Jane Lee
Rebecca Starr
Group #24

Petfunder

HTML Link

http://web.engr.oregonstate.edu/~leeja6/cs340-petfunder/index.html

Project Step 3 (Design/HTML Interface) - Feedback (Actionable: Highlighted)

***TA** - Andrew Jung*

Great job! Some suggestions:

- You should aim to use names of things instead of ID numbers which are not very helpful unless you already know what those ID numbers are referencing
- Use dropdown options to dynamically load values from the database for FK values. A user should not be expected to manually enter these values.
- Create a template navigation bar that remains the same on every page rather than having links change on every page.

Peers

Adam Okasha

- Does the UI utilize a SELECT for every table in the schema? In other words, data from each table in the schema should be displayed on the UI. Note: it is generally not acceptable for just a single query to join all tables and displays them.
  - Yes it looks like there is a select for every table in the schema. By tabbing at the top we can see data for pets, sponsors, sponsorships and shelters.
- Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?
  - Yes, the sponsors tab allows filtering by sponsorId.
- Does the UI implement an INSERT for every table in the schema? In other words, there should be UI input fields that correspond to each table and attribute in that table.
  - Yes, there is a form for pets, sponsors, sponsorships and shelters to allow inserting new rows.
- Does each INSERT also add the corresponding FK attributes, including at least one M:M relationship? In other words if there is a M:M relationship between Orders and Products, INSERTing a new Order (e.g. orderID, customerID, date, total), should also INSERT row(s) in the intersection table, e.g. OrderDetails (orderID, productID, qty, price and line_total).
  - Yes, there is a M:M relationship between sponsors and pets. I can see from the sponsorships page that these can be added. Since the ids are inserted manually, there is nothing precluding a user from adding data to have that M:M relationship
- Is there at least one DELETE and does at least one DELETE remove things from a M:M relationship? In other words, if an order is deleted from the Orders table, it should

also delete the corresponding rows from the OrderDetails table, BUT it should not delete any Products or Customers.
    - Yes. On the sponsorship (M:M sponsors to pets) page there is a delete button which will get rid of a row.
- Is there at least one UPDATE for any one entity? In other words, in the case of Products, can productName, listPrice, qtyOnHand, e.g. be updated for a single ProductID record?
    - It looks like there are edit buttons for rows of all entities though it does not appear to go anywhere at the moment. It just needs implementation.
- Is at least one relationship NULLable? In other words, there should be at least one optional relationship, e.g. having an Employee might be optional for any Order. Thus it should be feasible to edit an Order and change the value of Employee to be empty.
    - Yes. The relationship between pets and sponsors is nullable. A pet can exist without a sponsor and vice versa.
- Do you have any other suggestions for the team to help with their HTML UI?
    - I like the simple design and clarity of your schema. I think having the forms on the same page will require ajax and updating the tables client-side javascript (like a single page application). This makes the UX nicer.
      Putting the forms on separate pages and redirecting and injecting the new data back into the template might be easier. Either approach is fine though. Great job!

Kate Meyer

Does the UI utilize a SELECT for every table in the schema? In other words, data from each table in the schema should be displayed on the UI. Note: it is generally not acceptable for just a single query to join all tables and displays them.
Yes. Each entity has a table. The Sponsorships page allows the user to toggle between the ShelterSponsorships and PetSponsorships tables.
Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?
Yes, there is the ability to filter by sponsorID on the Sponsorships page.
Does the UI implement an INSERT for every table in the schema? In other words, there should be UI input fields that correspond to each table and attribute in that table.
Yes, users are able to create a new Sponsor, Pet, Shelter, and Sponsorship for both types of sponsorships with all of the attributes listed in the schema.
Does each INSERT also add the corresponding FK attributes, including at least one M:M relationship? In other words if there is a M:M relationship between Orders and Products, INSERTing a new Order (e.g. orderID, customerID, date, total), should also INSERT row(s) in the intersection table, e.g. OrderDetails (orderID, productID, qty, price and line_total).
Yes. When inserting into the ShelterSponsorships table the foreign keys shelterID and sponsorID are added. When inserting into the PetSponsorships table the foreign keys petID and sponsorID are added. These are both M:M relationships. At this time adding a new sponsor, pet, or shelter does not update the ShelterSponsorship or PetSponsorships tables.

