Vincent Ta
Fauzi Kliman
CS 340

Dixon Recreation Center Database

URL: http://flip1.engr.oregonstate.edu:8261

Project Outline:

We will be creating a database that represents the Dixon Recreation Center at Oregon State University. Dixon is a recreational center where members can workout and participate in physical activities. There are classes, clubs, and personal trainers available for members to sign up for. The purpose of this database is to create a system for the rec center to keep track of all members as well as all the on going classes and activities that are available.

Database Outline, in Words:

The entities in our database are:

- Member - All members of Dixon who have access to the facility will use the Member entity in our database.
    - ID: Each member will have a unique id associated to them when recorded to the database which will be the primary key.
    - Trainer ID: Int used as foreign key to the Trainer table to indicate a trainer the member may have.
    - First Name: The first name of the member which is a string with 100 characters max. It cannot be blank and there is no default.
    - Last Name: The last name of the member which is a string with 100 characters max. It cannot be blank and there is no default.
- Class - The different classes taught at Dixon for members will have Class as the entity.
    - ID: Each class will have a unique id associated to them when recorded to the database which will be the primary key.
    - Instructor ID: Int used as foreign key to the Instructor table to indicate the instructor for the class.

    - Name: The name of the class which is a string with 100 characters max. It cannot be blank and there is no default.
    - Description: Brief description of the class. A string with 500 characters max. It can be blank and default is blank.
    - Price: The cost for the class per session which is an integer. It cannot be blank and there is no default.
- Trainer - The personal trainers working at Dixon will have the Trainer entity in the database.
    - ID: Each trainer will have a unique id associated to them when recorded to the database which will be the primary key.
    - First Name: The first name of the trainer which is a string with 100 characters max. It cannot be blank and there is no default.
    - Last Name: The last name of the trainer which is a string with 100 characters max. It cannot be blank and there is no default.
    - Sex: The sex for the trainer which is a string of maximum 6 characters and it can only be either of the two values: Male or Female. It cannot be blank and there is no default.
    - Description: A brief description about the trainer. A string with 500 characters max. It can be blank and default is blank.
    - Hourly Rate: The cost of the trainer per hour which is an integer. It cannot be blank and there is no default.
- Club - The different clubs that members can be apart of will be the Club entity in the database.
    - ID: Each Club will have a unique id associated to them when recorded to the database which will be the primary key.
    - Name: The name of the club which is a string with 100 characters max. It cannot be blank and there is no default.
    - Description: Brief description of the club. A string with 500 characters max. It can be blank and default is blank.
- Instructor - The instructors who teach classes will have the Instructor entity in the database.
    - ID: Each instructor will have a unique id associated to them when recorded to the database which will be the primary key.
    - First Name: The first name of the instructor which is a string with 100 characters max. It cannot be blank and there is no default.
    - Last Name: The last name of the instructor which is a string with 100 characters max. It cannot be blank and there is no default.

