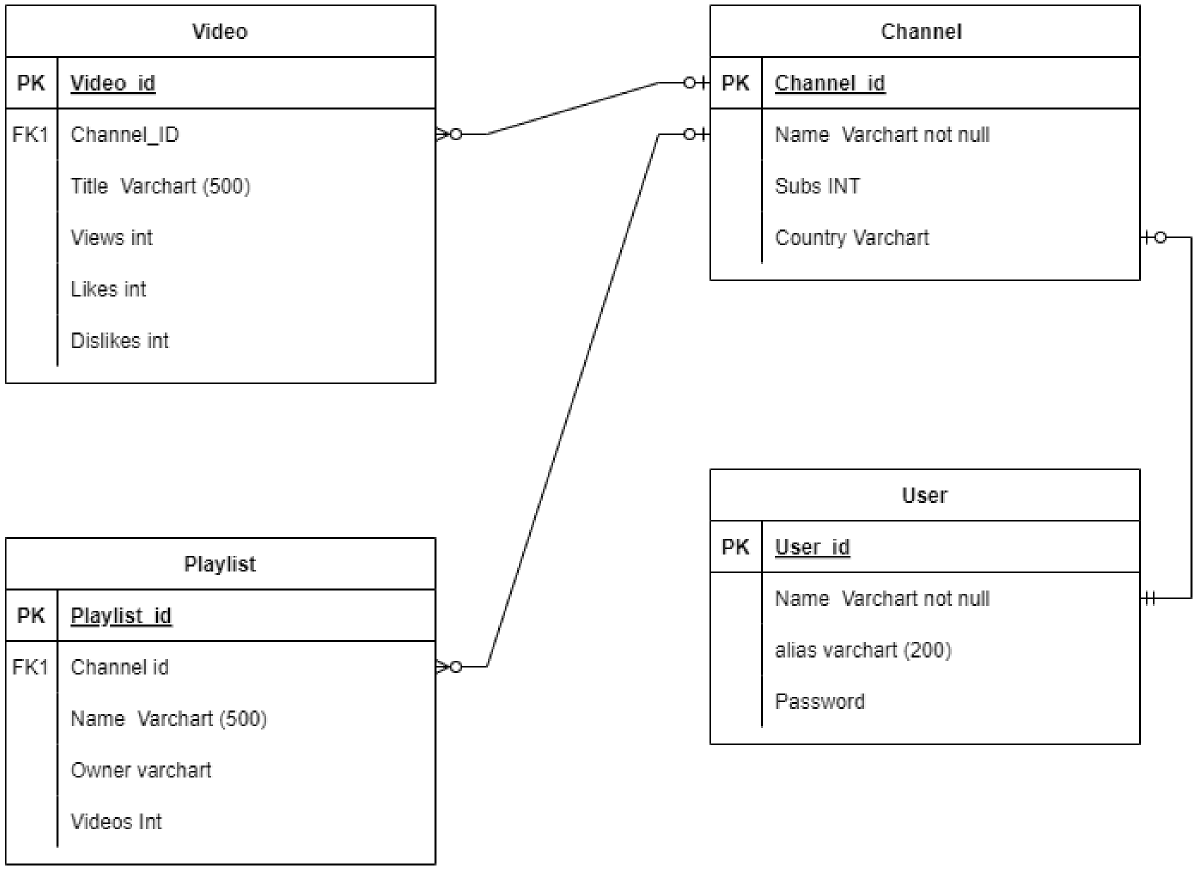
David Santiago Buitrago C. 20221020085

Workshop 3

1.



2.
For the develop of this database model, i used the following steps

1. Types of data:

I used this types of data for the development of the DM
Varchart:For attributes like title, description, username, email address, and playlist title.
Integer: For numerical values like view count, like count, dislike count, subscriber count, and duration (in seconds).

2. Entities

Video: Represents a single video uploaded to the platform. It contains attributes such as video ID, title, views, likes, dislikes, and channel id
Channel: Represents a user's channel on the platform. It includes attributes like channel ID, name, subscribers, and country
User: Represents a registered user of the platform. It includes attributes like user ID, username(alias) , password (hashed).
Playlist: Represents a collection of videos curated by a user. It includes attributes like playlist ID, name , User id, videos and channel id.

