**University of British Columbia, Vancouver**
Department of Computer Science

# CPSC 304 Project Cover Page

Milestone #: 2

Date: October 15th, 2024

Group Number: 25

| Name | Student Number | CS Alias (Userid) | Preferred E-mail Address |
|---|---|---|---|
| Jeff Kim | 70668132 | m3v1i | Jeffkim7@hotmail.com |
| Jessica Patricia | 81731218 | l7j4y | jessicapatricia012@gmail.com |
| Hansel Poe | 82673492 | l7z7n | hpoe01@student.ubc.ca |

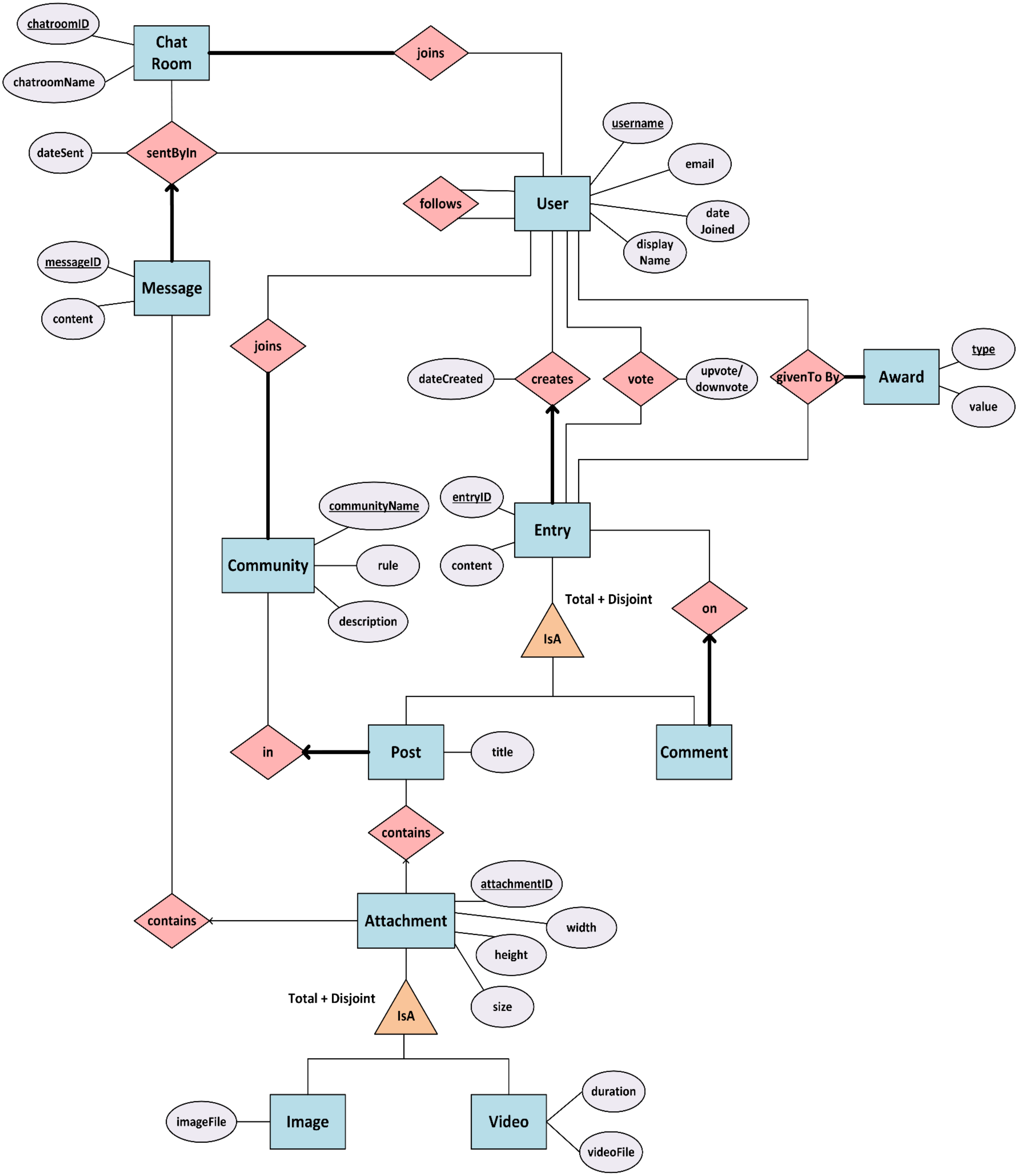
By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