Is there at least one DELETE and does at least one DELETE remove things from a M:M relationship? In other words, if an order is deleted from the Orders table, it should also delete the corresponding rows from the OrderDetails table, BUT it should not delete any Products or Customers.
Yes. Every entity has the option to DELETE which includes the M:M relationships for ShelterSponsorships and PetSponsorships. The DELETE function is not working for Pets and Shelters right now, but it is working for all other entities.
Is there at least one UPDATE for any one entity? In other words, in the case of Products, can productName, listPrice, qtyOnHand, e.g. be updated for a single ProductID record?
Yes. Every entity has the option to UPDATE. This does not function yet.
Is at least one relationship NULLable? In other words, there should be at least one optional relationship, e.g. having an Employee might be optional for any Order. Thus it should be feasible to edit an Order and change the value of Employee to be empty.
Yes. There is a NULLable relationship between Sponsors and Shelters, and between Sponsors and Pets. Shelters and Pets do not need a sponsor to be in the database.
Do you have any other suggestions for the team to help with their HTML UI?
I did not initially notice the toggle options on the Sponsorships page. You could add a description at the top of the page, similar to the home page.

Edward Yeow

- Does the UI utilize a SELECT for every table in the schema? In other words, data from each table in the schema should be displayed on the UI. Note: it is generally not acceptable for just a single query to join all tables and displays them.

Yes, the UI displays each entity table in separate pages and there are SELECTS for all the tables. All data/attributes for each entity are displayed appropriately.

- Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?

Yes, each table has a search function and the option to apply a filter; this will help in searching out specific entities.

- Does the UI implement an INSERT for every table in the schema? In other words, there should be UI input fields that correspond to each table and attribute in that table.

Yes, each table is on its own respective page and has its own corresponding INSERT function.

- Does each INSERT also add the corresponding FK attributes, including at least one M:M relationship? In other words if there is a M:M relationship between Orders and Products, INSERTing a new Order (e.g. orderID, customerID, date, total), should also INSERT row(s) in the intersection table, e.g. OrderDetails (orderID, productID, qty, price and line_total).

The Insert options do allow the addition of IDs, which will then become FK attributes. With these present, I am confident that at least one M:M relationship is being represented in the UI.

- Is there at least one DELETE and does at least one DELETE remove things from a M:M relationship? In other words, if an order is deleted from the Orders table, it should also delete the corresponding rows from the OrderDetails table, BUT it should not delete any Products or Customers.

Each entry in the table has its own DELETE button and will delete both things in a M:M relationship.

- Is there at least one UPDATE for any one entity? In other words, in the case of Products, can productName, listPrice, qtyOnHand, e.g. be updated for a single ProductID record?

Along with the DELETE button, each entry has its own UPDATE option to allow the editing of current attributes.

- Is at least one relationship NULLable? In other words, there should be at least one optional relationship, e.g. having an Employee might be optional for any Order. Thus it should be feasible to edit an Order and change the value of Employee to be empty.

Yes, the Pet to Shelter relationship is NULLABLE, as a Pet is not required to be linked to a shelter. Currently not able to edit the value to be empty, but am confident that it will be.

- Do you have any other suggestions for the team to help with their HTML UI?

This HTML mockup is very well done! I wish my mockup was as well done as this. I do agree with Kevin that the changing buttons at the top is a bit jarring, and I'd just suggest keeping the button order static to avoid confusion and maintain visual consistency in each page. Other than that, I think your INSERT functions are well formatted and your tables are well constructed with the necessary functions.

Kevin Zhu

- Does the UI utilize a SELECT for every table in the schema? In other words, data from each table in the schema should be displayed on the UI. Note: it is generally not acceptable for just a single query to join all tables and displays them.
    - There are SELECTS for all tables in the schema including the two relationship tables PetSponsorships and ShelterSponsorships.
- Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?
    - Yes, there is a sponsorID filter on the Sponsorships page that dynamically populates the table of sponsorships.
- Does the UI implement an INSERT for every table in the schema? In other words, there should be UI input fields that correspond to each table and attribute in that table.
    - There is an insert for every entity table in the schema, however it seems that there is no Yes the UI implements an INSERT for every table in the schema.
- Does each INSERT also add the corresponding FK attributes, including at least one M:M relationship? In other words if there is a M:M relationship between Orders and Products, INSERTing a new Order (e.g. orderID, customerID, date, total), should also INSERT row(s) in the intersection table, e.g. OrderDetails (orderID, productID, qty, price and line_total).
    - Yes, on the sponsorship page, there is an option to either insert into the PetSponsorship M:M relationship table or the ShelterSponsorship M:M relationship table.
- Is there at least one DELETE and does at least one DELETE remove things from a M:M relationship? In other words, if an order is deleted from the Orders table, it should also delete the corresponding rows from the OrderDetails table, BUT it should not delete any Products or Customers.
    - There are delete buttons on every table on the pages, including the sponsorships page which allows deletion from both M:M relationship tables (Sponsorships tables).

