University of Puerto Rico - Mayagüez
Electrical and Computer Engineering
Department

ICOM 5016/ CIIC 4060
Introduction to Database Systems

Term Project – Backend System for Twitter-Style Service
Phase II
Initial System Implementation

Andrea C. Mirada Acevedo
Emmanuel Nieves González
Jean C. Rivera Hernández
Carlos M. Santiago López

Monday April 12, 2021

Mapping ER to Tables:

Entities:

- **Users**

  Since Users is as entity a table Users is created that contains parameters: uid which is users unique id, first_name, last_name, email, username and password.

  -create table Users(uid serial primary key , first_name varchar(20) , last_name varchar(20),email char(54),username char(15),password char(15));
- **Messages**

  Since Messages is an entity, a table is created with parameters: Mid which is messages unique id, message which is the message that is written by the user, date which is the date of when the message was written.

  - create table Messages(mid serial primary key,mdate char(10), message varchar(20),uid integer references Users(uid));

Relationships:

One to Many:

- The relation Posts is a one to many relationship between Users(one side) and Messages(many side), where a user posts a message and even if the same user posts the same message their Id's are different. Since post/messages cannot exist if they are no messages that means that messages has total participation. In this case a table is not created, but the one side primary key uid from Users is passed to the Messages table as a foreign key. The table is shown in Messages**.**

Many to Many:

- The relation Shares is a many to many relationships between Users and Messages since a share is done by a user and what is sharing is a message. Since is many to many a table for Shares is created with

parameters uid that references Users id and mid that references Messages mid and shares unique id sid.
- create table Shares(sid serial primary key,uid integer references Users(uid),mid integer references Messages(mid));

- The relation Reply is a many to many relationships between Users and messages where a user’s can reply to a message from other user or even the same user. Since is many to many a table Reply is created with parameters: rid which is the table unique id, uid which references Users uid, mid which references Messages mid and rmessage which is the message that the user replies to the message.
  - create table Reply(rid serial primary key, rmessage varchar(240), uid integer references Users(uid), mid integer references Messages(mid));
- The relation Follows is a many to many self-relationship of Users since a user follows another user. Since is many to many a table is created with parameters fid it’s unique id, followingid that refers to Users uid as the person who is following and the other parameter followerid which references to Users uid too, but this one to signify the one who is the follower.
  - create table Follows(fid serial primary key, followingid integer references Users(uid), followerid integer references Users(uid));
- The relation Blocks is a many to many self-relationship of Users similar to Users since a user blocks another user. A table is created with parameters: bid which is the unique id, blockingid which references to Users ui in the meaning the one that will be blocked and uid which references to Users uid referencing the one that is blocking.
  - create table Blocks(bid serial primary key, blockingid integer references Users(uid), uid integer references Users(uid));
- The relation Reacts is a many to many relation, therefore a table was created. The parameters were rid which is the unique key, uid that refers to Users uid to get track of who gave the reaction, the mid that refers to Messages mid to get track of the message and isLiked a boolean attribute that if it’s true is a like, if it’s false is a unlike.

- create table Reacts(rid serial primary key, uid integer references Users(uid), mid integer references Messages(mid), isLiked boolean);

Changes to ER Diagrams:

- Reply was changed to many to many of users and messages, because during implementation it made more sense that a user replies to a message, instead of a self-relation of message.
- Some attributes like join_date of Users and mdate for Messages were removed, because there are not of a lot of use for this implementation.
- Reacts a relation between Users and Messages was changed from one to many to many to many, since a user can react to a message and a message can have different reactions from other users.
- The Explanation of the tables done were the ones used for this phase 2.