CS 186 Fall 2020

Sample Exam.

Midterm 2

INSTRUCTIONS

This is your exam. Complete it either at exam.cs61a.org or, if that doesn't work, by emailing course staff with your solutions before the exam deadline.

This exam is intended for the student with email address `<EMAILADDRESS>`. If this is not your email address, notify course staff immediately, as each exam is different. Do not distribute this exam PDF even after the exam ends, as some students may be taking the exam in a different time zone.

For questions with **circular bubbles**, you should select exactly *one* choice.

○ You must choose either this option

○ Or this one, but not both!

For questions with **square checkboxes**, you may select *multiple* choices.

☐ You could select this choice.

☐ You could select this one too!

You may start your exam now. Your exam is due at <DEADLINE> Pacific Time. Go to the next page to begin.

Preliminaries

1 CS 186 Fall 2020 Midterm 2 (Online)

`Do not share this exam until solutions are released.`

1.0.1 Contents:

- The midterm has 5 questions, each with multiple parts, and worth a total of 100 points.

1.0.2 Taking the exam:

- You have 110 minutes to complete the midterm.
- You may print this exam to work on it.
- For each question, submit only your final answer on examtool.
- For numerical answers, do not input any suffixes (i.e. if your answer is 5 I/Os, only input 5 and not 5 I/Os or 5 IOs)
- Make sure to save your answers in examtool at the end of the exam, although the website should autosave your answers as well.

1.0.3 Aids:

- You may use two pages (double sided) of handwritten notes as well as a calculator.
- You must work individually on this exam.

1.0.4 Grading Notes:

- All I/Os must be written as integers. There is no such thing as 1.02 I/Os – that is actually 2 I/Os.
- 1 KB = 1024 bytes. We will be using powers of 2, not powers of 10
- Unsimplified answers, like those left in log format, will receive a point penalty.

(a) What is your full name?

(b) What is your student ID number?

(c) Who do you believe will win the election?

1. (21 points) Iterators/Joins

Poll volunteer Jerry is trying to register eligible voters for the election. He has a list of all residents in his county who have registered (Table X) which takes up 25 pages and a list of eligible voters (Table Y) which takes up 30 pages.

For all parts of this question, do not include the units in your answer. For example, if the answer is 5 I/Os, just write "5" or if the answer is 4 passes, just write "4."

(a) (2 pt) Jerry wishes to join table X (25 pages) and table Y (30 pages) on social security number. His computer has 10 buffer pages. How many IOs will a Page Nested Loops Join of these two tables cost?

(b) (1 pt) Poll volunteer Saurav tells Jerry that Block Nested Loops Join is better than Page Nested Loops Join as it will always reduce IO cost. Jerry knows this is true only for a certain range of buffer sizes B.

With our fixed table sizes of 25 and 30 pages, give a range for B in which the IOs for BNLJ is less than the IOs for PNLJ. You should express your answer as "L<=B<=U" (no spaces) where L and U correspond to the lower and upper bounds for which the statement is true. You may use INF to express infinity in your solution if needed.

(c) (9 points)

Jerry wishes for his output to be ordered, and he knows he can use Sort Merge Join refinements and optimizations to potentially reduce IO counts.

For all of the sort merge join questions, apply any SMJ optimizations/refinements possible. You may also assume that there are no duplicates in either of the two tables.

i. (2 pt) How many IOs will Sort Merge Join of X (25 pages) and Y (30 pages) with 10 buffer pages take? You should not assume that these tables are already sorted.

ii. (1 pt) Express your answer to the previous question in terms of M and N which represents the number of IOs needed to sort Table X and Table Y respectively. Please do not insert spaces in between (example answer: "M+N+100").

iii. (2 pt) Just before he starts, Jerry's computer dies. He turns to his phone to do the join.

How many IOs will Sort Merge Join of X (25 pages) and Y (30 pages) with 3 buffer pages take? You should not assume that these tables are already sorted.

iv. (1 pt) Express your answer to the previous question in terms of M and N which represents the number of IOs needed to sort Table X and Table Y respectively. Please do not insert spaces in between (example answer: "M+N+100").

v. (2 pt) Soon after, Jerry gets access to a large supercomputer!

How many IOs will Sort Merge Join of X (25 pages) and Y (30 pages) with 100 buffer pages take? You should not assume that these tables are already sorted.

vi. (1 pt) Express your answer to the previous question in terms of M and N which represents the number of IOs needed to sort Table X and Table Y respectively. Please do not insert spaces in between (example answer: "M+N+100").

(d) Poll manager Alvin finds that Jerry is doing such a good job that he gives him the information for all adjacent counties as well. Jerry feels that Sort Merge Join might be too slow, so he decides to execute Grace Hash Join instead.

