Database Development and Design (CPT201)

Tutorial 5: Prepare for Final Exam

Dr. Wei Wang

Department of Computing

Q1

- Consider the following two relations and their catalogue information:
  - *account(<u>account_Number</u>, customer_Name, balance, branch_ID)*
  - *branch(<u>branch_ID</u>, branch_Name, branch_City, postcode)*
- The "*account_Number*" is the key for the *account* relation, and the "*branch_Name*" is the key for the *branch* relation. The *account* relation contains 300,000 records stored in 60,000 blocks, and the *branch* relation contains 500 records stored in 50 blocks. Assume that both relations are sequentially stored by the key attributes.
- Answer the following questions:
  - A) Suppose that the linear search algorithm is used to evaluate the selection $\delta_{balance>5,000}$ in the *account* relation, how many block transfers are needed? How many seeks are needed?

Q1 – cont'd

- B) Suppose that none of the relations can fit in memory, and the nested loop join algorithm is used to evaluate "*account branch*". Which relation should be used as outer relation? How many block transfers are needed? How many seeks are needed?
- C) Suppose that the external sort merge algorithm is used to sort the *account* relation on the *account_Number* attribute. Assume that the memory size $M=30$ and the buffer for reading and writing $b_b=2$. How many block transfers are needed? How many seeks are needed?
- D) Suppose that the hash join algorithm is used to evaluate "*account branch*", the number of partitions, $n_h=70$, and the size of the buffer for reading and writing, $b_b=2$. How many block transfers are needed? How many seeks are needed?

Q1 Solutions

- A) 60,000 block transfers and 1 seek.
- B) The smaller relation branch should be used as the outer relation (denoted as *r*). Number of block transfers: $n_r*b_s+b_r$ = 500*60,000+50 = 30,000,050; Number of seeks: b_r+n_r = 50+500 = 550
- C) Number of block transfers: $b_r(2\lceil \log_{\lfloor M/bb \rfloor -1}(b_r/M)\rceil + 1)$ = $60{,}000(2*\log_{14}(60{,}000/30))+1)$ = 420,000; Number of seeks: $2\lceil b_r/M\rceil + \lceil b_r/b_b\rceil(2\lceil \log_{\lfloor M/bb \rfloor -1}(b_r/M)\rceil -1)$ = 2*(60,000/30) + $60{,}000/2(2*\log_{14}(60{,}000/30))-1)$ = 4,000+30,000*5 = 154,000
- D) Number of block transfers: $3(br+bs)+4*nh$ = $3(50+60{,}000)+4*70$ = 180,430; Number of seeks: $2(\lceil br/bb\rceil+\lceil bs/bb\rceil)+2nh$ = 50+60,000+2*70 = 60,190

Xi'an Jiaotong-Liverpool University 西交利物浦大学

Q2

- Consider the following three relations and their catalog information.
  - *staff(<u>ID</u>, name, email, department)*
  - *teaches(ID, module_Code)*
  - *module(<u>module_Code</u>, module_Title, level)*
- where *staff.ID* is the key for *staff*, and *module_Code* is the key for *module; teaches.ID* and *teaches.module_Code* are the foreign keys referencing *staff* and *module*, respectively. A staff member can teach at most 10 modules.
  - the number of records in *staff*, n_{staff} = 1,000;
  - the number of blocks in *staff*, b_{staff} =200
  - the number of distinct values for the attribute *department* in the *staff* relation, *V(department, staff) = 50*
  - index: a three-level primary B^+-tree index (height=3) on the *ID* attribute of *teaches* relation.
  - the number of records in *teaches*, $n_{teaches}$ = 3,000
  - the number of blocks in *teaches*, $b_{teaches}$ = 100
  - the number of records in *module*, n_{module} = 1,500
  - the number of blocks in *module*, b_{module} = 150

Q2 – cont'd

- Consider the following relational algebra expression and answer the questions below.

