NAME:

UID:

UMD College Park Department of Computer-Science
CMSC 424: Database Design Spring 2018
Second Midterm Exam

There are 8 questions (most of them with multiple parts) and 11 pages in this exam (including this page). The maximum number of points you can receive is 50 (before bonus points). To receive credit for a question, answer it according to the instructions given. You have 70 minutes to answer the questions (we will not start until 4 or 5 mins into the class period).

Write your name on this cover sheet (see above) AND at the top of each page of this exam. Also, please also write your UID on this page (but only this page). This exam will be scanned and graded in Gradescope. Gradescope uses OCR technology to read the text that you write. So it's very important that you write your answers in clear handwriting. Otherwise, Gradescope may make mistakes when it reads your text. This is also true when you write your name above on this cover sheet --- Gradescope reads your name to figure out whose exam this is. Please write your name very clearly on this page. On the rest of the pages, you must write your name in order to receive the bonus points for this exam (see below), but there, your name will only ever be read by a human if your exam gets mixed up with exams from other students after the staples are removed. So for the rest of the pages, writing your name clearly is not important. But for this one, it is important.

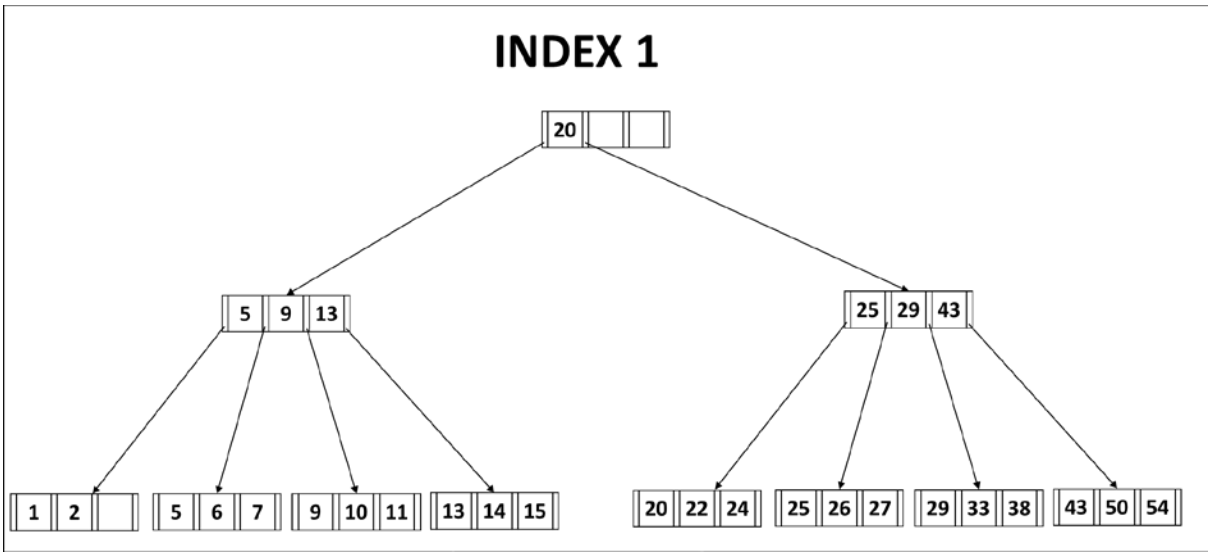
Please do not remove the staple from this exam, and please do not rip out pages. This exam has been carefully written (with relevant material copied on each page) so that you will not be tempted to do this. Pages that are ripped out significantly slow down the destapling and scanning process.

There are boxes in which your answer to each question are to be written. Gradescope will only look inside those boxes to see your answer to that question. If there are important assumptions that you are making about a question, we have provided a box at the end of the exam in which you can write down whatever you want (make sure you indicate which question you are writing about), and a human will look at the contents of that box and adjust the grading for the question you are discussing if necessary. Anything that you write outside of an answer box or the miscellaneous box at the end of the exam will be ignored by both the human and computer graders.

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 ask the professor during the exam or write down any assumptions you make in the miscellaneous box at the end of the exam.

