Experiment No. 1

Group: - 1 from T4 Batch

0077 – Aakash Joshi

2039 – Akanksha Lokhande

Aim: -

ER Modeling and Normalization:

Decide a case study related to real time application in a group of 2-3 students and formulate a problem statement for application to be developed. Propose a Conceptual Design using ER features using tools like ERD plus, ER Win etc. (Identifying entities, relationships between entities, attributes, keys, cardinalities, generalization, specialization etc.) Convert the ER diagram into relational tables and normalize Relational data model.

Date: - 18/8/2023

Theory: -

ER Diagram: -

An Entity-Relationship (ER) diagram is a visual representation of the structure of a database, illustrating the relationships between different entities. Let's create an ER diagram for a Twitch database, incorporating four key tables: User, Stream, Video, and Chat.

1. User Table:

- The User table represents Twitch users and stores their personal information, such as username, email, and date of registration. It includes a primary key 'UserID' for uniquely identifying users.

2. Stream Table:

- The Stream table records information about live streams on Twitch. It includes attributes like 'StreamID' as the primary key, 'Title,' 'StreamURL,' and 'ViewerCount' to track live viewers.

3. Video Table:

- The Video table stores details about recorded videos or VODs (Video On Demand). Each video is associated with a stream and includes attributes like 'VideoID' (primary key), 'Title,' 'Duration,' and 'UploadDate.'

4. Chat Table:

- The Chat table represents the chat interaction during a live stream. It is linked to a specific stream using a foreign key 'StreamID' and contains attributes like 'ChatID' (primary key), 'Message,' 'Timestamp,' and 'UserID' (foreign key referring to the User table). This helps associate messages with specific users.

Relationships:

- Users are related to Streams through a one-to-many relationship. A user can have multiple streams, but each stream is associated with one user.

- Streams have a one-to-many relationship with Videos, as a stream can be recorded and turned into multiple videos.

- The Chat table has a many-to-one relationship with Streams, allowing multiple chat messages to be associated with a single live stream.

This ER diagram provides a clear understanding of how users, streams, videos, and chat messages are interconnected within the Twitch database, enabling efficient data management for the platform's streaming and social interaction features.

Normalization of Database: -

Normalization is the process of organizing a relational database in such a way that data redundancy is minimized. This ensures data integrity, reduces anomalies, and improves the efficiency of database operations. Database normalization is typically carried out in a series of steps, each referred to as a "normal form." There are several normal forms, from First Normal Form (1NF) to Fifth Normal Form (5NF). I'll explain each step-in detail:

1. First Normal Form (1NF):

- At this stage, each column in a table must contain only atomic (indivisible) values. There should be no repeating groups, and each row should be unique. For example, a table that stores information about people should have separate columns for first name and last name, rather than a single "Name" column with comma-separated values.

2. Second Normal Form (2NF):

- In 2NF, a table is already in 1NF, and it ensures that all non-key attributes (columns) are fully functionally dependent on the entire primary key. This means breaking the table into smaller tables, where each sub-table has a single purpose. For instance, if you have a table with student information that includes both student ID and course information, you'd create separate tables for students and courses.

3. Third Normal Form (3NF):

- 3NF builds on 2NF and further eliminates transitive dependencies. A transitive dependency occurs when a non-key attribute depends on another non-key attribute. In this step, you should ensure that non-key attributes are only dependent on the primary key. This may involve creating more tables and reorganizing the data structure to avoid redundancy.

4. Boyce-Codd Normal Form (BCNF):

   - BCNF is an advanced form of normalization that deals with certain types of anomalies. A table is in BCNF if, for every non-trivial functional dependency ($A \rightarrow B$), A is a superkey. In simpler terms, this means that for any non-trivial dependency, the left-hand side must be a superkey.

5. Fourth Normal Form (4NF):

   - 4NF deals with multi-valued dependencies. If a table has multi-valued attributes (attributes that can contain multiple values for a single row), 4NF is used to separate these attributes into their own tables.

6. Fifth Normal Form (5NF):

   - 5NF deals with join dependencies. It aims to minimize the number of joins required to retrieve data from a database. This form is rarely used in practice and is typically reserved for very complex data structures.

Each step of normalization aims to reduce data redundancy and minimize anomalies, but it's important to remember that normalization can sometimes lead to complex database structures and may not always be the best approach for all database systems. The level of normalization achieved depends on the specific requirements of the system and the trade-offs between data integrity and performance.

