Information and Database Management Systems I (CIS 4301)  
(Spring 2017)  
Instructor: Dr. Markus Schneider  
TA: Yang Peng  

Homework 2 Solutions

Name:  
UFID:  
Email Address:  

Pledge (Must be signed according to UF Honor Code)  
On my honor, I have neither given nor received unauthorized aid in doing this assignment.

_______________________________________________  
Signature  

For scoring use only:

<table>
<thead>
<tr>
<th>Exercise</th>
<th>Maximum</th>
<th>Received</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exercise 1</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>Exercise 2</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>Exercise 3</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>Exercise 4</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>
Exercise 1 (Knowledge Questions) [25 points]

1. [5 points] What is a DBMS? What are the processes that a DBMS facilitates?
   A DBMS is a collection of programs that enables users to create and maintain a database. The DBMS is a general-purpose software system that facilitates the processes of defining, constructing, manipulating, and sharing databases among various users and applications.

2. Consider a database of employees in which we need to record information about employees’ addresses. Name at least two conditions that would cause you to make “address” an entity set of its own rather than an attribute of the employee entity set.
   - An employee may have more than one address and all of them are supposed to be stored in the database.
   - Attribute address is composed of street, city, state, etc. Components of an address may also be interesting to some queries.
   - Address of an employee may be shared by another entity.

3. [5 points] What is the difference of generalization and aggregation?
   Generalization defines an is-a-kind of relationship in which one class shares its structure and/or behavior with one or more other classes. Aggregation is a special relationship set which associates each superior entity set with several subordinate entity sets.

4. [5 points] List the basic operations of the Relational Algebra with their name and their correct symbolic notation.
   The basic operations are $\cup$ (union), $-$ (difference), $\times$ (Cartesian product), $\pi$ (projection), and $\sigma$ (selection).

5. [5 points] What is a key? Should a key contain only a single value? If yes, explain why. If not, give an example.
   A key is a minimal set of attributes whose values uniquely characterize the associated entity among all entities of its type. A key need not to contain only one single value, it can be a set of attributes. For example: \{company\_brand, sequence\_number\} can be a key to characterize a laptop.
Exercise 2 (Relational Algebra) [20 Points]

Consider the following database schema:

**Flights** (flightNumber: string, travelFrom: string, travelTo: string, distance: integer, departs: time)

**Aircraft** (planeId: string, planeName: string, range: integer)

**Pilots** (employeeId: string, planeId: string)

**Employees** (employeeId: string, employeeName: string, salary: integer)

Express the following colloquial queries in Relational Algebra:

1. [5 points] Find the employee Id’s of pilots who can operate the aircrafts of types "Boeing 747" and "Boeing 777".

   \[ \pi_{employeeId} (\sigma_{planeName = 'Boeing 747'} ('Aircraft' \bowtie 'Pilots')) \cap \pi_{employeeId} (\sigma_{planeName = 'Boeing 777'} ('Aircraft' \bowtie 'Pilots')) \]

2. [5 points] Find the id of the pilot, or pilots, with the highest salary.

   \[ \pi_{employeeId} ('Pilots') \bowtie (\pi_{employeeId} ('Employees') - \pi_{e2.employeeId} (\sigma_{e1.salary > e2.salary} (\rho_{e1} ('Employees') \bowtie \rho_{e2} ('Employees')))) \]

3. [5 points] Find the airplane Id's of aircrafts that cannot fly non-stop from ATL to JFK.

   \[ \pi_{planeId} (\sigma_{range < distance} ('Aircraft' \bowtie 'Flights')) \]

4. [5 points] Find the names of pilots who can operate planes with a range greater than or equal to 1,500 miles but cannot operate "Boeing 747" and "Boeing 777" aircrafts.

   \[ \pi_{employeeName} ('Employees' \bowtie (\pi_{employeeId} (\sigma_{range \geq 1500} ('Aircraft' \bowtie 'Pilots'))) - (\pi_{employeeId} (\sigma_{planeName = 'Boeing 747'} ('Aircraft' \bowtie 'Pilots'))) \cap \pi_{employeeId} (\sigma_{planeName = 'Boeing 777'} ('Aircraft' \bowtie 'Pilots'))) ) \]
Exercise 3 (Relational Algebra) [30 points]

Consider the following database schema for a simplified Twitter database (primary keys are underlined):

- **Person(pname, city, ttext)** – Assume the **pname** is unique
- **Follows(pname1, pname2)** – Person **pname1** follows person **pname2**
- **Tweets(tid, title, ttext)** – Tweet with tid has title **title** and text **ttext**
- **PersonTweets(pname, tid, ts)** – Person **pname** posted tweet tid at timestamp **ts**
- **TweetTag(tid, tagname)** – Tweet tid had tagname in its list of tags

Users post ‘tweets’, that is short pieces of text. They may tag their tweets with zero or more tags of their own choice. For example, a user tweeting about the G20 summit may decide to use the tag ‘G20’. A user ‘u’ may follow zero or more other users, which means that their ‘tweets’ are visible to user ‘u’ when he/she logs in.