## SUMMARY

We are making a database for a social media platform inspired by the popular application Reddit with similar features, but with our own tweaks. Our domain will model basic social networking concepts, such as users, communities, posts, comments, messages, attachments, chat rooms, and awards. The user interactions are modeled in the relationships between entities where users can join communities, follow other users, make posts and comments with attachments, give awards to and upvote/downvote posts and comments, and enter chat to message other users.

## ER DIAGRAM


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

NOTE:

- Both IsA have a total + disjoint constraint
- vote has a boolean attribute of upvote/downvote. 1 if a user upvotes and 0 if a user downvotes
- Extra constraint: attachment has to belong to one of either a post or a message

Changes:

- Merged the previously two on relationships involving Comment into one to reduce redundancy since both Comment and Post are subclasses of Entry. We now have “Comment on Entry” instead of “Comment on Post” and “Comment on Comment”.
- Deleted the sub-entity URL because it is a text and can be considered as a part of the content of Message or Entry.
- Deleted belongs (from chatroom to community) to simplify the ER diagram. Initially inspired by Reddit’s Channel, we decided to add this relationship. Then we noticed this relationship was not as significant and none of us want this feature.
- Added attribute Content of type text on Entry and Message, and attribute Image File and Video File of type BLOB on Image and Video respectively. They are necessary to store the content/file.
- Moved the attributes Resolution and Size from the sub-entities Image and Video to their parent entity attachment, now that we do not have URL. Then replace Resolution to Width and Height of type integers
- Added a ternary relationship sentByIn involving Message, Chatroom, and User to replace the binary relatonships sends and in. This reduces redundancy of tables and makes more sense.
- Added a ternary relationship givenToBy involving Award, User, and Entry to replace the binary relationships gives and to. Same reason as above.
- Changed attribute names like ID or name to include the entity name for clarity (e.g. entryID, communityName)
- Deleted dateAdded attribute of Attachment because it’s irrelevant, we only care about the date of Post or Message
- Moved the date/timeWritten attribute of Message and Entry to the relationships sentByIn and Create respectively, and renamed them to dateSent and datecreated. This implementation is more intuitive.

**SCHEMA**

- User (username: varchar(20), email: varchar(20), dateJoined: date, displayName: varchar(20))
  **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).

- 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  
  **Note :** The participation constraint on chat room must be expressed using assertions

- 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

  **Note:** The participation constraint of Award must be expressed using assertions

- 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

- 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
- email → username, dateJoined, displayName

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:

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

From VideoContainedBy:

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

**NORMALIZATION (to 3NF)**

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

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

Closures:

- attachmentID += imageFile, width, height, size, entryID, messageID
- imageFile += width, height, size

attachmentID is key

Normalize to 3NF by Loseless Join Method:

imageFile → width, height, size is not in 3NF because imageFile is not a superkey and width, height, size are not part of superkey.

Minimal cover:

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

imageFile → width, imageFile → height and imageFile → size violate 3NF because imageFile is not a superkey, and width, height, size are not a part of superkey.

