**Problem Statement**

**Product Dissection for top leading Platforms**

Welcome to this case study on dissecting and designing products for top leading platforms. In this case study, you will delve into the intriguing world of schema design for a prominent platform of your choice. Your task is to choose a top leading platform, research its features, and meticulously craft a schema design that encapsulates the essence of its functionality. By focusing on key entities, attributes, and relationships, you will gain invaluable insights into how data architecture drives the platform's effectiveness.

**Step 1: Choose a Leading Platform**

Select a leading platform that could span various domains such as social media, e-commerce, finance, or any other industry. This choice will form the foundation of your exploration into its schema design.

**Step 2: Research:**

Thoroughly research the platform you have selected. Investigate its core features, functionalities, and user interactions. Identify the top features that define its user experience and contribute significantly to its popularity.

**Step 3: Product Dissection and Real World Problems solved by the platform**

In this step, you will meticulously analyse the platform's standout features and how they provide innovative solutions to real-world challenges. By identifying key functionalities that resonate with users, you'll unravel how the platform effectively addresses problems and enhances user experiences. This dissection will serve as the foundation for understanding how the schema design aligns with the platform's core objectives.

**Step 4: Case Study on the real world problems and approach to solving them**

In this pivotal step, you will expand on the real-world challenges uncovered in Step 3 through a comprehensive case study. Delve into specific instances where users encountered difficulties and showcase how the platform's unique features provided effective solutions. By dissecting the approach taken by the platform to overcome these challenges, you'll gain a deeper appreciation for the platform's user-centric design philosophy and how it shapes the schema design.

**Step 5: Schema Design Based on Top Features**

Based on the features you have identified, craft a schema design that reflects the platform's data structure. Focus on the key entities, attributes, and relationships that underpin the chosen features. Your schema should capture the essence of how the platform organises and utilises its data.

**Step 6: Rationale Behind the Design**

While creating the schema design, consider the rationale behind the platform's choices. Reflect on why certain entities and relationships were chosen and how they align with the platform's goals. This will help you understand the strategic decisions driving the schema's architecture.

**Step 7: Create an ER Diagram**

Utilise tools like the Miro platform or similar applications to create an illustrative Entity-Relationship (ER) diagram. This diagram should vividly depict the entities, attributes, and relationships present within your schema design. The ER diagram will serve as a visual representation of your insights.

**Step 8: Presentation of Findings**

Present your findings in a clear and concise manner. Showcase your understanding of how the schema design impacts the platform's functionality and user experience. Explain how your chosen features are integrated into the schema and how the schema's structure supports the platform's objectives.

**Task Details:**

1. **Answer Submission:** Your submission should include well-structured solutions for all provided questions related to product schema designs.
2. **Video Creation:** Create an informative and engaging video where you thoroughly explain the Case Study.
3. **Depth and Clarity:** Ensure your solutions are detailed and showcase your understanding of product schema design principles. Similarly, in the video, provide clear explanations that are easy to understand for a wide audience.
4. **Creativity Encouraged:** You are welcome to utilise visuals, diagrams, or creative elements to enhance the clarity and impact of your explanations.

**Note:**

1. Duplicate this document and proceed to write your solutions and prepare your video.
2. Include the video link in this document before final submission.

Best of luck in completing this project and showcasing your prowess in dissecting and designing product schema for leading platforms! **For reference, we have also conducted a case study on Instagram, which you can find below. This case study will provide you with valuable insights into how schema design plays a pivotal role in shaping the functionality and success of a prominent platform.**



**Product Dissection for Discord**

### **Discord: Company Overview**

**Founded:** May 13, 2015  
**Founders:** Jason Citron and Stanislav Vishnevskiy  
**Headquarters:** San Francisco, California, USA

#### **Vision and Mission**

* **Vision:** To create a platform that connects people with their communities and passions.
* **Mission:** To provide a communication tool where users can create spaces for real-time interaction through text, voice, and video.

