Technical Report and ERM

DataBase Foundations

Professor:

Eng. Carlos Andrés Sierra, M.Sc.

Author:

Montoya Amaya Oscar Santiago - 20212020076

Hernández Manosalva Adiel Valentín - 20201020144

Francisco José de Caldas District University

Faculty of Engineering

Systems Engineering

September 2024

Introduction

This report presents a detailed design of the Twitch database, the objective of which is to efficiently model the main entities and relationships that support the daily operation of the service. The primary focus is on identifying key platform components such as users, channels, videos, and messaging, as well as critical functionalities related to subscriptions, donations, and interactive events.

Throughout this document, the most important entities that make up the data model are broken down, along with their respective attributes, in order to provide a coherent and adaptable structure.

Twitch Business Model

Twitch is a live-streaming platform that generates revenue through subscriptions, donations, advertising, and brand partnerships.

User Ecosystem:

The Twitch ecosystem includes three main types of users:

- Streamers: Content creators who broadcast live streams and monetize through subscriptions, donations, and ads.
- Viewers: Users who watch streams, engage via chat, subscribe to their favorite channels, and donate to support streamers.
- Brands/Advertisers: Companies that reach Twitch's audience by running ads during streams or sponsoring streamers, taking advantage of the platform’s highly engaged user base.

Value Proposition

- For streamers, Twitch provides a platform to build an audience and monetize their content through subscriptions, donations, and ads.
- For viewers, it offers real-time interaction with streamers and access to exclusive content through subscriptions and community engagement.
- For brands and advertisers, Twitch offers access to a highly engaged and niche audience. Brands can place targeted ads, sponsor popular streamers, and interact with viewers through authentic, live content.

Revenue Streams

- Subscriptions: Viewers pay monthly fees to support their favorite streamers and access exclusive perks.
- Bits (Donations): Viewers purchase and donate virtual items called Bits to show support.
- Ads: Twitch runs pre-roll and mid-roll ads during streams, sharing the revenue with streamers.
- Brand Partnerships: Streamers collaborate with brands for sponsorships, product placements, and promotional activities.

Processes and Information Required in the Application

Key Processes

- User registration: Both streamers and viewers need to create accounts, providing details such as email, username, and payment information.
- Streaming process: A streamer starts a live broadcast, which is then made available to viewers in real time.
- Real-time interaction: Viewers can engage via chat, subscribe to streamers, or send donations (Bits).
- Monetization and payouts: Streamers receive payouts based on subscriptions, donations, and ads.

Required Information

- User data: Name, email, payment details.
- Streaming data: Stream ID, duration, content category, number of viewers.
- Financial data: Subscription payments, donations, ad revenue distribution.

Steps

Designing an efficient and well-structured database for a platform as complex as Twitch requires a methodical approach. To ensure that all key aspects of the platform are properly represented and that the relationships between the various components are clear, several key steps are followed.

In this process, we begin by identifying the main components that define how Twitch works, such as users, channels, videos, subscriptions, and other critical elements. From these components, we proceed to define the entities that will be responsible for modeling these elements in the database, as well as the attributes that describe the specific characteristics of each entity.

Finally, the information is organized, assigning the corresponding attributes to each entity to ensure the consistency and integrity of the data model, where everything is seen.

1. Define components

a. User Management
b. Streaming
c. Video (VOD) Management
d. Chat System
e. Subscriptions
f. Followers
g. Donations
h. Moderation
i. Emotes and Interaction
j. Reports and Violations

2. Define entities

a. Users (**e1**)
b. Streams (**e2**)
c. Videos (VODs) (**e3**)
d. Categories (**e4**)
e. Followers (**e5**)
f. Subscriptions (**e6**)
g. Chats (**e7**)
h. Donations (**e8**)
i. Emotes (**e9**)

j. Moderators (**e10**)
k. Reports (**e11**)

3. Define attributes per entity

a. Users

i. username
ii. email
iii. password hash
iv. account type (streamer/viewer)
v. sign up date
vi. profile picture v
vii. bio

b. Streams

i. title
ii. category (game/creative/music/etc.)
iii. start time
iv. end time
v. view count
vi. status (live/offline)

c. Videos (VODs)

i. video url
ii. upload date
iii. view count
iv. duration
v.