- Is there at least one UPDATE for any one entity? In other words, in the case of Products, can productName, listPrice, qtyOnHand, e.g. be updated for a single ProductID record?
    - Each row on each table has an edit button which allows for updating the attributes of a single row in line.
- Is at least one relationship NULLable? In other words, there should be at least one optional relationship, e.g. having an Employee might be optional for any Order. Thus it should be feasible to edit an Order and change the value of Employee to be empty.
    - Yes, it is possible for a pet to not be connected to any shelter and thus have a NULL value in the shelterID attribute of Pets.
- Do you have any other suggestions for the team to help with their HTML UI?
    - Very well designed and easy to use website overall. However, I keep getting thrown off and confused by the changing navigation bar buttons at the top of the page. They change to remove the button for the current page, and make it so that users have to make sure they are clicking the correct button, since the positions change. I would suggest keeping a static bar with 5 links for all pages, and not removing the link to the current page. So every page would have a navigation bar with links to Home, Sponsors, Sponsorships, Pets, and Shelters regardless of what page you are on.

Project Step 3 - Actions Based on Feedback

- Include a navigation bar rather than changing buttons for navigating between pages (Kevin, Edward, TA) - implemented
- Include foreign keys in dropdown pulled from database instead of free text for entry (TA) - implemented
- Reference foreign keys by name instead of only ID (TA) - implemented
- DELETE function not working for Pets and Shelters (Kate) - implemented
- Include description for Sponsorship toggle (Kate) - in order to not impact the consistency, we didn't add a description but we put the toggle in a box and bolded the text to make it more obvious
- Implement edit functionality (Adam) - we aren't currently implementing this in the design/HTML interface but it will be implemented once we put in the server functionality
- Putting forms/tables on different pages (Adam) - this was a suggestion that it may make the coding easier but we will keep the current design for the sake of our UX. We'll keep this recommendation in mind for the future if it's too challenging to maintain the current design

Projects Step 3 - Updates to Draft Version

- Included references to names of things instead of IDs (Sponsorships, Pets page)
- Implemented dropdowns for pulling in FK values (Sponsorships, Pets page)
- Put in a consistent navigation bar that stays the same across all pages instead of changing button links between pages (all pages)
- Included working DELETE functionality for sample data on the Pets and Shelters page (Pets, Shelters, already implemented on Sponsorships and Sponsors)
- Put the Sponsorships toggle options (for creating sponsorships and displaying sponsorships by pet or shelter) in a box and bolded the text so that it is more immediately noticeable to the user (Sponsorships page)

Project Step 2 - Peer Feedback (Actionable Feedback is Highlighted)

Lucian Haj

- Does the overview describe what problem is to be solved by a website with DB back end?

Many shelters have lots of pets which need sponsors and keeping track of the animals status can become complicated if the animal moves shelters or is adopted. To help more animals get sponsors and be adopted this website will make tracking pets relationship to shelters/sponsors easier.

- Does the overview list specific facts?

Yes, references the database's anticipated user count with actual numbers as well as the anticipated quantity of all entities.

- Are at least four entities described and does each one represent a single idea to be stored a s a list?

Yes, pets, sponsors, shelters, and sponsorships can all be represented as a single idea.

- Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints and describe relationships between entities?

Yes, Sponsors are related to none or more pets, shelters sponsored by to none/one or many sponsors, pets can have sponsorships, and sponsorships are related to both pets and shelters.

- Are 1:M relationships correctly formulated? Is there at least one M:M relationship?

Yes, sponsors correctly relates with multiple sponsorships and shelters has multiple sponsorships. There is also a M:M relationship between pets and sponsors.

- Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?

All entities maintain capitalization and are all plural. Entity names also maintain spelling correctness, but entities change count when joined with other entities. For example, Shelters becomes Shelter in the ShelterSponsorships table. The same goes for Pets and PetSponsorships.

