Information and Database Management Systems I (CIS 4301)  
(Spring 2017)  

Instructor: Dr. Markus Schneider  
TA: Yang Peng  

Exam 1 Part 1 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>Question</th>
<th>Maximum</th>
<th>Received</th>
</tr>
</thead>
<tbody>
<tr>
<td>Question 1</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>Question 2</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>Question 3</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>50</td>
<td></td>
</tr>
</tbody>
</table>
**Question 1 (Knowledge Questions)**  [20 points]

1. [4 points] What are the different data abstraction levels in a database system? Characterize them briefly.
   - External/view levels describe the part of the DB, which is relevant for the user.
   - Conceptual/logical level gives information about existing data and relationships in the DB.
   - Physical/internal level describes how data are physically stored.

2. [4 points] How can we create a table in Oracle? What other operations can we do on tables?
   - Use the DDL (data definition language to create the table). Descriptions of “Create Table” statements are also OK.
   - Insertion, Deletion, Update, Select, etc.

3. [4 points]. Describe the difference between strong entity sets and weak entity sets.
   - The entity set which does not have sufficient attributes to form a primary key is called a weak entity set.
   - An entity set that has a primary key is called as Strong entity set.

4. [4 points] Briefly list the four stages of database design.
   - Requirements Analysis
   - Conceptual Design
   - Logical Design
   - Physical Design

5. [4 points] What is the difference between an inner join and an outer join?
   - Inner join: the result does not contain those tuples that did not find a partner.
   - Outer join: the result also contains those tuples that did not find a partner. The result tuples are “filled” with null values.
Question 2 (ER Model)  [20 points]

Suppose that Alachua County wants to centrally control the gyms of the area, and towards this goal you are hired to design a database to store information about the gyms and their employees. Below we describe the entities and the relations that should be stored in the database.

- Each gym has a name, street number, street name, ZIP code, and one or more phone numbers. The gym names are unique.
- An employee is uniquely defined by her SSN. Moreover, we store her name.
- An employee may work at several gyms of the Alachua County. For instance, Smith is working in the mornings at “X Shadyside” and in the evenings at “Fitness Factory”.
- For every employee we record the percentage of time he or she works at each gym. Thus, employee ‘Smith’ above would be recorded as working at 50% at “X Shadyside” and 50% at “Fitness Factory”.
- Some employees may specialize in one of the following specialties: manager, receptionist, or personal trainer. Each employee has zero or one specialization.
- Every manager manages one or more gyms.
- Each gym has exactly one manager, that is, it cannot be without a manager.
- For a personal trainer we also store the type(s) of certification he/she has. Some examples of certification are yoga, aerobics, and sports nutrition. A trainer may have zero or one certification.
- The information stored for a customer is: SSN (unique), name, age.
- Each customer may be going to more than one gym. For example, Alice attends group exercise classes both in “Fitness Planet” and “X Shadyside”, while Bob has always been going to “Club One Fitness”.
- The gyms also allow each customer to have guests/friends associated with him/her. These guests can use the facilities of the gyms their host goes. The guests are not considered customers of the gym. Only the name and age of the guests are stored in the database, and we assume that for each customer the pair (guest_name, guest_age) is unique.

[20 points] Give the whole E.R. diagram for the small scenario described above. Make sure to indicate the primary and partial keys, cardinality constraints, weak entities (if applicable), and participation constraints. If you make any assumptions, state them clearly.
[Each entity set needs to have a primary/partial key. Grading criteria:
* Missing keys/attributes: -1 each, -5 max
* Bad design: -1~10, depending on how bad it is.
* Missing relationship/entity: -2 each.
* Missing/incorrect cardinalities: -1 each, -5 max.]
Question 3 (Relational Algebra) [10 points]

The Relational Algebra operator ÷ (quotient/division) is defined as follows:
\[ R ÷ S = \pi_{R - S}(R) - \pi_{R - S}((\pi_{R - S}(R) \times S) - R) \]

Explain in your own words but precisely how this relational algebra expression computes the intended result of the quotient operator. For this, take parts of the relational algebra expression above and explain what they do. Examples are not allowed.

Solution:

The intended result of the division operator consists of the restrictions of tuples in R to the attribute names unique to R, i.e., in the header of R but not in the header of S, for which it holds that all their combinations with tuples in S are present in R.

In detail, the algebra expression above can be explained as follows:

1. \( R - S \) is the schema that contains all attributes in R that are not in S. It is the result schema of the quotient.
2. \( \pi_{R - S}(R) \) reduces all tuples in R to the attributes that are in the schema of R but not in S. Note that \( S \subseteq R \) holds. Let us call the result tuples prefix tuples of R.
3. \( (\pi_{R - S}(R) \times S) \) returns the Cartesian product of the prefix tuples \( \pi_{R - S}(R) \) and S, i.e. all possible combinations of the prefix tuples and S. The schema of this Cartesian product is equal to the schema of R.
4. The expression \( (\pi_{R - S}(R) \times S) - R \) subtracts all tuples of R from this Cartesian product of all possible combinations of the prefix tuples and S. Let \( t \in \pi_{R - S}(R) \). The Cartesian product \( (\pi_{R - S}(R) \times S) \) will contain all combinations of \( t \) with all tuples \( s \in S \). We can distinguish two cases: If all these combinations are also in R, they will be completely eliminated in the result of \( (\pi_{R - S}(R) \times S) - R \). This means there will be no tuple with the prefix \( t \) any more in \( (\pi_{R - S}(R) \times S) - R \). Otherwise, if not all of these combinations are in R, the prefix \( t \) can still be found in \( (\pi_{R - S}(R) \times S) - R \).
5. This means that the expression \( \pi_{R - S}((\pi_{R - S}(R) \times S) - R) \) keeps all prefix tuples \( t \in \pi_{R - S}(R) \) that could not be completely eliminated by R in the Cartesian product \( (\pi_{R - S}(R) \times S) \). In other words, \( \pi_{R - S}((\pi_{R - S}(R) \times S) - R) \) returns those prefix tuples with attributes unique to R that have the property that not all of their combinations with tuples in S are in R. This means that \( \pi_{R - S}((\pi_{R - S}(R) \times S) - R) \) determines the set of all disqualifying prefix tuples.
6. Therefore, \( R ÷ S = \pi_{R - S}(R) - \pi_{R - S}((\pi_{R - S}(R) \times S) - R) \), by removing all disqualifying prefix tuples from \( \pi_{R - S}(R) \), computes the intended result, namely those prefix tuples that in combination with all tuples from S can be found in R.