d. Categories

i. name
ii. description

e. Followers

i. follow date

f. Subscriptions

i. subscription date
ii. subscription type (tier1/tier2/tier3)

g. Chats

i. message
ii. timestamp

h. Donations

i. amount
ii. currency
iii. donation date

i. Emotes

   i. emote name
   ii. image url

j. **Moderators**
   i. assigned date

k. **Reports**
   i. reason
   ii. report date
   iii. status (pending/resolved/closed)

4. Define relationships

	e1	e2	e3	e4	e5	e6	e7	e8	e9	e10	e11
e1		✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
e2	✔		✔	✔			✔	✔		✔	
e3	✔	✔		✔							
e4	✔	✔	✔								
e5	✔					✔		✔		✔	✔
e6	✔				✔						
e7	✔	✔						✔	✔	✔	✔
e8	✔	✔			✔		✔				
e9	✔						✔				
e10	✔	✔			✔		✔				✔
e11	✔				✔		✔			✔	

5. Define relationships types

Types

- **1 to many (1--- M)**
- **many to many (M---M)**
- **1 to 1 (1---1)**

Users (e1):

- **e1 to e2 (Streams): 1---M**
- **e1 to e3 (Videos): Indirect (through Streams)**
- **e1 to e4 (Categories): Indirect (through Streams)**
- **e1 to e5 (Followers): M---M**
- **e1 to e6 (Subscriptions): M---M**
- **e1 to e7 (Chats): 1---M**
- **e1 to e8 (Donations): 1---M**
- **e1 to e9 (Emotes): M---M**
- **e1 to e10 (Moderators): 1---M**
- **e1 to e11 (Reports): 1---M**

Streams (e2):

- **e2 to e3to e4 (C (Videos):** 1---M
- **e2 (Categories):** M---1
- **e2 to e7 (Chats):** 1---M
- **e2 to e8 (Donations):** 1---M
- **e2 to e10 (Moderators):** M---M

Videos(e3)

- **e3 to e4 (Categories):** M---1

Followers (e5):

- **e5 to e6 (Subscriptions):** M–1
- **e5 to e8 (Donations):** M–1
- **e5 to e10 (Moderators):** 1—1
- **e5 to e11 (Reports):**M—M

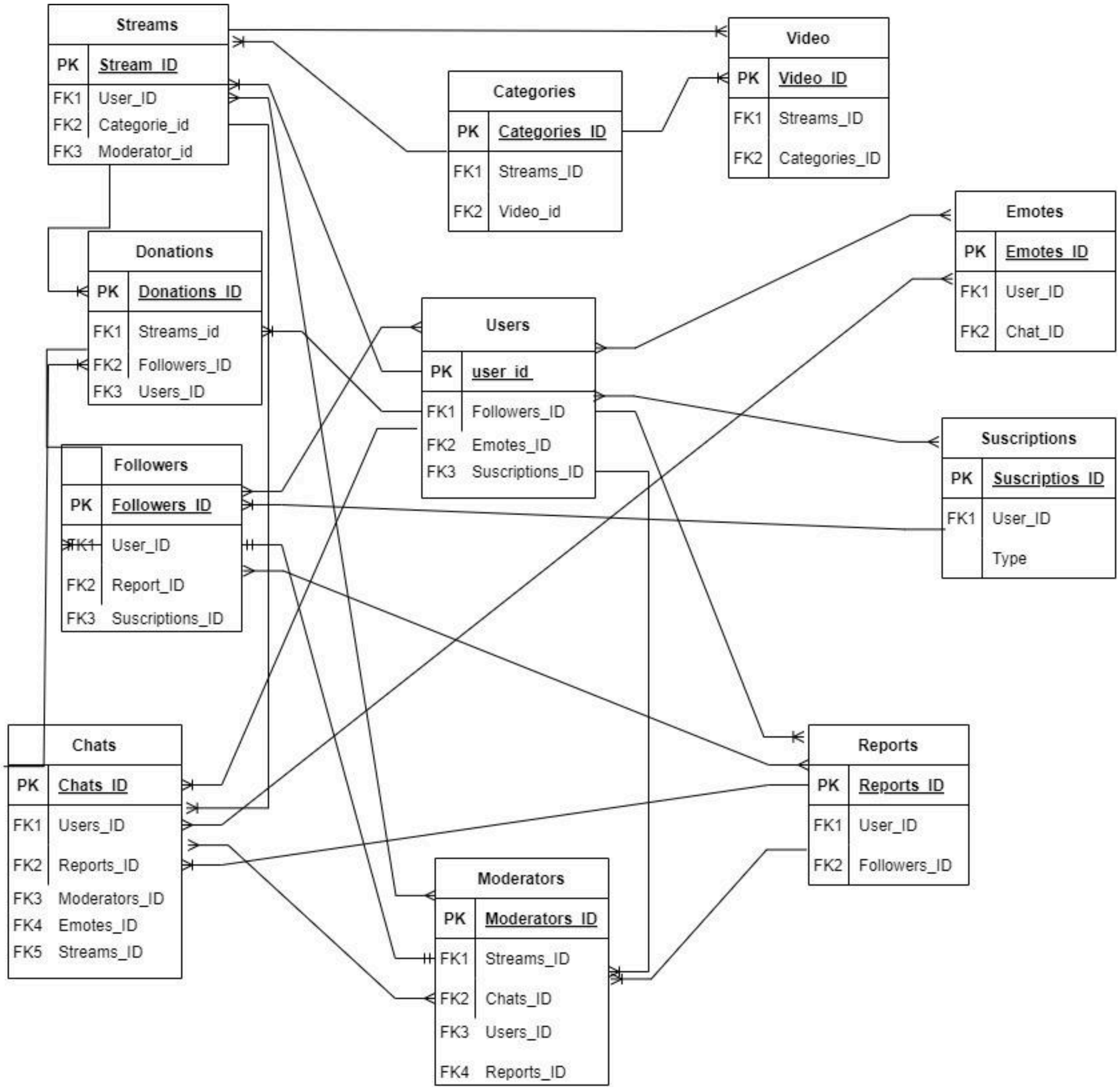
Chats (e7):