  $\Pi_{\text{name, module\_Title}}(\sigma_{\text{department="Industrial Design"} \wedge \text{level="4"}}(\text{staff} \bowtie \text{teaches} \bowtie \text{module})$

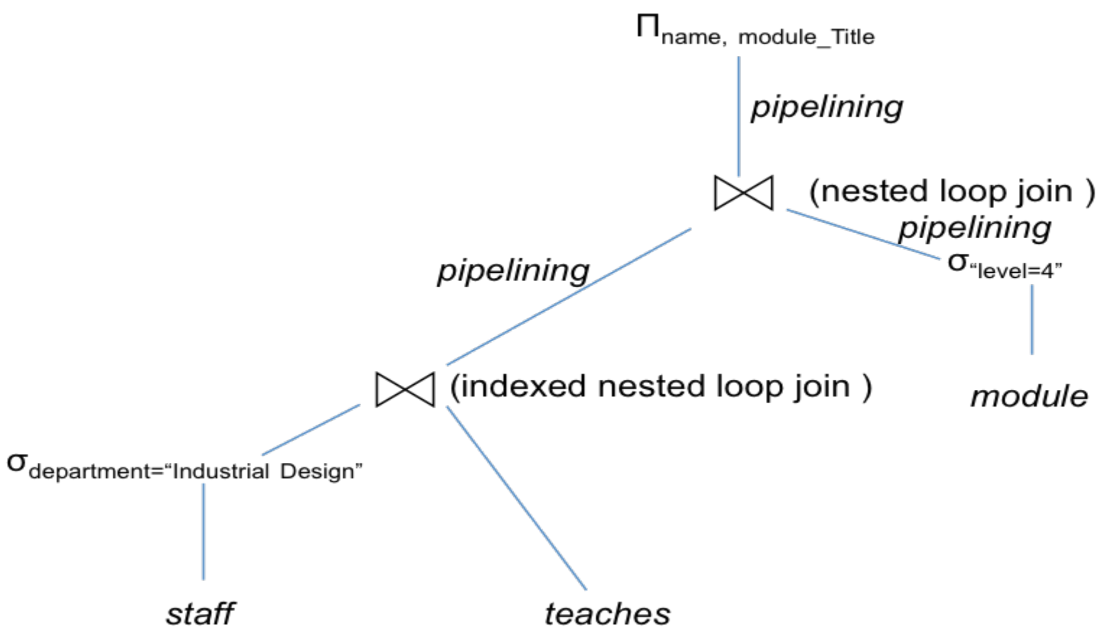
  - A) One of the heuristic rules for query optimisation is to perform selection operations as early as possible. Write the equivalent algebra expression for the given expression based on this heuristic rule and the equivalence rules.
  - B) Suppose that the join between relations *staff* and *teaches* is evaluated using the indexed nested loop join algorithm; the join between the result and the relation *module* is evaluated using the nested loop join algorithm. Also, assume that pipelining can be used for projection and nested loop join. Draw an annotated evaluation tree for the relational algebra expression obtained from Question 3.a).

Q2 – cont'd

- C) Based on the given catalog information and query evaluation tree, what is the estimated size of the selection $\sigma_{department=\text{"Industrial Design"}}(staff)$? How many blocks would be needed to store the results?
- D) Based on the given catalog information and query evaluation tree, what is the estimated size of the join "*staff* $\bowtie$ *teaches*" using the indexed nested loop join algorithm?
- E) Based on the results from Question 3.D), what is the maximum size of the nested loop join with the relation *module*? Justify your answer.
- F) Assume all sections are evaluated using linear scan, what would be the total number of block transfers for the entire evaluation plan?

