CS145 Midterm Examination

Autumn 2009, Prof. Widom

- Please read all instructions (including these) carefully.
- There are 7 problems on the exam, with a varying number of points for each problem and subproblem for a total of 75 points to be completed in 75 minutes. *You should look through the entire exam before getting started, in order to plan your strategy.*
- The exam is closed book and closed notes, but you may refer to your three pages of prepared notes.
- Please write your solutions in the spaces provided on the exam. Make sure your solutions are neat and clearly marked. You may use the blank areas and backs of the exam pages for ungraded scratch work.
- *Simplicity and clarity of solutions will count.* You may get as few as 0 points for a problem if your solution is far more complicated than necessary, or if we cannot understand your solution.
- Throughout the exam you should assume and use "pure" SQL, XPath, XQuery, and XSLT as covered in class—not dialects of these languages supported by a particular implementation (such as mySQL or Saxon).

NAME ____

In accordance with both the letter and spirit of the Honor Code, I have neither given nor received assistance on this examination.

SIGNATURE: ____

Problem	1	2	3	4	5	6	7	TOTAL
Max. points	5	8	18	18	8	12	6	75
Points								

1. **Relational Algebra** (5 points)

   Consider two relations $R(A, B)$ and $S(A, B)$. You would like to compute their intersection $R \cap S$, but unfortunately you only have four relational algebra operators at your disposal: σ, π, $\times$, and $\bowtie$ (natural join). Is it possible to compute $R \cap S$ using just these four operators? If so, show the simplest equivalent expression you can come up with. If not, briefly explain why not.

2. **SQL** (8 points)

   Consider a table `T(A char(1))`. Write a SQL query to find the largest number of duplicated values in T. (For example, if T contains three `A`'s, five `C`'s, and four `E`'s, then your query should return the number 5.) Your query will be graded on simplicity as well as on correctness.

3. **Query Equivalences** (18 points)

Each row of the table below shows two queries. In the blank third column of the table write "YES" if the two queries are equivalent, and "NO" if they are not equivalent. Remember that two queries are equivalent if they always return exactly the same answer on all databases.

All queries refer to relations $R(A,B)$ and/or $S(A,B)$.

- In both relations R and S, attribute A is a key and attribute B is not a key.
- No attributes are permitted to contain NULL values.
- Do not make any other assumptions about the data.

To lessen the score for guessing, you will earn 1.5 points for each correct answer, -1 points for each incorrect answer, and 0 points for each answer left blank.

Query 1	Query 2	Equiv.?
$\pi_A(R-S)$	$\pi_A(R) - \pi_A(S)$	
$\pi_B(R-S)$	$\pi_B(R) - \pi_B(S)$	
$\pi_B(R \cup S)$	$\pi_B(R) \cup \pi_B(S)$	
$\pi_{R.A,S.A}(R \times S)$	$\pi_A(R) \times \pi_A(S)$	
$\pi_A(R) \cup \rho_A \pi_B(S)$	`(Select A From R) Union` `(Select Distinct B As A From S)`	
$\pi_{R.A}(\sigma_{R.B=S.B}(R \times S))$	`Select R.A From R,S Where R.B=S.B`	
`Select B From R` `Where A In (Select A From S)`	`Select R.B From R,S Where R.A=S.A`	
`Select A From R` `Where A not in (Select A From S)`	`Select R.A From R,S Where R.A<>S.A`	
$\pi_B(R) -$ $\pi_{R1.B}(\sigma_{R1.B \geq R2.B}(\rho_{R1}(R) \times \rho_{R2}(R)))$	`Select Min(B) From R`	
$\pi_B(R) -$ $\pi_{R1.B}(\sigma_{R1.B < R2.B}(\rho_{R1}(R) \times \rho_{R2}(R)))$	`Select Max(B) From R`	
`Select A From R`	`Select A From R Group By A`	
`Select B From R`	`Select B From R Group By B`	

4. **XML, XPath, and XQuery** (18 points)

Consider the following Document Type Descriptor (DTD) for XML documents containing information about students.

```
<!DOCTYPE Students [
   <!ELEMENT Students (Student+)>
   <!ELEMENT Student (BlogPost*, Friends)>
   <!ATTLIST Student Name CDATA #REQUIRED>
   <!ELEMENT BlogPost (#PCDATA)>
   <!ELEMENT Friends (Friend*)>
   <!ELEMENT Friend (Name)>
   <!ELEMENT Name (#PCDATA)>
]>
```

(a) (4 points) Specify the smallest example XML document you can come up with (no header needed) that is valid with respect to the `Students` DTD.

(problem continues on next page)

(b) (6 points) You are to write a query in *XPath* to be executed on a document "students.xml" that conforms to the `Students` DTD above. Your query should return the names of all students who have at least three friends and the third one's name is "Ghost", or who have a blog post consisting of the single phrase "Trick-or-treat". Your expression will be graded on simplicity as well as correctness.