Decompose ImageContainedBy on imageFile → width:

- ImageContainedBy1(imageFile, width)
- ImageContainedBy2(attachmentID, imageFile, height, size, entryID, messageID)

Decompose ImageContainedBy2 on imageFile-> height (violates BCNF):

- ImageContainedBy3(imageFile, height)
- ImageContainedBy4(attachmentID, imageFile, size, entryID, messageID)

Decompose ImageContainedBy4 on imageFile-> size (violates BCNF):

- ImageContainedBy5(imageFile, size)
- ImageContainedBy6(attachmentID, imageFile, entryID, messageID)

Result:

- ImageContainedBy1(imageFile, width)
- ImageContainedBy3(imageFile, height)
- ImageContainedBy5(imageFile, size)
- ImageContainedBy6(attachmentID, imageFile**, entryID, messageID**)

*Primary keys are underlined. Foreign keys are **bolded**. No other candidate key except of primary keys.*

VideoContainedBy (attachmentID, imageFile,width, height, size, duration, entryID, messageID):

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

Closures:

- attachmentID += videoFile, duration, width, height, size, entryID, messageID
- videoFile += width, height, size, duration

attachmentID is key

Normalize to 3NF by Loseless Join Method:

videoFile → width, height, size, duration is not in 3NF because videoFile is not a superkey and width, height, size, duration are not part of superkey.

Minimal cover:

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

videoFile → width, videoFile → height, videoFile → size, and videoFile → duration violate 3NF because videoFile is not a superkey, and width, height, size, duration are not a part of superkey.

Decompose videoContainedBy on videoFile → width:

- VideoContainedBy1(<u>videoFile</u>, width)
- VideoContainedBy2(<u>attachmentID</u>, videoFile, height, size, duration, entryID, messageID)

Decompose VideoContainedBy2 on videoFile-> height (violates BCNF):

- VideoContainedBy3(<u>videoFile</u>, height)
- VideoContainedBy4(<u>attachmentID</u>, videoFile, size, duration, entryID, messageID)

Decompose VideoContainedBy4 on videoFile → size:

- VideoContainedBy5 (<u>videoFile</u>, size)
- VideoContainedBy6 (<u>attachmentID</u>, videoFile, duration, entryID, messageID)

Decompose VideoContainedBy6 on videoFile → duration:

- VideoContainedBy7 (<u>videoFile</u>, duration)
- VideoContainedBy8 (<u>attachmentID</u>, videoFile, entryID, messageID)

Result:

- VideoContainedBy1 (<u>videoFile</u>, width)
- VideoContainedBy3 (<u>videoFile</u>, height)
- VideoContainedBy5 (<u>videoFile</u>, size)
- VideoContainedBy7 (<u>videoFile</u>, duration)
- VideoContainedBy8 (<u>attachmentID</u>, videoFile**, entryID, messageID**)

*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.

