CSE 132C
Database System Implementation

Arun Kumar

Review Exercise
## Common Info: Netflix Schema

### 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>UName</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>

---

---
• All attributes in the given Netflix database are of fixed length and 8 bytes long, except for Director, Name, and UName, which are 40 bytes long.
• The number of tuples in R, U, and M are 10 billion, 50 million, and 1 million, respectively.
• All tables are stored as heap files with the pages storing the fixed-length records in *unpacked* layout.
• The foreign keys have no dangling references.
• In the given instance, every UID in U and every MID in M arises at least once in R.
Exercise

Q1) Page size is 4096 bytes.

A. [9pts] What is the size of each table in pages? Round to the nearest thousand.

B. [4pts] Which 2-table key-foreign key join has the larger output in terms of cardinality? In terms of arity? In terms of size?

C. [5pts] What is the size (in pages) of the output of the 3-table star join in pages? Round to the nearest thousand.

D. [8pts] What is the largest possible size (in pages) of the output of the following query? Round to the nearest thousand.

\[ \pi_{\text{UID}}(U) \times \pi_{\text{MID}}(M) \ominus \pi_{\text{UID},\text{MID}}(R) \]
Q2) Page size is 4096 bytes. You have 1 million buffer frames.

A. [6pts] What is the lowest possible I/O cost (in pages) of sorting R on Stars using any of the optimizations discussed in the lectures? Round to the nearest thousand.

B. [10pts] Suppose page pointers are 8 bytes long and record IDs are 12 bytes long. What is the lowest possible size (in pages) of an extendible hash index built on Users with IndexKey UID using the AltRID alternative? Assume the hash function enables uniform hashing without skews. Round to the nearest thousand. (Hint: Count the number of slots first).
Exercise

Q3) Page size is 4096 bytes. You have 2 million buffer frames for each query (independent of the others). What is the lowest possible I/O cost (in pages, rounded to the nearest thousand) of each of the following queries regardless of the data distributions using only the operator implementations discussed in the lectures? Include the cost of writing the output in C and D.

A. [6pts] \( \pi_{\text{JoinDate}}(U) \)

B. [8pts] \( \gamma_{\text{Director}, \text{COUNT}(\ast)}(M \bowtie R) \)

C. [10pts] \( R \bowtie U \bowtie M \)

D. [10pts] \( \pi_{\text{UID}}(U) \times \pi_{\text{MID}}(M) \)
Exercise

Q4) Are these pairs of queries equivalent? If you say yes, explain why. If you say no, provide a counterexample.

A. [5pts] \( \pi_{UID}(U) \times \pi_{MID}(M) \)
\( \pi_{UID, MID}(U \times M) \)

B. [6pts] \( \gamma_{Year, \text{MAX}(Stars)}(R \Join M) \)
\( \gamma_{Year, \text{MAX}(Stars)}(M \Join \gamma_{MID, \text{MAX}(Stars)}(R)) \)

C. [6pts] \( \gamma_{\text{COUNT}(\ast)}(R \Join U \Join M) \)
\( \gamma_{\text{COUNT}(\ast)}(\pi_{UID, MID}(R) \Join U \Join M) \)