Now, Table X has 90 pages and Table Y has 100 pages. Assume we have 11 buffers. You may assume a perfectly uniform hash function in all passes of Grace Hash Join and no key skew.

i. **(1 pt)** After the first pass of Grace Hash Join, how many partitions of X have been created?

ii. **(1 pt)** After the first pass of Grace Hash Join, how many pages are in each partition of X?

iii. **(1 pt)** How many IOs will this partitioning of X take?

iv. **(1 pt)** After the first pass of Grace Hash Join, how many partitions of Y have been created?

v. **(1 pt)** After the first pass of Grace Hash Join, how many pages are in each partition of Y?

vi. **(1 pt)** How many IOs will this partitioning of Y take?

vii. **(1 pt)** Is recursive partitioning necessary? If yes, which table will be repartitioned?

viii. **(2 pt)** After executing the rest of the Grace Hash Join (including recursive partitioning, if necessary) how many IOs in total will Grace Hash Join take?

2. (21 points) Query Optimization

For this question, let's look at tables R, S, and T with the following properties. Assume that each column's values are uniformly distributed, and that all of the column distributions are independent of each other.

Schema	Pages	Indexes	Table Stats
CREATE TABLE **R**(a INTEGER, b FLOAT, c INTEGER)	50 pages with 100 records each	**R.b:** Alt 1 index with h = 3 and 50 leaf pages and **R.c:** Alt 3, clustered index with h = 2 and 20 leaf pages	**R.b:** min = 0, max = 20, 75 unique values and **R.c:** min = 10, max = 50, 20 unique values
CREATE TABLE **S**(a INTEGER, b FLOAT, c INTEGER)	100 pages with 100 records each	**S.b:** Alt 2, clustered index with h = 2 and 10 leaf pages	**S.b:** min = 1, max = 50, 50 unique values
CREATE TABLE **T**(a INTEGER, b FLOAT, c INTEGER)	25 pages with 100 records each	**T.c:** Alt 2, unclustered index with h = 2 and 10 leaf pages	**T.c:** min = 1, max = 100, 100 unique values

(a) (4 points) Selectivity Estimation

Estimate the **number of pages of output** each of the following queries will produce.

i. (1 pt) `SELECT * FROM R WHERE R.b <= 10`

ii. (1 pt) `SELECT * FROM T WHERE T.c <= 20`

iii. (2 pt) `SELECT R.a, S.b, S.c  FROM R INNER JOIN S on R.a = S.a WHERE R.b <= 10`

iv. (0 pt) Feel free to show your work for the previous question below for partial credit. You may leave this blank if you'd like.

(b) (5 points) System R Query Optimizer - Pass 1

For the next sections, we will optimize the following query using the System R (aka Selinger) query optimizer.

```
SELECT R.a, S.b, T.c
    FROM R INNER JOIN S ON R.a = S.a
           INNER JOIN T ON R.b = T.b
    WHERE R.b <= 10 AND T.c <= 20
    GROUP BY S.b
```

i. (3 pt) What is the estimated I/O cost of performing an index scan on T using the index on T.c?

ii. (0 pt) Feel free to show your work for the previous question below for partial credit. You may leave this blank if you'd like.

Option	Access Plan	I/O Cost

iii. **(2 pt)** Ignore your answer for the previous problem and **assume the cost of an index scan on T.c is 30 I/Os** (as indicated in the table below). Assume that we have the following I/O costs for accessing single tables:

Option	Access Plan	I/O Cost
a	R: Full scan	50 I/Os
b	R: Index scan on R.b	28 I/Os
c	R: Index scan on R.c	72 I/Os
d	S: Full scan	100 I/Os
e	S: Index scan on S.b	102 I/Os
f	T: Full scan	25 I/Os
g	T: Index scan on T.c	30 I/Os

Which of the following plans will be kept at the end of Pass 1?

☐ Option a

☐ Option b

☐ Option c

☐ Option d

☐ Option e

☐ Option f

☐ Option g

☐ None of the above

(c) (12 points) System R Query Optimizer - Pass 2/3

Let's now look at passes 2 and 3 of the System R Query Optimizer. Assume that:

- We have **B = 5** buffer pages.
- Your earlier answer to the number of pages of R after applying the `R.b <= 10` predicate is **10 pages**
- Your answer to the number of pages of T after applying the `T.c <= 20` predicate is **20 pages**

For your convenience, here is a copy of the query we are optimizing:

```
SELECT R.a, S.b, T.c
    FROM R INNER JOIN S ON R.a = S.a
           INNER JOIN T ON R.b = T.b
    WHERE R.b <= 10 AND T.c <= 20
    GROUP BY S.b
```

i. (3 pt) What is the estimated I/O cost of R BNLJ S? Use the full scan on R and the full scan on S as the single table access plans for R and S (as given in the table for Pass 1).

ii. (0 pt) Feel free to show your work for the previous question below for partial credit. You may leave this blank if you'd like.

iii. (1 pt) Will R BNLJ S produce an interesting order?

