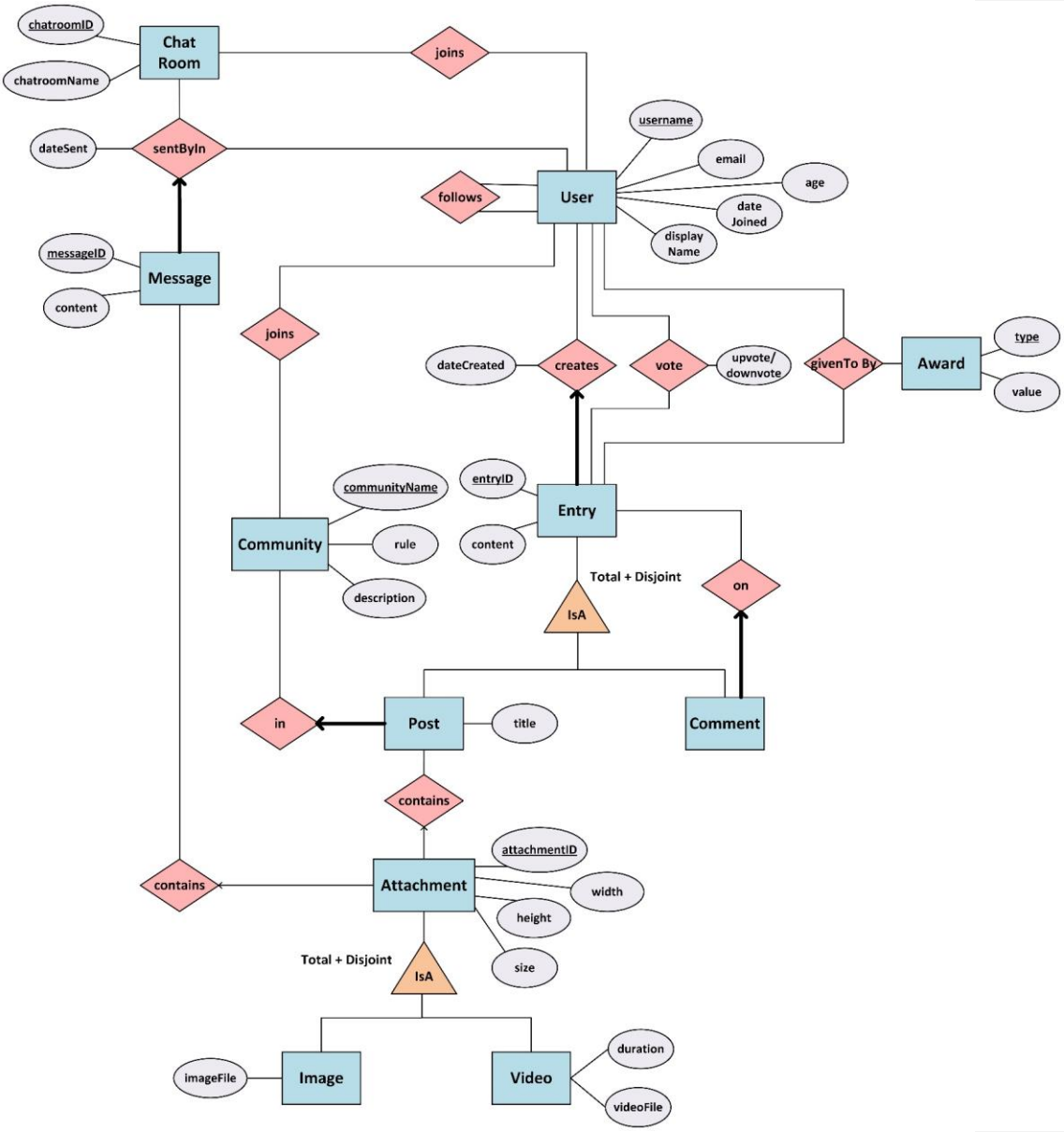
chatroomID
Chat Room
chatroomName
joins
username
email
age
date Joined
display Name
dateSent
sentByIn
follows
User
messageID
Message
content
joins
dateCreated
creates
vote
upvote/ downvote
givenTo By
Award
type
value
communityName
entryID
Entry
Community
rule
content
description
Total + Disjoint
IsA
on
in
Post
title
Comment
contains
attachmentID
contains
Attachment
width
height
size
Total + Disjoint
IsA
imageFile
Image
Video
duration
videoFile

