Ryan Howerton

11/11/2018

CS 340

Project Step 4 Final

URL: http://flip2.engr.oregonstate.edu:3436

Note that you must be on the OSU network.

Feedback by the reviewers:

“Hi Ryan,

Your site looks great! The form on the insert page works and I can see the new pokemon displayed after I submit. The links on the top of the pokedex page don't work, but I am sure you are aware of that! Looks like you also have to implement Inserts for types and moves, but since the pokemon page was done so well I am sure those will be quick work. Seem to be on the right track, keep it up!” – David Sahni

“Hi Ryan,

Create functionality works nicely for the pokemon.
I'm assuming the links that lead to 404 pages are ones that you are just currently working on.
Your INSERT queries look good.
I'm looking forward to seeing your other pages. Keep up the good work!” – Joseph Shin

“Hi Ryan,

Site looks great! I like that you have a bunch of attributes for each Pokemon, makes me miss playing it back in the day. And I am a fan of the search bar, it works well!

I did notice, however, that searching then going back to the original page was tedious since I had to press back several times. Don't mean to sound nit-picky.

Queries look good, and I'm looking forward to your next revision of the site.” – Patrick Kim

“What you have looks good, I'm going to assume your National Pokedex Draft homepage will eventually have a description of the site or something. I would set a blank box for secondary and have it as the default, since not all Pokemon have a second type. Good job having the add and delete work.” – Le-Chuan Chang

“Hi Ryan,

I love how easy this site is. The create functions worked great and I loved the inclusion of the search bar. There could be some minor user improvements, but nothing major.” – Forrest Hellard

“Hi Ryan. Good work here! It looks like you can add and delete for pokemon and the many to many relationship works well. It also looks like, as is the case for most of us, you have more work to do in order to display the rest of the information in the database, but you've got a good start! I'm sorry , but I deleted your pikachu. I put him back, but he's a little disfigured.” – Benjamin Alley

Actions based on the feedback:

1. I added all of the missing pages (except regions), and gave them full insert, delete, and search functionality.
2. I added a null box for Secondary Type that submits null to the database.

Upgrades to the Draft version:

N/A

a. **Fixes based on Feedback from Step 3:**

- The most major change from step 3 is that I moved my app onto the flip server section, as I didn't want to figure out if all the requisite packages worked on the ENGR web space, which does require being on the OSU network.
- I moved my navigation links to the top of the pages.

b. **Project Outline and Database Outline, ERD, and Schema:**

Pokémon Database

Project Outline:

I will be making a database based the Pokémon games (property of The Pokémon Company, which is co-owned by Nintendo, Game Freak, and Creatures, Inc.). In the main Pokémon video games, the player character is a child travelling the world (split into regions), completing some sort of challenge to become the best Pokémon trainer of the land. Pokémon are creatures with essentially what equate to superpowers that the player uses to battle other trainers to advance. There are now hundreds of Pokémon, each with their own abilities that players can find and add to a team, so this website is an attempt to organize them into a quick reference.

The website will allow a user to search up any Pokémon, and see information such as their type(s), stats, possible moves, and what regions the Pokémon can be found in. Every Pokémon has one or two types, a set of 6 numbers that define their stats in-game. Pokémon game focuses on a particular region, which is essentially just a different map for the player to explore, but many Pokémon can be found in different regions. Lastly, a user can search up a move to find its type, its power, and its category (described below).

Database Outline:

Pokémon: this will contain a list of every Pokémon currently available and is the core of the database. This list will have a relation with all other lists in the database. It has the following attributes:

- id: a unique number assigned to the Pokémon as it is added to the database. This will be the primary key.
- name: the name of the Pokémon in English. The name cannot be null, cannot be longer than 15 characters, and has no default.
- type1: the primary type of the Pokémon. This cannot be null but does not have a default. The value will be the id of the type as found in the Types table.
- type2: the secondary type of the Pokémon. This can be null and does not have a default. The value will be the id of the type as found in the Types table.
- hp: the first stat number, showing how many health points a Pokémon will have relative to others. This is an integer value that cannot be null.
- atk: the second stat number, showing how much physical attack power a Pokémon will have relative to others. This is an integer value that cannot be null.

- def: the third stat number, showing how much physical defense a Pokémon will have relative to others. This is an integer value that cannot be null.
- sp_atk: the fourth stat number, showing how much special attack power a Pokémon will have relative to others. This is an integer value that cannot be null.
- sp_def: the fifth stat number, showing how much special defense a Pokémon will have relative to others. This is an integer value that cannot be null.
- spe: the sixth stat number showing how fast a Pokémon is relative others. This is an integer value that cannot be null.

Types: this will contain a list of every type that can currently be found in the Pokémon games. It has the following attributes:

- id: a unique number automatically assigned to the type. This cannot be null, and auto-increments. This is the primary key.
- name: the name of the type in English. This is a string value that cannot be null and can be a maximum of 8 characters but does not have a default value.

Regions: the list of regions from every main Pokémon game.

- id: an automatically incrementing number assigned to each region. This number cannot be null and is the primary key.
- name: the name of the region in English. This is a string value that cannot be more than 6 characters.

Locations: Pokemon can be found in many different regions, and this lists what regions each Pokémon can be found in.