- **e7 to e8 (Donations):** 1—-M
- **e7 to e9 (Emotes):** M—M
- **e7 to e10 (Moderators):** M—M
- **e7 to e11 (Reports):** M—1

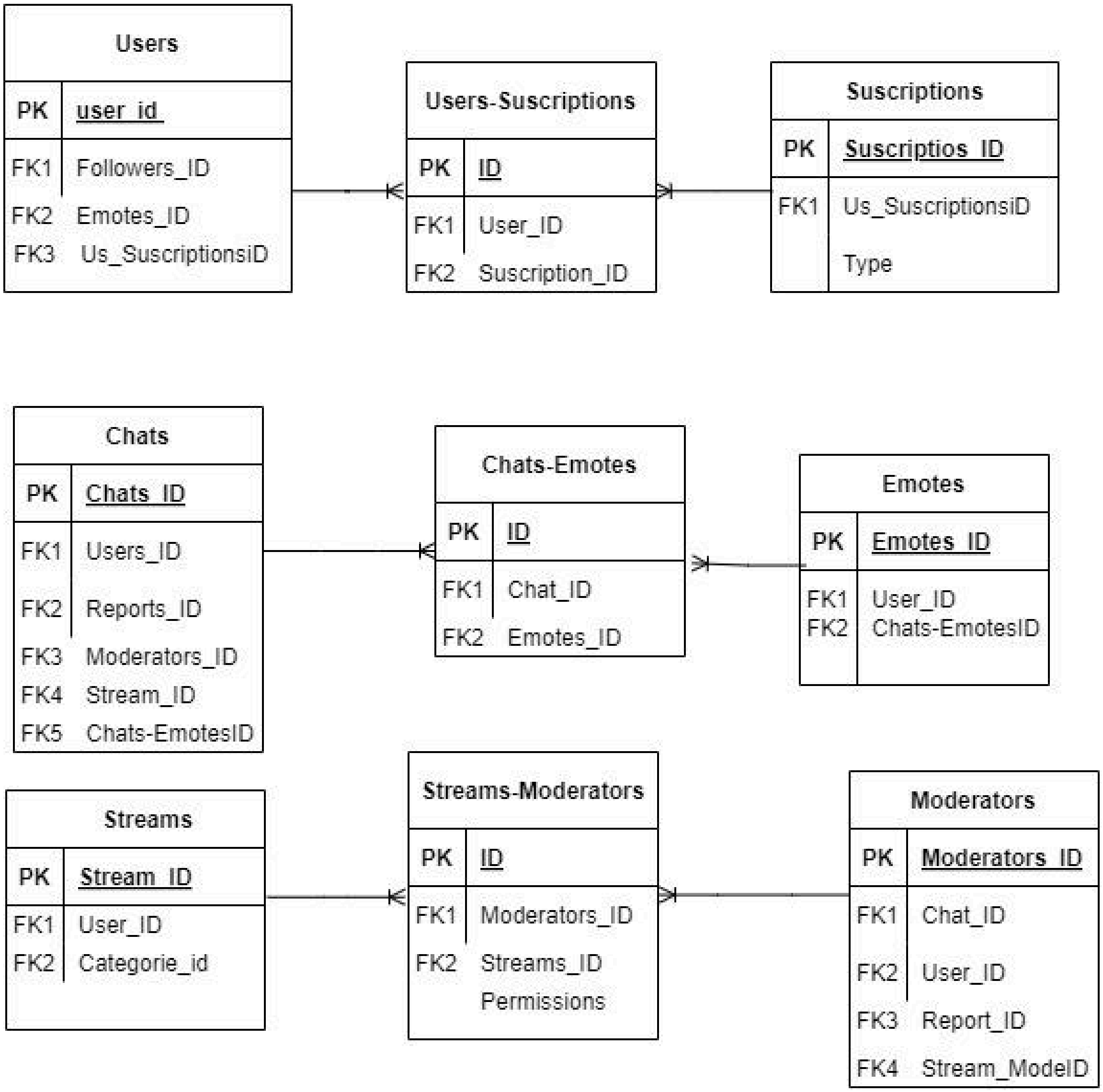
Moderators (e10)

