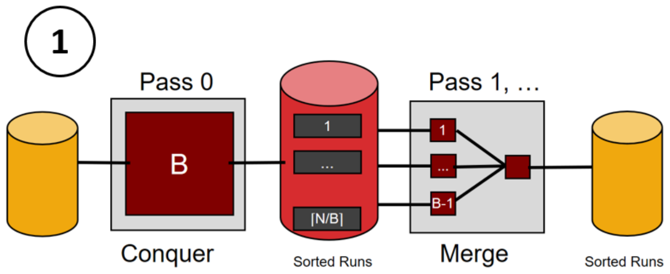
CS150: Database and Data Mining
Final Exam Solutions

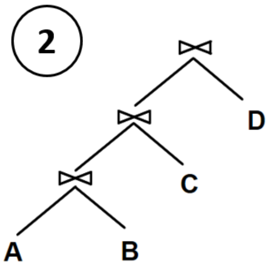
December 28, 2021

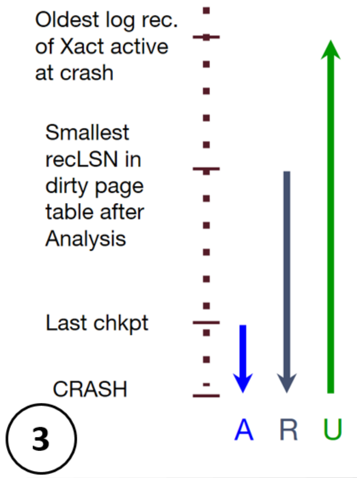
I Basics [10 points]

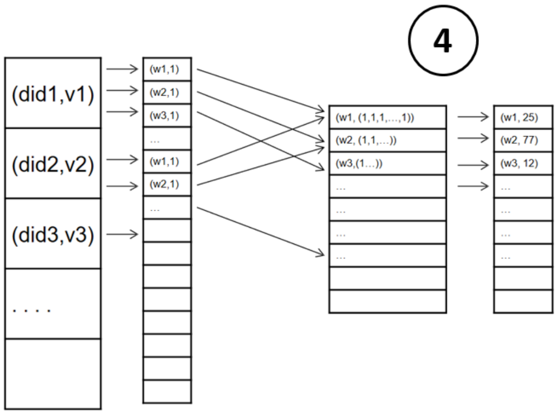
For each image on the next page, select the letter corresponding to the best description.
A. Left Deep Tree
B. Key Compression
C. B+ Tree
D. Spark
E. Nested Loops Join
F. Sort Merge Join
G. Page Nested Loop Join
H. Slotted Page
I. Variable Length Tuple
J. Fixed Length Tuple
K. Buffer Frame
L. Sort-based Group-by
M. Recovery
N. Two Phase Lock
O. JSON
P. ISAM
Q. External Sort
R. MapReduce

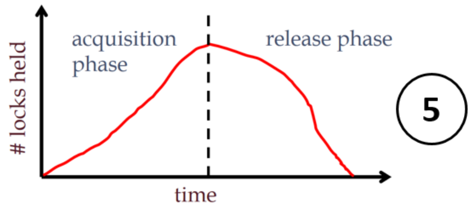
Solution
1. Q
2. A
3. M
4. R
5. N
6. H
7. O
8. C
9. D
10. G

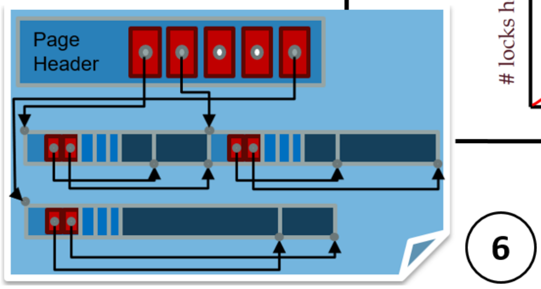





Oldest log rec. of Xact active at crash
Smallest recLSN in dirty page table after Analysis
Last chkpt
CRASH
A
R
U
3

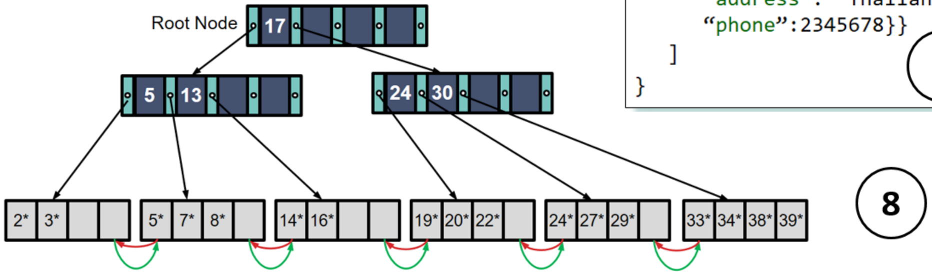






```
{"person":
    [ {"name": "Mary",
      "address":
        {"street":"Maple",
         "no":345,
         "city": "SF"}},
     {"name": "John",
      "address": "Thailand",
      "phone":2345678}}
    ]
}
```

7

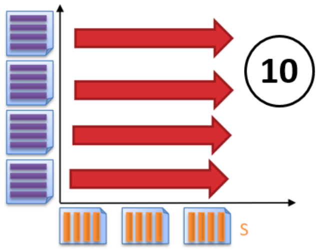


R(A,B)
S(A,C)

SELECT count(*) FROM R, S
WHERE R.B > 200 and S.C < 100 and R.A= S.A

```
R = s.read().textFile("R.csv").map(parseRecord).persist();
S = s.read().textFile("S.csv").map(parseRecord).persist();
RB = R.filter(t -> t.b > 200).persist();
SC = S.filter(t -> t.c < 100).persist();
J = RB.join(SC).persist();
J.count();
```

9



II SQL [12 points]

1. [**2 points**] Which of the following expressions computes the matrix vector product:

$$(\mathbf{Ax})_i = \sum_{k=1}^{d} A_{ik} x_k$$

Assume **A** and **x** have compatible dimensions and there is only one correct answer.

