CSE 444 – Homework 1
Relational Algebra, Heap Files, and Buffer Manager

Name: Eric Boris

Question	Points	Score
1	10	
2	15	
3	25	
Total:	50	

1 Simple SQL and Relational Algebra Review

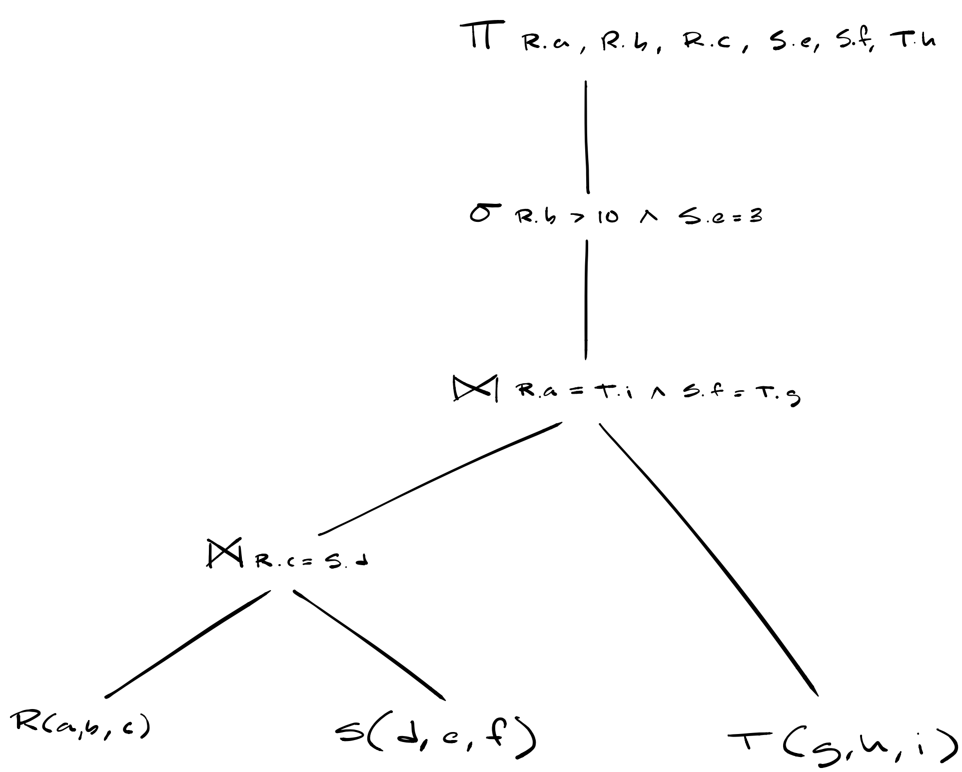
1. (10 points)

   When a user (or application) submits a SQL query to a relational DBMS, the SQL query takes the form of a *string*. Through a series of steps, the DBMS translates this string into a logical query plan. In this exercise, we practice manually translating two simple SQL queries into logical query plans.

   Consider relations `R(a,b,c)`, `S(d,e,f)`, and `T(g,h,i)`.

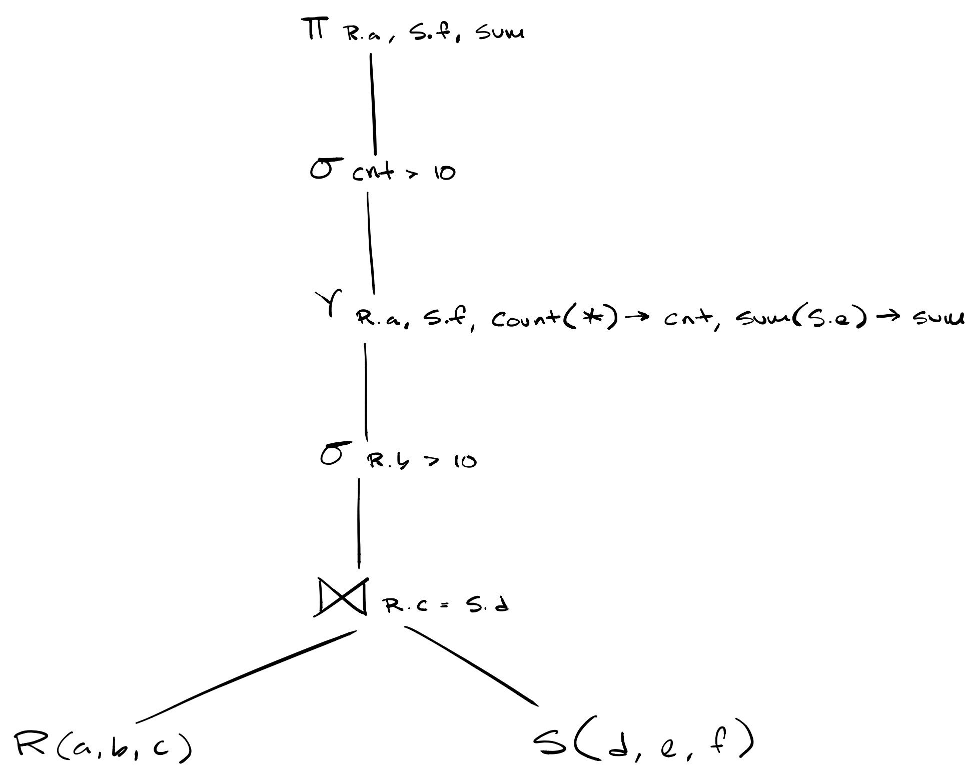
   (a) (5 points) Write a Relational Algebra expression in the form of a *logical query plan* that is equivalent to the SQL query below.

```
SELECT R.a, R.b, R.c, S.e, S.f, T.h
FROM R, S, T
WHERE R.c = S.d
AND S.f=T.g
AND T.i=R.a
AND R.b > 10
AND S.e = 3
```



