CS145 Midterm Examination

Autumn 2005, Prof. Widom

- Please read all instructions (including these) carefully.
- There are 8 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 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	8	TOTAL
Max. points	10	10	5	4	14	12	14	6	75
Points									

1. **Relational Algebra** (10 points)

   Consider a relation `Taking(student,class)` in which neither attribute alone is a key. Write a relational algebra expression that returns all pairs of students who have no classes in common. Make sure to include each pair only once—for example, if your expression returns `<Mary,Fred>` then it should not also return `<Fred,Mary>`. Remember that simplicity and clarity count, as well as correctness.

2. **SQL** (10 points)

   Consider a table `Exams(student,score)` containing student exam scores. Each student may have any number of scores recorded in the table. Write a SQL query to find the student with the highest score differential, i.e., the student with the largest spread between his or her highest and lowest scores, among all students with scores in the table. Assume there is a unique student with the highest spread and return that student only once. Your query will be graded on simplicity and understandability as well as on correctness.

3. **More SQL** (5 points)

Consider the following two SQL queries over table $R(K, A)$ where K is a key.

```
Q1: select * from R
    where A >= all (select A from R)

Q2: select * from R as R1
    where A > all (select A from R as R2 where R1.K <> R2.K)
```

Are these two queries equivalent? That is, do they return the same answer on all possible instances of R? (circle one) YES NO

If you circled YES, briefly justify why the queries are equivalent. If you circled NO, show the smallest instance of R you can find that gives different answers for `Q1` and `Q2`.

4. **XML** (4 points)

Here is an XML DTD:

```
<!DOCTYPE X [
  <!ELEMENT X (Y, Z*)*>
  <!ELEMENT Y (Z+)
  <!ELEMENT Z (#PCDATA) ]>
```

Give an example of the smallest data set (i.e., the fewest number of elements) you can think of that is valid with respect to this DTD and includes at least one each of `X`, `Y`, and `Z` elements. Write your answer as well-formed XML.

5. **XPath and XQuery** (14 points)

Consider querying XML documents containing information about students in classes. The documents conform to the following DTD:

```
<!DOCTYPE Classes [
  <!ELEMENT Classes (Class*)>
  <!ELEMENT Class (Topic, Students)>
  <!ATTLIST Class Number ID #REQUIRED Units CDATA #REQUIRED>
  <!ELEMENT Topic (#PCDATA)>
  <!ELEMENT Students (Student+)>
  <!ELEMENT Student (FirstNm, LastNm)>
  <!ELEMENT FirstNm (#PCDATA)>
  <!ELEMENT LastNm (#PCDATA)> ]>
```

For each of the query pairs in (a)–(e), circle YES if the XPath and XQuery expressions are equivalent (i.e., they are guaranteed return the same result over any XML document conforming to the above DTD), and circle NO if they are not equivalent (i.e., there is some document conforming to the DTD for which they will return different results).

For equivalence don't take into account details of answer presentation (such as <`result`> tags), just consider whether the query results contain the same set of elements. Also don't worry about `doc(..)` specifications or type coercions.

Scoring: To discourage guessing on parts (a)–(e), you will receive 2 points if you circle the correct answer and −2 points if you circle the incorrect answer.

(a)
```
XPath:
  /Classes/Class[Students/Student[LastNm="Smith"]]/Topic
XQuery:
  for $c in /Classes/Class
  where every $n in $c/Students/Student/LastNm satisfies $n="Smith"
  return $c/Topic
```

Queries are equivalent? YES NO

(b)
```
XPath:
  //*[@Number="1234"]//Student
XQuery:
  for $c in /Classes/Class
  where $c/@Number="1234"
  return $c/Students/Student
```

Queries are equivalent? YES NO

(c) XPath:

```
/Classes/Class
  [Students/Student/FirstNm != Students/Student/FirstNm
   and Students/Student/LastNm != Students/Student/LastNm]/Topic
```

XQuery:

```
for $c in /Classes/Class
for $s1 in $c/Students/Student
for $s2 in $c/Students/Student
where $s1/FirstNm != $s2/FirstNm and $s1/LastNm != $s2/LastNm
return $c/Topic
```

Queries are equivalent? YES NO

(d) XPath:

```
/Classes/Class[@Units="5"][3]/Topic
```

XQuery:

```
for $c in /Classes/Class[3][@Units="5"]
return $c/Topic
```

Queries are equivalent? YES NO

(e) XPath:

```
/Classes/Class[@Number="1234"]/@Units
```

XQuery:

```
let $c := /Classes/Class[@Number="1234"]
return max($c/@Units)
```

Queries are equivalent? YES NO

(f) *(4 points)* For the same DTD, consider the following XQuery expression evaluated over a data set representing one class with 3 students. List all of the tag names that would appear in the result, including any duplicates.

```
for $s in //(Students | Student)
for $x in $s/preceding-sibling::*
return name($x)
```

Tag names:

6. **Dependencies and Normal Forms** (12 points – 2 for each subpart)

Consider the following two relational schemas:

Schema 1: $R(A, B, C)$

Schema 2: $R_1(A, B)$, $R_2(A, C)$

(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.

i. Schema 1 is in (circle the strongest one): 3NF, BCNF, 4NF, none of these

ii. Schema 2 is in (circle the strongest one): 3NF, BCNF, 4NF, none of these

(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.

i. Schema 1 is in (circle the strongest one): 3NF, BCNF, 4NF, none of these

ii. Schema 2 is in (circle the strongest one): 3NF, BCNF, 4NF, none of these

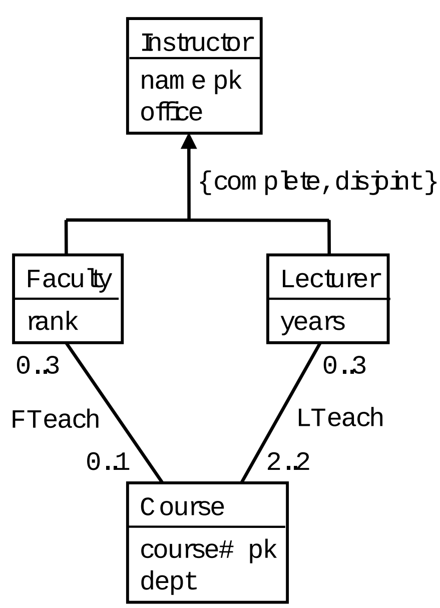
(c) Now suppose that the only dependencies (functional or multivalued) that hold on the relations in these schemas are $A \twoheadrightarrow B$, $A \twoheadrightarrow C$, and all dependencies that follow from these two.

i. Schema 1 is in (circle the strongest one): 3NF, BCNF, 4NF, none of these

ii. Schema 2 is in (circle the strongest one): 3NF, BCNF, 4NF, none of these

7. **UML** (14 points)

Consider the following UML diagram.



(a) *(2 points)* According to the diagram, what are the minimum and maximum total number of instructors for a given course?

Minimum: [ ] Maximum: [ ]

(b) *(2 points)* According to the diagram, what is the minimum and maximum teaching load (number of courses) for lecturers? For professors?

Lecturer minimum: [ ] Lecturer maximum: [ ]

Professor minimum: [ ] Professor maximum: [ ]

(c) *(10 points)* Convert the diagram to a relational schema, with the following requirements.

- Use as few relations as possible, while still conforming to one or more of the translation schemes discussed in class.
- Underline a minimal key for each relation.
- Suppose that by default attribute values cannot contain NULL. If your translation scheme requires any attributes to permit NULL values, circle them.

Write your relational schema in the box: