

a) Project Outline and Database Outline - Updated Version

i) Overview

WorkoutTracker is a web application allowing users to record their workouts and track their performance in a journal-like webapp. Users will be able to create a new workout entry everytime they workout, add in exercises performed (deadlift, bench press etc) and record specific notes for a given workout. Exercises can be grouped into different categories (strength, cardio, flexibility) and users can search through them based on those categories, or they can create new exercises.

One of the biggest thrills of working out is being able to see yourself improve, and WorkoutTracker helps you do just that. After signing up for a free account, users can immediately start recording workouts and tracking their progress. Users will be able to create an unlimited number of journal entries so even the most active users will never run out of space. After recording each workout, users will have their information all in one place where it's easy to view. Our database of exercises will keep their entries organized and easy to search.

Having a relational database backing this web application not only enables keeping records specific to a user, but also the normalization allows for data storage optimization. For example many users can create different exercises utilizing the "squat rack" equipment, but only one instance of that entity needs to exist in the database. With users potentially completing hundreds of workouts every year, the app would help them keep track of their progress in an easily accessible online database. Since the database is pre populated with lots of exercises, equipment and categories, it gives users a standardized way to record their workouts.

This design also allows for many different users to keep track of their data given how relationships scope information.

ii) Database Outline, in Words

- 1) **Users:** information about a specific user of the application
 - (a) userID: int, auto_increment, unique, not NULL, PK
 - (b) email: varchar, unique, not NULL
 - (c) firstName: varchar, not NULL
 - (d) lastName: varchar, not NULL

- (e) Relationship: a 1:M relationship between Users and Workouts is implemented with the User entity userID attribute as a foreign key attribute on the Workout entity. (a user can have many workouts, a workout belongs to one user).

2) Workouts: records information about a specific workout session of a user

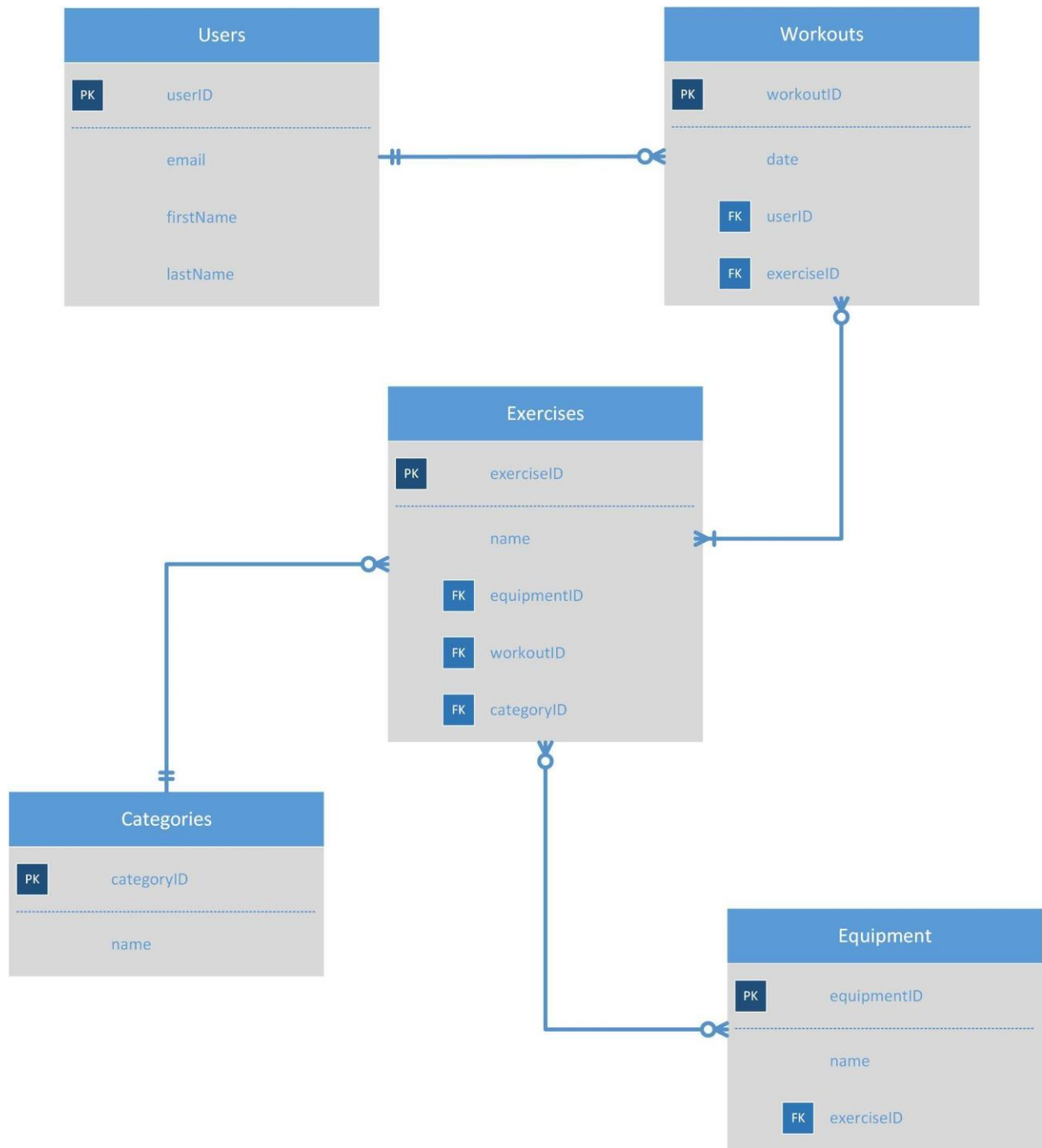
- (a) workoutID: int, auto_increment, unique, not NULL, PK
- (b) date: DATE, not NULL
- (c) Description: varchar
- (d) Relationship: a 1:M relationship between Users and Workouts is implemented with the User entity userID attribute as a foreign key attribute on the Workout entity. (a user can have many workouts, a workout belongs to one user).
- (e) Relationship: a M:M relationship between Workouts and Exercises is implemented with a join table with two columns; workoutID and exerciseID. The workoutID and exerciseID are each foreign keys to their respective entities, and together they form a composite primary key within the join table. (a workout can have many exercises, an exercise can be a part of many workouts)