A.

```
SELECT A.row AS row, SUM(A.value * x.value) AS value
FROM A JOIN x
ON A.col = x.row
GROUP BY A.row
```

B.

```
SELECT A.row AS row, SUM(A.value * x.value) AS value
FROM A JOIN x
ON A.row = x.row
GROUP BY A.col
```

C.

```
SELECT x.row AS row, SUM(A.value * x.value) AS value
FROM A JOIN x
ON A.col = x.row
GROUP BY A.col
```

D.

```
SELECT A.row AS row, A.value * x.value AS value
FROM A JOIN x
ON A.row = x.row
```

Solution
A: The final answer should be grouped by the row of A and therefore sum over the product along the columns.

2. [**2 points**] Suppose we wanted to compute the element-wise sum of the vectors **x** and **y** using the SQL expression:

```
SELECT x.row AS row, SUM(x.value + y.value) AS value
FROM x JOIN y
ON x.row = y.row
```

Which of the following statements about this query are true? (You may mark zero (ϕ), one or more than one of the choices.)
A. Some non-zero entries may be omitted from the final result.
B. The correct query should use LEFT OUTER JOIN.
C. The correct query should use FULL OUTER JOIN.
D. There is nothing wrong.

Solution
A, C: There was a typo in this question (sum should be removed). The issue with this query is the combination of two factors. First, the two vectors may have zeros in different places, and in a sparse vector the zero entries are represented by the absence of information (missing row IDs in the table). Second, by default, joins in SQL will discard a tuple from one table unless it has a match in the other table; OUTER JOINs preserve these non-matching tuples.
Consider the following:

Vector a

Row	Value
1	3.0
3	4.0

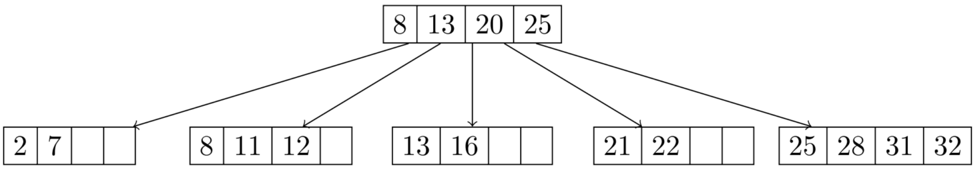
Vector b

Row	Value
2	2.0
5	8.0

These two tables produce no outputs on a typical (inner) join, representing a vector of all 0's. We need to do a full outer join to ensure that we get rows 1, 2, 3 and 5 in the output.

III Indexes and B+ Trees [9 points]

1. [**4 points**] Consider the following B+ tree of order 2.



(a) [**1 point**] How many nodes split when you insert 27?

Solution

- To insert 27, we follow the rightmost pointer in the tree from the root node because $27 > 8$, $27 > 13$, $27 > 20$, $27 > 25$.
- We get to the leaf node containing (25, 28, 31, 32) and attempt to insert 27, but the leaf node is at maximum capacity and the insertion breaks the occupancy invariant. Therefore we must split the leaf into (25, 27) and (28, 32, 21) and copy up key 27 because this is a leaf node.
- We attempt to insert 27 into the parent node containing (8, 13, 20, 25) but the root is also at maximum capacity. We split it into (8, 13) and (25, 28) pushing up key 20 because it is an inner node. Our final tree looks like and we split two nodes.

(b) [**1 point**] After inserting 27 into the tree, you also insert 26. How many nodes split as a result of inserting 26?

Solution

- To insert 26, we first look at the root node and follow the right pointer because $26 \geq 20$. We then follow the middle pointer at the inner node below the root node because $26 > 25$ but $26 < 28$. We get to the leaf node containing (25, 27)
- We attempt to insert 26 into the leaf node containing (25, 27) and since 3 is less than the occupancy invariant 4 we insert 26 and are done. We did not split any nodes.

(c) [**2 points**] Assume that after inserting 26 and 27, you insert the keys 34, 35, 36, . . ., 100. After all these insertions, what keys are in the leftmost leaf node?

Solution
You should be able to identify that we are only inserting keys into the right side of the tree. This is similar to bulk loading in that the left leaf nodes will not be touched. Therefore, we know that the leftmost leaf node will not be modified and will contain (2, 7) after the insertions to the right side of the tree.

IV Buffer Mangagement [8 points]

1. [**6 points**]

   Supposed we have a buffer pool size of 4 pages, and the following access pattern:
   A P P L E S A N D B A N A N A S A N D O R A N G E S
   Assume that pages are unpinned immediately (ignore pinning).

   (a) [**2 points**] What is the number of cache hits if we use an LRU replacement policy? Assume we are starting from a cold (empty) cache.

   Solution
   8

   (b) [**2 points**] What is the number of cache hits if we use an MRU replacement policy? Assume we are starting from a cold (empty) cache

   Solution
   5

   (c) [**2 points**] What is the number of cache hits if we use a CLOCK replacement policy? Assume we are starting from a cold (empty) cache.

   Solution
   8

   (d) [**2 points**] What is the number of set reference bits at the end of (c)?

   Solution
   4

V External Sorting [8 points]

1. **[6 points]** Assume that each page is 4 KB large, and that you have a 24KB buffer pool (with 6 frames).

   (a) **[2 points]** How many passes would it take to externally sort an 512KB file? Include the initial sorting pass and subsequent merging passes in your answer. You need to simplify your answer.

   > **Solution**
   > 3: The file contains 128 pages, and B = 6, and thus we need $\left(1 + \left\lceil \log_5 \left(\left\lceil \frac{128}{6} \right\rceil\right) \right\rceil = 3\right)$ passes.

   (b) **[2 points]** What would be the total cost in I/Os for this external sort?

   > **Solution**
   > 768: 2∗128∗#passes = 256*3 = 768 (I/Os).

   (c) **[2 points]** What is the minimum number of additional buffer frames we require to reduce the number of passes (from part 1) by 1?

   > **Solution**
   > 6: In order to sort the file in two passes, we need B buffer frames where $B(B-1) \geq 128$. The smallest such B is 12. Thus, the number of additional pages is $12 - 6 = 6$.

VI Join Algorithms [8 points]

1. **[5 points]** Consider a relation R with attributes (x, y) and a relation S with attributes (y, z). Column y in S is a key and the set of values of y in R are the same as the set of values of y in S. Assume that there are no indexes available and that there are 25 pages in the buffer available. Table R is 1,500 pages with 50 tuples per page. Table S is 400 with 100 tuples per page. Compute the I/O costs for the following joins. Assume the simplest cost model, where pages are read and written one at a time.

