DietBase - Personal Diet and Nutrition Tracker

By: Zach Angha

Student ID: 922988765

GitHub Username: zachangha

Version History:

Checkpoint#	Date Submitted
Checkpoint I	02/20/24
Checkpoint II	3/5/24

Table of Contents

Product Description

Ever since I have been going to the gym, I have found using a personal diet and nutrition tracker app to be very beneficial to make sure I am getting all the right nutrients. Since I am lifting multiple times a week and looking to gain muscle, I have tried to make sure I am in a caloric surplus and eating a high-protein diet. On the other hand, I have many friends who use these apps for the opposite reasons to lose weight, meaning they want to be in a caloric deficit. This shows that these apps can be used for both people trying to bulk up and people who are aiming to lose weight. As someone who has been using many nutrition-tracking apps for the past couple of years, I understand all the standard features that come with them, but all pretty much have the same functionality and features. For these companies to separate themselves from the competition, they could look to implement a new database system that can open the gate for new features that would attract more consumers.

The DietBase database system is perfect for helping people with their busy lives, assisting them plan meals or to just track what they have available to them, while offering the same features offered by any other diet or nutrition tracker. With the current dieting, tracking calories and nutrients has become very easy, but what if you want more precise tracking where you can input what you buy from the store like a virtual fridge or pantry? DietBase will allow more precise tracking of food by implementing a tracking table where we will store all the foods and ingredients that people buy from the store collecting what groceries they currently have all in one place. We offer this database to dieting software and apps that want to add new and innovative features for their users, to improve their customer satisfaction, saving them time and energy. What separates DietBase from other dieting databases is that we will allow users to input and track what they have in their fridge or pantry for possible recipes, track freshness dates, or just take inventory of what they currently have. Features like alerts of when you run out of an ingredient and adding it directly to your shopping list, notifying you of expired foods, or recommending recipes that you can make with what you have can all be added with this new tracker. Existing software like MyFitnessPal and WeightWatchers would greatly benefit from this feature because they'd be able to offer their customers something that would make their apps and systems more personal, allowing them to track their groceries through the app.

Use Cases:

1. **Use Case:** Food Waste Reduction

   Actor: Environmentalist (Steven)

   Description: Steven is very concerned about the environment and reducing food waste is one of the many ways for him to do his part in saving it. Steven will use anything and everything he buys to prevent the waste in resources, food, and money. Steven wants to have a way to track the expiration date of all the foods he buys from the grocery store, so he can be sure nothing he buys goes to waste in his house.

   DietBase is the solution to his problem as it has tables made to track all the foods you buy from the store and all the expiration dates, so you can view how close the food gets to losing its freshness. Apps could implement features with this database that will let Steven view expiration dates and be notified when his food is getting close to becoming expired.

2. **Use Case:** Limited Meals

   Actor: Late Night Worker (Josh)

   Description: Josh works late shifts at his job and when he gets home, he just wants to eat something before he goes to sleep. All the restaurants near Josh are closed by the time he gets off, so he's limited to what he has at home. Sometimes Josh doesn't have many ingredients and is restricted by what he must work with.

   DietBase can help him find recipes that fit within his constrictions. The DietBase database system saves the ingredients in recipes as singular food entities, allowing apps to develop a system that will query and find recipes that he can make with what he has available.

3. **Use Case:** Grocery List

   Actor: Full-time Student (Jessica)

   Description: Jessica is a full-time student and has class almost every day of the week. She doesn't have convenient access to a grocery store, so she must make sure she gets everything she needs, the one time she goes a week.

   The DietBase system is a solution to this problem as there is a built-in grocery list table. Developers could implement the feature for Jessica to let her create her own grocery list in the app, so she never forgets anything again.

