Samuel Grzelak
SWDV 691
Database Re-Design
July 23, 2023

What changes were made in the re-design process?

- I utilized a free program to create a cleaner, more detailed ERD.
- Some table names were changed or dropped and combined with others.
    - The rewards table was dropped, but rewards are now included in the goals table and associated with each unique goal.
    - client_relations table stores all client and mentor relationships.

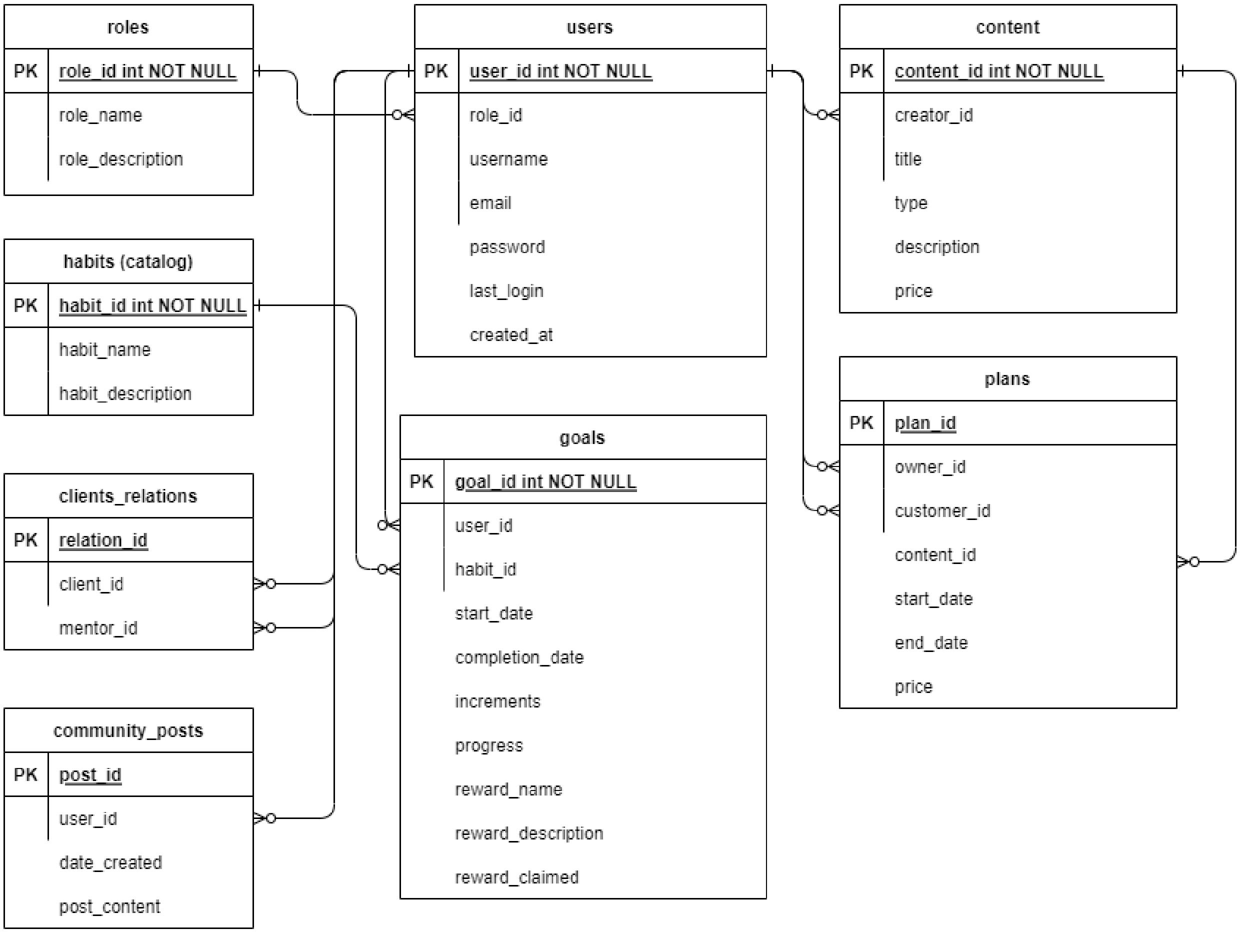
Database Technology Choice: MySQL

Explanation: For the Healthy Habits web application, MySQL is chosen as the appropriate database technology. The justification for my choice is as follows:

1. Relational Database Management System: MySQL is a widely used and mature relational database management system (RDBMS) that offers excellent performance and scalability. It provides a robust set of features for managing structured data, making it suitable for handling the relational nature of the application's data.

