Homework #1 Solutions

Database Systems, CSCI-4380-01

Homework Description

Question 1. Suppose you are given the following relations. Write down tables that correspond to these relations in SQL with CREATE TABLE statements by listing each attribute with an appropriate data type, listing all keys by primary key constraints and giving the appropriate rationale (you can assume names are unique for this homework to simplify the question). The length of character strings is not important, use varchar(255) for all.

```
CREATE TABLE person (
        name varchar(255),
        dateOfBirth date,
        PRIMARY KEY (name)
    )
```

Each person has a unique name but multiple people may be born on the same day.

```
CREATE TABLE club (
        name varchar(255),
        dateFounded date,
        PRIMARY KEY (name)
    )
```

Each club has a unique name but clubs may be founded on the same day.

```
CREATE TABLE clubMember (
        personName varchar(255),
        clubName varchar(255),
        PRIMARY KEY (personName, clubName)
    )
```

Each person has a unique name and each club has a unique name. Each person can only be a member of each club once.

```
CREATE TABLE clubActivity (
        clubName varchar(255),
        title varchar(255),
        date date,
        time time,
        PRIMARY KEY(clubName, title, date)
    )
```

An activity should only occur once a day for each club but the time that an activity occurs can change from one day to the next (as meeting times commonly do). If, however, a club wishes to meet for a different activity on the same day (say board games wants to play *Battlestar Galactica* at 4pm then *Arkham Horror* at 6pm both on 8-01-2010, the same day as *Carcassone*) then this should be (and is) allowed.

Question 2. Assuming the above data model and instance, for each query below, write down (1) what is a good English sentence that describes what the query is returning, (2) what is the result of the query (i.e. which attributes and tuples are in the result)?

a. $(\Pi_{name}\ Club) - (\Pi_{name}(Club \bowtie_{name=ClubName \text{ and } date>dateFounded}\ ClubActivity))$

The name of any club that has not had an activity since its founding.

Result:

name
skiing

b. $Person \bowtie_{name<>n1 \text{ and } dateOfBirth=d1}\ (\rho_{P1(n1,d1,p1,a1)}(Person))$

Records of any persons born on the same date but having different names.

Result:

n1	d1	p1	a1
joe	01-01-1985	jane	01-01-1985

Question 3. Write the following queries using relational algebra assuming the above data model:

a. Find the name of all people who are part of a club founded after 2010.

What "after 2010" means is ambiguous. I give two answers based on the two most obvious interpretations.

If "after 2010" means "after the year became 2010" then the query is:
$(\Pi_{personName}(\sigma_{dateFounded>=01-01-2010}(club \bowtie_{name=clubName} clubMember)))$

If "after 2010" means "after the end of the year 2010" the query is:
$(\Pi_{personName}(\sigma_{dateFounded>=01-01-2011}(club \bowtie_{name=clubName} clubMember)))$

b. Find people who are born after 1980 and are member of a club. Return all the attributes of the people table.

$N := \Pi_{personName}(clubMember)$;;`All members of any club`
$Y := \sigma_{dateOfBirth>=01-01-1980}(person)$;;`All people born after 1980`
$\Pi_{name,dateOfBirth}\ (N \bowtie_{N.personName=Y.name}\ (Y))$;;`Extract members of Y from N`

c. Find all clubs that have no members and have no activities. Return their name.

$C(clubname) := \Pi_{name}(club)$;;`Names of all clubs`
$A := \Pi_{clubname}(clubMember \bowtie_{clubMember.clubName=clubActivity.clubname} clubActivity)$;;`Names of all clubs that have members and activities`
$C - A$;;`Clubs with no members and no activities`