   (a) **[1 point]** Block nested loops join with R as the outer relation and S as the inner relation.

   Solution
   $|R| + \lceil |R|/B \rceil \times |S| = 1500 + \lceil 1500/25 \rceil \times 400 = 25500$ OR
   $|R| + \lceil |R|/(B-1) \rceil \times |S| = 1500 + \lceil 1500/24 \rceil \times 400 = 26700$ OR
   $|R| + \lceil |R|/(B-2) \rceil \times |S| = 1500 + \lceil 1500/23 \rceil \times 400 = 27900$
   Since, these three equations all appears in some materials, any were acceptable.

   (b) **[1 point]** Sort merge join with S as the outer relation and R as the inner relation.

   Solution
   12500: Same as (c).

   (c) **[1 point]** Hash join with S as the outer relation and R as the inner relation.

   Solution
   5700: $3(\lceil R \rceil + \lceil S \rceil) = 3(1500 + 400) = 5700$.

2. **[3 points]** Consider a new case, i.e. B>4 pages worth of buffer space, and relations M and N of size > B. Please fill the blanks below with "always", "sometimes" or "never".

   (a) **[1 point]** Block nested loop join is ______ better than page-oriented nested loop join.

   Solution
   Always: Block nested loop join is page-oriented on steroids, and steroids are always good.

   (b) **[1 point]** Sort-merge join is ______ better than hash-join.

   Solution
   Sometimes: Hash join is cooler if one relation is really small and one is very large. Sort dominates if you have lots of duplicate join keys.

   (c) **[1 point]** Hybrid Hash-Join is ______ better than block-nested loops join.

   Solution
   Sometimes: Nested loops works for non-equijoins and hash does not. For equijoins, hash is often better since it only makes a small number of passes over each relation, whereas block nested-loops still may visit the inner relation many times. If one relation fits in memory, the two algorithms are about equivalent.

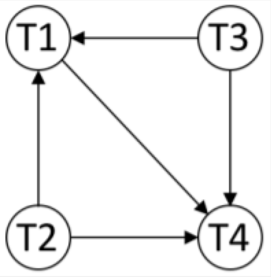
VII Transactions and Concurrency [20 points]

1. [**4 points**] **Transactions**
Consider the following schedule. (For each of the questions below, you may mark zero (ϕ), one or more than one of the choices.)

	T1	T2	T3	T4
1	R(A)			
2		R(A)		
3			R(C)	
4			W(C)	
5		R(B)		
6		W(B)		
7	R(B)			
8				R(B)
9	R(C)			
10				R(C)
11				W(B)
12		commit		
13			commit	
14	commit			
15				commit

(a) [**3 points**] Please draw the conflict dependency graph of the above schedule?

Solution



(b) [**3 points**] Which of the following schedules below are conflict equivalent to the schedule above?
A. T3, T1, T2, T4
B. T2, T3, T1, T4
C. T4, T3, T1, T2
D. T1, T2, T3, T4
E. T3, T2, T1, T4

Solution
B, E.
Just by doing a topological sort, we know that T4 has to be last and T1 has to be second to last. Whether T2 or T3 comes first is irrelevant since they have no incoming edges, so we have 2 possible conflict equivalent schedules.

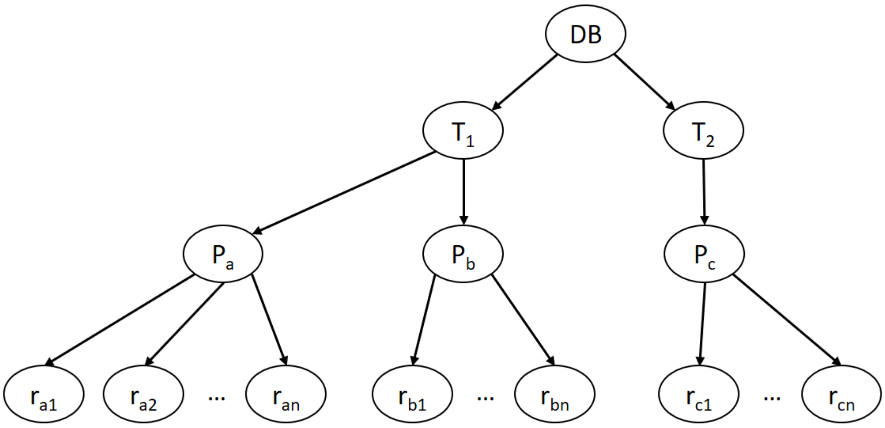
(c) [**3 points**] Mark all the correct choices.
A. In Strict 2PL, we can give up locks after aborting but before rollback is complete.
B. Schedules that are conflict serializable can not produce a cyclic dependency graph.
C. 2PL does not promise conflict serializability.
D. Strict Two Phase Locking ensures that we do not have deadlocks.

Solution

B.

- A is false. For A, you must wait until rollback is complete before giving up locks.
- B is correct. For B, 2PL enforces conflict serializability but may not allow all conflict serializable schedules (e.g. W1(X), R2(X), W1(Y), R2(Y) is impossible under 2PL).
- C is false.
- D is false. For D. Strict 2PL does not help us avoid deadlocks: if T1 wants X(A), X(B), and T2 wants X(B), X(A), we can get into a deadlock if T1 acquires X(A) and T2 acquires X(B).

2. [4 points] **Lock Manager**
Given the database system shown below:



(a) Which lock modes (including IS, IX, or SIX) on which resources are necessary to read P_a?

(b) Which lock modes (including IS, IX, or SIX) held by other transactions on P_a would prevent us from modifying r_{a1}?

Solution

(a) We would need the IS lock mode on DB and T1, and the S lock mode on P_a. This allows us to read from P_a while restricting other transactions as little as possible.

(b) S, SIX, and X lock modes held by other transactions on P_a would prevent us from holding an X lock on r_{a1}, which is necessary to modify r_{a1}. IX and IS locks would not prevent us, as the actual X or S locks held by other transactions are not necessarily on r_{a1}.

