Received Feedback

“I don’t see the userName attribute listed in the database outline.” -Hannah Lee

“For instance, I don’t see the totalFollowers and dateFollowed attributes from the database outline illustrated in the UI. Although this is a small detail, I noticed is that the followerID is spelled differently in the ERD, followersID. I also noticed these differences in the other tables when comparing to the schema and database outline”. -Hannah Lee

“The missing tables that are listed in the schema are ProfileEngagement, PostEngagement, as well as the intersection tables”. -Hannah Lee

“So instead of INSERT INTO (userID, userName, dateCreated, email), you could just need INSERT INTO (userName, dateCreated, email)” Hannah Lee

“Similar to the SELECTs, the ProfileEngagements, PostEngagements, UserFollowing, and UserFollowers do not exist yet and by default are missing INSERTS in both html and DML. The pages and DML will need to be created for each of these with INSERTS” - Vincent Nguyen

“For the ProfileEngagements and PostEngagements we could have 0 engagements in the beginning, so all the total attributes could be NULLABLE.” -Vincent Nguyen

Project Step 3 Draft
CS_340

Group Name: Team 86
Group Members: Giovanita Bell and Kevin Lee
Database Title: BeaverGram Statistics

Overview

BeaverGram is the next popular social media platform started by and created for Oregon State University students. The goal of this database is to track a user's account and to determine the user's engagement with other users/followers. BeaverGram is designed to have users post images/text/videos that other users can interact with using likes/comments. Users can follow other users to view posts from each other.

This database for this platform is focused around finding the average audience engagement statistics, by tracking a user's audience engagement.
The database is meant to track how frequently an audience engages each individual post and the user's profile. This allows a user to track any fluctuations in popularity.

Example:
A user on BeaverGram has the following data from their profile that is meant to be entered into the database:

- 160,000 followers
- 200 following
- 900,000 likes
- 2.6 M views
- 6 M profile views

With this data, the database is able to find the following:

- Average, max, and min number of likes per post
- Average, max, and min number of comments per post
- Average, max, and min number of gained followers per month
- Average, max, and min number of lost followers per month
- Average, max, and min number of profile views
- Ratio of followers to following
- Comprehensive total of likes and comments for a user
- Rate of change and trends of user's profile engagement from followers
    - Will be able to determine if comments, views, and likes have increased or decreased compared to previous month
    - Will be able to identify the most active months within a year

<u>Database Outline</u>

Entities and Attributes

- Users - Retains the user's information
  - userID: int, auto_increment, UNIQUE, not NULL, PK
  - userName: varchar(50), UNIQUE, not NULL
  - dateCreated: date, not NULL
  - email: varchar, not NULL

- Followers - Retains the number of followers the user has, and helps identify the frequency of followers gained or lost
  - followersID: int, auto_increment, unique not NULL, PK
  - dateFollowedChange: date, not NULL
  - isGainedFollower: tinyint(1)
  - followerCount: int, auto_increment, not NULL

  - Relationships:
    - M:N - Followers(PK) to User(FK)

- Following - Retains the number of other accounts followed by the user
  - followingID: int, auto_increment, unique not NULL, PK
  - totalFollowingAccount: int, auto_increment, not NULL
  - dateFollowing: date, not NULL
  - Relationships:
    - M:M - Following(PK) to User(FK)

- Profile Engagement - Tracks the average and total engagement of a user profile, by combining total likes and comments for the entire page
  - profileEngagementID: int, auto_increment, unique, not NULL, PK
  - totalPosts: int, auto_increment, not NULL
  - totalLikes: int, auto_increment, not NULL
  - totalViews: int, auto_increment, not NULL
  - totalComments: int, auto_increment, not NULL
  - Relationships:
    - 1:M - Profile Engagement (PK) to User (FK)

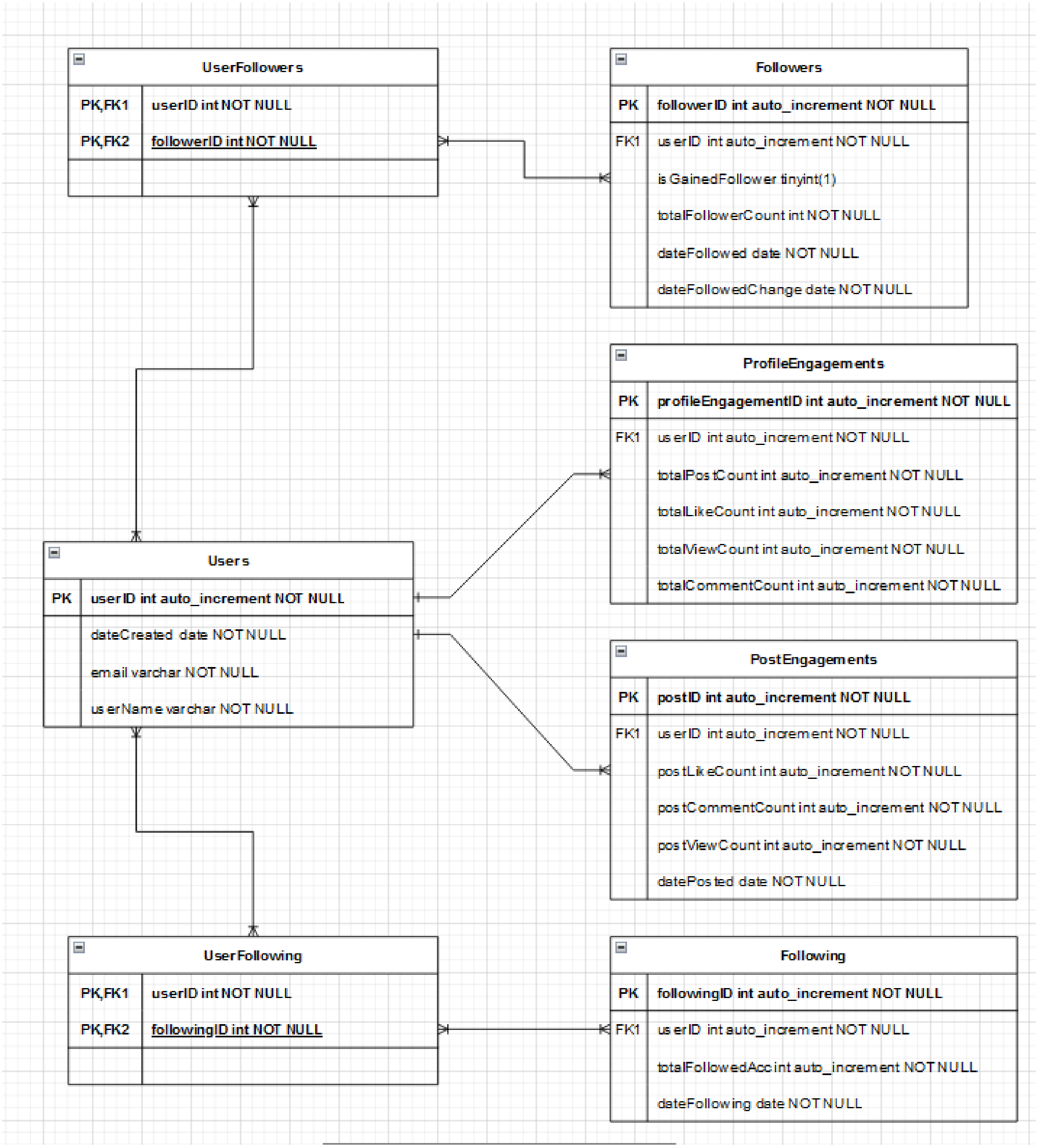
- Post Engagement - Tracks the average and total engagement for individual posts. This enters the number of likes for each post, as well as the average number of likes for all posts
    - postID: int, auto_increment, unique, not NULL, PK
    - totalPostLikes: int, auto_increment, not NULL
    - totalPostComments: int, auto_increment, not NULL
    - totalPostViews: int, auto_increment, not NULL
    - datePosted: date, not NULL
    - Relationships:
        - 1:M User (PK) can have multiple posts