Functional Database Requirements

1. User: Strong
    1.1. A user shall create one account.
    1.2. A user shall login to many devices.
2. Devices: Strong
    2.1. A device shall log into many users.
3. Account: Weak
    3.1. An account shall be created by at only one user.
    3.2. An account shall be a general user or an admin.
    3.3. An account shall have many allergies and restrictions.
    3.4. An account shall select one time zone at a time.
    3.5. An account shall select one location at a time.
    3.6. An account shall select one and only one measurement system.
    3.7. An account shall save at least one measurement.
    3.8. An account shall have at least one long-term goal.
    3.9. An account shall have only one set of nutritional goals.
    3.10. An account shall input at least one weight log.
    3.11. An account shall input many fasting logs.
    3.12. An account shall input many diary entries.
    3.13. An account shall input many water intakes.
    3.14. An account shall have many step counters.
    3.15. An account shall input many sleep logs.
    3.16. An account shall create many recipes, foods, medications, and/or supplements.
    3.17. An account shall input many recipes, foods, medications, and/or supplements.
    3.18. An account shall save many recipes.
    3.19. An account shall create many exercises.
    3.20. An account shall input many exercises.
    3.21. An account shall have many meal plans.
    3.22. An account shall have many exercise plans.

3.23. An account shall make many inputs to their grocery list.

3.24. An account shall make many inputs to their food inventory.

3.25. An account shall add many friends with other accounts.

3.26. An account shall create many groups.

3.27. An account shall join many groups.

3.28. An account shall create many posts.

3.29. An account shall create many comments.

3.30. An account shall upload multiple multimedia's.

3.31. An account shall be sent many notifications.

3.32. An account shall accept many food challenges.

3.33. An account shall accept many exercise challenges.

3.34. An account shall view many newsfeeds.

4. General User: Weak

4.1. An account shall be created by at only one user.

4.2. An account shall be a general user or an admin.

4.3. An account shall have many allergies and restrictions.

4.4. An account shall select one time zone at a time.

4.5. An account shall select one location at a time.

4.6. An account shall select one and only one measurement system.

4.7. An account shall save at least one measurement.

4.8. An account shall have at least one long-term goal.

4.9. An account shall have only one set of nutritional goals.

4.10. An account shall input at least one weight log.

4.11. An account shall input many fasting logs.

4.12. An account shall input many diary entries.

4.13. An account shall input many water intakes.

4.14. An account shall have many step counters.

4.15. An account shall input many sleep logs.

4.16. An account shall create many recipes, foods, medications, and/or supplements.

4.17. An account shall input many recipes, foods, medications, and/or supplements.

4.18. An account shall save many recipes.

4.19. An account shall create many exercises.

4.20. An account shall input many exercises.

4.21. An account shall have many meal plans.

4.22. An account shall have many exercise plans.

4.23. An account shall make many inputs to their grocery list.

4.24. An account shall make many inputs to their food inventory.

4.25. An account shall add many friends with other accounts.

4.26. An account shall create many groups.

4.27. An account shall join many groups.

4.28. An account shall create many posts.

4.29. An account shall create many comments.

4.30. An account shall upload multiple multimedia's.

4.31. An account shall be sent many notifications.

4.32. An account shall accept many food challenges.

4.33. An account shall accept many exercise challenges.

4.34. An account shall view many newsfeeds.

5. Admin: Weak

5.1. An admin shall do everything a general user can do.

5.2. An admin shall delete posts.

5.3. An admin shall delete comments.

5.4. An admin shall delete accounts.

5.5. An admin shall delete recipes.

5.6. An admin shall delete foods.

6. Allergies and Restrictions: Strong

6.1. An allergy and/or restriction shall be assigned to many accounts.

7. Time Zones: Strong

7.1. A time zone shall be selected by many accounts.

8. Locations: Strong
    8.1. A location shall be selected by many accounts.
9. Measurement System: Strong
    9.1. A measurement system shall be selected by many accounts.
10. Measurements: Weak
    10.1. A measurement shall be assigned to only one account.
11. Long-Term Goals: Weak
    11.1. A long-term shall be assigned to only one account.
12. Nutritional Goals: Weak
    12.1. A set of nutritional goals shall be assigned to only one account.
13. Weight Log: Weak
    13.1. A weight log shall be input by only one account.
14. Fasting Log: Weak
    14.1. A fasting log shall be input by only one account.
15. Diary Entry: Weak
    15.1. A diary entry shall be input by only one account.
16. Water Intake: Weak
    16.1. A water intake shall be input by only one account.
17. Step Counter: Weak
    17.1. A step counter shall be assigned to only one account.
18. Sleep Log: Weak
    18.1. A sleep log shall be input by only one account.
19. Food: Strong
    19.1. A food shall be created by at most one account.
    19.2. A food shall be input by many accounts.
    19.3. A food shall have at most one complete nutritional information.
    19.4. A food shall be used in many recipes.
    19.5. A food shall be in many grocery lists.
    19.6. A food shall be in many food inventories.

