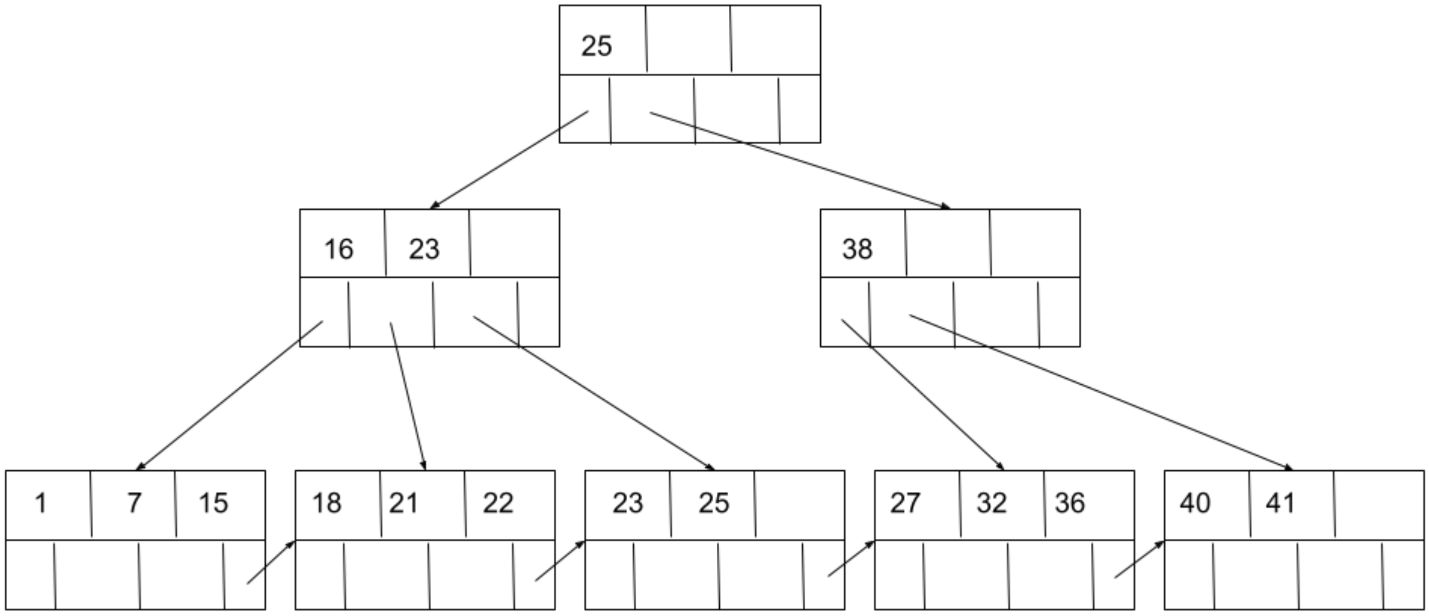
(d) ⟨13, 42, 87⟩

(e) ⟨27, null, null⟩

Question 2

You have the B+-tree shown below. What sequence of keys is visited by a range query, [16, 26]?

B+-Tree for Question 2



(a) ⟨25, 16, 18, 21, 22, 23, 25⟩

(b) ⟨1,7,15,18, 21, 22, 23, 25⟩

(c) ⟨25, 23, 18, 21, 22, 23, 23, 25, 23, 25, 38⟩

(d) ⟨25, 16, 23, 18, 21, 22, 23, 25⟩

(e) ⟨25, 16, 23, 18, 21, 22, 23, 25, 27⟩

Question 3

Assume that no indexes exist in the database. The query below runs too slowly. You have some statistics that can inform a solution. For the purpose of accelerating the query, which attribute would benefit most from a secondary index?

```
T(R) = 1000
T(S) = 10000
T(T) = 100000
V(R,x) = 1000
V(S,x) = 1000
V(S,y) = 10000
V(T,y) = 10000
```

```
SELECT R.x
FROM R
  JOIN S ON (R.x = S.x)
  JOIN T ON (S.y = T.y);
```

(a) R.x

(b) S.x

(c) S.y

(d) T.y

(e) Indexes will not accelerate this query

Question 4

Relation R occupies 10 blocks, relation S occupies 6 blocks, and you have three blocks of memory available. Moreover, both R and S fit four tuples per block. How many I/O's are saved by executing the query below with block-nested loops join rather than a nested loops join?

```
SELECT *
FROM R
  NATURAL JOIN S;
```

