CS W186 Fall 2018 Midterm 2

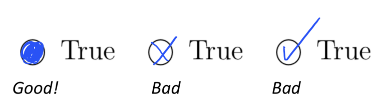
Do not turn this page until instructed to start the exam.

Contents:

- You should receive one *double-sided answer sheet* and a 12-page *exam packet*.
- The midterm has *5 questions*, each with multiple parts.
- The midterm is worth a total of *61 points*.

Taking the exam:

- You have *80 minutes* to complete the midterm.
- All answers should be written on the answer sheet. The exam packet will be collected but not graded.
- For each question, place only your *final answer* on the answer sheet; do not show work.
- For multiple choice questions, please *fill in the bubble or box completely* as shown on the left below. *Do not mark the box with an X or checkmark.*



- Use the blank spaces in your exam for scratch paper.

Aids:

- You are allowed **two** 8.5" × 11" double-sided pages of notes.
- The **only** electronic devices allowed are basic scientific calculators with simple numeric readout. No graphing calculators, tablets, cellphones, smartwatches, laptops, etc.

1 Iterators and Joins (15 points)

1. (5 points) For each of the following five questions, mark True or False.
   A. Chunk Nested Loops join will always perform at least as well as Page Nested Loops Join when it comes to minimizing I/Os.
   B. Grace hash join is usually the best algorithm for joins in which the join condition includes an inequality (i.e. col1 < col2).
   C. In choosing a join order for nested loops join to minimize I/Os, it is best to make the smaller relation the "outer" part of the loop.
   D. Suppose we are joining two tables that are very different in size. In choosing a join order for index nested loops join to minimize I/Os, if both relations have indexes on their join column, it is best to query the index of the smaller relation.
   E. If we can call the next() method on an iterator, then we are using a streaming (on-the-fly) algorithm.

In the next five questions, assume we have the following two database tables with the corresponding details below.

`Students: (sid, sname, syear)`

`Enrolled: (sid, cid, semester)`

variable	symbol	value
pages of `Students` table	$[S]$	200
tuples per `Students` page	p_S	10
pages of `Enrolled` table	$[E]$	100
tuples per `Enrolled` page	p_E	60
pages in memory to perform the join	B	7
I/Os needed to access the leaf of a B+tree	L	2

We want to join `Students` and `Enrolled` on `Students.sid = Enrolled.sid`. Attribute `sid` is the primary key for table `Students`. For every tuple in `Students`, assume there are 3 matching tuples in `Enrolled`. There is an unclustered B+tree index on `E.sid`.

Note: For these questions, do **NOT** include the cost of writing matching output, but **DO** include for the cost of scanning the tables.

2. (2 points) How many **I/Os** will a **grace hash join** take? Assume perfect hash functions, and be sure to choose the best relation for "building" to minimize cost.

3. (2 points) What is the **minimum** number of total **pages** in RAM that it would take to reduce the number of I/Os for **grace hash join**?

4. (2 points) How many disk **I/Os** are needed to perform an **index nested loops join** using the B+tree on `E.sid`?

5. (2 points) After examining the data, you realize that both the `Students` table and `Enrolled` table are sorted by `sid`. To account for this, you want to use **Sort Merge Join**. How many **I/Os** will this join take?

6. (2 points) Suppose that I wanted to do a **Chunk Nested Loops Join** in at most 1100 I/Os. What is the **minimum** number of **pages** in RAM I would need to accomplish this?

2 Parallel Query Processing (10 points)

For the following questions, assume that you have access to the following relations:
`Players (name, team, position, salary, agent)`
`Coaches (name, team, salary)`
Important parameters for this question are summarized in this table:

variable	symbol	value
Number of machines	M	4
Size of Page	s	4 KB
Pages in RAM per machine for joins	B	8 pages
Size of `Players`	[P]	128 pages
Size of `Coaches`	[C]	4 pages
Time for each I/O	t	5 ms

Assume we have 4 machines, each with 8 pages in memory for joins. We will need to measure time, so assume that an I/O takes 5ms. For the following questions, when we ask for execution time we are only concerned with the time associated with I/Os (e.g. assume CPU and other costs are negligible). Assume that we can send individual tuples over the network with no overhead in terms of network cost.

