First Letter of your last name:_____

Your Full Name: _____

COM S 363: Exam 2

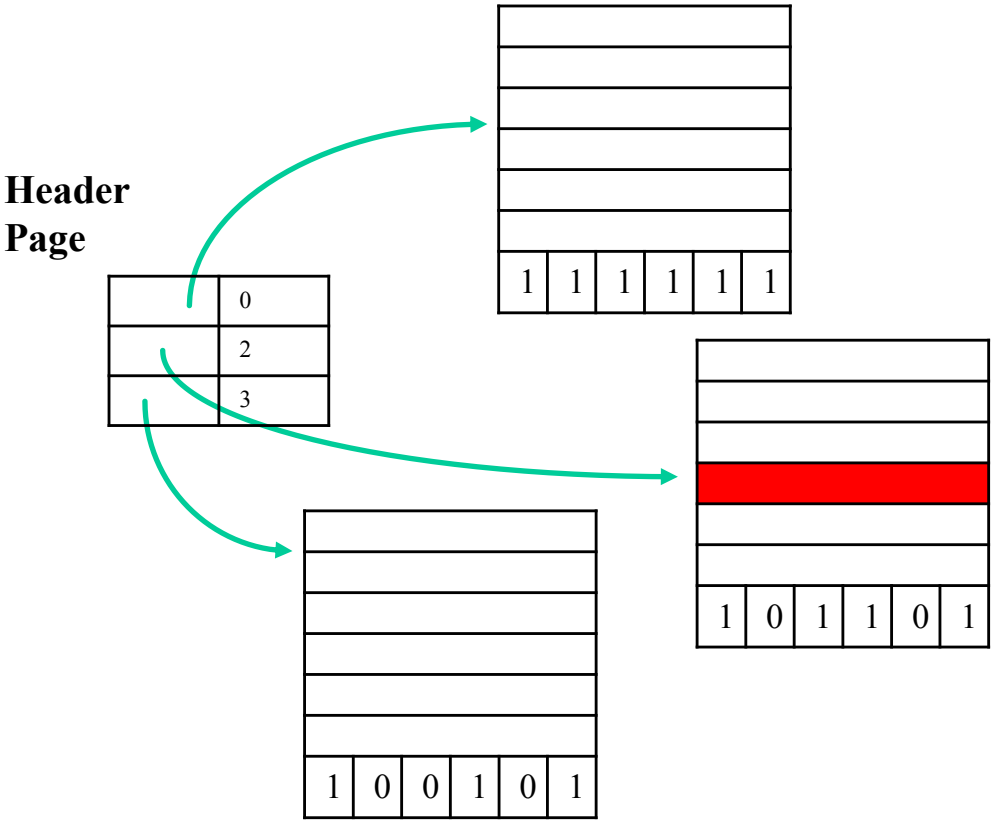
Time: 75 minutes

NOTES:

•DO NOT OPEN THIS EXAM UNTIL YOU ARE TOLD TO OPEN
•This is a closed book closed notes exam.
•Write your name and answer legibly; if the grader can't read, you receive no point.
•Attempt all problems. Write solutions on these sheets.
•Fill in your name now, but do not turn the page until the signal is given.

Problem	Max Points	Points
1	15	
2	15	
3	10	
4	10	
5	10	
6	15	
7	10	
8	15	
Total	100	

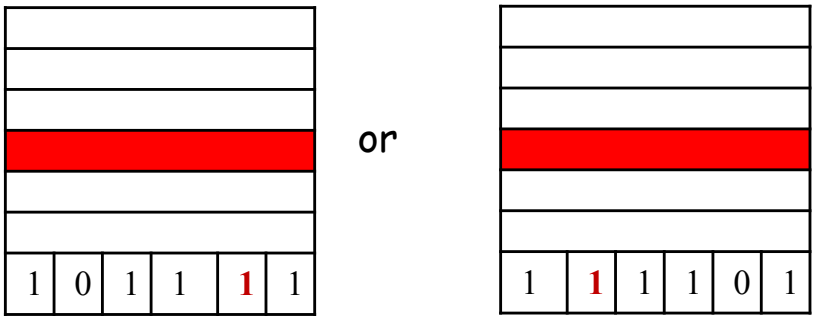
1. (15 points) Suppose a relation is stored in heap files showing below.



(1) (5 points) From the illustration, can you decide if the records are fix length or variable length, packed or unpacked?
(2) (5 points) Suppose a user inserts another record. What the result files should look like?
(3) (5 points) Suppose a user wants to delete a record. This record is located at the highlighted slot. What is the procedure?

1. The records are fixed length (2 points), unpacked (3 points)

2. Either one is fine



3. Locate the bit corresponding to the page, change the bit to 0.

2. (15 points) Consider a relation with this schema, Employee(ename:string, eid:integer, age:integer, salary:integer). Suppose ename is indexed by a sparse B^+-tree and eid is indexed by a dense and unclustered B^+-tree, as illustrated by the figure showed in the next page.

   Suppose the relation has 1,000 pages, and each page holds 4 records. Estimate the total number of pages that need to be read from the disk in order to answer the following queries. Explain your answer. For each query, you can assume the selection factor is 0.1, i.e., out of 1,000 * 4 records, 1,000 * 4 * 0.1 = 400 records will satisfy the query condition.

   a) (5 points) Find all employees whose salary is in between 80k and 100k.

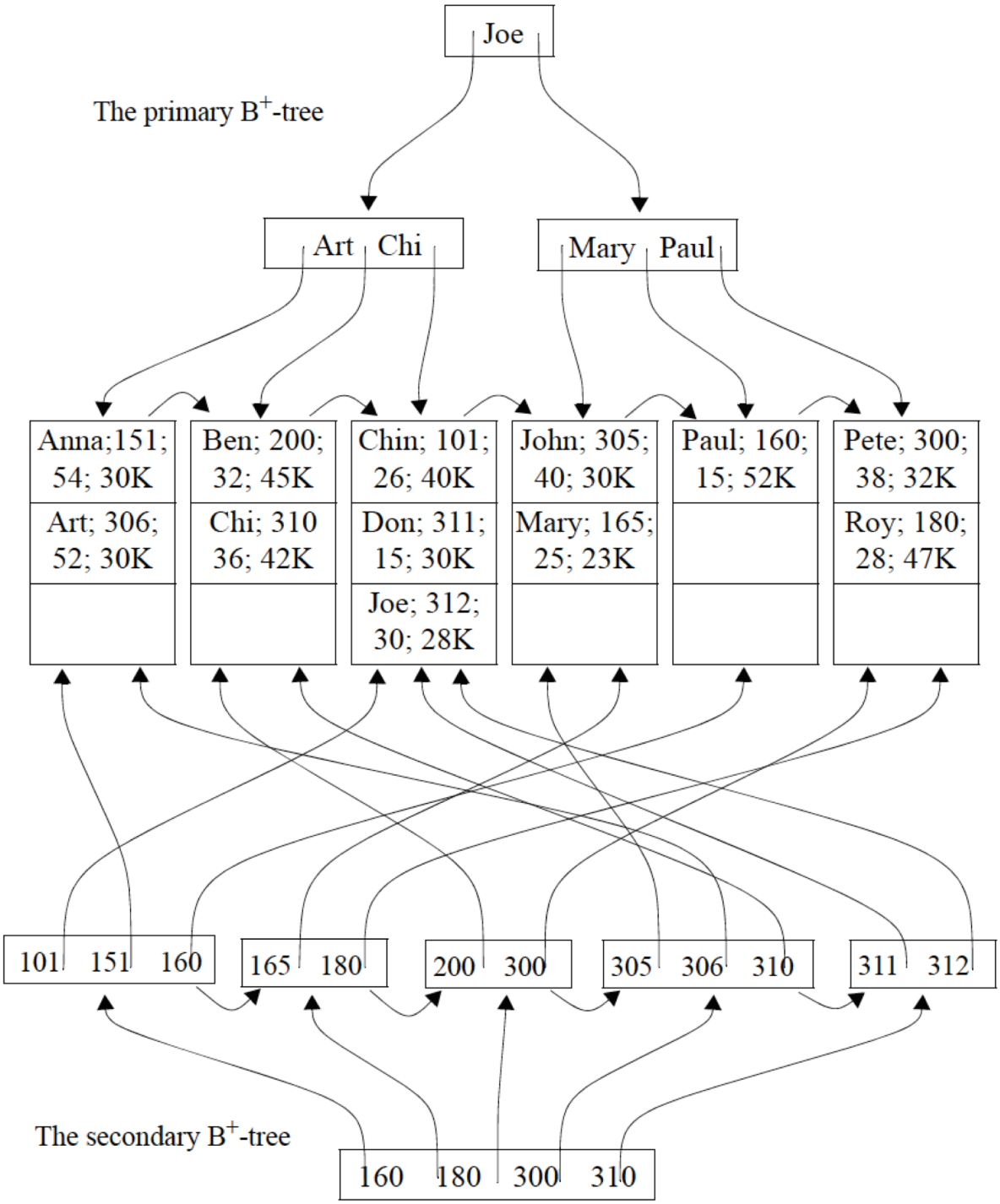
   Since there is no index on salary and it is not sorted (2 points), so sequential scan is needed to answer the query. The cost is 1,000 pages (3 points).