Alan Kuo

Peer Review:

1. Yes, the overview does a good job in describing a problem to be solved with a database-backed website.
2. Yes, the overview lists some very specific facts and figures regarding the number of pets/shelters involved that helps the reader understand the scope of the website/database
3. Five entities are listed (Pets, Sponsors, Shelters, PetSponsorships, and ShelterSponsorships). Pets, Sponsors, and Shelters can be understood pretty easily as single ideas. PetSponsorships and ShelterSponsorships were a little harder to understand as single ideas for me. From my understanding, they represent the relationship between a Pet & Sponsor or a Shelter & Sponsor - capturing information about the amount/duration/etc. of the relationship while also facilitating the M:M relationship as composite entities. I'm not as about these two entities representing single stand-alone ideas, but this might just be me.
4. Yes, the outline very clearly details the purpose, attributes, and relationships for each entity listed.

5. Yes, the 1:M relationships appear to be correctly formulated. There are two M:M relationships (Pets/Sponsors and Shelters/Sponsors) implemented with the composite entities PetSponsorships and ShelterSponsorships.
6. Yes, the naming appears consistent throughout. Entities (Pets, Sponsors, Shelters, etc) are consistently plural, and attributes are consistently singular. Names with multiple words are consistently named with camelCase

Elizabeth De Laurell
This is an awesome idea! The overview provides a thorough description of the problem that can be solved through a database backed website. Your use of numbers and statistics are well done and help further describe the vastness of the problem the website is trying to solve. There are five entities with three (Pets, Sponsors, Shelters,) as true entities and two (PetSponsorships and ShelterSponsorships) representing relationships but having their own attributes as well which transform them into entities. They all represent a single idea that can be portrayed as a list. The outline does a good job of describing the purpose of each entity. It also provides datatypes and constraints for each attribute such as Not NULL and unique. The 1:M relationships seem to be formulated correctly and there are two M:M relationships. These relationships appear to be correct with a FK from two entities. One thing I did notice was in your ERD the relationship between Shelters and Pets, you state shelters can have 0 or more pets. In your "Deletions"' description however, you state that a shelter would be deleted if it has no pets in the database, so the relationship would have to have participation of at least one. For each entity and attribute, the naming is consistent. Entities are plural and attributes are singular. Also, another idea, some sponsors may want to donate anonymously, so maybe you could add an attribute to the Sponsors entity that is a boolean value for anonymous. That's not necessary at all though! Nice work!!

Felipe Teixeira Groberio
Hi Rebecca and Jane,
Does the overview describe what problem is to be solved by a website with DB back end?
I like your project, very cool and honorable idea. The project overview describes de problem clearly and gives a high level insight of the database. I also like that your project database different types of clients, shelters and pet sponsors.
Does the overview list specific facts?
The project overview lists specific facts and numbers about pets, adoption, sponsoring, and shelters in the United States.
Are at least four entities described and does each one represent a single idea to be stored as a list?
Yes, more than four entities are described. Though, I would like to suggest that you make an entity for addresses, for two reasons: one, so you can reduce the number of attributes for Shelters; and two, because Sponsors entity may need address information in the future.
Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints and describe relationships between entities?
The database entity outline describes in details the purpose of each entity, and lists all the attributes while noting the datatypes. Relationships are also well annotated.
Are 1:M relationships correctly formulated? Is there at least one M:M relationship?

The One-to-Many relationships are correctly formulated. There exists two Many-to-Many relationship on this project between shelter and sponsors, and between pets and sponsors. Well done.
Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?
Entities are in plural and attributes are in singular, while in camel case naming convention. I could not find any naming inconsistency.
Closing remarks...
Your ER diagram and schema are very well done. Good luck with the project, it is looking great already!

Actions Based on Peer Feedback

"Entities change count when joined with other entities. For example, Shelters becomes Shelter in the ShelterSponsorships table. The same goes for Pets and PetSponsorships." (Lucian Haj) - We are not changing our draft based on this feedback. I'm not sure if this is meant to be a correction or just a call-out due to the question, but regardless the naming convention is because Pets is referring to an entity that is holding multiple Pets whereas PetSponsorships is referring to an entity that is holding multiple Sponsorships, each of which is only corresponding to one Pet. It is an intentional decision on our part based on our database design.