Q2 Solutions

- A) $\Pi_{name,\ module\_Title}((\sigma_{department=\text{"Industrial Design"}}(staff) \bowtie teaches) \bowtie (\sigma_{level=\text{"4"}}(module))$
- B)



Q2 Solutions – cont'd

- C) As V(department, staff) = 50, there are 1,000 records in the staff relation. So the estimated size is 20. One block can hold 5 (1,000/200) tuples, the result may occupy 20/5=4 blocks.
- D) As the join attribute *ID* is the foreign key for *teaches* relation, so the size of "*staff* $\bowtie$ *teaches*""is 3,000. V(department, staff) = 50, So the estimated size is 60.
- E) The estimated size of the indexed nested loop join is 60; the size of the module is 1,500. The join attribute *module_Code* is the foreign key referencing the *module* relation. One tuple in the result of "*staff* $\bowtie$ *teaches*" can join with one tuple in the relation *module*. So the size of the nested loop join is 60.

Q2 Solutions – cont'd

- F)
  - Selection $\sigma_{department=\text{"Industrial Design"}}(staff)$ needs 200 block transfers.
  - The result of the selection (20 tuples) needs to be materialised onto the disk, the number of block transfers is 4 (from answer 2.C).
  - For the indexed nested loop join, the number of block transfers needed is $b_r+n*c = 4+20(3+1) = 84$.
  - Selection $\sigma_{level=\text{"4"}}(module)$ needs 150 block transfers.
  - As pipelining is used elsewhere, so the total number of block transfer is 200+4+84+150 = 438.

Xi'an Jiaotong-Liverpool University 西交利物浦大學

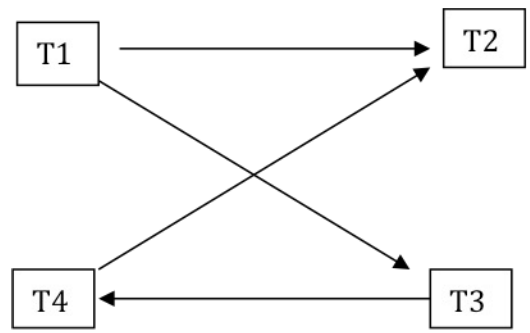
Q3

- A) Draw the precedence diagram for the schedule below. Is it conflict serialisable? Justify your answer.
- B) Is the schedule recoverable? Justify your answer.

T1	T2	T3	T4
Read(x)			
Write(x)			
Read(y)			
	Read(x)		
		Write(y)	
	Read(z)		
			Read(y)
			Read(w)
			Write(w)
	Read(w)		

Q3 Solutions

- A) Yes, it is conflict serialisable. Justification: because the precedence diagram is acyclic.



- B) A recoverable schedule is one where, for each pair of transactions *Ti* and *Tj* such that *Tj* reads a data item previously written by *Ti*, the commit operation of *Ti* appears before the commit operation of *Tj*. It is not possible to determine if the schedule is recoverable or not as the commit operations of the transactions are not given.

Q4

- A) Consider the following schedule.
  - *T1:write(X); T1:write(Y); T2:read(X); T2:write(Y); T2:read(Z); T1:write(Z); T1:commit; T3:read(Y); T2: commit; T3:write(Z); T4:read(Z); T3:commit; T4:abort.*
  - Draw the precedence diagram for the schedule. Is it conflict serialisable? Justify your answer.
- B) Is the schedule in Question 4.C recoverable? Justify your answer.
- C) Is the schedule in Question 4.C cascadeless? Justify your answer.