(b) (5 points) Write a Relational Algebra expression in the form of a *logical query plan* that is equivalent to the SQL query below.

```
SELECT R.a, S.f, sum(S.e) as sum
FROM R, S
WHERE R.c = S.d
AND R.b > 10
GROUP BY R.a, S.f
HAVING count(*) > 10
```



2 Data Storage: Heap Files

2. (15 points)

   Consider a relation S with the following schema:

   `S(a int, b char(10), c char(20))`

   Consider the following small instance of S:

a	b	c
1	Orange	First
2	Red	Second
3	Blue	Third

Assume that S is stored in a Heap file on disk. Assume also a page size of 8KB (8192 bytes) and 4-byte integers.

(a) (5 points) Assume the same on-disk representation as used in SimpleDB (please refer to the lab1 instructions), draw a *schematic* representation of the Heap file **page** on disk storing the S instance. For examples of what to draw, see lecture 4, slides on "Page Formats", but check the SimpleDB documentation to figure out the format of the page that you should draw! Fill in the fields to show the three tuples on the page. Show where the empty space is on the page (if any). Show padding/unused space (if any). Do not worry about endianness nor about getting all proportions right. If you want, you may specify byte offsets for the records but you do NOT have to show that information.

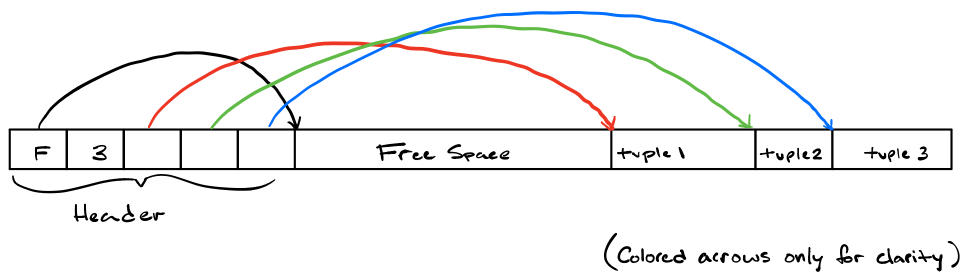
- 4 B + 10 B + 20 B = 34 B per tuple
- 3 tuples · 34 B per tuple = 102 B to store S
- Each tuple in S is fixed length
- Hence,

Header	tuple 1	tuple 2	tuple 3	Free Space	# of Records

(b) (5 points) The size of each S tuple in bytes is 4+10+20 or 34 bytes (this is different form SimpleDB where all strings are 128+4 bytes in length). How many S tuples fit on one page? What is the size of the page *header* in bytes? How many pages are necessary to hold an instance of S with 1000 tuples (remember the space necessary for the page header)?

- Tuple Size = 34 B/Page
- Page Size = 8 kB = 8192 B
- Tuples Per Page $= \left\lfloor \frac{\text{PageSize} \cdot 8}{\text{TupleSize} \cdot 8 + 1} \right\rfloor = \left\lfloor \frac{8192\text{ B} \cdot 8}{34\text{ B/Page} \cdot 8 + 1} \right\rfloor = 240$ tuples per Page
- Page Header $= \left\lceil \frac{\text{TuplesPerPage}}{8} \right\rceil = \left\lceil \frac{240\text{ Tuples/Page}}{8} \right\rceil = 30$ B
- PagesToHold1000TuplesOfS $= \left\lceil \frac{1000\text{ TuplesOfS}}{\text{TuplesPerPage}} \right\rceil = \left\lceil \frac{1000\text{ TuplesOfS}}{240\text{ Tuples/Page}} \right\rceil = 5$ pages

(c) (5 points) Now imagine that we wanted to extend SimpleDB to support *variable-length* tuples, draw a *schematic* represenation of the modified Heap file page on disk storing the S instance (the one with the three tuples). Assume the strings are now variable lengths (VARCHAR). Do not worry about the detailed representation of the records themselves, though.



3 Buffer Manager and Simple Query Execution

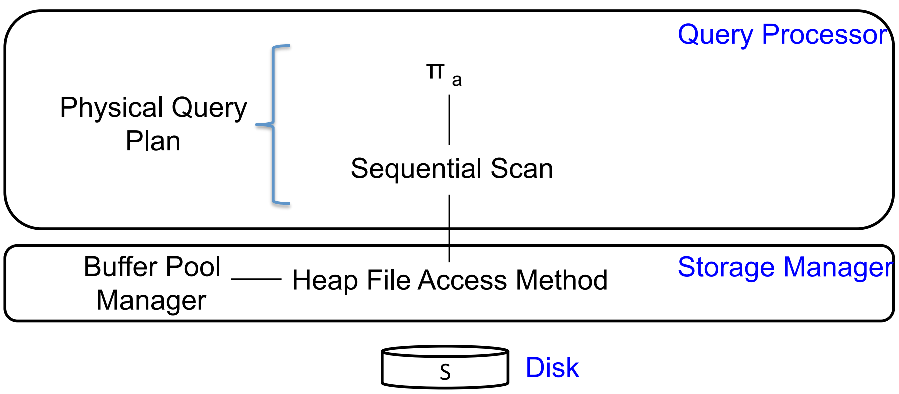
3. (25 points)

Consider a buffer pool large enough to hold 10 pages. The buffer pool is initially empty. Consider also a buffer pool manager that uses an LRU page replacement policy.

Consider the same S relation as in the earlier question and consider the following SQL query:

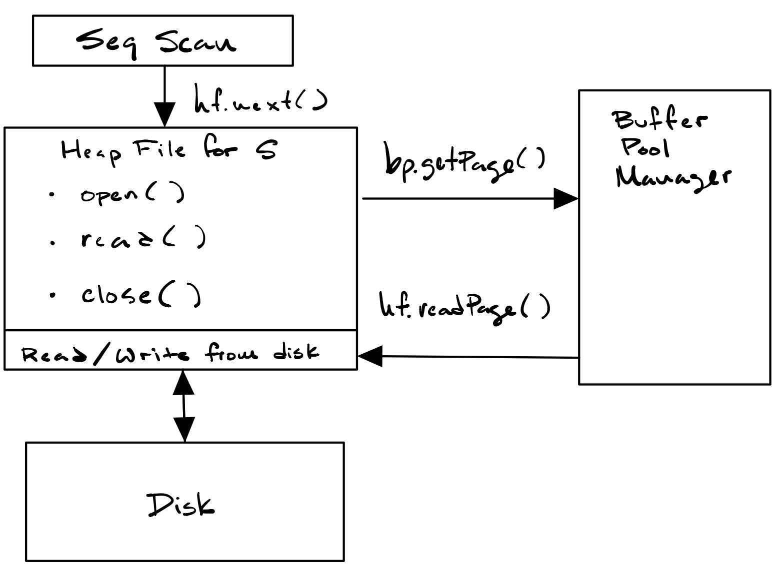
```
SELECT S.a
FROM S
```

A simple *physical query plan* for this query is the following. We show the query plan in the context of the relevant DBMS architecture components:



To execute the query, the system will call `open()` and then `next()` on the project operator. We ignore `hasNext()` in this exercise.

Consider that relation S is stored in a heap file on disk.



(a) (5 points) Explain how the execution of this query will proceed as the system calls `open()` and then `next()` on the topmost, project operator. You only need to describe what happens on the call to`open()` and then on the first call to `next()`. You do not need to describe subsequent calls to `next()`.

Your explanation should describe the control flow (who calls whom and when) between (1) the project operator, (2) the sequential scan operator, (3) the heap file access method, and (4) the buffer pool manager. Similar to the SimpleDB design, consider that the buffer pool manager will call the heap file to actually read a page from disk.

To process query Select S.a From S:

open() is called on Project Operator (PO)

PO calls open() on Sequential Scan Operator (SSO)

SSO calls open on Heap File (HF) access method

PO calls next() on SSO

SSO calls next() on HF access method

HF access method calls getPage() on Buffer Pool Manager (BPM)

if requested Page (P) is not in Buffer Pool (BP)

BPM calls readPage() on HF

P is returned to BP

P is returned to SSO

SSO returns Tuple to PO

(b) (5 points) What will be the content of the buffer pool after the first `next()` call on the project operator returns a tuple.

BP contains first page of Relation S that contains the tuple requested of the PO.

(c) (5 points) What will be the content of the buffer pool after the second `next()` call on the project operator returns a tuple.

Depends. If the first and second tuple fit on the same page then the Buffer Pool will have one page with both tuples. Otherwise, it will have two pages with one tuple a piece. This prevents against needing to access main memory if next is called again.

(d) (5 points) Assuming S contains 10,000 tuples, what will be the content of the buffer pool after 1000 next() calls on the project operator.

BP contains the first 5 Pages. i.e. the last 1000 Tuples filling 5 pages as computed in 2b.

(e) (5 points) Assuming S contains 10,000 tuples, what will be the content of the buffer pool after the query completes?

10,000 Tuples fill $\left\lceil \frac{10000}{240} \right\rceil$ = 42 Pages. The BP holds 10 Pages maximum. BP contains only the last 10 pages, including the contents of the last Page of Relation S.