3) Exercises: describes a specific activity that can be performed during a workout

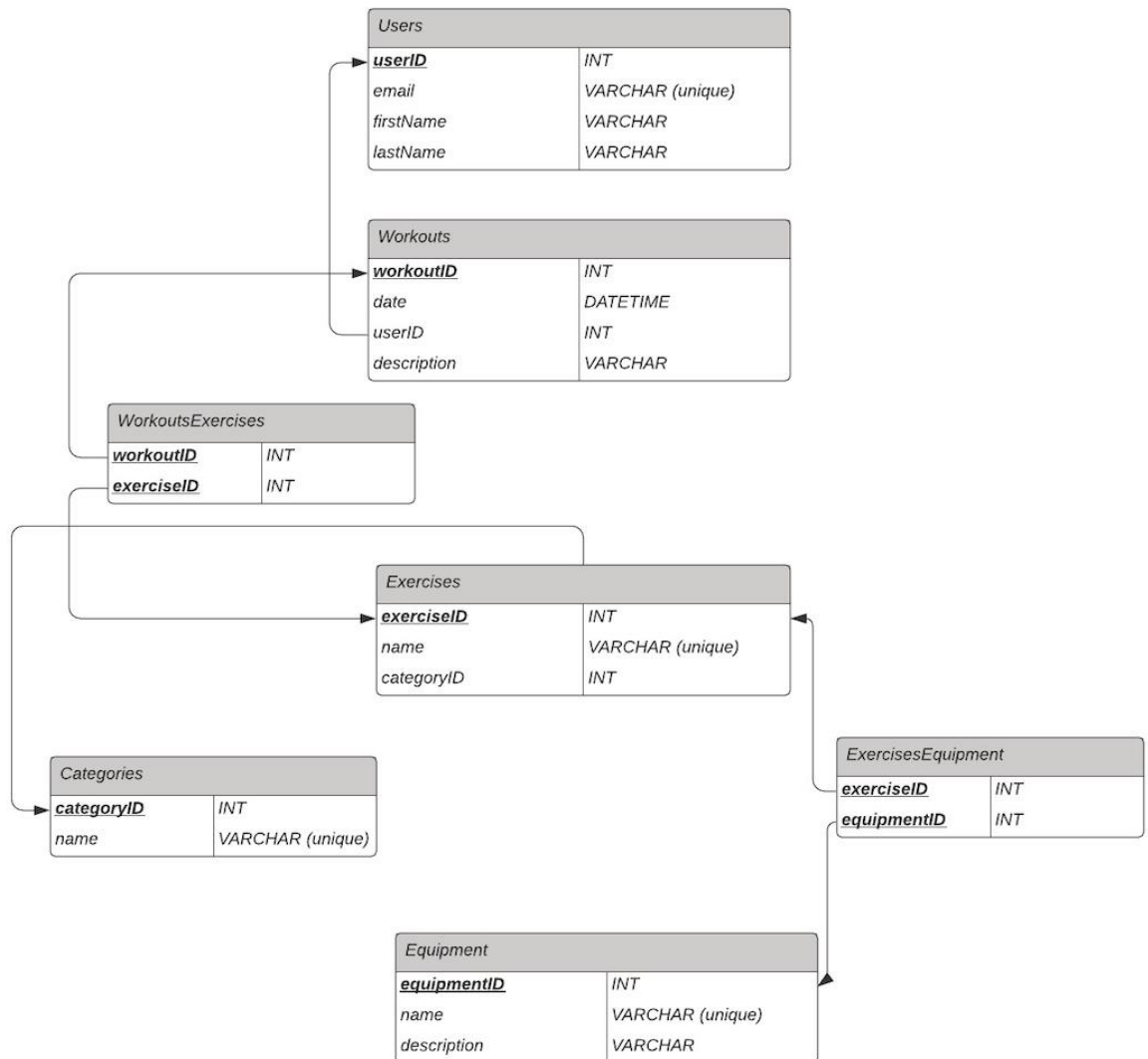
- (a) exerciseID: int, auto_increment, unique, not NULL, PK
- (b) name: varchar, unique, not NULL
- (c) Relationship: a M:M relationship between Workouts and Exercises is implemented with a join table with two columns; workoutID and exerciseID. The workoutID and exerciseID are each foreign keys to their respective entities, and together they form a composite primary key within the join table. (a workout can have many exercises, an exercise can be a part of many workouts)
- (d) Relationship: a M:M relationship between Exercises and Equipment is implemented with a join table with two columns; exerciseID and equipmentID. The exerciseID and equipmentID are each foreign keys to their respective entities, and together they form a composite primary key within the join table. (an exercise can have many different types of equipment {like bench and barbell} but may not be

- needed, equipment can be used for many different types of exercises but can exist without being a part of an exercise)
- (e) Relationship: a 1:M relationship between Exercises and Categories is implemented with the Category entity categoryID attribute as a foreign key attribute on the Exercise entity. (a category can have many exercises, an exercise belongs to one category).
- 4) **Equipment:** equipment used when performing an exercise (bench, squat rack, dumbbell etc.), but not always necessary (jogging outside, stretching etc.).
- (a) equipmentID: int, auto_increment, unique, not NULL, PK
 - (b) name: varchar, unique, not NULL
 - (c) Relationship: a M:M relationship between Exercises and Equipment is implemented with a join table with two columns; exerciseID and equipmentID. The exerciseID and equipmentID are each foreign keys to their respective entities, and together they form a composite primary key within the join table. (an exercise can have many different types of equipment {like bench and barbell} but may not be needed, equipment can be used for many different types of exercises but can exist without being a part of an exercise)
- 5) **Categories:** represents a grouping of exercises with similar movements or goals (cardio, weight training, stretching)
- (a) categoryID: int, auto_increment, unique, not NULL, PK
 - (b) name: varchar, unique, not NULL
 - (c) Relationship: a 1:M relationship between Exercises and Categories is implemented with the Category entity categoryID attribute as a foreign key attribute on the Exercise entity. (a category can have many exercises, an exercise belongs to one category).

b) Entity-Relationship Diagram



c) Schema



d) Screenshots

Home page with entity index

Welcome to Workout App!