You receive 25% credit for any question that you leave **COMPLETELY** blank. This credit goes away once you try to start to answer it. In addition, every student that successfully writes their name on every page of this exam will receive a bonus amount of points based on a complicated formula based on the average number of questions (weighted by point value) left empty across the class. On the previous exam, it came out to 5 bonus points per student. This exam will likely come out to slightly fewer bonus points (probably 2 or 3).

NAME:



For all questions on B+ trees on this exam, use the algorithm from lecture/your textbook.

Question 1: This question is based on the B+ tree index marked as INDEX 1 above. Pointers between leaf nodes are not shown, since they are not relevant for this question. For this index, leaf nodes must have between 2 and 3 keys (they are considered "underfull" if they have 0 or 1 key), and non-leaf nodes must have between 2 and 4 pointers (they are considered "underfull" if they have 0 or 1 pointer).

1A [1 point] What is the height of the index? 3

1B [1 point] How many non-leaf nodes are there in this index? 3

1C [1 point] True or False: (Assume that the data in the table is sorted by one and only one of the attributes.) Since the data is sorted in this index, we can conclude that this is a primary index.

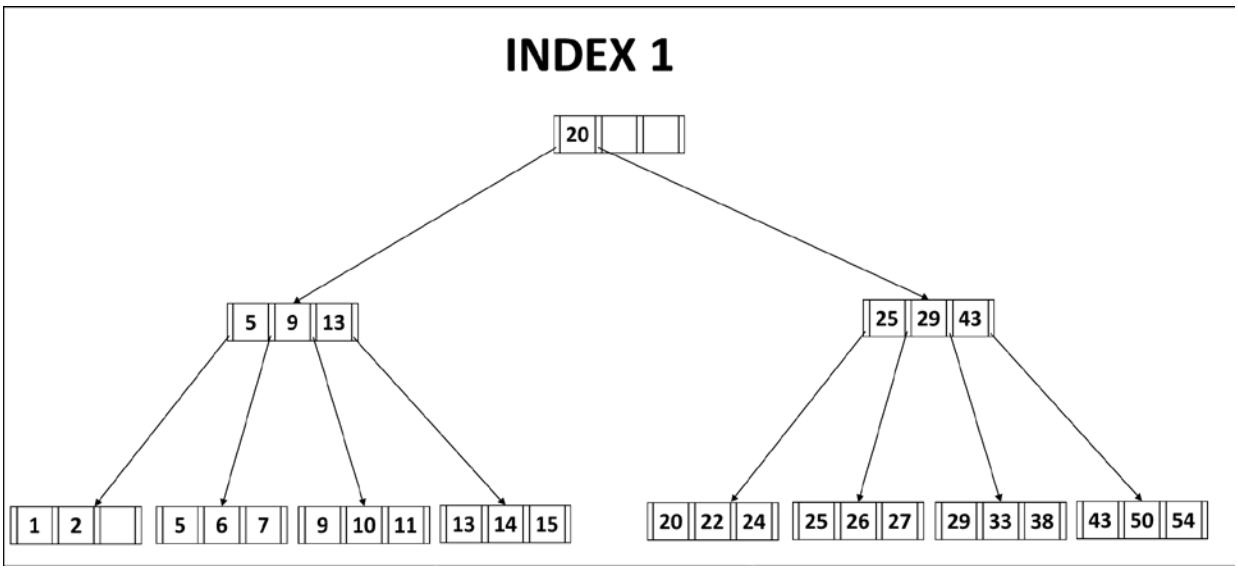
false

For the following questions, we will ask you to fill in the contents inside a node of the index. Please do not worry about pointers for these questions. Just write all the search keys in that node, separated by a '+'. So for example, if we asked you what is in the current node in INDEX 1 which is the parent of the leaf node with key value 33, you would write: 25 + 29 + 43

1D [2 points] What are the contents of the **parent** of the leaf node with key value 7 after a key with value 8 has been inserted into the tree? 5 + 7

1E [2 points] What are the contents of the **root** node after a key with value 60 has been inserted into the **original** tree shown in INDEX 1?
(Pretend that the insert from 1D never happened) 20 + 43

NAME:



(The above image is a copy of the index on the previous page. We are not done asking questions about it, and so we are copying it to this page as a convenience, so that you don't have to flip back and forth between the pages!)

