Anthony Frusti
Noah Strong

Rogue Agents Global Gym

https://web.engr.oregonstate.edu/~stronoah/cs340/index.html

Feedback from Peers-part 1

Hello,

I think an exercise tracking app would be really useful!

- **Does the overview describe what problem is to be solved by a website with DB back end?**

Your overview describes the problem that it is trying to solve so that someone who goes to the gym can track their workouts over time so that they can make adjustments to a workout routine.

- **Does the overview list specific facts?**

There aren't really specific facts listed, though, and you've noted that the gym has $4 in annual sales, which doesn't seem like very much. I think having more numerical facts listed (i.e. how many patrons do they expect to have per day/week/month/year, how much equipment do they have, etc.) would give a clearer idea of your project.

- **Are at least four entities described, and does each one represent a single idea to be stored as a list?**

You have five entities listed, but your calories' entity only has a 1:1 relationship with workouts. It may be better to just put all the attributes in calories under the workout entity, as I don't know that there is a need for the calorie entity. I think only in rare instances is a 1:1 relationship necessary.

- **Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints and describe relationships between entities?**

You describe the purpose of each of your entities and the list attribute datatype. However, I'm not sure if a bool datatype for Equipid attribute under equipment is the best option; perhaps int or varchar?

In regards to the relationships, it may help to also list all of the relationships in the exercise entity that it has with the other entities.

Anthony Frusti
Noah Strong

- **Are 1:M relationships correctly formulated? Is there at least one M:M relationship? Does the ERD present a logical view of the database?**

I believe that your 1:M relationships are correctly formulated and you have at least one M:M (between bodyparts and exercise and between exercise and equipment).

Your ERD presents a logical view of your database. I like that in your ERD that you noted that the relationship between exercise and equipment is optional (as your example, for pushups).

- **Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?**

In the ER Model exploration in Module 2, it's recommended that entities are plural since they are a list, so that may be helpful for your entities. All of your attributes are singular which is recommended, but I believe they should be snake or camel case, so for example your Workid attribute could be workID or work_id, as long as it's consistent throughout. I think that the entities can be capitalized and all of your entities are.

Great job and I look forward to your project!

Best,

Vanessa Dowd

Great work group 5! Your database design looks good so far. Here are some notes I have: - your attributes should have consistent capitalization/undercase (ex. body_part_name vs Category) I suggest sticking to all lowercase for attributes. You also should stick to either camelCase or snake_case for consistency - Similarly your entity names should be consistent in pluralization (Exercise vs BodyParts) - I'm wondering if the Calories entity is actually necessary or would it be enough to include it as an attribute in Workouts

Katie Russell

Actions and Changes based on the Feedback

Some of the things that we have changed in this report has been based on peer review feedback and our own reflection on how we could improve our draft. In the overview, we have added more detail and other specifications that we saw were lacking, and our peers agreed. In the Data Outline, we have reduced our entity count by 1 to add an attribute to one of them instead, specifically adding calories to Workouts. We have redone the Relationships tab, settled on data types after some input from our peers, and updated it based on the loss of the calorie entity. We changed our ER diagram to reflect these adjustments. The naming conventions for our entities and attributes also needed some work. We updated our entities to have pluralized names with camel case and our attribute names to follow snake case.

Upgrades to the Draft version

We made a small change to the ER diagram by adding an M:M between the BodyParts entity and the Workouts entity. We originally included the workout_id attribute as a part of the BodyParts entity, but we did not establish a relationship between the two. We felt it was necessary to show that a body part can be used in many workouts, and many workouts can be used for many different body parts.

Feedback from Peers -Part 2

Good work group 5. Some notes: - An M:M relationship is facilitated by intersection table with the two ids from each entity. Your Exercises to Equipment needs an intersection table such as Exercises_has_Equipment with the FKs equipmentID and exerciseID. Because we have these ids in the intersection table to facilitate the M:M relationship, you should not have the equipment id in exercises nor the exerciseid in equipment - you have a calId in your ERD which I believe should be renamed to calorie_count according to your outline - your schema should be a visualization of how your database is formed and how each relationship is related to each other. You can easily do so through phpmyadmin by clicking designer in the navigation bar at the top, you can also check this out in week 5: ERD vs Schema

Katie Russell, May 7 at 10:09pm

Actions and Changes based on the Feedback

The first thing that was changed with this newest draft was that we took the advice of our Ta and created new table to host the many to many relationship that would satisfy the conditions set by the new normal forms. So the Exercises_has_equipment was born to satisfy this need. The calorie_count mistake was fixed, we changed the foreign key commit problem that we saw in some of our reviews in the ed discussion that had FK problems. We have created a new schema for the document rather than rely on the database to do itle