◯ Yes

◯ No

iv. (3 pt) Jerry is trying to estimate the I/O cost for R SMJ T. He thinks it's 300 I/Os (the cost of sorting R) + 150 I/Os (the cost of sorting T) + 75 I/Os (cost of merging) = 525 I/Os.

However, Jennifer disagrees and thinks we can compute R SMJ T in fewer I/Os.

List 2 optimizations we can make that will lower the I/O cost of R SMJ T.

v. (1 pt) Will R SMJ T produce an interesting order?

◯ Yes

◯ No

vi. (2 pt) Assume that we have the following I/O costs for joining 2 tables together:

Option	Access Plan	Sorted on	I/O Cost
a	$R \bowtie S$	N/A	800 I/Os
b	$S \bowtie R$	S.b	750 I/Os
c	$S \bowtie T$	N/A	700 I/Os
d	$T \bowtie S$	S.b	650 I/Os
e	$R \bowtie T$	N/A	150 I/Os
f	$R \bowtie T$	R.b	160 I/Os
g	$T \bowtie R$	T.c	170 I/Os

Using just this information, along with the query, which of the following plans will be kept at the end of Pass 2?

☐ Option a

☐ Option b

- ☐ Option c
- ☐ Option d
- ☐ Option e
- ☐ Option f
- ☐ Option g
- ☐ None of the above

vii. (2 pt) Assume that the access plans retained at the end of Pass 2 were S $\bowtie$ R, S $\bowtie$ T, and R $\bowtie$ T. Which of the following access plans will be considered in Pass 3 of the System R query optimizer?

- ☐ (R $\bowtie$ S) $\bowtie$ T
- ☐ (S $\bowtie$ R) $\bowtie$ T
- ☐ (R $\bowtie$ T) $\bowtie$ S
- ☐ T $\bowtie$ (S $\bowtie$ R)
- ☐ R $\bowtie$ (S $\bowtie$ T)
- ☐ S $\bowtie$ (R $\bowtie$ T)
- ☐ None of the above

3. (20 points) Concurrency or Procurrency

(a) Tripping (up) on ACID

For the following three questions, select **ALL** ACID properties that could be violated by the database, if any.

i. (2 pt) You have a special database that currently has one table `students` with a primary key `sid`. You run three transactions and the resulting `students` table (after the transactions commit) consists of every row in the previous table duplicated three times. You decide you don't like this new table and restart your computer. Luckily, when you reboot the database, the `students` table has returned to the way it was (only one copy of each row).

☐ Atomicity

☐ Consistency

☐ Isolation

☐ Durability

☐ None

ii. (2 pt) You have a special database that commits all active transactions prematurely and flushes their updates as soon as it detects any deadlock. The database uses strict 2PL.

☐ Atomicity

☐ Consistency

☐ Isolation

☐ Durability

☐ None

iii. (2 pt) Instead of conflict serializability, your database only accepts schedules that are write-write equivalent to some serial schedule. Two schedules are write-write equivalent if they involve the same actions of the transactions in the same order, and every pair of writes across two different transactions on the same resource is ordered the same way.

☐ Atomicity

☐ Consistency

☐ Isolation

☐ Durability

☐ None

(b) Don't Bank on It

Tara is about to participate in a 3 way transfer at her bank. She is supposed to receive $50 each from her friends Utne and Valdo. Though she is only supposed to receive $100, she knows that the bank transactions are **not atomic** and she knows the ordering of operations ahead of time. Therefore, she knows that she can possibly make a profit here. The transfer is done in **one transaction** as follows (T is Tara's bank account, U is Utne's, and V is Valdo's). Keep in mind that R stands for read and W stands for write.

Timestamp:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
T1	R(U)	R(V)	U=U-100	W(U)	R(T)	V=V-100	T=T+200	W(V)	T=T-50	V=V+50	W(V)	W(T)	U=U+50	W(U)	T=T-50	W(T)

i. (2 pt) What is the maximum profit Tara could make? (don't include the dollar sign)

ii. (2 pt) Select all possible values that Utne can see in his bank account after the transfer is done.

☐ U

☐ U - 50

☐ U - 100

☐ U - 150

☐ U - 200

☐ None of the above

iii. (1 pt) Assume that the bank upgrades their infrastructure and now separates the transfer into three separate transactions. This time we have a four way transfer between four bank accounts (T, U, V, and A). Once again, R stands for read and W stands for write. The schedule is as follows:

	1	2	3	4	5	6	7	8	9	10
T1	R(T)				R(V)			R(A)		W(U)
T2			W(V)				R(T)		R(A)	
T3		R(U)		W(A)		W(T)				

Is this schedule conflict serializable?

◯ Yes

◯ No

iv. (2 pt) If you marked no in the previous answer, please provide the smallest set of timestamps of the operations which, once removed, will make this schedule conflict serializable.