1F [2 points] What are the contents of the **parent** of the leaf with key value 15 after the key with value 2 has been deleted from the **original** tree shown in INDEX 1? (Pretend that the inserts from 1D and 1E never happened)

6 + 9 + 13

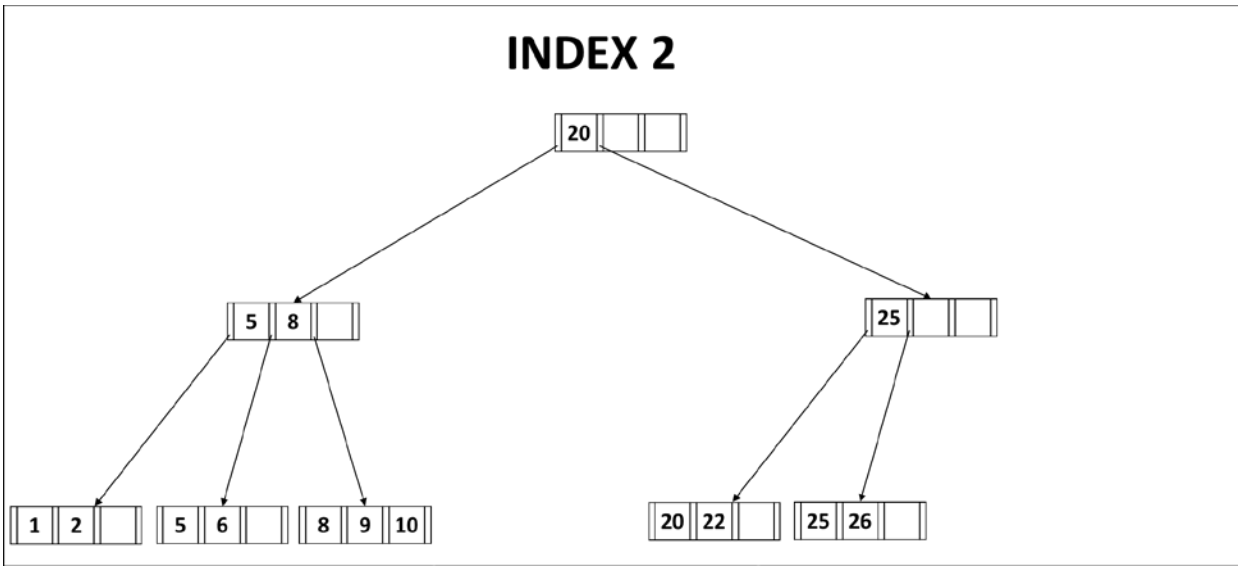
1G [1.5 points] What is the minimum number of new keys that could be inserted into the **original** tree shown in INDEX 1 where the last of these inserts would cause the height of the tree to increase by 1 (Pretend that the inserts from 1D and 1E, and the delete from 1F never happened)

4

1H [1.5 points] What is the maximum number of new keys that could be inserted into the **original** tree shown in INDEX 1 such that the height of the tree will not change (Pretend that the inserts from 1D, 1E, and 1G along the delete from 1F never happened)

25

NAME:



Question 2: This question is based on the B+ tree index marked as INDEX 2 above. Pointers between leaf nodes are not shown, since they are not relevant for this question. For this index (similar to INDEX 1), leaf nodes must have between 2 and 3 keys (they are considered "underfull" if they have 0 or 1 key), and non-leaf nodes must have between 2 and 4 pointers (they are considered "underfull" if they have 0 or 1 pointer).

2A [2 points] What are the contents of the **parent** of the leaf node with key value 2 after the key with value 1 has been deleted from INDEX 2?

8

2B [2 points] What are the contents of the **root** node of the index after the key with value 20 has been deleted from the **original** tree shown in INDEX 2? (Pretend that the delete from 2A never happened)

5 + 8 + 20

2C [2 points] What is the maximum number of keys that could be deleted from the **original** tree shown in INDEX 2 such that the height of the tree will not change (Pretend that the deletes from 2A and 2B never happened)

3

NAME:

Question 3: Short answers

3A [1.5 points] True or false: No matter what has been inserted or deleted from a B+ tree in the past, it is impossible for the same key value to appear in multiple (different) **non-leaf** nodes of B+ tree.

true

Assume a table , A, with 100,000 tuples, that is sorted by attribute A.x. The following query is sent to the database system, for which 10,000 tuples are returned:

```
SELECT A.x, A.z
FROM A
WHERE A.y = 50
```