19.7. A food shall be assigned many food types.

20. Medications: Strong

20.1. A medication shall be created by at most one account.

20.2. A medication shall have at most one complete nutritional information.

20.3. A medication shall be input by many accounts.

21. Supplements: Strong

21.1. A supplement shall be created by at most one account.

21.2. A supplement shall be input by many accounts.

21.3. A supplement shall have at most one complete nutritional information.

22. Recipes: Weak

22.1. A recipe shall have at least one food.

22.2. A recipe shall be created by at most one account.

22.3. A recipe shall be input by many accounts.

22.4. A recipe shall have at most one nutritional information.

22.5. A recipe shall be saved by many accounts.

22.6. A recipe shall be used in many meal plans.

22.7. A recipe shall be created by at most one account.

22.8. A recipe shall be assigned many meal categories.

22.9. A recipe shall be assigned many meal types.

23. Exercises: Strong

23.1. An exercise shall be created by at most one account.

23.2. An exercise shall be input by many accounts.

23.3. An exercise shall be put in many exercise plans.

24. Meal Plans: Weak

24.1. A meal plan shall have many recipes.

24.2. A meal plan shall be assigned to many accounts.

24.3. A meal plan shall be assigned many meal categories.

25. Exercise Plans: Weak

25.1. An exercise plan shall be assigned to many accounts.

25.2. An exercise plan shall hold many exercises.

26. Groups: Strong

26.1. A group shall be created by at most one account.

26.2. A group shall have many accounts.

27. Posts: Strong

27.1. A post shall be created by at most one account.

27.2. A post shall have many comments.

27.3. A post shall have at most one multimedia.

27.4. A post shall be on many newsfeeds.

28. Comments: Weak

28.1. A comment shall be assigned to only one post by only one account.

29. Multimedia: Strong

29.1. A multimedia shall be uploaded by at most one account.

29.2. A multimedia shall be on at most one post.

30. Notification: Weak

30.1. A notification shall be sent to an account.

31. Food Challenges: Strong

31.1. A food challenge shall be accepted by many accounts.

32. Exercise Challenges: Strong

32.1. An exercise challenge shall be accepted by many accounts.

33. Nutritional Information: Weak

33.1. A nutritional information shall be assigned to only one food, recipe, supplement, or medication.

34. Meal Category: Strong

34.1. A meal category shall be assigned to many recipes.

34.2. A meal category shall be assigned to many meal plans.

35. Food type: Strong

35.1. A food type shall be assigned to many foods.

36. Newsfeed: Strong

36.1. A newsfeed shall be viewed by many accounts.

36.2. A newsfeed shall have many posts.

Non-functional Requirements

1. Performance
    1.1. The database shall support concurrent inputs.
    1.2. The database shall respond to the user's input in a reasonable amount of time.
    1.3. The database shall be optimized to find the needed information in a reasonable amount of time.
    1.4. The database shall take in the user's information in a reasonable amount of time.
    1.5. The database shall handle concurrent user activity without having a significant impact on the performance.
2. Security
    2.1. The database shall protect the user's measurements from other people.
    2.2. The database shall only store encrypted passwords.
    2.3. The database shall be backed up every day at 11:59 pm.
    2.4. The database shall only insert values that are consistent with the attribute's datatype and domain.
    2.5. Administrative functions and commands are only given to those that are authorized to do so.
3. Scalability
    3.1. Regardless of the number of users the database shall function as expected.
    3.2. The performance of the database shall remain consistent as the number of entries grows.
    3.3. The database shall allow an increase of more servers to distribute the workload as the user base grows.
    3.4. The database shall be able to expand with new added features.
    3.5. As the database grows the amount of memory allocated to the tables should grow with it.
4. Capability
    4.1. The database shall support different data types.
    4.2. The database shall query to find the needed information efficiently.

