Xi'an Jiaotong-Liverpool University

PAPER CODE	EXAMINER	DEPARTMENT	TEL
CPT201		Computing	

FIRST SEMESTER 2022/2023 FINAL EXAMINATIONS

BACHELOR DEGREE – Year 3

DATABASE DEVELOPMENT AND DESIGN

Exam Duration: ***2 Hours***

Crash Time Allowed: ***30 Minutes (for online exam only)***

INSTRUCTIONS TO CANDIDATES

1、 **This is a close-book exam. Please tick the integrity disclaimer *when uploading your answers on LM Core* and complete the assessment independently and honestly.**

2、 **Total marks available are 100. This will count for 70% in the final assessment.**

3、 **Answer ALL questions in both Part A and Part B.**

4、 **The number in the column on the right indicates the marks for each section.**

5、 **The university approved calculator - Casio FS82ES/83ES can be used.**

6、 **All the answers must be in English.**

7、 **The duration is *2 hours*, and an additional *30-minute* crash time beyond the exam duration will be allowed for you to report and resolve minor technical issues which may be encountered during the online exam. Where there are any major problems preventing you from continuing the exam or submitting your answers in time, please do not hesitate to email the Assessment Team of Registry (assessment@xjtlu.edu.cn).**

THIS PAPER MUST NOT BE REMOVED FROM THE EXAMINATION ROOM

PART A: Short Answer Questions

[40 marks]

1) Relation *bankCustomer* has 50,000 tuples, which are stored as fixed length and fixed format records; each has the length of 350 bytes. Tuples contain the non-key attribute *name* with length of 15 bytes. The tuples are stored sequentially in a number of blocks, ordered by *name*. 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 *bankCustomer*?

$\lfloor \frac{4096}{350} \rfloor = 11$ $\lceil \frac{50000}{11} \rceil = 4546$

[2/40]

2) With the same information in Part A, Question 1), suppose that a primary index using B+tree on the *name* attribute is to be created. A 10-byte pointer to actual tuples (an 8 byte block id and 2 byte offset) is needed for each index entry. Each index entry is also fully contained in one block. If the primary index is sparse, i.e. one index entry for one block, what would be the maximum number of blocks needed to store the index?

10+15 = 25 bytes $\lfloor \frac{4096}{25} \rfloor = 163$ $\lceil \frac{4546}{163} \rceil = 28$

[2/40]

3) With the same information in Part A, Question 2), what would be the minimum number of blocks needed to store the index? (Hint: in this case, all tuples have the same name)

only 1 block

[2/40]

4) Removal of a search key from a B+ tree may cause a non-leaf node to become underfull. What are the two strategies to restore the balance of the tree?

[2/40]

5) Briefly describe a popular application of the R-Tree in database systems.

[2/40]

6) What is the main purpose of query optimisation in relational database systems?

[2/40]

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

[2/40]

8) What is meant by 'pipelining' in query optimisation in relational database systems?

[2/40]

9) What are the two simple rules used in heuristic optimisation that transform query evaluation trees to improve execution performance?

[2/40]

10) What are the four properties of transactions in relational database systems?

Atomicity, consistency, isolation, durability.

[2/40]

11) Briefly describe how to test conflict serialisability of a schedule.

[2/40]

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

[2/40]

13) Briefly describe the two-phase locking protocol for concurrency control in relational database systems.

[2/40]

14) In log-based recovery algorithm, what is meant by 'undoing' and 'redoing' a log record of the form '*<Ti, X, V1, V2>*', where T stands for a transaction, X for the data item, and V for values?

[2/40]

15) Describe what a recovery system would do when performing checkpointing.

[2/40]

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

[2/40]

17) Briefly describe how RDF (Resource Description Framework) graph datasets and linked data can be queried.

[2/40]

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

[2/40]

19) What are the two categories of consensus algorithms used in public blockchain based storage systems?

Proof of work

Proof of stake

[2/40]

20) Suppose that an online company has stored millions of movie reviews in its database. After purchasing a reasonable amount of labeled review data, it wants to analyse if reviews in its database are positive or negative, e.g. for sentimental analysis. Propose two algorithms for this task.

[2/40]

PART B: Problem-Solving and Quantitative Questions

[60 marks]

Question 1. A small online mobile video creation and sharing platform called 'VideoBook' has the following three relations in its database. The relevant catalog information is given: "*userId*" is the candidate key for relation *user*; "*videoID*" is the candidate key for relation *video*; the two attributes in relation *creates* are the foreign keys referencing *user* and *video*, respectively. In all relations, no records span over one block.

- *user (<u>userID</u>, name, email, city)*
- *creates (userID, videoID)*
- *video (<u>videoID</u>, title, category, creationDate, format, URL)*

Number of records in *user*, n_{user} = 7,000; each disk block can hold up to 300 tuples; blocks = 24

Number of records in ~~*user*~~ creates, $n_{\cancel{user}}$ creates = 90,000; each disk block can hold up to 650 tuples; blocks = 139

Number of records in *video*, n_{video} = 50,500; each disk block can hold up to 70 tuples; blocks = 722

Index: a primary B+-tree index of height 6 on the *userID* attribute of the *creates* relation;
The number of distinct values, *V(city, user)* = 10.
Answer the following questions.