VIII Logging and Recovery [20 points]

1. [4 points] **ARIES Algorithm**
   Select all the true statements:
   A. Write Ahead Logging describes a protocol where updated pages must be written to disk before a crash.
   B. During a transaction abort, we undo all data updates made by the transaction.
   C. In ARIES, UPDATE log records contain no information of the previous state of the page.
   D. The recovery manager is responsible for Atomicity and Consistency, as defined by the ACID acronym.

   Solution
   B

2. [5 points] **General Logging and Recovery.**
   Mark the boxes for all true statement(s):

   (a) Schedules produced by two phase locking are guaranteed to prevent cascading aborts.

   Solution
   F: Strict 2PL is needed to guarantee this.

   (b) Strict two phase locking is both necessary and sufficient to guarantee conflict serializability.

   Solution
   F: Sufficient but not necessary.

   (c) In a system that uses strict two-phase locking, if a transaction aborts, it releases all of its locks as soon as rollback is complete.

   Solution
   T.

   (d) In a system that uses strict two-phase locking, a transaction that only performs reads can never enter a deadlock cycle.

   Solution
   F.

   (e) When aborting a transaction, it is necessary to modify pages on disk.

   Solution
   T. steal policy.

   (f) During recovery, the ARIES protocol redo aborted transactions.

   Solution
   T.

   (g) When a transaction commits, any modified buffer pages must be written to durable storage.

   Solution
   F: no force policy.

   (h) In ARIES recovery, after the analysis phase, the recLSN of each page in the dirty page table must be larger than the pageLSN of the corresponding page.

   Solution
   F: The page could have been updated and flushed from the buffer pool between the last checkpoint and time of crash. The flushed page would have a pageLSN from its most recent update, which is after the recLSN in the checkpoint.

   (i) If PageLSN is greater than the max LSN flushed so far (flushedLSN), we can safely write this page to disk.

Solution

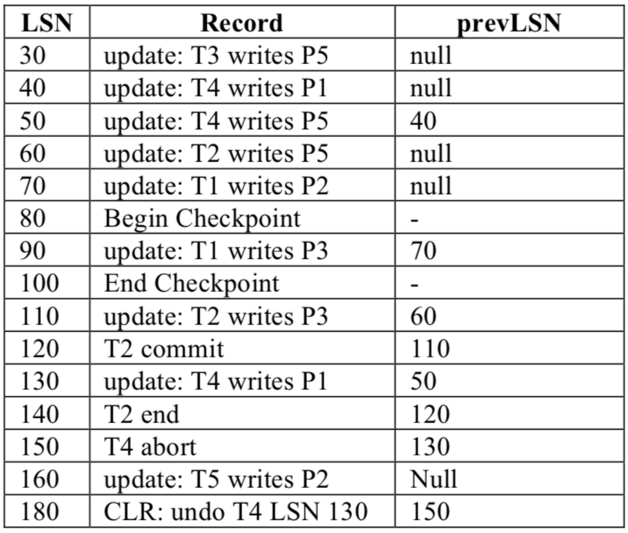
T: WAL requires that.

(j) Write-Ahead Logging (WAL) guarantees that a transactions log records are flushed to disk before the transaction commit.

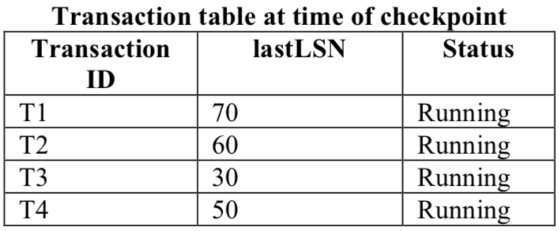
Solution

T.

3. [**8 points**] **Recovery.**
Your database server has just crashed due to a power outage. You boot it back up, find the following log and checkpoint information on disk, and begin the recovery process. Assume we use a STEAL/NO FORCE recovery policy.

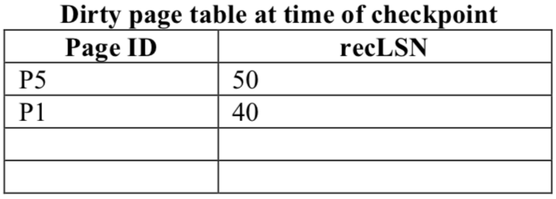


LSN	Record	prevLSN
30	update: T3 writes P5	null
40	update: T4 writes P1	null
50	update: T4 writes P5	40
60	update: T2 writes P5	null
70	update: T1 writes P2	null
80	Begin Checkpoint	-
90	update: T1 writes P3	70
100	End Checkpoint	-
110	update: T2 writes P3	60
120	T2 commit	110
130	update: T4 writes P1	50
140	T2 end	120
150	T4 abort	130
160	update: T5 writes P2	Null
180	CLR: undo T4 LSN 130	150



Transaction table at time of checkpoint

Transaction ID	lastLSN	Status
T1	70	Running
T2	60	Running
T3	30	Running
T4	50	Running

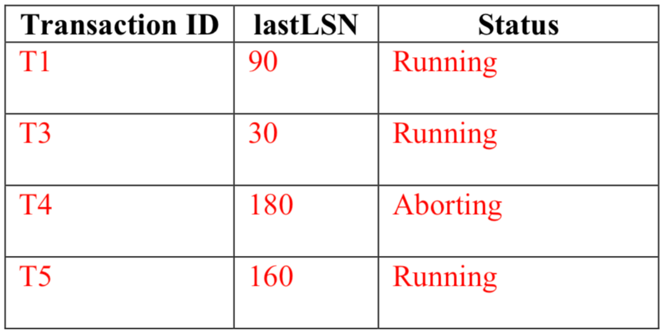


Dirty page table at time of checkpoint

Page ID	recLSN
P5	50
P1	40

(a) [**2 points**] At the end of the Analysis phase, what transactions will be in the transaction table, and with what lastLSN and Status values?

Solution

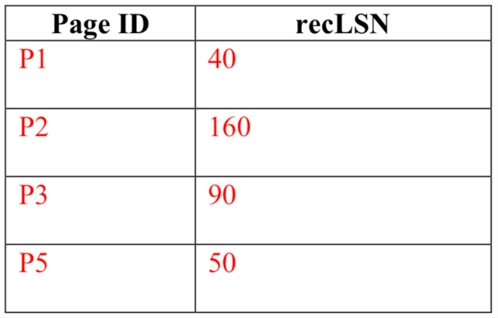


Transaction ID	lastLSN	Status
T1	90	Running
T3	30	Running
T4	180	Aborting
T5	160	Running

We also accept the answer by moving the status of transactions in the above table from "Running" to "Aborting".