4.3. The database shall support multiple languages.

4.4. Applications should be able to integrate the database system.

4.5. The database shall work with many computer and server setups.

5. Environmental

5.1. The database shall run in a reliable space.

5.2. The database shall optimize its performance when it can.

5.3. The database shall run as a parallel database.

5.4. The data center locations shall be set up in areas with high connectivity outreach.

5.5. The database shall be available around the world.

6. Coding Standards

6.1. A coding style and format shall be used across the entire system.

6.2. Code shall be reusable when it can be to reduce redundancy.

6.3. Code will have comments to explain.

6.4. All errors will be documented and handled.

6.5. All code will be reviewed to see if it’s up to coding standards.

7. Media Storage

7.1. Different form of media shall be stored on the database.

7.2. Different file types of media shall be stored on the database.

7.3. Media stored shall be compressed to reduce size.

7.4. All media shall have a 200 MB limit.

7.5. All media shall have submission dates stored with.

8. Privacy

8.1. All data stored will be in compliance with the user.

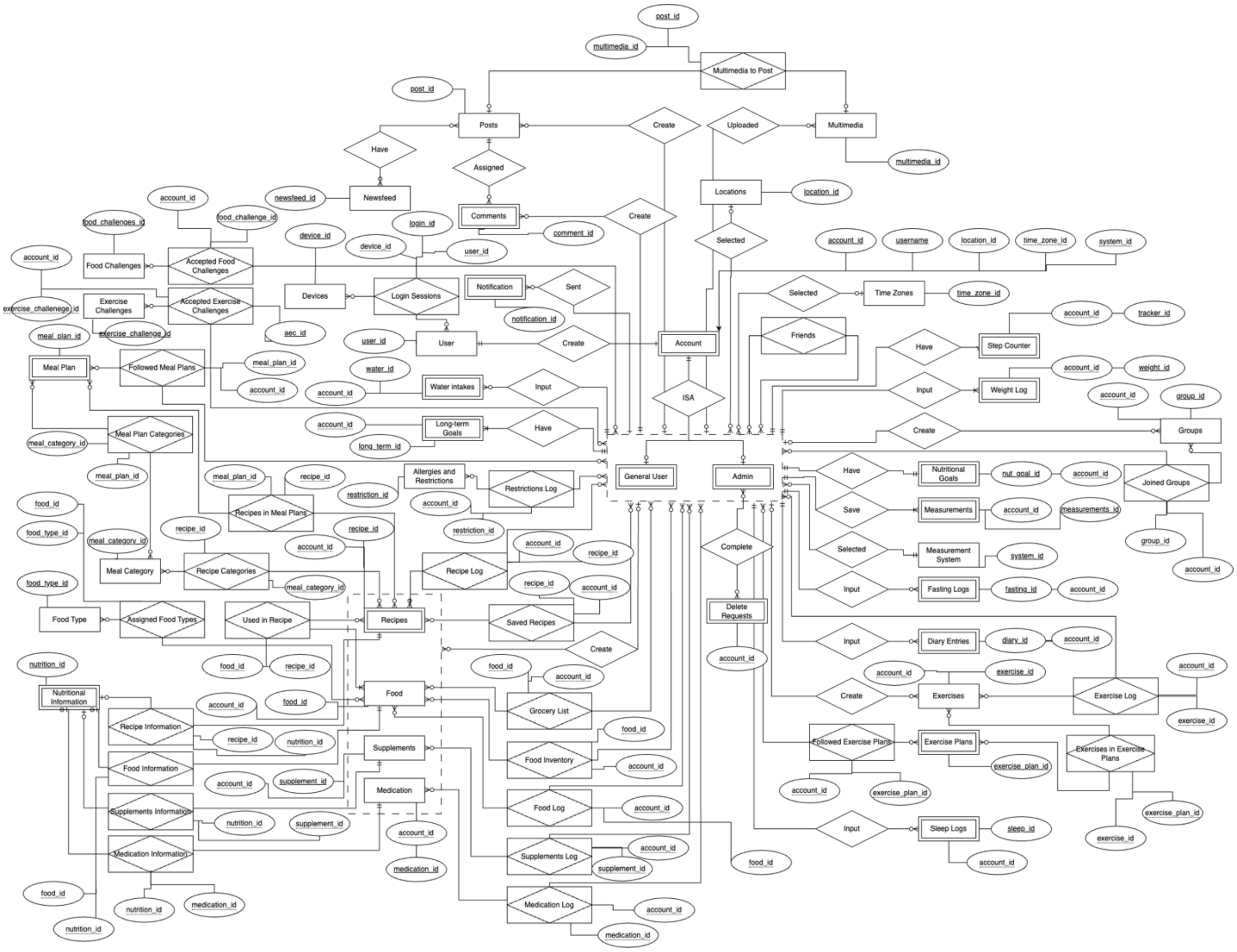
8.2. If requested from the user, an admin shall delete their data.

