Database Management Systems
Practice Exam Questions

Practice problems (solutions are on the book's web page): Chapters 1 (all odd numbered exercises) , 8 (all odd numbered exercises), 9 (all odd, except 9.5), 11 (all odd), 12 (all odd), 13.1, 13.5, 14 (all odd except hybrid hash-join), 15 (all odd), 16 (all odd), 17.5, 17.9, 17.11, 18 (all odd), 22.1.3, 22.1.4. 22.3.3.

The exam will also include questions on the practical learning articles on the web page and questions related to the marking questions for the PLs.

Question 1 Answer true (T) or false (F).

____ 1. Parallel RDBMS are generally more fault-tolerant than map-reduce systems.

____ 2. Map-reduce provides the ACID properties.

____ 3. Index selection is important to the performance of map-reduce programs.

____ 4. Recoverable schedules are used to ensure transactions eventually commit.

____ 5. Using WAL (Write-Ahead Logging), when a page is written in the buffer (memory) pool, its log record must be on disk.

____ 6. Immediately after a transaction commits, all data pages written by the transaction must be on non-volatile storage to ensure transaction durability.

____ 7. Checkpointing is used to limit the amount of work required to record redo actions in the log.

____ 8. In a DBMS, recovery is used to ensure that a set of writes done by a transactions appear to a user to be atomic.

Question 2 Data Models (Article on Keyword Queries and Information Retrieval)

The Relational data model is often used for structured data. A unstructured data model uses bags of (key, value) pairs to model information. Describe an application for which the Relational data model would be best, and one for which an unstructured model would be best.

Question 3 Parallel DB

1. Define the terms scale-up and speed-up.
2. Define "shared-nothing architecture".
3. Why is a shared-nothing architecture attractive for parallel database systems?

Question 4 For this question consider a map-reduce systems (MR) that has 16 nodes and a parallel DBMS (PDBMS) that is running on 16 nodes. Assume the nodes are the same (i.e., the same physical hardware configurations/power).

1. We learned in class that typically the *response time* of queries will be faster (in some cases much faster) on the PDBMS than on MR. Explain why.
2. We also learned in class that the *fault tolerance* of a MR system is typically better than the PDBMS. Explain why and explain what types of faults this applies to.

Question 5 Map Reduce and Parallel DBMS.

An important part of web search is computing the reputation for a page. A simple form of reputation for page P is to count the number of other pages that link to P. Assume we have as input for each page, a key which is the page's unique URL, and a value which is a list of links (that is, a list of URLs) found on that page. Then the following map reduce program computes the reputation of each page.

As an example of the input, on Steve Cook's homepage there are links to the Computer Science Dept page and to the department's theory group. So the input to the map function for Steve's page would contain the key "http://www.cs.toronto.edu/∼sacook/" (the URL for his page) and the value would be a list of the links (URLs) on Steve's page ("http://www.cs.toronto.edu", "http://www.cs.toronto.edu/theory/", ...). This map reduce program computes the reputation of all pages (URLs) that are linked to by at least one other page.

```
map(String key, String value):
    for each URL in value:
        EmitIntermediate(URL, key);

reduce(String key, Iterator values):
    int result = 0:
    for each v in values:
        if key ≠ v then result++:
    Emit(key, AsString(result));
```

Suppose we want to compute the same result using a relational Parellel DBMS. We start by normalizing the input data into a BCNF relational table `Page(URL, Link)` where the primary key is `(URL, Link)` and the `Link` attribute contains a single link.

1. Give an SQL query that computes the same result as the map reduce program above.
2. Next discuss what changes you would make to the map reduce program and to the SQL to output the URLs where the rank is above a threshold θ_R.

Question 6 Transaction Management Indicate whether each of the following schedules is *serial*, *view serializable*, *conflict serializable*, *recoverable*, *avoids cascading aborts*, *2PL*, *strict 2PL*. Recall that a 2PL schedule is one that could be produced by a two-phase locking scheduler. A strict 2PL schedule is one that could be prodcued by a two phase locking scheduler which holds write locks until end of transaction.

1. $R_1(A), W_2(A), R_1(B), Commit_1, W_3(B), R_3(B), W_3(A), Commit_3, R_2(C), Commit_2$.
2. $R_1(A), W_2(B), R_1(B), Commit_1, Commit_2$
3. $R_1(A), W_2(B), R_1(B), Commit_2, Commit_1$
4. $R_1(A), W_2(B), Commit_2, R_1(B), W_1(B)_1, Commit_1$
5. $W_1(A), W_2(A), Commit_2, W_1(A), Commit_1$

Question 7 Recovery Consider the following sequence of undo/redo log records (where each record for a transaction write includes the LSN, the transaction id, the log record type, the previous LSN. For updates, the record also includes the object id, the before-image (former value) of the object and the after-image (new value) of the object). Compensation log records contain the same information as update records with one final value which is the undonextLSN (the next LSN that needs to be undone). For brevity, we don't include the length and offset of the change. The transaction ids are S, T, U and V. The object ids are A, B, C, D, E and F. The number at the start of each log record indicates the log sequence number.

