

Exercise Recommendation System

Naga Vara Pradeep Yendluri

Tharun Gunishetti

Siva Kumar Chaini

Sai Deekshitha Kuchuru

Lakshmi Deepika Pagadala

Department of Computer Science and Engineering, University of North Texas

INFO 5707.001: Data Modeling for Information Professionals

Dr. Tozammel Hossain, PhD

April 18, 2024

Database Overview – Objectives and Scope

The objectives and scope of the Exercise Recommendation System's database are designed to support the overarching goal of providing personalized, engaging, and effective fitness recommendations to users.

Objectives of the Database System:

Personalization: To store comprehensive user profiles that include not only basic demographic information but also detailed health parameters (e.g., weight, medical conditions, fitness levels) to enable personalized exercise recommendations.

Adaptability: To dynamically adjust exercise recommendations based on user feedback, progress, and changes in health parameters, ensuring the recommendations remain relevant and effective over time.

Support for Machine Learning: To provide a structured and accessible data repository that can be used by various machine learning algorithms to analyze user data, identify patterns, and generate intelligent exercise recommendations.

Data Security and Privacy: To ensure the highest standards of data security and privacy protection for users' personal and health information, adhering to legal and ethical guidelines.

Scope of the Database:

User Data Management: The database will store and manage extensive user data, including personal details, fitness goals, health parameters, and medical conditions, allowing for a highly personalized user experience.

Exercise Repository: It will contain a comprehensive catalog of exercises, including descriptions, targeted muscle groups, and difficulty levels, which can be used to assemble personalized workout plans.

Workout Plans and Recommendations: The system will generate and store customized workout plans and exercise recommendations for each user, based on machine learning algorithms and user-specific data.

Progress Tracking: The database will record user progress over time, including completed exercises, improvements in fitness levels, and feedback on the exercise regimen, enabling continuous refinement of recommendations.

Machine Learning Integration: It will support the integration of machine learning models by storing data needed for training algorithms (e.g., user responses, effectiveness of recommendations) and by applying these models to generate new recommendations.

User Requirements

Storing Detailed User Profiles: The database will hold comprehensive data on each user, including personal information, fitness goals, health parameters (e.g., weight, age, medical conditions), and exercise preferences.

Exercise and Workout Plan Records: A catalog of exercises, including detailed attributes like type, targeted muscle groups, and difficulty level, alongside user-specific workout plans, will be maintained, enabling easy generation and adjustment of personalized plans.

Health Parameters Record: Capture static health parameters at the time of user registration or update, such as weight, height, and known medical conditions. These parameters are used to tailor exercise recommendations.

User Preferences: Record user exercise preferences, such as favorite types of workouts (e.g., yoga, running) and available equipment, to personalize exercise recommendations without the need for complex algorithms.

Exercise Plans: Store exercise plans linked to user profiles, consisting of a set of exercises selected based on the user's health parameters and preferences. Plans can be manually adjusted by users or administrators.

Progress Logs: Keep logs of exercises completed by the user, including the date and any user feedback on the exercise difficulty and enjoyment. This data helps track progress without requiring real-time analysis.

Dietary Information: Optionally, store basic dietary preferences or restrictions to complement exercise recommendations with general health tips. This feature does not include dynamic meal planning but can offer static content tailored to broad user groups.

Static Health Content: Include a repository of health and fitness articles, tips, and guidelines that users can access for information. This content is not personalized but selected to cover a wide range of interests and needs.

Administrative Controls: Provide functionalities for system administrators to update exercise catalogs, health content, and manage user accounts. This allows for system maintenance and content refresh without involving complex data operations.

User Goals Tracking: Implement a feature to store user-set fitness goals, such as weight loss targets, running distances, or strength achievements.

Community Feature Data: Create a basic structure to support a community feature where users can share achievements, such as completing a fitness challenge or reaching a personal goal.

Business Rules

Business rules define the constraints and guidelines under which the Exercise Recommendation System operates. These rules are critical for maintaining data integrity, ensuring user privacy, and guiding system interactions.

Unique User Identification: Each user must have a unique identifier (UserID) to ensure personal data is accurately associated with the correct individual and to prevent unauthorized access.

Data Validation for Health Parameters: Health parameters (e.g., weight, height, age) must be validated for format and range to ensure the integrity of the data used in exercise recommendations.

Privacy Compliance: All user data, especially sensitive health information, must be stored and processed in compliance with relevant data protection regulations (e.g., GDPR, HIPAA) to ensure user privacy and system security.

Exercise Recommendation Limits: Exercise recommendations should be aligned with recognized fitness and health guidelines to prevent suggesting workouts that could be harmful based on a user's health profile and fitness level.

User Feedback Incorporation: User feedback on exercises and workout plans should be periodically reviewed and incorporated into the system to improve exercise recommendations and user satisfaction.

Access Control: Users must only have access to their personal information and exercise plans. Administrative access is required for any modifications outside of standard user functionalities.

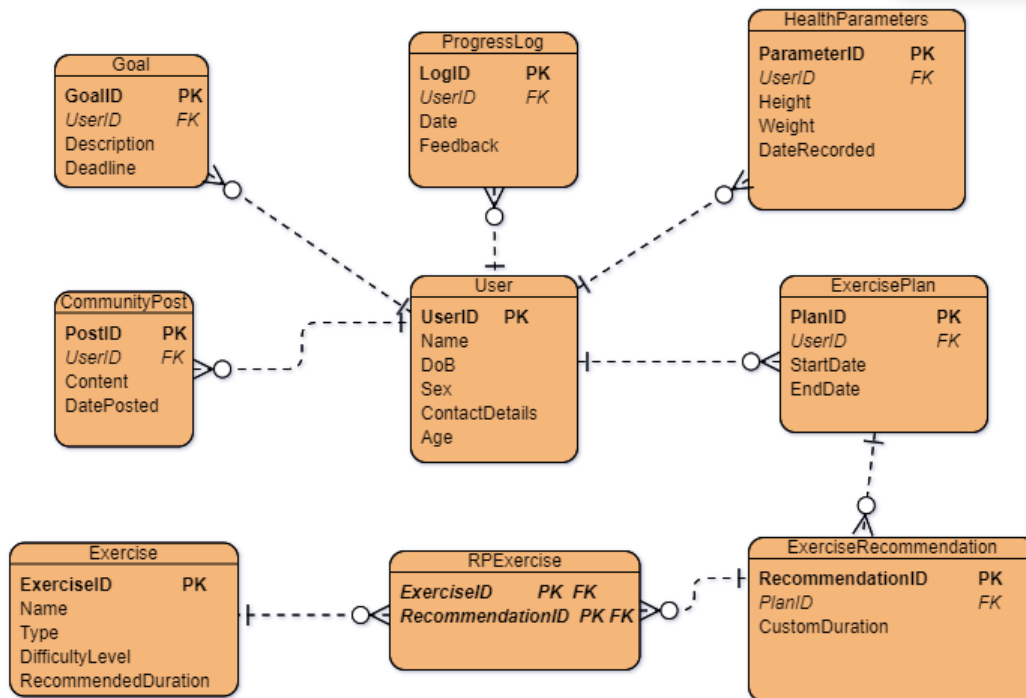
Progress Tracking Updates: Users must log their exercise progress to keep their profiles updated and ensure the relevance of future exercise recommendations. This tracking is based on user input and does not require real-time monitoring.

Exercise Database Integrity: The exercise database must be maintained with accurate descriptions, difficulty levels, and classifications. Any additions or changes to the exercise database require validation for safety and effectiveness based on established fitness standards.

Goal Setting Guidelines: Users can set personal fitness goals within the system, but these goals must comply with safe and realistic guidelines to avoid promoting unhealthy or unattainable targets.

Notification Preferences: Users have the right to set and modify their preferences for receiving notifications from the system to avoid unwanted communication and respect user privacy.

