

# GROUP PROJECT 1

P Kumar Sachin

Due: April 15, 2020

## TEAM NAME: *Group 6*

Benjamin Nodvin

Harrison Wright

Will Spagnoli

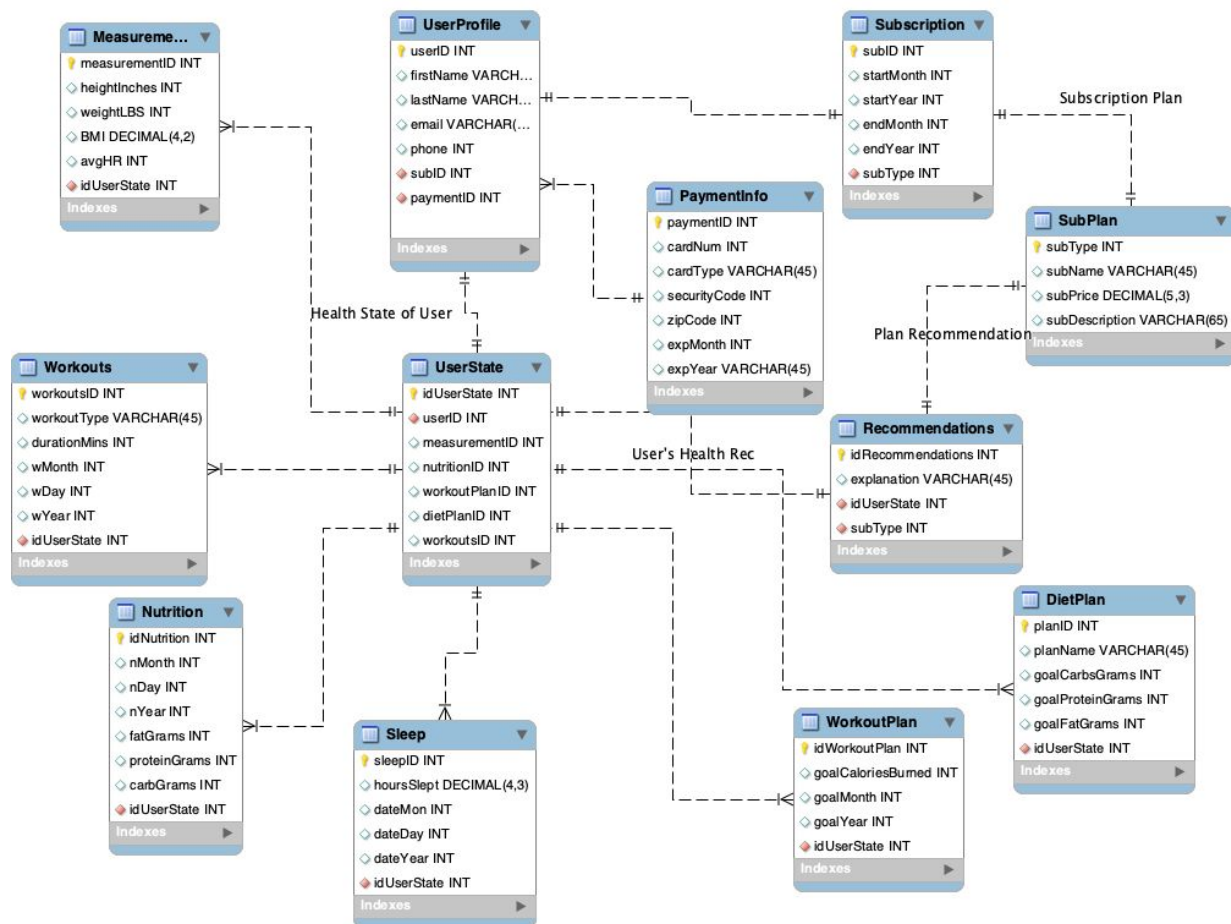
Daniel Saul

## Problem Description

For the fitness application FitBit, the community revolving around the company has voiced their dissatisfaction over issues regarding FitBit's subscription plans and personal coaching. Sometimes, the company's databases within the application may either nonexistent or not working properly. FitBit Premium is a somewhat more expensive subscription that includes programs, insights, challenges, additional sleep score details, and personal coaching. Those who spend the extra funds on these programs would like to enjoy the bonuses without having issues purchasing and running them, benefiting both the company and customer. In some cases, premium plans are not working and disappearing from older application interfaces. Customers are showing confusion with which payment plans to purchase based off of whether or not they work and are the most efficient.