### **Key Milestones**

1. **2015** - Discord launched as a voice communication tool primarily for gamers.
2. **2017** - Expanded to serve broader communities beyond gaming, introducing robust text and video features.
3. **2020** - Rebranded with the tagline "Your place to talk and hang out," signaling its evolution beyond gaming.
4. **2021** - Valued at over $15 billion after funding rounds; explored acquisition talks with Microsoft (deal did not materialize).
5. **2022 and Beyond** - Continued development of community features, Nitro subscriptions, and integrations with apps like YouTube and Spotify.
6. **2023**: Achieved over 300 million registered users and 150 million monthly active users.

### 

### 

### 

### **Core Features**

1. **Voice and Video Communication**: Low-latency, high-quality voice and video calls.
2. **Text Channels**: Organized discussions via topic-specific text channels.
3. **Community Servers**: Spaces for groups to gather and interact, each customizable with roles and permissions.
4. **Screen Sharing and Streaming**: Useful for gaming, presentations, and virtual events.
5. **Integration with Apps**: Supports bots, APIs, and integrations with services like Spotify and YouTube.
6. **Mobile and Desktop Support**: Available across major platforms for consistent user experiences.

### **Business Model**

Discord operates on a **freemium model**, offering core features for free while monetizing through **Nitro subscriptions**, which provide:

1. Enhanced customization (e.g., custom emojis, animated avatars).
2. Higher-quality streaming and larger file uploads.
3. Additional perks like custom server boosts.

### **Target Audience**

1. **Gamers:** The original user base, is still a major demographic.
2. **General Communities:** Groups focused on hobbies, fandoms, and professional interests.
3. **Educators and Students:** A growing audience for virtual learning and collaboration.
4. **Businesses:** Small to medium-sized teams using Discord as a cost-effective collaboration tool.

### **Company Culture**

* Known for its user-first approach, Discord prioritizes privacy, accessibility, and inclusivity.
* Actively promotes mental health and safe online spaces through initiatives and partnerships.

### 

### **Key Statistics (2023 Estimates)**

* **Users:** Over 300 million registered users, with 150 million monthly active users.
* **Servers:** Over 19 million active servers.
* **Revenue:** Estimated to surpass $200 million annually, largely driven by Nitro subscriptions.

### **Competitors**

* Slack: Focused on professional team collaboration.
* Microsoft Teams: A workplace communication tool.
* Reddit: A forum-based community platform.
* Zoom/Google Meet: Popular for video conferencing.

Discord’s emphasis on community, user engagement, and flexible communication has positioned it as a leader in the social and collaborative software market, with continued growth and diversification expected in the years ahead.

### **Case Study: Real-World Problems and Discord's Innovative Solutions**

Discord, a platform originally designed for gamers, has become a versatile communication tool that addresses several real-world problems. Here's a product dissection and an overview of the issues it solves.

### 

### **1. Remote Communication and Collaboration**

* **Problem:** Teams and communities need help to stay connected remotely.
* **Solution:** Discord provides seamless voice, video, and text communication tools. Businesses, educational groups, and project teams use it as an alternative to Slack or Zoom for real-time collaboration.

### **2. Community Building and Engagement**

* **Problem:** People seek platforms for interest-based communities and engagement.
* **Solution:** Discord allows users to create servers tailored to specific hobbies, fandoms, or professional interests, fostering vibrant, interactive communities.

### **3. Accessibility to Learning and Knowledge Sharing**

* **Problem:** Limited platforms for live, interactive learning.
* **Solution:** Discord supports virtual classrooms, workshops, and study groups through its text and voice channels, screen sharing, and bot integrations for quizzes or educational content.

### **4. Gaming and Content Sharing**

* **Problem:** Gamers need a reliable communication platform.
* **Solution:** Discord was initially built for gamers and still excels in providing low-latency voice chat and streaming, making it ideal for coordinating gameplay or hosting game nights.