Questions 1 and 2 will deal with the following query:
`SELECT p.name, c.name FROM Players p, Coaches c WHERE c.team = p.team`

1. (4 points) Assuming the `Players` and `Coaches` relations are both round-robin partitioned across 4 machines by an adversary who is trying to maximize our network costs, what is the largest amount of data we ship across the network, **in KB**, of the query above, assuming we execute a sort-merge join?

2. (2 points) Assuming the `Players` relation starts out hash-partitioned on the `position` key across the 4 machines, and that 75% of `Players` gets mapped to one machine, how long **in ms** will it take to complete a parallel scan of `Players`?

Suppose we add another table with the following schema: `Fans (name, team)`, which has 40,000 pages and is round-robin partitioned across 4 **new** machines with only this data.

Questions 3-4 deal with the following query:

```
Select f.name, c.name from Coaches c, Fans f where f.team = c.team
```

3. (2 points) What is the **name** of the join strategy that provides lowest possible network cost (amount of data shipped) to execute the query above?

4. (2 points) What is the amount of data shipped **in KB** to execute that join strategy?

3 Query Optimization (11 points)

1. (2.5 points) For each of the following assertions about left-deep plans, answer True or False.

   A. Two left-deep plans can differ in the order of relations and produce the same output.

   B. Two left-deep plans can differ in the access method for each leaf operator and produce the same output.

   C. Two left-deep plans can differ in the join method for each join operator and produce the same output.

   D. The cheapest plan will always be among the left-deep plans.

   E. The concept of "interesting" orders is not relevant for left-deep plans.

2. (2 points) For each of the following assertions about the System R algorithm, answer True or False.

   A. System R never considers plans with cartesian products because they are suboptimal

   B. System R only explores left deep plans

   C. System R doesn't keep track of interesting orders as they do not reduce I/O cost

   D. The running time of the System R algorithm is at least exponential in the number of tables

Suppose the System R assumptions about uniformity and independence from lecture hold. Assume that costs are estimated as a number of I/Os, without differentiating random and sequential I/O cost.

Consider the following relational schema:

Table Schema	Table Stats	Pages	Indices
CREATE TABLE Customers (id INTEGER PRIMARY KEY, name STRING, age INTEGER, happiness INTEGER)	Nkeys: - id: 100 - name: 90 - age: 100 - happiness: see hist	10	- Clustered alternative 2 index of height 2 on **id** - Clustered alternative 2 index of height 2 on happiness
CREATE TABLE Purchases (order_id INTEGER PRIMARY KEY, customer_id INTEGER REFERENCES Customers(id), customer_name STRING, total_cost INTEGER)	Nkeys: - order_id: 1000 - customer_id: 50 - customer_name: 50 - total_cost: 1000	100	- Unclustered alternative 2 index of height 2 on **order_id**
CREATE TABLE Returns (return_id INTEGER PRIMARY KEY, order_id REFERENCES Purchases(order_id), customer_id REFERENCES Customers(id))	Nkeys: - return_id: 5000 - order_id: 5000 - customer_id:100	500	- Unclustered alternative 2 index of height 2 on **return_id**

Assume the distribution on `Customers.happiness` is as shown in Figure 1. Each bin is inclusive of the min and exclusive of the max, [min, max).

[1-2)	[2-5)	[5-7)	[7-9)	≥ 9
5%	15%	10%	30%	40%

Figure 1: Histogram on `Customers.happiness`

Suppose you're executing the following query:

```
SELECT id, name
FROM Customers c, Purchases p
WHERE c.happiness >= 2
   AND c.name = p.customer_name
   AND c.happiness < 7
```

3. (1 point) What will be the selectivity for the predicate `c.name = p.customer_name`?

4. (1 point) What will be the selectivity of `c.happiness` $\geq$ `2 AND c.happiness` $<$ `7`?

5. (1 point) How many tuples do we estimate to be in the output of the query? Choose *one* of the options below.

   A. (answer to q3) $*$ (answer to q4) $*$ |`Customers`| $*$ |`Purchases`|

   B. (selectivity of `c.happiness` ≥ 2) $*$ (selectivity of `c.happiness` < 7) $*$ |`Customers`|$*$|`Purchases`|

   C. (answer to q3) $*$ (selectivity of `c.happiness` ≥ 2)
   $*$ (selectivity of `c.happiness` < 7) $*$ |`Customers`| $*$ |`Purchases`|

For problems 6-7, refer to the following query:

```
SELECT c.id, r. return_id, p.order_id
FROM Customers c, Purchases p, Returns r
WHERE c.id = p.customer_id
     AND p.order_id = r.order_id
     AND r.customer_id = c.id
     AND c.happiness < 2
ORDER BY c.id
```

6. (2.5 points) Which of these table scans will output an interesting order? Mark True for correct answers and False for incorrect answers.

