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>25</td>
<td></td>
</tr>
<tr>
<td>Exercise 3</td>
<td>25</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. Briefly describe the terms primary key, alternate key, and foreign key. Give one simple example of each kind of key. [5 points]
   
   Primary Key – Selected candidate key that identifies tuples uniquely within a relation. No two tuples within a relation can have the same primary key. UFID for students at UF would be a primary key.

   Alternate Key – Candidate keys that are not selected to be primary, but still will have uniqueness within the relation. Email addresses for students at UF would be an alternate key.

   Foreign Key – A set of attributes within one relation that matches a candidate key of some other, or possibly same, relation. For entities Staff(staffNo, fName, lName, position, sex, DOB) and Client(clientNo, fName, lName, telNo, eMail, staffNo), with staffNo being a primary key for Staff, staffNo for client would be a foreign key that references Staff(staffNo).

2. Define Participation and Disjoint constraints, including the two variations of each constraint. [5 points]

Participation Constraints – Determines whether every member in the superclass must participate as a member of a subclass. Can be either mandatory or optional. Mandatory indicates the members must participate, optional indicates they may or may not participate as a member of a subclass.

Disjoint Constraints – Describes relationship between members of the subclasses and indicates whether member of a superclass can be a member of one or more than one subclass. Can be either disjoint or non-disjoint. Disjoint states that an instance of a superclass can be a member of only one subclass. Non disjoint rule states that an instance of a superclass can be a member of more than one subclass.

3. List the basic operations of the Relational Algebra with their name and their correct symbolic notation [5 points].

The basic operations are \( \cup \) (union), \(-\) (difference), \(\times\) (Cartesian product), \(\pi\) (projection), and \(\sigma\) (selection).

4. Describe how to define the Intersection and Division operations using only the five basic operations listed above. [5 points]

Intersection: \( R \cap S = R - (R - S) \)

Division:
\[
T1 \leftarrow \Pi_{R \cdot S}(R)
\]
\[
T_2 \leftarrow \Pi_{R \cdot S}((S \times T_1) - R) \\
T \leftarrow T_1 - T_2
\]

5. What are aggregate functions? Give examples of some aggregate functions in SQL [5 points].

Aggregate functions are functions that take a set of values and that yield a single value in a result tuple. Examples: SUM, AVG, COUNT, MIN, MAX.

Exercise 2 (Relational Algebra) [25 Points]

Assume the following database schema for this exercise:

**Employee** (**personName**: string, **street**: string, **city**: string)

**Work** (**personName**: string, **companyName**: string, **salary**: integer)

**Company** (**companyName**: string, **city**: string)

**Manage** (**personName**: string, **managerName**: string)

Give an expression in the relational algebra to express each of the following queries.

1. Find the names of all employees who work for Sunshine Co. [5 points]
   \[\pi_{\text{personName}} (\sigma_{\text{companyName} = 'Sunshine Co.'} (\text{Work}))\]

2. Find the names of all employees who live in the same city as their managers. [5 points]
   \[\pi_{\text{personName}} ((\text{Employee} \bowtie \text{Manage}) \bowtie \text{managerName}=\text{employee2.personName} \land \text{Employee.city} = \text{employee2.city}) \rho_{\text{employee2}} (\text{Employee}))\]

3. Assume the company may be located in several cities. Find all companies located in every city in which Sunshine Co. is located. [5 points]
   \[\pi_{\text{companyName}} (\text{Company} \div (\pi_{\text{city}} (\sigma_{\text{companyName} = 'Sunshine Co.'} (\text{Company}))))\]

4. Assume a person may work for more than one company. Find the names of all employees who do not work for Sunshine Co. [5 points]
   \[\pi_{\text{personName}} ((\sigma_{\text{companyName} \neq 'Sunshine Co.'} (\text{Work}))\]

