Databases: Assignment #3

CS 338 Fall 2015

Solutions on November 10 in class

Problem 1: more Aggregate Queries

All the queries have to be formulated with respect to the bibliography database introduced in class; here is the schema of our database again:

```
author(AID INTEGER, NAME CHAR(22), URL CHAR(42))
wrote(AUTHOR INTEGER, PUBLICATION CHAR(8))
publication(PUBID CHAR(8), TITLE CHAR(70))
book(PUBID CHAR(8), PUBLISHER CHAR(50), YEAR INTEGER)
journal(PUBID CHAR(8), VOLUME INTEGER, NO INTEGER, YEAR INTEGER)
proceedings(PUBID CHAR(8), YEAR INTEGER)
article(PUBID CHAR(8), CROSSREF CHAR(8), STARTPAGE INTEGER,
                                          ENDPAGE INTEGER)
```

Note that queries 1-15 coincide with Assignment 1 (so it is sufficient to transliterate your *relational calculus* solutions to SQL).

Queries:

Write the following queries in SQL:

1. List authors who wrote the most books (including all ties).

   Answer: Author id and name

2. Find authors who wrote the longest article (again, include ties); find **two** solutions to this query, one using the `MAX` aggregate and another using **subqueries**.

   Answer: Author id and name

Problem 2: Duplicates and NULLs

(A) Do the two queries below always return the same answer? (without assuming any integrity constraints on the underlying tables?)

```
Q1: ( SELECT a.aid, a.name, p.pubid
      FROM   author a, wrote w, article p
      WHERE  a.aid=w.author and w.publication=p.pubid )
    UNION ALL
    ( SELECT a.aid, a.name, p.pubid
      FROM   author a, wrote w, book p
      WHERE  a.aid=w.author and w.publication=p.pubid )

Q2:  SELECT a.aid, a.name, w.publication
     FROM   author a, wrote w
     WHERE  a.aid=w.author and
      ( w.publication in ( SELECT pubid from article ) OR
        w.publication in ( SELECT pubid from book ) )
```

(B) List *integrity constraints* that will make `Q1` and `Q2` behave the same (include *all* constraints needed, including those mentioned in class).

(C) Show how a query (over two binary relations R(a,b) and `S(b,c)`)

```
SELECT R.a, R.b, S.c
FROM   R FULL OUTER JOIN S ON R.b=S.b
```

can be expressed *without* the use of `FULL OUTER JOIN`.

Problem 3: ER-modeling

You have been asked by the Waterloo Recreational Sports Association (WRSA) to help them with their database design. The WRSA has come up with a database specification and included it in the next section. You will be required to create an ER model based on the specification. Once you have created the ER model you will map it to a relational model using the steps from lecture.

Requirements/Specification

The WRSA oversees several sports. Each sport has a unique name, the number of referees required to referee matches of that sport, and a list of locations where the sport can be played. For simplicity, each location is a single string. The WRSA is comprised of many teams. Each team plays exactly one sport, and every sport has some team associated with

it. Each team has a unique name. A team may choose to have one or more coaches. However, since the WRSA is recreational, teams are not required to have a coach. Coaches have a name, address, phone number, and certification date. Players who are registered in the WRSA are members of teams. Each player can be a member of multiple teams, but they must be a member of at least one team. Players have a name, age, phone number, and address. Coaches and players are covered by the WRSAs insurance and each have an insurance number. Each player must have at least one emergency contact. The emergency contacts each have a name, address, relation to the player, and phone number. A player cannot be a coach and cannot have two emergency contacts with the same name. The WRSA selects several players to act as mascots. Each mascot must have a nickname. Every team plays in matches with other teams. A match is given a unique match number, a location, and a date and time. Matches are always between two different teams and the outcomes are recorded. Each match is refereed by a coach that is not involved in that specific match.

Part 1: ER model

Create a ER model based on the database description. Make a registrant entity that is a superclass of players and coaches. Give registrants unique IDs.

For each entity you create, ask yourself:

- what are its attributes?
- what attributes can make up its key?
- is the entity weak?
- is the entity a subentity of another entity?
- what relationships does it have with other entities?
- are any attributes multivalued?

For each relation you create, ask yourself:

- what are the cardinality/participation constraints?
- is this an identifying relationship?

Come up with at least one derived attribute and include it in your ER model.

Part 2: Relational model

Create a relational schema based on your ER model. For each constraint implied by your ER model, ask yourself:

- is the constraint captured in your relational design? (if nor, briefly explain why)