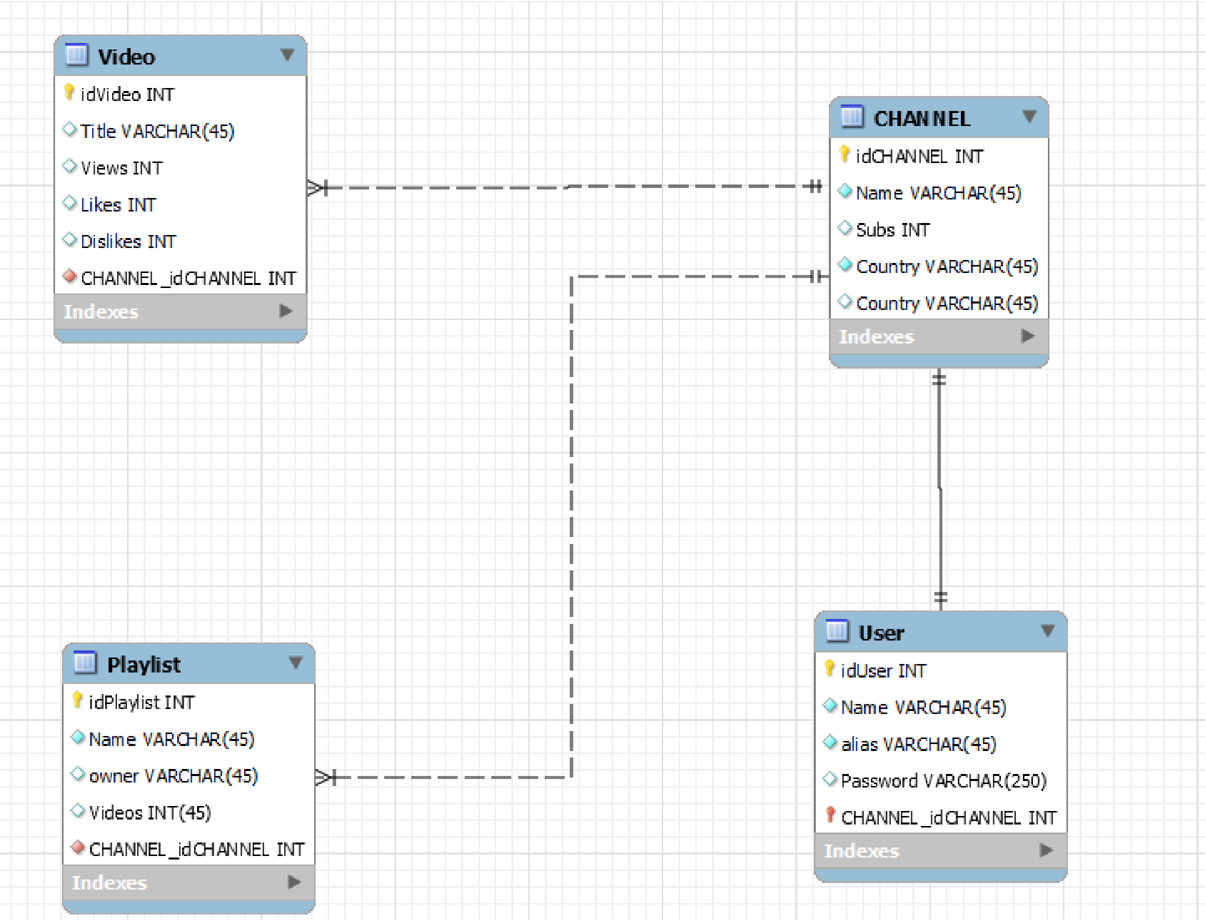
3. Relationships

User-Channel: One user can create a channel but the user is not obligated to create a channel and a channel just have a user.
Channel-Video: A channel can have a lot videos but a video only **belongs to** a single channel
Channel- playlist :.A channel can have a lot PLAYLIST but a playlist only **belongs to** a single channel

4.

In this database model, i used 1 foregoing key that was channel id.

3. Screenshot of dbms (MYSQL)



4. SQL Script

```
CREATE TABLE IF NOT EXISTS `YoutubeDB`.`CHANNEL` (
  `idCHANNEL` INT NOT NULL AUTO_INCREMENT,
  `Name` VARCHAR(45) NOT NULL,
  `Subs` INT NULL,
  `Country` VARCHAR(45) NOT NULL,
  `Country` VARCHAR(45) NULL,
  PRIMARY KEY (`idCHANNEL`))
ENGINE = InnoDB
```

```
# -*- coding: utf-8 -*-
# MySQL Workbench Python script
# <description>
# Written in MySQL Workbench 8.0.34

import grt
#import mforms

CREATE TABLE IF NOT EXISTS `YoutubeDB`.`Playlist` (
  `idPlaylist` INT NOT NULL,
  `Name` VARCHAR(45) NOT NULL,
  `owner` VARCHAR(45) NULL,
  `Videos` INT(45) NULL,
  `CHANNEL_idCHANNEL` INT NOT NULL,
  PRIMARY KEY (`idPlaylist`),
  INDEX `fk_Playlist_CHANNEL1_idx` (`CHANNEL_idCHANNEL` ASC) VISIBLE,
  CONSTRAINT `fk_Playlist_CHANNEL1`
    FOREIGN KEY (`CHANNEL_idCHANNEL`)
    REFERENCES `YoutubeDB`.`CHANNEL` (`idCHANNEL`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB

CREATE TABLE IF NOT EXISTS `YoutubeDB`.`User` (
  `idUser` INT NOT NULL AUTO_INCREMENT,
  `Name` VARCHAR(45) NOT NULL,
  `alias` VARCHAR(45) NOT NULL,
  `Password` VARCHAR(250) NULL,
  `CHANNEL_idCHANNEL` INT NOT NULL,
  PRIMARY KEY (`idUser`, `CHANNEL_idCHANNEL`),
  INDEX `fk_User_CHANNEL1_idx` (`CHANNEL_idCHANNEL` ASC) VISIBLE,
  CONSTRAINT `fk_User_CHANNEL1`
    FOREIGN KEY (`CHANNEL_idCHANNEL`)
    REFERENCES `YoutubeDB`.`CHANNEL` (`idCHANNEL`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
```

```
ENGINE = InnoDB

CREATE TABLE IF NOT EXISTS `YoutubeDB`.`Video` (
  `idVideo` INT NOT NULL,
  `Title` VARCHAR(45) NULL,
  `Views` INT NULL,
  `Likes` INT NULL,
  `Dislikes` INT NULL,
  `CHANNEL_idCHANNEL` INT NOT NULL,
  PRIMARY KEY (`idVideo`),
  INDEX `fk_Video_CHANNEL_idx` (`CHANNEL_idCHANNEL` ASC) VISIBLE,
  CONSTRAINT `fk_Video_CHANNEL`
    FOREIGN KEY (`CHANNEL_idCHANNEL`)
    REFERENCES `YoutubeDB`.`CHANNEL` (`idCHANNEL`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
```