"I'm not as about these two entities (PetSponships and ShelterSponsorships) representing single stand-alone ideas, but this might just be me." (Alan Kuo) - We are not updating our draft based on this feedback. Please see the Project 2a (Updates based on Feedback on Project Step 1) section. The splitting of the sponsorships into the two sections is an intentional decision based on Pets and Shelters being two distinct entities and necessary so that we can use petID and shelterID as FK in their respective sponsorship tables. This was based on a recommendation from the TA.

"One thing I did notice was in your ERD the relationship between Shelters and Pets, you state shelters can have 0 or more pets. In your "Deletions'" description however, you state that a shelter would be deleted if it has no pets in the database, so the relationship would have to have participation of at least one." (Elizabeth DeLaurell) - We are not updating our draft based on this feedback. The relationship between Shelters and Pets is intentional since in the deletions section of our Shelters entity, we say "If a shelter can no longer be sponsored and has no pets in the Pets database, then it is removed from the Shelters database". A shelter is allowed to be sponsored without any pets in the database, so a shelter can be associated with 0 or more pets.

"Also, another idea, some sponsors may want to donate anonymously, so maybe you could add an attribute to the Sponsors entity that is a boolean value for anonymous." (Elizabeth DeLaurell) - We think this is a cool idea and we will add a boolean anonymous attribute to our Sponsors table.

"Though, I would like to suggest that you make an entity for addresses, for two reasons: one, so you can reduce the number of attributes for Shelters; and two, because Sponsors entity may need address information in the future." (Felipe Teixeira Groberio) - We are not updating our draft based on this feedback. The first reason does not seem sufficient for us to create a separate entity and if we want to include address info in the Sponsors entity we can add those attributes to the Sponsors table

directly. We do not want to combine addresses for the two distinct entities in one table otherwise it would be unclear whether to do a FK based on shelterID or sponsorID. It would be necessary for us to have a shelterAddresses and sponsorAddresses table. At this point, we think it is fine to keep the address attributes in the entity tables themselves.

Upgrades to the Draft Version

- We added the “anonymous” attribute to the Sponsors table.
- We added a “registrationDate” attribute to the Pets table.
- We updated “age” in the Pets table to “birthday”.
- We changed the 1:M relationship between Shelters and Pets so that it is a nullable 1:M relationship. This is in order to fulfill the project requirements in the event that any shelters are closed/no longer associated with a pet that is still sponsorable and so that ongoing sponsorships for sponsorable pets may still continue. In that event, the shelterID for a pet will be set to NULL and the relationship between the shelter and the pet will be removed. Therefore Shelters now has partial participation in Pets.
- We updated our outline to include primary key notation for composite keys (sponsorID, petID) in the PetSponsorships table and (sponsorID, shelterID) in the ShelterSponsorships table.
- We updated our schema to include arrows between FKs and PKs.

Project Step 2a (Updates based on Feedback on Project Step 1)

These are the changes that we are making based on the TA feedback for our design:

- Style-related
    - Included more numerical facts to estimate numbers entered in the database in the overview
    - More clearly defined the "goal" attribute in the "Pets" entity
- Design-related
    - Broke down our "Sponsorships" entity which previously existed as one entity to capture the M-M relationship between Pets OR Shelters and Sponsors (Sponsors would have a sponsorship for at least one Pet or one Shelter and Pets and Shelters could be sponsored by any number of Sponsors) into two separate entities, PetSponsorships and ShelterSponsorships. This is so that we can use petID from Pets as a FK in the PetSponsorships entity and shelterID from Shelters as a FK in the ShelterSponsorships entity.
    - Removed the sponsorshipID attribute from the Sponsorships entity (now broken out as the PetSponsorships and ShelterSponsorships entities) -- no separate primary key is needed for this table since (sponsorID, petID) can function as a PK for PetSponsorships and (sponsorID, shelterID) can function as PK for ShelterSponsorships.

Overview

Approximately 1.5 million animals die in shelters every year. Our app aims to address this by providing animal rescues and animal shelters a source of animal-loving sponsors who will provide monthly sponsorship gifts to animals at these shelters who have not yet found a home or to the shelters themselves. Any public or private animal rescue organizations located in the United States can join our app by registering for the shelter database (est. max of 3500, the number of animal shelters in the United States). They can register their unadopted pets in our pet database (est. max of

6.5 million, the number of pets entering shelters in the US every year). Individuals who choose to sponsor one or more of these pets or shelters will be registered to the sponsor database (est. max of 50 million, the number of gofundme sponsors donating through 2017) and sign up for various sponsorships recording the monthly amounts they wish to sponsor for and the recipients of their sponsorship, which can be any pet or sponsorable shelter in the pet and shelter databases (est. max of 100 million sponsorships assuming average of 2 sponsorships/sponsor).