(a) 180 blocks

(b) 36 blocks

(c) 210 blocks

(d) 246 blocks

(e) 250 blocks

Question 5

Given the statistics provided below, what is the estimated size of the following relational algebra expression?

```
T(R) = 1000
T(S) = 10000
T(T) = 100000
V(R,x) = 1000
V(S,x) = 1000
V(S,y) = 10000
V(T,y) = 10000
```

$$\pi_{S.x}(R \bowtie_{R.x=S.x} S \bowtie_{S.y=T.y} T)$$

(a) 10

(b) 100

(c) 1000

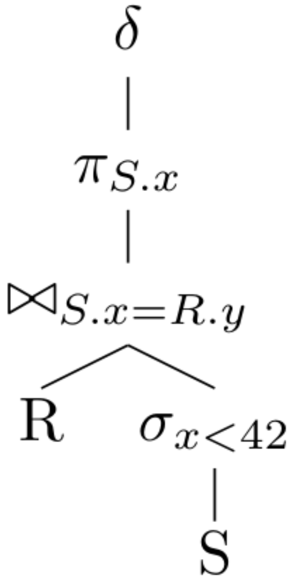
(d) 10000

(e) 100000

Question 6

You are given the logical query plan below. Which of the provided SQL queries does it represent?

Question 6 Logical Query Plan



(a) `SELECT DISTINCT S.x FROM R JOIN S ON (S.x = R.y) WHERE S.x < 42;`

(b) `SELECT DISTINCT S.x FROM R JOIN (SELECT x FROM S WHERE x < 42) AS T ON (R.y = T.x);`

(c) `SELECT DISTINCT S.x FROM (SELECT y AS x FROM R WHERE y < 42) AS T NATURAL JOIN S;`

(d) `SELECT DISTINCT * FROM (SELECT y AS x FROM R) AS R1 NATURAL JOIN (SELECT x FROM S where x < 42) AS S1;`

(e) All of the above

Question 7

Assume the operations below have taken place. Which of the following options could be the contents of an UNDO log if the power went out immediately after the second OUTPUT, i.e., before the second FLUSH LOG?

```
READ(A, t) // A has value 100
t := t + 10
WRITE(A, t)
READ(B, u) // B has value 10
u := u + t
```

```
WRITE(B,u)
FLUSH LOG
OUTPUT(A)
OUTPUT(B)
FLUSH LOG
```

(a) ⟨<START T>, <T, A, 100>, <T, B, 10>⟩

(b) ⟨<START T>, <T, A, 110>, <T, B, 120>⟩

(c) ⟨<START T>, <T, A, 100>, <T, B, 10>, <COMMIT T>⟩

(d) ⟨<START T>, <T, A, 110>, <T, B, 120>, <COMMIT T>⟩

(e) The log would be empty (even absent any corruption to the disk)

Question 8

Assume that you lose connectivity with the database and need to restore from a REDO log, which is shown below. The state of database elements on disk is A=10, B=20, and C=30, but this may not be consistent. What is the state on disk after recovery?

```
<START T1>
<T1, A, 1>
<T1, B, 20>
<T1, C, 3>
<START CKPT(T1)>
<START T2>
<COMMIT T1>
<END CKPT>
<START T3>
<T2, A, 50>
<T3, C, 11>
<COMMIT T3>
```

(a) A=10, B=20, C=30

(b) A=1, B=20, C=3

(c) A=10, B=20, C=11

(d) A=1, B=20, C=11

(e) A=50, B=20, C=30

Question 9

You have a RAID 6 scheme in place with the following layout:

```
Disk 1: 0 1 1
Disk 2: 1 0 1
Disk 3: 1 1 1
Disk 4: 1 1 0
Disk 5: 0 0 1
Disk 6: 0 1 0
Disk 7: 1 0 0
```

Two disks go down concurrently and the database state is as given below. Which of the following are the contents of one of the *lost* disks after recovery?

```
Disk 1: 0110
Disk 2: 1011
Disk 3: ????
Disk 4: ????
Disk 5: 1111
Disk 6: 0001
Disk 7: 1001
```

(a) 0000

(b) 1010

(c) 0110

(d) 1111

(e) The disks cannot be recovered in this case

Question 10

When would you prefer pipelining over materialisation of intermediate query results?

(a) Whenever possible

(b) When you expect the same query might be issued again in the future

(c) When all relations have the same number of tuples

(d) When one of the operands is a view instead of a table

(e) Never

------------- END OF EXAMINATION -------------