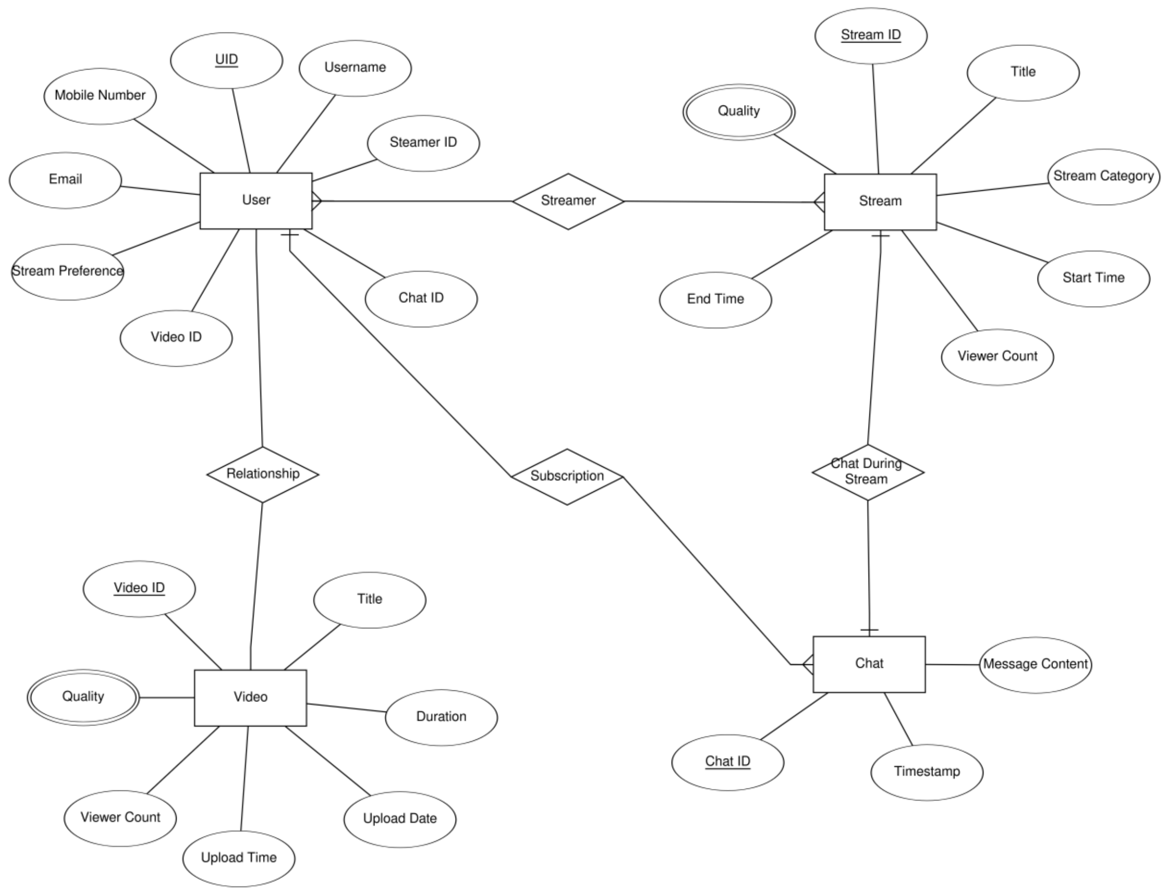
Conclusion:-

In conclusion, the process of ER modeling and normalization plays a pivotal role in designing efficient and robust relational databases for real-time applications. This topic, often undertaken by small teams of students, offers invaluable learning experience in database design and management.

Finally, the ER diagram is converted into relational tables, where the process of normalization comes into play. Normalization, in its various forms, helps eliminate data redundancy, ensures data integrity, and contributes to the overall efficiency of the database. Each step of normalization is a carefully calculated move to enhance the database structure, making it more scalable, maintainable, and performant.

In essence, the combination of ER modeling and normalization provides a solid foundation for creating databases that not only store and manage data effectively but also serve as a critical component in the development of real-time applications. This educational journey equips students with valuable skills for tackling complex data management challenges and lays the groundwork for sound database design principles.

Output: -


UID
Username
Mobile Number
Steamer ID
Email
User
Stream Preference
Chat ID
Video ID
Streamer
Stream ID
Title
Quality
Stream Category
Stream
End Time
Start Time
Viewer Count
Relationship
Subscription
Chat During Stream
Video ID
Title
Quality
Video
Duration
Viewer Count
Upload Time
Upload Date
Chat
Message Content
Chat ID
Timestamp