If you marked yes, write 0.

For example, if you believe the minimum number of operations you need to remove to make the schedule conflict serializable is 3 and removing T1's R(T) and W(U) and T3's W(A) will satisfy this constraint, provide as your answer 1,4,10.

(c) Pop, Lock, and Drop It (with a heavy emphasis on lock it)

Refer to the following schedule of transactions for the next 3 questions. These transactions use shared and exclusive locks on the resources A, B, C, and D. **Assume our lock manager's wait queue policy is always first-in-first-out**.

	1	2	3	4	5	6	7	8	9	10
T1	S(C)									X(A)
T2		S(B)						X(B)		
T3			S(A)		X(C)				X(D)	
T4				S(A)		S(C)	X(C)			

i. (2 pt) Directly after timestep 6 but before timestep 7, what transactions are in the wait queue for resource C?

☐ T1

☐ T2

☐ T3

☐ T4

☐ None of the above

ii. (1 pt) You are contemplating between using wound-wait and wait-die for deadlock avoidance for this schedule. You would like to maximize the number of transactions aborted. Both wound-wait and wait-die are ordered on lock acquisition. This means that a transaction held the lock first gets priority. Assume aborted transactions don't restart until long after timestamp 10. Also assume that waiting transactions do not try to acquire other locks. Between wound-wait and wait-die, which strategy will result in the most number of aborted transactions?

◯ Wait-die

◯ Wound-wait

◯ They both result in the same number.

iii. (2 pt) Which transactions are aborted in the maximum case?

For example, if you answered wait-die previously and if you believe wait-die aborts 3 transactions; T1, T2, and T4; your answer would be T1,T2,T4 or any permutation thereof. If you answered 'They both would result in the same number,' provide the transactions that would be aborted by Wait-die.

(d) Project SIX

In this subpart, we will explore 2 potential transactions and concurrency design choices.

i. **(2 pt)** In Project 4, upgrading a lock has priority over other queued requests. Your friend believes this is unfair. She proposes that the currently held lock is released and the request for the upgraded lock is added to the queue. To put it more concretely, let's assume T1 holds an S lock on resource A and T2 is waiting for an X lock on resource A. If T1 then requests an upgrade from S to X for the held lock, then, in normal circumstances, T1's upgrade would have priority and T1 would immediately be granted the X lock on A. However, your friend proposes that the lock is instead released, the new request gets added to the wait queue, and T2 receives their X lock on A. Your friend believes this will promote fairness across transaction requests so transactions won't get stalled for too long. Why might this not be a good idea? Answer in 1-2 sentences.

ii. **(2 pt)** We know from lecture that a schedule is conflict serializable if and only if its dependency graph is acyclic. Your friend suggests the following process to identify the conflict equivalent serial schedule for a schedule with an **acyclic** dependency graph: find a node with only outgoing edges, then perform depth-first search on it. He states that this will always return one such conflict equivalent serial schedule. Is your friend right? If yes, explain why. If not, propose another way of finding a conflict equivalent serial schedule. Answer in 1-2 sentences.

4. (14 points) Logging and Recovery

(a) (4 points) True or False

Choose True or False for each of the statements below.

i. (1 pt) Undo logging relies on flushing changes made by each transaction to the disk before it commits.

◯ True

◯ False

ii. (1 pt) Redo logging relies on flushing changes made by each transaction to the disk before it commits.

◯ True

◯ False

iii. (1 pt) For redo logging, we only need to store the new value and not the previous value in each update record in the log.

◯ True

◯ False

iv. (1 pt) Write-ahead logging means updates to in-memory pages are written to the disk before the corresponding log record is written to the disk.

◯ True

◯ False

(b) (10 points) Recovery

Consider the recovery log below:

```
1:  <START T1>
2:  <T1 X 5>
3:  <START T2>
4:  <T1 Y 8>
5:  <T2 X 9>
6:  <START T3>
7:  <T3 Z 112>
8:  <COMMIT T1>
9: <T2 X 13>
10: <ABORT T2>
11: <T3 Y 17>
[DB CRASHED]
```

i. (3 pt) Under write-ahead logging, and assuming that the recovery manager is aware of the status of all transactions before it starts recovery (e.g., by storing a separate "transaction status table" that flushed to the disk each time when the log is flushed).

Under UNDO logging, what is the earliest line the recovery manager must read to ensure full recovery of the database? Write down the line number below. No explanation needed. For instance, answering N means the recovery manager must read from the end of the log up to and including line N for full recovery.

ii. (3 pt) With the same assumptions as above, under REDO logging, what is the latest line the recovery manager must read to ensure full recovery of the database? Write down the line number below. No explanation needed. For instance, answering N means the recovery manager must read from the beginning of the log up to and including line N for full recovery.

iii. (2 pt) Under UNDO logging, what will be the value of X after recovery? Write down the answer below. No explanation needed.

iv. (2 pt) Under REDO logging, what will be the value of X after recovery? Write down the answer below. No explanation needed.