### **5. Decentralized and Secure Community Management**

* **Problem:** Centralized platforms may have limited customization or privacy concerns.
* **Solution:** Discord empowers server owners with extensive moderation and privacy controls, allowing decentralized and self-managed communities.

### **6. Mental Health and Social Isolation**

* **Problem:** Social isolation and lack of mental health support are prevalent issues.
* **Solution:** Many servers on Discord focus on peer support, mental health discussions, and creating safe spaces for people to connect and share experiences.

### **7. Cost-Effective Communication for Small Groups**

* **Problem:** Expensive software solutions for small teams or groups.
* **Solution:** Discord’s free tier is feature-rich and sufficient for many small organizations or personal projects, making it a budget-friendly option.

#### **Conclusion:**

In conclusion, Discord has evolved from a niche platform for gamers into a versatile tool that addresses a wide array of real-world problems. Its robust features—ranging from seamless communication and community building to integration with third-party apps and customizable server management—enable it to cater to diverse audiences, including educators, businesses, hobbyists, and support groups.

By solving issues such as remote collaboration, social isolation, and cost-effective communication, Discord has established itself as a powerful platform for fostering connection and engagement in a digital world. Its accessibility, scalability, and user-centric design ensure its continued relevance and adaptability to emerging needs, making it a valuable asset in both personal and professional contexts.

### **Top Features of Discord:**

### **Core Features**

1. **Voice, Video, and Text Communication**
   * Provides high-quality voice, video, and text chat options.
   * Servers are organized into channels, making it easy to separate topics and activities.
2. **Server-Based Community Structure**
   * Users can create or join servers tailored to specific interests or communities.
   * Servers have customizable roles, permissions, and moderation tools.
3. **Screen Sharing and Streaming**
   * Enables screen sharing for presentations, gaming, or collaboration.
   * Users can stream games or other activities to groups.
4. **Integration with Third-Party Apps**
   * Connects with services like Spotify, YouTube, and Twitch.
   * Offers custom bots and APIs for automation and added functionalities.
5. **Mobile and Desktop Compatibility**
   * Cross-platform usability with apps for Windows, macOS, Linux, iOS, and Android.
6. **Freemium Model**
   * Core features are free, with Nitro subscriptions offering premium benefits like custom emojis, better-streaming quality, and larger file uploads.

### **Schema Description:**

The schema for Discord involves multiple entities that represent different aspects of the platform. These entities include User Entity, Server Entity, Channel Entity, Message Entity, Reaction Entity, Role Entity, Membership Entity and more. Each entity has specific attributes that describe its properties and relationships with other entities.

**User Entity:**

Represents a user account on Discord, including their username, email, password, and other personal details:

* **user\_id:** Unique identifier for the user.
* **username:** Display the name of the user.
* **email:** User's email address.
* **password\_hash:** Hashed password for security.
* **avatar\_url:** URL to the user's avatar image.
* **bio:** User's brief description.
* **created\_at:** Timestamp of user account creation.
* **last\_login:** Timestamp of the user's last login.
* **is\_verified:** Boolean flag indicating whether the user's email is verified.
* **is\_bot:** Boolean flag indicating whether the user is a bot account.

**Server Entity:**

Represents a server on Discord, including its name, owner, description, and member list:

* **server\_id:** Unique identifier for the server.
* **server\_name:** Display the name of the server.
* **owner\_id:** User ID of the server owner.
* **description:** Brief description of the server.
* **icon\_url:** URL to the server's icon image.
* **created\_at:** Timestamp of server creation.
* **member\_count:** Number of members in the server.
* **region:** Server's region (e.g., US West, EU West).

**Channel Entity:**

Represents a channel within a server, including its name, type (text or voice), and topic:

* **channel\_id:** Unique identifier for the channel.
* **server\_id:** Server ID to which the channel belongs.
* **channel\_name:** Display the name of the channel.
* **channel\_type:** Type of channel (text, voice, or category).
* **topic:** Topic or description of the channel.
* **position:** Position of the channel in the channel list.
* **parent\_id:** ID of the parent category channel (if applicable).

