CSCI 585: Database Systems HW1

UCS ID: 3725520208
USC Email: annes@usc.edu

Description of the (E)ER Diagram for YouTube.

1. The **Users** table contains all the basic details of the users such as First_name, Last_name, Street_address, City, State, Country, Email, age, **Is_a_creator?** Which will determine if the User is a content creator, **Is_a_consumer?** Which shall decide if the User is a content consumer.
2. The **primary key** for the **Users** table shall be **UniqueID** which is a unique and not null value.
3. Since a user can be both a content creator and a content consumer, they shall result in an **overlapping entity subtype**. As they are overlapping subtype entities, **both the columns Is_a_creator?** and **Is_a_consumer? Will be required for subtype classification** of the users **into Video_creators and Video_consumers** tables.
4. Since there are only two categories of users (Video_Creators and Video_Consumers), we use two lines to underline the subtype discriminator(**total completeness**).
5. The **UniqueID** (primary key of the Users table) shall also be **passed on to the child tables Video_creators and Video_consumers as a primary key.**
6. The **Video_creators** table contains details about the revenue each creator has earned, the total number of subscribers the creator has, and the number of videos uploaded.
7. On the other hand, the **Video_consumers** table contains information about the content consumers, like the number of subscriptions a consumer has, the amount_of_time_spent on watching videos, and the number_of_liked_videos.
8. The channels act as dashboards for content creators to publish their videos. The **Channels** table contains data like channel_name, channel_description, channel_creation_date, number_of_subscribers, no_of_videos_uploaded, total_views, and type of subscriptions that the Channel has like free or paid that are defined by the columns, has_a_paid_subscription? And has_a-free_subscription? For every Channel when it is created, a random ID is generated (**primary key) called ChannelID**, and the owner of the Channel is identified using the **UniqueID (foreign key)** column. This foreign key can be used to extract the user/owner information from the **Users table.**
9. **Every Video_Creator can have multiple channels, but every Channel can have one and only one Video_Creator. So, it is a (1:N) relationship**. Assuming a user is designated as a video creator, if they shall own at least one Channel.
10. **Video_Consumers can subscribe to 0 or more channels, and every Channel can have '0' or more subscribers, so the relationship between the Channels and Video_Consumers is (1:N).**
11. The **Subscriptions** table acts as a bridge between the **Channels and the Video_Consumers.** The **SubscriptionID** is the primary key given to each subscription, the foreign key **ChannelID** helps to

identify to which **Channel** a particular subscription belongs, and the Foreign key **UniqueID** helps to identify the **User** who has subscribed to that specific Channel.

12. The column **Subscription_type** column will be used to identifies the type of subscription the user has taken.
13. Videos are an integral part of YouTube, and **every channel can host either 0 or more videos, so it is a (1:N) relationship.** All the **metadata regarding a video** is stored in the **Videos** table.
14. In the **Videos** table, the **VideoID** acts as a primary key. Foreign key **UniqueID** is used to determine the video's owner, and foreign key **ChannelID** is used to determine to which Channel a video belongs. **Apart from this** information, the **Videos** table **also contains data** about the Video_url, Video_duration, Video_title, Video_description, Video_upload_date, Video_upload_time, Video_category, number_of_likes, number_of_dislikes, number_of_comments, number_of_shares, number_of_views and since relational DBs cannot store images directly, the attribute Video_thumbnail_image_location can identify the location of the video thumbnail image.
15. Every video can be categorized into different categories like Entertainment and Informational. Since these **are disjoint subtype entities, the field Video_category can be used as a subtype discriminator to split tables into the Informational_Videos table and the Entertainment_Videos table and other categories as well.**
16. Since there can also be other subtypes, we represent the subtype discriminator with only a single line (**partial completeness**).
17. Every **Video can have either 0 or more comments, so it is a (1:N) relationship**. The details of the comments are stored in a separate table called **Comments**. The **CommentID** column is the primary key, which shall be unique for each comment, the **VideoID** will be used to determine to which video a particular comment belongs, and the **UniqueID** foreign key column will be helpful to determine the User who has put up that comment.
18. Apart from this information, the **Comments** table contains the comment_text, comment_date, comment_time, number_of_likes_for_comment, number_of_dislikes_for_comment, and comment_sentiment data for every comment that is recorded.
19. In the **Sponsors** table, the **SponsorID** acts as a primary key, It has information about Sponsor_name, Sponsor_address, Sponsor_email_id, Sponsor_phone.
20. The **Sponsored_Videos** table has details of the Sponsor(**SponsorID**), Video(**VideoID**) and the amount sponsored for a specific video**.**
21. **The relationship between the Videos and the Sponsored_Videos table shall be (1:1) relationship and relationship between the Sponsors and the Sponsored_Videos table shall be (1:M) relationship.**