3B [1 point] True or false: If we have an index on A.y, then using the index for this query will probably be faster than doing a sequential scan.

false

3C [1 point] True or false: If we have an index on A.x, but not on A.y, then using the index on A.x for this query will probably be faster than doing a sequential scan.

false

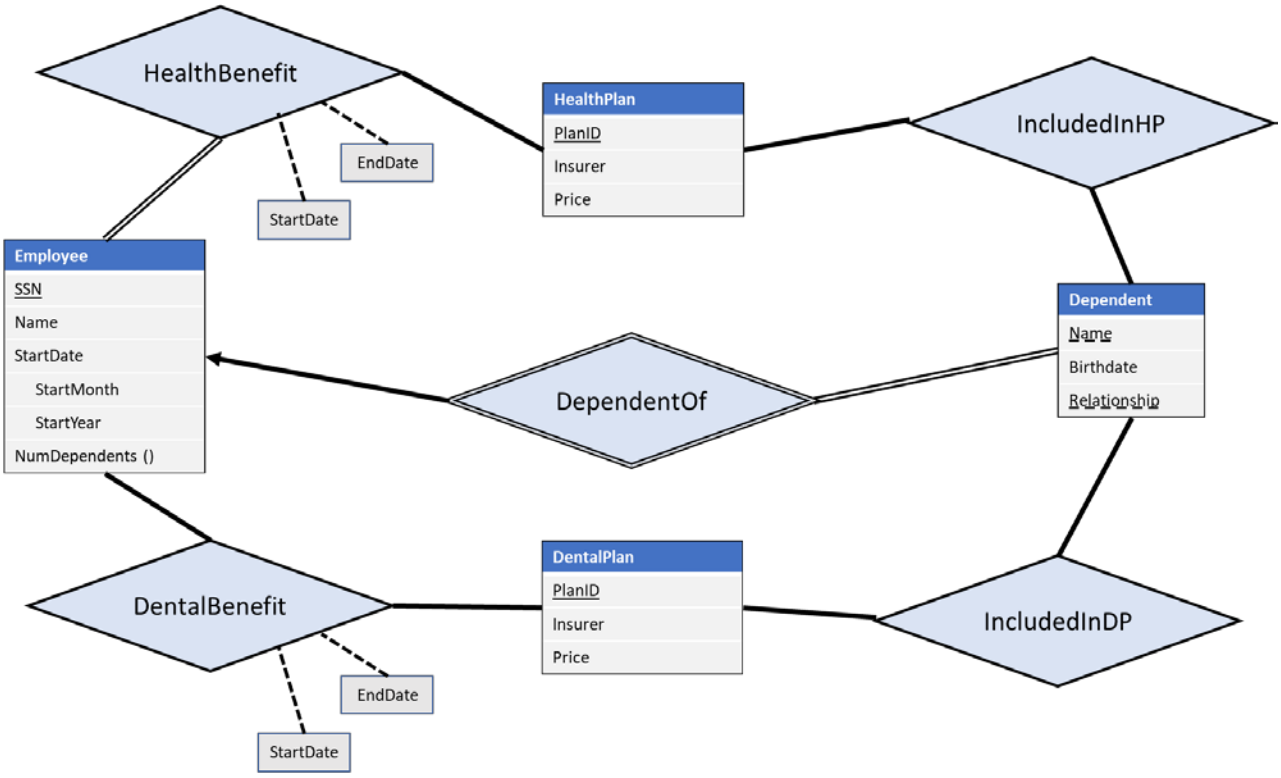
3D [3 points]: Throughout Chapter 12, the cost of a query plan is calculated in terms of t_S and t_T. where t_S is the number of milliseconds to prepare the disk for a random access (disk seek time and rotational latency combined) and t_T is the time (in milliseconds) to transfer a block of data. Let's say that a particular disk rotates at Z rpm. Please calculate the cost of a sequential scan (in milliseconds) of a table that is stored contiguously on 25% of a single disk track. (You can assume for this question that once the disk head is in the correct location of the start of a sequential series of blocks to read, blocks can be transferred at the same speed that a disk can rotate.)

Assume that
t_S = X milliseconds
t_T = Y milliseconds

So your formula may include variables X,Y, and Z, but **no other variables**.