- **e10 to e11(Reports):** M—1

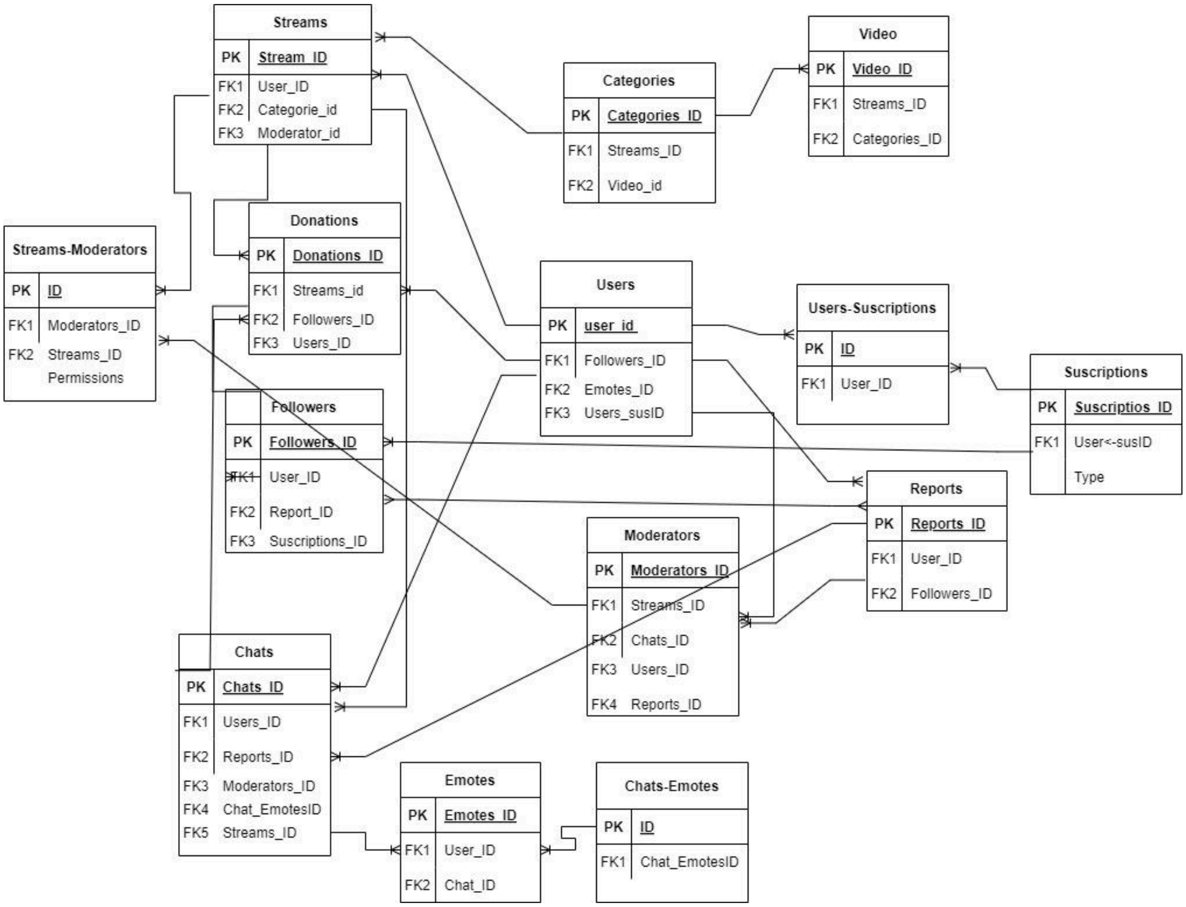
6. First Entity-Relationship Draw



