COMP 3005: Database Management Systems **(Due: Friday Dec. 10, 2021 (11:59 PM))**

Assignment #5

Instructor: Ahmed El-Roby *Name:* Ryan Lo, *ID:* 101117765

Instructions: Read all the instructions below carefully before you start working on the assignment, and before you make a submission.

- The accepted format for your submission is pdf.
- If you use the tex file, make sure you edit line 28 to add your name and ID. Only write your solution and do not change anything else in the tex file. If you do, you will be penalized.
- Late submissions are allowed for 24 hours after the deadline above with a penalty of 10% of the total grade of the assignment. Submissions after more than 24 are not allowed.

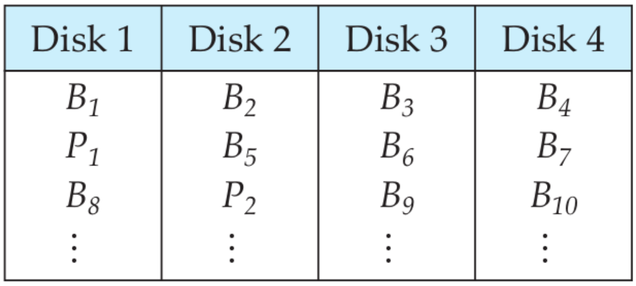
Q 1: (4 points)

RAID systems can support replacing failed disks without the system going offline. Which of the RAID levels better support this operation with the least amount of interference between the rebuild and ongoing disk accesses? Explain your answer.

RAID level 1 (reflecting) is the one which facilitates rebuilding of a failed disk with least interference with the on-going disk accesses. In the case where a drive fails, the data does not have to rebuild itself, since In RAID level 1 you have two copies of the data itself, the data just has to be copied to the replacement drive.

Q 2: (4 points)

Consider the following arrangement for four disks, where B_i is a data block, and P_i is the parity block for the 4 data blocks that precedes it. What problem will this arrangement cause?

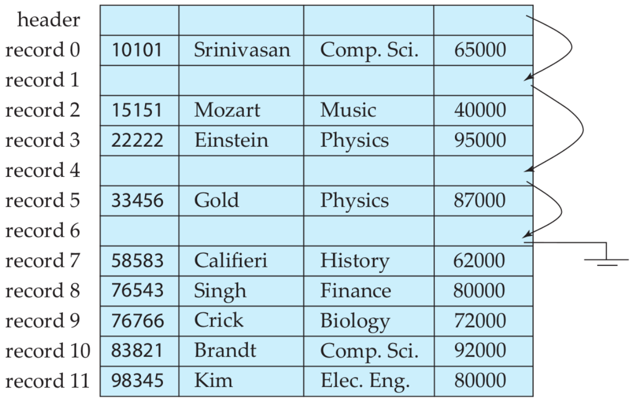


Disk 1	Disk 2	Disk 3	Disk 4
B_1	B_2	B_3	B_4
P_1	B_5	B_6	B_7
B_8	P_2	B_9	B_{10}
⋮	⋮	⋮	⋮

This arrangement of parity blocks is more prune to losing information if some blocks get corrupted. Since some of the rows in the disk arrangement does not include a parity block, blocks in those rows would be hard to recover if those blocks were to be corrupted.

Q 3: (8 points)

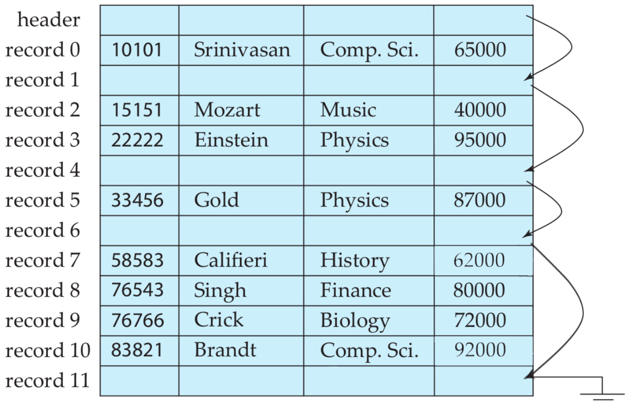
Consider the following file organization using free list.