   b) (5 points) Find all employees whose name starts with a character that is in between "*A*" and "*C*".

   Since name is clustered, once located a record in a page, changes are all records in the page satisfy the query condition (2 points). The cost is 400 records/(4 records/page) = 100 pages (3 points)

   c) (5 points) Find all employees whose eid is in between 100 to 200.

   Since eid is not clustered, the records are scattered among the disk page. For each record, need to load a whole page. The cost is 400 pages.


Joe
The primary B+-tree
Art Chi
Mary Paul
Anna;151; 54; 30K
Art; 306; 52; 30K
Ben; 200; 32; 45K
Chi; 310 36; 42K
Chin; 101; 26; 40K
Don; 311; 15; 30K
Joe; 312; 30; 28K
John; 305; 40; 30K
Mary; 165; 25; 23K
Paul; 160; 15; 52K
Pete; 300; 38; 32K
Roy; 180; 28; 47K
101 151 160
165 180
200 300
305 306 310
311 312
The secondary B+-tree
160 180 300 310

3. (10 points) Derive the time cost of a block nested loops join of relations R and S given the following variables, which you may or may not use all of them. Ignore the CPU time costs. R is the outer relation and the results of the join are written back to the disk.

   |R|: Number of tuples in R
   |S|: Number of tuples in S
   M: Number of pages in R
   N: Number of pages in S
   B: Number of available memory in pages
   P: Number of pages that the join results occupy
   Cr: Time cost of reading a page from disk into memory in seconds
   Cw: Time cost of writing a page from memory into disk in seconds

   1 page reserved for S and 1 page for output

   So the total pages for R is B-1. The number of times of loading R to fill B-1 pages is Ceiling(M/R). The total read cost of R is M (3 points)

   For each fill, needs to scan S in whole. So the total read cost for S is Ceiling(M/R) * N (2 points)

   The total read cost for both R and S is M+Ceiling(M/R)*N (3 points)

   The total write cost is P (2 points)

   So the total time cost is

   [M+Ceiling(M/R)*N] * Cr + P * Cw

4. (10 points) Grace Hash Join consists of two phases: partition and join. Explain them in detail.

Let R and S be two relations to join

Partition (5 points):

- Divide R into R1, R2, ..., Rn with a hash function f1
- Divide S into S1, S2, ..., Sn with the same hash function f1

Join

- Join Ri and Si (2 points), with a different hash function f2 (3 points)

5. (10 points) Explain in detail how to sort 8 pages of data (e.g., numbers) with 3 pages of main memory. Your algorithm needs to be as efficient as possible.

1) Sort individual pages (4 points)

Load the first three pages of the data into the main memory, sort them. Let P1, P2, P3 be the sorted pages,

Load the next three pages of the data into the main memory, sort them. Let P4, P5, P6 be the sorted pages.

Load the next two pages of the data into the main memory, sort them. Let P7, P8 be the sorted pages.