Feedback from Peers -Part 3

Good work group 5. A few notes: - your schema is looking a bit better, but there are still a few problems. Exercises should not have the id EquipmentId as it is in the intersection table to facilitate your M:M relationship. Your ERD also has the wrong notation for the M:M relationship, right now you are showing two 1:1s. instead it should look like Exercises --< Exercises_has_equipment >-- Equipment. Exercises should also have the Workout id as an FK to facilitate the 1:M relationship, - you need to also include your intersection table as a separate page on your website - FKs should be user friendly, meaning that instead of showing their id you should be showing a name. ex. in Categories, instead of id 1 for exercise, use the name "bench press" If you need any more clarification or help please feel free to drop by my office hours or send me a message/email
- you should also have an insert on each page, these inserts should have all the attributes in each entity ex. for category, it should include values for body_part_name and exercise id where exercise id is ideally a drop down menu - also now that you no longer have the body entity, you need to get rid of the bodyID and just include bodypartname

Actions and Changes based on the Feedback

We have reviewed everything that the TA has noted and attempted to fully submit something that is within expectations for this assignment

Feedback from Peers -Part 4

Hello, Noah and Anthony!

- Do the implemented CRUD steps function as the team expects (e.g. if the team stated that a CRUD step worked but you found an error, please tell them)?
  - No, CUD out of the CRUD steps did not function. Specifically for the Equipment entity, which was excepted to work. I tried Insertion, Update and Deletion under Exercises_have_Equipment and nothing happened.
- Would a user easily be able to use the UI to complete the step? If not or you have suggestions for how the UI can be improved, please elaborate.
  - I do think the pages are easy to navigate and clear. My only suggestion would be to using a drop-down box option for the exercise name and exercise equipment that would make it easier to update and delete.

Anthony Frusti
Noah Strong

- What suggestions do you have for the team in any areas where they are blocked or having difficulty? Detailed helpful feedback will receive higher credit. If the team is not blocked or having difficulty encouraging and supportive comments would be a better response than NO feedback.
  - I would recommend to look at https://github.com/osu-cs340-ecampus/nodejs-starter-app, this helped my group implement Create, Read, Delete with a few adjustments depending on the entity. We have not successfully been able to Update using this, but we are getting close to making it work. Looking at your zip file, if you follow the guide that I linked it will also change the structure of your folders. It also looks like you have two different index.js, and I am not sure if that might cause an issue for you in the future. I can definitely see where the vision is for this project and best of luck implementing the CRUD, I know it's challenging.

Actions and Changes based on the Feedback

After receiving this feedback, we were dismayed but encouraged to have a start and have adjusted our code to get one our code working before the rest and we think we have succeeded this time with one functionality. However we realized that is not up to standard and we are working on rectifying that by scheduling overtime after this as we have both been incredibly busy this term. We will be working to update it to the most implementation standard and to start work on full working project

1)Overview

Rogue Agents Global Gym (RAGG) is a startup in the fitness industry that has made significant strides with approximately $4 million in annual sales. To further its success, the company is launching a new exercise web app that can record workout data by logging Calories, Exercise, BodyParts, and Equipment to Workouts. The app's database stores all workout data, making it simple to track progress over time and modify routines as necessary.

With its user-friendly interface and scalable design, the app is expected to attract at least 4,000 users by the end of the year. Once the project passes its initial release, it has the potential for even further expansion. RAGG aims to revolutionize the fitness industry by making tracking fitness progress and achieving fitness goals more accessible than ever before.

The ease-of-use is expected to be a major factor in the app's success, as it encourages more people to use it and recommend it to their friends and family. The scalability of the app is also a key feature, allowing it to easily handle numerous users without experiencing performance issues. The launch of this new exercise web app is a promising development for RAGG and the fitness industry as a whole, with its user-friendly interface, scalable design, and powerful database, it has the potential to transform the way people track their fitness, progress and achieve their fitness goals.