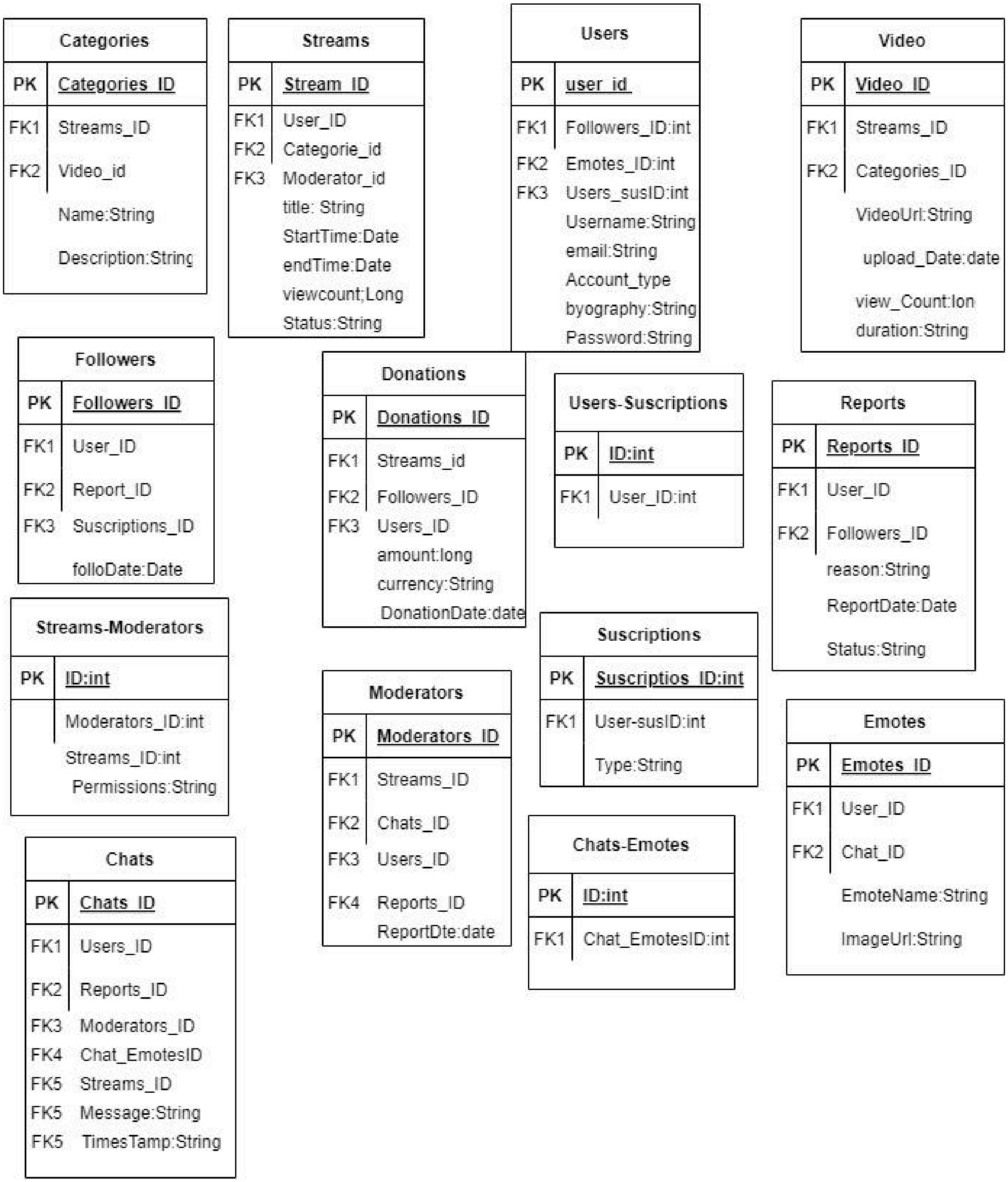
7. First Split Many-to-Many Relationships