Outline

Pets: Entity recording the attributes for the adoptable animal registered by the shelter

- Attributes
    - petID: int, auto_increment, unique, not NULL, primary key
    - registrationDate: date, not NULL
    - name: varchar, not NULL
    - birthday: date, not NULL
    - animal: varchar, not NULL (cat,dog,etc.)
    - breed: varchar
    - personality: varchar, NULL
    - adoptable: boolean, not NULL
    - goal: varchar, not NULL (this is what the sponsor money will go towards, e.g. toys, medicine, food, etc. - entered text value)
    - shelterID: foreign key
- Relationships: Each pet can be sponsored by any number of sponsors (including none). A sponsor can sponsor any number of pets (including none). Each pet can belong to at most one shelter. A pet can be in multiple sponsorships (or none) but each pet sponsorship contains one pet.
    - a M:1 relationship between Pets and Shelters is implemented with shelterID as a FK inside of Pets
    - a M:M relationship between Pets and Sponsors is implemented with petID as a FK and sponsorID as a FK inside of PetSponsorships
    - a 1:M relationship between Pets and Sponsorships with petID as a FK inside of PetSponsorships
- Deletions: If a pet can no longer be sponsored (is adopted, etc.), it is removed from the Pets database and any sponsorships associated with the pet are also removed from the PetSponsorships database.

Sponsors: Entity recording the attributes for the sponsors registered on the app

- Attributes
    - sponsorID: int, auto_increment, unique, not NULL, primary key
    - firstName: varchar, not NULL
    - lastName: varchar, not NULL
    - anonymous: boolean, not NULL (for sponsors that choose to have their identities hidden to the pets/shelters that they are sponsoring)

- Relationships: Each sponsor can sponsor any number of pets and shelters. A sponsor can have multiple sponsorships (they must have at least one sponsorship to be in the database) but each sponsorship contains only one sponsor.
    - a M:M relationship between Pets and Sponsors is implemented with petID as a FK and sponsorID as a FK using the PetSponsorships table
    - a M:M relationship between Shelters and Sponsors is implemented with shelterID as a FK and sponsorID as a FK using the ShelterSponsorships table
    - a 1:M relationship between Sponsors and PetSponsorships is implemented using sponsorID as a FK inside of PetSponsorships
    - a 1:M relationship between Sponsors and ShelterSponsorships is implemented using sponsorID as a FK inside of ShelterSponsorships
- Deletions: A sponsor is removed from the Sponsors database if the sponsor ends all associated sponsorships

PetSponsorships: Entity recording the existing sponsorships that have been set up for sponsors describing which pets they are sponsoring and the monthly amounts they are sponsoring them for

- Attributes
    - sponsorID: foreign key, not NULL
    - petID: foreign key, not NULL
    - (sponsorID, petID): primaryKey, not NULL
    - amount: numeric, not NULL (amount sponsor is donating for pet per month)
    - beginDate: date, not NULL (date that sponsorship is initiated)
    - endDate: date (for sponsorships that have a specified end date)
- Relationships: Each sponsorship relates to one sponsor and one pet.
    - a M:M relationship between Pets and Sponsors is implemented with petID as a FK and sponsorID as a FK inside of PetSponsorships
    - a 1:M relationship between Sponsors and PetSponsorships is implemented using sponsorID as a FK inside of PetSponsorships
    - a 1:M relationship between Pets and PetSponsorships using petID as a FK inside of PetSponsorships
- Deletions: A sponsorship is removed from the PetSponsorships database if the sponsor ends the sponsorship, the sponsored pet is no longer accepting sponsors (pet is removed from Pets database), or the current date is past the endDate

ShelterSponsorships: Entity recording the existing sponsorships that have been set up for sponsors describing which shelters they are sponsoring and the monthly amounts they are sponsoring them for