Intersection Tables

The following entity-relationships will have intersection tables:

- User-Following:
    - Intersection table will have userID and followingID as foreign keys
    - 1:M between Users and User-Following
    - 1:M between Following and User-following
- User-Followers:
    - Intersection table will have userID and followersID as foreign keys
    - 1:M between Users and User-Followers
    - 1:M between Following and User-followers
- User-Posts:
    - Intersection table will have userID and postID as foreign keys
    - 1:M between Users and User-Posts
    - 1:M between PostEngagements and User-Posts

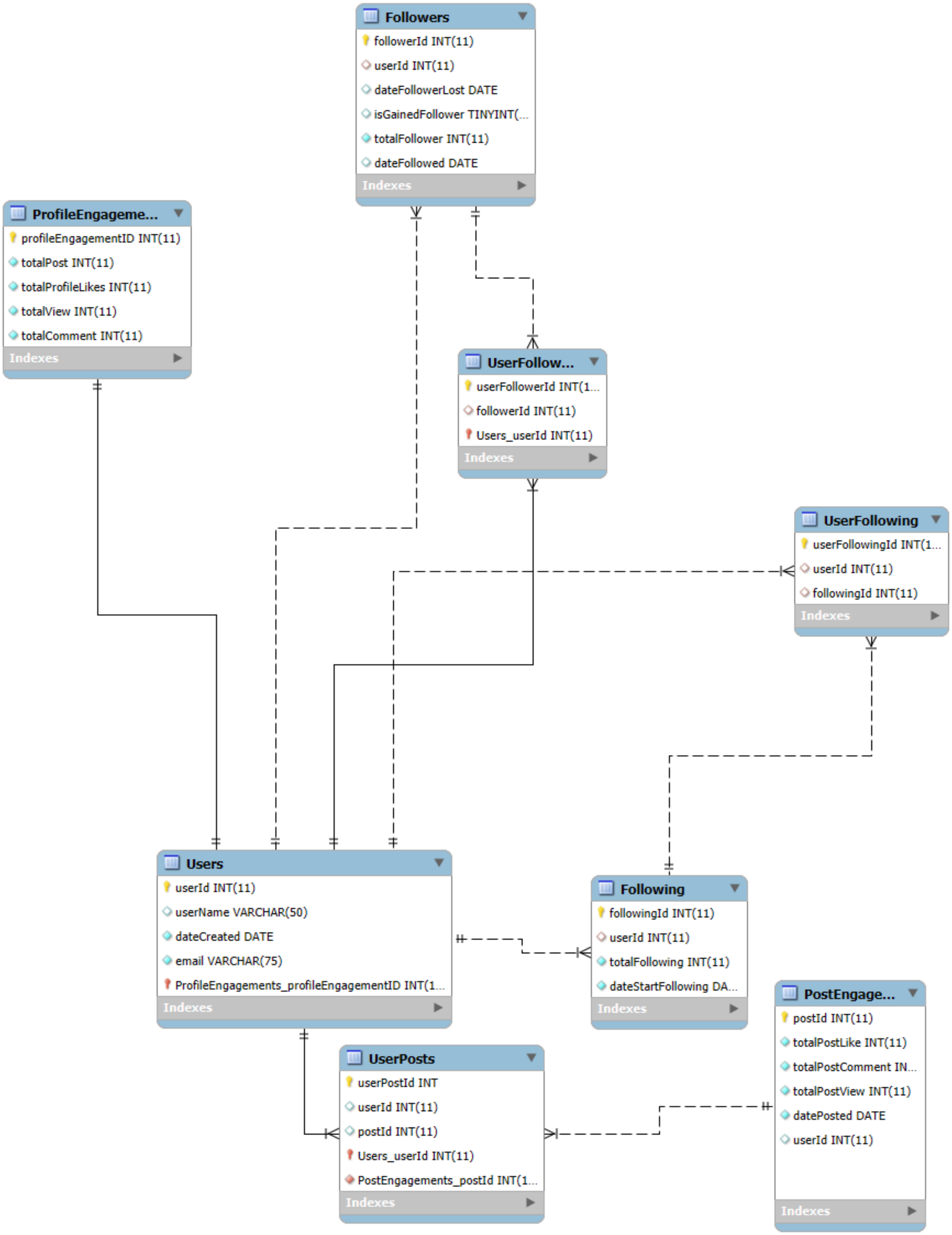
ERD Diagram


UserFollowers
PK,FK1 userID int NOT NULL
PK,FK2 followerID int NOT NULL
Followers
PK followerID int auto_increment NOT NULL
FK1 userID int auto_increment NOT NULL
isGainedFollower tinyint(1)
totalFollowerCount int NOT NULL
dateFollowed date NOT NULL
dateFollowedChange date NOT NULL
ProfileEngagements
PK profileEngagementID int auto_increment NOT NULL
FK1 userID int auto_increment NOT NULL
totalPostCount int auto_increment NOT NULL
totalLikeCount int auto_increment NOT NULL
totalViewCount int auto_increment NOT NULL
totalCommentCount int auto_increment NOT NULL
Users
PK userID int auto_increment NOT NULL
dateCreated date NOT NULL
email varchar NOT NULL
userName varchar NOT NULL
PostEngagements
PK postID int auto_increment NOT NULL
FK1 userID int auto_increment NOT NULL
postLikeCount int auto_increment NOT NULL
postCommentCount int auto_increment NOT NULL
postViewCount int auto_increment NOT NULL
datePosted date NOT NULL
UserFollowing
PK,FK1 userID int NOT NULL
PK,FK2 followingID int NOT NULL
Following
PK followingID int auto_increment NOT NULL
FK1 userID int auto_increment NOT NULL
totalFollowedAcc int auto_increment NOT NULL