5. (20 points) The Electoral College of Database Design

We have the following tables in our database:

`Candidates`(**id, name, homeStateID, partyID, age**)

`States`(**sid, name, population, prevWinnerID, prevMargin**)

`Counties`(**cid, sid, name, population, prevWinnerID, prevMargin**)

`Polls`(**pid, sid, cid, projectedMargin, projectedWinnerID**)

For the entire question, you may assume the following:

- External sorting will always take exactly 2 passes.
- The SMJ optimization is not used.
- External hashing will never involve recursive partitioning.
- For external hashing, the number of pages written will always equal the number of pages read.
- The size of each relation, **including joined relations**, is at least 100 times the number of buffer pages.
- There are no indexes on any column.
- The database uses the **System R** query optimizer.

(a) Joins

Hint: For the next 6 questions, assume selections and projections do not change the size of the relation and do not worry about the selectivity of the join(s).

i. (1 pt) Which of the following joins will the query optimizer use to execute **Query A**?

Query A

```
SELECT C.name, C.partyID, S.name, S.prevMargin
FROM Candidates AS C INNER JOIN States AS S
ON C.homeStateID != S.sid
WHERE S.population > 5000000 AND S.prevMargin < 5;
```

◯ Simple Nested Loop Join

◯ Page Nested Loop Join

◯ Block Nested Loop Join

◯ Index Nested Loop Join

◯ Sort Merge Join

◯ Grace Hash Join

ii. (1 pt) Explain your answer for **Query A** in 2-4 sentences.

iii. (1 pt) Which of the following joins will the query optimizer use to execute **Query B**?

Query B

```
SELECT S.sid, MAX(P.projectedMargin), MIN(P.projectedMargin)
FROM Polls AS P INNER JOIN States AS S
ON P.sid = S.sid
INNER JOIN Counties AS C
ON S.sid = C.sid
WHERE S.name LIKE 'A%' AND P.cid IS NOT NULL
GROUP BY S.sid;
```

◯ Simple Nested Loop Join

◯ Page Nested Loop Join

◯ Block Nested Loop Join

◯ Index Nested Loop Join

◯ Sort Merge Join

◯ Grace Hash Join

iv. (2 pt) Explain your answer for **Query B** in 2-6 sentences.

v. (1 pt) Which of the following joins will the query optimizer use to execute **Query C**?

Query C

```
SELECT S.sid, AVERAGE(C.population)
FROM States AS S INNER JOIN Counties AS C
ON S.sid = C.sid
GROUP BY S.sid
ORDER BY AVERAGE(C.population);
```