To solve this problem and help the business, we advise that the premium plans be fixed or new plans should be created. We have implemented two new entities, "Recommendations" and "Sleep." The "Recommendations" entity consists of attributes that provide explanations for which plans to purchase as well as new payment plans to introduce into FitBit's business model. The "Sleep" entity is meant to create an entirely new database for FitBit where customers can track every detail of their sleep schedule more efficiently. This is an upgrade from the company's current sleep tracking database. With care and expertise, we can help FitBit fix their subscription and customer satisfaction issues.

To begin, our data model for FitBit is centered around the new addition of a user's current health state. The UserState entity includes ID's of measurements, nutrition, workout plan, diet plan, workouts, and the user's ID. The UserState is tailored to the user of the app, only one user can currently have a singular state of health. One user can have different types of payment info because most people own more than one credit or debit card. A user can only have one subscription and the subscription can only have one subscription plan. Both of these entities include typical information available in a subscription such as names, dates, prices, and descriptions. Each subscription plan comes with one plan recommendation and one recommendation can be linked to a certain user health state. But a user's health state can give many and different recommendations. The user's state can have multiple workout plans which include goal dates and calorie burning goals. The user's state can also have many sleep schedule inputs, as well as many nutritional inputs containing the dates at which certain amounts of fat, proteins, and carbs are consumed. A user can also enter many workouts and measurements of their body with typical info related to those entities--each relating to the one unique user state.



## Data Dictionary

Table: **DietPlan**

Column Name	Description	Data Type	Size	Format	Key
planID	Unique sequential number indicating the diet plan of the user	Text	11		PK
planName	Name of the diet plan based on user goals	Text	45		
goalCarbsGrams	User's goal carb intake in grams	Numeric	11		
goalProteinGrams	User's goal protein intake in grams	Numeric	11		
goalFatGrams	User's goal fat intake in grams	Numeric	11		
idUserState	Unique number indicating a specific user	Text	11		FK (ref UserState)

Table: **Measurements**

Column Name	Description	Data Type	Size	Format	Key
measurementID	Unique number linking a user to their measurement info	Text	11		PK
heightInches	User's height in inches	Numeric	11		
weightLBS	User's current weight in pounds	Numeric	11		
BMI	User's body mass index	Numeric	11		
avgHR	User's average heart rate in beats per minute	Numeric	11		

idUserState	Unique number indicating a specific user	Text	11		FK (ref UserState)
-------------	--	------	----	--	--------------------

Table: **Nutrition**

Column Name	Description	Data Type	Size	Format	Key
idNutrition	Unique number of individual's nutrition levels	Text	11		PK
nMonth	Date's month to track nutrition	Date	2	11	
nDay	Date's day to track nutrition	Date	2	11	
nYear	Date's year to track nutrition	Date	4	1111	
fatGrams	User's grams of ingested fat on given date	Numeric	11		
proteinGrams	User's grams of ingested protein on given date	Numeric	11		
carbGrams	User's grams of ingested carbs on given date	Numeric	11		
idUserState	Unique number indicating a specific user	Text	11		FK (ref UserState)

Table: **PaymentInfo**

Column Name	Description	Data Type	Size	Format	Key
paymentID	Unique number relating to a unique monetary transactions	Text	11		PK

cardNum	Unique number linked to user's credit card	Text	12	1111-1111-1111-1111	
cardType	Brand of credit card	Text 45			
cardExpMonth	Month of credit card's expiration	Date	2	11	
cardExpYear	Year of credit card's expiration	Date	4	1111	
securityCode	Unique security code of a credit card	Text	3	111	
zipCode	Unique zip code linked to a credit card	Text	5	11111	

Table: **Recommendations**

Column Name	Description	Data Type	Size	Format	Key
idRecommendations	Unique recommendation code	Text	11		PK
explanation	Explanation of recommendation received by a user	Text	45		
idUserState	Unique number indicating a specific user	Text	11		FK (ref UserState)
subType	Type of subscription held by user	Text	11		FK (ref SubPlan)

Table: **Sleep**

Column Name	Description	Data Type	Size	Format	Key
sleepID	Unique sleep ID linked to a user	Text	11		PK

hoursSlept	Hours slept by a user on a given date	Text	2	11	
dateMon	Date's month on a given night of sleep	Date	2	11	
dateDay	Date's day on a given night of sleep	Date	2	11	
dateYear	Date's year on a given night of sleep	Date	4	1111	
idUserState	Unique number indicating a specific user	Text	11		FK (ref UserState)

