1 Relational Algebra [20 Points]

Since user-centered recommendation systems are gaining unprecedented popularity, it is high
time to create a "CMUQ venue recommendation system," which will record the preferences
of current students for various recreational places in Education City. The ultimate goal is to
recommend venues to new students based on the similarity of their preferences to more senior
students. We start with the following relation schemas:

\[
\text{PLACES (category: string, bname: string, popularity: integer)}
\]

- \text{category}: category of the venue (e.g., "Cafe", "Library").
- \text{bname}: building where the venue is located (e.g., "LAS", "CMUQ").
- \text{popularity}: popularity score of the place (between 1 and 10).

\[
\text{STUDENT_LIKES (sname: string, category: string)}
\]

- \text{sname}: student’s full name (e.g., John Brown). You may assume that every
  name in the database is unique.
- \text{category}: category of the venue of interest (e.g., "Cafe," "Library").

Tables 1 and 2 present instances of \text{PLACES} and \text{STUDENT_LIKES} respectively. Note
that a place is identified by its building and category.

<table>
<thead>
<tr>
<th>Category</th>
<th>Bname</th>
<th>Popularity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Café</td>
<td>Recreation Center</td>
<td>7</td>
</tr>
<tr>
<td>Library</td>
<td>LAS</td>
<td>5</td>
</tr>
<tr>
<td>Gym</td>
<td>Recreation Center</td>
<td>10</td>
</tr>
<tr>
<td>Library</td>
<td>CMUQ</td>
<td>8</td>
</tr>
<tr>
<td>Indoor Activity</td>
<td>Al-Awsaj</td>
<td>10</td>
</tr>
<tr>
<td>Gym</td>
<td>CMUQ</td>
<td>3</td>
</tr>
<tr>
<td>Bookstore</td>
<td>Student Center</td>
<td>5</td>
</tr>
<tr>
<td>Movies</td>
<td>Student Center</td>
<td>4</td>
</tr>
<tr>
<td>Café</td>
<td>CMUQ</td>
<td>4</td>
</tr>
<tr>
<td>Football Field</td>
<td>Al-Awsaj</td>
<td>10</td>
</tr>
</tbody>
</table>

\text{Table 1: A PLACES relation instance P. A place is identified by its category and building, and has a popularity score.}
<table>
<thead>
<tr>
<th>sname</th>
<th>category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oliver Stark</td>
<td>Café</td>
</tr>
<tr>
<td>Noah Morgan</td>
<td>Gym</td>
</tr>
<tr>
<td>Fabiana Dallas</td>
<td>Gym</td>
</tr>
<tr>
<td>Timmy Jones</td>
<td>Library</td>
</tr>
<tr>
<td>Timmy Jones</td>
<td>Indoor Activity</td>
</tr>
<tr>
<td>Adrian McCarthy</td>
<td>Library</td>
</tr>
<tr>
<td>Adam Harrison</td>
<td>Café</td>
</tr>
</tbody>
</table>

Table 2: A Student_Likes relation instance S.

1. What will the following relational algebra expression output? [3 Points]

\[ \pi_{\text{bname}}(\sigma_{\text{popularity}>5}(\text{PLACES})) \]

(a) The names of buildings in which all corresponding places have popularity scores greater than 5.
(b) The names of buildings whose corresponding places have at most one category with a popularity score greater than 5.
(c) The names of buildings whose corresponding places have only one category with a popularity score greater than 5.
(d) None of the above (hence, write-down your answer).

2. Write a relational algebra expression which returns all student names who have exactly one category of interest. [10 Points]

3. Consider the following relational algebra expression and a relation instance of PLACES as shown in Table 1: [7 Points]

\[ \pi_{\text{category,bname}}(\text{PLACES}) \div (\pi_{\text{bname}}[\sigma_{\text{category}}=\text{"Gym"}](\text{PLACES})) \]

(i) What does the above expression entail (in English)? A possible answer is: "Select all (building, category) combinations that do not have a Gym." [3 Points]

(ii) How many field(s) are (is) returned in the output? [1 Point]

(iii) What are (is) the field(s) returned in the output? [1 Point]

(iv) How many record(s) are (is) returned in the output? [1 Point]

(v) What are (is) the record(s) returned in the output? [1 Point]
2 Tuple Relational Calculus (TRC) [20 Points]

For this question, we will consider the same relation schemas used in Question 1.

1. Write a TRC expression that selects all the records from STUDENT_LIKES who like the "Gym". [3 Points]

2. Consider the following TRC expression and an instance S of STUDENT_LIKES as shown in Table 2: [7 Points]

\[
\{ S \mid S \in \text{STUDENT\_LIKES} \land \exists S' \in \text{STUDENT\_LIKES} (S'.\text{category} = S.\text{category} \land S'.\text{name} = 'AdrianMcCarthy' \land S.\text{name} \neq 'AdrianMcCarthy') \}
\]

(a) What does the above expression entail (in English)? A possible answer is: "Select (Adrian McCarthy, category) if he likes more than one category". [3 Points]

(b) How many field(s) are (is) returned in the output? [1 Point]

(c) What are (is) the field(s) returned in the output? [1 Point]

(d) How many record(s) are (is) returned in the output? [1 Point]

(e) What are (is) the record(s) returned in the output? [1 Point]

3. Write a TRC expression that returns all categories that are liked by at most two Students. [10 Points]
3 Relational Algebra, TRC and SQL [60 Points]

Moving from CMUQ’s recommendation system, we shall now consider a simplified version of a
known social application, Twitter! At a high-level, Twitter works as follows:

- Users post tweets that are short pieces of text.
- Users may tag their tweets with zero or more tags of their own choices. A tag must begin
  with the hash tag sign '#'. For example, a user tweeting about the Database Applications
  course may decide to tag the tweet with #DBApps #socool.
- Users may follow zero or more other users. The tweets of the former are visible to the
  latter.

Given the above Twitter’s description, we define the following relation schemas:

<table>
<thead>
<tr>
<th>Relation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Users</td>
<td>(uname, city, street) - you may assume that uname is unique</td>
</tr>
<tr>
<td>Follows</td>
<td>(uname1, uname2) - user with uname1 follows user with uname2</td>
</tr>
<tr>
<td>Tweets</td>
<td>(tid, t_title, t_text) - tweet with tid has title t_title &amp; content t_text</td>
</tr>
<tr>
<td>UserTweets</td>
<td>(uname, tid, ts) - user uname posted a tweet with tid at time ts</td>
</tr>
<tr>
<td>TweetTags</td>
<td>(tid, tag) - tweet with tid has tag in its list of tags.</td>
</tr>
</tbody>
</table>

Now, we would like to extract some useful information from the database and we leave this
job to our database expert (you!). For each of the following questions, write (a) a relational
algebra expression, (b) a TRC expression, and (c) an SQL query. State the reason clearly if an
expression and/or query cannot be expressed.

1. Find all users (uname) who posted a tweet with tag "#DBApps". [10 Points]
2. Find all distinct tags ever used since the launch of Twitter. [10 Points]
3. Considering a particular user 'Donald Trump’, find all distinct tags of all tweets by users
   whom Donald follows. In other words, find Donald’s reading interests. [10 Points]
4. Find all users (uname and city) who follow users who follow user 'Donald Trump'.
   [10 Points]
5. Find all users who read about iPhone X before its launch (i.e., before Nov. 3, 2017).
   [10 Points]
6. Find all users (uname and city) who follow at least everyone that user 'Donald Trump’
   follows. [10 Points]