header				
record 0	10101	Srinivasan	Comp. Sci.	65000
record 1				
record 2	15151	Mozart	Music	40000
record 3	22222	Einstein	Physics	95000
record 4				
record 5	33456	Gold	Physics	87000
record 6				
record 7	58583	Califieri	History	62000
record 8	76543	Singh	Finance	80000
record 9	76766	Crick	Biology	72000
record 10	83821	Brandt	Comp. Sci.	92000
record 11	98345	Kim	Elec. Eng.	80000

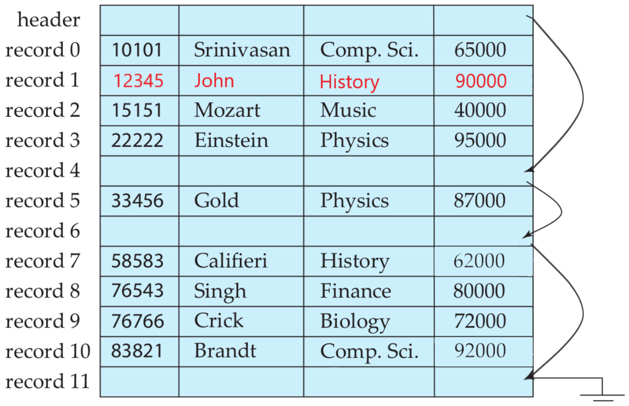
Show the structure of the file after each of the following operations (**they follow one another**):

(a) Delete record 11. (2 marks)



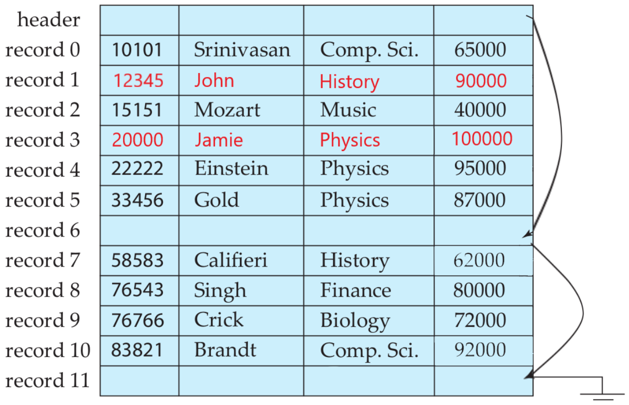
header				
record 0	10101	Srinivasan	Comp. Sci.	65000
record 1				
record 2	15151	Mozart	Music	40000
record 3	22222	Einstein	Physics	95000
record 4				
record 5	33456	Gold	Physics	87000
record 6				
record 7	58583	Califieri	History	62000
record 8	76543	Singh	Finance	80000
record 9	76766	Crick	Biology	72000
record 10	83821	Brandt	Comp. Sci.	92000
record 11				

(b) Insert (12345, John, History, 90000) (3 marks)



header				
record 0	10101	Srinivasan	Comp. Sci.	65000
record 1	12345	John	History	90000
record 2	15151	Mozart	Music	40000
record 3	22222	Einstein	Physics	95000
record 4				
record 5	33456	Gold	Physics	87000
record 6				
record 7	58583	Califieri	History	62000
record 8	76543	Singh	Finance	80000
record 9	76766	Crick	Biology	72000
record 10	83821	Brandt	Comp. Sci.	92000
record 11				

(c) Insert (20000, Jamie, Physics, 100000). (3 marks)



header				
record 0	10101	Srinivasan	Comp. Sci.	65000
record 1	12345	John	History	90000
record 2	15151	Mozart	Music	40000
record 3	20000	Jamie	Physics	100000
record 4	22222	Einstein	Physics	95000
record 5	33456	Gold	Physics	87000
record 6				
record 7	58583	Califieri	History	62000
record 8	76543	Singh	Finance	80000
record 9	76766	Crick	Biology	72000
record 10	83821	Brandt	Comp. Sci.	92000
record 11				

Q 4: (3 points)

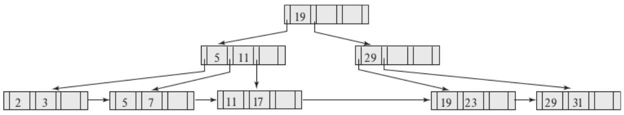
In variable-length record representation, the record starts with offset and length pairs of variable-size attributes, followed by fixed-size attributes, then the null bitmap, and finally the variable-size attributes. How can we improve this representation if our application is expected to store tables with large number of attributes, most of which are nulls?

