Q 1. [35pts] For the following questions, clearly circle True or False.

1. Locks on phantoms are what distinguish the SERIALIZABLE isolation level from the REPEATABLE READ isolation level.
   True

2. In the unpacked record layout on pages for fixed-length records, deleting records alters the record IDs of other records on the same page.
   False

3. Schema matching is a harder problem than schema mapping.
   False

4. A recoverable schedule is always guaranteed to not have cascading aborts.
   False

5. An avoiding-cascading-aborts (ACA) schedule is always guaranteed to not have cascading aborts.
   True

6. An avoiding-cascading-aborts (ACA) schedule is always guaranteed to be a recoverable schedule.
   True

7. A recoverable schedule is always guaranteed to be an avoiding-cascading-aborts (ACA) schedule.
   False

8. In the delimiter-based record format, we need to scan the record from the start even to retrieve a single field in the middle.
   True

9. A hash index is typically very efficient for answering selection queries with \( \neq \) predicates (denoted \( <> \) in SQL).
   False
10. Spark’s API mostly subsumes the MapReduce programming abstraction.
   True

11. Apart from schema information, the RDBMS catalog also stores statistics about both relation instances and indexes.
   True

12. All four SQL isolation levels guarantee recoverability.
   False

13. All four SQL isolation levels guarantee serializability.
   False

14. All four SQL isolation levels guarantee that there will be no WW conflicts during a concurrent execution of transactions.
   True

15. A serializable schedule is not sufficient to ensure compliance with ACID properties.
   True

16. The goal of information extraction is to convert a relation into a piece of unstructured text.
   False

17. It is impossible to have a B+ tree index that is a clustered index with respect to more than one IndexKey.
   False

18. A typical magnetic hard disk has one arm.
   True

19. Variance is an algebraic aggregate that can be computed in just one pass over the table.
   True

20. A serializable schedule is always equivalent to exactly one serial schedule.
   False

   False

22. Recoverability implies serializability.
   False
23. Rename is the only operator in relational algebra whose semantics depends only on the database schema and not the database instance.
   True

24. Global-As-View is much more flexible for adding or removing data sources compared to Local-As-View when handling schema alignment.
   False

   False

26. Blocking heuristics in entity matching workflows help avoid a computationally expensive all-pairs comparison for the pairwise checker.
   True

27. The performance gap between random access and sequential access observed on magnetic hard disks is much narrower on non-volatile memory.
   True

28. Redistribution of index keys among siblings following an underflow of a node is the first preference for reorganizing a B+ tree index after a delete operation.
   True

29. Given a logical query plan, there is only one physical query plan that it can be translated to.
   False

30. Open information extraction differs from closed information extraction in that the attributes of entities may not be known in the former.
   True

31. The average rotational delay of a disk is a function of its RPM.
   True

32. Different value representations for the same entity is one of the issues that makes data cleaning hard.
   True

33. It is sometimes possible to obtain superlinear speedup in a parallel RDBMS.
   True

34. All primary indexes are clustered indexes.
   False
35. The pain of programming with the user-defined function/aggregate abstraction of most RDBMSs is one of the major reasons for the popularity of Spark.

True

Q 2. [50pts] Clearly circle the correct answer for each of the following questions (only one option is correct).

1. Which improvement discussed in class for the external merge sort is motivated by the idea of exploiting direct memory access?

(a) Internal replacement sort  (b) Blocked I/O  (c) Double buffering

ANSWER: (c) Double buffering.

2. Which of the following relational operators usually can not be sped up (in terms of I/O costs) with a B+ tree index at all?

(a) $\sigma$  (b) $\gamma$  (c) $\times$  (d) $\Join$  (e) $\pi$

ANSWER: (c).

3. Given a relation $R(A, B, C)$, a hash index on which IndexKey does not match the following query: $\sigma_{A=a, B=b}(R)$?

(a) On $B$  (b) On $A$  (c) On $(B, A)$  (d) On $(B, C)$

ANSWER: (d) On $(B, C)$.

4. What is the minimum number of hash functions needed for the two-phase “improved” hash join implementation in a parallel RDBMS?

(a) 0  (b) 1  (c) 2  (d) 3  (e) 4  (f) 5

ANSWER: (d) 3.
5. In practice, what is the minimum percentage occupancy of a page typically maintained by a B+ tree index?

(a) 10%    (b) 25%    (c) 50%    (d) 75%    (e) 100%

**ANSWER:** (c) 50%.

6. Which of the following invariants/conditions always hold for an extendible hash index? GD and LD stand or global and local depths respectively.

(a) All LD ≤ GD    (b) All LD < GD    (c) All LD ≥ GD    (d) All LD > GD

**ANSWER:** (a) All LD ≤ GD.

7. Which of the following improvements of external merge sort does not affect the fan-in of the merge phase?

(a) Internal replacement sort    (b) Blocked I/O    (c) Double buffering

**ANSWER:** (a) Internal replacement sort.

8. Given this four-table join query $R \bowtie S \bowtie T \bowtie U$, how many right deep tree orderings exist (assume we only use the hash join implementation)?

