CS145 Midterm Examination

Spring 2004, Prof. Widom

- Please read all instructions (including these) carefully.
- There are 9 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 scratch work. Please do not use any additional scratch paper.
- *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, and XQuery as covered in class—not dialects of these languages supported by a particular implementation (such as Oracle, mySQL, or Quip).

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	8	9	TOTAL
Max. points	6	6	8	18	6	5	13	5	8	75
Points										

1. **Relational Algebra** (6 points)

   Consider a relation `Temp(regionID,name,date,high,low)` that records high and low temperatures for various regions on various dates. Regions have a `name` but are identified by their `regionID`, so a key for the relation is `<regionID,date>`. Consider the following relational algebra expression, written in *linear notation* as covered in the textbook.

   $$\texttt{T1(rID,date,h)} := \pi_{\texttt{regionID,date,high}}(\texttt{Temp})$$
   $$\texttt{T2(rID,date,l)} := \pi_{\texttt{regionID,date,low}}(\texttt{Temp})$$
   $$\texttt{T3(regionID)} := \pi_{\texttt{rID}}(\texttt{T1} \bowtie_{\texttt{h<high}} \texttt{Temp})$$
   $$\texttt{T4(regionID)} := \pi_{\texttt{rID}}(\texttt{T2} \bowtie_{\texttt{l>low}} \texttt{Temp})$$
   $$\texttt{T5(regionID)} := \pi_{\texttt{regionID}}(\texttt{Temp}) - \texttt{T3}$$
   $$\texttt{T6(regionID)} := \pi_{\texttt{regionID}}(\texttt{Temp}) - \texttt{T4}$$
   $$\texttt{Answer(n)} := \pi_{\texttt{name}}(\texttt{Temp} \bowtie (\texttt{T5} \cup \texttt{T6}))$$

   State in English what this expression computes as the final `Answer`. Please state the English interpretation succinctly and in terms of the real-world data being represented. Under no circumstances should your description discuss details of expression evaluation or matching. The correct answer can be expressed in a single short sentence.

2. **Relational Algebra and Functional Dependencies** (6 points)

   Consider a relation $R(A, B, C)$. Write a relational algebra expression that returns an empty result if and only if the functional dependency $A \rightarrow B$ holds on R. Please write a single algebraic expression—do not use linear or tree notation.

3. **SQL** (8 points)

The *anti-semijoin* of two relations R and S is defined to contain the "dangling" tuples in R: those tuples in R that have no joining tuples in S, and no others. Consider $R(A, B, C)$ and $S(A, B, D)$ joining on shared attributes A and B, and assume $\langle A, B\rangle$ is a key for each relation. For each of the following SQL queries, write YES in the second column if the query correctly implements the anti-semjoin, and write NO otherwise. Note that `R.*` in a SQL `select` clause denotes all attributes of relation R.

You will receive 1 point for each correct answer, -1 *point for each incorrect answer, and 0 points for each blank answer. In other words, it is not to your advantage to guess!*

Query Implements Anti-Semijoin?	**YES or NO**

`select * from R`
`where <A,B> not in (select A,B from S)`	

`select R.* from R,S`
`where R.A <> S.A and R.B <> S.B`	

`select R.* from R,S`
`where R.A <> S.A or R.B <> S.B`	

`select * from R`
`where not exists`
`(select * from S where S.A = R.A and S.B = R.B)`	

`select * from R`
`where not exists`
`(select * from S where S.A = R.A or S.B = R.B)`	

`select * from R`
`where A <>any (select A from S)`
`and B <>any (select B from S)`	

`select * from R`
`where A <>all (select A from S)`
`or B <>all (select B from S)`	

`select * from R`
`where 1 > (select count(*) from S`
`where S.A = R.A and S.B = R.B)`	

4. **SQL, Dependencies, and Normal Forms** (18 points)

Consider a table recording how many hours students sleep at night:

```
Sleep(student,major,date,#hours)
```

For now make the following two assumptions and no others:

(A1) `<student,date>` is a key.

(A2) Functional dependency `student` → `major` holds.

(a) (10 points) Write a SQL query to find all students whose average night's sleep is shorter than the average over all students with the same major. Your query will be graded on simplicity as well as correctness.

(problem continues on next page)

(b) (4 points) Based on assumptions (A1) and (A2) above, and no others, is `Sleep` in Boyce-Codd Normal Form (BCNF)?

Circle one: YES NO

If you circled YES: In the space below, state the smallest possible change to the assumptions about keys and dependencies so that `Sleep` is no longer in BCNF. Your new assumptions should "make sense" in the real world, meaning it could be reasonable to require the data in `Sleep` to satisfy these assumptions.

