Meghna Malhotra
Jugal Lodaya

08.01.2016
Project Report

HealthyMe

Application URL: http://resin.cci.drexel.edu/~jjl354/healthy_me

Description:

We have created a utilitarian personal health and fitness monitoring application. Our database has been designed to keep track of information about a user's health, including basic body statistics, daily nutrition, physical activities, steps, sleep schedule, and heart rate.

We assume that each user has a fitness tracker such as a FitBit that will record their physical activities, steps, sleep schedule and heart rate, and sync the collected data with our database. Additionally, users will be able to manually log their meals and activities on our UI. We will use this information to determine a user's current heath and fitness status. Users will be able to check their BMI, change in weight, calories consumed, calories burned, average steps walked, heart rate during activity, and resting heart rate.

Our application is user-focused and all the entities we have modeled relate to our main entity - Users. They are: BodyStats, Activities, Nutrition, Sleep, Steps, and HeartRate.

Entity Sets, Relationship Sets and Business Rules:

1) Users:
   - Primary key: user_id
   - Attributes: first_name, last_name, age
   - Constraints: Users have at least one value for BodyStats, Steps, Nutrition, Sleep and HeartRate

2) Have_BodyStats:
   - Primary key: stat_id
   - Foreign key: user_id from Users
   - Attributes: height, weight
   - Constraints: Each body statistic belongs to exactly one user

3) Walk_Steps:
   - Primary key: step_id
   - Foreign key: user_id from Users
   - Attributes: num_steps, calories_burned, date
   - Constraints: Each step count belongs to exactly one user

4) Perform_Activities:
   - Primary key: activity_id
   - Foreign key: user_id from Users
   - Attributes: name, date, start_time, end_time, calories_burned
   - Constraints: Each activity is performed by exactly one user

5) Need_Nutrition:
   - Primary key: meal_id
   - Foreign key: user_id from Users
   - Attributes: food_name, meal_type, date, calories
   - Constraints: Each meal is eaten by exactly one user

6) Need_Sleep:
   - Primary key: sleep_session_id
   - Foreign key: user_id from Users
   - Attributes: date, start_time, end_time
   - Constraints: Each sleep session has exactly one user

7) Have_HeartRate:
   - Primary key: hr_id
   - Foreign key: user_id from Users
   - Attributes: heart_rate, date, start_time, end_time
   - Constraints: Each heart rate value belongs to exactly one user

Translation of the ER Diagram to a RDB Schema:

We used a straightforward approach when translating the entity relationship diagram into the relational database schema. We started with our core entity Users, and wrote SQL code that captured all the constraints we wanted this entity to embody. For the rest of the tables, we chose to combine the relationship set and entity set into the same table. In this manner, we created the following tables:

1. have_BodyStats
2. walk_Steps
3. need_Nutrition
4. perform_Activities
5. need_Sleep
6. have_HeartRate

We chose to implement our design in this format because we felt it awarded us increased flexibility in terms of writing queries for our database.

Data Acquisition:

We manually generated realistic data. We found websites with nutritional information of different foods and emulated these in our database. We added different types of workouts and the average calories burned based on personal experience and online exercise guides. We wrote insert table statements for all tables except one. For this last table – have_HeartRate – we created a CSV file with the relevant data and imported this file into our DB using the Postgres “\copy” command.

User Interface:

We have created a clean and simple user interface that provides 15 different functions and interactions. They are as follows:

1. Add a user to the database that doesn’t already exist.
2. Add body statistics for a user that already exists.
3. Add activity for a user.
4. Add nutrition for a user.
5. Calculate BMI (Body Mass Index) for a specified date.
6. Determine change in weight between 2 specified dates.
7. Get a breakdown of calories consumed in each meal, for a specified date.
8. Get the total calories consumed on a specified date.
9. Get the total calories burned on a specified date.
10. Calculate the average steps walked per day, within a specified date range.
11. Calculate average calories burned per day, within a specified date range.
12. Calculate average calories consumed per meal, within a specified date range.
13. Get the maximum heart rate during an activity period.
14. Get the average heart rate during periods of sleep.
15. Calculate your resting heart rate.

IMPORTANT NOTE:

We start by inserting 10 different users in our database. In the interest of saving space, all tables except Users carry data only for 2 different users. Also, the data is only spread over a 5-day period from 2016-07-04 (Monday) to 2016-07-08 (Friday).
This means that functions 5-15 will only work with the following values:

- Users:
  - ‘Ford Prefect’
  - ‘Daenerys Targaryen’
- Dates:
  - ‘2016-07-04’
  - ‘2016-07-05’
  - ‘2016-07-06’
  - ‘2016-07-07’
  - ‘2016-07-08’