[30 marks]

a) Assume that in the worst case, the memory can only hold one block for any of the relations. 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 "*user* ⋈ *creates*", respectively?

transfer: 24 + 7000 × 139 = 973024   Seek: 24 + 7000 = 7024

[4/30]

b) Assume that in the worst case, the memory can only hold one block for any of the relations. Using the blocked nested loop join algorithm and the small relation as the outer relation, how many block transfers and seeks would be needed to evaluate "*user* ⋈ *creates*", respectively?

transfer: 24 + 24 · 139 = 3360   Seek: 24 × 2 = 48

[4/30]

c) Assume that the indexed nested loop join algorithm is used based on the available B+ tree index on *creates*. How many block transfers and seeks at minimum would be needed to evaluate "*user* ⋈ *creates*", respectively?

transfer: 24 + 7000 × (6+1) = 49024   seek: 24 + 7000 × (6+1) = 49024

[4/30]

d) Assume that both relations *user* and *creates* are already sorted on disk. With the merge join algorithm and buffer size b_b=1, how many block transfers and seeks would be needed to evaluate "*user* ⋈ *creates*", respectively?

transfer: 139 + 24 = 163

[4/30]

Seek: $\lceil \frac{139}{1} \rceil + \lceil \frac{24}{1} \rceil = 163$

e) Based on the results obtained from a), b), c) and d), discuss which join algorithm is the most efficient in evaluating "*user* $\bowtie$ *creates*".

[4/30]

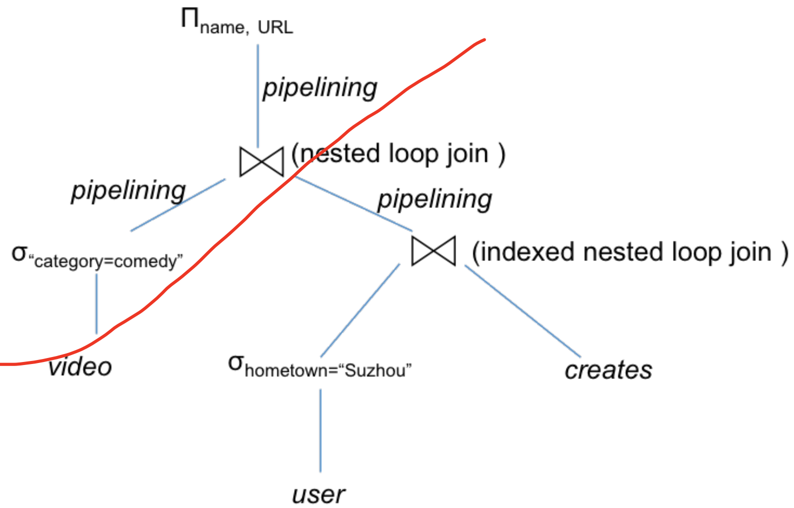
Seek 和 transfer 大小关系

f) Assume that the technician at 'VideoBook' created an optimised evaluation plan for a query, which is shown in the following diagram. Assume that linear scan is used to evaluate all the selection operations. The use of pipelining is also shown in the diagram. What is the total number of block transfers for the whole evaluation plan? Justify your answer.

$722+24+3+3+700\times 7$

$= 4900+752$

$= 5652$



[10/30]

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

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

[30 marks]

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

$T_1 \to T_2$ Yes

T_3 T_4

[5/30]

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

Yes. T_1 T_2 T_3, T_4

[5/30]

c) Is the schedule recoverable? Justify your answer.

Yes.

[5/30]

d) Is the schedule cascadeless? Justify your answer.

No.

[5/30]

e) Consider the following partial transaction logs produced by a schedule during some time period. Answer the following questions: (1) What are the transactions in the checkpoint L1{}? (2) What are the transactions in the checkpoint L2{}? (3) When recovering from the system crash, in the redo pass, which operations 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 is the meaning of the logs '<T4, B, 700>, <T4 abort>'? (7) What logs need to be inserted to stable storage after the successful recovery?

(1). T_1, T_2

(2). T_2, T_3, T_4, T_5

(3). <T5 C 200 100>
<T6 A 100 300>
<T4 B 700>
<T3, D, 90, 19>

(4). T_3 T_6

(5). T_3 T_6

(6). Roll back and make B=700 in T_4

(7). <T3 D 90>
<T6 A 100>
<T6 abort>
<T3 abort>

Start of the logs
<T1 start>
<T2 start>
<T1, B, 300, 700>
<checkpoint L1{…}>
<T1 commit>
<T4 start>
<T5 start>
<T4, B, 700, 1900>
<T2, D, 30, 90>
<T3 start>
<checkpoint L2{…}>
<T5, C, 200, 100>
<T5 commit>
<T2 commit>
<T6 start>
<T6, A, 100, 300>
<T4, B, 700>
<T4 abort>
<T3, D, 90, 19>
← System crash, start recovery

[10/30]

END OF EXAM PAPER