Xi’an Jiaotong-Liverpool University

PAPER CODE	EXAMINER	DEPARTMENT	TEL
CPT201			

_1st_ SEMESTER _2023/2024 RESIT_ EXAMINATION

DATABASE DEVELOPMENT AND DESIGN

TIME ALLOWED: _2 Hours_

INSTRUCTIONS TO CANDIDATES

1、 **This is a _closed-book_ examination, _which is to be written without books or notes_.**

2、 **Total marks available are 100.**

3、 **This exam consists of _two_ sections:**

Section A consists of 20 short answer questions worth 2 marks each for a total of 40 marks.

Section B consists of 2 problem-solving and quantitative questions worth 30 marks each for a total of 60 marks.

4、 **Answer all questions. There is NO penalty for providing a wrong answer.**

5、 **Only English solutions are accepted. Answer should be written in the answer booklet(s) provided.**

6、 **All materials must be returned to the exam invigilator upon completion of the exam. Failure to do so will be deemed academic misconduct and will be dealt with accordingly.**

Section A: Short Answer Questions

[40 marks]

1) Relation *books* has 30,000 tuples, which are stored as fixed length and fixed format records; each has the length of 360 bytes. Tuples contain the non-key attribute *title* with length of 20 bytes. The tuples are stored sequentially in a number of blocks, ordered by *title*. Each block has the size of 4,096 bytes and each tuple is fully contained in one block. What is number of disk blocks needed to store the relation student? Justify your answer.

$\lfloor \frac{4096}{360} \rfloor = 11$ $\lceil \frac{30000}{11} \rceil = 2728$ blocks

[2/40]

2) With the same information in Part A, Question 1), suppose that a primary sparse index (i.e. one index entry for one block) on the *title* attribute is to be created. An 8-byte pointer to the actual tuple is needed for each index entry. Each index entry is also fully contained in one block. Assume that all 30,000 tuples have different titles, what is the number of blocks needed to store the primary index?

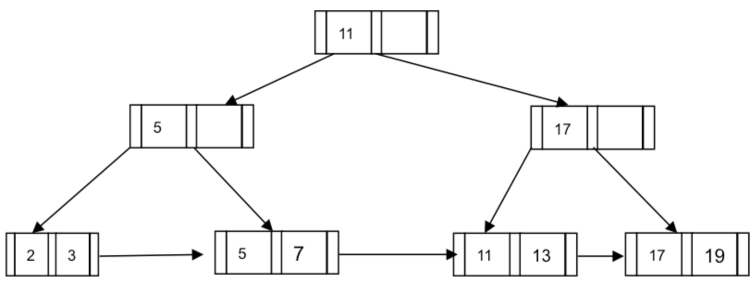
$20+8=28$ bytes $\lfloor \frac{4096}{28} \rfloor = 146$ $\lceil \frac{2728}{146} \rceil = 19$

[2/40]

3) With the following B+tree, briefly describe the process of deleting search key "11".

[2/40]

Initial B+ tree



4) Name two spatial indexing techniques which can support range queries.

R-Tree   QuadTree

[2/40]

5) In evaluating selection operations using a secondary index on the non-candidate key with equality, what would be the cost in terms of block transfers and seeks, respectively?

[2/40]

transfer: $b_r \cdot (h_i+n)$ seek: $b_r \cdot (h_i+n)$

6) Briefly describe how the comparative selection $\delta_{A \geq V}(r)$ can be efficiently evaluated using a secondary index, where *A* stands for the attribute name of the relation *r*, and *V* for a constant.

Find the first leaf node satisfy $\delta_{A \geq V}(r)$, and scan from this leaf to get the pointers. Then use these pointers to find the tuples.

[2/40]

7) Assume that two relations are not sorted and are too large to fit in memory. Briefly describe how to join the two relations using the merge-join algorithm. Detail for the merge-join is not needed.

[2/40]

8) Following the information given in Part A Question 7), which of the following two algorithms is likely to be more efficient in performing the join: (1) Hash Join; (2) Merge-Join? Why?

[2/40]

9) Let r and s denote two relations, and R and S the attributes of r and s, respectively. If $R \cap S$ is a candidate key for r, what would be the number of tuples in '$r \bowtie s$'? Justify your answer.

[2/40]

10) What is meant by isolation of transactions in relational database systems?

[2/40]

11) What is meant by 'a schedule S is conflict serialisable'?

[2/40]

12) Briefly describe how to determine if a schedule is cascadeless.

[2/40]

13) Briefly describe how deadlocks can be detected by using the wait-for graph.

[2/40]

14) In relational databases, the lock manager usually maintains a lock table to record granted locks and pending requests. What data structure can be usually used to implement the lock table?

[2/40]

15) What needs to be done by 'undoing' a transaction in log-based failure recovery algorithms with checkpoints?

[2/40]

16) In distributed database systems, what are the responsibilities of a transaction coordinator?

[2/40]

17) With coordinator failure in distributed database systems, briefly describe what is meant by the blocking problem.

[2/40]

18) Briefly describe the concept of 'Eventual Consistency' in big data storage systems.

[2/40]

19) In blockchain based storage systems, briefly describe how is 'irrefutability' ensured?

[2/40]

20) An online short-video website plans to implement a recommendation functionality for its users. Assume that the website has information about users' past viewing history. Briefly describe a data mining technique that would be helpful in implementing the recommendation.

[2/40]

Section B: Problem-Solving and Quantitative Questions

[60 marks]

Question 1. Consider the following two relations in a university database for teaching assistant management.

