Group 16 - COMP 421 - Database Systems
Demetrios Koziris (260584555)
Sami Riaz (260630430)
Brennan Nichyporuk (260504951)
Vivien Traineau (260623886)

Project 1: Database Design and Data Modelling

I. Requirements Analysis

1. Introduction

The application is a social media platform that can be compared to the likes of Facebook or Twitter. It is however meant to be based solely on the interactions between the users without any unnecessary additions. It allows the website to be light and efficient, using little data while allowing users to communicate just as well as any other social media website. With a core set of features that we believe are sufficient to give the users a great experience, we intend to stay away from the data collection of the users' habits, the prediction of their actions and the personalization of their feeds.
In summary, our application is a minimalistic social media with all the benefits of current social medias, but keeping in mind both the importance of their privacy and the idea that spending more time than necessary on social media is not healthy.

2. Database Description

Entity sets:
Creator: A creator is-a user or is-a page. A creator has a unique handle. A creator has exactly one wall. A creator can optionally schedule one or more events. A creator can create a post on a wall.
User: A user is-a subcategory of the more general creator entity. A user has a first name, a last name, a gender, a birthdate, a email, and a password. A user can manage a page or can follow a page (or both). A user can subscribe to an event by RSVPing 'going', 'maybe', or 'not going'. A user can befriend another user. A user can like a submission. A user can comment on a post. A user can view a feed of posts. A user can get notifications.
Page: A page is-a subcategory of the more general creator entity. A page has a title, a description, and number of followers. The number of followers attribute is present for performance reasons. A page is managed by exactly one user. A page can be followed by any number of users.
Submission: A submission is-a post or is-a comment. A submission has text, a datetime, a number of likes, and a submission_id. A submission can be liked by a user. The number of likes attribute is present for performance reasons
Post: A post is-a subcategory of the more general submission entity. A post can additionally contain an image. A post can link to an attachment.
Comment: A comment is-a subcategory of the more general submission entity. A comment is associated with exactly one user and post.
Wall: A wall is related to a single creator or a single event (but not both). A wall has a description and a wall_id.

Event: An event is scheduled by a creator. An event has exactly one wall. An event has a location, a description, a title, a start time, a end time, a creation time, and an event_id.
Notification: A notification is related to a single user. A notification has text, a link, datetime, a read_status, and a n_id.

Relationships
has: an event has a wall. This is a one-to-one relationship because an event is associated with exactly one wall, and a wall is associated with exactly one event.
schedules: a creator schedules an event. This is a many-to-one relationship because a creator can schedule many events, but an event is associated with exactly one creator.
owns: a creator owns a wall. This is a one-to-one relationship because an creator is associated with exactly one wall, and a wall is associated with exactly one creator.
subscribes: a user subscribes to an event. This is a many-to-many relationship because a user can subscribe to multiple events, and a event can have multiple users subscribe.
creates: a creator creates a post on a wall. A many-to-many relationship exists between a creator and wall since a creator can post on multiple walls and since a wall can be posted on by multiple creators. A participation/key constraint exists on a post since a post is related to exactly one wall.
befriends: a user can befriend another user. This is a many-to-many relationship since a user can have many friends.
manages: a user manages a page. This is a many-to-one relationship since a user can manage many pages, but a page is managed by exactly one user.
follows: a user follows a page. This is a many-to-many relationship since a user can follow many pages and a page can be followed by many users.
likes: a user can like a submission. This is a many-to-many relationships because a user can like multiple submissions and a submission can be liked by many users.
gets: a user gets a notification. This is a many-to-one relationship because a user can get many notifications, but a notification is associated with exactly one user.
comments on: a user can comment on a post. A many-to-many-to-one relationship exists since: a user can comment on a post multiple times, a post can be commented on by multiple users, but a comment is associated with a single user / post.
views feed of: a user views a feed of posts. This is a many-to-many relationships since a user can view multiple posts in a feed, and since a post can be on many users feed.

II. E/R Diagram

Please see png file provided in assignment submission.
Constraints not expressed in the E/R Diagram

- Covering Constraint for Creator ISA hierarchy (i.e. every entity of Creator entity set must be one of the subclass entity sets User or Page) is not expressed.
- Covering Constraint for Submission ISA hierarchy (i.e. every entity of Submission entity set must be one of the subclass entity sets Post or Comment) is not expressed.
- A Wall entity should have exactly one association with either an Event entity or with a Creator entity (but not both) but this participation constraint cannot be not expressed in the E/R Diagram.
- A User should not be able befriend themselves but this constraint is not expressed in the E/R Diagram.

