**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.

**A diagram of different colored squares

Description automatically generated with medium confidenceER DIAGRAM**

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 boolean attribute containedType to Attachment to indicate whether it is attached to Post or Message. 1 for Post and 0 for Message.
* 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

* PostInCreatedByIn(entryID: integer,dateCreated: date,content: text,username: varchar[20], title: text, communityName: varchar(20)) (

**Primary Key:** entryID

**Foreign key:** communityName, username

**Not NULL:** dateCreated, content, username,title, communityName

**Note:** This table combines the Post and Entry entity, and creates and in relationships. We use method 3 of ISA schema implementation

* CommentOnBy (entryID: integer, dateCreated: date, content: text, username: varchar[20], onEntryID: integer)

**Primary Key:** entryID

**Foreign key:** onEntryID, username

**Not NULL:** dateCreated, content,username, onEntryID

**Note** : This table combines the Comment and Entry entities, and the on and creates relationships. We use method 3 of ISA schema implementation

* ImageContainedBy (attachmentID: integer, imageFile: mediumblob, width: integer, height: integer, size: integer, containedID:integer , containedType: boolean)

**Primary Key:** attachmentID

**Foreign key:** containedID

**Not NULL:** imageFile, containedID, containedType

**Note** : This table combines the attachement and image entities, as well as the contains relationships. An attachment must be contained in one of either post or message (see note). We use containedType to express this 1 for Post and 0 for Message. We do this because messageID and entryID might be the same (in which case we need to use containedType to distinguish them).

* VideoContainedBy (attachmentID: integer, videoFile: largeblob, duration: time, width: integer, height: integer, size: integer, containedID: integer, containedType: boolean)

**Primary Key:** attachmentID

**Foreign key:** containedID

**Not NULL:** videoFile, contained

**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

* ChatroomJoinedBy (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: integer, value: integer)

**Primary Key:** awardType

* 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

* joins (username: varchar[20], communityName: varchar[20])

**Primary Key:** username, communityName

**Foreign Key :** username, communityName

**FUNTIONAL DEPENDENCIES**

givenToBy, chatroomJoinedBy, joins 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 PostInCreatedByIn

* entryID → title, communityName, dateCreated, content, username

From CommentOnBy

* entryID → onEntryID, dateCreated, content, username

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, containedID, containedType

From VideoContainedBy:

* videoFile → width, height, size, duration
* attachmentID → videoFile, width, height, size,duration, containedID, containedType

**NORMALIZATION (to 3NF)**

ImageContainedBy (attachmentID, imageFile, width, height, size, containedType, containedID):

* attachmentID → imageFile, width, height, size, containedType, containedID
* imageFile → width, height, size

Closures:

* attachmentID += imageFile, width, height, size, containedType, containedID
* 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 → containedType
* attachmentID → containedID
* 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:

* R1(imageFile, width)
* R2(attachmentID, imageFile, height, size, containedType, containedID)

Decompose R2 on imageFile-> height (violates BCNF):

* R3(imageFile, height)
* R4(attachmentID, imageFile, size, containedType,containedID)

Decompose R4 on imageFile-> size (violates BCNF):

* R5(imageFile, size)
* R6(attachmentID, imageFile, containedType,containedID)

Result:

* R1(imageFile, width)
* R3(imageFile, height)
* R5(imageFile, size)
* R6(attachmentID, imageFile, containedType,containedID)

*Primary keys are underlined. No other candidate key except of primary keys, and no foreign keys.*

VideoContainedBy (attachmentID, imageFile,width, height, size, duration, containedType, containedID):

* attachmentID → videoFile, duration, width, height, size, containedType, containedID
* videoFile → width, height, size, duration

Closures:

* attachmentID += videoFile, duration, width, height, size, containedType, containedID
* 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 → containedType
* attachmentID → containedID
* 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:

* S1(videoFile, width)
* S2(attachmentID, videoFile, height, size, duration, containedType, containedID)

Decompose S2 on videoFile-> height (violates BCNF):

* S3(videoFile, height)
* S4(attachmentID, videoFile, size, duration, containedType,containedID)

Decompose S4 on videoFile → size:

* S5(videoFile, size)
* S6(attachmentID, videoFile, duration, containedType, containedID)

Decompose S6 on videoFile → duration:

* S7(videoFile, duration)
* S8(attachmentID, videoFile, containedType, containedID)

Result:

* S1(videoFile, width)
* S3(videoFile, height)
* S5(videoFile, size)
* S7(videoFile, duration)
* S8(attachmentID, videoFile, containedType, containedID)

