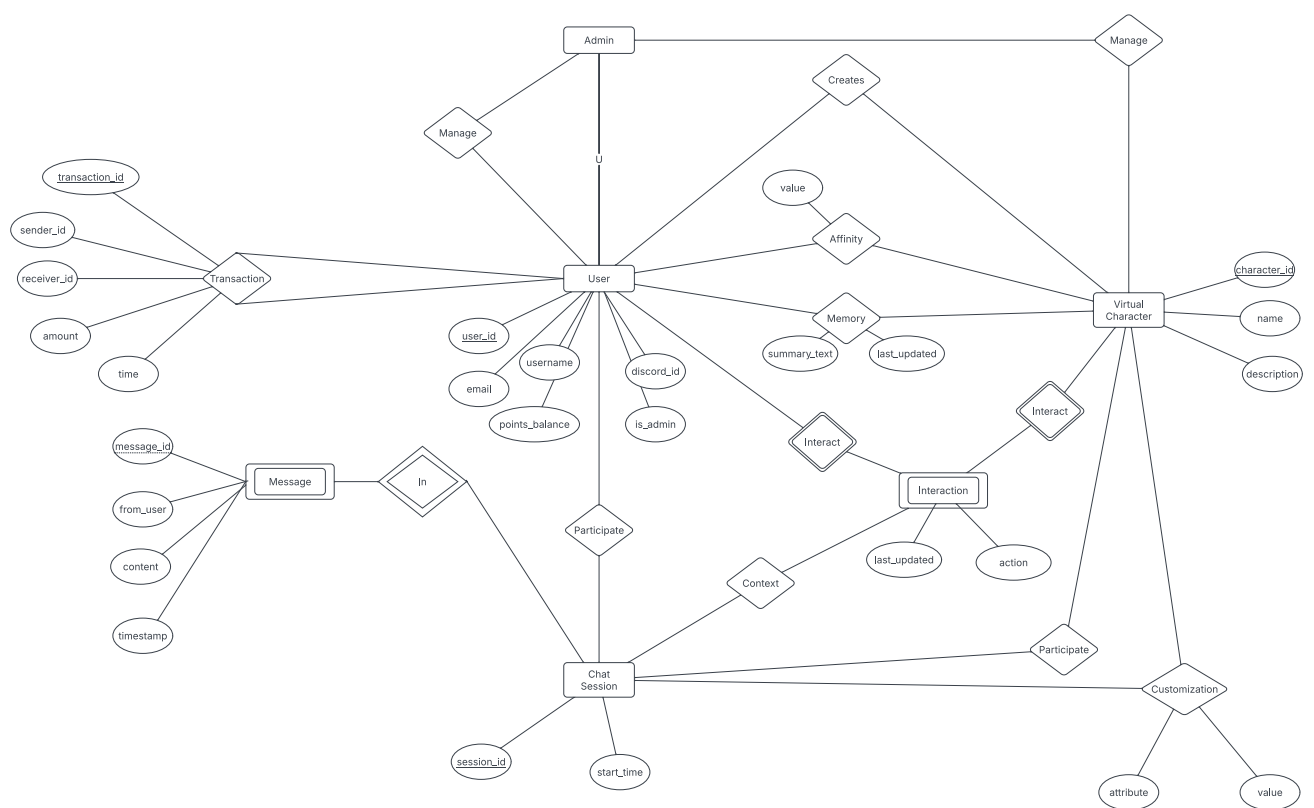


## Project Phase 2 : Group 8

Chengxu Lan, Yulong Cao, Rohan Verma, Harshal Jorwekar

### Step 5: EER modeling



## Step 6: Creating Relations

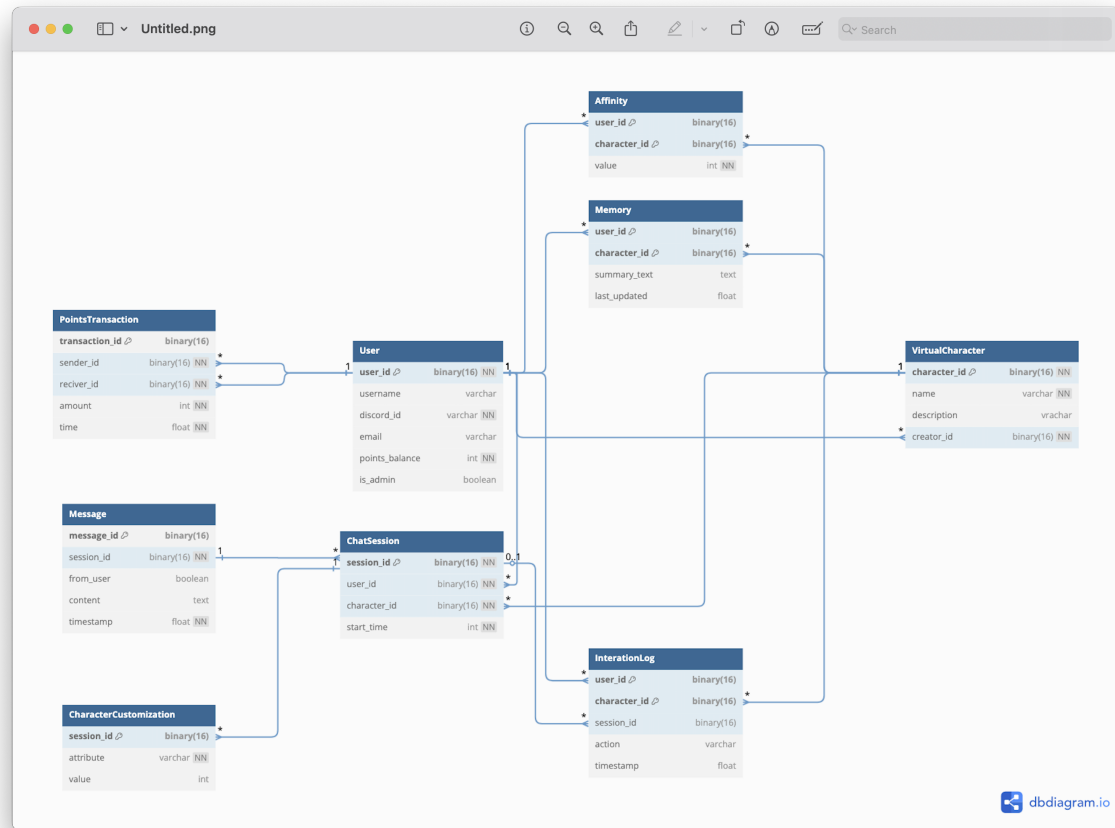


Figure 1: ER Diagram

## Relational Schema

### User

(user\_id, username, email, points\_balance, discord\_id, is\_admin)

### Virtual\_Character

(character\_id, name, description)

### Transaction

(transaction\_id, sender\_id, receiver\_id, amount, time)

FOREIGN KEY (sender\_id) REFERENCES User(user\_id)

FOREIGN KEY (receiver\_id) REFERENCES User(user\_id)

### Message

(message\_id, from\_user, content, timestamp)

FOREIGN KEY (from\_user) REFERENCES User(user\_id)

### Chat\_Session

(session\_id, start\_time)

### Interaction

(user\_id, character\_id, timestamp, action, context)

FOREIGN KEY (user\_id) REFERENCES User(user\_id)

FOREIGN KEY (character\_id) REFERENCES Virtual\_Character(character\_id)

### Memory

(user\_id, character\_id, summary\_text, last\_updated)

FOREIGN KEY (user\_id) REFERENCES User(user\_id)

FOREIGN KEY (character\_id) REFERENCES Virtual\_Character(character\_id)

### Customization

(character\_id, attribute, value)

FOREIGN KEY (character\_id) REFERENCES Virtual\_Character(character\_id)

### Affinity

(user\_id, character\_id, value)

FOREIGN KEY (user\_id) REFERENCES User(user\_id)

FOREIGN KEY (character\_id) REFERENCES Virtual\_Character(character\_id)

## Candidate Keys

- **User:** {user\_id}, {email}, {discord\_id}
- **Virtual\_Character:** {character\_id}, {name}
- **Transaction:** {transaction\_id}
- **Message:** {message\_id}
- **Chat\_Session:** {session\_id}
- **Interaction:** {user\_id, character\_id, timestamp}
- **Memory:** {user\_id, character\_id}
- **Customization:** {character\_id, attribute}
- **Affinity:** {user\_id, character\_id}

