

# CSC261 Project Milestone 2

## Project Details

**Name of the project:** JsTube

**Team number:** 17

**Team name:** Jays Club

**Group members:**

- Junfei Liu (jliu137): algorithm & schema (33%)
- Jinghan Lu (jlu54): schema & mapping (33%)
- Jinghao Jiang (jjiang27): algorithm & er (33%)

## Task A: Draw an ER Diagram

### Entities:

- **videos**(video\_name:varchar(50), id:int, user\_id:int, upload\_time:datetime, membership\_requirement: boolean, link:varchar(2083))
- **user**(id:int, user\_name:varchar(20), email:varchar(20), password:varchar(20), membership: boolean)
- **membership**(user\_id: int, level: int, valid\_through: date)
- **administrator**(id:int, password: varchar (20), email:varchar(20))
- **case**(case\_id: int, administrator\_id: int, type:int,video\_id:int)

### Relations:

- User — **upload** — video
  - Each user may not upload a video or a user can upload unlimited number of videos, so there should be (0, N) uploads
  - Each video must and must only be uploaded by one user, so it is (1,1)
- User — **view** — video
  - Each user can choose to not watch any video or watch an infinite amount of videos, so it should be (0, N)
  - Each video can be viewed by many users but can also not be viewed by any user so it is (0, M)
- Video — **require** — membership
  - Each video may or may need a membership to be viewed, and there is at most one membership can be requires, so it is (0, 1)
  - Each membership status might be or might not be needed to view the video, so it is (0, 1)
- User — **have** — membership
  - Given membership is a boolean variable, a user can either have or not have a membership, so it is (0, 1)



## Task B: Relational Database Design Using ER-to-Relational Mapping

### Step 1: Mapping of Regular Entity Types

All entities in the ER diagram are regular (strong) entities and the primary keys are underlined in the table below.

#### 1) USER

<u>User_id</u>	User_name	Email	Password	Membership
----------------	-----------	-------	----------	------------

#### 2) VIDEO

Video_name	<u>Video_id</u>	User_id	Upload_day	Upload_month	Upload_year	Membership_requirement	Link
------------	-----------------	---------	------------	--------------	-------------	------------------------	------

#### 3) MEMBERSHIP

<u>User_id</u>	Level	Valid_to_day	Valid_to_month	Valid_to_year
----------------	-------	--------------	----------------	---------------

#### 4) ADMINISTRATOR

<u>Administrator_id</u>	Email	Password
-------------------------	-------	----------

#### 5) CASE

<u>Case_id</u>	Administrator_id	Video_id	Type
----------------	------------------	----------	------

### Step 2: Mapping of Weak Entity Types

Since there is no weak entity type in the ER Diagram, this step is not applicable.

### Step 3: Mapping of Binary 1:1 Relationship Types

- `have` relationship between USER and MEMBERSHIP
  - Given MEMBERSHIP has total participation on the relationship, choose MEMBERSHIP as *S* and include the primary key of USER, `User_id`, into MEMBERSHIP, which is equivalent to `user_id` and name it as `member_id`
- `is_reported_as` relationship between CASE and VIDEO
  - Given CASE has total participation on the relationship, choose CASE as *S* and include the primary key of VIDEO into CASE and name it as `Report_user_id`

### Step 4: Mapping of Binary 1:N Relationship Types

- `upload` relationship between USER and VIDEO

- Given VIDEO is on the N side, choose VIDEO as relation *S* and then include the primary key of USER into VIDEO, which is the same as `user_id`
- `manage` relationship between ADMINISTRATOR and CASE
  - Given CASE is on the N side, choose CASE as relation *S* and then include the primary key of ADMINISTRATOR into CASE, which is the same as `Administrator_id`
- `report` relationship between USER and CASE
  - Given CASE is on the N side, choose CASE as relation *S* and then include the primary key of USER into CASE and name it `Report_user_id`

### Step 5: Mapping of Binary M:N Relationship Types

- `view` relationship between USER and VIDEO
  - Create a new relation VIEW, add and combined the primary key of USER and VIDEO, `User_id` and `Video_id`, to form the primary key, and include the attribute `Times`