Table: **SubPlan**

Column Name	Description	Data Type	Size	Format	Key
subType	Type of subscription plan held by user	Text	11		PK
subName	Name of subscription plan	Text	11		
subPrice	Price of subscription plan	Text	5	\$111.11	
subDescription	Description of subscription plan	Text	65		

Table: **Subscription**

Column Name	Description	Data Type	Size	Format	Key
subID	Subscription held by a user	Text	11		PK
startMonth	Starting month of a user's subscription	Date	2	11	
startYear	Starting year of a user's subscription	Date	4	1111	

endMonth	Final month of a user's subscription	Date	2		
endYear	Final year of a user's subscription	Date	4		
subType	Type of subscription plan held by user	Text	11		FK (ref SubPlan)

Table: **UserProfile**

Column Name	Description	Data Type	Size	Format	Key
userID	Unique number linking a user's personal information to their account	Text	11		PK
firstName	User's first name	Text	45		
lastName	User's last name	Text	45		
email	User's email account	Text	45		
phone	User's phone number	Text	10	(111) 111-1111	
paymentID	Unique number relating to a unique monetary transactions	Text	11		FK (ref PaymentInfo)
subID	Subscription held by a user	Text	11		FK (ref Subscription)

Table: **UserState**

Column Name	Description	Data Type	Size	Format	Key
idUserState	Unique number indicating a specific user	Text	11		PK
workoutsID	Unique number indicating a user's	Text	11		FK (ref Workouts)

	workout on a given date				
idNutritionID	Unique number of individual's nutrition levels	Text	11		FK (ref Nutrition)
planID	Unique sequential number indicating the diet plan of the user	Text	11		FK (ref DietPlan)
idWorkoutPlan	Unique number indicating a user's workout routine				
userID					

Table: **WorkoutPlan**

Column Name	Description	Data Type	Size	Format	Key
idWorkoutPlan	Unique number indicating a user's workout routine	Text	11		PK
goalCaloriesBurned	User's goal calories to burn	Text	11		
goalMonth	Monthly calories to burn in order to obtain desired result	Date	2	11	
goalYear	Yearly calories to burn in order to obtain desired result	Date	4	1111	
idUserState	Unique number indicating a specific user	Text	11		FK (ref UserState)

Table: **Workouts**

Column Name	Description	Data Type	Size	Format	Key
-------------	-------------	-----------	------	--------	-----



workoutsID	Unique number indicating a certain workout routine	Text	11		PK
workoutType	Type of workout	Text	45		
durationsMins	Duration of workout in minutes	Numeri c	11		
wMonth	Date's month of given workout	Date	2	11	
wDay	Date's day of given workout	Date	2	11	
wYear	Date's year of given workout	Date	4	111	
idUserState	Unique number indicating a specific user	Text	11		FK (ref UserState)

## Queries

**Query 1:** Show total minutes of work outs for users who have worked out with the app in order of first name

### Query:

```
CREATE PROCEDURE totalMinutes()
SELECT UserProfile.firstName, UserProfile.Lastname, sum(Workouts.durationMins) AS "Total Workout Minutes"
FROM UserProfile
JOIN UserState ON UserProfile.userID = UserState.userID
JOIN Workouts ON Workouts.idUserState = UserState.idUserState
GROUP BY UserProfile.userID
ORDER BY UserProfile.firstName;
```

### Response:

	firstName	Lastname	Total Workout Minutes
►	Jane	Smith	60
	John	Johns	65
	Mary	Miller	70

**Query 2:** Show dates where users got less sleep than average user sleep

**Query:**

```
CREATE PROCEDURE sleeplessNights()
SELECT UserProfile.firstName, UserProfile.lastName, Sleep.dateMon AS "Month",
Sleep.dateDay AS "Day", Sleep.dateYear AS "YEAR", Sleep.hoursSlept AS "Hours Slept"
FROM Sleep
JOIN UserState ON UserState.idUserState = Sleep.idUserState
JOIN UserProfile ON UserProfile.userID = UserState.userID
WHERE Sleep.hoursSlept < (
    SELECT AVG(Sleep.hoursSlept)
    FROM Sleep);
```

**Response:**

	firstName	lastName	Month	Day	YEAR	Hours Slept
▶	Jane	Smith	12	23	2018	5.000

**Query 3:** Selects list of people who have a subscription

**Query:**

```
CREATE PROCEDURE hasSubscriptions()
SELECT Outter.firstName, Outter.lastName
FROM UserProfile AS Outter
WHERE EXISTS (
    SELECT Subscription.subID
    FROM Subscription
    WHERE Subscription.subID = Outter.userID);
```