Entity Relationship Diagram



Exercise Recommendation System

Representations in Visual-paradigm.com
 Bold: Primary Key
 Italic: Foreign Key

In RPEExercise Table both ExerciseID, RecommendationID combined act as primary key

Data Dictionary

| Table Name | Attribute Name | Description | Data Type | Data Format | Required | PK or FK | Example | |
|------------------------|---------------------|--|--------------|---------------|----------|----------|---------------------------------------|------------|
| User | UserID | Unique identifier for each user | INTEGER | 123 | Y | PK | | 152 |
| | Name | User's full name | VARCHAR(255) | xxxx xxxxx | Y | | SMITH | |
| | Age | User's age | INTEGER | 23 | | | | 12 |
| | DoB | User's Date of Birth | DATE | 08-06-1995 | | | | 04-05-1999 |
| | ContactDetails | Users contact details | VARCHAR(255) | xxx xxxxx | | | 940-950-4590 | |
| Goal | GoalID | Goal ID | INTEGER | 156 | Y | PK | | 12 |
| | UserID | User ID to which goal is created | INTEGER | 154 | Y | FK | | 14 |
| | Description | description about goal | VARCHAR(255) | xxx sxxxxsxxx | | | Complete 3 laps | |
| | Deadline | deadline to complete goal | DATE | 08-06-1999 | Y | | | 05-03-1999 |
| CommunityPost | PostID | post id | INTEGER | 12 | Y | PK | | 23 |
| | UserID | User id who posted | INTEGER | 45 | Y | FK | | 34 |
| | Content | content of post | VARCHAR(255) | xxxxxxxxxx | | | Completed exercise plan for this week | |
| | PostDate | post date | DATE | 05-08-1999 | Y | | | 05-04-1999 |
| ProgressLog | LogID | log id | INTEGER | 56 | Y | PK | | 12 |
| | UserID | user id who logged | INTEGER | 54 | Y | FK | | 12 |
| | Date | date of logging | DATE | 05-05-1999 | Y | | | 05-04-1999 |
| | Feedback | feedback of exercise plan | VARCHAR(255) | xxxxxxxxxxxxx | | | Good Exercise plan | |
| HealthParameters | ParameterID | parameter id to store user's health parameters | INTEGER | 59 | Y | PK | | 12 |
| | UserID | user id | INTEGER | 45 | Y | FK | | 12 |
| | Height | user's height | DECIMAL | 45.55 | | | | 45.55 |
| | Weight | user's weight | DECIMAL | 59.69 | | | | 59.69 |
| | Date | date of record | DATE | 05-05-1999 | Y | | | 05-04-1999 |
| ExercisePlan | PlanID | plan id | INTEGER | 45 | Y | PK | | 12 |
| | UserID | user id to which plan assigned | INTEGER | 84 | Y | FK | | 12 |
| | StartDate | start date | DATE | 09-08-1999 | | | | 05-04-1999 |
| | EndDate | end date | DATE | 07-08-1960 | | | | 05-04-1999 |
| ExerciseRecommendation | RecommendationID | recommendation id | INTEGER | 45 | Y | PK | | 12 |
| | PlanID | plan id | INTEGER | 45 | Y | FK | | 12 |
| | CustomDuration | duration recommended | INTEGER | 89 | Y | | | 123 |
| Exercise | ExerciseID | exercise id | INTEGER | 78 | Y | PK | | 12 |
| | Name | exercise name | VARCHAR(255) | xxxxxxxxxxxxx | Y | | Running | |
| | Type | exercise type | VARCHAR(255) | xxxxxxxx | Y | | Cardiac | |
| | DifficultyLevel | difficulty level | VARCHAR(255) | xxxxxxxx | | | Easy | |
| | RecommendedDuration | recommended duration | INTEGER | 59 | Y | | | 15 |
| RPExercise | ExerciseID | exercise id | INTEGER | 78 | Y | PK | | 12 |
| | RecommendationID | recommendation id | INTEGER | 45 | Y | PK | | 12 |

Entity Generation and Data Entry

Create Statements:

```

root@racknerd-e93057:~# sqlite3 --version
3.22.0 2018-01-22 18:45:57 0c55d179733b46d8d0ba4d88e01a25e3
root@racknerd-e93057:~# sqlite exercise.db
-bash: sqlite: command not found
root@racknerd-e93057:~# sqlite3 exercise.db
SQLite version 3.22.0 2018-01-22 18:45:57
Enter ".help" for usage hints.
sqlite> -- Create the User table
sqlite> CREATE TABLE User (
...>   UserID INTEGER PRIMARY KEY,
...>   Name TEXT NOT NULL,
...>   Age INTEGER,
...>   DoB DATE,
...>   Sex TEXT,
...>   ContactDetails TEXT
...> );
sqlite>
sqlite> -- Create the Exercise table
sqlite> CREATE TABLE Exercise (
...>   ExerciseID INTEGER PRIMARY KEY,
...>   Name TEXT,
...>   Type TEXT,
...>   DifficultyLevel INTEGER,
...>   RecommendedDuration INTEGER
...> );
sqlite>
sqlite> -- Create the Goal table
sqlite> CREATE TABLE Goal (
...>   GoalID INTEGER PRIMARY KEY,
...>   UserID INTEGER,
...>   Description TEXT,
...>   Deadline DATE,
...>   FOREIGN KEY (UserID) REFERENCES User(UserID)
...> );
sqlite>

```

```

sqlite> -- Create the HealthParameters table
sqlite> CREATE TABLE HealthParameters (
...>   ParameterID INTEGER PRIMARY KEY,
...>   UserID INTEGER,
...>   Height REAL,
...>   Weight REAL,
...>   DateRecorded DATE,
...>   FOREIGN KEY (UserID) REFERENCES User(UserID)
...> );
sqlite>
sqlite> -- Create the ProgressLog table
sqlite> CREATE TABLE ProgressLog (
...>   LogID INTEGER PRIMARY KEY,
...>   UserID INTEGER,
...>   Date DATE,
...>   Feedback TEXT,
...>   FOREIGN KEY (UserID) REFERENCES User(UserID)
...> );
sqlite>
sqlite> -- Create the CommunityPost table
sqlite> CREATE TABLE CommunityPost (
...>   PostID INTEGER PRIMARY KEY,
...>   UserID INTEGER,
...>   Content TEXT,
...>   DatePosted DATE,
...>   FOREIGN KEY (UserID) REFERENCES User(UserID)
...> );
sqlite>
sqlite> -- Create the ExercisePlan table
sqlite> CREATE TABLE ExercisePlan (
...>   PlanID INTEGER PRIMARY KEY,
...>   UserID INTEGER,
...>   StartDate DATE,
...>   EndDate DATE,
...>   FOREIGN KEY (UserID) REFERENCES User(UserID)
...> );

```

```

sqlite> -- Create the ExerciseRecommendation table
sqlite> CREATE TABLE ExerciseRecommendation (
...>   RecommendationID INTEGER PRIMARY KEY,
...>   PlanID INTEGER,
...>   CustomDuration INTEGER,
...>   FOREIGN KEY (PlanID) REFERENCES ExercisePlan(PlanID)
...> );
sqlite>
sqlite> -- Create the RPEExercise table
sqlite> CREATE TABLE RPEExercise (
...>   ExerciseID INTEGER,
...>   RecommendationID INTEGER,
...>   PRIMARY KEY (ExerciseID, RecommendationID),
...>   FOREIGN KEY (ExerciseID) REFERENCES Exercise(ExerciseID),
...>   FOREIGN KEY (RecommendationID) REFERENCES ExerciseRecommendation(RecommendationID)
...> );

```

Users table: Stores information about each user, including their ID, name, age, date of birth, gender, and contact details. Serves as a central table linking users to their health records, goals, posts in the community, and exercise plans.

Exercise Table: Contains details of different exercises, such as exercise name, type (e.g., cardio, strength), difficulty level, and the recommended duration for performing the exercise. Helps in cataloging different exercises that can be recommended within exercise plans.

Goal Table: Records the personal goals of users, such as losing weight or improving stamina, along with a deadline for achieving these goals. Links each goal to a specific user and can help track the progress toward these goals.

HealthParameters Table: Tracks key health parameters of users over time, such as height and weight. Each entry includes a date when these measurements were recorded, allowing for monitoring changes and trends in health metrics.

ProgressLog Table: Logs entries related to the user's progress, including feedback on their achievements or challenges faced. These logs are dated and provide a historical record of the user's journey and improvements over time.

CommunityPost Table: Manages posts made by users in a community forum, facilitating interaction and sharing of information or experiences among users. Each post includes content, the posting date, and the ID of the user who posted it.

ExercisePlan Table: Outlines exercise plans for users, including start and end dates, to structure when and how users should engage in their exercises. These plans are user-specific and can be customized according to individual needs and goals.

ExerciseRecommendation Table: Provides customized exercise recommendations within an exercise plan, potentially including adjustments to the duration for which an exercise should be performed.

