Department of Electrical Engineering and Computer Science

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

6.5830/31 Database Systems: Fall 2023 Quiz I

There are 13 questions and 12 pages in this quiz booklet. To receive credit for a question, answer it according to the instructions given. *You can receive partial credit on questions.* You have **80 minutes** to answer the questions.

Write your name on this cover sheet AND at the bottom of each page of this booklet.

Some questions may be harder than others. Attack them in the order that allows you to make the most progress. If you find a question ambiguous, be sure to write down any assumptions you make. Be neat. If we can't understand your answer, we can't give you credit!

THIS IS AN OPEN BOOK, OPEN NOTES QUIZ.
LAPTOPS MAY BE USED FOR NOTES AND SLIDES; NO PHONES, INTERNET, NOR ON LAPTOP LLMS, SQL SHELLS, NOR OTHER ASSISTANTS.

Name:

I Cost Model

Assume a row-oriented disk-based database with the following tables:

```
CREATE TABLE airport{
    code CHAR(4) PRIMARY KEY,
    name CHAR(115) NOT NULL,
    country-code CHAR(2) NOT NULL
}

CREATE TABLE flight{
    nb CHAR(8) PRIMARY KEY,
    origin CHAR(4) REFERENCES airport(code) NOT NULL,
    destination CHAR(4) REFERENCES airport(code) NOT NULL
}

CREATE TABLE passenger{
    pid INT PRIMARY KEY,
    name CHAR(105) NOT NULL,
    flight-nb CHAR(8) REFERENCES flight(nb) NOT NULL
}
```

The airport table has 1K records, flights has 1M records, and passengers table 100M records and every flight has exactly 100 passengers.

All the data is organized as a heap file with a page size of 64KB, all fully filled except potentially the last one. While each record has a header (e.g., for the rid) the heapfile itself does NOT have any additional meta-data/header. Chars are 1 byte and ints are 8 bytes.

The heap file is stored on a spinning disk with the following characteristics:

Seek: 10ms

Bandwidth: 100MB/sec

Throughout this exam we use KB = 1000 bytes, MB = 1000 KB, etc.

Name:

1. [9 points]: Calculate (1) the number bytes for each record in the respective tables assuming a 4 byte header per record, (2) the time to scan the tables including seek, and (3) the number of pages (assume all pages are dense except potentially the last one). Write your answer into the following table.

Table	Bytes	Scan-time	# pages
airport			
flight			
passenger			

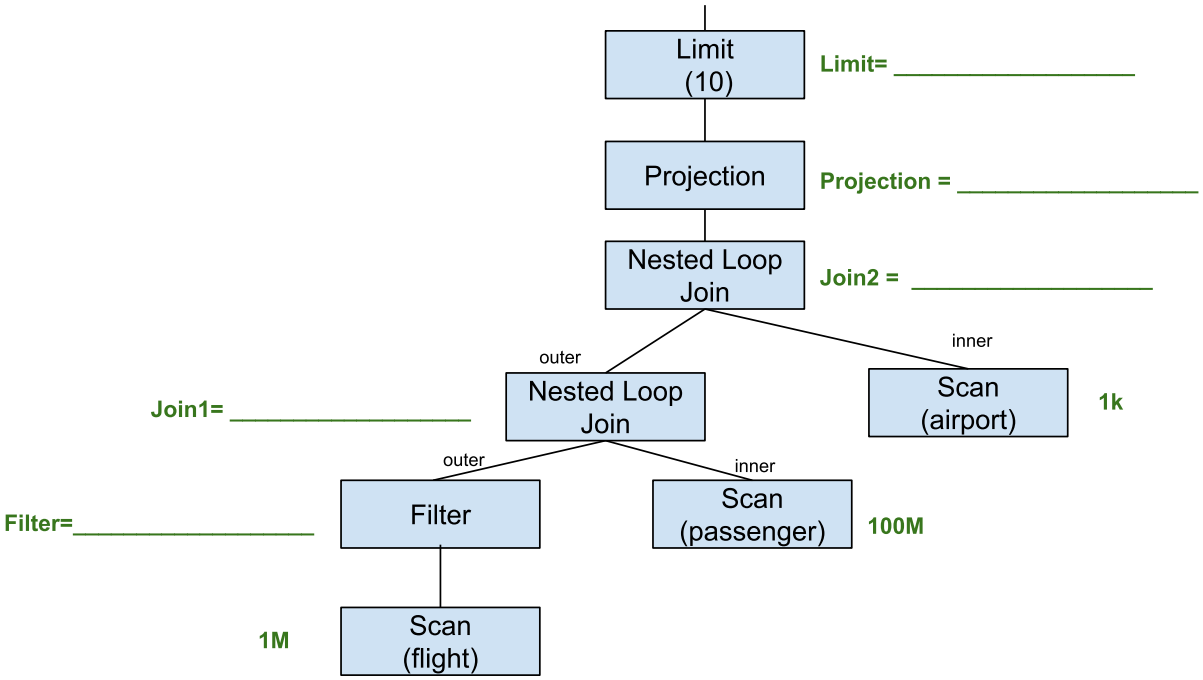
Name:

2. [5 points]:

Consider the following query:

```
SELECT p.name, a.code
FROM airport a, flight f, passenger p
WHERE p.flight-nb = f.nb
AND f.origin=a.code
AND f.flight-nb in ('TP1023','TP343','TP225','TP3423',
'TP107','LH130','LH1034','UA455','UA230','UA260')
LIMIT 10
```

(Fill in the blanks to indicate the output cardinality of each intermediate result in the the following physical execution plan for this query)



Name:

3. [10 points]:

What is the estimated execution time considering only I/O for the query plan above assuming a cold (empty) cache, a 2000 page cache (with LRU), and that each operator computes and stores the entire (intermediate) result in memory before the next operation begins? That is, the operators do NOT implement an iterator tuple-at-a-time model. [You may round results up/down to second granularity]

(Write your answer in the space below.)

4. [6 points]: Assume the query plan above with the same assumptions (cold cache etc.), but this time we execute it using a tuple-at-the-time iterator model (like in GoDB). Would the query execute faster or slower?

(Write your answer and short explanation below.)

5. [8 points]: From the list below, select all *unclustered* indexes which would help to improve the execution time of the query above (the indexes may result in a different physical plan). Provide a brief explanation of why you selected the indexes you did:

A. Index on airport(code)

B. Index on airport(name)

C. Index on flight(nb)

D. Index on flight(origin)

E. Index on passenger(pid)

F. Index on passenger(flight-nb)

Explanation:

6. [4 points]: Assume the query plan above with the same assumptions (cold cache etc.), but this time we execute it using an early materialization column-store design without compression. Would the query execute faster or slower?

(Write your answer and short explanation below.)

II Iterator Model

7. [10 points]:

Consider a database that implements a tuple-at-a-time iterator model as in GoDB. Which of the following statements are true?

(Circle 'T' or 'F' for each choice.)

T F The iterator model can help to minimize the amount of memory needed for temporary results

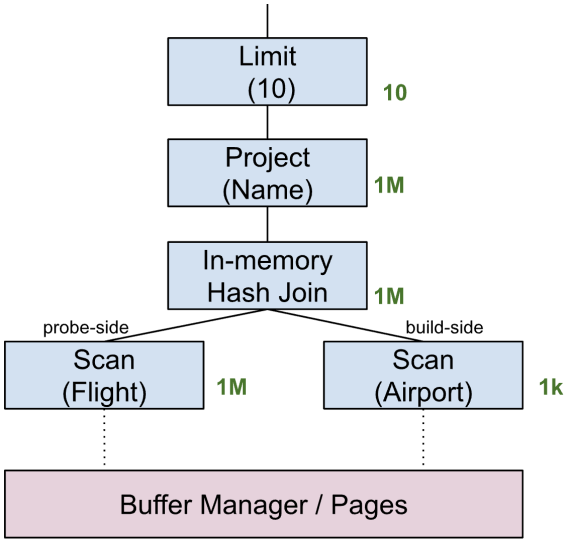
T F The iterator model can cause a lot of (virtual) function calls per tuple, which are hard to optimize for modern compilers

T F The iterator model might fetch fewer pages from disk than alternative execution models (e.g., computing one operator at a time)

T F The iterator model helps to take full advantage of the CPU cache for queries that operate on large amounts of data with many joins

T F The iterator model allows for a lot of parallelism and can be easily SIMD-optimized

8. [10 points]: Consider the following iterator with the limit operator at the root and scan operators at the bottom. Here we use the same schema for the airports database as above. The numbers to the right of each operator indicate cardinality estimates (e.g., the airport scan yields 1k tuples). The join between flights and airports is on the *origin* column and every flight has exactly one airport it originates from. Assume the iterator tree implements the tuple-at-a-time next() API and that the scan operator internally uses a shared buffer manager that also implements the next() API to get tuples from a particular page (like the GoDB implementation). Here, "build-side" means the relation the hash table is built on.



How many next() invocations are required until the query is completely computed? Note, that each next() call on every operator counts as one invocation. For example, if projection.next() calls filter.next() this counts as two invocations.

(Write your answer below.)

Name:

III Optimization

Consider the following database about animals, the hats they wear, and the magical effects of these hats.

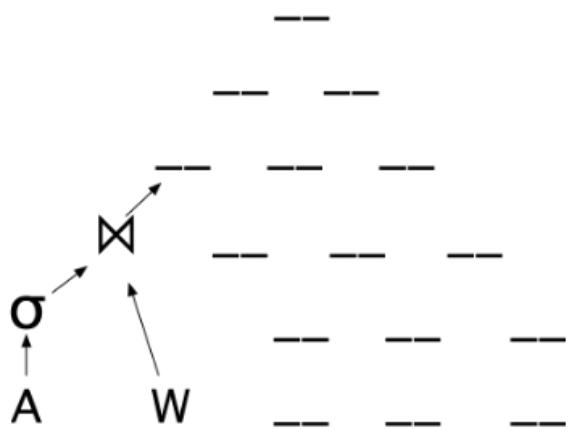
```
animals(AId, AName, ASpecies)
hats(HId,HColor,HSize)
wears(W_AId, W_HId)
magic(MId, MDesc)
hatmagic(HM_HId,HM_MId)
```

Here underlined fields are a part of the primary key and *italic* fields are a foreign key reference.

You want to find the magic that the animal "Tim the Beaver" gets, e.g. evaluate this query:

```
SELECT MDesc FROM
animals A JOIN wears W ON AId = W_AId
JOIN hats H ON HId = W_HId
JOIN hatmagic HM ON HM_HId = HId
JOIN magic M ON MId = HM_MId
WHERE AName = "Tim the Beaver"
```

