

CSE 532: Project 1 - Dating Services Support System

Spring 2006

Deadline: February 27

In this project, you will design and implement database support for an electronic dating system. You will be using the Datalog language supported by the XSB system (<http://xsb.sourceforge.net>). The lecture slides give a brief introduction to the use of XSB's Datalog. To simplify the matters, you are not required to build any GUI or store data in a real database. You can keep the data in a file along with the rules for defining views and the intermediate predicates used in your queries.

1 General Description

The Dating Services Support System (DSSS) is intended to represent information about the clients of the dating service; their preferences; their past and present dates; and queries that are used as part of running the service. The system has two main functions:

1. Enable the clients to find potential “matches” based on preferences and features.
2. Enable analytical querying by the DSSS personnel (presumably in order to work our better algorithms and make the matching process more accurate).

2 System Users

There are two kinds of users: DSSS clients and DSSS personnel. In this project you do not need to build any GUI and you do not need to support security and user account features. You only need to distinguish between DSSS clients who are looking for partners and DSSS personnel.

3 Required Data

The data items required by your system roughly fall into these categories:

- *Information about DSSS clients and their preferences.*
- *Information about the DSSS personnel.*
- *Information about the past and present dates of the clients.*

Note that DSSS personnel can also be clients, and the database schema should reflect that fact. Every person in the system has

- Name
- Address
- Login information

A client, in addition, has

- A set of preferences (the features that the client wants the potential dates to have). For instance, preferences can include sports, hiking, reading, dance, tall, short, slim, etc.

- A set of client’s own features. A client’s feature can be an interest (e.g., hiking, fishing) or a physical feature (e.g., tall, slim, blond).
- The set of other DSSS clients whom this client was dating in the past. This information should include the time intervals when dating was taking place (for instance, March 7, 2005 to June 22, 2005).

For simplicity, in this project you can represent time with integers of the following form: YYYYMMDD. For instance, 20060213 for February 13, 2006. This way you can do date comparison by simply using $<$ and $>$ for integers.

- Information about DSSS clients whom this client is currently dating. This should indicate when dating began.

4 Queries

You are to implement the following queries.

1. List all clients whom Joe Public was dating between 20050307 and 20050622 (does not mean every day).
2. List all clients whom Joe Public was dating as of 20050307 and such that they had at least one common feature with Joe.

Here, dating a certain person as of 20050307 (or as of some other day) means that Joe started dating that person at time prior or equal to 20050307 and has not stopped dating on 20050307 or before.

3. Same as above, but now we want Joe’s dates to satisfy at least two of his preferences.
4. Find all potential “perfect matches” for Joe Public.

A perfect match is a DSSS client such that the set of all Joe’s preferences is included in the set of that client’s features.

This query involves a non-trivial use of negation like the ones in the lecture slides and the textbook.

5. Find Joe Public’s *indirect* dates as of 20050307.

An indirect date at time T , is either a person whom Joe was dating directly at T ; or a person who was dating another person, p , at T and p was dating Joe directly at that time; or a person who was dating p_1 at T , who was dating p_2 , who was dating ..., etc., who was dating Joe.

Note that this is a recursive query.

5 Documentation and Submission Instructions

Your project should be accompanied by a short document, which should appear at the top of the file between the comment markers `/* ... */`. The document should detail the database schema used for the project (i.e., relation names, the meaning of the columns, and their types). Also, state what you expect to be the keys in each relation. You do not need to express them in Datalog (we did not discuss the representation of constraints in this language).

It is likely that in order to express the last two queries, which are relatively involved, you will need to include rules to define some intermediate relations. In some cases, the intent of those rules may not be obvious and you would have to explain their meaning with a comment line or two.

Pay attention to the aesthetics. Poorly formatted or designed works will be penalized.

You will submit the documentation to the TA (Zhiquan Gao, zgao@cs.sunysb.edu) via email. The data, the rules, and the queries should be in one **text** (not Word!) file, which should be sent as an attachment. The file must be directly loadable and executable in XSB without any editing.

At the top of the file (in a comment block) include your name, student Id, email, and this statement:

I pledge my honor that all parts of this project were done by me individually and without collaboration with others.

6 Teaming

This project must be done **individually** — no partners.

7 Planning Your Work

This is not a hard project, but, since it is likely that Datalog is new to you, you will encounter numerous problems trying to get things right. Therefore, start right away and do not delay. There will be no deadline extensions.

We will provide an initial test data set. However, you might need to add more information to ensure that your queries have enough data to work with.