This representation could be improved if expected to store tables with large number of attributes and most of which are nulls by keeping the null bitmaps updated.

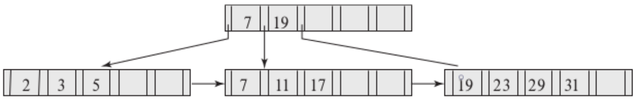
Q 5: (12 points)

Construct a B^+-tree for the following set of key values: (2, 3, 5, 7, 11, 17, 19, 23, 29, 31). The tree is initially empty and values are added one value at a time in ascending order. Consider the following values of n:

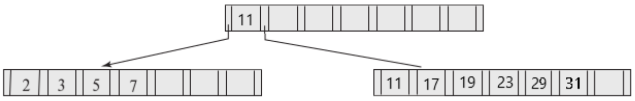
(a) $n = 4$. (4 points)



(b) $n = 6$. (4 points)

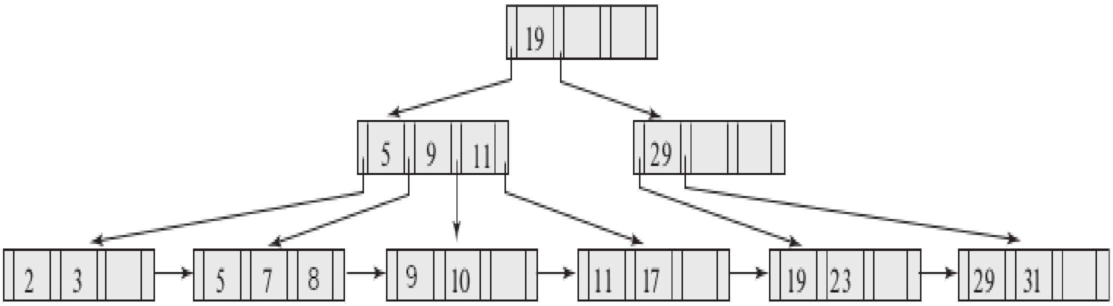


(c) $n = 8$. (4 points)

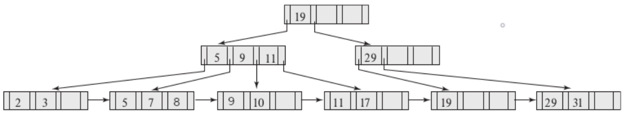


Q 6: (8 points)

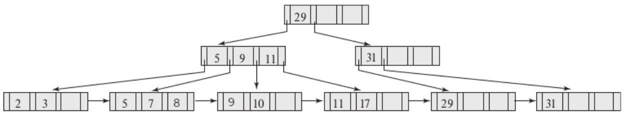
Consider the following B^+-tree with $n = 4$:



(a) Delete 23. (4 points)

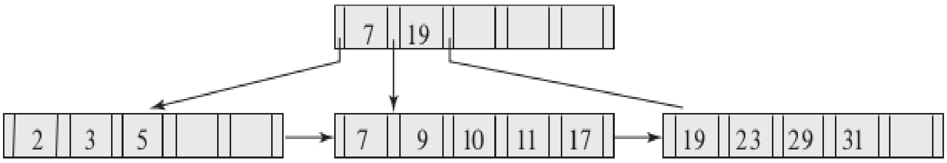


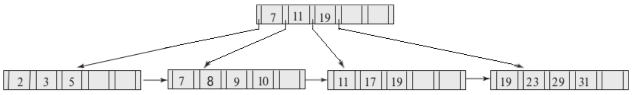
(b) Delete 19 (after the previous deletion in (a)). (4 points)



Q 7: (4 points)

Consider the following B^+-tree with $n = 6$: Insert 8 into this tree.


7
19
2
3
5
7
9
10
11
17
19
23
29
31


7
11
19
2
3
5
7
8
9
10
11
17
19
19
23
29
31