(problem continues on next page)

(c) (8 points) Still considering a document "students.xml" conforming to the `Students` DTD, but now using *XQuery*, write a query that returns all students who have posted a blog that is entirely contained in some other student's blog. To keep things simple, your result may have the following form, and don't worry about duplicates:

```
<Thieves>
   <Student> all-student-stuff-here </Student>
   <Student> all-student-stuff-here </Student>
                     ...
</Thieves>
```

5. **XML Schema and DTDs** (8 points)

Continue with the `Students` DTD from the previous problem:

```
<!DOCTYPE Students [
   <!ELEMENT Students (Student+)>
   <!ELEMENT Student (BlogPost*, Friends)>
   <!ATTLIST Student Name CDATA #REQUIRED>
   <!ELEMENT BlogPost (#PCDATA)>
   <!ELEMENT Friends (Friend*)>
   <!ELEMENT Friend (Name)>
   <!ELEMENT Name (#PCDATA)>
]>
```

(a) (5 points) Suppose you decide to use the `IDREF` concept instead of subelement structure to represent friends. Modify the DTD accordingly. If you prefer to show only the modified parts of the DTD you may do so, but be extremely clear. Try to make your XML structure captured by the DTD as simple as possible, without losing information.

(problem continues on next page)

(b) (3 points) Consider the syntactically correct XML Schema specification below. Is this specification satisfied by exactly the same set of XML documents that conform to the original `Students` DTD? Answer "YES" or "NO." If you answer "NO", give a brief explanation why not.

Hint: You don't need to examine the specification in detail. Scan for key features of XML Schema.

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
 <xsd:element name="Students">
  <xsd:complexType>
   <xsd:sequence>
    <xsd:element name="Student">
     <xsd:complexType>
      <xsd:sequence>
       <xsd:element name="BlogPost" type="xsd:string"/>
       <xsd:element name="Friends">
        <xsd:complexType>
         <xsd:sequence>
          <xsd:element name="Friend">
           <xsd:complexType>
            <xsd:sequence>
             <xsd:element name="Name" type="xsd:string"/>
            </xsd:sequence>
           </xsd:complexType>
          </xsd:element>
         </xsd:sequence>
        </xsd:complexType>
       </xsd:element>
      </xsd:sequence>
      <xsd:attribute name="Name" type="xsd:string" use="required"/>
     </xsd:complexType>
    </xsd:element>
   </xsd:sequence>
  </xsd:complexType>
 </xsd:element>
</xsd:schema>
```

6. **Functional and Multivalued Dependencies** (12 points; 6 for each part)

The relation $R(A, B, C)$ satisfies an unknown set of functional and multivalued dependencies. All we know about R is that it allows at least the following two instances:

A	B	C
1	2	3
1	3	4

A	B	C
1	3	3
2	2	4
3	3	3

(a) Based on R's schema alone, enumerate all possible completely nontrivial functional dependencies that have a single attribute on the right-hand side. Then, based on the instances above, for each FD you listed, label whether it:

H: Definitely holds in R.

NH: Definitely does not hold in R.

CD: Cannot be determined from the information given whether or not it holds in R.

(b) Based on R's schema alone, enumerate all possible nontrivial multivalued dependencies. Then, based on the instances above, for each MVD you listed, label it according to the $\{$**H**, **NH**, **CD**$\}$ scheme from part (a).

7. **Normal Forms** (6 points; 2 for each part)

Consider the following two relational schemas:

```
Schema 1: R(A,B,C)
Schema 2: R1(A,B), R2(A,C)
```

(a) Suppose that the only dependencies (functional or multivalued) that hold on the relations in these schemas are $A \rightarrow BC$ and all dependencies that follow from this one. Circle exactly two of the following.

(1) Schema 1 is in neither BCNF nor 4NF
(2) Schema 1 is in BCNF but not 4NF
(3) Schema 1 is in 4NF but not BCNF
(4) Schema 1 is in both BCNF and 4NF
(5) Schema 2 is in neither BCNF nor 4NF
(6) Schema 2 is in BCNF but not 4NF
(7) Schema 2 is in 4NF but not BCNF
(8) Schema 2 is in both BCNF and 4NF

(b) Now suppose that the only dependencies (functional or multivalued) that hold on the relations in these schemas are $BC \rightarrow A$, $A \rightarrow C$, and all dependencies that follow from these two. Circle exactly two of the following.

(1) Schema 1 is in neither BCNF nor 4NF
(2) Schema 1 is in BCNF but not 4NF
(3) Schema 1 is in 4NF but not BCNF
(4) Schema 1 is in both BCNF and 4NF
(5) Schema 2 is in neither BCNF nor 4NF
(6) Schema 2 is in BCNF but not 4NF
(7) Schema 2 is in 4NF but not BCNF
(8) Schema 2 is in both BCNF and 4NF

(c) Now suppose that the only dependencies (functional or multivalued) that hold on the relations in these schemas are $A \twoheadrightarrow B$ and all dependencies that follow from this one. Circle exactly two of the following.

(1) Schema 1 is in neither BCNF nor 4NF
(2) Schema 1 is in BCNF but not 4NF
(3) Schema 1 is in 4NF but not BCNF
(4) Schema 1 is in both BCNF and 4NF
(5) Schema 2 is in neither BCNF nor 4NF
(6) Schema 2 is in BCNF but not 4NF
(7) Schema 2 is in 4NF but not BCNF
(8) Schema 2 is in both BCNF and 4NF