◯ Simple Nested Loop Join

◯ Page Nested Loop Join

◯ Block Nested Loop Join

◯ Index Nested Loop Join

◯ Sort Merge Join

◯ Grace Hash Join

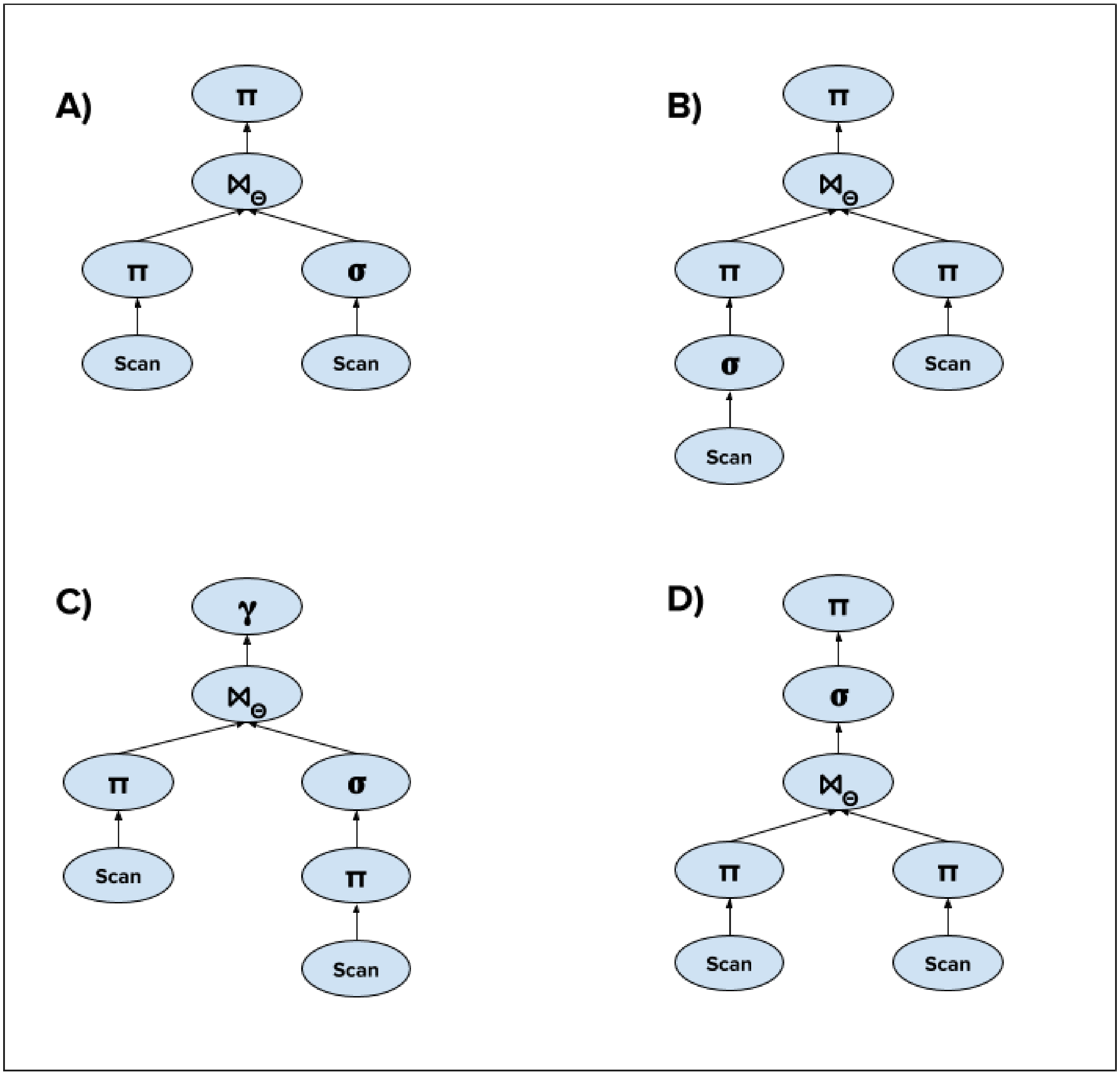
vi. (2 pt) Explain your answer for **Query C** in 2-4 sentences.

(b) Query Plan

i. (1 pt) Which of the following query plans will System R choose in order to execute **Query A**?

Query A

```
SELECT C.name, C.partyID, S.name, S.prevMargin
FROM Candidates AS C INNER JOIN States AS S
ON C.homeStateID != S.sid
WHERE S.population > 5000000 AND S.prevMargin < 5;
```

