

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.

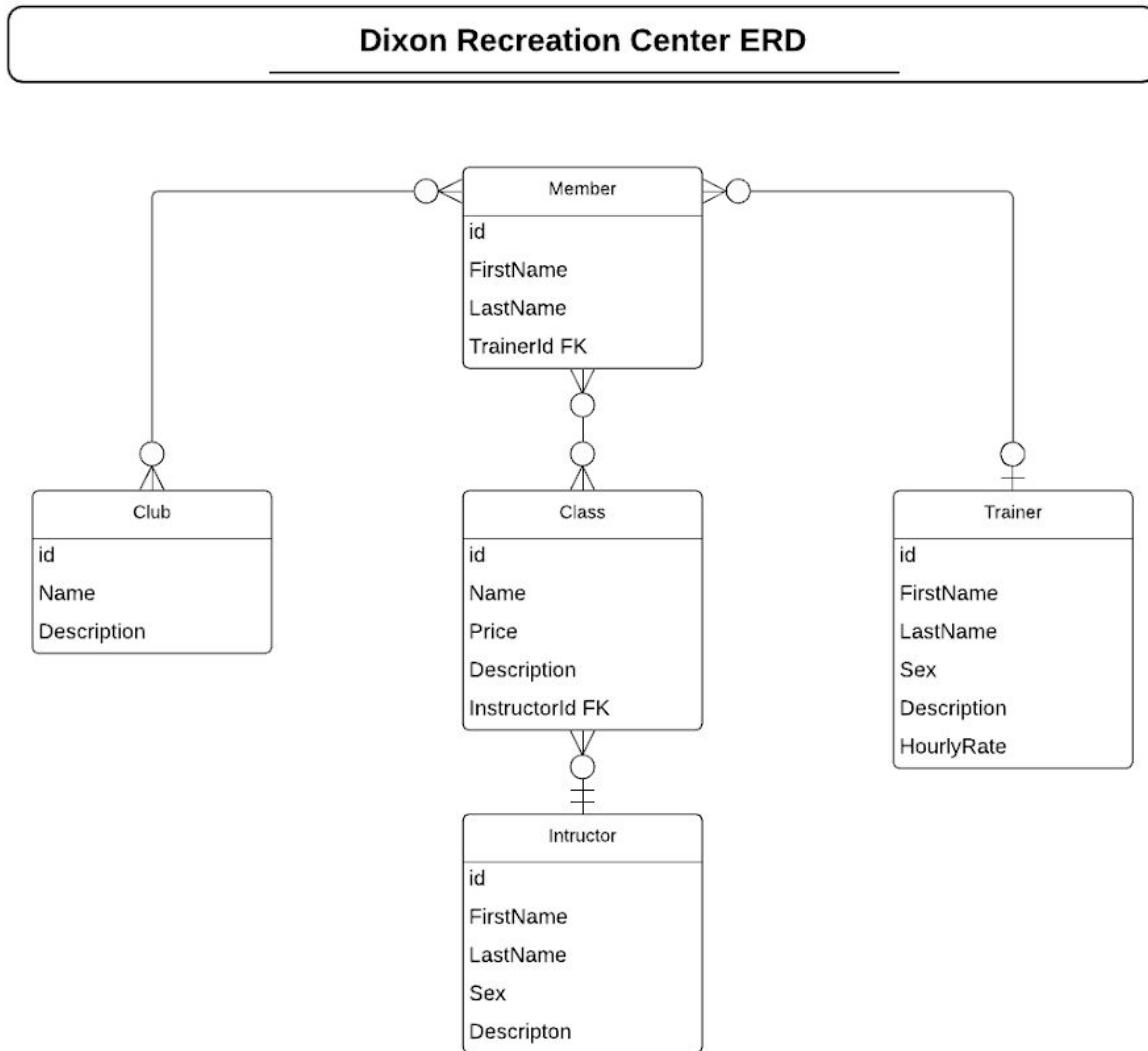
- Name: The name of the class which is a string with 100 characters max. It cannot be blank and there is no default.
 - Description: Brief description of the class. A string with 500 characters max. It can be blank and default is blank.
 - Price: The cost for the class per session which is an integer. 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.