2. SQL Compatibility: MySQL is fully compatible with SQL, which is a widely adopted and standardized language for querying and manipulating relational databases. This compatibility ensures that the application can leverage the power of SQL for efficient data retrieval and manipulation.

3. Community Support and Documentation: MySQL has a large and active community of developers and users, providing ample resources, documentation, and support. This availability of community support makes it easier to troubleshoot issues, optimize performance, and stay up to date with best practices.



roles

PK role_id int NOT NULL

role_name

role_description

habits (catalog)

PK habit_id int NOT NULL

habit_name

habit_description

clients_relations

PK relation_id

client_id

mentor_id

community_posts

PK post_id

user_id

date_created

post_content

users

PK user_id int NOT NULL

role_id

username

email

password

last_login

created_at

goals

PK goal_id int NOT NULL

user_id

habit_id

start_date

completion_date

increments

progress

reward_name

reward_description

reward_claimed

content

PK content_id int NOT NULL

creator_id

title

type

description

price

plans

PK plan_id

owner_id

customer_id

content_id

start_date

end_date

price

users Table:

- Columns: user_id (PRIMARY KEY), role_id (FOREIGN KEY), username, email, password, last_login, created_date
- Data Structure:
  - user_id: Integer or UUID to uniquely identify each user.
  - role_id: Integer referencing the role_id from the roles table.
  - username: String to store the username of the user.
  - email: String to store the email address of the user.
  - password: String to store the hashed password of the user.
  - last_login: DATE to store the last time a user logged in.
  - created_date: DATE storing the time of account creation.

Explanation: The users table is modified to include a role_id column, which references the roles table to identify the role of each user. This modification enables assigning roles to users and determining their permissions and access levels within the application.

roles Table:

- Columns: role_id (PRIMARY KEY), role_name
- Data Structure:
  - role_id: Integer or UUID to uniquely identify each role.
  - role_name: String to store the name of the role.

Explanation: The roles table is added to store information about different user roles available in the application. Each role has a unique role_id, and the role_name field holds the name of the role (e.g., Trainer, Nutritionist, Healthy Professional).

content Table:

- Columns: content_id (PRIMARY KEY), creator_id (FOREIGN KEY), title, description, content_type, price
- Data Structure:
    - content_id: Integer or UUID to uniquely identify each content item.
    - creator_id: Integer or UUID referencing the user_id from the users table.
    - title: String to store the title or name of the content item.
    - description: Text or varchar to provide a description of the content.
    - content_type: String to specify the type of content (e.g., workout plan, diet plan, recipe).
    - price: Numeric or decimal to store the price of the content item.

Explanation: The content table is created to allow Trainers/Nutritionists/Healthy Professionals to create and manage their content items. Each content item has a unique content_id. The user_id field establishes a relationship with the users table, indicating which user created the content. The title and description fields provide information about the content item. The content_type field helps categorize the type of content (e.g., workout plan, diet plan, recipe). The price field stores the price of the content item.

plans Table:

- Columns: purchase_id (PRIMARY KEY), user_id (FOREIGN KEY), content_id (FOREIGN KEY), purchase_date
- Data Structure:
  - plan_id: Integer or UUID to uniquely identify each purchase.
  - owner_id: Integer or UUID referencing the user_id from the users table.
  - customer_id: Integer or UUID referencing the user_id from the users table.
  - content_id: Integer or UUID referencing the content_id from the content table.
  - start_date: Date or datetime to store the start date of the plan (date of purchase).
  - end_date: Date or datetime to store the end date of the goal.

Explanation: The purchases table is added to track user purchases of content items. Each purchase has a unique purchase_id. The user_id field establishes a relationship with the users table, indicating which user made the purchase. The content_id field establishes a relationship with the content table, indicating which content item was purchased. The purchase_date field stores the timestamp of the purchase.

With these modifications and new tables, the Healthy Habits application allows users with the Trainer/Nutritionist/Healthy Professional role to create and monetize their content. They can create various types of content items (workout plans, diet plans, recipes) and set prices for them. Users can then make purchases either through subscription plans or one-time purchases, and the purchases table tracks their transactions.

client_relations Table:

- Columns: relation_id (PRIMARY KEY), client_id (FOREIGN KEY) mentor_id (FOREIGN KEY)
- Data Structure:
  - relation_id: Integer or UUID to uniquely identify each client/mentor relation.
  - client_id: Integer or UUID referencing the user_id from the users table for the assigned client.
  - mentor_id: Integer or UUID referencing the user_id from the users table for the assigned trainer.

Explanation: The clients table is created to store information about the clients assigned to a Trainer/Nutritionist/Healthy Professional. Each relationship has a unique relation_id. The client_id and mentor_id field establishes a relationship with the users table, indicating the assigned trainer for the client.

habits Table:

- Columns: habit_id (PRIMARY KEY), habit_name, habit_description
- Data Structure:
  - habit_id: Integer or UUID referencing the habit_id from the habits table.
  - habit_name: String to store the name or title of the habit.
  - habit_description: Text or varchar to provide additional details or description of the habit.

Explanation: The habits table acts as the catalog of previously created habits and goals. Users can choose to use an already created habit or create a new one which will be stored in the habits table for future use by other community members.

goals Table:

- Columns: goal_id (PRIMARY KEY), client_id (FOREIGN KEY), habit_id (FOREIGN KEY), start_date, completion_date, increments, progress, reward_name, reward_description, reward_claimed
- Data Structure:
    - goal_id: Integer or UUID to uniquely identify each client's goal.
    - user_id: Integer or UUID referencing the user_id from the users table.
    - habit_id: Integer or UUID referencing the habit_id from the habits table.
    - start_date: Date or datetime to store the start date of the goal.
    - end_date: Date or datetime to store the end date of the goal.
    - increments: Numeric or decimal to store the number of times the user should repeat the habit to complete the goal.
    - progress: Numeric or decimal value (out of 100) incremented each time a user repeats the habit until completion.
    - reward_name: String to store the name or title of the reward.
    - reward_description: Text or varchar to provide additional details or description of the reward.
    - reward_claime: Integer (0 or 1) to indicate whether reward was claimed or not after completion of goal.

Explanation: The client_goals table is added to track the goals set by clients for specific habits. Each goal has a unique goal_id. The client_id field establishes a relationship with the clients table, indicating which client set the goal. The habit_id field references the habit_id from the habits table to link the goal to a specific habit. The goal_value field stores the value or target set by the client for the habit.

community_posts Table:

- Columns: post_id (PRIMARY KEY), user_id (FOREIGN KEY), date created, post_content
- Data Structure:
    - post_id: Integer or UUID to uniquely identify each post.
    - user_id: Integer or UUID referencing the user_id from the users table.
    - date_created: Date or datetime to store the creation date of the post.
    - post_content: varchar (###) containing string element of content written by the poster.

Explanation: The community_posts table will contain all posts from user accounts. These posts will appear on user profile feeds. The content of these posts will range from status updates, progress on goals, and goal completions / rewards that users can choose to share along their journey. The community feed will pull from the community_posts for users to see what their mentors are posting about and any new content available for them to view / purchase.

With these tables, the assigned Trainer/Nutritionist/Healthy Professional users can perform the following functions:

1. Invite Clients: They can invite clients to join the application by creating client accounts and creating a relationship stored in the client_relations table.
2. Manage Client Goals: They can set and track goals for each client by creating entries in the goals table, linking the client, habit, and goal value.
3. Manage Client Rewards: They can assign rewards to clients for achieving their goals by adding a reward upon goal creation in the goals table, associating the client and reward details. They can also check if the client has claimed the reward once they've reached their goal (ie. "cheat day").
4. Manage Client Plans: They can enroll clients in specific plans or content items by adding entries in the client_plans table, specifying the client, content, and plan duration. Client will review the created plans to confirm the amount and payment if applicable.

These tables enable the Trainer/Nutritionist/Healthy Professional users to have dedicated control and tracking over their clients' goals, rewards, and plans within the Healthy Habits application.