(b) [**2 points**] At the end of the Analysis phase, what pages will be in the dirty page table, and with what recLSN values?

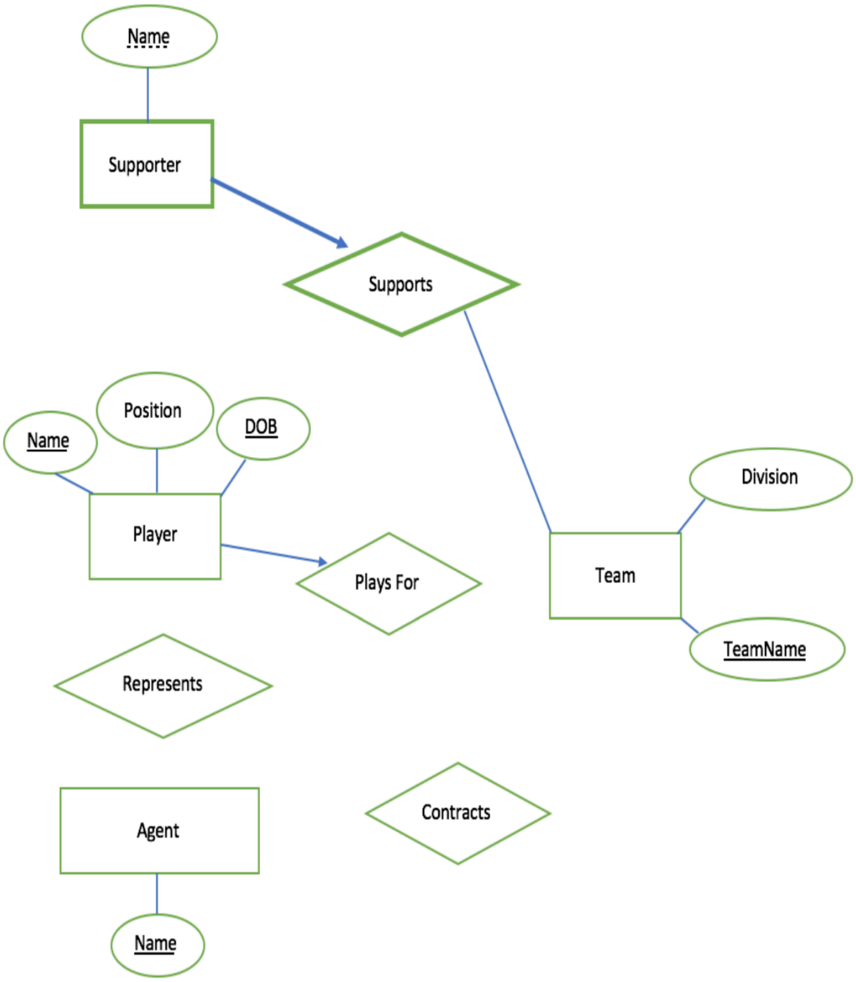
Solution



Page ID	recLSN
P1	40
P2	160
P3	90
P5	50

IX ER Modeling and Functional Dependencies [20 points]

1. **[4 points] ER Modeling**
   In this question, we will choose to model elements of a baseball league using ER-Diagrams. We have 4 entities: supporters, teams, players, contract and agents. Refer to the diagram below for the following question.

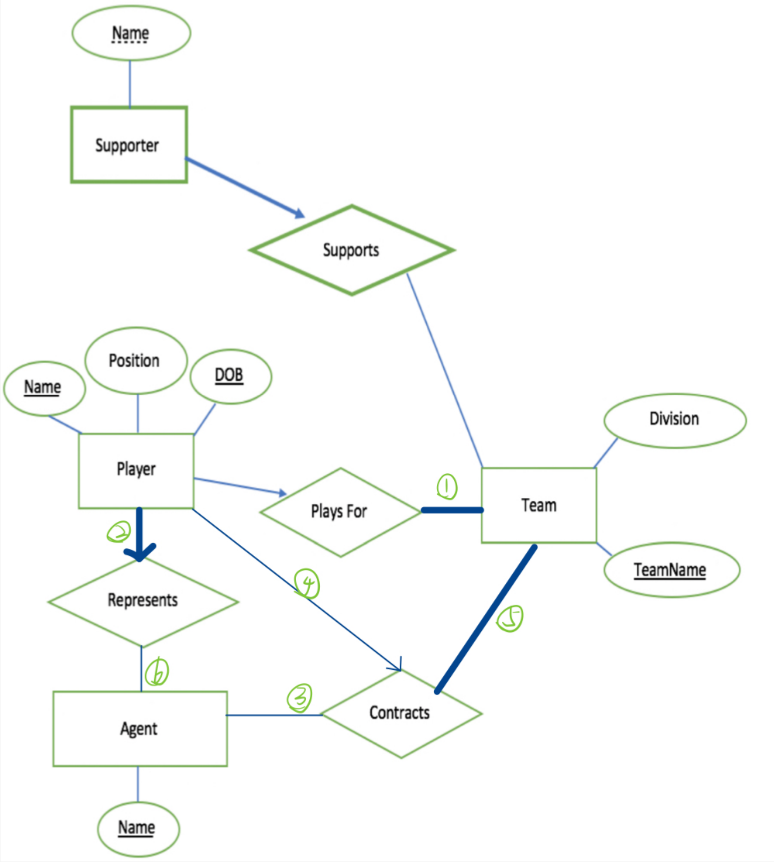


   (a) There are some elements of the ER-diagram are missing, fill them in so that the ER-diagram satisfies the following constraints.

   - Each team must have at least one players on its roster.
   - Each player must have exactly one agent to represent him/her.
   - Each agent can sign as many contracts as he or she wants to, or none at all.
   - Each player can have at most one contract.
   - A team has at least one contract.
   - Relation **Represents** involves entities **Player** and **Agent**; relation **Contracts** involves entities **Player**, **Agent** and **Team**; relation **Plays For** involves entities **Team** and **Player**

   Hint: you should add 6 edges in total.

