SWE3003 Database Systems - Final Exam Spring 2020

[Note 1] Write down your student ID and name on top of each answer sheet.
[Note 2] You are not allowed to leave WebEx meeting until the exam is over at 14:45.
[Note 3] For the problems that require justification, you should explain how you reached the answer. Without correct justification, you will get no point.
[Note 4] No questions are allowed. If you find a question is not clear and needs some assumptions, you should make and write down your own assumptions and solve it based on the assumptions.
[Note 5] For the scheduling problems, you should assume each transaction starts with the first operation shown in the tables.

1. [Storage and File Structures, 20pts] You have a database table that consists of 20,000 records. Each 4 KB disk page can hold 20 records, thus the database table is stored in 1,000 disk pages. Suppose the time to read or write each disk page is 2 msec, and your DBMS does not use the buffer cache.

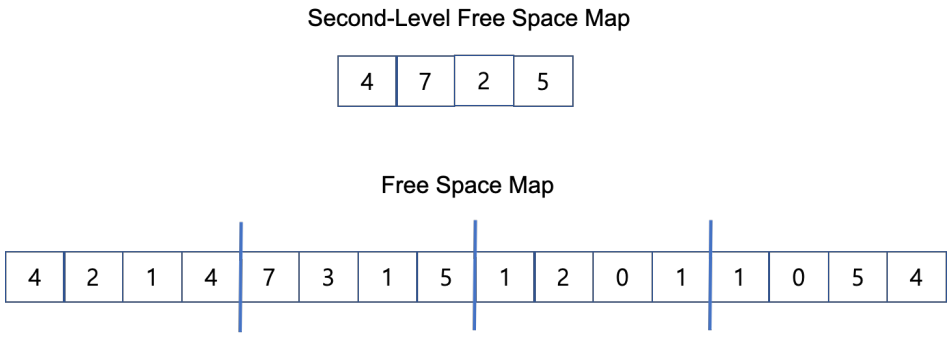
   (a) If your database table is organized as a "Heap File", what is the average cost (in terms of msec) of inserting a new record into the table? Justify your answer.

   (b) If your database table is organized as a "Heap File", what is the average cost (in terms of msec) of finding a record in the table? Justify your answer.

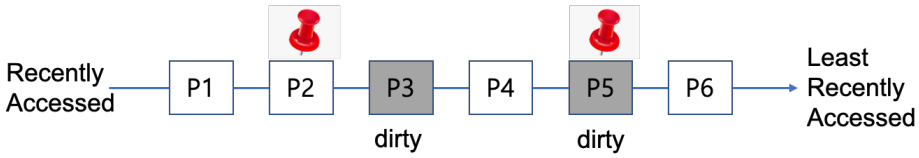
   (c) If your database table is organized as a "Sequential File", what is the average cost (in terms of msec) of inserting a new record into the table? Justify your answer.

   (d) If your database table is organized as a "Sequential File", what is the average cost (in terms of msec) of finding a record in the table? Justify your answer.

2. [Storage and File Structures, 10pts] The following figure shows a free space map for a heap file. A query that inserts a new record of 3500 bytes is submitted. If the disk page size is 4 KB, how the free space map will be changed? Justify your answer.



3. [Buffer Cache, 15pts] Suppose a buffer manager can hold up to six pages in a DBMS, as shown in the figure below, and the buffer manager is using the LRU replacement policy. Note that the dark colored pages are dirty, and the white colored pages are clean. The red pins indicate which pages are pinned by a suspended transaction, which will not wake up until the next transaction T1 commits. Note that all the pages accessed by uncommitted transactions must reside in the buffer manager until they commit.



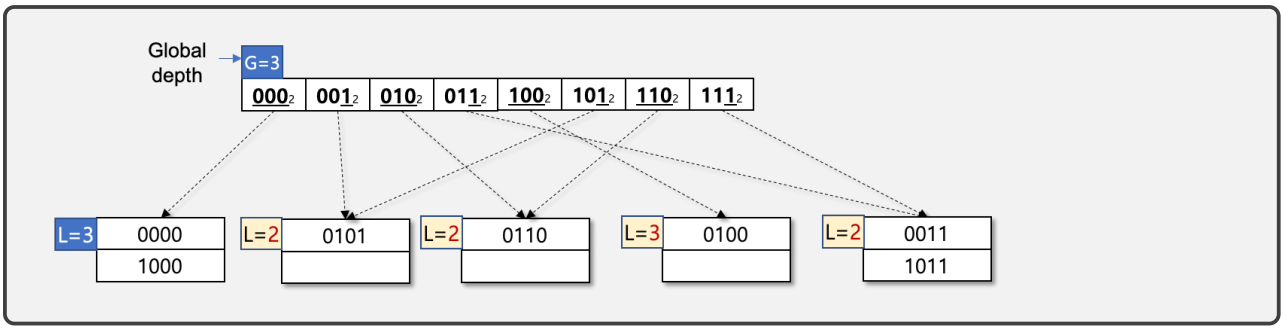
(a) If a new transaction T1 reads page P7 and P8, and then commits when the buffer manager's LRU list is as shown in the figure, how many disk pages will be flushed to disks? Draw the status of LRU list after the transaction T1 commits.