X + 60000/4Z

NAME:



Question 4: The above figure shows an ER Diagram of a simplified employee benefits database. It uses the notation from the textbook, such as: (1) An identifying relationship of a weak entity set is represented with two diamonds rather than one (2) composite attributes have their component attributes indented underneath the composite attribute name (3) derived attributes are represented with () after the attribute name (4) primary keys have a solid underline (5) discriminators have a dashed underline (6) total participation is represented as a double line (7) the "one" side of a many-to-one or one-to-one relationship ends in an arrow (8) attributes of relationships are connected to the relationship via a dotted line.

4A [1 point] What is the name of the weak entity set in this diagram?

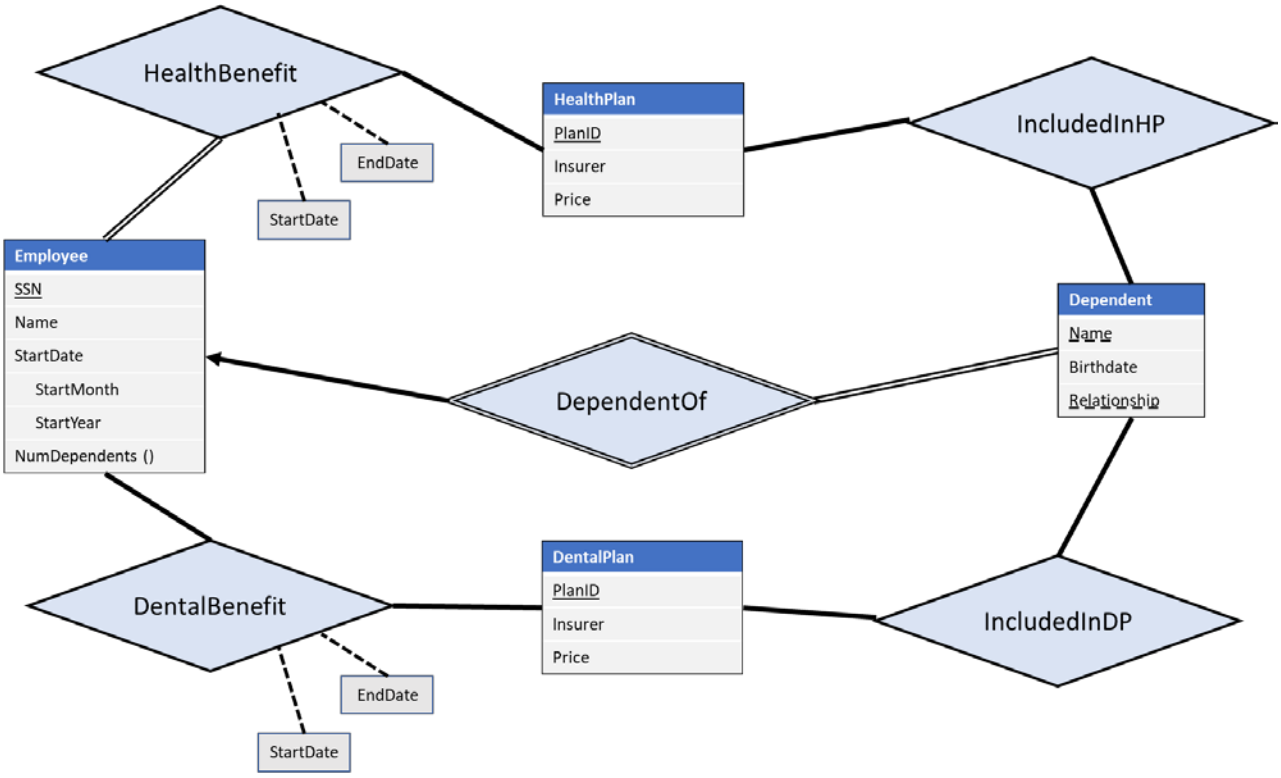
dependent

The next few questions ask you about the result of converting this ER diagram into a relational schema. For all of these questions, assume that schema redundancy is dealt with by combining schemas according to the algorithm that we discussed in class / your textbook such that the combination of schemas will **not** necessitate the introduction of new null values to the tables.