2) Merge sorted pages (6 points)
Merge two sorted lists L1=[P1, P2, P3] and L2=[P4, P5, P6] with the 3 pages of the main memory.

- Use one page for L1, one page for L2, one page for output
- First load P1 and P4, merge them and write to the output
    - If P1 is finished, load P2, and then P3
    - If P4 is finished, load P5, and then P6
    - If the output buffer is full, write to the disk

Let the sorted 6 pages be P1, P2, P3, P4, P5, P6.

Merge two sorted lists L1=[P1, P2, P3, P4, P5, P6] and L2=[P7, P8] 3 pages of the main memory.

6. (15 points) Consider three relations Students(sid, sname, dob, addr), Courses (cid, cname), and Register (sid, cid, semester). Write a relational algebraic expression for each of the following queries.

   a) (5 points) Find the names of students who registered the course whose cid = "COMS363"

$PROJECT_{sname}(SELECT_{cid="COMS363"}(Register)\ JOIN\ Student)$

   a) (5 points) Find the names of students who reserved at least one course in semester "Spring2020"

$PROJECT_{sname}(SELECT_{Semester="Spring2016"}(Register)\ JOIN\ Student)$

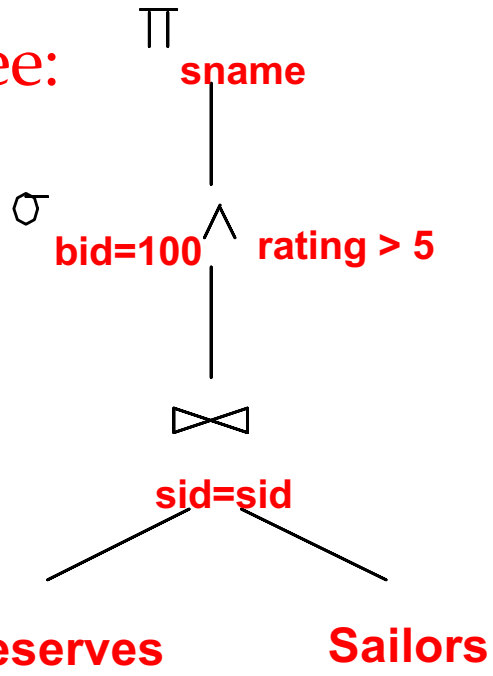
   a) (5 points) Find the names of students who have not registered any course

$PROJECT_{sname}(Students) - PROJECT_{sname}(Register\ JOIN\ Student)$

7. (10 points) Draw a relational algebraic tree that represents the following SQL statement.

```
SELECT  S.sname
FROM  Reserves R, Sailors S
WHERE  R.sid=S.sid AND
   R.bid=100 AND S.rating>5
```

RA Tree:

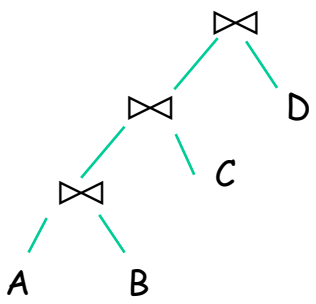


8. (15 points) Finding the best plan for join operations is prohibitively expensive. A practical solution is to avoid the worst plans by considering only the left-deep plans.

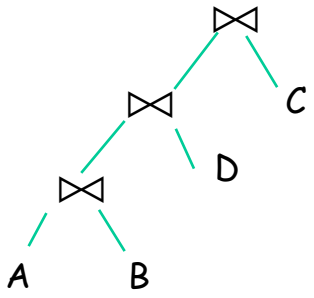
   a) (5 points) Explain the main benefit of left-deep plans.

Allow piplelining, without having to write intermediate results to harddrive.

   b) (10 points) Consider joining 4 relations A, B, C, and D. Suppose A ⋈ B is known to have the least disk I/Os among the joins of any two relations. Given this information, draw all possible left-deep join trees that will be considered by IBM System R.



5 points



5 points

For each correct tree, give 5 points. For each wrong tree, reduce 2 points, up to 10 points

This page is intentionally left blank. Use it as a scratch paper.