Changes:

- Added attribute age to User that indicates the age of the account (not user), just like how Reddit has cake day (dateJoined) and Reddit age. This change was made so that dateJoined->age and we could normalize User. Previously, we had ImageContainedBy and VideoContainedBy normalized, but there are some concerns regarding it (https://piazza.com/class/m01ini8hsm72gv/post/569)
- Removed the total participation constraint on ChatRoom to joins, Award to givenToBy, and Community to joins because we can't use assertion on Oracle. These changes are for simplification, and our diagram still make sense (in some cases, more sense): we still keep the data on a chatroom even if no user are in it, an Award entitiy should still exist even if no user has given/received it (we still keep the data on the type and value associated with it), an empty community can exist and we should retain its data (a user can join that community anytime).

UPDATED SCHEMA

- User (username: varchar(20), email: varchar(20), dateJoined: date, displayName: varchar(20), age: integer)
  Primary Key: username
  Candidate Key: email (one email can only be associated with one account)
  Unique: email
  Not NULL: dateJoined, email

- Community (communityName: varchar(20), rule: text, description: text)
  Primary Key: communityName

- EntryCreatedBy (entryID: integer, dateCreated: date, content:text, username: varchar(20))
  Primary Key: entryID
  Foreign Key: username
  Not Null: username
  - PostIn (**entryID**: integer, title:text, communityName: varchar(20))
    Primary Key: entryID
    Foreign Key : entryID**,** communityName
    Not NULL : title, communityName
    Note: Uses method 2 of ISA and also combines in relationships of Post
  - CommentOn (entryID: integer, onEntryID: integer)
    Primary Key : entryID
    Foreign Key: entryID, onEntryID
    Not NULL: onEntryID
    Note : This table uses method 2 of ISA and combines the Comment and the on relationship

- ImageContainedBy (attachmentID: integer, imageFile: mediumblob, width: integer, height: integer, size: integer, entryID: integer, messageID: integer)

  Primary Key: attachmentID
  Foreign key: entryID, messageID
  Not NULL: imageFile

  Note : Uses method 3 of ISA,This table combines the attachment and image entities, as well as the contains relationships. An attachment must be contained in one of either post or message (see note).

Commented [JP1]: I made 1 schema per entity/relationship but we could maybeee merge some though I don't understand how. There are prob more constraints we need to add too (unique, not null, foreign key) but idk I could only figure out some

Commented [JP2R1]: And for schemas involving relationships with total participation, idk how we're supposed to show that constraint but I made it according to this post https://piazza.com/class/m01ini8hsm72gv/post/187

- VideoContainedBy (attachmentID: integer, videoFile: largeblob, duration: time, width: integer, height: integer, size: integer, entryID: integer, messageID: integer)

  Primary Key: attachmentID
  Foreign key: entryID, messageID
  Not NULL: videoFile

  Note : This table combines the Attachment and Video entities, as well as the contains relationships (see ImageContainedBy for similar setup)

- MessageSentByIn (messageID: integer, dateSent: date, content: text, username: varchar(20), chatroomID: integer)
  Primary Key: messageID
  Foreign key : chatroomID, username
  Not NULL: dateSent, content, username, chatroomID

  Note : This table combines the Message entity with sentByIn relationship.