Each recommendation is linked to a specific exercise plan.

RPExercise Table: Provides customized exercise recommendations within an exercise plan, potentially including adjustments to the duration for which an exercise should be performed. Each recommendation is linked to a specific exercise plan.

Insert Statements:

User table:

```
sqlite> INSERT INTO User (UserID, Name, Age, DoB, Sex, ContactDetails) VALUES
...> (1, 'Alice Johnson', 28, '1996-04-15', 'F', 'alice.johnson@exercise.com'),
...> (2, 'Bob Smith', 34, '1990-08-22', 'M', 'bob.smith@exercise.com'),
...> (3, 'Carol Taylor', 22, '2002-07-30', 'F', 'carol.taylor@exercise.com'),
...> (4, 'David Lee', 45, '1979-02-17', 'M', 'david.lee@exercise.com'),
...> (5, 'Eva Lopez', 31, '1993-03-25', 'F', 'eva.lopez@exercise.com'),
...> (6, 'Frank Harris', 27, '1997-05-09', 'M', 'frank.harris@exercise.com'),
...> (7, 'Grace Brown', 36, '1988-08-14', 'F', 'grace.brown@exercise.com'),
...> (8, 'Henry Davis', 29, '1995-12-01', 'M', 'henry.davis@exercise.com'),
...> (9, 'Isabel Martinez', 41, '1983-11-21', 'F', 'isabel.martinez@exercise.com'),
...> (10, 'Jack Clark', 19, '2005-04-18', 'M', 'jack.clark@exercise.com'),
...> (11, 'Kimberly White', 29, '1995-05-21', 'F', 'kimberly.white@exercise.com'),
...> (12, 'Lucas Green', 32, '1992-09-13', 'M', 'lucas.green@exercise.com'),
...> (13, 'Megan Turner', 24, '2000-01-26', 'F', 'megan.turner@exercise.com'),
...> (14, 'Nathan Hall', 37, '1987-03-08', 'M', 'nathan.hall@exercise.com'),
...> (15, 'Olivia Scott', 26, '1998-07-19', 'F', 'olivia.scott@exercise.com'),
...> (16, 'Peter Young', 41, '1983-11-11', 'M', 'peter.young@exercise.com'),
...> (17, 'Quinn Edwards', 23, '2001-04-02', 'F', 'quinn.edwards@exercise.com'),
...> (18, 'Rachel King', 35, '1989-06-30', 'F', 'rachel.king@exercise.com'),
...> (19, 'Steven Wright', 30, '1994-10-17', 'M', 'steven.wright@exercise.com'),
...> (20, 'Tina Walker', 28, '1996-12-05', 'F', 'tina.walker@exercise.com'),
...> (21, 'Ulysses Carter', 38, '1986-02-14', 'M', 'ulysses.carter@exercise.com'),
...> (22, 'Violet Gibson', 31, '1993-08-09', 'F', 'violet.gibson@exercise.com'),
...> (23, 'Walter Bishop', 29, '1995-11-23', 'M', 'walter.bishop@exercise.com'),
...> (24, 'Xena Morales', 22, '2002-05-16', 'F', 'xena.morales@exercise.com'),
...> (25, 'Yvonne Jenkins', 43, '1981-09-01', 'F', 'yvonne.jenkins@exercise.com');
```

Exercise Table:

```

sqlite> INSERT INTO Exercise (ExerciseID, Name, Type, DifficultyLevel, RecommendedDuration) VALUES
...> (1, 'Running', 'Cardio', 3, 30),
...> (2, 'Swimming', 'Cardio', 2, 45),
...> (3, 'Cycling', 'Cardio', 4, 60),
...> (4, 'Jump Rope', 'Cardio', 3, 20),
...> (5, 'Squats', 'Strength', 2, 15),
...> (6, 'Bench Press', 'Strength', 4, 45),
...> (7, 'Deadlift', 'Strength', 5, 60),
...> (8, 'Pull Ups', 'Strength', 4, 30),
...> (9, 'Push Ups', 'Calisthenics', 1, 10),
...> (10, 'Plank', 'Core', 2, 5),
...> (11, 'Crunches', 'Core', 1, 10),
...> (12, 'Leg Raise', 'Core', 3, 15),
...> (13, 'Lunges', 'Strength', 2, 15),
...> (14, 'Yoga', 'Flexibility', 1, 60),
...> (15, 'Pilates', 'Flexibility', 2, 60),
...> (16, 'Tai Chi', 'Balance', 1, 30),
...> (17, 'Boxing', 'Cardio', 5, 30),
...> (18, 'Rowing', 'Cardio', 3, 60),
...> (19, 'Elliptical Trainer', 'Cardio', 2, 30),
...> (20, 'Stair Climber', 'Cardio', 4, 20),
...> (21, 'High Knees', 'Cardio', 1, 10),
...> (22, 'Burpees', 'Calisthenics', 5, 15),
...> (23, 'Mountain Climbers', 'Calisthenics', 4, 15),
...> (24, 'Jumping Jacks', 'Cardio', 1, 10),
...> (25, 'Kickboxing', 'Cardio', 4, 45);
sqlite>

```

Goals Table:

```

sqlite> INSERT INTO Goal (GoalID, UserID, Description, Deadline) VALUES
...> (1, 1, 'Run 5k without stopping', '2024-09-01'),
...> (2, 1, 'Lose 10 pounds', '2024-06-30'),
...> (3, 2, 'Gain 5 pounds of muscle', '2024-12-31'),
...> (4, 2, 'Improve marathon time by 10 minutes', '2024-11-15'),
...> (5, 3, 'Complete a triathlon', '2024-07-20'),
...> (6, 3, 'Master yoga inversions', '2024-08-14'),
...> (7, 4, 'Swim 1km', '2024-05-05'),
...> (8, 4, 'Hold a plank for 3 minutes', '2024-04-22'),
...> (9, 5, 'Cycle 50 miles', '2024-10-10'),
...> (10, 5, 'Deadlift 200 pounds', '2024-09-15'),
...> (11, 6, 'Join a sports league', '2024-08-30'),
...> (12, 6, 'Complete 10 pull-ups', '2024-07-12'),
...> (13, 7, 'Hike a high peak', '2024-06-18'),
...> (14, 7, 'Take a dance class', '2024-05-27'),
...> (15, 8, 'Go rock climbing', '2024-10-25'),
...> (16, 8, 'Start a meditation habit', '2024-09-01'),
...> (17, 9, 'Go vegan for a month', '2024-07-15'),
...> (18, 9, 'Walk 10,000 steps daily', '2024-08-09'),
...> (19, 10, 'Learn to cook healthy meals', '2024-04-30'),
...> (20, 10, 'Do a handstand', '2024-05-25'),
...> (21, 11, 'Finish a fitness challenge', '2024-11-01'),
...> (22, 11, 'Reduce body fat by 5%', '2024-12-07'),
...> (23, 12, 'Improve posture', '2024-06-15'),
...> (24, 12, 'Increase flexibility', '2024-07-31'),
...> (25, 13, 'Build a home gym', '2024-09-10'),
...> (26, 13, 'Learn a martial art', '2024-10-17'),
...> (27, 14, 'Complete a 30-day yoga journey', '2024-05-20'),
...> (28, 14, 'Run a half marathon', '2024-08-23'),
...> (29, 15, 'Master a new sport', '2024-04-05'),
...> (30, 15, 'Achieve a balanced diet', '2024-06-30');
sqlite>