(b) If a new transaction T1 reads page P4, P7 and P8, and then commits when the buffer manager's LRU list is as shown in the figure, how many disk pages will be flushed to disks? Draw the status of LRU list after the transaction T1 commits.

(c) If a new transaction T1 **writes** page P4, reads P7 and P8, and then commits when the buffer manager's LRU list is as shown in the figure, how many disk pages will be flushed to disks? Draw the status of LRU list after the transaction T1 commits.

4. [Extendable Hashing, 10pts] Suppose we insert the following records into an empty extendable hash table. The initial global depth of directory is 1 and there are 2 empty buckets whose local depths are also 1. We use the least significant bits of keys and $\%2^G$ as the hash function. If each bucket can hold up to 2 records, how will the extendable hash table change after all the following records are inserted? Please draw an extendable hash table that holds the following records. Don't forget writing global depth and local depths.

Record	Hash key
r0	0000
r3	0011
r4	0100
r5	0101
r6	0110
r8	1000
r11	1011



5. [Linear Hashing, 10pts] Suppose we insert the following records into an empty linear hash table. The initial number of buckets (N) is 2 when the level (L) is 0. I.e., the hash function family is $\%2^L \times N$. If each bucket can hold up to 2 records, how will the linear hash table change after all the following records are inserted? Please draw a linear hash table that holds the following records.

Record	Hash key
r0	0000
r3	0011
r4	0100
r5	0101
r6	0110
r8	1000
r11	1011

6. [Indexing, 10pts] Suppose you have a table that has a very large number of attributes and a millions of records. If your application runs the following type of queries frequently, which indexing would perform the best? Justify your answer.

```
SELECT attr1
FROM myTable
WHERE attr2 = C2 AND attr3 = C3 AND ... AND attrN = CN;
```

7. [Cache Replacement, 5pts] Which relational operator works best for MRU strategy? Justify your answer.
   (a) SELECT
   (b) JOIN
   (c) UNION
   (d) SET DIFFERENCE

8. [Join, 10pts] Suppose you are joining a `student` table and `department` table. `student` table has 1,000 records and `department` table has 100 records. Each disk page can hold maximum 10 student records or 10 department records. Both `student` and `department` tables do not have any index and they are organized as heap files. What is the expected number of disk accesses if we employ Sort-Merge Join using 4 input buffer pages and 1 output buffer page? Show how you reached the answer.

9. [Query Optimization, 15pts] Suppose you are have `student`, `department`, `takes`, and `course` tables. `student` table has thousands of records and it is stored in a B+tree file. Its primary key is the student ID. `department` table has dozens of records and it is stored in a heap file without a secondary index. `takes` table has millions of records in a heap file without a secondary index. It has student ID and course ID. `course` table has thousands of records stored in a hash table, where course ID is used as the hash key. Consider the following query.

```
SELECT student.name
FROM student JOIN department JOIN takes JOIN course
WHERE course.course_id = 'swe3003' AND takes.year = 2020
      AND department.dept_name = 'software';
```

   Draw an expression tree for a query execution plan that you think the most efficient. Explain why your query plan will be efficient.

10. [Serializability, 10pts] Consider the schedule shown in Figure 1.

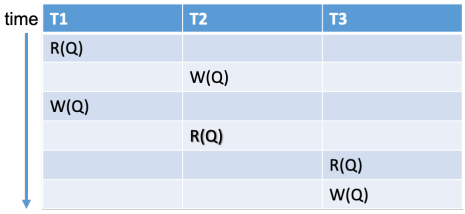


Figure 1: Schedule 1

(a) Is this schedule conflict serializable? Justify your answer.

(b) Is this schedule view serializable? Justify your answer.

11. [2PL, 15pts] Consider the schedule shown in Figure 2.

T1	T2
Lock-S(A)	
R(A)	
Lock-X(B)	
Unlock-S(A)	
	Lock-S(A)
W(B)	
Unlock-X(B)	
	Lock-X(B)
	W(B)
	R(A)
	Ulock-S(A)
	Ulock-X(B)
	Commit
Abort	

Figure 2: Schedule 2

(a) Can this schedule be generated by two-phase locking (2PL) protocol? Justify your answer.

(b) Can this schedule be generated by strict two-phase locking (strict-2PL) protocol? Justify your answer.

(c) Is this a cascadeless schedule? Justify your answer.

12. [2PL, 15pts] Consider the schedule shown in Figure 3.