If you circled NO: In the space below, decompose `Sleep` into relations that are in BCNF, following the decomposition algorithm covered in class and in the textbook. Show the result of your decomposition only, not the process.

(c) (4 points) Now drop all assumptions about keys and dependencies from parts (a) and (b) of this problem. Instead make the following assumption and no others:

(A3) Multivalued dependency `student` $\twoheadrightarrow$ `major` holds.

Under this assumption only, answer the following two questions.

(i) Is `Sleep` in Fourth Normal Form (4NF)? Circle one: YES NO

(ii) Regardless of whether you circled YES or NO for part (i), is the instance of table `Sleep` shown below legal under assumption (A3) and no others?

Circle one: YES NO

student	*major*	*date*	*#hours*
Jane	CS	5/1/04	6.0
Jane	EE	5/1/04	6.0
Jim	CS	5/1/04	6.0

5. **XML Data** (6 points)

Consider the following DTD for XML documents:

```
<!DOCTYPE Univ [
   <!ELEMENT University (Dept+)>
   <!ATTLIST Dept Name ID #REQUIRED>
   <!ELEMENT Dept (Faculty | Staff | Student)*>
   <!ATTLIST Faculty Name ID #REQUIRED>
   <!ELEMENT Faculty (Office, Salary)>
   <!ATTLIST Staff Name ID #REQUIRED>
   <!ELEMENT Staff (Office, Salary)>
   <!ATTLIST Student Name ID #REQUIRED Advisor IDREF #REQUIRED>
   <!ELEMENT Student (Dorm)>
   <!ELEMENT Office (#PCDATA)>
   <!ELEMENT Salary (#PCDATA)>
   <!ELEMENT Dorm (#PCDATA)> ]
```

We refer to this DTD as the *University DTD*. It is used for this problem and the following two problems.

Specify an example XML document that is valid with respect to the University DTD. Your example should include exactly one Department, one Faculty member, and one Student.

6. **XPath** (5 points)

   Consider documents conforming to the University DTD introduced in the previous problem. Write an XPath expression to find the names of all Faculty members who share an office with a Staff member. Do not worry about "`document(...)`" specifications—you may start path expressions with "`/University/...`" as in class. Your solution will be graded on simplicity as well as correctness.

7. **XQuery** (13 points)

   Consider the following XQuery expression over documents conforming to the University DTD used in the previous two problems. As in the previous problem, we are omitting "`document(...)`" specifications.

```
for $d1 in /University/Dept
for $d2 in /University/Dept[@Name <> $d1/@Name]
for $f in $d1/Faculty
let $s1 := $d1/Student[@Advisor=$f/@Name]
let $s2 := $d2/Student[@Advisor=$f/@Name]
where count($s2) > count($s1)
return <Mystery>
         <F>{$f/@Name}</F>
         <D>{$d2/@Name}</D>
       </Mystery>
```

   (a) (5 points) State in English what is computed by this query. Your answer will be graded on clarity and conciseness, as well as on correctness.

(problem continues on next page)

(b) (2 points) Is it possible to rearrange the `for` and `let` clauses in the above query so at least one `let` appears before at least one `for`, without changing the meaning or correctness of the query?

Circle one: YES NO

(c) (2 points) Can the above query produce duplicate `Mystery` elements in its result?

Circle one: YES NO

(d) (4 points) Briefly describe what could change in the query result if the condition `[@Name <> $d1/@Name]` is removed from the second `for` clause. Your answer will be graded on clarity and conciseness, as well as on correctness.

(scratch space)

8. **Functional Dependencies and Keys** (5 points)

Consider a relation $R(A, B, C, D, E, F, G, H)$ with the following functional dependencies:

$$A \rightarrow BCD$$
$$AD \rightarrow E$$
$$EFG \rightarrow H$$
$$F \rightarrow GH$$

(a) (3 points) Based on these functional dependencies, there is one key for R. What is it?

(b) (2 points) One of the four functional dependencies can be removed without altering the key. Which one?

9. **Normal Forms** (8 points)

Consider a schema consisting of two relations: $R_1(A, B, C)$ and $R_2(B, D)$.

(a) (4 points) Suppose that the only functional dependencies that hold on the relations in this schema are $A \rightarrow B$, $A \rightarrow C$, $B \rightarrow A$, $A \rightarrow D$, and all dependencies that follow from these. Is this schema in Boyce-Codd Normal Form (BCNF)?

Circle one: YES NO

(b) (4 points) Suppose that the only functional and multivalued dependencies that hold on the relations in this schema are $A \rightarrow BD$, $D \rightarrow C$, $C \twoheadrightarrow AB$, $A \twoheadrightarrow B$, $B \twoheadrightarrow D$, and all dependencies that follow from these. Is this schema in Fourth Normal Form (4NF)?

Circle one: YES NO