```

Health Parameters Table:

```

sqlite> INSERT INTO HealthParameters (ParameterID, UserID, Height, Weight, DateRecorded) VALUES
...> (1, 1, 165.0, 70.0, '2024-04-01'),
...> (2, 1, 165.0, 68.5, '2024-05-01'),
...> (3, 2, 180.0, 80.0, '2024-04-01'),
...> (4, 2, 180.0, 79.5, '2024-05-01'),
...> (5, 3, 170.0, 60.0, '2024-04-01'),
...> (6, 3, 170.0, 59.0, '2024-05-01'),
...> (7, 4, 175.0, 88.0, '2024-04-01'),
...> (8, 4, 175.0, 87.0, '2024-05-01'),
...> (9, 5, 160.0, 55.0, '2024-04-01'),
...> (10, 5, 160.0, 54.5, '2024-05-01'),
...> (11, 6, 182.0, 85.0, '2024-04-01'),
...> (12, 6, 182.0, 84.0, '2024-05-01'),
...> (13, 7, 158.0, 52.0, '2024-04-01'),
...> (14, 7, 158.0, 51.0, '2024-05-01'),
...> (15, 8, 170.0, 76.0, '2024-04-01'),
...> (16, 8, 170.0, 75.0, '2024-05-01'),
...> (17, 9, 174.0, 65.0, '2024-04-01'),
...> (18, 9, 174.0, 64.0, '2024-05-01'),
...> (19, 10, 168.0, 90.0, '2024-04-01'),
...> (20, 10, 168.0, 89.0, '2024-05-01'),
...> (21, 11, 162.0, 62.0, '2024-04-01'),
...> (22, 11, 162.0, 61.0, '2024-05-01'),
...> (23, 12, 177.0, 70.0, '2024-04-01'),
...> (24, 12, 177.0, 69.5, '2024-05-01'),
...> (25, 13, 165.0, 55.0, '2024-04-01'),
...> (26, 13, 165.0, 54.0, '2024-05-01'),
...> (27, 14, 183.0, 82.0, '2024-04-01'),
...> (28, 14, 183.0, 81.0, '2024-05-01'),
...> (29, 15, 158.0, 58.0, '2024-04-01'),
...> (30, 15, 158.0, 57.0, '2024-05-01'),
...> (31, 16, 176.0, 75.0, '2024-04-01'),
...> (32, 16, 176.0, 74.5, '2024-05-01'),
...> (33, 17, 169.0, 63.0, '2024-04-01'),
...> (34, 17, 169.0, 62.5, '2024-05-01'),
...> (35, 18, 172.0, 70.0, '2024-04-01');
sqlite>

```

Progress Log Table:

```

sqlite> INSERT INTO ProgressLog (LogID, UserID, Date, Feedback) VALUES
...> (1, 1, '2024-04-01', 'Felt great after the workout. '),
...> (2, 1, '2024-04-08', 'Struggled with motivation today. '),
...> (3, 2, '2024-04-01', 'Completed my first 5k run. '),
...> (4, 2, '2024-04-08', 'Improved my running time. '),
...> (5, 3, '2024-04-01', 'Yoga session was relaxing. '),
...> (6, 3, '2024-04-08', 'Felt more flexible today. '),
...> (7, 4, '2024-04-01', 'Gym session was tough. '),
...> (8, 4, '2024-04-08', 'Managed to lift heavier weights. '),
...> (9, 5, '2024-04-01', 'Swimming laps felt good. '),
...> (10, 5, '2024-04-08', 'Improved swimming speed. '),
...> (11, 6, '2024-04-01', 'Great cycling session. '),
...> (12, 6, '2024-04-08', 'Beat my previous cycling time. '),
...> (13, 7, '2024-04-01', 'Started learning tennis. '),
...> (14, 7, '2024-04-08', 'Better at forehand shots now. '),
...> (15, 8, '2024-04-01', 'Took a long hike. Challenging but rewarding. '),
...> (16, 8, '2024-04-08', 'Hiked without any breaks. '),
...> (17, 9, '2024-04-01', 'Dance class was fun. '),
...> (18, 9, '2024-04-08', 'Learning new dance moves. '),
...> (19, 10, '2024-04-01', 'Jogging felt easier today. '),
...> (20, 10, '2024-04-08', 'Managed to jog for an extra mile. '),
...> (21, 11, '2024-04-01', 'Started a low-carb diet. '),
...> (22, 11, '2024-04-08', 'Adjusting well to the new diet. '),
...> (23, 12, '2024-04-01', 'Kickboxing is intense. '),
...> (24, 12, '2024-04-08', 'Faster reaction times in kickboxing. '),
...> (25, 13, '2024-04-01', 'Pilates is challenging. '),
...> (26, 13, '2024-04-08', 'Core feels stronger after Pilates. ');
sqlite>

```

Community Post Table:

```

sqlite> INSERT INTO CommunityPost (PostID, UserID, Content, DatePosted) VALUES
...> (1, 1, 'This is my first post about fitness!', '2024-04-01'),
...> (2, 2, 'Excited to join this health community.', '2024-04-01'),
...> (3, 1, 'Here's another post about healthy eating.', '2024-04-02'),
...> (4, 3, 'What a great day to post about mental health!', '2024-04-02'),
...> (5, 2, 'Looking forward to your feedback on my workout routine.', '2024-04-03'),
...> (6, 4, 'Can anyone help me with nutrition tips?', '2024-04-03'),
...> (7, 1, 'Sharing my experience on weight loss strategies.', '2024-04-04'),
...> (8, 3, 'A beautiful day to write about wellness!', '2024-04-04'),
...> (9, 5, 'Starting my journey in personal health today!', '2024-04-05'),
...> (10, 2, 'Another post to share my thoughts on yoga.', '2024-04-05'),
...> (11, 6, 'Hello world of health!', '2024-04-06'),
...> (12, 1, 'Deep dive into the benefits of meditation.', '2024-04-06'),
...> (13, 3, 'Discussing the benefits of herbal supplements.', '2024-04-07'),
...> (14, 4, 'Here's how you set up a home gym.', '2024-04-07'),
...> (15, 5, 'Insights on alternative medicine.', '2024-04-08'),
...> (16, 2, 'Let's talk about the science of sleep.', '2024-04-08'),
...> (17, 1, 'Back again with more dieting tips!', '2024-04-09'),
...> (18, 6, 'A brief post about managing stress.', '2024-04-09'),
...> (19, 3, 'Today I learned something cool about physiotherapy!', '2024-04-10'),
...> (20, 4, 'Exploring new health gadgets.', '2024-04-10'),
...> (21, 5, 'My views on the future of healthcare technology.', '2024-04-11'),
...> (22, 2, 'Is anyone else excited about advancements in medical science?', '2024-04-11'),
...> (23, 1, 'Efficiency in exercise routines.', '2024-04-12'),
...> (24, 6, 'Debating the best diets for overall health.', '2024-04-12'),
...> (25, 3, 'The importance of cybersecurity in healthcare.', '2024-04-13'),
...> (26, 4, 'Essential tools for healthcare professionals.', '2024-04-13'),
...> (27, 5, 'Why choose organic food?', '2024-04-14'),
...> (28, 2, 'How to stay updated with health news.', '2024-04-14'),
...> (29, 6, 'Mobile health apps vs. traditional healthcare.', '2024-04-15'),
...> (30, 1, 'My final post of the series on holistic health.', '2024-04-15');
sqlite>

```

Exercise Plan Table:

```

sqlite> INSERT INTO ExercisePlan (PlanID, UserID, StartDate, EndDate) VALUES
...> (1, 1, '2024-05-01', '2024-05-31'),
...> (2, 2, '2024-05-05', '2024-06-05'),
...> (3, 3, '2024-05-10', '2024-07-10'),
...> (4, 4, '2024-05-15', '2024-08-15'),
...> (5, 5, '2024-05-20', '2024-09-20'),
...> (6, 6, '2024-05-25', '2024-10-25'),
...> (7, 7, '2024-06-01', '2024-06-30'),
...> (8, 8, '2024-06-05', '2024-07-05'),
...> (9, 9, '2024-06-10', '2024-08-10'),
...> (10, 10, '2024-06-15', '2024-09-15'),
...> (11, 11, '2024-06-20', '2024-10-20'),
...> (12, 12, '2024-06-25', '2024-11-25'),
...> (13, 1, '2024-07-01', '2024-07-31'),
...> (14, 2, '2024-07-05', '2024-08-05'),
...> (15, 3, '2024-07-10', '2024-09-10'),
...> (16, 4, '2024-07-15', '2024-10-15'),
...> (17, 5, '2024-07-20', '2024-11-20'),
...> (18, 6, '2024-07-25', '2024-12-25'),
...> (19, 7, '2024-08-01', '2024-08-31'),
...> (20, 8, '2024-08-05', '2024-09-05'),
...> (21, 9, '2024-08-10', '2024-10-10'),
...> (22, 10, '2024-08-15', '2024-11-15'),
...> (23, 11, '2024-08-20', '2024-12-20'),
...> (24, 12, '2024-08-25', '2025-01-25'),
...> (25, 1, '2024-09-01', '2024-09-30'),
...> (26, 2, '2024-09-05', '2024-10-05'),
...> (27, 3, '2024-09-10', '2024-11-10'),
...> (28, 4, '2024-09-15', '2024-12-15'),
...> (29, 5, '2024-09-20', '2025-01-20'),
...> (30, 6, '2024-09-25', '2025-02-25');
sqlite>
sqlite> |