2)Database Outline

- Workouts: Details the various list of exercises (prebuilt or customized)
    - workout_id: int, auto-increment, unique, NOT NULL, PK
    - workout_name: varchar(50), NOT NULL
    - calorie_count: "BIGINT UNSIGNED

      Relationships:
        - 1:M with Exercises using workout _id
- Exercises: Details the specific movements required to complete the exercise
    - exercise_id: varchar(50), auto-increment ,unique, NOT NULL, PK
    - workout_id: int, FK
    - exercise_name: varchar(50), NOT NULL
    - equipment_id: varchar(50), NOT NULL,FK

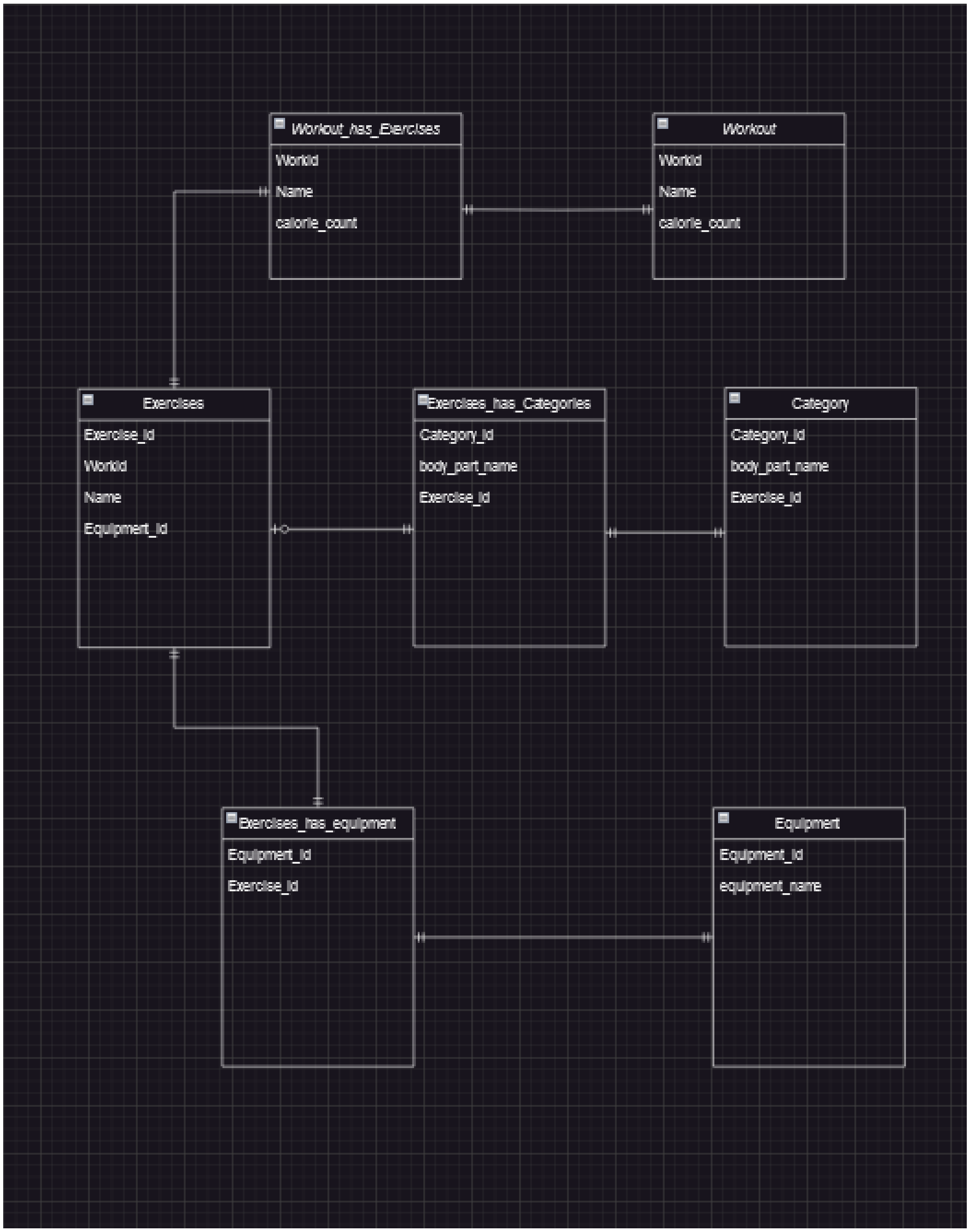
      Relationships
        - 1:M with Workouts using workout_id
        - 1:M with Exercises_has_Categories using category_id
        - 1:M relationship with Exercise_has_Equipment using equipment_id
- Category: Which body parts and muscles are going to be needed and worked on
    - category_id: int, NOT NULL, AUTO_INCREMENT,PK
    - body_part_id: varchar(50), auto-increment, NOT NULL
    - body_part_names: varchar(50), NOT NULL
    - exercise_id: int, FK