Q4 Solutions

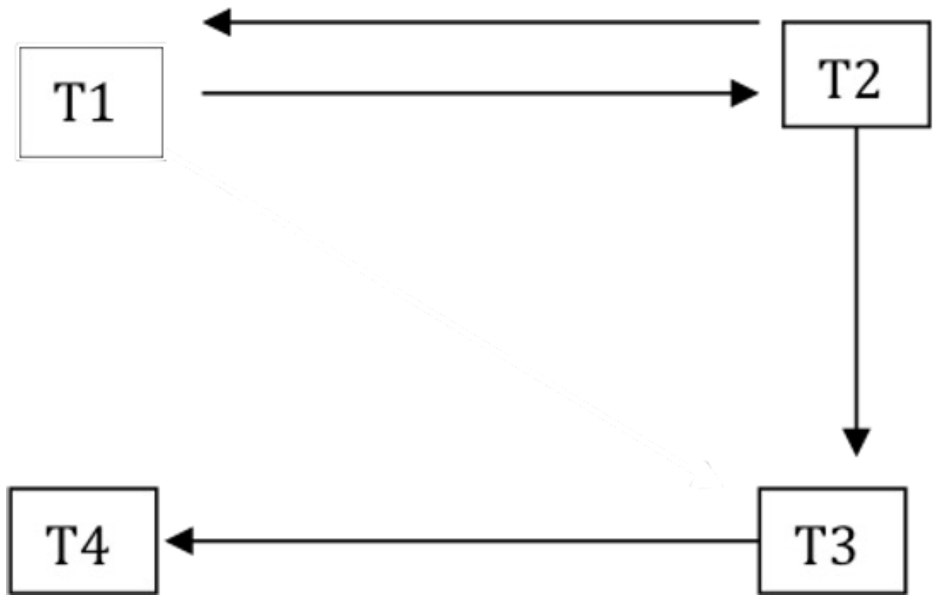
First, you need to list all the transactions and their instructions in a table.

- W: write
- R: read
- *Do this on draft paper*

T1	T2	T3	T4
W(X)			
W(Y)			
	R(X)		
	W(Y)		
	R(Z)		
W(Z)			
COMMIT			
		R(Y)	
	COMMIT		
		W(Z)	
			R(Z)
		COMMIT	
			ABORT

Q4 Solutions – cont'd

- A)



No, it is not conflict serialisable. Justification: there is a cycle in the graph (between T1 and T2).

Q4 Solutions – cont'd

- B) The above schedule is recoverable.
  - T2 reads X which is written by T1 and T1 commits before T2 does.
  - T3 reads Y which is written by T2 and T2 commits before T3 does.
  - T4 reads Z which is written by T3 and T3 commits before T4 aborts.
- C) The above schedule is not cascadeless.
  - T2 reads X which is written by T1 but T1 commits after T2 reads X.
  - T3 reads Y which is written by T2 but T2 commits after T3 reads Y.
  - T4 aborts later so it is not considered.

Q5

- Consider the schedule below and answer the following questions. Transactions that are not shown in the table should be ignored. Initially, X=10 and Y=20.
  - (1) Write the logs that would be generated by the execution of the schedule.
  - (2) Which transactions are in the list *L* at checkpoint 1 and checkpoint 2, respectively?
  - (3) Assume that the failure happens immediately after t=52. When recovering from the system crash, which transactions need to be redone?
  - (4) What needs to be undone?
  - (5) What transaction logs need to be added after the successful recovery?

Q5 – cont'd

Time	T1	T2	T3	T4
30	start			
31	read(X)			
32	X=2*X			
33			start	
34			read(Y)	
35			Y=Y+7	
36	***Checkpoint 1***			
37			write(Y)	
38			commit	
39				start
40				read(Z)
41		start		
42		read(X)		
43		read(Y)		
44		Y=Y*X		
45		display(Y)		
46		write(Y)		
47		commit		
48	read(Y)			
49	X=X-Y			
50	display(X)			
51	write(X)			
52	***Checkpoint 2***			
53				read(Y)

Q5 Solutions

- (1)
  - <T1 start>
  - <T3 start>
  - <checkpoint 1 {T1, T3}>
  - <T3, Y, 20, 27>
  - <T3 commit>
  - <T4 start>
  - <T2 start>
  - <T2, Y, 27, 270>
  - <T2 commit>
  - <T1, X, 10, -250>
  - <checkpoint 2 {T1, T4}>
- (2)
  - checkpoint 1: {T1, T3}
  - checkpoint 2: {T1, T4}

Xi'an Jiaotong-Liverpool University 西交利物浦大學

Q5 Solutions – cont'd

- (3)
  - No transactions needs to be redone.
- (4)
  - T1 and T4 need to be undone.
- (5)
  - <T1, X, 10>
  - <T4 abort>
  - <T1 abort>
  - *Order of the logs does matter. Answers are considered as wrong if the order is wrong.*