   Solution



(b) As **Supporter** is a weak entity that must have unique and total participation in the **Supports** relation, its primary key is determined by which attribute(s) in the ER-diagram?
A. Name
B. TeamName
C. Name, TeamName
D. None of them

Solution
C

2. [**4 points**] **FD Properties**
Select all the FD's that follow from Armstrong's Axiom. Select all true statement(s):
A. if X → Y and Z → W, then XZ → YW
B. if XZ → YZ, then X → Y
C. if X → YZ, then X → Y and X → Z
D. if X → Y and WY → Z, then WX → Z
E. if X → Y and W → Y, then X → W

Solution
A, C, D

3. [**4 points**] **FD Example**
We have a relation R(A, B, C, D, E). We are told that the set of functional dependencies is F = E → BC, A → B, C → D, AD → C.

(a) What are A+, C+ and E+

(b) Is the attribute set ACE the superkey for relation R?

(c) Is relation R already in Boyce-Codd Normal Form (BCNF)?

Solution

(a) A+: AB, C+: CD, E+: BCDE.

(b) Yes. ACD+: ABCDE.

(c) No. None of the the FDs are a superkey of R and none of them are trivial FDs.

4. [**4 points**] **Normalization and Decomposition**
Decompose ABCDEFGH into Boyce-Codd Normal Form (BCNF) in the order of the following FDs: G → H; EF → B; EA → C; FH → D.

Solution
At step 1, we get ABCDEFG and GH.
At step 2, we get ACDEFG and EFB and GH.
At step 3, we get ADEFG and EAC and EFB and GH.

X Parallel Query Processing [20 points]

1. **[4 points] Partition**
Assume that we have 5 machines and a 1000 page **students(sid, name, gpa)** table. Initially, all of the pages start on one machine. Assume pages are 1KB.

   (a) How much network cost does it take to round-robin partition the table?

   (b) How many IOs will it take to execute the following query:

```
SELECT *
FROM students
WHERE name = 'John'
```

   (c) Suppose that instead of round robin partitioning the table, we hash partitioned it on the name column instead, How many IOs would the query from (b) take? (Assume that all records with value 'John' can fit on one machine.)

   (d) Assume that an IO takes 1ms and the network cost is negligible. How long will the query in part 2 take if the data is round-robin partitioned and if the data is hash partitioned on the name column? (For simplicity, assume that each machine spends the same amount of time on scanning.)

Solution

(a) 800 KB.
In round robin partitioning the data is distributed completely evenly. This means that the machine the data starts on will be assigned 1/5 of the pages. This means that 4/5 of the the pages will need to move to different machines, so the answer is: 4/5 * 1000 * 1 = 800 KB.

(b) 1000 IOs.
When the data is round robin partitioned we have no idea what machine(s) the records we need will be on. This means we will have to do full scans on all of the machines so we will have to do a total of 1000 IOs.

(c) 200 IOs.
When the data is hash partitioned on the name column, we know exactly what machine to go to for this query (we can calculate the hash value of 'John' and find out what machine is assigned that hash value). This means we will only have to a full scan over 1 machine for a total of 200 IOs. Of course, this assumes all records with value 'John' can fit on one machine.

(d) 200ms.
Under both partitioning schemes it will take 200ms. Each machine will take the same amount of time to do a full scan, and all machines can be run at the same time so for runtime it doesn't matter how many machines we need to query.

2. **[4 points] Parallel Algorithm**
Assume that we have a 100 page table R that we will join with a 400 page table S. All the pages start on machine 1, and there are 4 machines in total with 30 buffer pages each.

   (a) How many passes are needed to do a parallel unoptimized SMJ on the data? For this question a pass is defined as a full pass over either table (so if we have to do 1 pass over R and 1 pass over S it is 2 total passes).

   (b) How many passes are needed to a parallel Grace Hash Join on the data?

   (c) We want to calculate the max in parallel using hierarchical aggregation. What value should each machine calculate and what should the coordinator do with those values?

Solution

(a) 7 passes.
2 passes to partition the data (1 for each table). Each machine will then have 25 pages for R and 100 pages for S. R can be sorted in 1 pass but S requires 2 passes. Then it will take 2 passes to merge the tables together (1 for each table). This gives us a total of 2 (partition) + 1 (sort R) + 2 (sort S) + 2 (merge relations) = 7 passes.

(b) 4 passes.
We will need 2 passes to partition the data across the four machines and each machine will have 25 pages for R and 100 for S. We don't need any partitioning passes because R fits in memory on every machine, so we only need to do build and probe, which will take 2 passes (1 for each table). This gives us a total of 4 passes.

(c) Each machine should calculate the max and the coordinator will take the max of those maxes.

XI Distributed Transactions and NoSQL [20 points]

1. **[4 points] 2-Phase Commit**
   True or False:

   (a) If we are recovering and see a PREPARE record, it means we must have sent out a YES vote.

   (b) If the coordinator is recovering and sees a COMMIT record, we can locally commit and end the process there.

   (c) Suppose a participant receives a prepare message for a transaction and replies VOTE-YES. Suppose that they were running a wound-wait deadlock avoidance policy, and a transaction comes in with higher priority. The new transaction will abort the prepared transaction.

Solution

(a) False.
We could have crashed between logging PREPARE and sending VOTE-YES.

(b) False.
There is no guarantee that all participants received a COMMIT record, so we must rerun phase 2 first.

(c) False.
Transactions that are prepared must be ready to commit if the coordinator tells them to, so they cannot be aborted by anyone other than the coordinator.

2. **[4 points] OLTP and OLAP**
   Are the following workloads better characterized as Online Transaction Processing (OLTP) or Online Analytical Processing (OLAP)?

   (a) Placing orders and buying items in an online marketplace

   (b) Analyzing trends in purchasing habits in an online marketplace

Solution

(a) OLTP.
OLTP workloads involve high numbers of transactions executed by many different users. Queries in these workloads involve simple lookups more often than complex joins. In this case, when a user buys an item, the site might perform actions like looking up the item and updating its quantity, recording a new purchase, etc.

(b) OLAP.
OLAP workloads involve read-only queries and typically include lots of joins and aggregations. Often, workloads executed for analysis and decision making are OLAP workloads.