Manage Data

- [Users](#)
- [Workouts](#)
- [Exercises](#)
- [Equipment](#)
- [Categories](#)

Users index (READ Users)

Users Index

New user

- [Matt Yang - yangmat@oregonstate.edu](#)
- [Anusha P - test@gmail.com](#)

Create user

Create a User

Email address

First name

Last name

Submit

User detail (READ/DELETE User)

Matt Yang

yangmat@oregonstate.edu

Edit user

Delete User

Workouts:

[Create a workout](#)

- [Sat Aug 21 2021 00:00:00 GMT-0700 \(Pacific Daylight Time\)](#)

User

edit (READ/UPDATE User)

Home Users Workouts Exercises Equipment Categories Search

Edit User

Email address

First name

Last name

Workouts index (READ Workouts)

Home Users Workouts Exercises Equipment Categories Search

Workouts Index

New Workout

- [Sat Aug 21 2021 - Matt Yang yangmat@oregonstate.edu](#)
- [Wed Aug 04 2021 - Anusha P test@gmail.com](#)

Create workout (CREATE Workout, READ Users)

Create a Workout

Select User

Select a user

Date

mm / dd / yyyy

Description

Submit

Workout detail (READ/DELETE Workout, READ/CREATE/DELETE Workout ⇔ Exercise relationship, READ Exercises)

Workout

Sat Aug 21 2021

Push Ups 20 reps

Edit workout details

Delete Workout

Add Exercise

Select an exercise to add

Add

[Create new exercise](#)

Exercises:

- [Bench Press](#) Remove
- [Pull Up](#) Remove
- [Dead Lift](#) Remove
- [Running](#) Remove
- [Swimming](#) Remove

Edit workout (READ/UPDATE Workout, READ Users)

[Home](#) [Users](#) [Workouts](#) [Exercises](#) [Equipment](#) [Categories](#) [Search](#)

Edit Workout

Select User

Matt Yang - yangmat@oregonstate.edu

Date

08 / 21 / 2021

Description

Push Ups 20 reps

Submit

Cancel

Exercises index (READ Exercises)

[Home](#) [Users](#) [Workouts](#) [Exercises](#) [Equipment](#) [Categories](#) [Search](#)

Exercises Index

New Exercise

- [Bench Press](#)
- [Squat](#)
- [Push Up](#)
- [Pull Up](#)
- [Dead Lift](#)
- [Dip](#)
- [Overhead Press](#)
- [Running](#)
- [Swimming](#)
- [Yoga](#)
- [Stretching](#)
- [Burpees](#)
- [test](#)

Create exercise (CREATE Exercise, READ Categories)

Create an Exercise

Name

Select Category (optional)

No category 

Add exercise

Exercise detail (READ/DELETE Exercise, READ/CREADE/DELETE Exercise ⇔ Equipment relationship)

Bench Press

Category: [Strength](#)

Edit exercise details

Delete Exercise

Add Equipment

Add equipment 

Add

[Create new equipment](#)

Equipment:

Exercise has no equipment

Edit exercise (READ/UPDATE Exercise, READ Categories)

Edit Exercise

Name

Bench Press

Select Category (optional)

Strength



Update exercise

Cancel

Equipment index (READ Equipment)

Equipment Index

New Equipment

- [dumbbell](#)
- [kettlebell](#)
- [pull up bar](#)
- [rowing machine](#)
- [squat rack](#)
- [yoga ball](#)
- [yoga mat](#)

Create equipment (CREATE Equipment)

Add Equipment

Name

Add equipment

Equipment detail (READ/DELETE Equipment, READ Equipment ⇔ Exercises relationship)

dumbbell

Edit equipment

Delete

Exercises that use this equipment

- [Squat](#)
- [Overhead Press](#)

Equipment edit (UPDATE Equipment)

Edit Equipment

Name

Edit equipment

Categories index (READ Categories)

Category Index

New Category

- [Calisthenics](#)
- [Cardio](#)
- [Flexibility](#)
- [Strength](#)

Create category (CREATE Category)

Add a category

Name

Add category

Category detail (READ/DELETE Category, READ Categories ↔ Exercises relationship)

Calisthenics

Edit category

Delete

Exercises for this category

- [Pull Up](#)
- [Dip](#)
- [test](#)

Edit category (UPDATE Category)

Edit Category

Name

Search form

Search



Search results (SELECT/LIKE from Users/Workouts/Exercises/Equipment/Categories)

[Home](#) [Users](#) [Workouts](#) [Exercises](#) [Equipment](#) [Categories](#) [Search](#)

Search Results

Entity Searched: Equipment

Search term: yoga

Search Results

- [yoga ball](#)
- [yoga mat](#)