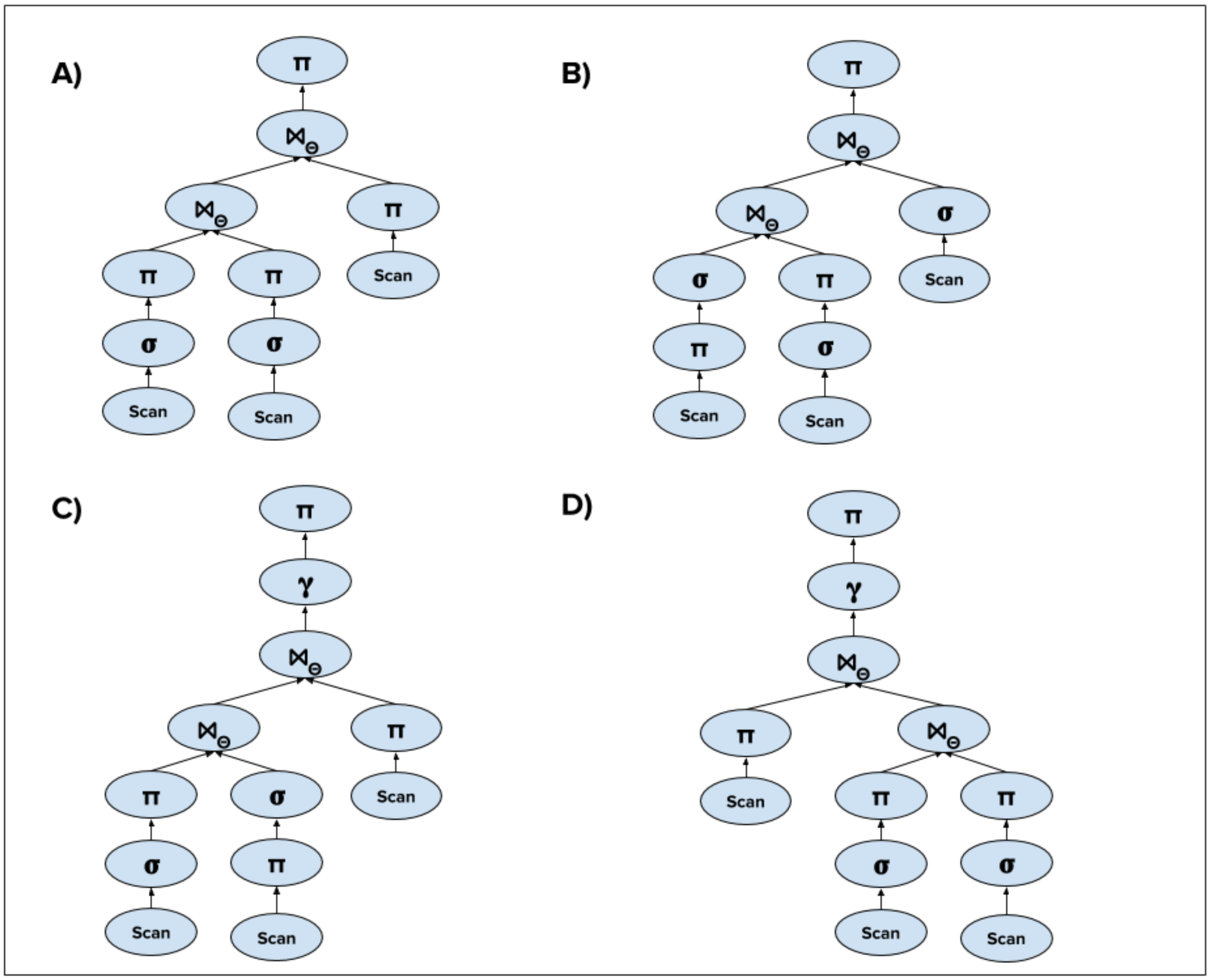


- ◯ A
- ◯ B
- ◯ C
- ◯ D
- ◯ None of the above

ii. (1 pt) Which of the following query plans will System R choose in order to execute **Query B**?

Query B

```
SELECT S.sid, MAX(P.projectedMargin), MIN(P.projectedMargin)
FROM Polls AS P INNER JOIN States AS S
ON P.sid = S.sid
INNER JOIN Counties AS C
ON S.sid = C.sid
WHERE S.name LIKE 'A%' AND P.cid IS NOT NULL
GROUP BY S.sid;
```