8. Second Entity-Relationship Draw



9. Get Data-Structure E-R M



	Categories
PK	Categories_ID
FK1	Streams_ID
FK2	Video_id
	Name:String
	Description:String

	Streams
PK	Stream_ID
FK1	User_ID
FK2	Categorie_id
FK3	Moderator_id
	title: String
	StartTime:Date
	endTime:Date
	viewcount;Long
	Status:String

	Users
PK	user_id
FK1	Followers_ID:int
FK2	Emotes_ID:int
FK3	Users_susID:int
	Username:String
	email:String
	Account_type
	byography:String
	Password:String

	Video
PK	Video_ID
FK1	Streams_ID
FK2	Categories_ID
	VideoUrl:String
	upload_Date:date
	view_Count:lon
	duration:String

	Followers
PK	Followers_ID
FK1	User_ID
FK2	Report_ID
FK3	Suscriptions_ID
	folloDate:Date

	Donations
PK	Donations_ID
FK1	Streams_id
FK2	Followers_ID
FK3	Users_ID
	amount:long
	currency:String
	DonationDate:date

	Users-Suscriptions
PK	ID:int
FK1	User_ID:int

	Reports
PK	Reports_ID
FK1	User_ID
FK2	Followers_ID
	reason:String
	ReportDate:Date
	Status:String

	Streams-Moderators
PK	ID:int
	Moderators_ID:int
	Streams_ID:int
	Permissions:String

	Moderators
PK	Moderators_ID
FK1	Streams_ID
FK2	Chats_ID
FK3	Users_ID
FK4	Reports_ID
	ReportDte:date

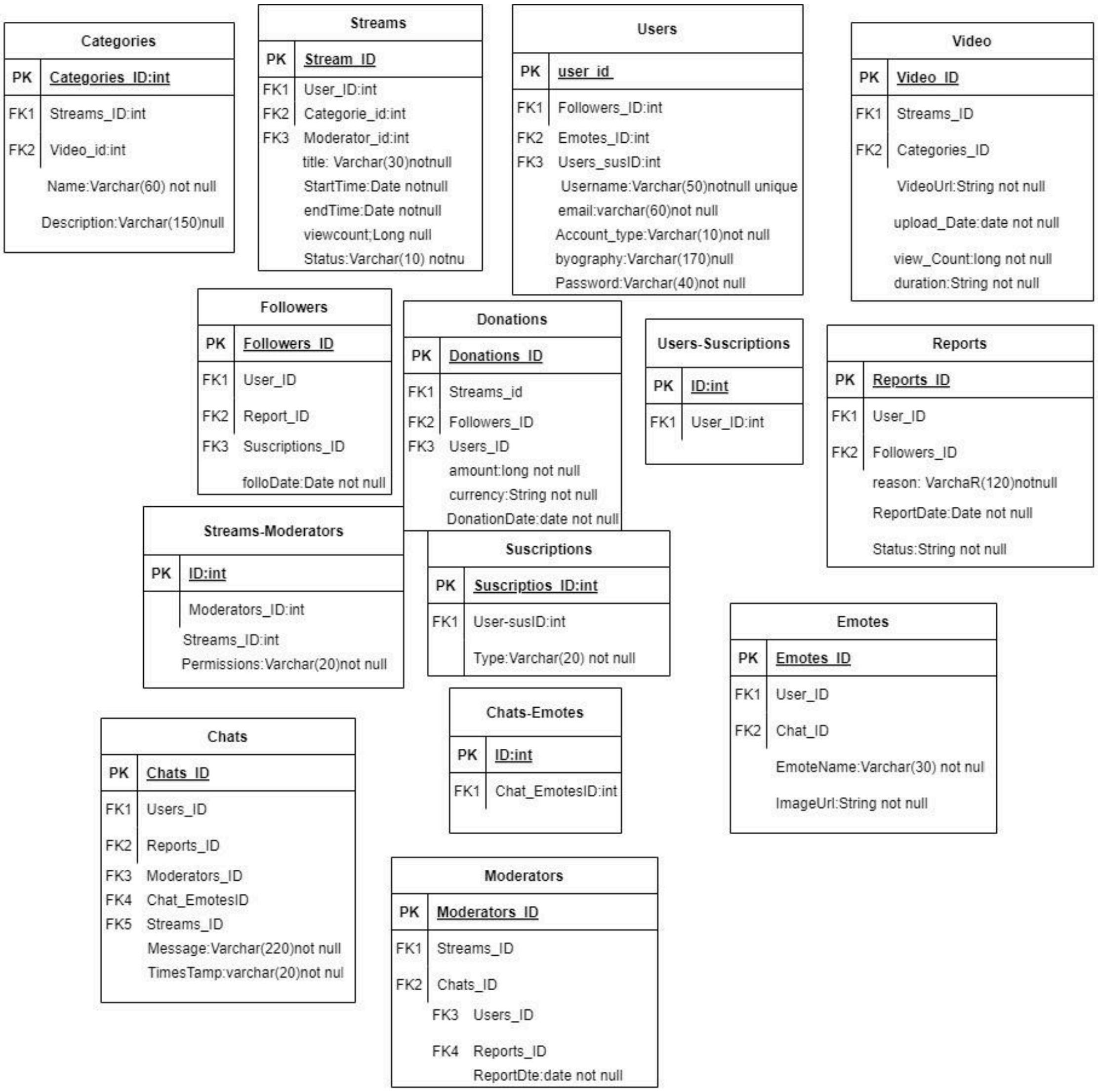
	Suscriptions
PK	Suscriptios_ID:int
FK1	User-susID:int
	Type:String