   A. Index scan on `happiness` for `Customers`

   B. Index scan on `id` for `Customers`

   C. Full table scan on `Customers`

   D. Index scan on `order_id` for `Purchases`

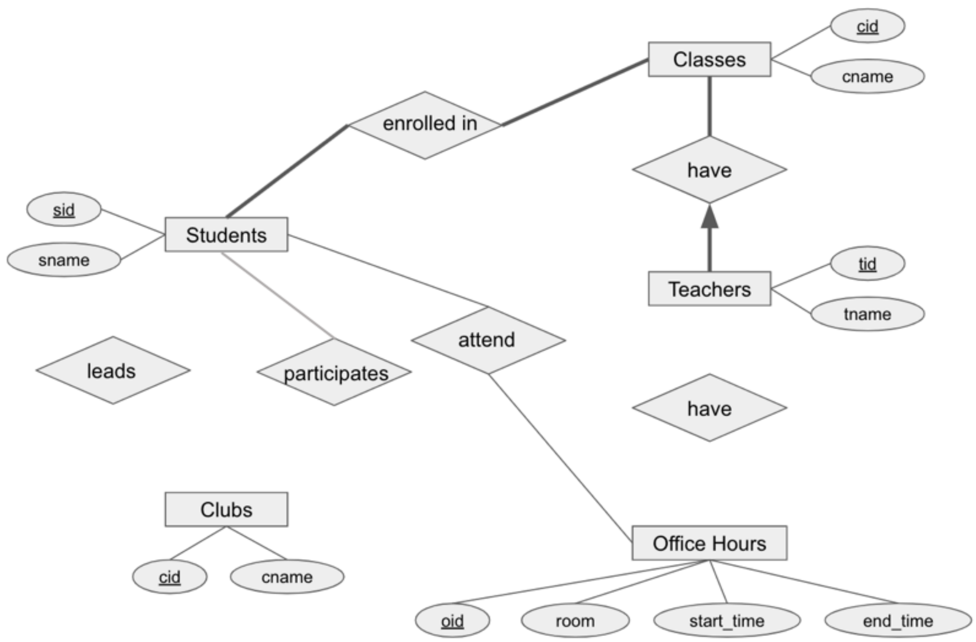
   E. Index scan on `return_id` for `Returns`

7. (1 point) True or False: Consider the SQL query above question 6. System R will choose to do a full table scan rather than an index scan for the Customers table.

4 ER Diagrams(14 points)

For questions 1-6, you will use the following ER Diagram, which represents the commitments that teachers and students have during the semester. (Hint: You might want to fill the diagram while you read these requirements here).

- Students may lead multiple clubs, and every club has one student leader.
- Students can also participate in multiple clubs, and every club has at least one student member.
- Every teacher has multiple office hours, and one teacher leads each office hour.



1. (1 point) Which edge should we draw to connect the **Clubs** entity with the **leads** relationship set?
   A. Thin Arrow
   B. Thick Arrow
   C. Thin Line
   D. Thick Line

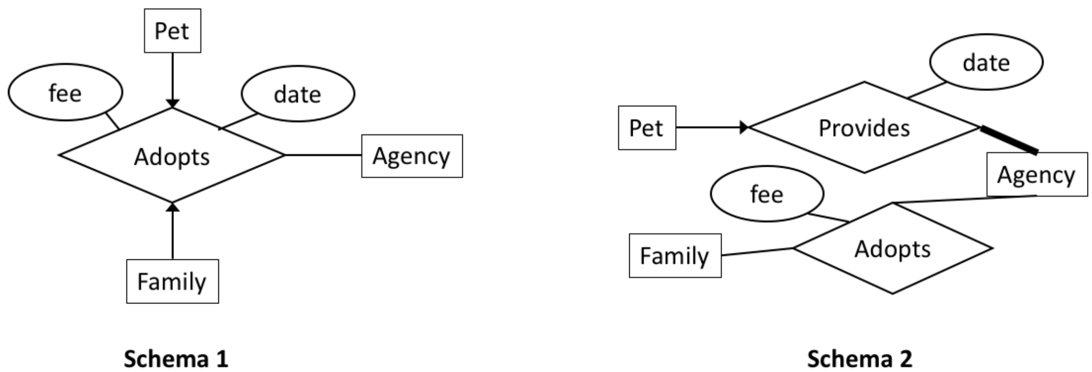
2. (1 point) Which edge should we draw to connect the **Students** entity with the **leads** relationship set?
   A. Thin Arrow
   B. Thick Arrow
   C. Thin Line
   D. Thick Line