[03,S,UPDATE, ∅, A,1,2]
[04,U,UPDATE, ∅, C,1,2]
[05,U,COMMIT,04]
[06,U,END,05]
[07,STARTCHECKPOINT]

[08,T,UPDATE, ∅, D,1,2]
[09,V,UPDATE,∅,C,2,3]
[10,T, COMMIT,T,08]
[11,V,UPDATE,09,E,1,2]
[12,ENDCHECKPOINT,DPT(C,04), XT(S,03)]
[13,S,UPDATE,03,A,2,3]
[14,STARTCHECKPOINT]
[15,V,ABORT,11]
[16,S,COMMIT,13]
[17,V,CLR,15,E, 2, 1, 09]

1. **Recovery** Suppose a memory crash occurs after the log record 17 is written to disk.
   (a) What is the set of transactions that must be undone? _____
   (b) What is the set of transactions that must be redone? _____
   (c) Would the set of redo-transactions change if the dirty page table were empty in LSN 12? _____
   (d) List, in order, the redo actions that must be done. Be specific by also giving all log records written as part of redo. For partial credit, show your work for the analysis phase.
   (e) List, in order, the undo actions that must be done. Be specific by also giving all log records written as part of undo.
2. Answer same question but suppose the crash occurs after the log record 13 is written to disk.

Question 8 Recovery

Consider a log produced by a write-ahead-log protocol where log records are forced to non-volatile storage at transaction commit. Assume page-level logging, that is, a log record may only contain undo/redo information for a single page.

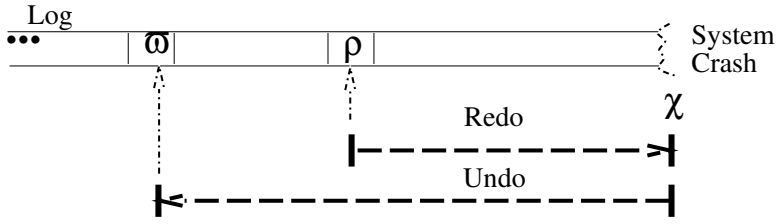


Figure 1: Redo and undo phases of recovery.

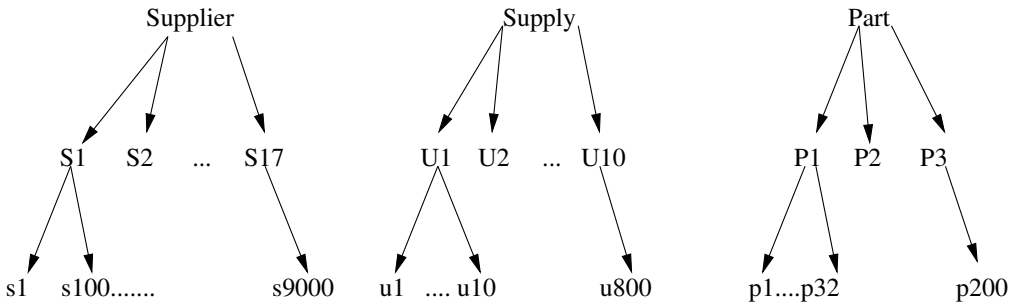
Assume the system crashes and all volatile memory is lost. During recovery:

1. Can ρ, the start of the redo phase, ever precede ϖ, the end of the undo phase, in the log? Why or why not?
2. Suppose the buffer manager uses a no steal policy so dirty pages are not written to disk before commit. Does this change eliminate the need for a redo phase? Why or why not? Does this change eliminate the need for an undo phase? Why or why not?

Question 9 Consider the following update.

```
UPDATE Part set price = price * 1.1
WHERE pid in ( SELECT U.pid
               FROM Supply U, Supplier S
               WHERE S.sid = U.sid AND S.city = 'Toronto' )
```

Suppose hierachical lock is used and that there are no indices available. The tuples are stored contiguously on pages and locks can be placed on the relations, their pages, or their tuples (the hierarchy is shown below). What locks would be obtained by a 2-phase locking scheduler to execute this update?



Question 10 Consider what indices might be useful for the update in the previous question. Indicate yes (Y) or no (N).

____ 1. Would an optimizer consider the use of a clustered hash index on Supply with key sid?

____ 2. Would an optimizer consider the use of a clustered B+tree index on Part keyed on price?

____ 3. Executing the update above with the update `insert into part values (101, 'hammer', 'Paris', 100)` could cause a deadlock.

____ 4. Assuming no indices are available, then this update could cause a phantom.

Question 11 Consider the following sequence of actions, listed in the order they are submitted (note this is submitted order, not scheduled order) to the DBMS:

Sequence S1: T1:R(X), T2:W(X), T2:W(Y), T3:W(Y), T1:W(Y)

For each of the following concurrency control mechanisms, describe how the concurrency control mechanism handles the sequence. Assume that the timestamp of transaction Ti is i (so Ti has higher priority than Tj if $i < j$). For lock-based concurrency control mechanisms, add lock and unlock requests to the above sequence of actions as per the locking protocol. The DBMS processes actions in the order shown. If a transaction is blocked, assume that all of its actions are queued until it is resumed; the DBMS continues with the next action (according to the listed sequence) of an unblocked transaction. Your answer should clearly indicated the **scheduled** sequence for each protocol.

1. Strict 2PL with timestamps used for deadlock prevention. Assume we use Wait-Die policy.
2. Strict 2PL with deadlock detection. (Show the waits-for graph if a deadlock cycle develops.)