```

Exercise Recommendation Table:

```

sqlite> INSERT INTO ExerciseRecommendation (RecommendationID, PlanID, CustomDuration) VALUES
...> (1, 1, 30),
...> (2, 2, 45),
...> (3, 3, 60),
...> (4, 4, 35),
...> (5, 5, 40),
...> (6, 6, 50),
...> (7, 7, 55),
...> (8, 8, 25),
...> (9, 9, 20),
...> (10, 10, 45),
...> (11, 11, 30),
...> (12, 12, 35),
...> (13, 1, 40),
...> (14, 2, 45),
...> (15, 3, 30),
...> (16, 4, 25),
...> (17, 5, 50),
...> (18, 6, 65),
...> (19, 7, 60),
...> (20, 8, 55),
...> (21, 9, 45),
...> (22, 10, 40),
...> (23, 11, 35),
...> (24, 12, 30),
...> (25, 1, 45),
...> (26, 2, 40),
...> (27, 3, 35),
...> (28, 4, 30),
...> (29, 5, 25),
...> (30, 6, 50);
sqlite>

```

RP Exercise Table:

```

sqlite> INSERT INTO RPExercise (ExerciseID, RecommendationID) VALUES
...> (1, 1),
...> (2, 2),
...> (3, 3),
...> (1, 4),
...> (2, 5),
...> (3, 6),
...> (1, 7),
...> (2, 8),
...> (3, 9),
...> (1, 10),
...> (2, 11),
...> (3, 12),
...> (1, 13),
...> (2, 14),
...> (3, 15),
...> (1, 16),
...> (2, 17),
...> (3, 18),
...> (1, 19),
...> (2, 20),
...> (3, 21),
...> (1, 22),
...> (2, 23),
...> (3, 24),
...> (1, 25),
...> (2, 26),
...> (3, 27),
...> (1, 28),
...> (2, 29),
...> (3, 30);
sqlite>

```

Data Retrieval and Simple Reports

1. Report on user participation in community posts:

Find out how many posts each user has made on the community board.

CODE:

```
SELECT u.UserID, u.Name, COUNT(p.PostID) AS NumberOfPosts
```

```
FROM User u
```

```
JOIN CommunityPost p ON u.UserID = p.UserID
```

```
GROUP BY u.UserID, u.Name;
```

```
sqlite> SELECT u.UserID, u.Name, COUNT(p.PostID) AS NumberOfPosts
...> FROM User u
...> JOIN CommunityPost p ON u.UserID = p.UserID
...> GROUP BY u.UserID, u.Name;
1|Alice Johnson|7
2|Bob Smith|6
3|Carol Taylor|5
4|David Lee|4
5|Eva Lopez|4
6|Frank Harris|4
sqlite>
```

2. Analysis of average weight change per user over recorded periods

Calculate the average weight change for each user based on their first and last recorded weights.

```
SELECT UserID, MIN(DateRecorded) AS FirstRecordDate, MAX(DateRecorded) AS LastRecordDate,
```

```
MIN(Weight) AS InitialWeight, MAX(Weight) AS LatestWeight,
```

```
(MAX(Weight) - MIN(Weight)) AS WeightChange
```

```
FROM HealthParameters
```

```
GROUP BY UserID;
```

```

sqlite> SELECT UserID, MIN(DateRecorded) AS FirstRecordDate, MAX(DateRecorded) AS LastRecordDate
...>     MIN(Weight) AS InitialWeight, MAX(Weight) AS LatestWeight,
...>     (MAX(Weight) - MIN(Weight)) AS WeightChange
...> FROM HealthParameters
...> GROUP BY UserID;
UserID|FirstRecordDate|LastRecordDate|InitialWeight|LatestWeight|WeightChange
1|2024-04-01|2024-05-01|68.5|70.0|1.5
2|2024-04-01|2024-05-01|79.5|80.0|0.5
3|2024-04-01|2024-05-01|59.0|60.0|1.0
4|2024-04-01|2024-05-01|87.0|88.0|1.0
5|2024-04-01|2024-05-01|54.5|55.0|0.5
6|2024-04-01|2024-05-01|84.0|85.0|1.0
7|2024-04-01|2024-05-01|51.0|52.0|1.0
8|2024-04-01|2024-05-01|75.0|76.0|1.0
9|2024-04-01|2024-05-01|64.0|65.0|1.0
10|2024-04-01|2024-05-01|89.0|90.0|1.0
11|2024-04-01|2024-05-01|61.0|62.0|1.0
12|2024-04-01|2024-05-01|69.5|70.0|0.5
13|2024-04-01|2024-05-01|54.0|55.0|1.0
14|2024-04-01|2024-05-01|81.0|82.0|1.0
15|2024-04-01|2024-05-01|57.0|58.0|1.0
16|2024-04-01|2024-05-01|74.5|75.0|0.5
17|2024-04-01|2024-05-01|62.5|63.0|0.5
18|2024-04-01|2024-04-01|70.0|70.0|0.0
sqlite> |

```

3. Summary of exercise plans by duration

Provide a list of exercise plans including their start and end dates, and the total duration in days.

```
SELECT ep.PlanID, ep.UserID, u.Name, ep.StartDate, ep.EndDate, julianday(ep.EndDate) -
```

```
julianday(ep.StartDate) AS DurationDays
```

```
FROM ExercisePlan ep
```

```
JOIN User u ON ep.UserID = u.UserID;
```

```

sqlite> SELECT ep.PlanID, ep.UserID, u.Name, ep.StartDate, ep.EndDate,
...>      julianday(ep.EndDate) - julianday(ep.StartDate) AS DurationDays
...> FROM ExercisePlan ep
...> JOIN User u ON ep.UserID = u.UserID;
PlanID|UserID|Name|StartDate|EndDate|DurationDays
1|1|Alice Johnson|2024-05-01|2024-05-31|30.0
2|2|Bob Smith|2024-05-05|2024-06-05|31.0
3|3|Carol Taylor|2024-05-10|2024-07-10|61.0
4|4|David Lee|2024-05-15|2024-08-15|92.0
5|5|Eva Lopez|2024-05-20|2024-09-20|123.0
6|6|Frank Harris|2024-05-25|2024-10-25|153.0
7|7|Grace Brown|2024-06-01|2024-06-30|29.0
8|8|Henry Davis|2024-06-05|2024-07-05|30.0
9|9|Isabel Martinez|2024-06-10|2024-08-10|61.0
10|10|Jack Clark|2024-06-15|2024-09-15|92.0
11|11|Kimberly White|2024-06-20|2024-10-20|122.0
12|12|Lucas Green|2024-06-25|2024-11-25|153.0
13|1|Alice Johnson|2024-07-01|2024-07-31|30.0
14|2|Bob Smith|2024-07-05|2024-08-05|31.0
15|3|Carol Taylor|2024-07-10|2024-09-10|62.0
16|4|David Lee|2024-07-15|2024-10-15|92.0
17|5|Eva Lopez|2024-07-20|2024-11-20|123.0
18|6|Frank Harris|2024-07-25|2024-12-25|153.0
19|7|Grace Brown|2024-08-01|2024-08-31|30.0
20|8|Henry Davis|2024-08-05|2024-09-05|31.0
21|9|Isabel Martinez|2024-08-10|2024-10-10|61.0
22|10|Jack Clark|2024-08-15|2024-11-15|92.0
23|11|Kimberly White|2024-08-20|2024-12-20|122.0
24|12|Lucas Green|2024-08-25|2025-01-25|153.0
25|1|Alice Johnson|2024-09-01|2024-09-30|29.0
26|2|Bob Smith|2024-09-05|2024-10-05|30.0
27|3|Carol Taylor|2024-09-10|2024-11-10|61.0
28|4|David Lee|2024-09-15|2024-12-15|91.0
29|5|Eva Lopez|2024-09-20|2025-01-20|122.0
30|6|Frank Harris|2024-09-25|2025-02-25|153.0