3. (1 point) Which edge should we draw to connect the **Clubs** entity with the **participates** relationship set?
    A. Thin Arrow
    B. Thick Arrow
    C. Thin Line
    D. Thick Line

4. (1 point) Which edge should we draw to connect the **Office Hours** entity with the **have** relationship set?
    A. Thin Arrow
    B. Thick Arrow
    C. Thin Line
    D. Thick Line

5. (1 point) Which edge should we draw to connect the **Teachers** entity with the **have** relationship set?
    A. Thin Arrow
    B. Thick Arrow
    C. Thin Line
    D. Thick Line

6. (1 point) True or False: Can a class be taught by multiple teachers?
    A. True
    B. False



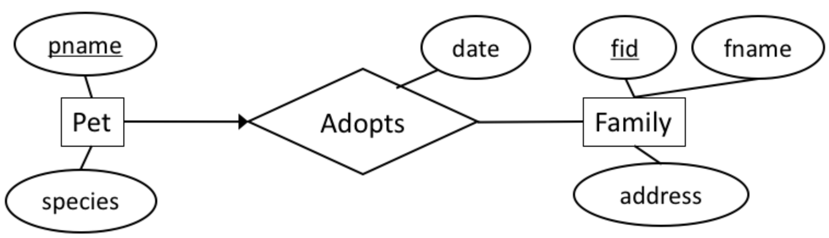
7. (4 points) Above are two alternative schemata[1] that represent pet adoptions. For each of the following assertions, mark True or False.

   A. In schema 1, a family can adopt at most one pet.

   B. In schema 2, if there are k agencies, a family can adopt at most k times.

   C. In schema 2, there is no record of which pet was adopted by which family.

   D. In schema 1, a family can adopt a pet without an agency being involved.



8. (4 points) To capture the ER Diagram above, we create three relations: Pet, Adopts and Family. For each of the following assertions, mark True or False.

   A. The Pet table's primary key includes the column fid.

   B. The Adopts table's primary key includes the column pname.

   C. The Adopts table's primary key includes the column fid.

   D. The Adopts table's column fid can be declared NOT NULL.

[1] "Schemata" is the plural of schema.

5 Text Search (11 points)

1. (5 points) For each assertion, fill in the corresponding bubble True or False.
   A. A postings list is a heap file of docIDs for a term.
   B. In IR's "bag of words" model, the word "running" is converted to "run", so "running" is an example of a stop word.
   C. IR is used mostly with unstructured text data.
   D. Inverted files are so named because they are structured with document IDs ordered in descending order.
   E. In general, relational DBMSs are faster at handling individual updates and deletes than Text Search Engines.

Questions 2 to 6 refer to finding all docs matching the following Boolean expression:

```
"Berkeley" AND ("Database" OR "Computer") AND NOT "Stanford"
```

Assume all term searches use index scans. Assume no optimizations are applied.

2. (1 point) How many index scans will be done to perform this search? Choose *one*.
   A. 1
   B. 2
   C. 3
   D. 4

3. (1 point) How many unions are performed? **Answer in a nonnegative integer.**

4. (1 point) How many intersections are performed? **Answer in a nonnegative integer.**

5. (1 point) How many set subtractions are performed? **Answer in a nonnegative integer.**

6. (2 points) Mark True or False for each of the following assertions regarding the efficiency of IR queries.
   A. To perform set operations we can use hash joins without the partitioning phase because postings lists are already hash partitioned.
   B. To perform set operations we can use merge joins without sorting because posting lists are already sorted.
   C. Performing a set operation on two postings lists requires no more than 3 I/O buffers in memory: two for input to the operation, one for output.
   D. The B+-tree containing the postings lists is perfectly clustered: that is, the heap file it points to is organized by (`term`, `docId`).