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.

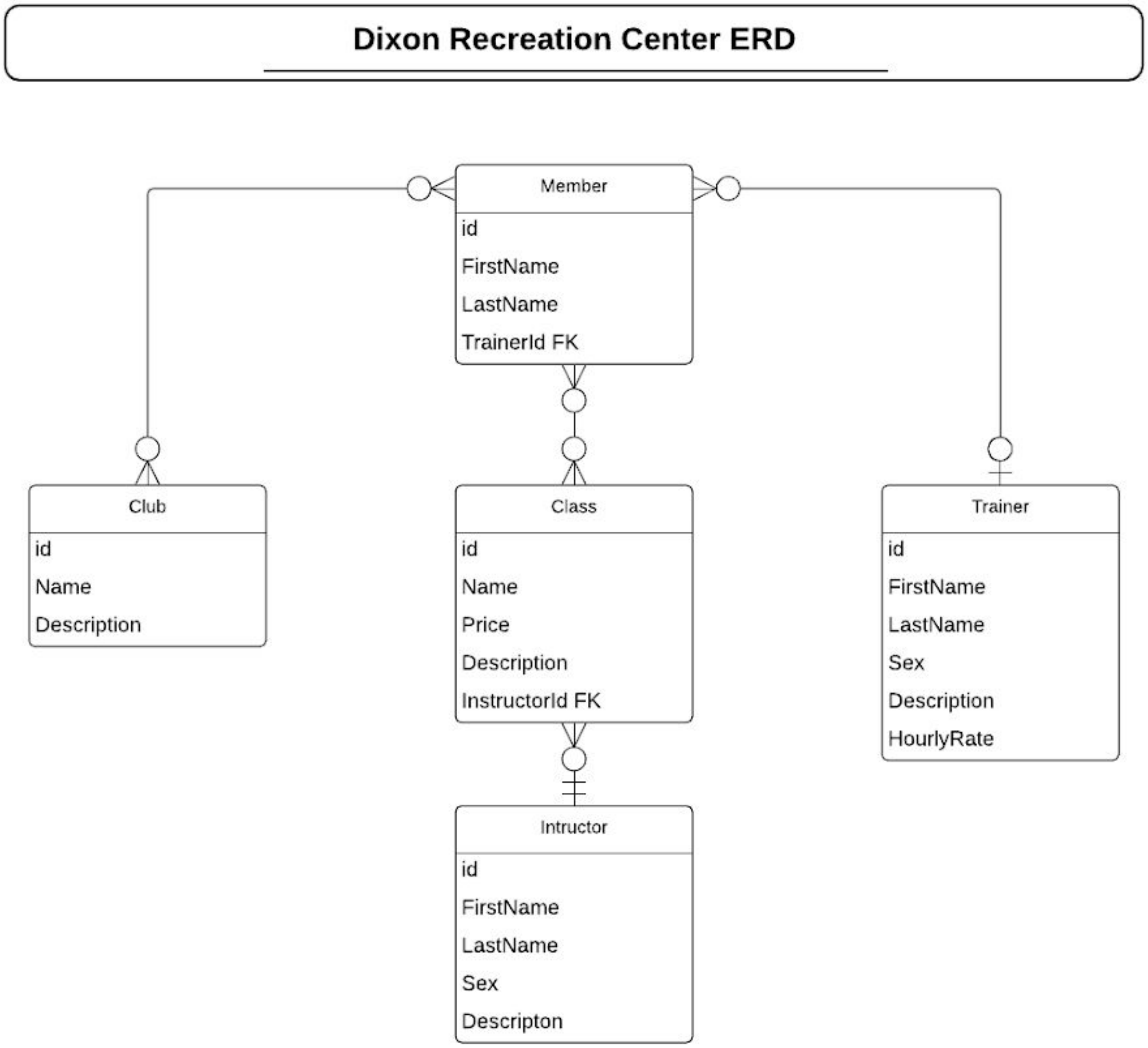
    - Sex: The sex for the instructor which is a string of maximum 6 characters and it can only be either of the two values: Male or Female. It cannot be blank and there is no default.
    - Description: Brief description about the instructor. A string with 500 characters max. It can be blank and default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

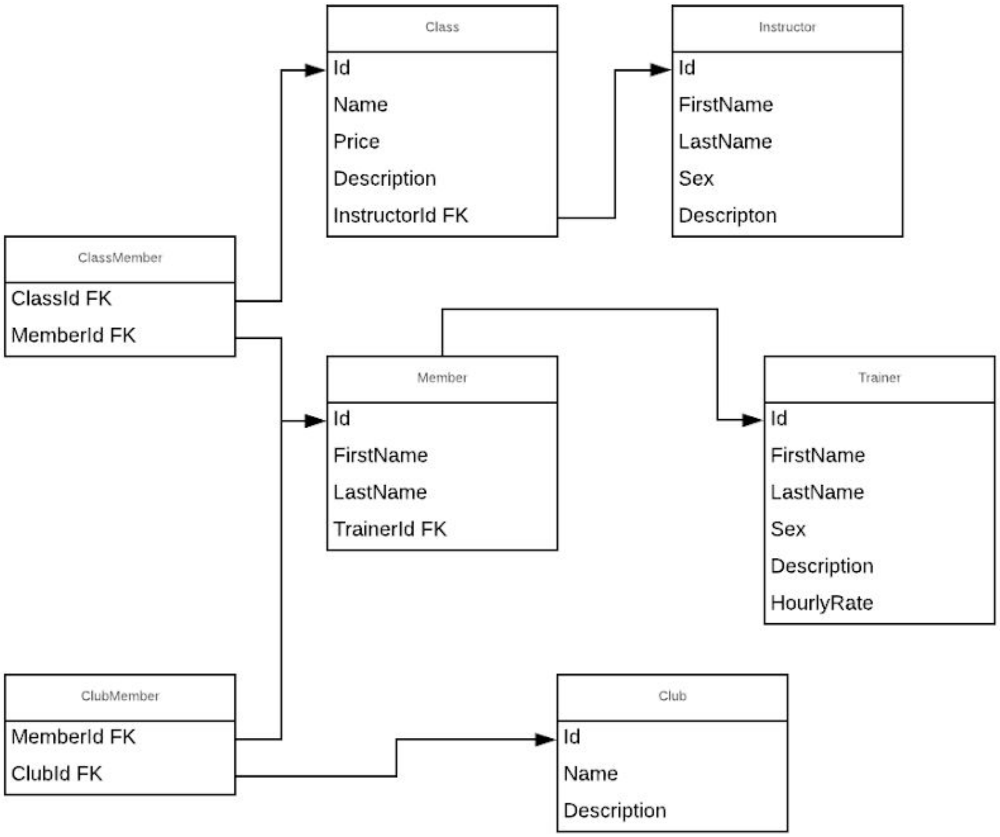
The relationships in our database are:

- 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 Great job on how your project is coming together! Not to be a squeaky wheel but I still think there are one too many composite tables for your relationships between classes and members which in the long run won't effect your project given the scale or our databases. Your html pages look great as well and I assume that once we get to entering some information in we will be able to see what information has been entered! Again great job so far!
David Chen

On the pages where you can sign members up for classes/trainers, it would be nice if you added a section where the user could view all members to make it easier for them to assign members to classes/trainers.

Number inputs might be more suitable for fields such as Price

For your sign for up classes section, you could use a dropdown for classes and let the user choose which class they want to add a member to instead of having them enter it themselves. This would prevent issues where entered names don't match the name of any class in your database, as well as issues where entered names match the names of more than one class in your database.

Joshua Fisher

It looks like you have managed to keep your SQL creation file relatively simple while still including all of the necessary elements! I also think your data manipulation queries are similarly well thought out yet simple. One thing that I would add, which I'm sure you've already done, is make sure that your SQL files work with the mySql database, as it looks like you have decided to write them out yourself rather than using the phpMyAdmin file dump functionality.

For the html, you will probably need to add a location where users can view their data. I suspect that you are planning on placing that on the same pages where the data is inputted by the user, which I think is a fine option. I like you're simple layout, and I can tell that your group is already well along the way for creating a great project for this course!

Jacob Souther

Your sql creation file is very clean. I like that you have done all of your key setting/auto increment/etc. in the create table queries instead of using alters after the fact. It also looks like you have accounted for all of your attributes, primary keys, and foreign keys so great job on the table creation. Your data manipulation queries are also quite thorough. Keep in mind you will eventually need UPDATE and DELETE queries as well, but they are not too functionally different than what you have already.

For your html, you have done a great job creating a sleek, easy to navigate site. Everything is easy to find and the inclusion of a nav bar is handy. I would suggest adding sections or pages which show the contents of each table so you can see all the available classes, instructors, etc. There is also a requirement to have search functionality so make sure you add that in the future. Otherwise, the site looks great!

Actions based on the feedback:

We will NOT be removing any tables as we feel that all tables present are needed for the project. Removing a composite table as suggested by a reviewer would limit the functionality of the project and thus not meet the required prerequisites of the final project. Each page for the entities of our database now reads and displays all the data from each entity table. Also, each page has the ability to insert a new person, class, or club to each corresponding table. We changed our Member page so that there is a drop down menu of available trainers to pick from when adding a new member. We also created a search bar that allows users to search for a member using their name as a search filter. The ability to insert into each of the tables through the UI was added. The ability to read all the tables from the UI was added as well.