```

4. Feedback frequency and content from users

List all feedback entries provided by users, showing the frequency of their feedback and the contents.

```
SELECT UserID, Date, Feedback, COUNT(LogID) AS FeedbackCount
```

```
FROM ProgressLog
```

```
GROUP BY UserID, Date
```

```
ORDER BY UserID, Date;
```



```

sqlite> SELECT UserID, Date, Feedback, COUNT(LogID) AS FeedbackCount
...> FROM ProgressLog
...> GROUP BY UserID, Date
...> ORDER BY UserID, Date;
UserID|Date|Feedback|FeedbackCount
1|2024-04-01|Felt great after the workout.|1
1|2024-04-08|Struggled with motivation today.|1
2|2024-04-01|Completed my first 5k run.|1
2|2024-04-08|Improved my running time.|1
3|2024-04-01|Yoga session was relaxing.|1
3|2024-04-08|Felt more flexible today.|1
4|2024-04-01|Gym session was tough.|1
4|2024-04-08|Managed to lift heavier weights.|1
5|2024-04-01|Swimming laps felt good.|1
5|2024-04-08|Improved swimming speed.|1
6|2024-04-01|Great cycling session.|1
6|2024-04-08|Beat my previous cycling time.|1
7|2024-04-01|Started learning tennis.|1
7|2024-04-08|Better at forehand shots now.|1
8|2024-04-01|Took a long hike. Challenging but rewarding.|1
8|2024-04-08|Hiked without any breaks.|1
9|2024-04-01|Dance class was fun.|1
9|2024-04-08|Learning new dance moves.|1
10|2024-04-01|Jogging felt easier today.|1
10|2024-04-08|Managed to jog for an extra mile.|1
11|2024-04-01|Started a low-carb diet.|1
11|2024-04-08|Adjusting well to the new diet.|1
12|2024-04-01|Kickboxing is intense.|1
12|2024-04-08|Faster reaction times in kickboxing.|1
13|2024-04-01|Pilates is challenging.|1
13|2024-04-08|Core feels stronger after Pilates.|1
sqlite> |

```

5. Exercise recommendations and actual durations

Compare the custom durations in exercise recommendations to the recommended durations for those exercises.

```

SELECT r.RecommendationID, r.PlanID, e.ExercisID, e.Name, e.RecommendedDuration,
r.CustomDuration
FROM RPEExercise rp
JOIN ExerciseRecommendation r ON rp.RecommendationID = r.RecommendationID
JOIN Exercise e ON rp.ExercisID = e.ExercisID;

```

```
sqlite> SELECT r.RecommendationID, r.PlanID, e.ExerciseID, e.Name, e.RecommendedDuration, r.CustomDuration
...> FROM RPExercise rp
...> JOIN ExerciseRecommendation r ON rp.RecommendationID = r.RecommendationID
...> JOIN Exercise e ON rp.ExerciseID = e.ExerciseID;
RecommendationID|PlanID|ExerciseID|Name|RecommendedDuration|CustomDuration
1|1|1|Running|30|30
2|2|2|Swimming|45|45
3|3|3|Cycling|60|60
4|4|1|Running|30|35
5|5|2|Swimming|45|40
6|6|3|Cycling|60|50
7|7|1|Running|30|55
8|8|2|Swimming|45|25
9|9|3|Cycling|60|20
10|10|1|Running|30|45
11|11|2|Swimming|45|30
12|12|3|Cycling|60|35
13|1|1|Running|30|40
14|2|2|Swimming|45|45
15|3|3|Cycling|60|30
16|4|1|Running|30|25
17|5|2|Swimming|45|50
18|6|3|Cycling|60|65
19|7|1|Running|30|60
20|8|2|Swimming|45|55
21|9|3|Cycling|60|45
22|10|1|Running|30|40
23|11|2|Swimming|45|35
24|12|3|Cycling|60|30
25|1|1|Running|30|45
26|2|2|Swimming|45|40
27|3|3|Cycling|60|35
28|4|1|Running|30|30
29|5|2|Swimming|45|25
30|6|3|Cycling|60|50
sqlite> |
```

[P.T.O DB CODE NEXT SECTION]

DB CODE

-- Create the User table

```
CREATE TABLE User (  
  UserID INTEGER PRIMARY KEY,  
  Name TEXT NOT NULL,  
  Age INTEGER,  
  DoB DATE,  
  Sex TEXT,  
  ContactDetails TEXT  
);
```

-- Create the Exercise table

```
CREATE TABLE Exercise (  
  ExerciseID INTEGER PRIMARY KEY,  
  Name TEXT,  
  Type TEXT,  
  DifficultyLevel INTEGER,  
  RecommendedDuration INTEGER  
);
```

-- Create the Goal table

```
CREATE TABLE Goal (  
  GoalID INTEGER PRIMARY KEY,  
  UserID INTEGER,  
  Description TEXT,  
  Deadline DATE,  
  FOREIGN KEY (UserID) REFERENCES User(UserID)  
);
```

-- Create the HealthParameters table

```
CREATE TABLE HealthParameters (  
    ParameterID INTEGER PRIMARY KEY,  
    UserID INTEGER,  
    Height REAL,  
    Weight REAL,  
    DateRecorded DATE,  
    FOREIGN KEY (UserID) REFERENCES User(UserID)  
);
```

-- Create the ProgressLog table

```
CREATE TABLE ProgressLog (  
    LogID INTEGER PRIMARY KEY,  
    UserID INTEGER,  
    Date DATE,  
    Feedback TEXT,  
    FOREIGN KEY (UserID) REFERENCES User(UserID)  
);
```

-- Create the CommunityPost table

```
CREATE TABLE CommunityPost (  
    PostID INTEGER PRIMARY KEY,  
    UserID INTEGER,  
    Content TEXT,  
    DatePosted DATE,  
    FOREIGN KEY (UserID) REFERENCES User(UserID)  
);
```

-- Create the ExercisePlan table

```
CREATE TABLE ExercisePlan (  
    PlanID INTEGER PRIMARY KEY,  
    UserID INTEGER,
```

```

StartDate DATE,
EndDate DATE,
FOREIGN KEY (UserID) REFERENCES User(UserID)
);

```

-- Create the ExerciseRecommendation table

```

CREATE TABLE ExerciseRecommendation (
  RecommendationID INTEGER PRIMARY KEY,
  PlanID INTEGER,
  CustomDuration INTEGER,
  FOREIGN KEY (PlanID) REFERENCES ExercisePlan(PlanID)
);

```

-- Create the RPExercise table

```

CREATE TABLE RPExercise (
  ExercisID INTEGER,
  RecommendationID INTEGER,
  PRIMARY KEY (ExercisID, RecommendationID),
  FOREIGN KEY (ExercisID) REFERENCES Exercise(ExercisID),
  FOREIGN KEY (RecommendationID) REFERENCES
ExerciseRecommendation(RecommendationID)
);

```

```

INSERT INTO User (UserID, Name, Age, DoB, Sex, ContactDetails) VALUES
(1, 'Alice Johnson', 28, '1996-04-15', 'F', 'alice.johnson@exercise.com'),
(2, 'Bob Smith', 34, '1990-08-22', 'M', 'bob.smith@exercise.com'),
(3, 'Carol Taylor', 22, '2002-07-30', 'F', 'carol.taylor@exercise.com'),
(4, 'David Lee', 45, '1979-02-17', 'M', 'david.lee@exercise.com'),
(5, 'Eva Lopez', 31, '1993-03-25', 'F', 'eva.lopez@exercise.com'),
(6, 'Frank Harris', 27, '1997-05-09', 'M', 'frank.harris@exercise.com'),
(7, 'Grace Brown', 36, '1988-08-14', 'F', 'grace.brown@exercise.com'),

```

(8, 'Henry Davis', 29, '1995-12-01', 'M', 'henry.davis@exercise.com'),
 (9, 'Isabel Martinez', 41, '1983-11-21', 'F', 'isabel.martinez@exercise.com'),
 (10, 'Jack Clark', 19, '2005-04-18', 'M', 'jack.clark@exercise.com'),
 (11, 'Kimberly White', 29, '1995-05-21', 'F', 'kimberly.white@exercise.com'),
 (12, 'Lucas Green', 32, '1992-09-13', 'M', 'lucas.green@exercise.com'),
 (13, 'Megan Turner', 24, '2000-01-26', 'F', 'megan.turner@exercise.com'),
 (14, 'Nathan Hall', 37, '1987-03-08', 'M', 'nathan.hall@exercise.com'),
 (15, 'Olivia Scott', 26, '1998-07-19', 'F', 'olivia.scott@exercise.com'),
 (16, 'Peter Young', 41, '1983-11-11', 'M', 'peter.young@exercise.com'),
 (17, 'Quinn Edwards', 23, '2001-04-02', 'F', 'quinn.edwards@exercise.com'),
 (18, 'Rachel King', 35, '1989-06-30', 'F', 'rachel.king@exercise.com'),
 (19, 'Steven Wright', 30, '1994-10-17', 'M', 'steven.wright@exercise.com'),
 (20, 'Tina Walker', 28, '1996-12-05', 'F', 'tina.walker@exercise.com'),
 (21, 'Ulysses Carter', 38, '1986-02-14', 'M', 'ulysses.carter@exercise.com'),
 (22, 'Violet Gibson', 31, '1993-08-09', 'F', 'violet.gibson@exercise.com'),
 (23, 'Walter Bishop', 29, '1995-11-23', 'M', 'walter.bishop@exercise.com'),
 (24, 'Xena Morales', 22, '2002-05-16', 'F', 'xena.morales@exercise.com'),
 (25, 'Yvonne Jenkins', 43, '1981-09-01', 'F', 'yvonne.jenkins@exercise.com');