4B [1 point] True or false: Since NumDependents is a derived attribute, we don't need to include it in the relational schema when we convert this ER diagram into a relational schema.

true

NAME:



The figure above is the same figure as the previous page, copied here for your convenience.

4C [2 points] Not counting NumDependents (if you answered "false" to 4B), how many attributes does the **Employee** table have after converting this ER diagram into a relational schema ?

4

4D [2 points] How how many attributes does the **Dependent** table have after converting this ER diagram into a relational schema? (If the table doesn't exist after conversion, write: 0)

4

4E [2 points] How many total tables are created after converting this ER diagram into a relational schema?

8

NAME:

A	B	C	D
1	1	1	1
2	2	2	1
3	1	1	1
4	4	3	1
5	4	3	1
6	3	7	1
6	3	5	1

Attribute names

Question 5A [3.5 points] The table above contains a table with 7 rows and 4 columns all of type integer. Please list all functional dependencies that exist in this instance of the table that involve attribute B, and are of the type X → Y (i.e. one attribute on the left side of the arrow, and one on the right side). Please write your answers in the box below, one functional dependency per line, sorted in order of the attribute on the left side of the arrow (if more than one of your answers has the same attribute on the left side of the arrow, then sort those answers by the attribute on the right side of the arrow).

A -> B
B -> B
B -> D
C -> B

Question 5B [2 points] Of all of the functional dependencies that you listed above, which one are you most suspicious of, in the sense that there is unlikely to be a real-world relationship between those two attributes? B → D

NAME:

ID	Name	Age	Major
1	Joe	18	CS
2	Jill	19	CS
3	Jane	18	CS
4	Jill	20	BIO
5	Jake	21	HIST
6	Jeff	20	BIO
7	Jewel	18	CS

Attribute names

Question 6 [1.5 points]: True or false: (Assume that ID has already been declared as the primary key for the table above.) We cannot be sure if the table above is already in BCNF.

Q ambiguous

ID	Name	Age	Major
1	Joe	18	CS
2	Jill	19	CS
3	Jane	18	CS
4	Jim	20	BIO
5	Joe	21	HIST
6	Jeff	20	BIO
4	Jim	20	CHEM

Attribute names

Question 7: When decomposing the table above using the BCNF decomposition method based on the functional dependency: ID → Name, Age, two tables are created:

Question 7A [1.5 points] List the attributes for the table which contains the age attribute:

ID, Name, Age

Question 7B [1.5 points] List the attributes for the table which does not contain the age attribute.

ID, Major

Question 7C [1 point]: True or false: We can be sure that the table which contains the age attribute is now in BCNF.

True

NAME:

Question 8: The following method attempts to use a PreparedStatement in JDBC to insert a row into a simple table called Students with just two attributes: an integer identifier attribute and a varchar (30) attribute corresponding to the student's name.

8A [1 point]: Please send the correct parameter(s) to con.prepareStatement. (You can assume that the connection was created correctly and successfully).

8B [3.5 points]: Please complete the missing code to ensure that the new row (the correct values of which are stored in the two variables at the beginning of the method) is inserted into the table correctly. (Assume that ps was created successfully, even if your answer to 8A was incorrect. You **must** use ps to perform this insert.)

```
 public void writeStudent() {
   int uid = 32145;
   String name = "John Doe";
   try {
     Connection con = getConnection(); //assume getConnection() works
     String sql = "INSERT INTO Students(id, name) VALUES(?,?);";

     PreparedStatement ps =
         con.prepareStatement(    sql           );
// Your code for Question 8A goes ^ (between the parentheses above)

/* Your code for Question 8B starts here
  ps.setInt(1, uid);
  ps.setString(2, name);
  ps.executeUpdate()









    Your code for Question 8B ends here */

     logger.trace("adding student with statement: "+ sql);
   }
   catch (SQLException e) {
     logger.warn(e.toString());
   }
 }
```

8C [2 points]: True or false: The third line of code is currently `String name = "John Doe";` If, instead, it was: `String name = "John Doe; DROP TABLE Students;--";` then all of the tuples in the Students table may get deleted. false

NAME:

Write in the box below any notes about any of your answers on this midterm exam that you want a human to read. Most students will probably leave this box empty, but if you do put something inside of it, please make sure to clearly indicate which question/answer you are talking about.

I really enjoyed taking this exam!