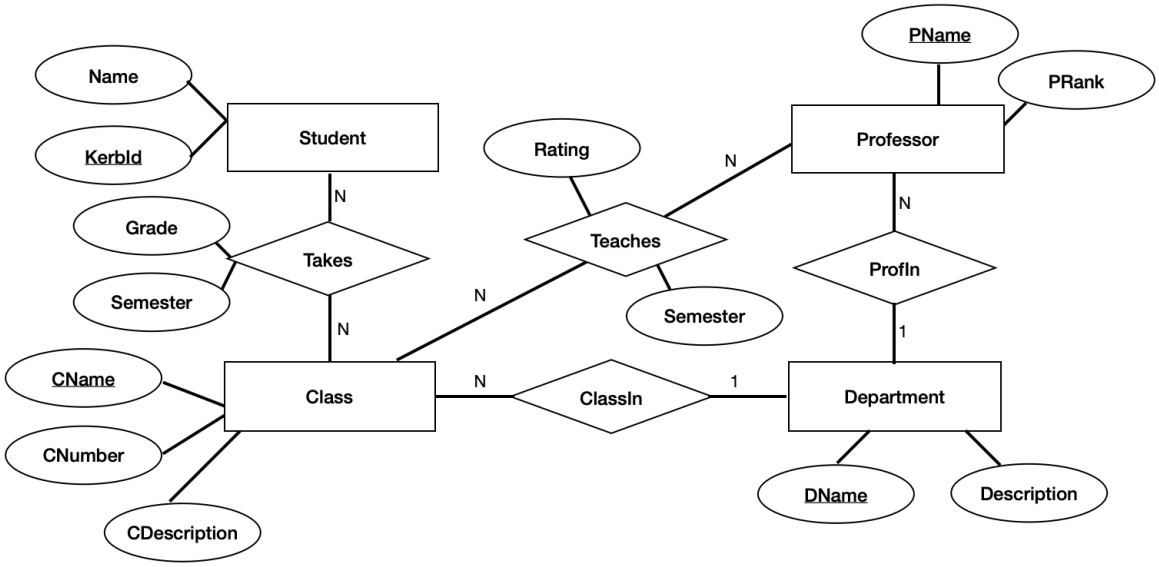
9. [6 points]: Suppose the Selinger optimizer chooses to compute the join between animals and wears first. Complete the following query plan with *one* plan the Selinger optimizer might generate (you can choose any plan it would produce). You are given no information about table sizes or join selectivities. Fill in the blanks with a join symbol (⋈) or the abbreviation of a table (HM, H, M). You will need to leave some blanks empty.



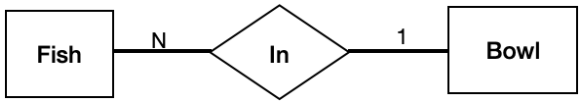
Name:

IV ER Diagrams

Consider the following ER diagram



Here underlined names are primary keys. Because the N:1 notation can be confusing, consider this example, which means "there are many fish, but each is in one bowl".



10. [6 points]: Which statements about a database that follows the constraints shown in the ER diagram can be true:

(Circle 'T' or 'F' for each choice.)

T F In the same semester, one student can take multiple classes, and one class can be taken by multiple students

T F One class can be in multiple departments

T F One professor can be in multiple departments

T F One professor can teach multiple classes in a semester

T F In a semester, a class's rating is independent of the professor who taught it

T F A professor's rank depends on the department they are in

Name:

11. [10 points]: Which of the following table definitions would exist in the schema for this diagram. If you circle false, write a brief explanation for why not.

T **F** Class : (CName, CNumber, CDescription, CProfessor)
Explanation if false:

T **F** Student : (KerbId, Name)
Explanation if false:

T **F** Professor: (PName, PRank, PDepartment)
Explanation if false:

T **F** ClassIn: (CName, DName, Semester)
Explanation if false:

T **F** Takes(KerbID, CName)
Explanation if false:

V SQL

Consider the following database representing people and the dorms they live in:

Dorm (DormID, Name, Address)
DormRooms (RoomId, RoomNo, *DRDormId*, SqFt, Beds)
Student (KerbId, SName, *SRoomId*)
ConnectsTo (*RoomIdA*, *RoomIdB*)

Here underlines represent primary key attributes, and *italics* represent foreign key attributes. ConnectsTo denotes that two rooms are connected by a door or corridor. Some rooms may have 0 beds, indicating that they are common areas. For convenience the ConnectsTo relation duplicates each connection in the opposite direction (e.g., if there is an entry "1,2", there is also an entry "2,1").

Complete the following queries:

12. [8 points]: Find students who do not have a dorm room:

```
SELECT KerbId FROM

Student _____ DormRooms

ON SRoomId = RoomID

WHERE _____

_____
```

13. [8 points]: Find the total square feet of all rooms connected to room 27:

```
SELECT _____ FROM

DormRooms dr _____

_____

WHERE dr.RoomID = 27

_____
```

End of Quiz I!

Name: