Exercise 3

Time tip: Roughly 45sec to 1min per 1pt
### Ratings / R

<table>
<thead>
<tr>
<th>RatingID</th>
<th>Stars</th>
<th>RateDate</th>
<th>UID</th>
<th>MID</th>
</tr>
</thead>
<tbody>
<tr>
<td>7254</td>
<td>4.5</td>
<td>12/15/19</td>
<td>839</td>
<td>123</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
</tbody>
</table>

### Users / U

<table>
<thead>
<tr>
<th>UID</th>
<th>Name</th>
<th>Age</th>
<th>JoinDate</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>Alvarez</td>
<td>39</td>
<td>11/02/14</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
</tbody>
</table>

### Movies / M

<table>
<thead>
<tr>
<th>MID</th>
<th>Name</th>
<th>Year</th>
<th>Director</th>
</tr>
</thead>
<tbody>
<tr>
<td>492</td>
<td>Parasite</td>
<td>2019</td>
<td>Bong Joon-Ho</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
</tbody>
</table>

**Common Info: Netflix Schema**
Q1) [2 x 2pts] Suppose you are given that NTtuples(R) = 1 billion. What is the selectivity of a selection predicate on R if it results in the following number of tuples in the output?

A. 50
B. 50 million
Q2) [4 x 6pts] Name 3 indexes each (at least 1 each of B+ tree and hash index) that match the following select operations.

A. SELECT * FROM R WHERE UID = 32 AND Stars >= 4
B. SELECT * FROM M WHERE NOT (Year < 2010 OR Year >= 2020)
C. SELECT * FROM U WHERE NAME LIKE ‘A%’ AND Age >=30 OR NAME LIKE ‘B%’ AND Age >= 30
D. SELECT * FROM R WHERE NOT (UID <> 8 OR MID = 9)
Q3) [6 x 4pts] You are given the following statistics of the number of pages in each relation and the allotted buffer memory size in pages $B$. Suppose all attributes are 8 bytes long, except U.Name, M.Name, and M.Director, each of is 40 bytes. Assume UID and MID are uniformly distributed in R. Ignore output write costs. What is the I/O cost (in number of pages) of the following operators?

\[(N_R, N_U, N_M, B) = (20000, 4000, 1000, 500)\]

A. Hash-based project of R on to RateDate
B. Sort-based project of R on to RateDate
C. Hash-based project of U on to Name
D. Sort-based project of U on to Name
E. Hash-based project of M on to Director
F. Sort-based project of M on to Director
Exercise

Q4) [6 x 4pts] You are given the following statistics of the number of pages in each relation and the allotted buffer memory size in pages $B$. Suppose all attributes are 8 bytes long, except $U$.Name, $M$.Name, and $M$.Director, each of is 40 bytes. Assume UID and MID are uniformly distributed in $R$. Ignore output write costs. What is the I/O cost (in number of pages) of the following operators?

$$(N_R, N_U, N_M, B) = (40000, 5000, 500, 800)$$

A. BNLJ for natural join of $U$ and $R$
B. SMJ for natural join of $U$ and $R$
C. HJ for natural join of $U$ and $R$
D. BNLJ for natural join of $M$ and $R$
E. SMJ for natural join of $M$ and $R$
F. HJ for natural join of $M$ and $R$
Exercise

Q5) [2 x 6pts] You are given the following statistics of the number of pages in each relation. Suppose all attributes are 8 bytes long, except U.Name, M.Name, and M.Director, each of is 40 bytes. Assume UID and MID are uniformly distributed in R. Ignore output write costs. Suppose the allowed buffer memory size in pages is B. Identify a value of B for which BNLJ will have a lower I/O cost than HJ or SMJ for the following join.

\[(N_R, N_U, N_M) = (50000, 10000, 2000)\]

A. Natural join of U and R
B. Natural join of M and R
Q6) [8pts] You have 4B + 1 buffer frames for the following hash join of R and U, which have sizes 16*B*r and 4*B*u pages, respectively. You also have 16Br >> 4Bu. The buffer pool is initially empty. You are also given that 3Fr = 4B − 1, where F is the hash table fudge factor. B is in the millions.

The distribution of UID in R is such that after the first hash partitioning phase, we get exactly 4B partitions each of R and U. Each partition of U is u pages long, but the partitions of R have differing lengths.

Suppose R gets partitioned as follows: B partitions of length r pages each and 3B partitions of length 5r pages each. What is the I/O cost of the above join using the regular hash join discussed in class? Exclude the cost of writing the output. Assume that perfect uniform splitting occurs during the recursive repartitioning and that we do not need to recurse more than once.