INSERT INTO Exercise (ExerciseID, Name, Type, DifficultyLevel,
 RecommendedDuration) VALUES

(1, 'Running', 'Cardio', 3, 30),
 (2, 'Swimming', 'Cardio', 2, 45),
 (3, 'Cycling', 'Cardio', 4, 60),
 (4, 'Jump Rope', 'Cardio', 3, 20),
 (5, 'Squats', 'Strength', 2, 15),
 (6, 'Bench Press', 'Strength', 4, 45),
 (7, 'Deadlift', 'Strength', 5, 60),
 (8, 'Pull Ups', 'Strength', 4, 30),
 (9, 'Push Ups', 'Calisthenics', 1, 10),

(10, 'Plank', 'Core', 2, 5),
 (11, 'Crunches', 'Core', 1, 10),
 (12, 'Leg Raise', 'Core', 3, 15),
 (13, 'Lunges', 'Strength', 2, 15),
 (14, 'Yoga', 'Flexibility', 1, 60),
 (15, 'Pilates', 'Flexibility', 2, 60),
 (16, 'Tai Chi', 'Balance', 1, 30),
 (17, 'Boxing', 'Cardio', 5, 30),
 (18, 'Rowing', 'Cardio', 3, 60),
 (19, 'Elliptical Trainer', 'Cardio', 2, 30),
 (20, 'Stair Climber', 'Cardio', 4, 20),
 (21, 'High Knees', 'Cardio', 1, 10),
 (22, 'Burpees', 'Calisthenics', 5, 15),
 (23, 'Mountain Climbers', 'Calisthenics', 4, 15),
 (24, 'Jumping Jacks', 'Cardio', 1, 10),
 (25, 'Kickboxing', 'Cardio', 4, 45);

INSERT INTO Goal (GoalID, UserID, Description, Deadline) VALUES

(1, 1, 'Run 5k without stopping', '2024-09-01'),
 (2, 1, 'Lose 10 pounds', '2024-06-30'),
 (3, 2, 'Gain 5 pounds of muscle', '2024-12-31'),
 (4, 2, 'Improve marathon time by 10 minutes', '2024-11-15'),
 (5, 3, 'Complete a triathlon', '2024-07-20'),
 (6, 3, 'Master yoga inversions', '2024-08-14'),
 (7, 4, 'Swim 1km', '2024-05-05'),
 (8, 4, 'Hold a plank for 3 minutes', '2024-04-22'),
 (9, 5, 'Cycle 50 miles', '2024-10-10'),
 (10, 5, 'Deadlift 200 pounds', '2024-09-15'),
 (11, 6, 'Join a sports league', '2024-08-30'),
 (12, 6, 'Complete 10 pull-ups', '2024-07-12'),
 (13, 7, 'Hike a high peak', '2024-06-18'),

(14, 7, 'Take a dance class', '2024-05-27'),
(15, 8, 'Go rock climbing', '2024-10-25'),
(16, 8, 'Start a meditation habit', '2024-09-01'),
(17, 9, 'Go vegan for a month', '2024-07-15'),
(18, 9, 'Walk 10,000 steps daily', '2024-08-09'),
(19, 10, 'Learn to cook healthy meals', '2024-04-30'),
(20, 10, 'Do a handstand', '2024-05-25'),
(21, 11, 'Finish a fitness challenge', '2024-11-01'),
(22, 11, 'Reduce body fat by 5%', '2024-12-07'),
(23, 12, 'Improve posture', '2024-06-15'),
(24, 12, 'Increase flexibility', '2024-07-31'),
(25, 13, 'Build a home gym', '2024-09-10'),
(26, 13, 'Learn a martial art', '2024-10-17'),
(27, 14, 'Complete a 30-day yoga journey', '2024-05-20'),
(28, 14, 'Run a half marathon', '2024-08-23'),
(29, 15, 'Master a new sport', '2024-04-05'),
(30, 15, 'Achieve a balanced diet', '2024-06-30');

INSERT INTO HealthParameters (ParameterID, UserID, Height, Weight,
DateRecorded) VALUES

(1, 1, 165.0, 70.0, '2024-04-01'),
(2, 1, 165.0, 68.5, '2024-05-01'),
(3, 2, 180.0, 80.0, '2024-04-01'),
(4, 2, 180.0, 79.5, '2024-05-01'),
(5, 3, 170.0, 60.0, '2024-04-01'),
(6, 3, 170.0, 59.0, '2024-05-01'),
(7, 4, 175.0, 88.0, '2024-04-01'),
(8, 4, 175.0, 87.0, '2024-05-01'),
(9, 5, 160.0, 55.0, '2024-04-01'),
(10, 5, 160.0, 54.5, '2024-05-01'),


```
(11, 6, 182.0, 85.0, '2024-04-01'),  
(12, 6, 182.0, 84.0, '2024-05-01'),  
(13, 7, 158.0, 52.0, '2024-04-01'),  
(14, 7, 158.0, 51.0, '2024-05-01'),  
(15, 8, 170.0, 76.0, '2024-04-01'),  
(16, 8, 170.0, 75.0, '2024-05-01'),  
(17, 9, 174.0, 65.0, '2024-04-01'),  
(18, 9, 174.0, 64.0, '2024-05-01'),  
(19, 10, 168.0, 90.0, '2024-04-01'),  
(20, 10, 168.0, 89.0, '2024-05-01'),  
(21, 11, 162.0, 62.0, '2024-04-01'),  
(22, 11, 162.0, 61.0, '2024-05-01'),  
(23, 12, 177.0, 70.0, '2024-04-01'),  
(24, 12, 177.0, 69.5, '2024-05-01'),  
(25, 13, 165.0, 55.0, '2024-04-01'),  
(26, 13, 165.0, 54.0, '2024-05-01'),  
(27, 14, 183.0, 82.0, '2024-04-01'),  
(28, 14, 183.0, 81.0, '2024-05-01'),  
(29, 15, 158.0, 58.0, '2024-04-01'),  
(30, 15, 158.0, 57.0, '2024-05-01'),  
(31, 16, 176.0, 75.0, '2024-04-01'),  
(32, 16, 176.0, 74.5, '2024-05-01'),  
(33, 17, 169.0, 63.0, '2024-04-01'),  
(34, 17, 169.0, 62.5, '2024-05-01'),  
(35, 18, 172.0, 70.0, '2024-04-01');
```

```
INSERT INTO ProgressLog (LogID, UserID, Date, Feedback) VALUES
```

```
(1, 1, '2024-04-01', 'Felt great after the workout.'),  
(2, 1, '2024-04-08', 'Struggled with motivation today.'),  
(3, 2, '2024-04-01', 'Completed my first 5k run.'),
```