For your convenience, you can use the follow abbreviations: P for Person, F for Follows, T for Tweets, PT for PersonTweets, TT for TweetTag. Express the following queries in relational algebra assuming set semantics.

1. [5 points] Find all the people (pname) who posted a tweet with tag ‘Obama’.
   
   **Solution:** \(\pi_{pname}(\sigma_{tagname='Obama'}^{\text{TT}} \land \text{T.tid} = \text{PT.tid} \land \text{PT.pname} = \text{PT.pname})\)

2. [5 points] Find all the tags ‘Larry Page’ uses in his tweets. (i.e. Larry’s tweeting interests)
   
   **Solution:** \(\pi_{tagname}(\sigma_{pname='LarryPage'}^{\text{PT.tid} = \text{TT.tid} \land \text{PT.tid}} \land \text{PT.pname} = \text{PT.pname})\)

3. [5 points] Find all the tags ‘Larry Page’ reads in the tweets of the people he follows. (i.e. Larry’s reading interests)
   
   **Solution:** \(\pi_{tagname}(\sigma_{pname='LarryPage'}^{\text{PT.tid} = \text{TT.tid} \land \text{PT.pname} = \text{PT.pname}} \land \text{PT.pname} = \text{PT.pname})\)

4. [5 points] Find all the people (pname and city) who follow people who follow ’Tim Cook’. (i.e. second-level followers).
   
   **Solution:** \(\pi_{\text{P.pname, city}}(\pi_{\text{P.pname}}(\sigma_{\text{F.pname} = \text{PT.pname}}^{\text{T.Followers}}))\)

5. [5 points] Find all the people (pname and city) who follow at least everyone that ‘Larry Page’ follows.
   
   **Solution:** \(\pi_{\text{P.pname, city}}(\sigma_{\text{P.pname} = \text{F.pname}}^{\text{F.Followers}})(\pi_{\text{P.pname}}(\sigma_{\text{P.pname} = \text{F.pname}}^{\text{F.Followers}}))\)

6. [5 points] Find all pairs of people (pname) who have at least one follower in common.
   
   **Solution:** \(\pi_{\text{F.pname1, F1.pname2}}(\sigma_{\text{F.pname1} = \text{F1.pname2}}^{\text{F.Followers}})\)
Exercise 4 (Relational Calculi) [25 points]

Consider the following database schema:

EMPLOYEE (emp_name, address_street, address_city, phone_no)
DEPARTMENT (dept_name, mgr_name, mgr_start_date)
WORKS_ON (emp_name, industry_name, salary)
INDUSTRY (industry_name, location_city)
MANAGES (emp_name, mgr_name)

Write Tuple Relational Calculus expressions for problems 1 and 2. Write Domain Relational Calculus expressions for problems 3, 4, and 5.

1. [5 points] List the names of all employees who don’t work for ‘Verizon Wireless’ and make more than $50,000 per year.

   TRC: {s.emp_name | WORKS_ON(s) ^ s.industry_name != 'Verizon Wireless ^ s.salary > 50000}

2. [5 points] Find the names and phone number of all employees who work for Intel and live in Santa Clara.

   TRC: {t.emp_name t.phone_no | (Ǝs) (WORKS_ON(s) ^ (s.emp_name = t.emp_name ^ s.industry_name = 'Intel' ^ t.address_city = 'Santa Clara')}

3. [5 points] Find the names of all employees who live in the same city and on the same street as their managers.

   DRC: {<t> | Ǝ s, c, m (<t, s, c> ∈ employee ∧ <t, m> ∈ manages ∧ <m,s,c> ∈ employee)}

4. [5 points] Find the names of all employees in this database who live in the same city as the industry they work on is located in.

   DRC: {<p> | Ǝ st, c, i, sa (<p, st, c> ∈ employee ∧ <p, i, sa> ∈ works_on ∧ <i, c> ∈ industry)}

5. [5 points] Find the names of all industries located in every city in which Intel is located.

   DRC: {<i> | ∀ c (<i1, c> ∈ industry ∧ i1 = Intel' ⇒ <i, c> ∈ industry)}