University of Toronto
CSC343, Fall 2015

Assignment 1

Due: Oct 16, at 10:00 pm sharp!

Introduction

For this assignment, you will write queries and relational constraints in relational algebra on a database for the Pan Am games of 2015. We are providing the schema for you. Later in the course, you will learn about how to develop your own schema based on knowledge of the domain. Even though developing a schema is really the first step in building a database, it is a more advanced task than querying an existing database; this is why we will be learning about it and practising it later.

Schema

The schema below represents data for Pan Am games 2015, where people can buy tickets for matches they would like to attend, and the results are recorded as events are completed. The schema attempts to represent the domain well, but in order to keep the assignment simple, some things are simplified and other details are not represented. For example, the fact that some sporting disciplines are team events, while others are individual competitions is not represented. Each athlete, whether competing in an individual event or a team competition, is awarded a medal for finishing in the top 3 (so multiple athletes can get a gold medal in the same event, if it happens to be a team event). Remember that all your queries will be written with respect to the schema below.

Relations

- Country(CID, cname)
  A tuple in this relation represents a country that participates in the competition. *CID* is the country id, *cname* is the country name.
- Athlete(AID, fname, lname, sport, CID, gold, silver, bronze)
  A tuple in this relation represents an athlete who is participating in the Pan Am games. *AID* is the athlete's ID, *fname* is their first name, *lname* is their last name, *sport* is the general sporting discipline the athlete is qualified to compete in (for example: athletics, swimming, fencing, diving, boxing, etc. you can assume no athlete is qualified for more than one discipline), and *CID* is the country they are representing (each athlete only represents one country). Finally, *gold/silver/bronze* represent the number of medals of each type won so far by that athlete.
- Stadium(SID, sname, address, capacity)
  A tuple in this relation represents a stadium where a match takes place. *SID* is the stadium's ID, *sname* is the stadium name, *address* represents its location within the city of Toronto, *capacity* is the capacity of the stadium (number of seats).
- Event(EID, date, time, sport, description, SID)
  A tuple in this relation represents a sporting event that took place or will be taking place in future. *EID* is the id of the event, *date* is the date the event is scheduled on, *time* is the time of the event, *sport* is the sporting discipline of the event, and *description* is the type of event in that sport (e.g., sport is athletics, description is 100m sprint). The *SID* is the id of the stadium where the event takes place.

- Ticket(TID, dateIssued, timeIssued, EID)
  A tuple in this relation represents a ticket that was purchased, (regardless if the event has taken place yet or not - ticket purchases can be made in advance). *TID* is the ticket's id, *dateIssued* is the date that it was purchased, *timeIssued* is the time that it was purchased, and *EID* is the EID of the event it was purchased for.
- Result(EID, AID, medal)
  This relation keeps track of all the results of completed events so far. A tuple in this relation represents the fact that a medal has been won in a specific event by a specific athlete. *EID* is the event id, *AID* is the athlete competing in this event, and *medal* is the medal type (gold, silver, bronze, or nomedal) which the respective athlete won in the competition.

Integrity constraints

- Athlete[CID] $\subseteq$ Country[CID]
- Event[SID] $\subseteq$ Stadium[SID]
- Ticket[EID] $\subseteq$ Event[EID]
- Result[EID] $\subseteq$ Event[EID]
- Result[AID] $\subseteq$ Athlete[AID]
- Π_{medal}Result $\subseteq$ {"gold", "silver", "bronze", "nomedal"}

Part 1: Queries [84% - 7 marks each]

Write the queries below in relational algebra. There are a number of variations on relational algebra, and different notations for the operations. You may only use the operations defined in Section 2.4 of the text and you must use the same notation as we have used in class and on the slides. You may use assignment, and the operators we have used in class: $\Pi, \sigma, \bowtie, \bowtie_{condition}, \times, \cap, \cup, -, \rho$. Assume all relations are sets (not bags), and do not use any of the extended relational algebra operations from Chapter 5 (for example, do not use the extended projection). Some of the queries cannot be expressed in the language that you are using. In those cases, simply write "cannot be expressed".

Additional points to keep in mind:

- Do not make any assumptions about the data that are not enforced by the original constraints given in the schema above. Your queries should work for any database that satisfies those constraints.
- Assume that every tuple has a value for every attribute. (For those of you who know some SQL, in other words, there are no null values.)
- Remember that the condition on a select operation may only examine the values of the attributes in one tuple (not whole columns), and that it can use comparison operators (such as $\leq$ and $\neq$) and boolean operators ($\vee$, $\wedge$ and $\neg$). Two relations in our schema have a date attribute. You may use comparison operators on such values. You may refer to the year component of a date d using the notation d.year.
- You are encouraged to use the assignment operator (:=) to define intermediate results, and it is a good idea to add commentary explaining what you're doing. This way, even if your final answer is not completely correct, you may receive part marks.

- The order of the columns in the result doesn't matter.
- When asked for a maximum or minimum, if there are ties, report all of them.

Write queries for each of the following:

1. Find the last names of the athlete(s) of the country(ies) that did not compete in any event yet.
2. Find the last names of the athlete(s) of the country(ies) that did not win any medals yet (either because they did not compete, or because their athletes did not rank in the top 3 in any event so far).
3. Find the stadium names of all the stadiums where exactly one event took place.
4. Find all the sporting disciplines that Canadian athletes have competed in so far.
5. Find the first and last name of the athletes whose sporting discipline is "swimming" and who have won the highest number of gold medals among all athletes who compete in the same sport.
6. Find the name of every country that has won at least one of every type of medal (gold, silver, and bronze).
7. Find the gold medalist country of the event for which the very first ticket out of all the tickets in the database was purchased.
8. Find the first and last name of the athlete representing "Mexico", who so far has the second highest number of gold medals (among athletes of the same country).
9. Find the sports disciplines for events for which at least two tickets were bought on the date of the event.
10. Find the athlete with the highest overall number of gold medals won so far, and report that athletes first and last name, country name, and number of gold medals won.
11. Find the discipline (sport) of the event for which the highest number of tickets was purchased.
12. Find the first and last name for all athletes who have won a gold medal in an event for which no tickets were sold.

Part 2: Additional Integrity Constraints [16% - 4 marks each]

Below are some additional integrity constraints on our schema. Express each of them using the notation from Section 2.5 of your textbook. If a constraint cannot be expressed using such notations, simply write "cannot be expressed".

1. An athlete cannot win more than one medal type in the same event.
2. All tickets for an event have to be purchased before the time of the event.
3. The number of tickets purchased for an event should not exceed the capacity of the stadium where the event takes place.
4. An athlete could not have competed in an event for a sporting discipline that they are not qualified to participate in.

Submission instructions

Your assignment must be typed; handwritten assignments will not be marked.

You may use any word-processing software you like. Many academics use LaTeX. It produces beautifully typeset text and handles mathematical notation well. Whatever you choose to use, you need to produce a final document in pdf format, and you must call it "a1.pdf" (with no capital letters).

You are allowed, and in fact encouraged, to work with a partner for this assignment. You must declare your team (whether it is a team of one or of two students) and hand in your work electronically using the MarkUs online system. Instructions for doing so are posted on the Assignments page of the course website. Well before the due date, you should declare your team and try commiting an empty a1.pdf file in your svn repository.

For this assignment, hand in just one file: "a1.pdf". Once you have submitted, be sure to check that you have submitted the correct version; new or missing files will not be accepted after the due date (or after grace tokens have been used up).

For more submission instructions, please consult the Assignment page on the course website.