erDiagram  
 USER {  
 string id PK  
 string username  
 string email  
 string password\_hash  
 date created\_at  
 date last\_login  
 json preferences  
 int streak\_days  
 int total\_points  
 }  
   
 MOOD\_ENTRY {  
 string id PK  
 string user\_id FK  
 date entry\_date  
 time entry\_time  
 int mood\_rating  
 string notes  
 json location\_data  
 json weather\_data  
 }  
   
 ACTIVITY {  
 string id PK  
 string name  
 string icon  
 string category  
 }  
   
 MOOD\_ACTIVITY {  
 string mood\_entry\_id FK  
 string activity\_id FK  
 int impact\_rating  
 }  
   
 ACHIEVEMENT {  
 string id PK  
 string name  
 string description  
 int points\_value  
 string badge\_icon  
 string criteria\_type  
 json criteria\_value  
 }  
   
 USER\_ACHIEVEMENT {  
 string user\_id FK  
 string achievement\_id FK  
 date earned\_date  
 boolean is\_viewed  
 }  
   
 THEME {  
 string id PK  
 string name  
 string description  
 json color\_scheme  
 boolean is\_system  
 }  
   
 USER\_THEME {  
 string user\_id FK  
 string theme\_id FK  
 boolean is\_active  
 date last\_used  
 }  
   
 REMINDER {  
 string id PK  
 string user\_id FK  
 time reminder\_time  
 json active\_days  
 boolean is\_enabled  
 string message  
 }  
   
 USER ||--o{ MOOD\_ENTRY : "tracks"  
 USER ||--o{ REMINDER : "sets"  
 USER ||--o{ USER\_ACHIEVEMENT : "earns"  
 USER ||--o{ USER\_THEME : "customizes"  
   
 MOOD\_ENTRY ||--o{ MOOD\_ACTIVITY : "includes"  
 ACTIVITY ||--o{ MOOD\_ACTIVITY : "categorizes"  
   
 ACHIEVEMENT ||--o{ USER\_ACHIEVEMENT : "awarded as"  
 THEME ||--o{ USER\_THEME : "selected as"

## Figure 4.2: Entity Relationship Diagram

The Entity Relationship Diagram (ERD) illustrates the database structure for the Pro Mood Tracker application, showing the relationships between different entities that store and organize application data.

### Primary Entities

1. **USER**:
   * Central entity storing user authentication and profile information
   * Tracks user engagement metrics like streak days and points
   * Stores user preferences as a JSON object for flexibility
2. **MOOD\_ENTRY**:
   * Core data entity capturing mood tracking information
   * Linked to users through a foreign key relationship
   * Stores contextual information like location and weather data
   * Contains the primary mood rating and optional notes
3. **ACTIVITY**:
   * Reference table for predefined and user-created activities
   * Categorized to allow for organizational grouping
   * Includes visual representation (icon) for UI display
4. **ACHIEVEMENT**:
   * Defines gamification elements to encourage app engagement
   * Contains criteria for earning achievements stored as flexible JSON
   * Associates point values and visual badges with accomplishments
5. **THEME**:
   * Enables UI customization through predefined and custom themes
   * Stores color schemes as JSON for flexible styling options
   * Distinguishes between system defaults and user-created themes

### Junction Tables

1. **MOOD\_ACTIVITY**:
   * Links mood entries with associated activities
   * Allows tracking impact rating of activities on mood
   * Enables many-to-many relationship between entries and activities
2. **USER\_ACHIEVEMENT**:
   * Tracks which achievements have been earned by which users
   * Records when achievements were earned
   * Tracks viewing status for notification purposes
3. **USER\_THEME**:
   * Manages theme preferences for each user
   * Tracks active theme and usage history

### Key Relationships

* Each user can have multiple mood entries, reminders, achievements, and themes
* Mood entries can be associated with multiple activities through the junction table
* Achievements and themes are reusable across users with personalized tracking
* All entity relationships maintain referential integrity through foreign key constraints

This database design supports the core functionality of mood tracking while enabling engagement features like achievements and customization options that enhance the user experience.