8.3. No user can access another user’s private information.

8.4. All data transferred to and from the database shall be encrypted.

8.5. Upon request of the user, their data can be sent to them.

Entity Relationship Diagram



Entity Description

1. User (Strong)
    - user_id: strong key, numeric
    - name: composite, alphanumeric
    - email: alphanumeric
    - password: alphanumeric
2. Devices (Strong)
    - device_id: strong key, numeric
    - device_type: alphanumeric
    - device_name: alphanumeric
3. Login Sessions
    - login_id: strong key, numeric
    - device_id: weak key, numeric
    - user_id: weak key, numeric
    - login_date: composite, numeric
    - logout_date: composite, numeric
4. Account (Weak)
    - account_id: strong key, numeric
    - username: alphanumeric
    - admin: boolean
    - dob: composite, timestamp
    - age: derived, numeric
    - date_joined: composite, timestamp
    - location_id: weak key, numeric
    - time_zone_id: weak key, numeric
5. General User (Weak)
    - account_id: strong key, numeric
    - username: alphanumeric
    - admin: boolean

- dob: composite, timestamp
- age: derived, numeric
- date_joined: composite, timestamp
- location_id: weak key, numeric
- time_zone_id: weak key, numeric

6. Admin (Weak)
    - account_id: strong key, numeric
    - username: alphanumeric
    - admin: boolean
    - dob: composite, timestamp
    - age: derived, numeric
    - date_joined: composite, timestamp
    - location_id: weak key, numeric
    - time_zone_id: weak key, numeric
7. Delete Requests
    - account_id: weak key, numeric
    - description: alphanumeric
    - completed: boolean
    - date_filed: composite, numeric
8. Allergies and Restrictions (Strong)
    - restriction_name: alphanumeric
    - restriction_id: strong key, numeric
    - restriction_type: alphanumeric
9. Restriction Log (Weak)
    - restriction_log_id: strong key, numeric
    - restriction_id: weak key, numeric
    - account_id: weak key, numeric
10. Time Zones (Strong)
    - time_zone_name: alphabetic

- ∗ time_zone_id: strong key, numeric
- ∗ uct_offset: numeric

11. Locations (Strong)
    - ∗ location_name: alphabetic
    - ∗ location_id: strong key, numeric
    - ∗ currency: alphanumeric
12. Measurement System (Strong)
    - ∗ system_name: alphabetic
    - ∗ system_id: strong key, numeric
13. Measurements (Weak)
    - ∗ measurements_id: strong key, numeric
    - ∗ account_id: weak key, numeric
    - ∗ height: numeric
    - ∗ starting_weight: numeric
    - ∗ gender: alphabetic
14. Long-Term Goals (Weak)
    - ∗ long_term_id: strong key, numeric
    - ∗ account_id: weak key, numeric
    - ∗ title: alphanumeric
    - ∗ description: alphanumeric
    - ∗ date_assigned: composite, timestamp
    - ∗ complete_by_date: composite, timestamp
    - ∗ completed: boolean
15. Nutritional Goals (Weak)
    - ∗ nut_goal_id: strong key, numeric
    - ∗ account_id: weak key, numerical
    - ∗ calories: numeric
    - ∗ macro_nutrients: composite, numeric
    - ∗ micro_nutrients: composite, numeric