T1	T2
Lock-S(A)	
R(A)	
Lock-X(B)	
W(B)	
Unlock-X(B)	
	Lock-X(B)
Unlock-S(A)	
	R(B)
	W(B)
	Lock-S(A)
	Unlock-X(B)
	R(A)
	Unlock-S(A)
	Commit
Abort	

Figure 3: Schedule 3

(a) Can this schedule be generated by two-phase locking (2PL) protocol? Justify your answer.

(b) Can this schedule be generated by strict two-phase locking (strict-2PL) protocol? Justify your answer.

(c) Is this a cascadeless schedule? Justify your answer.

13. [Multiple granularity locking, 20pts] Consider a database system holding the records shown below that implements hierarchical multiple granularity locking protocol. Suppose transaction T0 acquired locks on DB, A0 and F1 as shown in Figure 4. For each of the following transactions, indicate what kind of locks on which tree node it can acquire or not. Justify your answer.

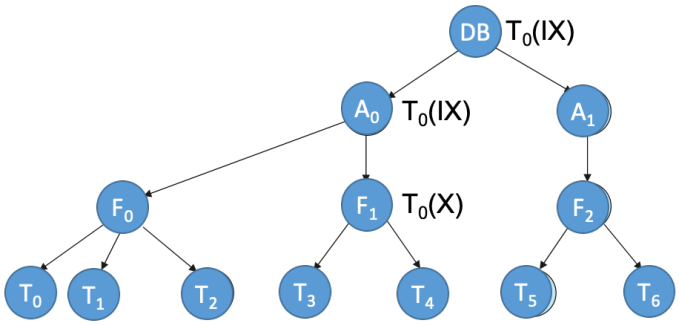


Figure 4: Granularity Hierarchy

(a) Read(T5)

(b) Write(T2)

(c) Add a new tree node A2 under DB.

(d) Read(DB) and Write(A1)

14. [Timestamp-ordering, 10pts] Consider the following schedule under timestamp-ordering protocol. Initially, the write and read timestamps for data `A` and `B` are all 0, and the timestamps of `T1` and `T2` are 1 and 2, respectively.

T1	T2
R(A)	
	R(A)
	W(A)
	R(B)
R(B)	
Display(A+B)	
	W(B)
	Commit
Commit	

Figure 5: Timestamp Ordering Protocol

(a) Is the schedule valid under timestamp-ordering protocol? Justify your answer.

(b) Transform this schedule into a serial schedule.

15. [Validation-based, 10pts] Consider the following schedule under validation-based protocol.

T1	T2
R(A)	
	R(A)
A=A+1	
	R(B)
	B=A+1
R(B)	
validate	
W(A)	
	validate
	W(B)

Figure 6: Validation-based Protocol

(a) Is this a valid schedule under the validation-based protocol? Justify your answer.

(b) Transform this schedule into a serial schedule.

16. [MVCC, 10pts] Is the following schedule valid under multi-version timestamp ordering protocol? Justify your answer.

T1	T2	T3
R(A)		
A=A+1		
	R(A)	
	A=A+1	
	W(A)	
W(A)		
		R(B)
		B=B+1
		W(B)
	R(B)	

Figure 7: Multi-version timestamp ordering protocol

17. [Snapshot isolation, 10pts] Consider the following schedule under snapshot isolation protocol.

T1	T2	T3
A=1		
W(A)		
	R(A)	
	A=A+2	
	W(A)	
DISPLAY(A)		
COMMIT		
		R(A)
		A=A+3
		W(A)
		DISPLAY(A)
		COMMIT
	DISPLAY(A)	
	COMMIT	

Figure 8: Snapshot isolation protocol

(a) What will be displayed by each transaction? Assume the initial value of A is 0. Justify your answer.

(b) Is it possible that the snapshot isolation suffers from cascading rollbacks? Justify your answer using the schedule shown above, i.e., assume the commit operation of each transaction fails, and explain what will happen.

18. [Recovery, 20pts] Consider the following log sequence for log-based recovery scheme.

```
LogSeqNo Log
  1     <T1 start>
  2     <T1, X, 0, 100>
  3     <T2, start>
  4     <T2, Y, 0, 10>
  5     <T1, X, 100, 200>
  6     <T1, commit>
  7     <checkpoint {T2}>
  8     <T3, start>
  9     <T2, Y, 10, 20>
  10    <T3, Z, 0, 1>
  11    <checkpoint {T2,T3}>
  12    <T2, Y, 10>
  13    <T3, Z, 1, 2>
>>>>> CRASH!!!!!!
```

(a) Suppose the system crashed after Log Sequence No 13. The checkpoint file created by Log Sequence No 11 is valid. How many times X, Y, and Z will be written by the recovery process? Justify your answer.

(b) Describe what log entries will the recovery scheme append to recover from the failure.

(c) Suppose the system crashed after Log Sequence No 13, and the last_checkpoint in the fuzzy checkpointing scheme is pointing to the file checkpointed by LogSeqNo 7. How many times X, Y, and Z will be written by the recovery process? Justify your answer.

(d) Describe what log entries will the recovery scheme append to recover from the failure.