- *teachingAssistant* (*ID*, *name, department, supervisor, email*)
- *supervises* (*ID, labCode*)

ID in relation *teachingAssistant* is the candidate key, and *supervises.ID* is the foreign key referencing *teachingAssistant*. It is required that a teaching assistant cannot supervise more than 10 labs. The catalog information is given as follows.

- number of tuples in relation *teachingAssistant*, $n_r = 230$, and number of blocks, $b_r = 20$;
- number of tuples in relation *supervises*, $n_s = 1{,}100$, and number of blocks, $b_s = 35$;
- index: a dense primary B+-tree index of height 3 on attribute *supervises.ID;*
- number of distinct values on attribute *supervisor*, V(*teachingAssistant*, *supervisor*) = 30.

Assume that the memory can only hold one block for each relation. Answer the following questions.

[30 marks]

a) Using the nested loop join algorithm and the smaller relation as the outer relation, how many block transfers and seeks would be needed to evaluate "*teachingAssistant* ⋈ *supervises*", respectively?

transfer: 20+230×35=8070

Seek: 20+230=250

[4/30]

b) Using the blocked nested loop join algorithm and the smaller relation as the outer relation, how many block transfers and seeks would be needed to evaluate "*teachingAssistant* ⋈ *supervises*", respectively?

transfer: 20+20×35: 720

Seek: 20×2=40

[4/30]

c) Assume that the smaller relation fits in memory. Using the nested loop join algorithm, how many block transfers and seeks would be needed to evaluate "*teachingAssistant* ⋈ *supervises*", respectively?

~~transfer: 20+230×35=8070~~ transfer: br+bs

~~Seek: 1+230=231~~ Seek: 2

[4/30]

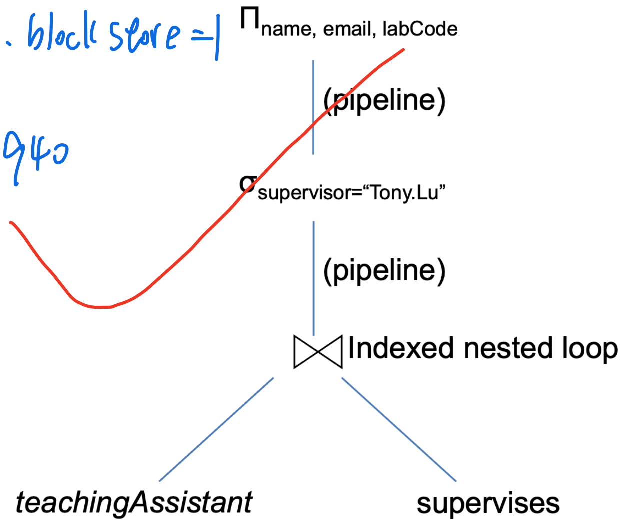
d) Assume that the external merge-sort algorithm is needed to sort the relation *supervises*, with memory size M=5 and buffer size b_b=2. How many block transfers and seeks would be needed, respectively?

[4/30]

e) Consider the evaluation plan as shown in the following diagram. Assume that all selection operations are evaluated using linear scan. Note that no intermediate results need to be stored using pipelining. Only an estimate is needed with the evaluation plan shown in the diagram below. What would be the total number of block transfers for the whole evaluation plan? Justify your answer.

$\lceil \frac{1100}{35} \rceil = 32$ tuple > 10 ∴ block store $= 1$

transfer: $20 + 230 \times (3+1) = 940$



[4/30]

f) Based on the relational algebra equivalence rules for selection operations, how can the evaluation plan shown in the diagram in Part B Question 1.e) be further optimised using "pushing selection early"? What would be total number of block transfers after optimisation? Justify your answer?

[10/30]

$\lceil \frac{230}{30} \rceil = 8$ tuples ∴ $20 + 1 + 1 + 8 \times (3+1) = 54$

$\lceil \frac{20}{30} \rceil = 1$ blocks

Question 2. Consider the following schedule and answer the following questions.

Schedule: T1:write(X); T1:write(Y); T2:read(X); T2:write(Y); T2:abort; T1:write(Z); T1:commit; T3:read(Y); T3:write(Z); T3:commit.

[30 marks]

a) Draw a precedence diagram for the schedule. Is the schedule above conflict serialisable? Justify your answer.

$T_1 \rightarrow T_2$, $T_1 \rightarrow T_3$

Yes. No cycle.

[6/30]

b) Is the schedule above view serialisable? If yes, to what serial schedule is it equivalent? Justify your answer.

Yes. Because a schedule is conflict serialisable, it is also a view serialisable.

T_1, T_2, T_3

[6/30]

c) Is the schedule above recoverable? Justify your answer.

No. The commit of T_1 is after T_2.

[4/30]

d) Is the schedule above cascadeless? Justify your answer.

No. Because commit of T_1 isn't before T_2:read(X)

[4/30]

e) Consider the following transaction logs and answer the following questions: (1) Which transaction(s) is in the checkpoint L1? (2) In the redo pass, which transactions need to be redone? (3) After the redo pass, which transaction(s) is left in the undo-list? (4) In the undo pass, which transactions need to be undone? (5) After successful recovery, what logs need to be added to the stable storage?

(1). T101 T104 T105

(2). T105 T107 T104

(3). T107

(4). T107

(5). <T107, A, 7006>

<T107 abort>

Start of the logs
<T101 start>
<T101, A, 5896, 7006>
<T104 start>
<T104, B, 1065, 1732>
<T105 start>
<checkpoint {L}>
<T105, C, 35, 190>
<T105 commit>
<T101 commit>
<T107 start>
<T107, A, 7006, 773>
<T104, B, 1732>
<T104 abort>
←System crash, start recovery

[10/30]

END OF EXAM PAPER