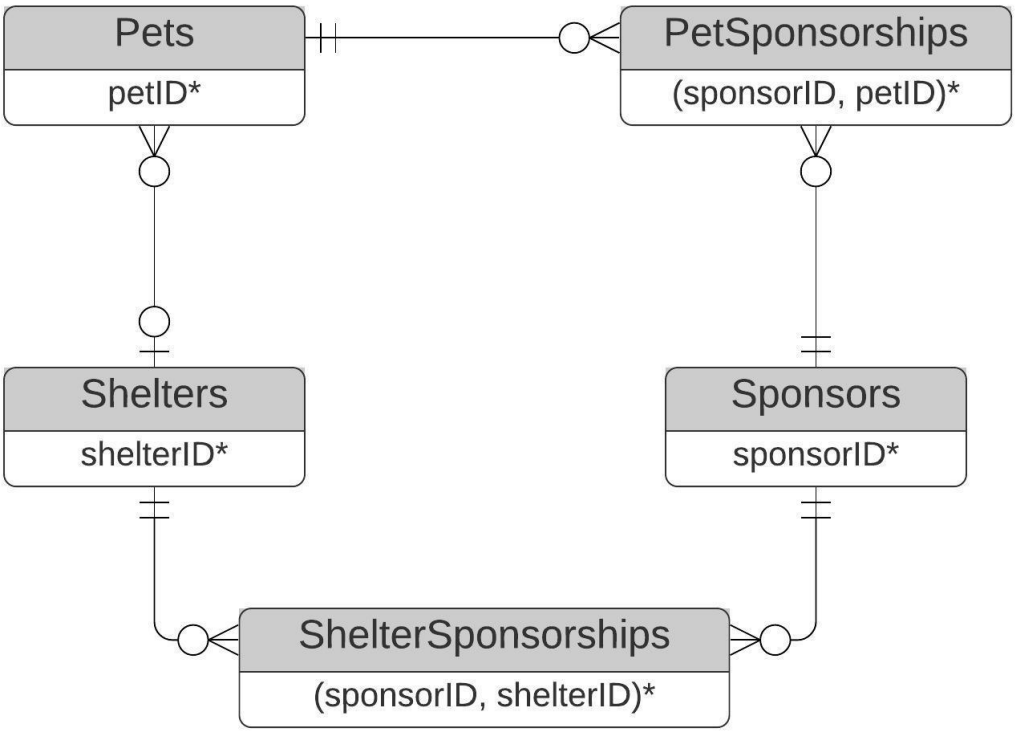
- Attributes
    - sponsorID: foreign key, not NULL
    - shelterID: foreign key, not NULL
    - (sponsorID, shelterID): primaryKey, not NULL
    - amount: numeric, not NULL (amount sponsor is donating for shelter per month)
    - beginDate: date, not NULL (date that sponsorship is initiated)
    - endDate: date (for sponsorships that have a specified end date)
- Relationships: Each sponsorship relates to one sponsor and one shelter.

    - a M:M relationship between Shelters and Sponsors is implemented with shelterID as a FK and sponsorID as a FK inside of ShelterSponsorships
    - a 1:M relationship between Sponsors and ShelterSponsorships is implemented using sponsorID as a FK inside of ShelterSponsorships
    - a 1:M relationship between Shelters and ShelterSponsorships using shelterID as a FK inside of ShelterSponsorships
- Deletions: A sponsorship is removed from the ShelterSponsorships database if the sponsor ends the sponsorship, the sponsored shelter is no longer accepting sponsors (shelter sponsorable field is set to False), or the current date is past the endDate

Shelters: Entity recording the attributes for the animal rescue/shelter registering for the app.

- Attributes:
    - shelterID: int, auto_increment, unique, not NULL, primaryKey
    - registrationDate: date, not NULL
    - name: varchar, unique, not NULL
    - streetAddress: varchar, not NULL
    - city: varchar, not NULL
    - state: char(2), not NULL
    - phoneNumber: varchar, not NULL
    - fax: varchar
    - email: varchar
    - sponsorable: boolean, not NULL (indicates whether shelter as an entity can be the sponsored party of a sponsorship)
- Relationships: Each shelter can be sponsored by any number of sponsors (including none). Each shelter can have any number of pets (including none). A shelter can be in multiple sponsorships (or none) but each shelter sponsorship contains one shelter.
    - a 1:M relationship between Shelters and Pets is implemented with shelterID as a FK inside of Pets
    - a M:M relationship between Shelters and Sponsors is implemented with shelterID as a FK and sponsorID as a FK using the ShelterSponsorships table
- Deletions: If a shelter can no longer be sponsored and has no pets in the Pets database, then it is removed from the Shelters database and any sponsorships associated with the shelter are also removed from the ShelterSponsorships database.

Petfunder ER diagram



Petfunder Schema