(4, 2, '2024-04-08', 'Improved my running time.'),
 (5, 3, '2024-04-01', 'Yoga session was relaxing.'),
 (6, 3, '2024-04-08', 'Felt more flexible today.'),
 (7, 4, '2024-04-01', 'Gym session was tough.'),
 (8, 4, '2024-04-08', 'Managed to lift heavier weights.'),
 (9, 5, '2024-04-01', 'Swimming laps felt good.'),
 (10, 5, '2024-04-08', 'Improved swimming speed.'),
 (11, 6, '2024-04-01', 'Great cycling session.'),
 (12, 6, '2024-04-08', 'Beat my previous cycling time.'),
 (13, 7, '2024-04-01', 'Started learning tennis.'),
 (14, 7, '2024-04-08', 'Better at forehand shots now.'),
 (15, 8, '2024-04-01', 'Took a long hike. Challenging but rewarding.'),
 (16, 8, '2024-04-08', 'Hiked without any breaks.'),
 (17, 9, '2024-04-01', 'Dance class was fun.'),
 (18, 9, '2024-04-08', 'Learning new dance moves.'),
 (19, 10, '2024-04-01', 'Jogging felt easier today.'),
 (20, 10, '2024-04-08', 'Managed to jog for an extra mile.'),
 (21, 11, '2024-04-01', 'Started a low-carb diet.'),
 (22, 11, '2024-04-08', 'Adjusting well to the new diet.'),
 (23, 12, '2024-04-01', 'Kickboxing is intense.'),
 (24, 12, '2024-04-08', 'Faster reaction times in kickboxing.'),
 (25, 13, '2024-04-01', 'Pilates is challenging.'),
 (26, 13, '2024-04-08', 'Core feels stronger after Pilates.'),

INSERT INTO CommunityPost (PostID, UserID, Content, DatePosted) **VALUES**

(1, 1, 'This is my first post about fitness!', '2024-04-01'),
 (2, 2, 'Excited to join this health community.', '2024-04-01'),
 (3, 1, 'Here's another post about healthy eating.', '2024-04-02'),
 (4, 3, 'What a great day to post about mental health!', '2024-04-02'),
 (5, 2, 'Looking forward to your feedback on my workout routine.', '2024-04-03'),
 (6, 4, 'Can anyone help me with nutrition tips?', '2024-04-03'),

(7, 1, 'Sharing my experience on weight loss strategies.', '2024-04-04'),
 (8, 3, 'A beautiful day to write about wellness!', '2024-04-04'),
 (9, 5, 'Starting my journey in personal health today!', '2024-04-05'),
 (10, 2, 'Another post to share my thoughts on yoga.', '2024-04-05'),
 (11, 6, 'Hello world of health!', '2024-04-06'),
 (12, 1, 'Deep dive into the benefits of meditation.', '2024-04-06'),
 (13, 3, 'Discussing the benefits of herbal supplements.', '2024-04-07'),
 (14, 4, 'Here's how you set up a home gym.', '2024-04-07'),
 (15, 5, 'Insights on alternative medicine.', '2024-04-08'),
 (16, 2, 'Let's talk about the science of sleep.', '2024-04-08'),
 (17, 1, 'Back again with more dieting tips!', '2024-04-09'),
 (18, 6, 'A brief post about managing stress.', '2024-04-09'),
 (19, 3, 'Today I learned something cool about physiotherapy!', '2024-04-10'),
 (20, 4, 'Exploring new health gadgets.', '2024-04-10'),
 (21, 5, 'My views on the future of healthcare technology.', '2024-04-11'),
 (22, 2, 'Is anyone else excited about advancements in medical science?', '2024-04-11'),
 (23, 1, 'Efficiency in exercise routines.', '2024-04-12'),
 (24, 6, 'Debating the best diets for overall health.', '2024-04-12'),
 (25, 3, 'The importance of cybersecurity in healthcare.', '2024-04-13'),
 (26, 4, 'Essential tools for healthcare professionals.', '2024-04-13'),
 (27, 5, 'Why choose organic food?', '2024-04-14'),
 (28, 2, 'How to stay updated with health news.', '2024-04-14'),
 (29, 6, 'Mobile health apps vs. traditional healthcare.', '2024-04-15'),
 (30, 1, 'My final post of the series on holistic health.', '2024-04-15');

INSERT INTO ExercisePlan (PlanID, UserID, StartDate, EndDate) **VALUES**

(1, 1, '2024-05-01', '2024-05-31'),
 (2, 2, '2024-05-05', '2024-06-05'),
 (3, 3, '2024-05-10', '2024-07-10'),
 (4, 4, '2024-05-15', '2024-08-15'),

```
(5, 5, '2024-05-20', '2024-09-20'),  
(6, 6, '2024-05-25', '2024-10-25'),  
(7, 7, '2024-06-01', '2024-06-30'),  
(8, 8, '2024-06-05', '2024-07-05'),  
(9, 9, '2024-06-10', '2024-08-10'),  
(10, 10, '2024-06-15', '2024-09-15'),  
(11, 11, '2024-06-20', '2024-10-20'),  
(12, 12, '2024-06-25', '2024-11-25'),  
(13, 1, '2024-07-01', '2024-07-31'),  
(14, 2, '2024-07-05', '2024-08-05'),  
(15, 3, '2024-07-10', '2024-09-10'),  
(16, 4, '2024-07-15', '2024-10-15'),  
(17, 5, '2024-07-20', '2024-11-20'),  
(18, 6, '2024-07-25', '2024-12-25'),  
(19, 7, '2024-08-01', '2024-08-31'),  
(20, 8, '2024-08-05', '2024-09-05'),  
(21, 9, '2024-08-10', '2024-10-10'),  
(22, 10, '2024-08-15', '2024-11-15'),  
(23, 11, '2024-08-20', '2024-12-20'),  
(24, 12, '2024-08-25', '2025-01-25'),  
(25, 1, '2024-09-01', '2024-09-30'),  
(26, 2, '2024-09-05', '2024-10-05'),  
(27, 3, '2024-09-10', '2024-11-10'),  
(28, 4, '2024-09-15', '2024-12-15'),  
(29, 5, '2024-09-20', '2025-01-20'),  
(30, 6, '2024-09-25', '2025-02-25');
```

```
INSERT INTO ExerciseRecommendation (RecommendationID, PlanID,  
CustomDuration) VALUES  
(1, 1, 30),  
(2, 2, 45),
```

(3, 3, 60),
(4, 4, 35),
(5, 5, 40),
(6, 6, 50),
(7, 7, 55),
(8, 8, 25),
(9, 9, 20),
(10, 10, 45),
(11, 11, 30),
(12, 12, 35),
(13, 1, 40),
(14, 2, 45),
(15, 3, 30),
(16, 4, 25),
(17, 5, 50),
(18, 6, 65),
(19, 7, 60),
(20, 8, 55),
(21, 9, 45),
(22, 10, 40),
(23, 11, 35),
(24, 12, 30),
(25, 1, 45),
(26, 2, 40),
(27, 3, 35),
(28, 4, 30),
(29, 5, 25),
(30, 6, 50);

INSERT INTO RPExercise (ExerciseID, RecommendationID) **VALUES**

(1, 1),
(2, 2),
(3, 3),
(1, 4),
(2, 5),
(3, 6),
(1, 7),
(2, 8),
(3, 9),
(1, 10),
(2, 11),
(3, 12),
(1, 13),
(2, 14),
(3, 15),
(1, 16),
(2, 17),
(3, 18),
(1, 19),
(2, 20),
(3, 21),
(1, 22),
(2, 23),
(3, 24),
(1, 25),
(2, 26),
(3, 27),
(1, 28),
(2, 29),
(3, 30);

-- Data Retrieval and Simple Reports

-- 1.

```
SELECT u.UserID, u.Name, COUNT(p.PostID) AS NumberOfPosts
FROM User u
JOIN CommunityPost p ON u.UserID = p.UserID
GROUP BY u.UserID, u.Name;
```

-- 2.

```
SELECT UserID, MIN(DateRecorded) AS FirstRecordDate, MAX(DateRecorded) AS
LastRecordDate,
    MIN(Weight) AS InitialWeight, MAX(Weight) AS LatestWeight,
    (MAX(Weight) - MIN(Weight)) AS WeightChange
FROM HealthParameters
GROUP BY UserID;
```

-- 3.

```
SELECT ep.PlanID, ep.UserID, u.Name, ep.StartDate, ep.EndDate,
    julianday(ep.EndDate) - julianday(ep.StartDate) AS DurationDays
FROM ExercisePlan ep
JOIN User u ON ep.UserID = u.UserID;
```

-- 4.

```
SELECT UserID, Date, Feedback, COUNT(LogID) AS FeedbackCount
FROM ProgressLog
GROUP BY UserID, Date
ORDER BY UserID, Date;
```

-- 5.

```
SELECT r.RecommendationID, r.PlanID, e.ExerciseID, e.Name,
e.RecommendedDuration, r.CustomDuration
FROM RPExercise rp
```

```
JOIN ExerciseRecommendation r ON rp.RecommendationID = r.RecommendationID  
JOIN Exercise e ON rp.ExerciseID = e.ExerciseID;
```