**Project Description**

**Changes**

* **Guest, Streamer and Moderator removed**
  + We combined all the user-related entities into one User entity since the entities above do not have any attributes. Therefore we reduced the number of redundant tables.
  + So all the relationship sets related to Guest, Streamer and Moderator now become binary relationship sets with User.
* **Password moved from RegisteredUser to User**
  + Every user should have a password.
* **numOfFollowers removed in RegisteredUser**
  + This can be calculated via the relationship Follows.
* **RegisteredUser merged into User**
  + There is no need to distinguish User and RegisteredUser.
* **ChatID added as a primary key of ChatMessage**
  + sendername and time are no longer the primary key of ChatMessage. Therefore deleting a user would not cause primary key of ChatMessage to be null.
  + sendername is removed since it is a foreign key from User.
* **numOfViewers removed in Channel**
  + This can be calculated via the relationship Watches.
* **Channel changed to a strong entity**
  + Since channel can be uniquely identified by channelID, there is no point to make it a weak entity.
* **C\_is\_of removed and Type removed**
  + Type is redundant. A Channel is either a GameChannel or a ShowChannel
* **Game renamed to GameChannel**
  + It is for better interpretation.
* **Show renamed to ShowChannel**
  + Same as above.
* **Channel\_Chat combined into Sends in ER diagram**
  + Therefore we have less relationships.
  + A ternary relationship makes more sense.
* **startTime and duration removed in Mutes**
  + The relationship set Mutes cannot have any attributes acting as its primary key.
  + We keep endTime and remove the other two for simplicity.
* **Mutes changed to an aggregation with GrantsPrivilege**
  + This makes more sense since a user cannot send chat message if he/she is muted.
* **Donates replaced by an entity Donation and a ternary relationship Makes**
  + Donates cannot have any attribute acting as its primary key.
  + Donation is added since we want to make a record for each donation.

**what the project accomplished**

* **users**:
  + join a channel
  + make donations
  + follow other registered users
  + send chat messages
  + check donations made/received
  + search for channels of specific type
  + search for specific user
  + join a channel
  + register
  + get top channels by donations received
  + get top channels by number of viewers
* **moderator**:
  + mute other viewers for a given duration
* **streamer**:
  + start/close stream
  + grant moderator privileges
  + create channels

**List of SQL Queries**

1. **Project and Selection**

select ID,streamer\_username from Channels where ID like '%$sid%'or streamer\_username like '%$sid%'

1. **Division**

Find users that all of your followees follow.

select u1.username

from Users u1,Follows f1

where u1.username=f1.followee\_username

AND f1.follower\_username

IN (select f2.followee\_username

from follows f2,Users u2

where u2.username=f2.follower\_username AND u2.username=’$uname’)

1. **Aggregation**

Find the number of viewers in this channel.

select count(id) from watches where id='$CID'

Find the amount of donations made to other users.

select sum(amount) from donations where donater\_username='$target\_name'

1. **Nested Aggregation**

Find top 10 channels by the number of viewers or by the amount of donation received from other users.

select \* from

(select c.id,c.title,count(w.watcher\_username)

from Channels c,Watches w

where c.id=w.id

group by c.id,c.title

ORDER BY count(w.watcher\_username) desc)

where rownum<=10

select \* from

(select c.id,c.title,sum(d.amount)

from Donations d,Channels c

where c.streamer\_username=d.donatee\_username

group by c.id,c.title

ORDER BY sum(d.amount) desc)

where rownum<=10

1. **Deletion**

delete from mutes where endtime < TO\_DATE('".$date."','YYYY/MM/DD')

delete from watches where id='$CID' and watcher\_username='$uname'