dateFollowing date NOT NULL

Example Data

```
INSERT into Users (userName, dateCreated, email)
VALUES
('Benny Beaver', '2020-01-01', 'bennybeaver@osu.com'),
('Moo Deng', '2020-01-01', 'moodeng@osu.com'),
('Pesto Penguin', '2020-01-01', 'pestopenguin@osu.com');

INSERT into Following (userId, totalFollowing, dateStartFollowing)
VALUES
(1, 1, '2020-01-01'),
(2, 2, '2020-01-01'),
(3, 0, '2020-01-01');

INSERT into Followers (userId, totalFollower, dateFollowed)
VALUES
(1, 1, '2020-01-01'),
(2, 1, '2020-01-02'),
(3, 1, '2020-01-01');

INSERT into ProfileEngagements(totalPost, totalProfileLikes, totalView,
totalComment)
VALUES
(10, 100, 1000, 5),
(1, 10, 100, 1),
(0, 0, 0, 0);

INSERT into PostEngagements(totalPostLike, totalPostComment, totalPostView,
datePosted)
VALUES
(5, 1, 10, '2020-01-02'),
(10, 0, 100, '2020-01-02'),
(25, 10, 20, '2020-01-02');
```

Schema


Followers
followerId INT(11)
userId INT(11)
dateFollowerLost DATE
isGainedFollower TINYINT(...
totalFollower INT(11)
dateFollowed DATE
Indexes
ProfileEngageme...
profileEngagementID INT(11)
totalPost INT(11)
totalProfileLikes INT(11)
totalView INT(11)
totalComment INT(11)
Indexes
UserFollow...
userFollowerId INT(1...
followerId INT(11)
Users_userId INT(11)
Indexes
UserFollowing
userFollowingId INT(1...
userId INT(11)
followingId INT(11)
Indexes
Users
userId INT(11)
userName VARCHAR(50)
dateCreated DATE
email VARCHAR(75)
ProfileEngagements_profileEngagementID INT(1...
Indexes
Following
followingId INT(11)
userId INT(11)
totalFollowing INT(11)
dateStartFollowing DA...
Indexes
PostEngage...
postId INT(11)
totalPostLike INT(11)
totalPostComment IN...
totalPostView INT(11)
datePosted DATE
userId INT(11)
Indexes
UserPosts
userPostId INT
userId INT(11)
postId INT(11)
Users_userId INT(11)
PostEngagements_postId INT(1...
Indexes