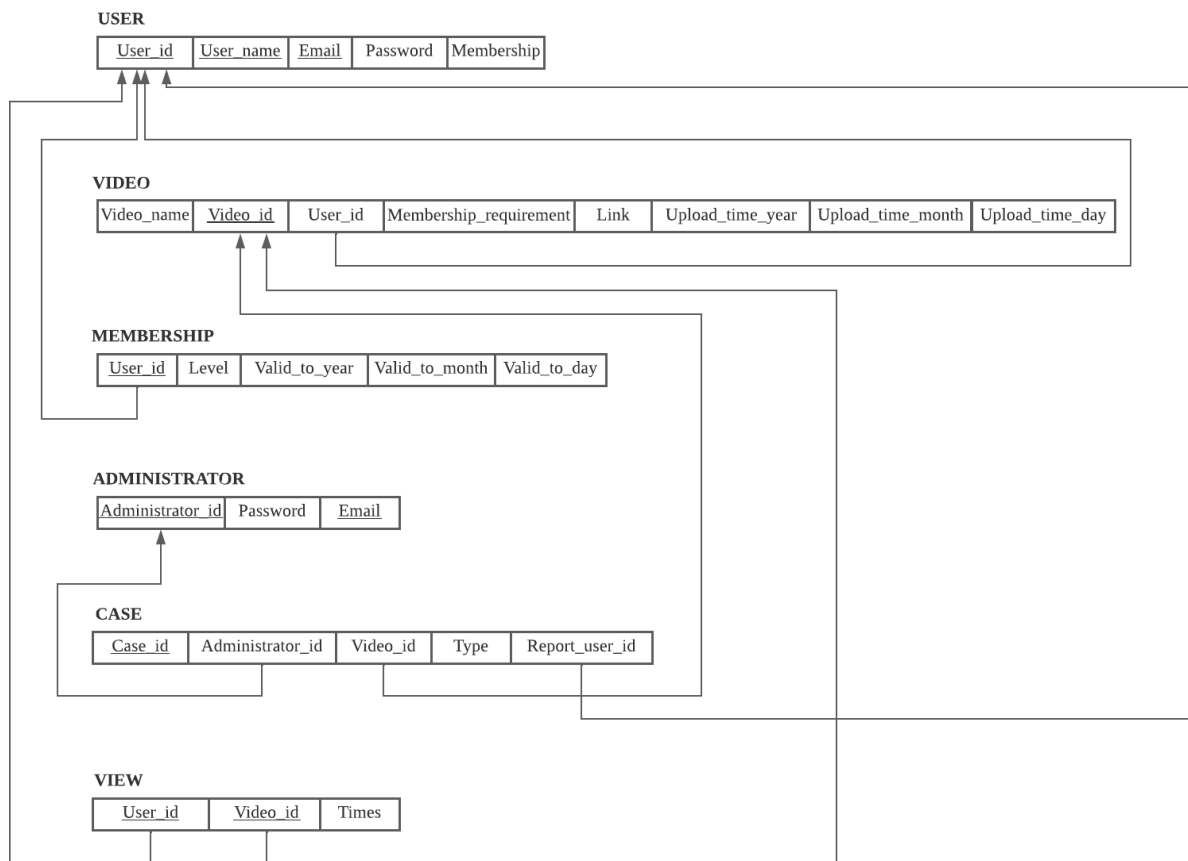
### Step 6: Mapping of Multivalued Attributes

There is no multivalued attribute in the ER Diagram, so this step is not applicable.

### Step 7: Mapping of *N*-ary Relationship Types

There is no *N*-ary attribute in the ER Diagram, so this step is not applicable.

Relation Name	RE Diagram Component
USER	E(USER)
VIDEO	E(VIDEO) + R(upload)
MEMBERSHIP	E(MEMBERSHIP) + R(has)
ADMINISTRATOR	E(ADMINISTRATOR)
CASE	E(CASE) + R(manage) + R(report) + R(is_reported)
VIEW	R(view)



## Schema of our database

### VIDEO

VIDEO				
Video_name		Title of video	varchar(50)	Can be NULL
Video_id	Primary key	Unique id of each video	int	
User_id	Foreign key	Unique id of user who uploaded the video	int	
Upload_time_year		Time the video was uploaded	datetime	
Upload_time_m				

onth				
Upload_time_day				
Membership_requirement		Whether the video requires a membership for user to view	boolean	Default is FALSE
Link		Unique link of the video	varchar(2083)	

Foreign key action:

- User\_id - set NULL

When user with the given user\_id is deleted from the USER table, the data of the specific video can be still be kept in our database

## USER

USER				
Id	Primary key	Unique id of user	int	
User_name		Name of user	varchar(20)	
Email		Login email	varchar(20)	
Password		Login password	varchar(20)	
Membership		Join or not	boolean	Default is FALSE

## MEMBERSHIP

MEMBERSHIP				
User_id	Foreign key Primary key	Unique id of user	int	
Level		Depends on how the user join a membership	int	Default is 0
Valid_to_year		Expiration time	date	

Valid_to_month				
valid_to_day				

Foreign key action:

- User\_id - DELETE CASCADE  
When the user unregistered, delete the membership information from our database

## ADMINISTRATOR

ADMINISTRATOR				
Administrator_id	Primary key	Unique id of administrator	int	
Password		Login password	varchar(20)	
Email		Login email	varchar(20)	

## CASE

CASE*				
Case_id	Primary key	Unique case id	int	
Administrator_id	Foreign key	Administrator who is responsible for this case	int	DEFAULT is id for administrator manager 0001
Video_id	Foreign key	Video involved in the case	int	
Type		Case type	int	

\*report case to administrators

Foreign key action:

- Administraot\_id - **set** DEFAULT  
When no administrator is assigned, the case will, by default, assigned to the manager
- Video\_id - DELETE CASCADE  
When the video which is involved in a case is deleted, then the case no longer exists

## VIEW

VIEW
------

User_id	Foreign key	Who viewed the video	int	
Video_id	Foreign key	Unique video id	int	
Times		The number of times a user view a certain video	time	Default is 0

Foreign key action:

- User\_id - no action  
Though the user registered from the platform, their viewing history will still be kept to for the purpose of documenting the overall views for a given video
- Video\_id - DELETE CASCADE  
Video has been deleted, so there is no need to keep the video viewing data.