      Relationships:
        - 1:M with Exercises_has_Categories using exercise_id
- Exercise_has_Equipment:To test whether the exercise needs equipment

    - equipment_id: varchar(50),FK
    - exercise_id: varchar(50),FK
    - **Relationships**:
        - 1:M with Equipment using equipment_id
        - 1:M with Exercise using exercise_id
- Exercises_has_Categories: to categorize the exercises
    - exercise_id: varchar(50), FK
    - category_id: int, FK
    - **Relationships**:
        - 1:M with Exercise using exercise_id
        - 1:M with Category using category_id
- Equipment: what types of equipment is necessary(if at all(think pushups))
    - equipment_id: varchar(50), auto-increment, unique, NOT NULL, PK
    - equipment_name: varchar(50), NOT NULL
    - **Relationships**:
        - 1:M with Exercise_has_Equipment using equipment_id

3)Entity-Relationship Diagram(ER)

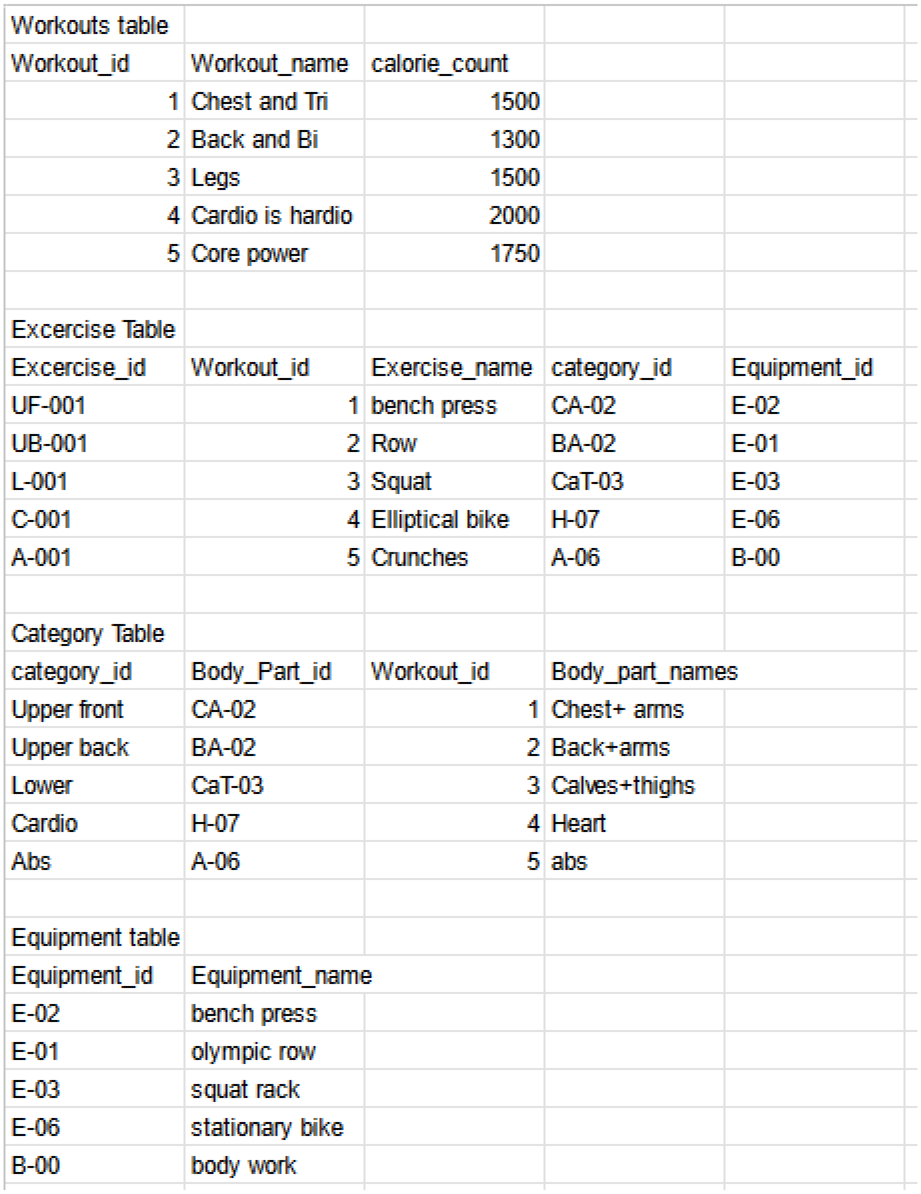
Anthony Frusti
Noah Strong



4)Schema

A)First to prove that the 1st normal form completion here is the table with some values that would be present at the beginning of the system