III. Relational Model

Entity Sets

Creator(handle)
User(user_handle, display_name, email, firstname, lastname, gender, birthdate, wall_id) (user_handle ref Creator(handle)) (wall_id ref Wall)
Page(page_handle, description, title, follower_count, user_handle) (page_handle ref Creator(handle)) (user_handle ref User)
Submission(submission_id, like_count, datetime, text)
Comment(comment_id, user_handle, post_id) (comment_id ref Submission(submission_id)) (user_handle ref User) (post_id ref Post)
Post(post_id, view_count, attachment_link, image, handle, wall_id) (post_id ref Submission(submission_id)) (handle ref Creator) (wall_id ref Wall)
Event(event_id, handle, wall_id) (handle ref Creator) (wall_id ref Wall)
Wall(wall_id, description)
Notification(notif_id, read_status, link, text, datetime, user_handle) (user_handle ref User)

Relationships

PageFollower(user_handle, page_handle) (user_handle ref User) (page_handle ref Page)
EventSubscription(user_handle, event_id, rsvp_status)(user_handle ref User)(event_id ref Event)
SubmissionLike(user_handle, submission_id) (user_handle ref User) (submission_id ref Submission)
FeedView(user_handle, post_id) (user_handle ref User) (post_id ref Post)
UserFriend(user_handle, friend_handle) (user_handle ref User**)** (friend_handle ref User(user_handle)**)**

Constraints not expressed in the Relational Model

- Participation constraint of Event in association with Wall is only enforce if foreign key ‘wall_id’ in Event is NOT NULL.
- Participation constraint of Creator in association with Wall is only enforced if foreign key ‘wall_id’ in Creator is NOT NULL.
- Participation constraint of Event in association with Creator is only enforced if foreign key ‘handle’ in Event is NOT NULL.
- Like in the E/R model, the participation constraint of the Wall is not enforced since it’s possible to create a Wall entity that is not referred to by exactly one of either an Event or a Creator.
- Participation constraint of Notification in association with User is only enforced if foreign key ‘user_handle’ in Notification is NOT NULL.
- Participation constraint of Post in association with Creator and Wall is only enforced if foreign key ‘handle’ and ‘wall_id’ in Post are both NOT NULL.
- Participation constraint of Comment in association with User and Post is only enforced if foreign key ‘user_handle’ and ‘post_id’ in Comment are both NOT NULL.
- A constraint keeping a User from being in a UserFriend relationship with themselves is not enforced by the Relational Diagram (friend_handle and user_handle would have to be guaranteed to be different).
- Covering Constraint for Creator (i.e. every entity of Creator entity set must be one of the subclass entity sets User or Page) is not expressed.

- Covering Constraint for Submission (i.e. every entity of Submission entity set must be one of the subclass entity sets Post or Comment) is not expressed.

Creativity and Complexity

- We have already combined several relations in order to reduce redundancy and enforce key constraints. For example, we do not have a seperate table for the association between Creator, Post, and Wall (Creator posts a Post on a Wall). Instead, each Post has a foreign key ‘handle’ referring to a Creator, and a foreign key ‘wall_id’ referring to a Wall. Likewise we do not have a seperate table for the association between User, Comment, and Post (User posts a Comment on a Post). Instead, each Comment has a foreign key ‘user_handle’ referring to a User, and a foreign key ‘post_id’ referring to a Post. This allows us to enforce the key constraint of Post in the above mentioned ternary relationship and the key constraint of Comment in the above mentioned ternary relationship. (The participation constraints of each are only enforced if the foreign keys are guaranteed to be not null, as mentioned above)
- These ternary relationships as well as our ISA hierarchy relations help our design stand out and provide the functionality we would want out of a minimalist social media site.
- We do not feel there is anyway to further combine any relations. For example, if we were to merge PageFollower, EventSubscription, SubmissionLike, or FeedView with User, it would no longer be possible for those associations to be many to many (i.e. many to many for a follow association between a User and a Page, a subscription association between a User and an Event, a like association between a User and a Submission, or a view association between a User and a Post).

Websites that inspired our design

Facebook and Twitter