Xi'an Jiaotong-Liverpool University

PAPER CODE	EXAMINER	DEPARTMENT	TEL
CPT201			

_1st_ SEMESTER _2023/2024 FINAL_ 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) A relational database contains two relations, namely *teacher* and *student*.

   In the relation *teacher*, tuples are stored as fixed-format, fixed-length records, each of 300 bytes; each tuple contains the key attribute *tid* of length 20 bytes; other fields and record headers make up the rest; there are 500 tuples in this relation.

   In the relation *student*, tuples are also stored as fixed-format, fixed-length records, each of 200 bytes; tuples contain foreign key attribute *tid*, referencing *teacher.tid*; there are 5,000 tuples in this relation.

   Assume that each teacher must supervise exactly 10 students. Tuples in the *teacher* relation are sequentially ordered by *tid* and the "clustered disk organisation" strategy is used. This means that, for each teacher record, the 10 student records also reside in the same block. A record does not span over more than one block, and the size of each block is 4,096 bytes.

   What is the number of disk blocks needed to store the two relations?

   $200 \times 10 + 300 = 2300$ bytes $\quad \left\lceil \frac{500}{1} \right\rceil = 500$

   $\left\lfloor \frac{4096}{2300} \right\rfloor = 1$

   [2/40]

2) Following the same information in Part A, Question 1), suppose that a dense primary index is to be created on *tid* for the *teacher* relation, i.e. one index entry for each tuple. Each index entry includes a key and a 15-byte pointer to data. How many disk blocks are needed to store the index?

   $15 + 20 = 35$ bytes $\quad \left\lfloor \frac{4096}{35} \right\rfloor = 117 \quad \left\lceil \frac{500}{117} \right\rceil = 5$

   [2/40]

3) Following the same information in Part A, Question 1) and 2), given a student name, discuss which of the following three indices is likely to be efficient in looking up the student tuple:
   - A dense primary index on *student.tid*
   - A sparse secondary index on *student.tid*
   - A dense primary index on *teacher.tid*

   [2/40]

4) Briefly explain under what circumstances the two strategies, "merge siblings" and "redistribute pointers" should be used during deletion operations in a B+ tree.

   [2/40]

5) Name two popular techniques used for spatial indexing.

   [2/40]

6) In query optimisation in relational database systems, what should be defined in an evaluation plan?

   [2/40]

7) In the context of query optimisation in relational database systems, briefly describe the key differences between *materialisation* and *pipelining*.

[2/40]

8) In cost-based query optimisation, besides statistical information about relations (e.g. number of tuples and number of distinct values for an attribute), name two other important factors in estimating the cost of an evaluation plan.

[2/40]

9) Name two rules used in heuristic optimisation that can transform query evaluation trees to improve execution performance.

[2/40]

10) In the context of transaction management in relational database systems, briefly describe what is meant by *isolation*.

[2/40]

11) In relational database systems, how can isolation be achieved?

[2/40]

12) In relational database systems, is a conflict serialisable schedule always recoverable? Justify your answer.

[2/40]

13) In the context of relational database systems, briefly describe the two-phase locking protocol for concurrency control.

[2/40]

14) In relational databases implementing log-based recovery algorithm, what needs to be done during normal transaction processing if there is a transaction rollback?

[2/40]

15) In concurrency control in relational databases, name two deadlock prevention strategies based on transaction timestamps.

[2/40]

16) What protocol is needed for failure recovery and concurrency control in distributed database systems?

[2/40]

17) There are four general principles in linked data design and publication, besides (1) Use URIs as names for things, (2) Use HTTP URIs so that people can look up those names, what are the 3rd and 4th principles?

[2/40]

18) Briefly describe why ‘Eventual Consistency’ is preferred in big data storage or No-SQL systems.

[2/40]

19) What are the four properties of blockchains?

[2/40]

20) Briefly describe the difference between association rule analysis and clustering.

[2/40]

Xi'an Jiaotong-Liverpool University

Section B: Problem-Solving and Quantitative Questions

[60 marks]

Question 1. The database of an online movie website has the following three relations. The catalog information is also provided.

i) *performer (pid, name, sex, agent, country, email)*
ii) *performsIn (pid, mid)*
iii) *movie (mid, title, year, language, producer)*

The attributes "*pid*" and "*mid*" are the keys for relations *performer* and *movie*, respectively. The two attributes in *performsIn* are the foreign keys referencing *performer* and *movie*, respectively. An agent for a performer is someone who signs a contract with and represents the performer.

- In relation *performer*, number of records $n_{performer}$ = 3,800; number of blocks $b_{performer}$ = 520;
- In relation *performsIn,* number of records $n_{performsIn}$ = 70,050; number of blocks $b_{performsIn}$ = 86;
- In relation *movie*, number of records, n_{movie} = 31,000; number of blocks b_{movie} = 2,700;
- Index: a primary B^+-tree index of height 7 on the *pid* attribute of the *performsIn* relation;
- Number of distinct values for the attribute *agent* in the *performer* relation *V(performer, agent) = 500;*

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 to evaluate "*movie* ⋈ *performsIn*", which relation should be used as the outer relation? How many block transfers and seeks would be needed, respectively? Justify your answer.

movie. transfer: 2700+31000×86= 2668700 seek: 2700+31000=33700

[6/30]

b) To evaluate "*performer* ⋈ *performsIn*", discuss which of the two algorithms, block nested loop join and indexed nested loop join, would be more efficient. How many block transfers and seeks would be needed, respectively? Justify your answer.