*Primary keys are underlined. No other candidate key except of primary keys, and no foreign keys.*

**SQL DDL STATEMENTS**

ImageContainedBy:

* R1(imageFile, width)
* R3(imageFile, height)
* R5(imageFile, size)
* R6(attachmentID, imageFile, containedType,containedID)

CREATE TABLE R1(

imageFile MEDIUMBLOB PRIMARY KEY,

width INTEGER

)

CREATE TABLE R3(

imageFile MEDIUMBLOB PRIMARY KEY,

height INTEGER

)

CREATE TABLE R5(

imageFile MEDIUMBLOB PRIMARY KEY,

size INTEGER

)

CREATE TABLE R6(

AttachmentID INTEGER PRIMARY KEY,

imageFile MEDIUMBLOB,

containedID INTEGER FOREIGN KEY,

containedType BOOLEAN,

NOT NULL (imageFile, containedID, containedType)

)

videoContainedBy:

* S1(videoFile, width)
* S3(videoFile, height)
* S5(videoFile, size)
* S7(videoFile, duration)
* S8(attachmentID, videoFile, containedType, containedID)

CREATE TABLE S1(

videoFile LARGEBLOB PRIMARY KEY,

width INTEGER

)

CREATE TABLE S3(

videoFile LARGEBLOB PRIMARY KEY,

height INTEGER

)

CREATE TABLE S5(

videoFile LARGEBLOB PRIMARY KEY,

size INTEGER

)

CREATE TABLE S7(

videoFile LARGEBLOB PRIMARY KEY,

duration TIME

)

CREATE TABLE S8(

AttachmentID INTEGER PRIMARY KEY,

videoFile LARGEBLOB,

containedID INTEGER FOREIGN KEY,

containedType BOOLEAN,

NOT NULL (videoFile, containedID, containedType)

)

**INSERT STATEMENTS**

ImageContainedBy:

* R1(imageFile, width)
* R3(imageFile, height)
* R5(imageFile, size)
* R6(attachmentID, imageFile, containedType,containedID)

INSERT INTO R1 (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 R3 (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 R5 (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 R6 (attachmentID, imageFile, containedType, containedID) VALUES

(1, [blob of cat image 1], 1, 100)

(2, [blob of cat image 2], 1, 101),

(3, [blob of dog image 1], 0, 102),

(4, [blob of dog image 2], 0, 103),

(5, [blob of snake image ], 1, 104);

VideoContainedBy:

* S1(videoFile, width)
* S3(videoFile, height)
* S5(videoFile, size)
* S7(videoFile, duration)
* S8(attachmentID, videoFile, containedType, containedID)

INSERT INTO S1 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 S3 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 S5 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 S7 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 S8 VALUES

(6, [blob of video 1], 1, 105),

(7, [blob of video 2], 1, 106),

(8, [blob of video 3], 0, 107),

(9, [blob of video 4], 0, 108),

(10, [blob of video 5], 1, 109);

Images I am using:

Cat image 1 (640 x 426) :

<https://cdn.pixabay.com/photo/2024/02/28/07/42/european-shorthair-8601492_640.jpg>

cat image 2 (539 x 360):

<https://t3.ftcdn.net/jpg/02/36/99/22/360_F_236992283_sNOxCVQeFLd5pdqaKGh8DRGMZy7P4XKm.jpg>

Dog Image 1 (710 x 340) :

<https://www.nylabone.com/-/media/project/oneweb/nylabone/images/dog101/10-intelligent-dog-breeds/golden-retriever-tongue-out.jpg>

Dog image 2 (600 x 400):

<https://www.dogster.com/wp-content/uploads/2023/08/red-toy-poodle-in-the-park_Mykhaylo_Kozelko_Shutterstock-600x400.jpg>

Snake image (1200 x 800):

<https://www.worldanimalprotection.ca/cdn-cgi/image/width=1280,format=auto/siteassets/shutterstock_2311286833.jpg>

Videos I am using:

Video 1: <https://youtu.be/xv1TQvROLfk> (4:15, 720p)

Video 2: <https://youtu.be/R44L-EovL88> (0:34, 720p)

Video 3: <https://youtu.be/N6igQn_2foE> (1:21, 1080p)

Video 4: <https://youtu.be/B0ed2CMuycg> (0:40, 1080p)

Video 5: <https://www.youtube.com/watch?v=53NLbM9QdbM> (4:14,720p)