	Emotes
PK	Emotes_ID
FK1	User_ID
FK2	Chat_ID
	EmoteName:String
	ImageUrl:String

	Chats-Emotes
PK	ID:int
FK1	Chat_EmotesID:int

	Chats
PK	Chats_ID
FK1	Users_ID
FK2	Reports_ID
FK3	Moderators_ID
FK4	Chat_EmotesID
FK5	Streams_ID
FK5	Message:String
FK5	TimesTamp:String

10.Define Constraints and Properties of Data



	Categories
PK	Categories_ID:int
FK1	Streams_ID:int
FK2	Video_id:int
	Name:Varchar(60) not null
	Description:Varchar(150)null

	Streams
PK	Stream_ID
FK1	User_ID:int
FK2	Categorie_id:int
FK3	Moderator_id:int
	title: Varchar(30)notnull
	StartTime:Date notnull
	endTime:Date notnull
	viewcount;Long null
	Status:Varchar(10) notnu

	Users
PK	user_id
FK1	Followers_ID:int
FK2	Emotes_ID:int
FK3	Users_susID:int
	Username:Varchar(50)notnull unique
	email:varchar(60)not null
	Account_type:Varchar(10)not null
	byography:Varchar(170)null
	Password:Varchar(40)not null

	Video
PK	Video_ID
FK1	Streams_ID
FK2	Categories_ID
	VideoUrl:String not null
	upload_Date:date not null
	view_Count:long not null
	duration:String not null

	Followers
PK	Followers_ID
FK1	User_ID
FK2	Report_ID
FK3	Suscriptions_ID
	folloDate:Date not null

	Donations
PK	Donations_ID
FK1	Streams_id
FK2	Followers_ID
FK3	Users_ID
	amount:long not null
	currency:String not null
	DonationDate:date not null

	Users-Suscriptions
PK	ID:int
FK1	User_ID:int

	Reports
PK	Reports_ID
FK1	User_ID
FK2	Followers_ID
	reason: VarchaR(120)notnull
	ReportDate:Date not null
	Status:String not null

	Streams-Moderators
PK	ID:int
	Moderators_ID:int
	Streams_ID:int
	Permissions:Varchar(20)not null

	Suscriptions
PK	Suscriptios_ID:int
FK1	User-susID:int
	Type:Varchar(20) not null

	Emotes
PK	Emotes_ID
FK1	User_ID
FK2	Chat_ID
	EmoteName:Varchar(30) not nul
	ImageUrl:String not null

	Chats
PK	Chats_ID
FK1	Users_ID
FK2	Reports_ID
FK3	Moderators_ID
FK4	Chat_EmotesID
FK5	Streams_ID
	Message:Varchar(220)not null
	TimesTamp:varchar(20)not nul

	Chats-Emotes
PK	ID:int
FK1	Chat_EmotesID:int

	Moderators
PK	Moderators_ID
FK1	Streams_ID
FK2	Chats_ID
FK3	Users_ID
FK4	Reports_ID
	ReportDte:date not null