[6/30]

block nested loop join:
transfer: 86+86×520=44806
seek: 86×2=172

indexed nested loop join:
transfer: 520+3800×8=30920
seek: 520+3800×8=30920

block nested loop join

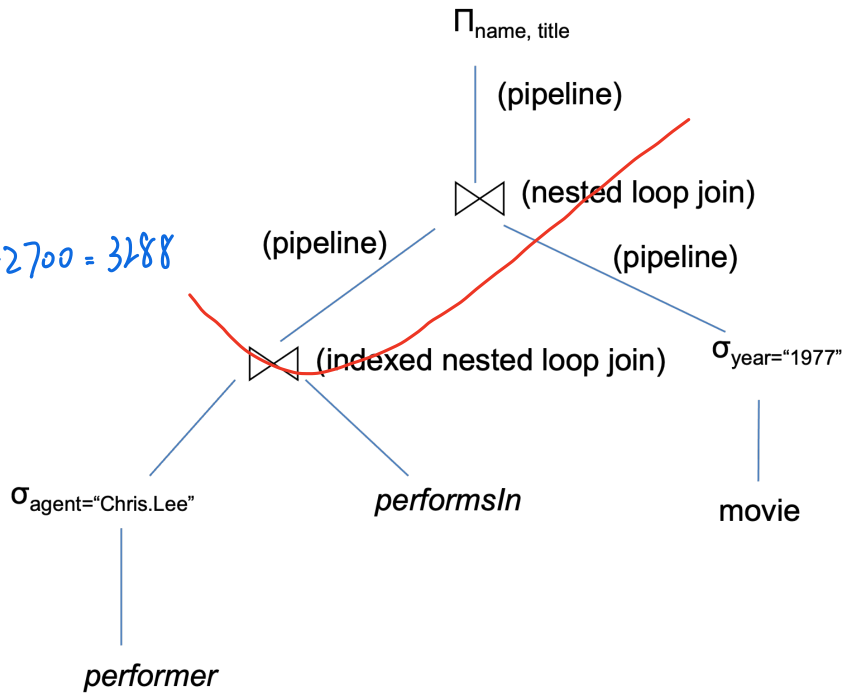
c) Assume that linear scan is used to evaluate all the selection operations, and no intermediate relations need to be stored as the result of using pipelining. Only an estimate is needed with the evaluation plan shown in the diagram below. What would be the total number of block transfers? Justify your answer.

$\lceil \frac{520}{500} \rceil = 2$

$tuple = \lceil \frac{3800}{500} \rceil = 8$

$8 \times 8 = 64$

$520 + 2 + 2 + 64 + 2700 = 3288$



[12/30]

d) Based on the evaluation plan shown in Part B, Question 1.c), describe what further optimisation could be done based on the equivalence rules in relational algebra. Justify your answer.

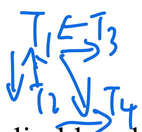
[6/30]

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

T1	T2	T3	T4
	start;		
	read(X);		
	read(Y);		
		start;	
start;			
read(X);			
read(Y);			
		read(X);	
write(X);			
	write(Y);		
	read(Z);		
	commit;		
write(Y);			
commit;			
			start;
			read(Y);
		write(X);	
			write(Y);
			abort;
		commit;	

[30 marks]

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



No

Because there is a cycle.

[8/30]

b) Is a view serialisable schedule recoverable? Justify your answer.

[3/30]

T1 T2
W(X)
R(X)
Commit W(X)
commit

c) Provide an example schedule and explain why cascadeless schedules are needed.

[4/30]

d) Consider the following partial schedule with log-based recovery. Assume that the initial values for x, y and z are all 1 (i.e. x=y=z=1). Answer the following questions: (1) What are the transactions in the checkpoint L1{}? (2) What are the transactions in the checkpoint L2{}? (3) Assume that a crash happens immediately after time=23. When recovering from the system crash, in the redo pass, which transactions need to be redone? (4) After the redo pass, what transactions are left in the undo list? (5) Which transactions need to be undone during the recovery? (6) What logs need to be inserted to stable storage after the successful recovery? *Hint: you should produce the log records with the given schedule on a draft paper first. Do NOT write the draft log records on the answer booklet.*

(1). T1 T2

(2). T1 T2 T3

(3). T2 T3

(4). T1 T2 T3

(5). T1 T2 T3

(6). <T3 y 2>

<T2 x 4>

<T3 z 1>

<T2 y 1>

<T3 abort>

<T1 z 1>

<T2 abort>

<T1 x 1>

Time	T1	T2	T3
1	Read(x)		
2	x=x+3		
3	Read(z)		
4	Write(x) 4		
5		Start	
6	------------------CHECKPOINT L1{...} ------------------		
7		Read(y)	
8	x=z-1		
9	Write(z) 1		
10		Read(x)	
11			Start
12			Read(z)
13		y=y*2	
14		Write(y) 2	
15			z=z+7
17			Read(y)
18			Write(z) 8
19	------------------CHECKPOINT L2{...} ------------------		
20		x=x-3	
21		Write(x) 1	
22			y=y/7
23			Write(y) $\frac{2}{7}$

[15/30]

END OF EXAM PAPER