

MPCS 53001: Assignment #2

October 10, 2017

Julian McClellan

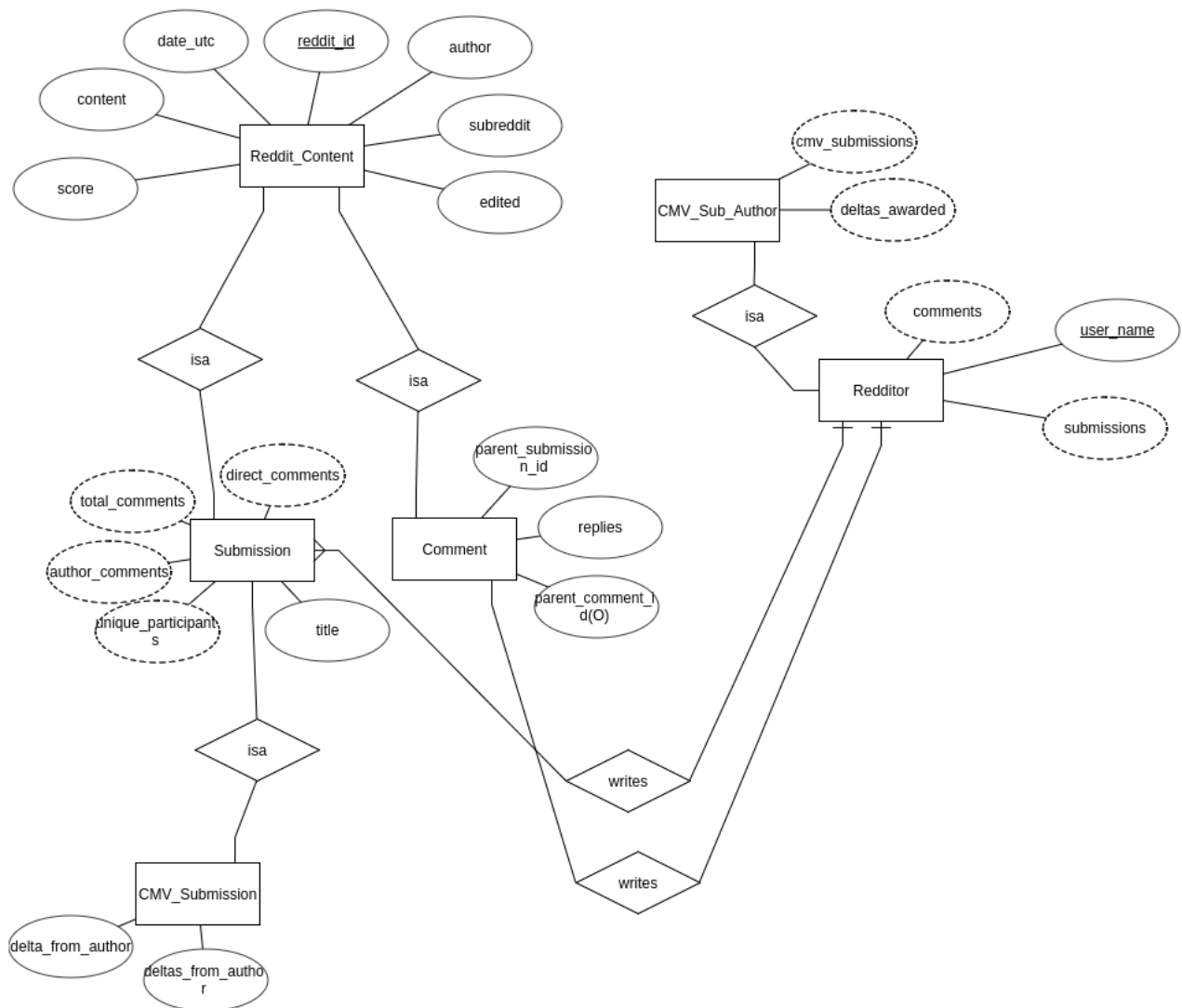
jmcclellan - 427478

1 Describe the app

The app I want to make would essentially allow users to delve into the subreddit `r/changemyview` and the Redditors who post their views there. Users would be able to explore the content of `r/changemyview` submissions, and be able to figure out whether or not the author of that submission ended up changing their view. Since I am parsing the comments of the submission to see if any of them changed the mind of the author, I would also like to allow users to access the comments of the submissions as well, either by some keyword search mechanism, sorting by score, or some other method. The exact mechanism of comment display probably depends on my capabilities with PHP and time. Baseline stats such as the total number of comments on a submission, the direct replies to the submission, total comments, and unique participants, are also stats I want to make available to users.

I would also allow users of the app to access the previous comment and submission history of `r/changemyview` authors. Users who are curious about the Reddit presence of an author can explore the author's past Reddit participation in depth, and should hopefully be able to search by date, keywords, different subreddits etc. I would also like to present some basic visualizations if possible to give users a quick-slice view of the author's overall activity level, and the author's prior deltas awarded (views changed). While *all* of a Redditor's previous post history is accessible, only ~ 1000 of their past comments are available, but I will still gather these to further flesh out their past Reddit participation.

2 Entity-relationship diagram



3 Transform the E/R diagram into a relational schema

3.1 Produce a set of relations for your database design

Submission(reddit_id, title, author, subreddit, date_utc, edited, score, content, total_comments, direct_comments, author_comments, num_unique_participants)

Comment(reddit_id, author, subreddit, date_utc, edited, score, content, num_replies, parent_submission_id, parent_comment_id(O))

CMV_Submission(reddit_id, title, author, subreddit, date_utc, edited, score, content, total_comments, direct_comments, author_comments, num_unique_participants, delta_from_author, num_deltas_from_author)

Redditor(user_name, num_comments, num_submissions)

For Redditors who are not CMV Submission authors these numbers will not be accurate, as they do not include submissions or comments outside r/changemyview

CMV_Sub_Author(user_name, num_comments, num_submissions, num_cmv_submissions, num_deltas_awarded)

3.2 Specify a set of nontrivial functional dependencies

The dependencies below here are rather trivial, as all the relations have a primary key that Reddit ascribes. (The Reddit API is rather developed, so this makes sense).

3.2.1 Submission

reddit_id → title, author, subreddit, date_utc, edited, score, content, total_comments, direct_comments, author_comments, num_unique_participants

3.2.2 Comment

reddit_id → author, subreddit, date_utc, edited, score, content, num_replies, parent_submission_id, parent_comment_id(O)

3.2.3 CMV_Submission

reddit_id → title, author, subreddit, date_utc, edited, score, content, total_comments, direct_comments, author_comments, num_unique_participants, delta_from_author, num_deltas_from_author

3.2.4 Redditor

user_name → num_comments, num_submissions

3.2.5 CMV_Sub_Author

user_name → num_comments, num_submissions, num_cmv_submissions, num_deltas_awarded