Anthony Frusti
Noah Strong

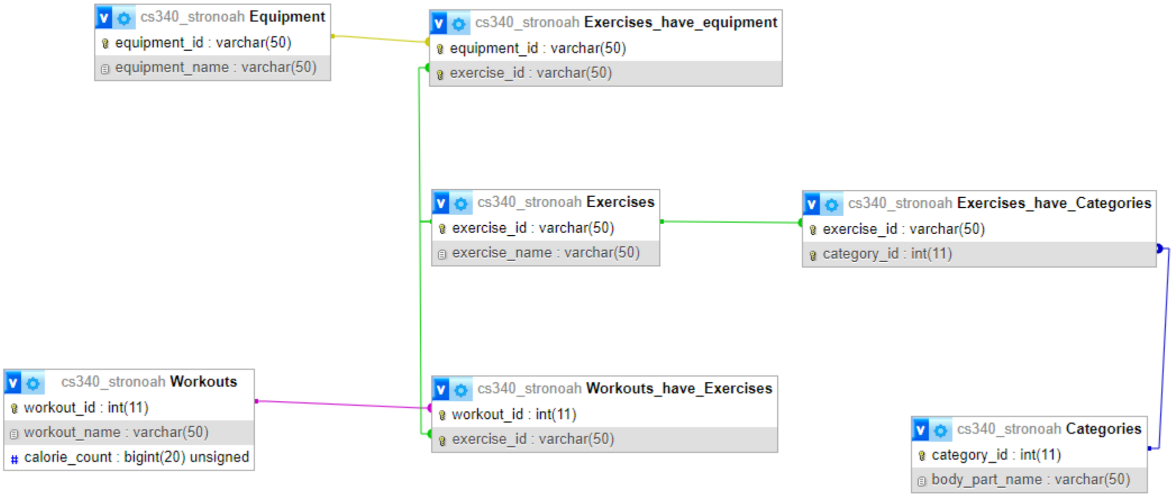


Workouts table				
Workout_id	Workout_name	calorie_count		
1	Chest and Tri	1500		
2	Back and Bi	1300		
3	Legs	1500		
4	Cardio is hardio	2000		
5	Core power	1750		
Excercise Table				
Excercise_id	Workout_id	Exercise_name	category_id	Equipment_id
UF-001	1	bench press	CA-02	E-02
UB-001	2	Row	BA-02	E-01
L-001	3	Squat	CaT-03	E-03
C-001	4	Elliptical bike	H-07	E-06
A-001	5	Crunches	A-06	B-00
Category Table				
category_id	Body_Part_id	Workout_id	Body_part_names	
Upper front	CA-02	1	Chest+ arms	
Upper back	BA-02	2	Back+arms	
Lower	CaT-03	3	Calves+thighs	
Cardio	H-07	4	Heart	
Abs	A-06	5	abs	
Equipment table				
Equipment_id	Equipment_name			
E-02	bench press			
E-01	olympic row			
E-03	squat rack			
E-06	stationary bike			
B-00	body work			

B)The changes that we have made in this instance is to completely change the body parts table to a category table to solve a redundancy problem. It fits more in the definition of body parts than in the description of the workout

To show the Second normal form completion, we have this table that shows that we have made the necessary foreign key changes

Anthony Frusti
Noah Strong



cs340_stronoah Equipment
equipment_id : varchar(50)
equipment_name : varchar(50)

cs340_stronoah Exercises_have_equipment
equipment_id : varchar(50)
exercise_id : varchar(50)

cs340_stronoah Exercises
exercise_id : varchar(50)
exercise_name : varchar(50)

cs340_stronoah Exercises_have_Categories
exercise_id : varchar(50)
category_id : int(11)

cs340_stronoah Workouts
workout_id : int(11)
workout_name : varchar(50)
calorie_count : bigint(20) unsigned

cs340_stronoah Workouts_have_Exercises
workout_id : int(11)
exercise_id : varchar(50)

cs340_stronoah Categories
category_id : int(11)
body_part_name : varchar(50)

B) The changes made to eliminate the partial dependencies was to cut category from exercise as one exercise does not have many categories, but there are many categories of exercise which is where our mistake in dependency lay