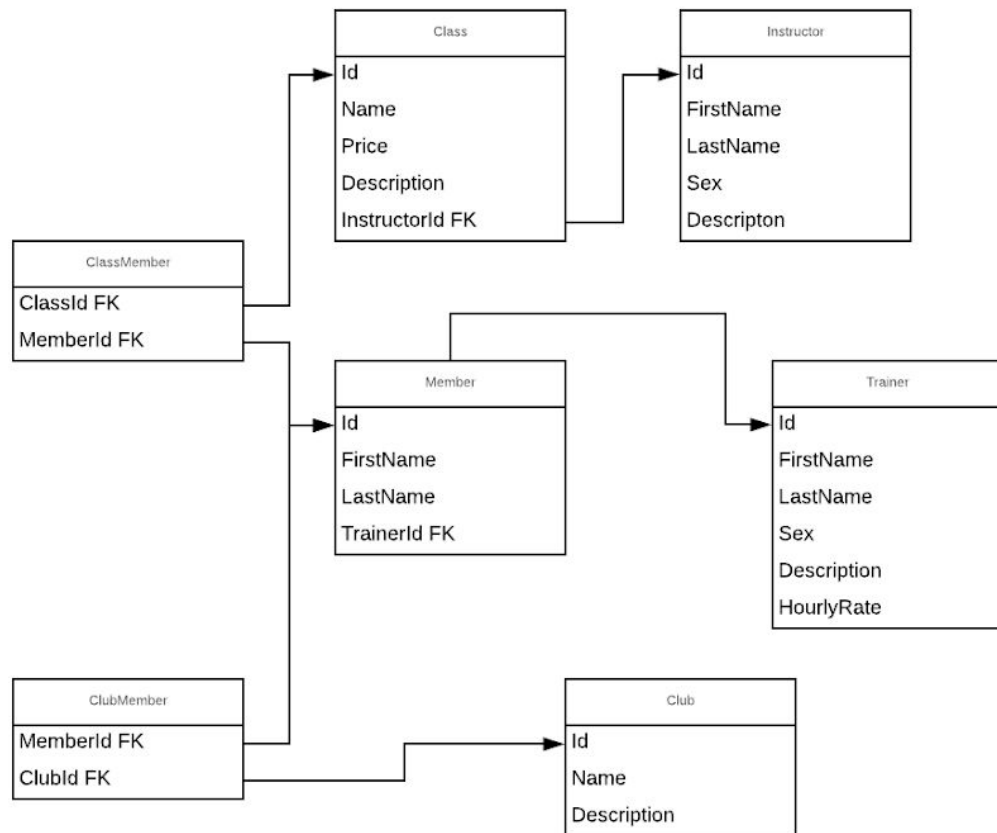
The relationships in our database are:

- 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:

Step 6) Feedback:

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.

Step 6) Actions based on Feedback:

For this step we implemented the delete functionality for the club page. Users are able to traverse to the club page, select the desired club from a drop down menu and delete the selected club from our database. Once the club is deleted, users can see that the club is removed from the list of clubs available. This notifies users that the club has been deleted.

Step 5) Feedback:

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.

Step 5) Actions based on 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.

Step 4) Feedback:

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!

Step 4) Actions based on 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.

Step 3) Feedback:

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 your 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!

Step 3) Actions based on 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. Search functionality for the members table will be added to meet the search requirements of the project. Some form inputs will be changed to be number inputs instead of text inputs. All rows for each entity will be displayed on their respective pages.

Step 2) Feedback:

Navine Rai

Some suggestions:

- Your Member table has an attribute named TrainerId which is included in your schema but not described in the text portion of your outline. The same goes for the InstructorId attribute in the Class table. I would add these fields to the text portion of your outline and describe which primary key they refer to (assuming they will be used as foreign keys).

- For your schema I would add some notation (arrows) to indicate which foreign keys refer to which primary keys. An example can be seen in the schemas lecture video at 07:02. Additionally I would indicate which attributes in each field serve as the primary key by underlining them or adding some sort of abbreviation such as PK.

Harrison Latimer

Your ER diagram looks great! The one thing I did wonder about is where the two tables came from for ClassMember and ClubMember. How do they relate to your ER diagram and what is their overall purpose to your schema? Could you possibly combine some things to make your table schema more simple? I'm struggling myself with the same issues

Joshua Fisher

Although the database is well planned, I think you should think about what the database will be used for. What will the website look like and what will users use it for? For example, if it is a website where users can look up classes or trainers, then maybe you could add more attributes in the database to describe the classes/trainers to help them make a decision (I think a string storing a description would be sufficient).

I would also consider not storing the sex of the users, unless you need it. Where I work we had genders stored in the database too when it was designed years ago, but we recently got into a controversy because it was limited to male/female and not all users were comfortable with the limited options. It may be easier to not worry about gender unless it is truly needed.

Lastly, you should consider showing the relationships in the schema somehow. I see the tables for classMember and clubMember, but what about the relationships between trainers and members etc.?

Iliana

I also like Joshua wondered why Sex was an attribute for Members, Trainers, and Instructors. Since classes and clubs don't seem like they can be exclusive to a certain gender, there doesn't seem to be any reason for the attribute to be in the database. Perhaps you can swap this out for other personally related attributes, such as keeping track of a Member's birthday.

Step 2) Actions based on Feedback:

We took out the sex attribute for members since it is not really necessary. Instead, the date of birth attribute has been added to members. Also, a description for classes, trainers, clubs, and instructors has been added to give more information about each entity. Entries for the foreign keys were added as well as arrows for the foreign keys in the schema. Entries for describing the ClassMember and ClubMember tables were added too.