- ChatRoom(chatroomID: integer, name: varchar[20])
  Primary Key : chatroomID

- JoinsChatroom (chatroomID: integer, username: varchar(20))
  Primary Key: chatroomID, username
  Foreign Key: chatroomID, username

- Award (awardType: varchar(10), value: integer)
  Primary Key: awardType
  Not NULL: value

- GivenToBy (awardType: integer, username: varchar(20), entryID: integer)
  Primary Key: awardType, entryID, username
  Foreign key: awardType, entryID, username
  Not NULL: awardType

- Follows (followingUsername: varchar[20], followedUsername: char[20])
  Primary Key: followingUsername, followedUsername
  Foreign Key: followingUsername, followedUsername

- Vote (username: varchar[20], entryID: integer, upvoteOrDownvote: boolean)
  Primary Key: username, entryID
  Foreign Key: username, entryID

Commented [HP3]: Ternary relationship combining In and Sends, has user and chatroom as foreign keys and participation and many to one constraint

Commented [HP4]: Combine Message and send relationship, user and in are added as foreign keys and both are not null.

Commented [JP5R4]: Changed in to chatroomID and user to username. Made similar changes on other schemas as well

Commented [HP6]: awardType should be integer

Commented [JP7R6]: I was thinking it wold be like “rose”, “star”, etc idk

Commented [JP8R6]: So is award type an int or varchar

Commented [HP9]: IMPORTANT CHANGE: This is a ternary relationship replacing gives and to. I’ve also tweaked the ER Diagram (see givenToBy). Note that total participation by award is not expressed (requires assertion statements according to slides). We haven’t learnt that yet so worry about it later

- JoinsCommunity (username: varchar[20], communityName: varchar[20])
  Primary Key: username, communityName
  Foreign Key : username, communityName

FUNTIONAL DEPENDENCIES

givenToBy, joinsChatroom, joinsCommunity and follows have all attributes as their keys, so FDs are trivial.

From User:

- username → email, dateJoined, displayName, age
- email → username, dateJoined, displayName, age
- dateJoined → age

From Community:

- communityName → rule, description

From EntryCreatedByIn:

- entryID → dateCreated, content, username,

From PostIn

- entryID → title, communityName

From CommentOn

- entryID → onEntryID

From MessageSentByIn:

- messageID → dateSent, content, username, chatroomID

From Chatroom:

- ChatroomID → name

From Award:

- awardType → value

From Vote:

- username, entryID→ upvoteordownvote

From ImageContainedBy:

- attachmentID → imageFile, width, height, size, entryID, messageID

From VideoContainedBy:

- attachmentID → videoFile, width, height, size,duration, entryID, messageID

Commented [JP10]: How about content?

Commented [HP11R10]: Already in its superclass

Commented [JP12R10]: Nvm im dumb

Commented [JP13]: And here?

Commented [HP14R13]: Super class -_-

Commented [JP15]: Added this (?)

Commented [HP16R15]: Idk FDs should kinda be between attributes in one table, So I'd say this is wrong

Commented [JP17R15]: Wdym image has attachmentID and imageFile

Commented [JP18R15]: The one that is wrong if you said so is imageFile->resolution, size bcs res and size is not on image

NORMALIZATION (to 3NF)

User (username, email, dateJoined, displayName, age)

- username → email, dateJoined, displayName, age
- email → username, dateJoined, displayName, age
- dateJoined → age

Closures:

- username += email, dateJoined, displayName, age
- email += username, dateJoined, displayName, age
- dateJoined += age

username is key

Normalize to 3NF by Loseless Join Method:

dateJoined → age is not in 3NF because dateJoined is not a superkey and age is not part of superkey.

Minimal cover:

- username → email
- ~~username → dateJoined~~
- ~~username → displayName~~
- ~~username → age~~
- email → username
- email → dateJoined
- email → displayName
- ~~email → age~~
- dateJoined → age

dateJoined → age violates 3NF because dateJoined is not a superkey and age is not part of superkey.

Decomper User on dateJoined → age:

- User1(<u>dateJoined</u>, age)
- User2 (<u>username</u>, email, dateJoined, displayName)

Result:

- User1(<u>dateJoined</u>, age)
- User2 (<u>username</u>, email, dateJoined, displayName)

*Primary keys are <u>underlined</u>. Foreign keys are **bolded**. No other candidate key except of primary keys.*

The rest of our relations are already in BCNF/3NF.