CSE344 Section 6 :)

Homework 4

- Reminder: Due on Saturday night, no late day
- Updated on lecture slides (lecture #13):
- http://www.cs.washington.edu/education/courses/cse344/12au/lectures/lecture13-xpath-xquery.pdf
- you can use 0/1/more "let" for the homework

Some useful hints

- Function reference, quick look:
- http://www.w3schools.com/xpath/xpath_functions.asp
- fn:normalize_space(string): “X ml” -> “X ml”
- Fn:tokenize(string, pattern): like string.split() function in java

Topics covered so far (maybe on the midterm)

- SQL, aggregates in SQL
- Basic Database Tuning
- Nested Query
- Relational Algebra, Datalog, and Relational Calculus
- XML, Xpath, XQuery

```
Paper(pid, title)
Reviewer(rid, name)
Reviews(rid, pid)
```

- `pid` is a unique paper identifier and the primary key of the `Paper` table.
- `rid` is a unique reviewer identifier and the primary key of the `Reviewer` table.
- `Reviews.rid` is a foreign key that references `Reviewer.rid`.
- `Reviews.pid` is a foreign key that references `Paper.pid`.
- A reviewer is assigned zero or more papers.
- A paper is assigned zero or more reviewers.

(a) (15 points) Write a **SQL query** that finds all papers with fewer than three reviewers assigned to them. The output of the query should be a list of paper titles. The result should **include papers without any reviewers** assigned to them.

(15 points) Write a **SQL query** that finds the reviewers with the most papers assigned to them. There can be more than one such reviewer. The output of the query should be a list of reviewer names. A reviewer should be listed if no other reviewer has strictly more papers to review.

Consider the following DTD, which describes the schema for a database containing a list of players. The rank of a player indicates whether the player is a "Beginner" or an "Advanced" player. The score is the total number of points accumulated by the player.

```
<!DOCTYPE game [
    <!ELEMENT game (player*)>
    <!ELEMENT player (rank,score)>
    <!ATTLIST player name CDATA #REQUIRED >
    <!ELEMENT rank (#PCDATA )>
    <!ELEMENT score (#PCDATA )>
]>
```

(a) (20 points) Write an XQuery expression that reformats a valid XML document, as per the above DTD, into one that matches the following DTD. In this new format, we want to group "Beginner" players into one category and "Advanced" players into another category:

```
<!DOCTYPE game [
    <!ELEMENT game  (category*)>
    <!ELEMENT category (rank, player*)>
    <!ELEMENT player (name, score) >
    <!ELEMENT rank     (#PCDATA )>
    <!ELEMENT name     (#PCDATA )>
    <!ELEMENT score    (#PCDATA )>
]>
```