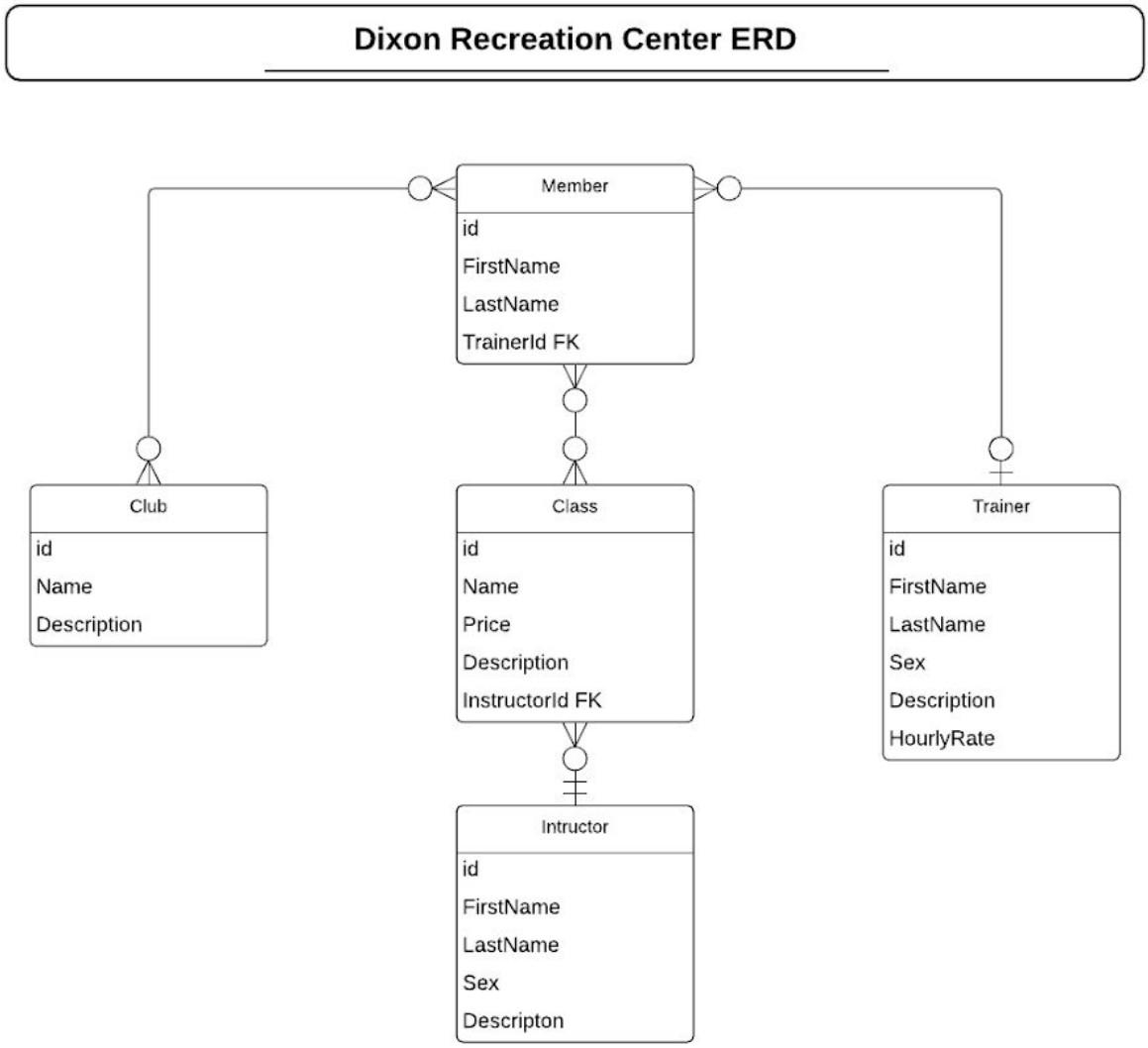
    - Sex: The sex for the instructor which is a string of maximum 6 characters and it can only be either of the two values: Male or Female. It cannot be blank and there is no default.
    - Description: Brief description about the instructor. A string with 500 characters max. It can be blank and default is blank.

- ClassMember - All members signed up for a class
    - Class ID: Int used as foreign key to the Class table to indicate the class the member is signed up for.
    - Member ID: Int used as foreign key to the Member table to indicate the member that is signed up to a class.
- ClubMember - All members signed up for a club
    - Club ID: Int used as foreign key to the Club table to indicate the club the member is signed up for.
    - Member ID: Int used as foreign key to the Member table to indicate the member that is signed up to a club.

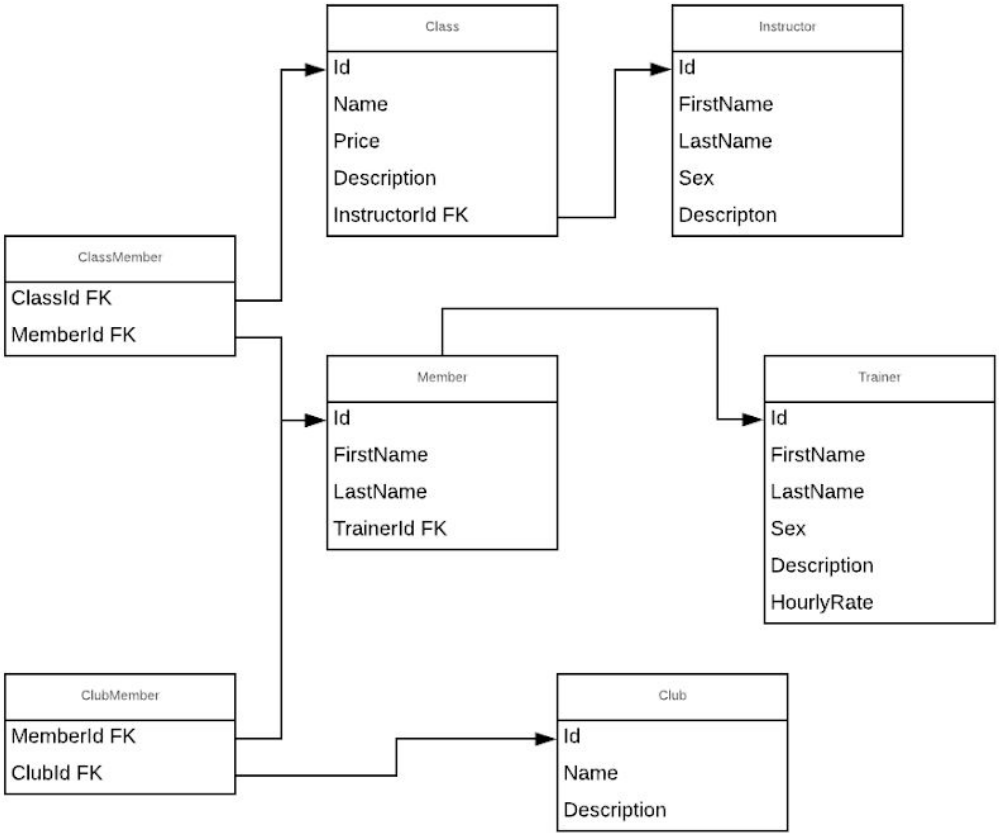
The relationships in our database are:

- Members can take classes - Members can sign up for multiple classes and a class can have multiple members in it. Member and Class entities are in a *many-to-many* relationship.
- Members can join clubs - Members can join multiple clubs and a club can have multiple members in it. Member and Club entities are in a *many-to-many* relationship.
- Members can hire a personal trainer - Members can hire at most one personal trainer and a trainer can have many members. Trainer and Member entities are in a *one-to-many* relationship.
- Classes must have one instructor - A class must have only one instructor and an instructor can teach multiple classes. Instructor and Class entities are in a *one-to-many* relationship

Entity-Relationship Diagram:



Schema:



Feedback by the reviewers:

Harrison Latimer

Hello Fauzi,

lots of ability to interact already with you site! There were two things I was thinking of when taking a look around. It seems like a few of your pages i.e. instructor / trainer registration maybe could all be put on the same page to reduce the number of places to look for similar information. It also isn't entirely clear what the flow or focus of the page is. Are the individuals interacting with your page primarily staff or users? If its

users then you probably wouldn't want them to enter information about trainers / staff / the price of a class. Other then this slight usage confusion, the site functions quite well! great job.
David Chen Adding new data using your forms doesn't seem to work yet. Ok, sounds good. Must've missed it cause I didn't refresh.
Joshua Fisher I like the changes that you've made to your site! It displays the database and it's functionalities elegantly. The only problem that I had while using it was when entering data there was no notification that the request went through, so I ended up hitting the submit button multiple times and inadvertently added multiple members with the same name and trainer. To solve this you could add something on the back end to check if the data already exists before adding it to the database to be safe, and also it would be helpful to see a confirmation on the front end that a change has been made. Great work!
Jacob Souther I like the simple design of your website. Everything is easy to find and your tables to display info are nicely formatted. It seems like most of the insert functions aren't present yet but no big deal. I like that you have set up all your drop down menus to pull info from the database as well, that will prevent the user from entering invalid values. Looks like you still need to add somewhere to edit/delete entries but I am sure they will come later. Off to a great start this week.

Actions based on the feedback:

For our update implementation, we made it so that it is possible for a member to sign up for a class. In the club and class page, users and select a member name and the desired class or club they would like to sign up for. After clicking the submit button, the club members or class members page will be updated to display the member being apart of the class or club. Based on our feedback, we also made our wage page dynamically refresh our table so that whenever someone adds a member to our database, the web page will automatically refresh and display the member in the table without having to manually refreshing the page.