

1 Authors and Links

The authors for this project are Joseph Heenan and Zainul Abi Din. The application may be found at: <http://resin.cci.drexel.edu:8080/~zad23/index.html>

2 Project Report

We created a dataset and relational schema, along with a web application, which allows a user to explore relationships between various entities in a fictional social network. Originally we planned for dynamic data acquisition and a "fakeness detection" scoring logic but as we were advised not to load large amounts of data on tux, we changed the application to be focused instead on querying a small, fictional dataset close to the Facebook Graph API data model and allowing for a robust number of queries. No insert or update queries are supported, mitigating the risk of SQL injection.

3 List of Business Rules

Many users may exist. A User may have none, 1 or many Education Experiences. Each Education Experience relates to exactly one User (key and participation constrain on Education Experience to User) A User may have none, 1 or many Work Experiences. Each Work Experience relates to exactly one User (key and participation constraint on Work Experience to User).

Many Groups may exist. Each Group is owned by exactly one user (key and participation constraint on Groups to Users). Note: only the "owns" relation is stored in this database, e.g "User1 owns Group1" can be modelled, by "User1 is a member of Group1" cannot.

Many Pages may exist. Each Page is owned by exactly one User (key and participation constraint on Page to User)

Many User Posts, Group Posts, and Page Posts may exist. Each "User Post" is associated with exactly one user (key and participation constraint) Each "Group Post" is associated with exactly one group (key and participation constraint) Each "Page Post" is associated with exactly one page (key and participation constraint)

4 How ER Diagram was Translated to Relational Database

We began by modelling the core User entity as its own "user_" table. Because "User Posts", "Group Posts" and "Page Posts" are owned by distinct users, groups and pages, respectively, we modeled these in a "user_posts", "group_posts" and "page_posts" table. Because education and work experiences have a key and participation constraint such that they map to exactly one user, we

created each with a `user_id` column that is a foreign key to the primary key (`user_id`) for the relevant user. Finally for modelling who owns a page and a group, we created table `Owns_Page`, which stores the information when known as to which User owns each Page. Furthermer we created a table `Owns_Group`, which stores the information when known as to which User owns each Group.

5 Data Acquisition Approach

Because we are modelling a fictional social network we looked at representative data from the Facebook social network and created a fictional dataset, which is loaded via statements in our `schema.sql` file.

6 User Interface

The user interface begins with a "show all" section which will allow a user to view all entities of a particular type. A user may see all Users, all Group Posts, etc., to get a feel for the underlying dataset.

Then a "Search By User Name" section is present. This functionality allows a user to find multiple users who may have the same name, and also allows a user to find the groups, pages, posts, work and education experiences owned by the relevant user(s). This is implemented via appropriate JOINS between the relevant tables.

In the third section "Search for Page Posts by Page Name" is present. This functionality allows a user to find posts made by one or more Page entities with a particular name.

Finally a "Search for Group Posts by Group Name" section is present. This functionality allows a user to find posts made by one or more Group entities with a particular name.