(a) 1    (b) 2    (c) 6    (d) 24    (e) 120

**ANSWER:** (d) 24. Since there are 4 spots, there are $4! = 24$ right-deep orderings.

9. In a B+ tree index, which nodes are allowed to have duplicates of IndexKey values?

(a) Root node    (b) Leaf nodes    (c) Non-leaf nodes except root

**ANSWER:** (b) Leaf nodes.

10. Column stores are typically used to speed up primarily which relational operation?

(a) $\bowtie$    (b) $\cup$    (c) $\cap$    (d) $-$    (e) $\gamma$


ANSWER: (e) γ.

11. Which of the following SQL capabilities have no counterpart in (extended) relational algebra?

(a) ORDER BY    (b) NATURAL JOIN    (c) GROUP BY    (d) SELECT DISTINCT

ANSWER: (a) ORDER BY.

12. Which conflict is not guarded against by the READ COMMITTED isolation level?

(a) WW    (b) WR    (c) RW    (d) None (all are avoided)

ANSWER: (c) RW.

13. Which type of information extraction requires extraction of instance values of entities’ attributes?

(a) Closed-world    (b) Closed    (c) Open    (d) All of them!

ANSWER: (d) All of them!

14. You are given two data sources with a table each that need to be integrated. The source schemas are S1 (LastName, FirstName, Salary) and S2 (FullName, GrossSalary, NetSalary, Age). Which of the following is a usable mediated schema for Local-As-View?

(a) FullName, GrossSalary, NetSalary

(b) LastName, FirstName, Age, GrossSalary, NetSalary

(c) LastName, FirstName, Salary, Age

(d) LastName, Salary, Age, GrossSalary

(e) LastName, FirstName, NetSalary, Age

ANSWER: (b).
15. Which attribute in this dataset surely needs data cleaning?

<table>
<thead>
<tr>
<th>Title</th>
<th>Year</th>
<th>Director</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avatar</td>
<td>2009</td>
<td>James Cameron</td>
</tr>
<tr>
<td>Inception</td>
<td>92093</td>
<td>Christopher Nolan</td>
</tr>
<tr>
<td>Moonlight</td>
<td>2016</td>
<td>Barry Jenkins</td>
</tr>
</tbody>
</table>

(a) Title      (b) Year      (c) Director    (d) None of them

**ANSWER:** (b) Year.

16. Given the relation $R(A, B, C, D)$ with primary key $A$ and alternate key $D$, which of the following terms does *not* apply to an index with IndexKey $(C, D)$?

(a) Composite index  (b) Primary index  (c) Secondary index  (d) Unique index

**ANSWER:** (b) Primary index.

17. You are given two tables $R$ and $S$ of sizes (in number of pages) $N_R$ and $0.5N_R$ respectively. The number of buffer pages available is $B + 1$. Which of the following is a necessary condition for $B$ for the hash join implementation to proceed without any partitioning of the base tables? Assume the fudge factor is 1.4.

(a) $B > 1.4N_R$  (b) $B > N_R$  (c) $B > 0.9N_R$  (d) $B > 0.7N_R$

**ANSWER:** (d) $B > 0.7N_R$.

18. Which index alternative almost always leads to the largest index size?

(a) AltRecord    (b) AltRID    (c) AltRIDList

**ANSWER:** (a) AltRecord.

19. Which join operator implementation will have the lowest I/O cost if you have infinite buffer memory available?

(a) PNLJ  (b) BNLJ  (c) HJ  (d) SMJ  (e) All have the same cost!

**ANSWER:** (e) All have the same cost!
20. What is touted as the biggest benefit of a learned index structure over a classical deterministic index structure?

(a) Lower lookup runtime complexity  (b) Lower space complexity
(c) Lower update runtime complexity  (d) Lower retrieval accuracy

**ANSWER:** (b) Lower space complexity.

21. You are given the table $R(RID, UID, MID, Time, Stars)$, wherein $Stars$ is a numeric attribute. You are also given an AltRID B+ tree index on it with the IndexKey $(MID, UID, Stars)$. The buffer pool is given to be empty. Which of the following aggregate queries can not always be executed using only a sequential scan of the leaf level of this index?

(a) $\gamma_{MID,AVG(Stars)}(R)$  (b) $\gamma_{UID,AVG(Stars)}(R)$
(c) $\gamma_{MID,UID,AVG(Stars)}(R)$  (d) $\gamma_{UID,MID,AVG(Stars)}(R)$

**ANSWER:** (b) $\gamma_{UID,AVG(Stars)}(R)$.

22. Given the same database, index, and buffer pool status as the previous question, which of these select-project queries may have an I/O cost higher than that of a sequential scan of the leaf level of the index?