**Response:**

	firstName	lastName
▶	Jane	Smith
	John	Johns
	Sam	Hunt
	Bill	Colin
	Erin	James
	Mary	Miller

**Query 4:** Shows count of all Free or Bronze subscriptions

**Query:**

```
CREATE PROCEDURE cheapSubscriptions()
SELECT SubPlan.subName, count(Subscription.subID)
FROM SubPlan
JOIN Subscription ON Subscription.subType = SubPlan.subType
GROUP BY SubPlan.subName HAVING SubPlan.subName = "Free" OR SubPlan.subName =
"Bronze";
```

**Response:**

	subName	count(Subscription.sub...
▶	Bronze	2
	Free	2

**Query 5:** Selects first and last names of profiles using a gmail account for their email (used to add in sign-in with google feature)

**Query:**

```
CREATE PROCEDURE gmailUsers()
SELECT UserProfile.firstName,UserProfile.lastName
FROM UserProfile
WHERE UserProfile.email RegExp 'gmail.com$';
```

**Response:**

	firstName	lastName
▶	Jane	Smith
	John	Johns
	Mary	Miller
	Sam	Hunt
	Bill	Colin
	Erin	James

**Query 6:** Returns the count of subscriptions created since 2015: useful for viewing growth statistics.

**Query:**

```
CREATE PROCEDURE recentSubscriptions()  
SELECT count(Subscription.subID) AS "New Subscriptions"  
FROM Subscription  
WHERE Subscription.startYear NOT IN (2012,2013,2014);
```

**Response:**

New Subscriptions	
▶	3

**Query 7:** Displays first and last name and contact info for all users.

**Query:**

```
CREATE PROCEDURE contactInfo()  
SELECT firstName, lastName, email, phone  
FROM UserProfile;
```

**Response:**

	firstName	lastName	email	phone
▶	Jane	Smith	jsmith@gmail.com	-5555
	John	Johns	jjohns@gmail.com	-5556
	Mary	Miller	mmiller@gmail.com	-5557
	Sam	Hunt	shunt@gmail.com	-5558
	Bill	Colin	bcolin@gmail.com	-5559
	Erin	James	ejames@gmail.com	-5550

**Query 8:** Displays first and last names and average daily grams consumed of carbs, protein, and fat

**Query:**

```
CREATE PROCEDURE averageConsumption()
SELECT UserProfile.firstName, UserProfile.lastName, AVG(Nutrition.fatGrams) AS "Fat",
AVG(Nutrition.proteinGrams) AS "Protein", AVG(Nutrition.carbGrams) AS "Carbs"
FROM Nutrition
JOIN UserState ON Nutrition.idUserState = UserState.idUserState
JOIN UserProfile ON UserProfile.userID = UserState.userID
GROUP BY UserProfile.userID;
```

**Response:**

	firstName	lastName	Fat	Protein	Carbs
▶	Bill	Colin	50.0000	180.0000	245.0000
	Sam	Hunt	86.0000	78.0000	60.0000
	Mary	Miller	22.0000	32.0000	53.0000
	John	Johns	21.0000	24.0000	66.0000

**Query 9:** Selects firstname, lastname, of users with their goal calories to burn by the end of the listed goal month/goal year

**Query:**

```
CREATE PROCEDURE goalTimelines()
SELECT UserProfile.firstName, UserProfile.lastName, WorkoutPlan.goalCaloriesBurned,
WorkoutPlan.goalMonth, WorkoutPlan.goalYear
FROM UserProfile
JOIN UserState ON UserState.userID = UserProfile.userID
JOIN WorkoutPlan ON WorkoutPlan.idUserState = UserState.idUserState;
```

**Response:**

	firstName	lastName	goalCaloriesBurn...	goalMonth	goalYear
►	Mary	Miller	2100	11	2022
	John	Johns	2850	6	2020
	Bill	Colin	1950	3	2021

**Query 10:** Selects first and last names of all users with the card types stored for their payment info: If you change your card processing provider to one who doesn't support some card types, you can quickly notify users affected by the transition.

**Query:**

```
CREATE PROCEDURE userCardTypes()
SELECT UserProfile.firstName, UserProfile.Lastname, PaymentInfo.cardType
FROM UserProfile
JOIN PaymentInfo ON PaymentInfo.paymentID = UserProfile.paymentID;
```

**Response:**

	firstName	Lastname	cardType
►	Sam	Hunt	VISA
	Mary	Miller	MasterCard
	Jane	Smith	AMEX
	Erin	James	AMEX
	Bill	Colin	VISA
	John	Johns	Discover