## Functional Dependencies

- **User:**
  - $\text{user\_id} \rightarrow \{\text{username}, \text{email}, \text{points\_balance}, \text{discord\_id}, \text{is\_admin}\}$
  - $\text{email} \rightarrow \{\text{user\_id}\}$
  - $\text{discord\_id} \rightarrow \{\text{user\_id}\}$
- **Virtual\_Character:**
  - $\text{character\_id} \rightarrow \{\text{name}, \text{description}\}$
  - $\text{name} \rightarrow \{\text{character\_id}\}$
- **Transaction:**
  - $\text{transaction\_id} \rightarrow \{\text{sender\_id}, \text{receiver\_id}, \text{amount}, \text{time}\}$
- **Message:**
  - $\text{message\_id} \rightarrow \{\text{from\_user}, \text{content}, \text{timestamp}\}$
- **Chat\_Session:**
  - $\text{session\_id} \rightarrow \{\text{start\_time}\}$
- **Interaction:**
  - $(\text{user\_id}, \text{character\_id}, \text{timestamp}) \rightarrow \{\text{action}, \text{context}\}$
- **Memory:**
  - $(\text{user\_id}, \text{character\_id}) \rightarrow \{\text{summary\_text}, \text{last\_updated}\}$
- **Customization:**
  - $(\text{character\_id}, \text{attribute}) \rightarrow \{\text{value}\}$
- **Affinity:**
  - $(\text{user\_id}, \text{character\_id}) \rightarrow \{\text{value}\}$

## Step 7: Relational Algebra

**Query 1: Find all users who have interacted with a specific character**

$$\pi_{\text{username}}(\sigma_{\text{character\_id}='X'}(\text{User} \bowtie \text{Interaction}))$$

**Query 2: Get all transactions for a specific user**

$$\pi_{\text{transaction\_id}, \text{amount}, \text{time}}(\sigma_{\text{sender\_id}='Y' \vee \text{receiver\_id}='Y'}(\text{Transaction}))$$

**Query 3: Find characters with high affinity scores**

$$\pi_{\text{name}, \text{value}}(\sigma_{\text{value}>0.8}(\text{Virtual\_Character} \bowtie \text{Affinity}))$$

**Query 4: Get recent chat sessions with messages**

$$\pi_{\text{session\_id}, \text{content}}(\sigma_{\text{timestamp}>'\text{current\_time} - 24\text{h}'}(\text{Chat\_Session} \bowtie \text{Message}))$$

**Query 5: Find customizations for a specific character**

$$\pi_{\text{attribute}, \text{value}}(\sigma_{\text{character\_id}='Z'}(\text{Customization}))$$

**Query 6: Find users with the highest points balance**

$$\pi_{\text{username, points.balance}}(\sigma_{\text{points.balance} \geq \text{ALL}(\pi_{\text{points.balance}}(\text{User}))}(\text{User}))$$

**Query 7: Get the most recent interaction for each user-character pair**

$$\begin{aligned} & \pi_{\text{user\_id, character\_id, action}}( \\ & \quad \sigma_{\text{i1.timestamp} = \text{max\_timestamp}}( \\ & \quad \quad \text{Interaction i1} \bowtie \\ & \quad \quad (\pi_{\text{user\_id, character\_id, max(timestamp) as max\_timestamp}}(\text{Interaction}) \text{ i2}) \\ & \quad ) \\ & ) \end{aligned}$$

**Query 8: Find users who have interacted with all characters**

$$\text{User} \div \pi_{\text{character\_id}}(\text{Virtual\_Character})$$

**Query 9: Get messages from users with high affinity characters**

$$\pi_{\text{content}}(\sigma_{\text{value} > 0.9}(\text{Message} \bowtie \text{User} \bowtie \text{Affinity}))$$

**Query 10: Find users who have never participated in a chat session**

$$\pi_{\text{username}}(\text{User}) - \pi_{\text{username}}(\text{User} \bowtie \text{Chat\_Session})$$

**Query 11: Get characters with complete customization profiles**

$$\begin{aligned} & \pi_{\text{name}}(\text{Virtual\_Character} \bowtie \\ & \quad (\pi_{\text{character\_id}}( \\ & \quad \quad \sigma_{\text{count} = n}( \\ & \quad \quad \quad \rho_{(\text{count})}(\mathcal{F}_{\text{COUNT}(* )}(\gamma_{\text{character\_id}}(\text{Customization}))) \times \\ & \quad \quad \quad \rho_{(n)}(\pi_{\text{count}}(\rho_{(\text{count})}(\mathcal{F}_{\text{COUNT}(\text{DISTINCT attribute})}(\text{Customization})))) \\ & \quad ) \\ & )) \end{aligned}$$

**Query 12: Find users who have made transactions but never received any**

$$\pi_{\text{username}}(\sigma_{\text{user\_id} \in (\pi_{\text{sender\_id}}(\text{Transaction})) \wedge \text{user\_id} \notin (\pi_{\text{receiver\_id}}(\text{Transaction}))}(\text{User}))$$