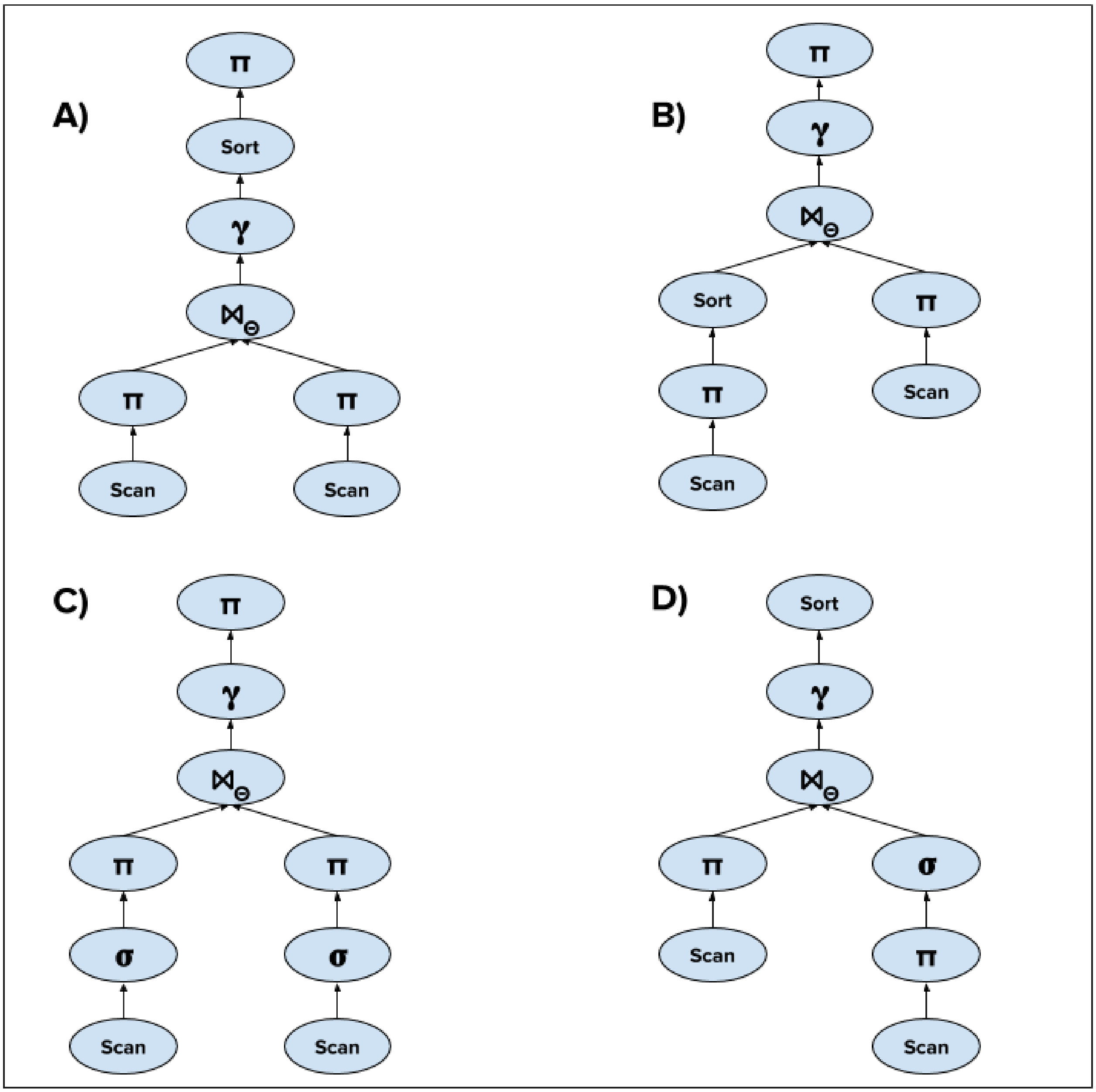


- ◯ A
- ◯ B
- ◯ C
- ◯ D
- ◯ None of the above

iii. (1 pt) Which of the following query plans will System R choose in order to execute **Query C**?

Query C

```
SELECT S.sid, AVERAGE(C.population)
FROM States AS S INNER JOIN Counties AS C
ON S.sid = C.sid
GROUP BY S.sid
ORDER BY AVERAGE(C.population);
```



- ◯ A
- ◯ B
- ◯ C
- ◯ D

◯ None of the above

(c) Concurrency

For the next 3 problems, consider the same 3 queries from above, which are copied below.

Query A

```
SELECT C.name, C.partyID, S.name, S.prevMargin
FROM Candidates AS C INNER JOIN States AS S
ON C.homeStateID != S.sid
WHERE S.population > 5000000 AND S.prevMargin < 5;
```

Query B

```
SELECT S.sid, MAX(P.projectedMargin), MIN(P.projectedMargin)
FROM Polls AS P INNER JOIN States AS S
ON P.sid = S.sid
INNER JOIN Counties AS C
ON S.sid = C.sid
WHERE S.name LIKE 'A%' AND P.cid IS NOT NULL
GROUP BY S.sid;
```

Query C

```
SELECT S.sid, AVERAGE(C.population)
FROM States AS S INNER JOIN Counties AS C
ON S.sid = C.sid
GROUP BY S.sid
ORDER BY AVERAGE(C.population);
```

i. **(1 pt)** Assume the **only** queries executed on the database are Queries A, B, and C. Queries are executed within a transaction, and each transaction consists of a **single** query. (i.e. Query A would be executed within its own transaction.)

Select **all** of the following that guarantee conflict serializability.

☐ 2PL

☐ Strict 2PL

☐ No locking

☐ None of the above

ii. **(1 pt)** Now, consider adding **Query D** shown below. Assume the **only** queries executed on the database are Queries A, B, C, and D. Again, queries are executed within a transaction, and each transaction consists of a **single** query. (i.e. Query A would be executed within its own transaction.)

Query D

```
INSERT INTO Polls
VALUES (20, 5, 186, 4, 3);
```

Select **all** of the following that guarantee conflict serializability.

☐ 2PL

☐ Strict 2PL

☐ No locking

☐ None of the above

iii. (1 pt) Assume the **only** queries executed on the database are Queries A, B, C, and D. Again, queries are executed within a transaction. However, there may be **multiple** queries (including repeats) within the same transaction. (i.e. Query A and B would be executed within the same transaction. Query A can also be executed twice within the same transaction.)

Select **all** of the following that guarantee conflict serializability.

☐ 2PL

☐ Strict 2PL

☐ No locking

☐ None of the above

(d) Performance

For the next 4 questions, indicate how the specified performance metric changes if the given change is implemented and explain your answer.

i. (1 pt) Reduce the number of active transactions once thrashing is observed.

◯ Throughput Increase

◯ Throughput Decrease

ii. (1 pt) Explain your previous answer in 2-3 sentences.

iii. (1 pt) Enforce view serializability instead of conflict serializability.

◯ Throughput Increase

◯ Throughput Decrease

iv. (1 pt) Explain your previous answer in 2-3 sentences.

v. (1 pt) Switch from Strict 2PL to 2PL, assuming there is never an edge in the dependency graph.

◯ Latency Increase

◯ Latency Decrease

vi. (1 pt) Explain your previous answer in 2-3 sentences.

vii. (1 pt) Switch from Undo Logging to Redo Logging, given an infinite buffer size.

◯ Latency Increase

◯ Latency Decrease

viii. (1 pt) Explain your previous answer in 2-3 sentences.

No more questions.