16. Weight Log (Weak)
    - weight_id: strong key, numeric
    - account_id: key, numeric
    - weight: numerical
    - weight_log_date: composite, timestamp
17. Fasting Log (Weak)
    - fasting_id: strong key, numeric
    - account_id: key, numeric
    - time_length: composite, numeric
    - fasting_date: composite, numeric
18. Diary Entry (Weak)
    - diary_id: strong key, numeric
    - account_id: weak key, numeric
    - entry: alphabetic
    - diary_date: composite, numeric
19. Water Intake (Weak)
    - water_id: strong key, numeric
    - account_id: weak key, numeric
    - water_amount: numeric
    - water_intake_date: composite, numeric
20. Step counter (Weak)
    - counter_id: strong key, numeric
    - account_id: weak key, numeric
    - step_amount: numeric
    - step_counter_date: composite, numeric
21. Sleep Log (Weak)
    - sleep_id: strong key, numeric
    - account_id: weak key, numeric
    - sleep_amount: composite, numeric

    - sleep_date: composite, numeric

22. Food (Strong)
    - account_id: weak key, numeric
    - name: alphanumeric
    - food_id: strong key, numeric
23. Food Type (Strong)
    - food_type_id: strong key, numeric
    - food_type_name: alphanumeric
    - food_type_description: alphanumeric
24. Assigned Food Types (Weak)
    - food_id: weak key, numeric
    - food_type_id: weak key, numeric
    - date_added: composite, numeric
25. Used in Recipe (Weak)
    - food_id: weak key, numeric
    - recipe_id: weak key, numeric
    - date_added: compsite, numeric
26. Grocery List (Weak)
    - food_id: weak key, numeric
    - account_id: weak key, numeric
    - priority level: numeric
    - food_gotten: boolean
27. Food Inventory (Weak)
    - food_id: weak key, numeric
    - account_id: weak key, numeric
    - expiration_date: composite, numeric
    - food_eaten: boolean
28. Food Log (Weak)
    - account_id: weak key, numeric

- food_id: weak key, numeric
- date_entered, composite, numeric
- meal_type, alphabetic

29. Food Information (Weak)
    - food_id: weak key, numeric
    - nutrition_id: weak key, numeric
    - date_added: composite, numeric
30. Medication (Strong)
    - account_id: weak key, numeric
    - name: alphanumeric
    - medication_id: strong key, numeric
    - instructions: alphanumeric
31. Medication Log (Weak)
    - account_id: weak key, numeric
    - medication_id: weak key, numeric
    - date_entered, composite, numeric
    - meal_type, alphabetic
32. Medication Information (Weak)
    - medication_id: weak key, numeric
    - nutrition_id: weak key, numeric
    - date_added: composite, numeric
33. Supplements (Strong)
    - account_id: weak key, numeric
    - name: alphanumeric
    - supplement_id: strong key, numeric
34. Supplement Log (Weak)
    - account_id: weak key, numeric
    - supplement_id: weak key, numeric
    - date_entered, composite, numeric

- meal_type, alphabetic

35. Supplement Information (Weak)

- supplement_id: weak key, numeric
- nutrition_id: weak key, numeric
- date_added: composite, numeric

36. Recipe (Weak)

- account_id: weak key, numeric
- name: alphanumeric
- recipe_id: strong key, numeric
- directions: alphanumeric

37. Recipe Log (Weak)

- account_id: weak key, numeric
- recipe_id: weak key, numeric
- date_entered, composite, numeric
- meal_type, alphabetic

38. Recipe Information (Weak)

- recipe_id: weak key, numeric
- nutrition_id: weak key, numeric
- date_added: composite, numeric

39. Saved Recipes (Weak)

- recipe_id: weak key, numeric
- account_id: weak key, numeric
- date_saved: composite, numeric

40. Meal Plans (Weak)

- meal_plan_id: strong key, numeric
- meal_plan_name: alphanumeric
- meal_plan_description: alphanumeric
- meal_plan_length: numeric

41. Recipes in Meal Plans (Weak)

- ∗ meal_plan_id: weak key, numeric
- ∗ recipe_id: weak key, numeric
- ∗ date_added: composite, numeric

