Vincent Ta
Fauzi Kliman
CS 340

Dixon Recreation Center Database

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

Part A) Project Outline and Database Outline, ERD and Schema Updated Version:

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.

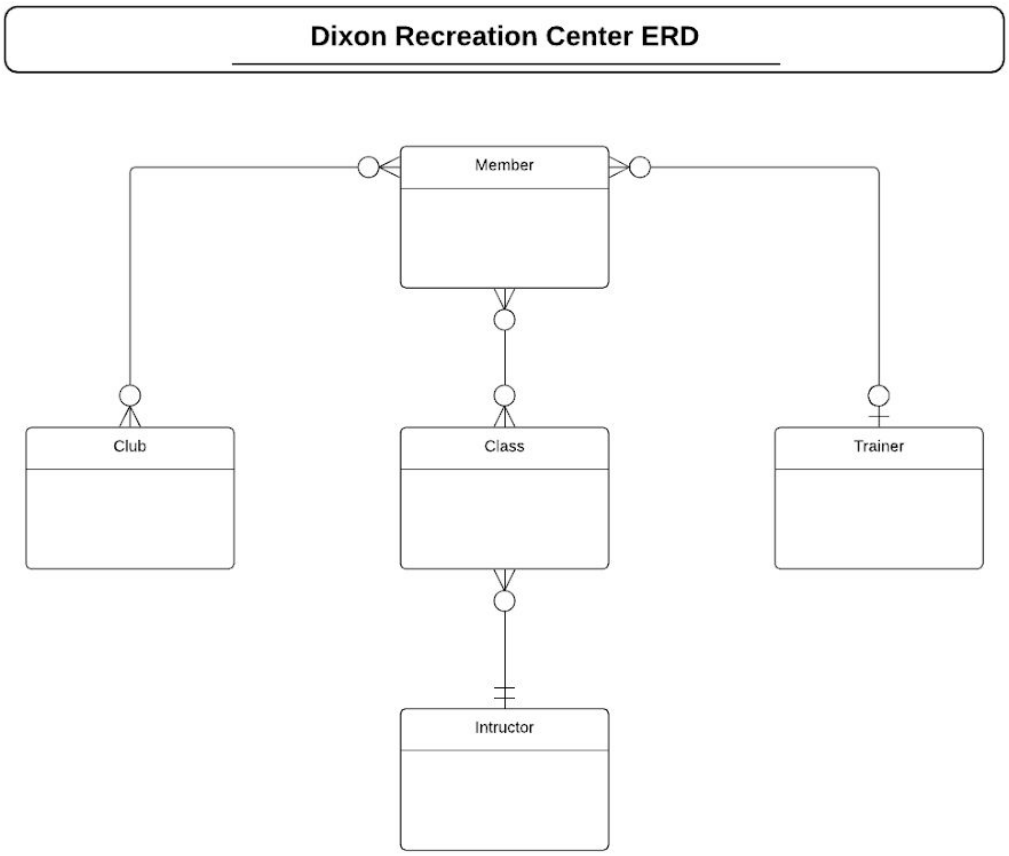
    - Last Name: The last name of the instructor which is a string with 100 characters max. It cannot be blank and there is no default.
    - Sex: The sex for the instructor which is a string of maximum 6 characters and it can only be either of the two values: Male or Female. It cannot be blank and there 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.

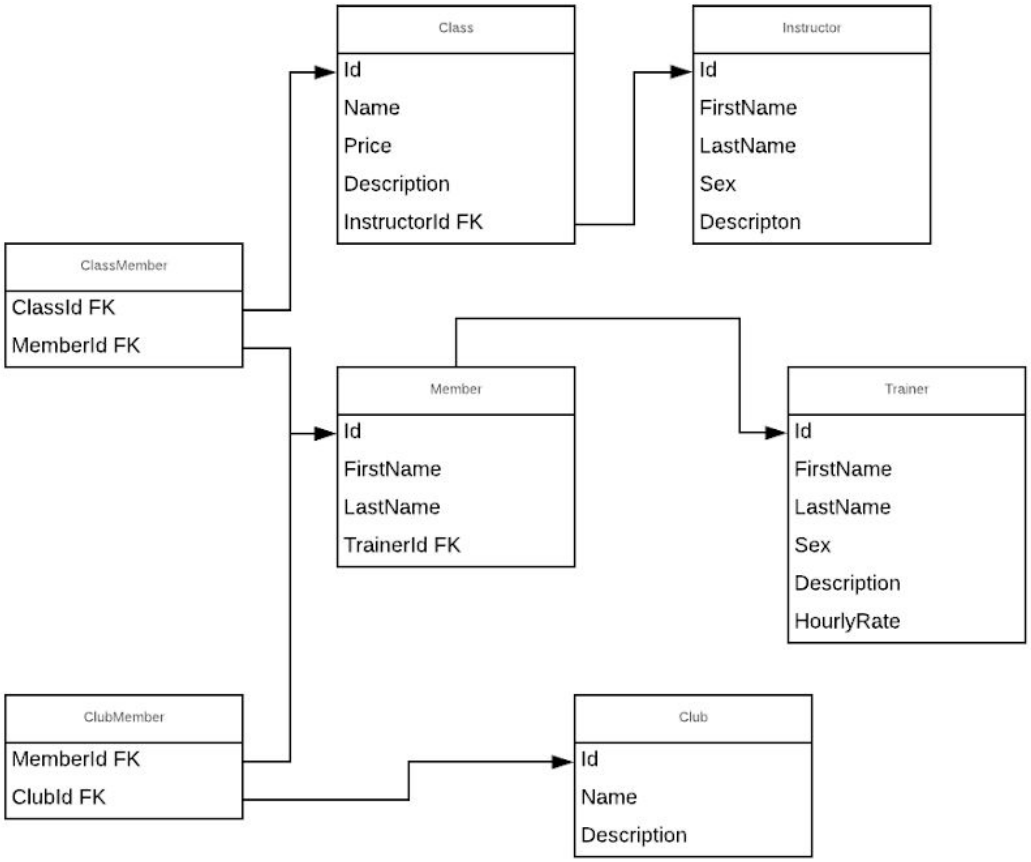
The relationships in our database are:

- 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:



Part B) **Fixes based on Feedback from Previous Steps:**

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 possible 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.

Fixes:

We took out the sex attribute for members since it is not really necessary and it will avoid future controversy. We decided to keep the sex attribute for both trainer and instructor because members might have a preference or feel more comfortable having a male or female leading their work outs. Also, a description attribute has been added for classes, trainers, clubs, and instructors 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.