(a) $\pi_{MID}(\sigma_{Stars>3.0}(R))$  (b) $\pi_{MID}(\sigma_{UID=1729}(R))$
(c) $\pi_{UID}(\sigma_{UID=1729}(R))$  (d) $\pi_{UID}(\sigma_{Stars>3.0}(R))$

**ANSWER:** (d) $\pi_{UID}(\sigma_{Stars>3.0}(R))$.

23. Which of the following relational algebra operators is not commutative?

(a) $\bowtie$  (b) $-$  (c) $\cup$  (d) $\times$  (e) $\cap$

**ANSWER:** (b) $-$. 

24. Which of the following paradigms of parallelism is followed by most parallel RDBMSs and dataflow systems?
(a) Shared-memory  (b) Shared-disk  (c) Shared-everything  (d) Shared-nothing

ANSWER: (d) Shared-nothing.

25. You are given two union-compatible tables $R(X, Y)$ and $S(X, Y)$ with attributes $X$ and $Y$ being of the same size. The sizes (in number of pages) of $R$ and $S$ are $2B$ and $10B$ respectively, where $B$ is the number of buffer pages given (in the millions). Assume there are no skews in any attribute distributions. Hash table fudge factor is 1.4. What is the I/O cost of executing the join $R \bowtie S$ using BNLJ (assume an in-memory hash tables is used on blocks)?

(a) 12B  (b) 14B  (c) 24B  (d) 32B  (e) 36B

ANSWER: (d) 32B. $R$ is the outer table for BNLJ, since it is smaller. I/O cost is then $N_R + N_S \cdot \lceil FN_R / (B - 2) \rceil$. Since $B$ is in the millions, $B / (B - 2) \approx 1$. So, the I/O cost is $2B + 10B \cdot 3 = 32B$.

Q 3. [15pts] Consider the following two transactions operating over two distinct data objects $A$ and $B$. For each question that follows, clearly circle the correct answer.

T1: R(A), W(B), R(A), W(A), Commit
T2: W(A), W(B), Commit

1. [3pts] Which of the following schedules has a WW conflict?
   
   (a) R1(A), W2(A), W1(B), W2(B), R1(A), W1(A), Commit2, Commit1  
   (b) W2(A), R1(A), W2(B), W1(B), R1(A), Commit2, W1(A), Commit1  
   (c) R1(A), W1(B), W2(A), W2(B), R1(A), W1(A), Commit1, Commit2  
   (d) W2(A), R1(A), W1(B), W2(B), R1(A), Commit2, W1(A), Commit1  
   (e) All of the above have WW conflicts!

ANSWER: (e). Every one of them has overwrites of dirty data!
2. [3pts] Which of the following schedules does not have an unrepeatable read conflict?

(a) R1(A), W2(A), W1(B), W2(B), R1(A), W1(A), Commit2, Commit1

(b) W2(A), R1(A), W2(B), W1(B), R1(A), Commit2, W1(A), Commit1

(c) R1(A), W1(B), W2(A), W2(B), R1(A), W1(A), Commit1, Commit2

(d) None of the above has an unrepeatable read conflict!

**ANSWER: (b).**

3. [3pts] Which of the following schedules is serializable?

(a) R1(A), W2(A), W1(B), W2(B), R1(A), W1(A), Commit2, Commit1

(b) W2(A), R1(A), W2(B), W1(B), R1(A), Commit2, W1(A), Commit1

(c) R1(A), W1(B), W2(A), W2(B), R1(A), W1(A), Commit1, Commit2

(d) W2(A), R1(A), W1(B), W2(B), R1(A), Commit2, W1(A), Commit1

(e) None of the above is serializable!

**ANSWER: (b).** It is equivalent to T2 → T1. You can verify that the others are not equivalent to any serial schedule.

4. [3pts] Which of the following schedules is not recoverable?

(a) R1(A), W2(A), W1(B), W2(B), R1(A), W1(A), Commit2, Commit1

(b) W2(A), R1(A), W2(B), W1(B), R1(A), Commit2, W1(A), Commit1

(c) R1(A), W1(B), W2(A), W2(B), R1(A), W1(A), Commit1, Commit2

(d) W2(A), R1(A), W1(B), W2(B), R1(A), Commit2, W1(A), Commit1
(e) All of the above are recoverable!

**ANSWER:** (c). T1 is reading dirty A from T2 but commits before T2 ends.

5. [3pts] Suppose we use strict 2-phase locking for concurrency control. A transaction acquires an appropriate lock on a data object at the time it first needs to access that data object. Assume a transaction can only get one kind of lock on a data object (exclusive or shared). Which of the following schedules will lead to a deadlock?

(a) R1(A), W2(A), W1(B), W2(B), R1(A), W1(A), Commit2, Commit1

(b) W2(A), R1(A), W2(B), W1(B), R1(A), Commit2, W1(A), Commit1

(c) R1(A), W1(B), W2(A), W2(B), R1(A), W1(A), Commit1, Commit2

(d) W2(A), R1(A), W1(B), W2(B), R1(A), Commit2, W1(A), Commit1

(e) None of them will lead to a deadlock!

**ANSWER:** (e). Note that both transactions start with an operation on A. Thus, the transaction that starts first will acquire a long exclusive lock on A. So, the other transaction will be forced to wait until the first one finishes, since it will also have to request a long exclusive lock on A (these two locks are not compatible). Thus, strict 2PL will transform each of the given schedule into what is effectively a serial schedule!