UNIVERSITY OF VICTORIA

EXAMINATION #4 (Physical Database Layer)

VERSION B - WITH SOLUTIONS

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 second child of the root node after the following sequence of insertions?

⟨26, 19, 17, 13, 11, 9, 6, 5, 4, 3, 27, 42⟩

(a) The root has only one child

(b) ⟨19, 27, null⟩

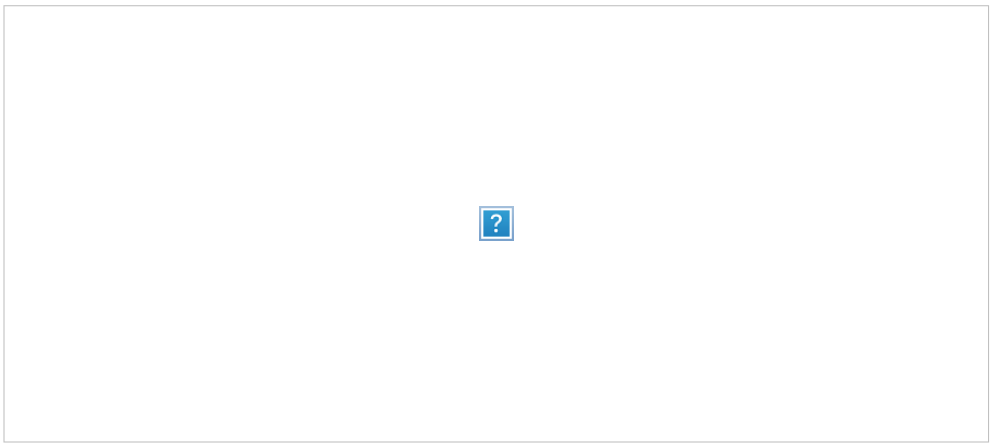
(c) ⟨13, 19, 27⟩ // 0.75 marks, key should be promoted

(d) ⟨13, 19, null⟩ // 0.6 marks, B-Tree, not B+-tree

(e) ⟨13, 27, null⟩

Question 2

You have the B+-tree shown below. What sequence of keys is visited by a lookup query for key 38?



(a) ⟨25, 38⟩

(b) ⟨25, 38, 36⟩

(c) ⟨25, 38, 27, 32, 36⟩

(d) ⟨25, 38, 27, 32, 36, 40⟩

(e) ⟨25, 38, 40⟩

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) = 10000
T(S) = 10000
T(T) = 10000
```

```
V(R,x) = 1000
V(S,x) = 100
V(S,y) = 1000
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 100 blocks, relation S occupies 60 blocks, and you have four blocks of memory available. Moreover, both R and S fit four tuples per block and S has a B+-tree index on the join key with height 3. How many I/O's are saved by executing the query below with an index join rather than block-nested loops join?

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

(a) 0 blocks

(b) 100 blocks

(c) 1760 blocks // 0.8 marks, misses chance to store root in memory

(d) 1800 blocks // 0.7 marks, also misses S in outer loop of BNL

(e) 2159 blocks

Question 5

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

```
                    V(R,x) = 100
T(R) = 100          V(S,x) = 100
T(S) = 1000         V(S,y) = 1000
T(T) = 10000        V(T,y) = 1000
```

$$\sigma_{x=2}(R) \bowtie S \bowtie \pi_y(T)$$

(a) 1000

(b) 1,000,000 // 0.6 marks, uses V(T,y) not T(T)

(c) 10,000,000

(d) 100,000,000

(e) 1,000,000,000

Question 6

Which of the following options is an accurate description of *pipelining*?

(a) Storing the result of each operation on disk until it is needed by another operation

(b) Dedicating as many memory buffers as possible to each operator

(c) Storing the results of one operator in a hash table for efficient lookup

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

(e) Interleaving operators so that intermediate tuples are not stored on disk

Question 7

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

```
READ(A, t) // A has value 50    READ(C, t) // C has value 5
READ(B, u) // B has value 50    u := u + t                    OUTPUT(A)
t := t + u                      WRITE(B,u)                    OUTPUT(B)
WRITE(A, t)                     FLUSH LOG                     FLUSH LOG
```

(a) ⟨<START T>, <T, A, 100>, <T, B, 55>, <CKPT>⟩ // 0.6 marks, needs nonquiescent checkpoint

(b) ⟨<START T>, <T, A, 100>, <T, B, 55>⟩ // 0.6 marks, commit missing

(c) ⟨<START T>, <T, A, 50>, <T, B, 50>⟩

(d) ⟨<START T>, <T, A, 50>, <T, B, 50>, <COMMIT T>⟩

(e) None of the above are consistent with the logging protocol

Question 8

Assume that you lose connectivity with the database and need to restore from an UNDO 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>                  <START T2>                  <T2, A, 50>
<T1, B, 20>                 <COMMIT T1>                 <T3, C, 11>
<T1, C, 3>                  <END CKPT>                  <COMMIT T3>
<START CKPT(T1)>            <START T3>                  <T2, B, 13>
```

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

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

(c) A=50, B=13, C=11

(d) A=50, B=13, C=3

(e) A=50, B=13, C=30

Question 9

You have a RAID 6 scheme in place with the layout and contents given below. Which of the following options are the contents of one of the *lost* disks after recovery?

```
Disk 1: 0 1 1                    Disk 1: ????
Disk 2: 1 0 1                    Disk 2: 1001
Disk 3: 1 1 1                    Disk 3: ????
Disk 4: 1 1 0                    Disk 4: 1111
Disk 5: 1 0 0                    Disk 5: 0110
Disk 6: 0 1 0                    Disk 6: 0001
Disk 7: 0 0 1                    Disk 7: 1001
```

(a) 0111

(b) 1011

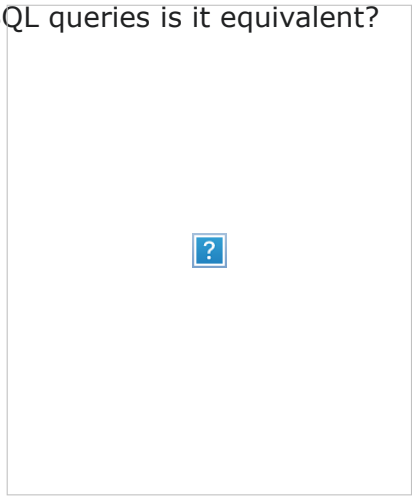
(c) 1101

(d) 1110

(e) The disks cannot be recovered in this case

Question 10

You are given the logical query plan below. To which of the provided SQL queries is it equivalent?



(a) `SELECT S.x, COUNT(*) AS total FROM R JOIN S ON (S.x = R.y) WHERE S.x < 42 AND R.y = 87;`strong>

(b) `SELECT S.x, COUNT(*) AS total FROM R JOIN (SELECT x FROM S WHERE x < 42) AS T ON (R.y = T.x) WHERE y = 87 GROUP BY x;`

(c) `SELECT S.x, COUNT(*) AS total FROM S WHERE x = 87 AND x < 42 GROUP BY x;` // 0.6 marks, assumes one-to-one relationship

(d) `SELECT x, COUNT(*) AS total FROM (SELECT y AS x FROM R WHERE x = 87) AS R1 NATURAL JOIN S GROUP BY x;`

(e) None of these queries matches the logical query plan given

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