3. **[4 points] NoSQL**
   True or False:

   (a) As methods for scaling dataset, partitioning is effective for write-heavy workloads, and replicaction is effective for read-heavy workloads.

   (b) Partitioning and replication are often used together in real systems to leverage the performance benefits and to make the system more fault-tolerant.

   (c) The consistency in the distributed systems is the same concept with the consistency found in ACID (Transactions).

   (d) JSON and XML are referred as semi-structured data formats, and can express complex data structures, which may be arranged into nested structures.

   (e) JSON is self-describing, and it is designed for efficient storage and retrieval from disk.

Solution

(a) True.

(b) True.

(c) False.

(d) True.

(e) False.

XII MapReduce and Spark [20 points]

1. **[4 points] Basics**
True or False:

   (a) In MapReduce, the Map phase applies a function in parallel to every element of a set of data, and the Reduce phase combines the results of the Map phase into the desired data output.

   (b) In Distributed File System (DFS), a file system in which large files are partitioned into smaller files, and then distributed and replicated several times on different nodes for fault tolerance.

   (c) Both the Map and Reduce phases can be split among dfferent workers on different machines, with workers performing independent tasks in parallel.

   (d) Since the Reduce phase must wait until the entire Map phase completes, we must restart the entire Map phase if one map task fails.

   (e) Resilient Distributed Datasets (RDDs) are not written to disk in intermediate steps but are rather stored in main memory.

   (f) Similar with MapReduce, Spark needs to write intermediate results to disk.

Solution

(a) True.

(b) True.

(c) True.

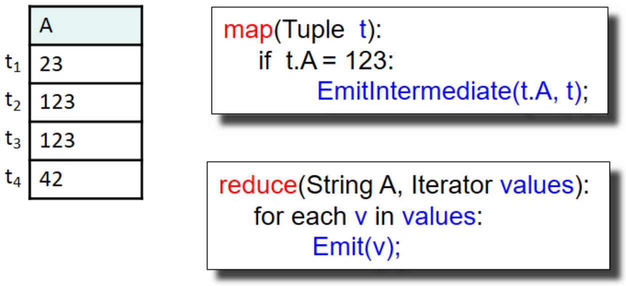
(d) False.

(e) True.

(f) False.

2. **[4 points] MapReduce**

   (a) Fig. 1 shows a relation $R(A)$ with the MapReduce implementation of the Selection operator $\sigma_{A=123}(R)$.



	A
t_1	23
t_2	123
t_3	123
t_4	42

```
map(Tuple  t):
    if  t.A = 123:
            EmitIntermediate(t.A, t);
```

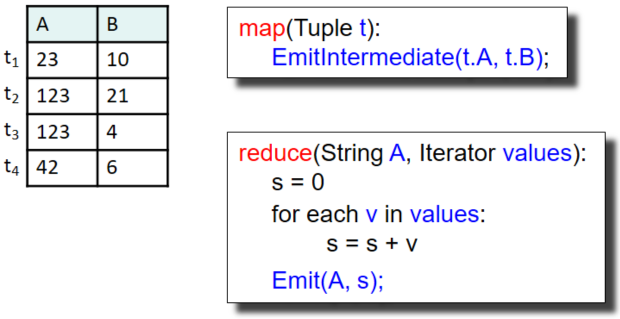
```
reduce(String A, Iterator values):
    for each v in values:
            Emit(v);
```

Figure 1: Selection

   (1) What is the output of the Map function?

   (2) What is the output of the Reduce function?

   (b) Fig. 2 shows a relation $R(A, B)$ with the MapReduce implementation of the Group-By operator $\gamma_{A=123,\text{SUM}(B)}(R)$.



	A	B
t_1	23	10
t_2	123	21
t_3	123	4
t_4	42	6

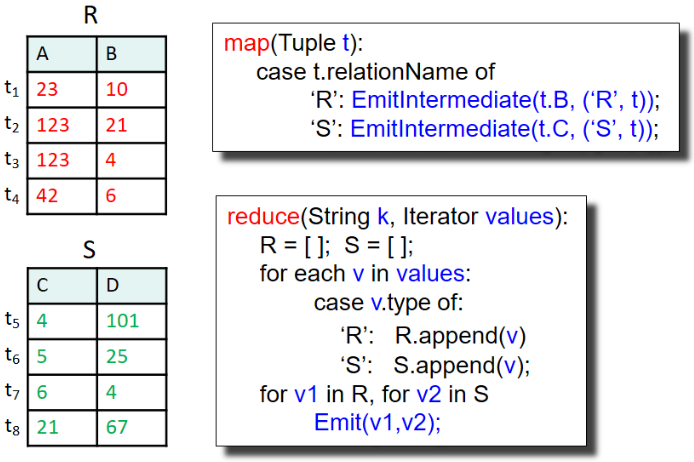
```
map(Tuple t):
    EmitIntermediate(t.A, t.B);
```

```
reduce(String A, Iterator values):
    s = 0
    for each v in values:
        s = s + v
    Emit(A, s);
```

Figure 2: Group-By

(1) What is the output of the Map function?
(2) What is the output of the Reduce function?

(c) Fig. 3 shows two relations $R(A, B)$ and $S(C, D)$ with the MapReduce implementation of the Hash-Join operator $R(A, B) \bowtie_{B=C} S(C, D)$.



R

	A	B
t_1	23	10
t_2	123	21
t_3	123	4
t_4	42	6

S

	C	D
t_5	4	101
t_6	5	25
t_7	6	4
t_8	21	67

```
map(Tuple t):
    case t.relationName of
        'R': EmitIntermediate(t.B, ('R', t));
        'S': EmitIntermediate(t.C, ('S', t));
```

```
reduce(String k, Iterator values):
    R = [ ];  S = [ ];
    for each v in values:
        case v.type of:
            'R':   R.append(v)
            'S':   S.append(v);
    for v1 in R, for v2 in S
        Emit(v1,v2);
```

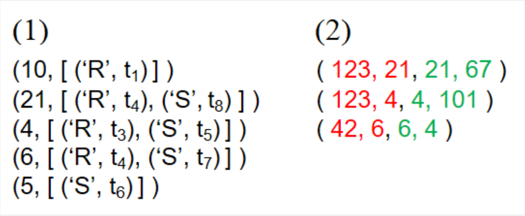
Figure 3: Hash-Join

(1) What is the output of the Map function?
(2) What is the output of the Reduce function?