**Message Entity:**

Represents a message sent within a channel, including its content, sender, timestamp, and reactions:

* **message\_id:** Unique identifier for the message.
* **channel\_id:** Channel ID where the message was sent.
* **user\_id:** User ID of the message sender.
* **content:** Text content of the message.
* **timestamp:** Timestamp of the message.
* **attachments:** Array of attached files (URLs or file IDs).
* **embeds:** Array of embedded rich content (e.g., images, videos, links).

**Reaction Entity:**

Represents a reaction to a message, including the user who reacted and the emoji used:

* **reaction\_id:** Unique identifier for the reaction.
* **message\_id:** Message ID to which the reaction is attached.
* **user\_id:** User ID of the user who reacted.
* **emoji:** Emoji used for the reaction (Unicode or custom emoji ID).

**Role Entity:**

Represents a role within a server, including its name, permissions, and color:

* **role\_id:** Unique identifier for the role.
* **server\_id:** Server ID to which the role belongs.
* **role\_name:** Display the name of the role.
* **permissions:** Bitmask representing the role's permissions (e.g., view channels, send messages, manage roles).
* **color:** Color code for the role (hexadecimal).
* **position:** Position of the role in the role hierarchy.
* **is\_hoisted:** Boolean flag indicating whether the role is displayed separately in the member list.

**Membership Entity:**

Represents a user's membership in a server, including their joined date, roles, and nickname:

* **membership\_id:** Unique identifier for the membership.
* **user\_id:** User ID of the member.
* **server\_id:** Server ID to which the membership belongs.
* **joined\_at:** Timestamp when the user joins the server.
* **nick:** Nickname of the user in the server.
* **roles:** Array of role IDs assigned to the user.

### **Rationale Behind the Design**

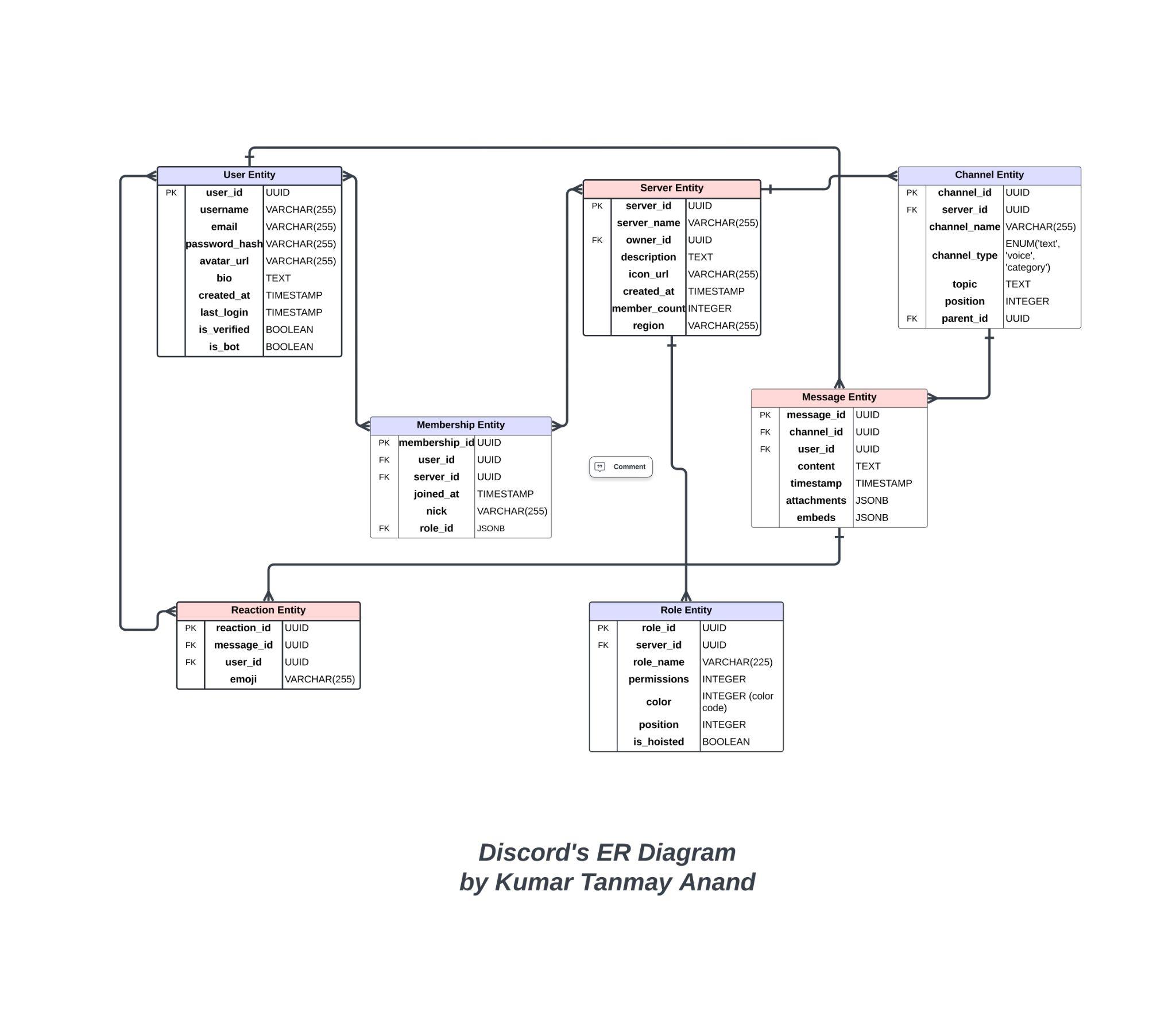
The schema design prioritizes user relationships, server structure, and message history. Key considerations include:

* **User-Server Relationships:** The **Membership** **entity** connects users to servers and their roles.
* **Channel Organization:** The **Channel entity** defines the structure of a server's communication channels.
* **Message History:** The **Message entity** stores the history of conversations within channels.
* **User Interactions:** The **Reaction entity** captures user reactions to messages.

## 

## **Understanding the ER Diagram for Discord**

**Entity-Relationship (ER) diagrams** are visual representations of data and the relationships between different data entities. In the context of Discord, the ER diagram helps us understand how different components of the platform, such as users, servers, channels, messages, and reactions, are interconnected.



**Made this ER diagram with the help of lucidchart**

### **Key Relationships:**

### **User-Server (Many-to-Many):** A user can be a member of many servers, and a server can have many members. This relationship is established through the **Membership** entity.

### **Server-Channel (One-to-Many):** A server can have many channels, but a channel belongs to only one server.

### **Channel-Message (One-to-Many):** A channel can have many messages, but a message belongs to only one channel.

### **User-Message (One-to-Many):** A user can send many messages, but a message is sent by only one user.

### **Message-Reaction (One-to-Many):** A message can have many reactions, but a reaction is associated with only one message.

### **User-Reaction (Many-to-Many):** A user can react to many messages, and a reaction can be associated with many users.

### **Server-Role (One-to-Many):** A server can have many roles, but a role belongs to only one server.

### 

### **Conclusion**

In conclusion, Discord’s journey from a gaming-focused communication platform to a versatile tool for communities, educators, businesses, and social groups exemplifies its adaptability and innovation. Discord has established itself as a leader in communication and social engagement by addressing key real-world problems like remote collaboration, community building, and cost-effective communication.

The platform’s user-centric design, freemium business model, and focus on inclusivity and privacy have played a critical role in its widespread adoption. With features like customizable servers, seamless voice and video communication, and integration with third-party tools, Discord continues to cater to a diverse and expanding audience.

As it evolves, Discord's ability to balance its core identity with the needs of broader user groups positions it well for sustained growth. Its success highlights the importance of flexibility, community focus, and innovation in meeting the challenges of a digitally connected world.