Xi'an Jiaotong-Liverpool University

PAPER CODE	EXAMINER	DEPARTMENT	TEL
CPT201		Computing	

FIRST SEMESTER 2020/2021 FINAL EXAMINATIONS

BACHELOR DEGREE – Year 3

DATABASE DEVELOPMENT AND DESIGN

Exam Duration: *2 Hours*

INSTRUCTIONS TO CANDIDATES

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

2、 **Answer ALL FOUR questions.**

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

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

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

THIS PAPER MUST NOT BE REMOVED FROM THE EXAMINATION ROOM

Question 1. Answer the following questions on indexing in database systems.

[25 marks]

A relation called *employee* has 10,000 tuples. The tuples are stored as fixed length and fixed format records; each has the length of 200 bytes. Tuples contain the key attribute *name* with length of 20 bytes. The tuples are stored sequentially in a number of blocks, ordered by *name*. Each block has the size of 4,096 bytes (i.e., 4 Kilobytes) and each tuple is fully contained in one block. Suppose that a primary index using B+ tree on the *name* attribute is to be created. The length of each index entry is 30 bytes: 20 bytes for the search key and 10 bytes for the pointer to data (an 8 byte block id and a 2 byte offset). Each index entry is also fully contained in one block.

a) Compute the number of tuples of the relation *employee* that one block can hold.

[2/25]

b) Compute the number of disk blocks needed to store the relation *employee*.

[3/25]

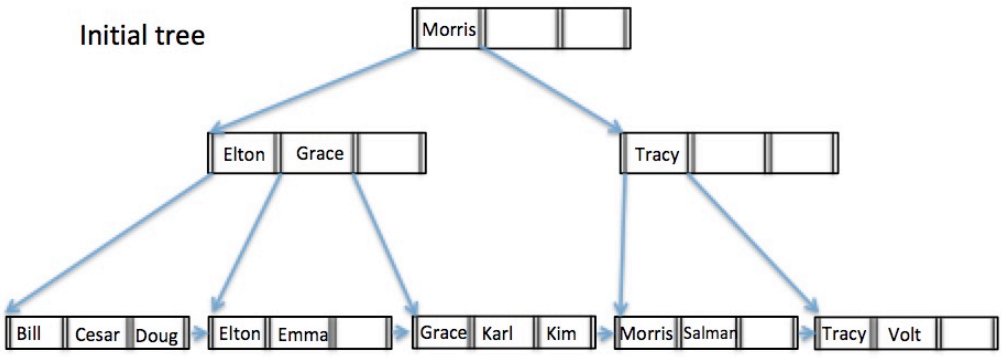
c) If the primary index on *name* is sparse, i.e., one index entry for one block, compute the number of blocks needed to store the primary index.

[4/25]

d) If the primary index on *name* is dense, i.e., one index entry for one tuple, compute the number of blocks needed to store the primary index.

[4/25]

e) Suppose at some point, the B+ tree index on *name* attribute is shown as follows. The number of pointers in a node is 4. Draw the B+ tree for each of following operations (in total there should be four trees): (1) insert 'Amy'; (2) insert 'Linda'; (3) delete 'Linda'; (4) delete 'Amy'. (Each of the operations, besides (1), must be performed on the B+ tree drawn for the previous operation and (1) must be performed on the following B+ tree).

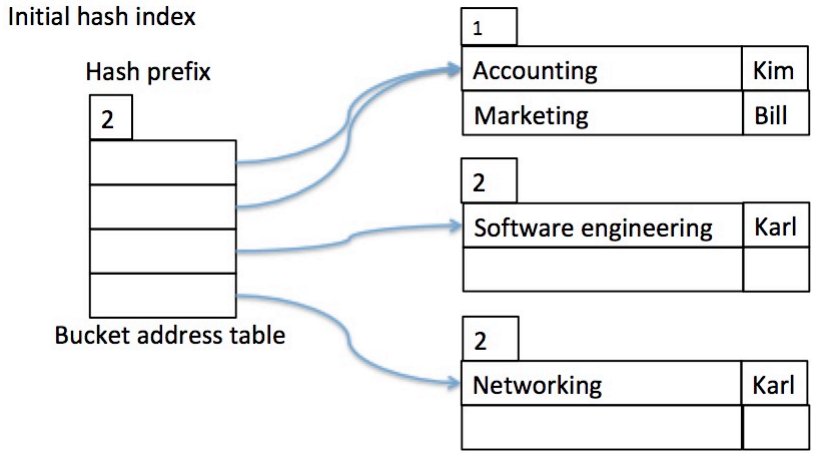


[8/25]

f) Suppose that an extendable hash index is created to keep track of the training courses that the employees have taken. Based on the hash values on the titles of the training courses in

the table below and the initial hash index, draw the hash index after insertion of the following tuples (tuple is of the form "*title_trainingCourses*, *employee.name*"): (1) 'Data mining, Amy'; (2) 'Data mining, Linda'.