**SQL DDL STATEMENTS**

```
CREATE TABLE User (
        username          VARCHAR(20)     PRIMARY KEY,
        email             VARCHAR(20)     NOT NULL,
        dateJoined        DATE            NOT NULL,
        displayName       VARCHAR(20),
        UNIQUE (email)
);
CREATE TABLE Community (
        communityName     VARCHAR(20)     PRIMARY KEY,
        rule              TEXT,
        description       TEXT
);
CREATE TABLE EntryCreatedBy (
        entryID           INTEGER         PRIMARY KEY,
        dateCreated       DATE,
        content           TEXT,
        username          VARCHAR(20)
);
CREATE TABLE PostIn (
        entryID           INTEGER         PRIMARY KEY,
        title             TEXT            NOT NULL,
        communityName     VARCHAR(20)     NOT NULL,
        FOREIGN KEY (communityName) REFERENCES EntryCreatedByIn,
        FOREIGN KEY (communityName) REFERENCES Community,
);
CREATE TABLE CommentOn (
        entryID           INETEGER        PRIMARY KEY,
        onEntryID         INTEGER         NOT NULL,
        FOREIGN KEY (onEntryID) REFERENCES EntryCreatedByIn,
        FOREIGN KEY (entryID) REFERENCES EntryCreatedByIn
);
```

```
CREATE TABLE MessageSentByIn (
        messageID           INETEGER          PRIMARY KEY,
        dateSent            DATE              NOT NULL,
        content             TEXT              NOT NULL,
        username            VARCHAR(20)       NOT NULL,
        chatroomID          INETEGER          NOT NULL,
        FOREIGN KEY (chatroomID) REFERENCES chatRoom,
        FOREIGN KEY (username) REFERENCES User
);
CREATE TABLE ChatRoom (
        chatroomID          INTEGER           PRIMARY KEY,
        name                VARCHAR(20)
);
CREATE TABLE JoinsChatRoom (
        chatroomID          INTEGER,
        username            VARCHAR(20),
        PRIMARY KEY (chatroomID, username),
        FOREIGN KEY (chatroomID) REFERENCES chatRoom,
        FOREIGN KEY (username) REFERENCES User,
);
CREATE TABLE Award (
        awardType           VARCHAR(10)        PRIMARY KEY,
        value               INTEGER           NOT NULL
);
CREATE TABLE GivenToBy (
        awardType           INETEGER          NOT NULL,
        username            VARCHAR(20),
        entryID             INETEGER,
        PRIMARY KEY (awardType, username, entryID),
        FOREIGN KEY (awardType) REFERENCES award,
        FOREIGN KEY (entryID) REFERENCES EntryCreatedBy,
        FOREIGN KEY (username) REFERENCES User,
);
```

```
CREATE TABLE Follows(
        followingUsername  VARCHAR(20),
        followedUsername   VARCHAR(20)
        PRIMARY KEY(followingUsername, followedUsername),
        FOREIGN KEY (followingUsername) REFERENCES User,
        FOREIGN KEY (followedUsername) REFERENCES User,
);

CREATE TABLE Vote (
        username           VARCHAR(20),
        entryID            INTEGER,
        upvoteOrDownvote   BOOLEAN,
        PRIMARY KEY(username, entryID),
        FOREIGN KEY (entryID) REFERENCES EntryCreatedBy,
        FOREIGN KEY (username) REFERENCES User,
);

CREATE TABLE JoinsCommunity(
        username           VARCHAR(20),
        communityName      VARCHAR(20),
        PRIMARY KEY (username, communityName),
        FOREIGN KEY (communityName) REFERENCES Community,
        FOREIGN KEY (username) REFERENCES User,
);
CREATE TABLE ImageContainedBy1(
        imageFile          MEDIUMBLOB    PRIMARY KEY,
        width              INTEGER
);
CREATE TABLE ImageContainedBy3(
        imageFile          MEDIUMBLOB    PRIMARY KEY,
        height             INTEGER
);
CREATE TABLE ImageContainedBy5(
        imageFile          MEDIUMBLOB    PRIMARY KEY,
        size               INTEGER
);
```

```
CREATE TABLE ImageContainedBy6(
        AttachmentID        INTEGER         PRIMARY KEY,
        imageFile           MEDIUMBLOB      NOT NULL,
        entryID             INTEGER,
        messageID           INETEGER,
        FOREIGN KEY (entryID) REFERENCES PostIn,
        FOREIGN KEY (messageID) REFERENCED MessageSentByIn
);
CREATE TABLE VideoContainedBy1 (
        videoFile           LARGEBLOB       PRIMARY KEY,
        width               INTEGER
);
CREATE TABLE VideoContainedBy3 (
        videoFile           LARGEBLOB       PRIMARY KEY,
        height              INTEGER
);
CREATE TABLE VideoContainedBy5 (
        videoFile           LARGEBLOB       PRIMARY KEY,
        size                INTEGER
);
CREATE TABLE VideoContainedBy7 (
        videoFile           LARGEBLOB       PRIMARY KEY,
        duration            TIME
);
CREATE TABLE VideoContainedBy8(
        attachmentID        INTEGER         PRIMARY KEY,
        videoFile           LARGEBLOB       NOT NULL,
        entryID             INTEGER,
        messageID           INETEGER,
        FOREIGN KEY (entryID) REFERENCES PostIn,
        FOREIGN KEY (messageID) REFERENCED MessageSentByIn
);
```