42. Followed Meal Plans (Weak)

- ∗ account_id: weak key, numeric
- ∗ meal_plan_id: weak key, numeric
- ∗ date_subscribed: composite, numeric

43. Meal Category (Strong)

- ∗ meal_category_id: strong key, numeric
- ∗ meal_category_name: alphanumeric
- ∗ meal_category_description: alphanumeric

44. Meal Plan Categories (Weak)

- ∗ meal_category_id: weak key, numeric
- ∗ meal_plan_id: weak key, numeric
- ∗ date_added: composite, numeric

45. Recipe Categories (Weak)

- ∗ meal_category_id: weak key, numeric
- ∗ recipe_id: weak key, numeric
- ∗ date_added: composite, numeric

46. Exercises (Strong)

- ∗ exercise_id: strong key, numeric
- ∗ account_id: weak key, numeric
- ∗ exercise_name: alphanumeric
- ∗ exercise_description: alphanumeric
- ∗ time_length: numeric

47. Exercise Log (Weak)

- ∗ account_id: weak key, numeric
- ∗ exercise_id: weak key, numeric
- ∗ date_entered: composite, numeric

48. Exercise Plans (Weak)
    - exercise_plan_id: weak key, numeric
    - exercise_plan_name: alphanumeric
    - exercise_plan_description: alphanumeric
    - exercise_plan_length: numeric
49. Exercises in Exercise Plans (Weak)
    - exercise_id: weak key, numeric
    - exercise_plan_id: weak key, numeric
    - date_added: composite, numeric
50. Followed Exercise Plans (Weak)
    - account_id: weak key, numeric
    - exercise_plan_id: weak key, numeric
    - date_subscribed: composite, numeric
51. Groups (Strong)
    - group_id: strong key, numeric
    - account_id: weak key, numeric
    - group_name: alphanumeric
    - group_description: alphanumeric
52. Joined Groups (Weak)
    - account_id: weak key, numeric
    - group_id: weak key, numeric
    - date_joined_group: composite, numeric
53. Posts (Strong)
    - account_id: weak key, numeric
    - post_id: strong key, numeric
    - post_title: alphanumeric
    - post_desctiption: alphanumeric
    - date_created: composite, numeric
54. Comments (Weak)

- comment_id: strong key, numeric
- account_id: weak key, numeric
- comment_content: alphanumeric
- date_commented: composite, numeric
- post_id: weak key, numeric

55. Multimedia (Strong)
    - file_name: alphanumeric
    - file_type: alphanumeric
    - multimedia_id: strong key, numeric
    - account_id: weak key, numeric
    - date_uploaded: composite, numeric
56. Multimedia to Post (Weak)
    - dimensions: composite, numeric
    - post_id: weak key, numeric
    - multimedia_id: weak key, numeric
57. Notification (Strong)
    - notification_id: strong key, numeric
    - account_id: weak key, numeric
    - notification_description: alphanumeric
58. Food Challenges (Strong)
    - food_challenge_id: strong key, numeric
    - food_challenge_name: alphanumeric
    - food_challenge_description: alphanumeric
    - food_challenge_length: numeric
59. Accepted Food Challenges (Weak)
    - food_challenge_id: weak key, numeric
    - account_id: weak key, numeric
    - date_started: composite, numeric
    - date_ended: composite, numeric

- completed: boolean

60. Exercise Challenges (Strong)
    - exercise_challenge_id: strong key, numeric
    - exercise_challenge_name: alphanumeric
    - exercise_challenge_description: alphanumeric
    - exercise_challenge_length: numeric
61. Accepted Exercise Challenges (Weak)
    - exercise_challenge_id: weak key, numeric
    - account_id: weak key, numeric
    - date_started: composite, numeric
    - date_ended: composite, numeric
    - completed: boolean
62. Nutritional Information (Weak)
    - nutrition_id: strong key, numeric
    - calories: numeric
    - macro_nutrients: composite, numeric
    - micro_nutrients: composite, numeric
63. Newsfeed (Strong)
    - newsfeed_id: strong key, numeric
    - newfeed_date: composite numeric
    - post_id: weak key, numeric