University of California, Berkeley

CS 186 Introduction to Databases, Spring 2014, Prof. Dan Olteanu

MIDTERM

- This is a closed book examination but you are allowed one 8.5" x 11" sheet of notes (double sided).
- No cell phones/calculators/electronic devices of any sort are allowed.
- Please display your student ID card.
- You should answer as many questions as possible. You should read all of the questions before starting the exam, as some of the questions are substantially more time-consuming than others.
- There are 100 points in total distributed over five questions.
- Write all of your answers in the designated space provided with each question. We will be grading only answers written in the designated space.
- You must put your CS 186 class account at the top of each page before you start the exam.

GOOD LUCK!

Question 1.[Potpourri] Minimum points: 0. Maximum points: 20.

For each of the following statements under **A**, **B**, and **C** below, each correct tick is worth 1 point, each wrong tick is worth -1 point, and no tick is worth 0 points.

A. Tick the correct TRUE/FALSE boxes for each of the following statements.

Statement	**TRUE**	**FALSE**
1. Multiple indexes can be maintained for a single table.	☐	☐
2. An unclustered index can be sparse.	☐	☐
3. Physical data independence is an important database concept and deals with hiding the details of the storage structure from user applications.	☐	☐
4. UB-Trees are equivalent to B^+-Trees over composite keys.	☐	☐
5. One-dimensional Point Quad Trees are unbalanced binary search trees.	☐	☐
6. Data entries in R-Trees are themselves regions.	☐	☐
7. When a node splits in a k-d tree index, it is along each of the k dimensions.	☐	☐
8. Division can be expressed in Relational Algebra using universal quantification.	☐	☐
9. Disk space management and buffer management in Database Systems are very similar to file management and virtual memory in Operating Systems.	☐	☐
10. A record id (rid) is a pair of page id and offset within the page.	☐	☐
11. On record deletion, the space taken by the record on the page is reclaimed and the page is compacted if necessary.	☐	☐

B. Consider relations R and S, with sizes $|R| = N$ and $|S| = M$ respectively, and $N > M$. Both relations have a primary key sid. Also consider the following table definitions:

```
CREATE TABLE A AS SELECT * FROM R INNER JOIN S ON R.sid=S.sid;

CREATE TABLE B AS SELECT * FROM R LEFT OUTER JOIN S ON R.sid=S.sid;

CREATE TABLE C AS SELECT * FROM R RIGHT OUTER JOIN S ON R.sid=S.sid;

CREATE TABLE D AS SELECT * FROM R FULL OUTER JOIN S ON R.sid=S.sid;
```

Tick the correct TRUE/FALSE boxes for each of the following statements. **TRUE** **FALSE**

12. $|A| < |B| < |C| < |D|$ ☐ ☐
13. $M \leq |A| \leq N$ ☐ ☐
14. $|B| = N$ ☐ ☐
15. $|D| = M$ ☐ ☐

C. Consider the Elevator Algorithm with the following parameters:

- Seek time formula: $1 + X/1000$ ms, where X is the number of cylinders traversed in transit;
- Average rotational delay is 4.75ms and the constant transfer time is 0.25ms.

We start at cylinder 0 and receive the following requests:

Request Name	Request (cylinder)	Time request arrives (ms)
A	5000	0
B	4000	8
C	3000	9
D	9000	9
E	8000	20

Tick the correct TRUE/FALSE boxes for each of the following statements on the relative order of processing times for the above requests. **TRUE** **FALSE**

16. C before D ☐ ☐
17. C before B ☐ ☐
18. D before E ☐ ☐
19. D before B ☐ ☐
20. Request D begins transferring data at 18.75 ms. ☐ ☐

Question 2.[Buffer Management] Maximum points: 16.

Consider a buffer manager that has a buffer pool large enough to hold 3 pages, and consider a file of 20 pages. Assume that the buffer pool is initially empty.

1. Is there a sequence of pin and unpin requests on at least 8 distinct pages such that the state of the buffer after completion of the sequence of requests is the same for all three replacement strategies CLOCK, LRU (Least Recently Used), FIFO (First In First Out)? If yes, write down the sequence (e.g., pin 1 unpin 1, pin 2, and so on), otherwise explain why not. (8 points)

Your request sequence is:

2. Give a sequence of pin and unpin requests such that the number of disk accesses is different for LRU and FIFO. For each policy, show the state of the buffer pool after each request. (8 points)

Your request sequence is:

Show the state of the buffer pool (i.e., state all its pages, its pinned pages, which page is at the head of the eviction queue) after requests in the following table:

Requests	FIFO	LRU

Question 3.[B$^+$-tree Indexing] Maximum points: 18.

Consider a B$^+$-tree of order 2 that is designed to index integer values. The index is initially empty (it holds no index entries). Assume that on overflow, you always split, and that on underflow, you always choose redistribution over merging if both options are available.

The sequence of integers 1 to 20 is first inserted into the B$^+$-tree and then all the prime numbers are deleted from the B$^+$-tree (from largest to smallest).

1. Draw the resulting B$^+$-tree after inserting 1 to 20. If you only show the index after *some* of the insertions, then please indicate exactly which integers in the sequence 1 to 20 you inserted. (10 points)

2. Draw the resulting B^+-tree after deleting 19, 17, 13, 11, 7, 5, 3, and 2. If you did not complete the previous task, then start with *any* B^+-tree index of order 2 with keys 1 to 20 at leaves. Indicate your initial B^+-tree for this task. Also, if you only show the index after *some* of the deletions, then please indicate which integers in the given sequence are deleted from your final index. (8 points)

Question 4.[Hash-based Indexing] Maximum points: 16.

Consider a hash index that uses the hash function $h(x) = x$ and in which each hash bucket can hold two data items. The index is initially empty (it holds no index entries) but consists of four buckets, and it initially uses the last two bits of the hash value $h(x)$. Consider the following sequence of insertions:

$$79 = 1\ 0\ 0\ 1\ 1\ 1\ 1_2$$
$$95 = 1\ 0\ 1\ 1\ 1\ 1\ 1_2$$
$$40 = 0\ 1\ 0\ 1\ 0\ 0\ 0_2$$
$$16 = 0\ 0\ 1\ 0\ 0\ 0\ 0_2$$
$$31 = 0\ 0\ 1\ 1\ 1\ 1\ 1_2$$
$$47 = 0\ 1\ 0\ 1\ 1\ 1\ 1_2$$
$$32 = 0\ 1\ 0\ 0\ 0\ 0\ 0_2$$
$$56 = 0\ 1\ 1\ 1\ 0\ 0\ 0_2$$

1. Consider that the index uses extendible hashing. Show the final state of the index after the insertions, including the directory, bucket entries, global and local depths in your diagram. If you only show the index after *some* of the insertions in the above sequence, then please state so. Assume that no overflow chains are allowed (so keep splitting until they are gone). (8 points)

2. Now consider that the index uses linear hashing. Initially, the "next" pointer points to the first bucket. A bucket split is triggered whenever an overflow page is created. Show the final state of the index after the insertions, including the bucket entries, index level, and the "next" pointer. If you only show the index after *some* of the insertions in the above sequence, then please state so. (8 points)

Question 5.[Query Languages] Maximum points: 30.

Consider the following relational database with information about applications, developers, and users (relation keys are underlined, types and column constraints are inlined in the definitions of relations):

```
Apps(aid (int), did (int) foreign key reference Developer,
      name (text), released (date))

Developers(did (int), name (text), city (text))

Users(uid (int), name (text), joined (date), city (text), device (text))

AppUsers(uid (int) foreign key references Users,
         aid (int) foreign key references Apps, downloaded (date))
```

Express each of the following natural language query in SQL and RA (relational algebra). Follow the exact notation that was specified for the Query Languages Homework.

1. Find the names of Developers who have created at least an App that has at least a User from London. (6 points)

SQL:

RA:

2. Find the names of Developers with the most used Apps. (6 points)

SQL:

RA:

3. How many Users have downloaded an App within a week of its release? Assume that the `date` type is closed under basic arithmetic operations and comparisons, in which days have the finest granularity. (6 points)

SQL:

RA:

4. Return the cities with the most number of Users. (6 points)

SQL:

RA:

5. How many developers are listed as users of their own application? (You may assume that names are unique.) (6 points)

SQL:

RA:

End of Exam Paper.

Extra Page for Scratch Work (to be discarded and not graded)

Extra Page for Scratch Work (to be discarded and not graded)

Extra Page for Scratch Work (to be discarded and not graded)