## INSERT STATEMENTS

```
INSERT INTO User VALUES
        (‘hpoe01’, ‘hpoe@outlook.com’, ‘2024-10-16’, ‘hansel’);
        (‘jpat02’, ‘jpat@outlook.com’, ‘2024-10-15’, ‘jessica’);
        (‘jkim03’, ‘jkim@gmail.com’, ‘2024-10-14’, ’jeff’ );
        (‘skat04’, ‘kat@gmail.com’, ‘2024-10-13’, ‘kat’);
        (‘ljon05’, ‘jon@gmail.com’, ‘2024-10-12’, ‘jon’);

INSERT INTO Community VALUES:
        (‘BuildAPC’, ‘1. No harrasment\n 2. No Politics…’, ‘Welcome to the largest sub on anything
        PC related’),
        (‘Math’, ‘1. No discussing other subjects\n 2. No Physics discussion...’ , ‘Discuss anything
        Math here!’),
        (‘CSCareers’, ‘1. No Grifting\n 2. No Politics…’, ‘Ask industry Professionals about CS
        careers’),
        (‘AMA’, ‘1. No boring questions\m 2. Be original…’, ‘Place for interesting people to answer
        questions’),
        (‘Gaming’, ‘1. Gaming topics only\n 2. No Politics…’, ‘Discuss the latest gaming news
        here!’);

INSERT INTO EntryCreatedBy VALUES
        (1, ‘2024-10-15’, [content], ‘hpoe01’),
        (2, ‘2024-10-14’, [content], ‘hpoe01’),
        (3, ‘2024-10-13’ [content], ‘jpat02’),
        (4, ‘2024-10-12’, [content], ‘skat04’),
        (5, ‘2024-10-11’ [content],’ljon05’),
        (6, ‘2024-10-15’, [content], ‘hpoe01’),
        (7, ‘2024-10-14’, [content], ‘hpoe01’),
        (8, ‘2024-10-13’ [content], ‘jpat02’),
        (9, ‘2024-10-12’, [content], ‘skat04’),
        (10, ‘2024-10-11’, [content],’ljon05’);

INSERT INTO PostIn VALUES
        (1,’New Fallout Game’, ‘ gaming’),
        (2,‘Stuck in this level, pls help’, ‘gaming’),
        (3,‘How to get a job?’,  ‘CSCareers’),
        (4,‘Stuck in this level, pls help’, ‘gaming’),
        (5,‘CPU Overheating?’,  ‘BuildAPC’);
```

```
INSERT INTO CommentOn VALUES
        (6, 1),
        (7, 2),
        (8, 3),
        (9, 4),
        (10, 5);
INSERT INTO ImageContainedBy1 (imageFile, width) VALUES
        ([blob of cat image 1], 640),
        ([blob of cat image 2], 539)
        ([blob of dog image 1], 710),
        ([blob of dog image 2], 600),
        ([blob of snake image], 1200);

INSERT INTO ImageContainedBy3 (imageFile, height) VALUES
        ([blob of cat image 1], 426),
        ([blob of cat image 2], 360),
        ([blob of dog image 1], 340),
        ([blob of dog image 2], 400),
        ([blob of snake image], 800);

INSERT INTO ImageContainedBy5 (imageFile, size) VALUES
        ([blob of cat image 1, 2097152]),
        ([blob of cat image 2, 2813651]),
        ([blob of dog image 1, 2907322]),
        ([blob of dog image 2, 3362163]),
        ([blob of snake image, 1342344]);

INSERT INTO ImageContainedBy6 (attachmentID, imageFile, entryID, messageID) VALUES
        (1, [blob of cat image 1],  1, NULL),