Solution

(a) (1) $(123, [t_2, t_3])$
(2) (t_2, t_3).

(b) (1) $(23, [t_1])$, $(42, [t_4])$, $(123, [t_2, t_3])$
(2) $(23, 10)$, $(42, 6)$, $(123, 25)$.



(1)
(10, [('R', t_1)])
(21, [('R', t_4), ('S', t_8)])
(4, [('R', t_3), ('S', t_5)])
(6, [('R', t_4), ('S', t_7)])
(5, [('S', t_6)])

(2)
(123, 21, 21, 67)
(123, 4, 4, 101)
(42, 6, 6, 4)

(c)

XIII Data Mining and Machine Learning [20 points]

1. [4 points] **General Data Mining and Machine Learning.**
   Mark the boxes for all true statement(s):

   (a) In multi-dimensional data model, it contains one fact table and multiple dimension tables.

   > **Solution**
   > True.

   (b) The order of the basic KDD (Knowledge Discovery in Database) process is Data Selection, Data Cleaning, Evaluation, Data Mining & ML.

   > **Solution**
   > False: Data Mining comes before Evaluation.

   (c) In supervised machine learning labeled data is used to train a model.

   > **Solution**
   > True: The labeled data is the "supervision" in the name.

   (d) Classification models would be a good candidate when trying to predict the total amount of time a user spends browsing a page.

   > **Solution**
   > False: Classification produces a discrete label (like "True/False"); to get a numeric value like an amount of time you might want to use regression.

   (e) The ultimate goal in machine learning is to find a model that best fits the training data.

   > **Solution**
   > False: Fitting the training data is a means to the end goal of predicting unknown information. As we learned, you can overfit the training data if you make that your ultimate goal.

   (f) The k-means algorithm is guaranteed to converge to the global optimum.

   > **Solution**
   > False: k-means can produce a sub-optimal answer, and it sensitive to the initial position of the centers; it does compute what's called a "local optimum".

   (g) The bag-of-words model encodes text as a vector.

   > **Solution**
   > True.

   (h) A common form of feature engineering on continuous data is one-hot-encoding.

   > **Solution**
   > False: One-hot encoding is used for capturing the presence of boolean features.

2. [8 points] **K-Means.**

   (a) [**2 points**] Which of the following can act as possible termination conditions in k-means? (you may mark zero (ϕ), one or more than one of the choices.)
   A. For a fixed number of iterations.
   B. Assignment of observations to clusters does not change between iterations. Except for cases with a bad local minimum.
   C. Centroids do not change between successive iterations.
   D. Terminate when the objective value falls below a threshold.

Solution
A, B, C, D.
All four conditions can be used as possible termination condition in K-Means clustering:
This condition limits the runtime of the clustering algorithm, but in some cases the quality of the clustering will be poor because of an insufficient number of iterations. Except for cases with a bad local minimum, this produces a good clustering, but runtimes may be unacceptably long. This also ensures that the algorithm has converged at the minima. Terminate when the objective value falls below a threshold. This criterion ensures that the clustering is of a desired quality after termination. Practically, it's a good practice to combine it with a bound on the number of iterations to guarantee termination.

(b) **[2 points]** In which of the following cases will k-means clustering fail to give good results? (you may mark zero (ϕ), one or more than one of the choices.)
A. Data points with outliers.
B. Data points with different densities.
C. Data points with round shapes.
D. Data points with non-convex shapes.

Solution
A, B, D.
K-Means clustering algorithm fails to give good results when the data contains outliers, the density spread of data points across the data space is different and the data points follow non-convex shapes.

(c) **[2 points]** The following is a set of one-dimensional points: $\{-4, 0, 1, 2, 3, 5, 7, 10, 13, 22\}$, perform two iterations of k-means on these points using the two initial centroids 1 and 7.

1) What are the two new centroids after the first iteration?
2) What are the two new centroids after the second iteration?

Solution
first iteration: 2/5; 57/5
second iteration: 7/6; 13

3. **[4 points] Linear Regression**
Given n independent and identically distributed (i.i.d.) training samples $\{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$, with the i-th sample $x_i, y_i \in \mathbb{R}$, $i = 1, 2, ..., n$. Consider the linear model:

$$y = x\theta + \epsilon \quad \epsilon \sim \mathcal{N}(0,\, \sigma^2), \tag{1}$$

where $\mathcal{N}(0,\, \sigma^2)$ denotes the Guassian distribution with mean 0 and variance σ^2. Assume that the training data has been centralized, such that the intercept can be ignored in the above linear model. Obviously, the probability density function of y conditioned on x and θ follows the Gaussian distribution $\mathcal{N}(x\theta, \sigma)$, which is defined by

$$p(y|x, \theta) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{1}{2\sigma^2}(y - x\theta)^2). \tag{2}$$

The model parameter θ can be estimated based on Maximum Likelihood Estimation (MLE), which aims to maximize the likelihood function:

$$\begin{aligned} L(\theta) &= \prod_{i=1}^{n} P(x_i, y_i|\theta) \\ &= \prod_{i=1}^{n} P(y_i|\theta, x_i)P(x_i) \\ &= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{(y_i - x_i\theta)^2}{2\sigma^2})P(x_i) \end{aligned} \tag{3}$$

(a) Show the log-likelihood function $\log L(\theta)$. (The base of the log function is e.)

(b) Using MLE to calculate the closed-form solution of θ.

Solution

(a) According to the definition of the likelihood function, we have

$$\begin{aligned}\log L(\theta) &= \sum_{i=1}^{n} -\frac{1}{2}\log 2\pi - \log\sigma - \frac{(y_i - x_i\theta)^2}{2\sigma} + \log P(x_i) \\ &= -\frac{1}{2\sigma}\sum_{i=1}^{n}(y_i - x_i\theta)^2 + C, \qquad (4)\end{aligned}$$

where C denotes a constant, and it is irrelevant to θ.

(b) Based on (a), we have

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \log L(\theta) = \operatorname{argmin} \sum_{i=1}^{n}(y_i - x_i\theta)^2. \qquad (5)$$

By seting the derivative of above objective function w.r.t. θ equal to 0, we have

$$\theta = \frac{\sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} x_i^2}. \qquad (6)$$