5. Find the names of all employees who earn more than every employee of Sunshine Co. [5 points]
   \[\pi_{\text{personName}} (\text{Work}) - (\pi_{\text{Work.personName}} (\text{Work} \bowtie \text{Work.salary} < \text{work2.salary} \land \text{work2.companyName}='Sunshine Co.')) \rho_{\text{work2}} (\text{Work}))\]
Exercise 3 (Relational Algebra) [25 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. Find the employee Id’s of pilots who can operate the aircrafts of types "Boeing 747" and "Boeing 777". [3 points]
   \[
   \pi_{employeeId} (\sigma_{planeName = 'Boeing 747'} (Aircraft \bowtie Pilots)) \cap \pi_{employeeId} (\sigma_{planeName = 'Boeing 777'} (Aircraft \bowtie Pilots))
   \]

2. Find the id of the pilot, or pilots, with the highest salary. [3 points]
   \[
   \pi_{employeeId} (Pilots) \bowtie (\pi_{employeeId} (Employees) - \pi_{e2.employeeId} (\sigma_{e1.salary > e2.salary} (\rho_{e1} (Employees) \times \rho_{e2} (Employees))))
   \]

3. Find the airplane Id's of aircrafts that cannot fly non-stop from ATL to JFK. [3 points]
   \[
   \pi_{planeId} (\sigma_{range < distance} (Aircraft \times \sigma_{travelFrom = 'ATL' \land travelTo = 'JFK'} (Flights)))
   \]

4. 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. [5 points]
   \[
   \pi_{employeeName} (Employees \bowtie (\pi_{employeeId} (\sigma_{range >= 1500} (Aircraft \bowtie Pilots)) - (\pi_{employeeId} (\sigma_{planeName = 'Boeing 747'} (Aircraft \bowtie Pilots)) \cap \pi_{employeeId} (\sigma_{planeName = 'Boeing 777'} (Aircraft \bowtie Pilots)))))
   \]

5. Find the flight numbers of flights that can be piloted by every pilot whose salary is under $100,000. [5 points]
   \[
   \pi_{flightNumber} (Flights) - \Pi_{flightNumber} ((\Pi_{flightNumber (Flights)} \times \pi_{employeeId} (\sigma_{salary < 100000} (Employees \bowtie Pilots))) - \pi_{flightNumber.employeeId} (\sigma_{range >= distance} (Aircraft \times Flights) \bowtie Pilots))
   \]
   or
   \[
   \pi_{flightNumber} (\sigma_{range >= distance} (Aircraft \times Flights) \bowtie \sigma_{salary < 100000} (Pilots \bowtie Employees))
   \]

6. Find the Id’s of the pilots that fly all aircrafts. [3 points]
   \[
   Pilots \div \pi_{planeId} (Aircraft)
   \]
7. Find the names of all employees who are not pilots. [3 points]

\[ \pi_{\text{employeeName}}(\text{Employee} \bowtie (\pi_{\text{employeeId}}(\text{Employee}) \setminus \pi_{\text{employeeId}}(\text{Pilot}))) \]

**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 the Tuple Relational Calculus expression for problems 1 and 2. Write the Domain Relational Calculus expression for problems 3, 4, and 5.

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

   **TRC:** \{s.emp_name | WORKS_ON(s) \land s.industry_name \neq ‘Verizon Wireless’ \land s.salary > 50000\}

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

   **TRC:** \{t.emp_name t.phone_no | (\exists s) (WORKS_ON(s) \land (s.emp_name = t.emp_name \land s.industry_name = ‘Intel’ \land t.address_city = ‘Santa Clara’))\}

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

   **DRC:** \{<t> | \exists s, c, m (<t, s, c> \in employee \land <t, m> \in manages \land <m, s, c> \in employee)\}

4. Find the names of all employees in this database who live in the same city [5 points]

   **DRC:** \{ <p> | \exists st, c, co, sa (<p, st, c> \in employee \land <p, co, sa> \in works \land <co, c> \in company)\}

5. Find all companies located in every city in which Intel is located. [5 points]

   **DRC:** \{<C> | \forall co2, ci2 (co2, ci2 \in company \land co2 = Intel' \Rightarrow <co2, ci2> \in company)\}