        (2, [blob of cat image 2], 2, NULL),
        (3, [blob of dog image 1], NULL, 3),
        (4, [blob of dog image 2], NULL, 4),
        (5, [blob of snake image ], 5, NULL);

INSERT INTO VideoContainedBy1 VALUES
        ([blob of video 1], 1280),
        ([blob of video 2], 1280),
        ([blob of video 3], 1920),
        ([blob of video 4], 1920),
        ([blob of video 5], 1280);

INSERT INTO VideoContainedBy3 VALUES
        ([blob of video 1], 720),
        ([blob of video 2], 720),
        ([blob of video 3], 1080),
        ([blob of video 4], 1080),
        ([blob of video 5], 720);

INSERT INTO VideoContainedBy5 VALUES
        ([blob of video 1], 431231231),
        ([blob of video 2], 213312321424),
        ([blob of video 3], 453255446),
        ([blob of video 4], 534563536),
        ([blob of video 5], 43524542352);

INSERT INTO VideoContainedBy7 VALUES (HH:mm:ss)
        ([blob of video 1], ‘00: 04:15’),
        ([blob of video 2], ‘00:00:34’),
        ([blob of video 3],  ’00:01:21’),
        ([blob of video 4], ’00:00:40’),
        ([blob of video 5], ’00:04:14’);

INSERT INTO VideoContainedBy8 VALUES
        (6, [blob of video 1], 1, NULL),
        (7, [blob of video 2], 2, NULL),
        (8, [blob of video 3], NULL, 3),
        (9, [blob of video 4], NULL,4),
        (10, [blob of video 5], 5,NULL);


INSERT INTO ChatRoom VALUES
        (1, ‘Group 1’),
        (2, ‘Group 2’),
        (3, ‘Group 3’),
        (4, ‘Group 4’),
        (5, ‘Group 5’);


INSERT INTO JoinsChatRoom VALUES
        (1,’hpoe01’),
        (1, ‘jpat02’),
        (2, ‘hpoe01’),
        (4, ‘skat04’),
        (5, ‘skat04’);

INSERT INTO MessageSentByIn VALUES
        (1, ‘2024-10-15’, [content], ‘hpoe01’, 1),
        (2, ‘2024-10-14’, [content], ‘hpoe01’, 1),
        (3, ‘2024-10-13’, [content], ‘jpat02’, 1),
        (4, ‘2024-10-12’, [content], ‘skat04’, 4),
        (5, ‘2024-10-11’, [content], ‘skat04’, 5);

INSERT INTO Award VALUES
        (‘Gold’, 100),
        (‘Silver, 75),
        (‘Bronze’, 50),
        (‘Balloon’, 25),
        (‘Rose’, 10);

INSERT INTO GivenToBy VALUES
        (‘Gold’,  ‘hpoe01’, 1),
        (‘Gold’,  ‘hpoe01’, 1),
        (‘Balloon’,  ‘jpat02’, 2),
        (‘Rose’,  ‘jpat02’, 2),
        (‘Rose’,  ‘skaat04’, 7);
INSERT INTO Follows VALUES
        (‘hpoe01’, ‘jpat02’),
        (‘jpat02’, ‘hpoe01’),
        (‘skat04’, ‘hpoe01’),
        (‘hpoe01’, ‘skat04’),
        (‘skat04’, ‘ljon05’);

INSERT INTO Vote VALUES
        (‘jpat02’, 1, 1),
        (‘jpat02’, 2, 1),
        (‘hpoe01’, 3, 0),
        (‘hpoe01’, 4, 0),
        (‘skat04’, 5, 1);
```

```
INSERT INTO JoinsCommunity VALUES
        (‘hpoe01’, ‘Gaming’),
        (‘hpoe01’, ‘BuildAPC’),
        (‘jpat02’, ‘CSCareers’),
        (‘skat04’, ‘Gaming‘),
        (‘ljon05’, ‘BuildAPC’);
```