Search key	Hash value
Accounting	0001 1001 0000 1111
Marketing	0101 1001 0101 1100
Operating system	0011 1011 0000 1100
Data mining	1100 1001 0101 1010
Networking	1110 0001 0101 1110
Software engineering	1010 0101 1111 0010



[4/25]

Question 2. A student report management system contains the following three relations.

i) student (SID, name, department, email)
ii) produces (SID, RID)
iii) report (RID, title, year, topic)

"*SId*" and "*RID*" are the keys for relations *student* and *report*, respectively. The two attributes in relation *produces* are the foreign keys referencing *student* and *report*, respectively. The catalog information about the three relations is given below.

- number of tuples in *student,* $n_{student}$ = 7,000, stored on 300 blocks;
- number of tuples in *produces,* $n_{produces}$ = 50,000, stored on 500 blocks;
- number of records in *report,* n_{report} = 35,000, stored on 1,000 blocks;
- index: a primary B^+-tree index of height 3 on the *SID* attribute of the *produces* relation*;*
- number of distinct values for the attribute *department* in the *student* relation, *V(student, department) = 50;*

Answer the following questions.

[25 marks]

a) For the query '*find all reports which were written after 2017 and before 2019*', describe how it can be evaluated in the most cost effective way by using a primary B+ tree index on *year* and a secondary B+ tree index on *year*, respectively.

[4/25]

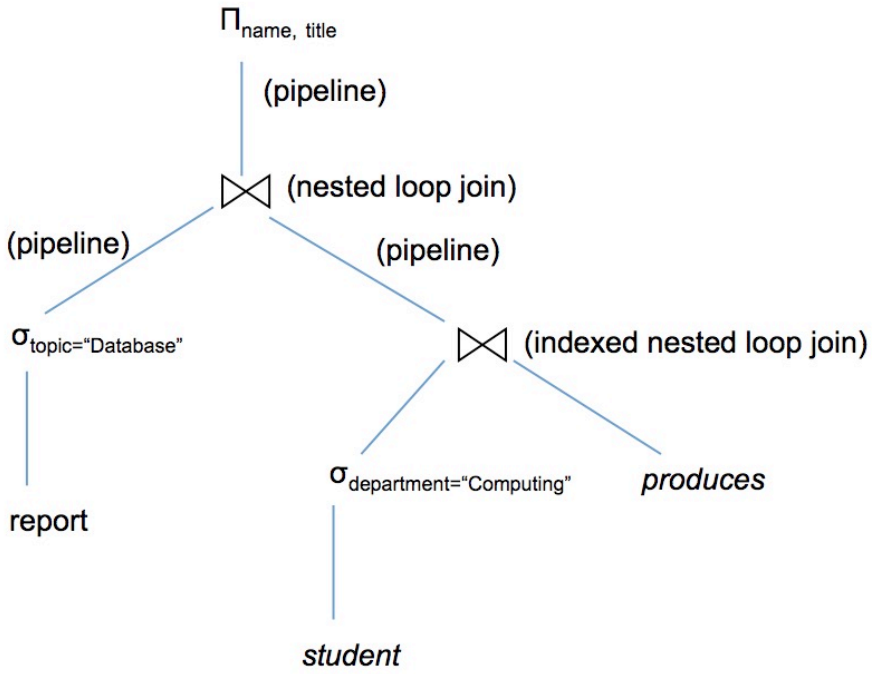
b) Assume that the memory can only hold one block for each relation. To evaluate the join operation "*student* $\bowtie$ *produces*", assume that the small relation is always used as the outer relation. Which of the two algorithms: (1) nested loop join, and (2) block nested loop join is more efficient? Justify your answer.

[8/25]

c) Assume that both relations are not sorted, memory size *M=10* blocks, buffer size $b_b=1$ block. To evaluate "*student* $\bowtie$ *produces*" using the merge join algorithm, how many block transfers and seeks would be needed, respectively?

[8/25]

d) An optimised query evaluation plan is shown below. Note that no intermediate relations need to be stored as the result of using pipelining. The B+-tree index of the *produces* relation is used to perform the indexed nested loop join. Assume that linear scan is used to evaluate all the selection operations. What is the total number of block transfers for the evaluation plan? Justify your answer.


Π name, title
(pipeline)
(nested loop join)
(pipeline)
(pipeline)
σ topic="Database"
(indexed nested loop join)
σ department="Computing"
produces
report
student

[5/25]

Question 3. Answer the following questions.

[25 marks]

a) Explain what problem will arise from the following schedule which uses the two-phase locking protocol.

T1	T2
lock-X(B);	
read(B);	
B = B − 50;	
write(B);	
	lock-S(A);
	read(A);
	lock-S(B);
lock-X(A);	
read(A);	
A = A + 50;	
write(A);	
unlock(B);	
unlock(A).	
	read(B);
	display(A + B);
	unlock(A);
	unlock(B).

[3/25]

b) Consider the following schedule and assume initially X=5 and Y=5. Can it be transformed to a serial schedule? What is the value of calling display(X+Y)?

T1	T2
read(X);	
	read(X)
	read(Y)
X = X − 1;	
write(X);	
	X = X + 1
	write(X);
read(Y);	
Y = Y + 1;	
write(Y);	
	read(X)
	display(X + Y);

[3/25]

c) Draw the precedence diagram for the following schedule. Determine if it is conflict serialisable and justify your answer.

	T1	T2	T3
1	write(A)		
2	read(B)		
3	read(C)		
4			write(B)
5	write(B)		
6			write(C)
7		read(C)	
8		write(B)	
9		write(C)	
10			read(A)

[4/25]

d) Consider the same schedule in Question 3.c), is it view serialisable? If yes, what serial schedule is it equivalent to? Justify your answer.

[5/25]

e) Consider the same schedule in Question 3.c), is it recoverable? Justify your answer.

[4/25]

f) Consider the following transaction logs. (1) What are the transactions in the checkpoint L{}? (2) When recovering from the system crash, in the redo pass, which operations need to be redone? (3) After the redo pass, what transactions are left in the undo list? (4) In the undo pass, which operations need to be undone? (5) What logs need to be added after the recovery is successful?

Start of the logs
<T17 start>
<T17, B, 300, 700>
<T12 start>
<T17 commit>
<T14 start>
<T15 start>
<T14, B, 700, 1700>
<T12, D, 70, 20>
<T13 start>
<T15, C, 200, 100>
<T15 commit>
<T12 commit>
<T16 start>
<checkpoint L{...}>
<T16, A, 100, 300>

<T14, B, 700>
<T14 abort>
<T13, D, 20, 19>
←***System crash, start recovery***

[6/25]

Question 4. Answer the following questions.

[25 Marks]

a) Santa Claus stores the names of all the children in the world and the toys they want, in distributed databases. Consider the following three relations at two sites:
Site 1:
Children (id, kid_name, age, address, city)
Request (id, toy_id)
Site 2:
Toys (toy_id, toy_name, toy_color, min_age)

Assume that the query "*find the names of children from the city of Beijing and their toy names*" is issued to the Site 1. It is also known that only a 'small' number of toys was wanted by the children from the Beijing city. Describe in detail how semi-join can be used to optimise the query.

[8/25]

b) Consider the following table for market basket analysis. Columns represent the items and rows represent transactions. Each cell represents whether an item is purchased in a transaction, "1" means yes and "0" means no. Compute the following:
(1) confidence (Bread ->> {Beer, Diapers})
(2) support (Milk ->> Cola)
(3) confidence ({Beer->> Milk})
(4) support (Eggs, Bread})
(5) confidence(Cola->>Milk)

Transaction	Bread	Milk	Diapers	Beer	Eggs	Cola
t1	1	1	0	0	0	0
t2	1	0	1	1	1	0
t3	0	1	1	0	0	1
t4	1	1	1	0	0	0
t5	1	1	0	1	0	1

[5/25]

c) **DHL** - an international shipping company uses a traditional relational database to store its business data, e.g. *item number* (unique), *weight*, *dimensions*, *destination*, *final delivery date*, and *transportation events* (i.e., flights, truck deliveries). As the amount of data keeps growing, it becomes difficult for the company to manage the complex relations among the data items, e.g. multiple items might have the same destination (country/city) but different addresses, they might be shipped from different areas, and the shipped items might make their way to their destinations via one or more standard DHL transportation events.

Suppose that you have been hired as a consultant to re-design the database system. The company proposed the solutions by using:

- linked data (or knowledge graph)
- object-oriented database

You are asked to make a final choice (only one of the above two), and then describe your design based on the following two aspects:

1) Data model - explain how the data is modeled and represented with your choice. Maximum 100 words.
2) Data storage and query – describe relevant techniques in supporting data storage and user queries. You may discuss this with an example. Maximum 100 words.

[12/25]

END OF EXAM PAPER