- rid: the id number of the region. This is an integer corresponding to the intended region. This cannot be null.
- pid: the id number of the Pokémon. This is an integer corresponding to the intended Pokémon. This cannot be null.

Moves: the list of all moves found across all Pokémon games.

- id: this is a unique number corresponding to each move. This number cannot be null and is automatically incrementing. This is the primary key.
- name: the name of the move in English. This is a character string that cannot be null.
- type1: the type of the move. This is an integer that corresponds to the id of the type of the move. Cannot be null.
- power: the power of the move. This is an integer number, which can be null.
- accuracy: the accuracy of the move. This is an integer number that can be null.
- category: the category of the move. This is an integer value that corresponds to the categories table, and cannot be null.

Categories: the possible categories of Pokémon moves.

- id: this is a unique auto-incrementing number to define the move. It cannot be null, and is the primary key.
- name: the actual name of the category. This is a string that should only be “Physical”, “Special”, or “Status”. Cannot be null.

The relationships in the database are:

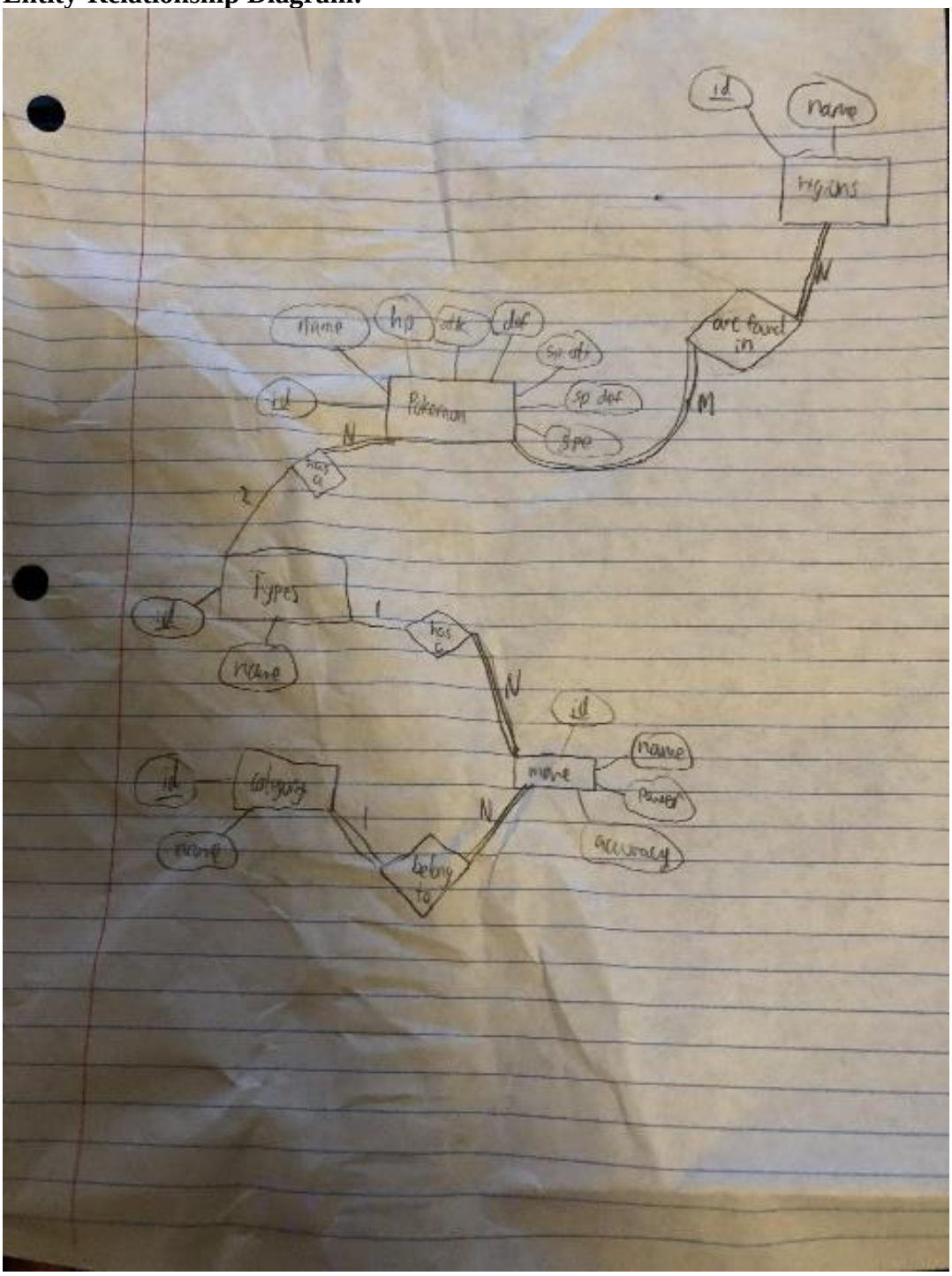
Pokemon have types: a Pokémon can be one or two types, but many Pokémon are the same type (if not the same combination). This is a one-to-many relationship, as a Pokémon can be one or two different types, but many Pokémon can be many different combinations of types.

Pokemon are from regions: a Pokémon are found in regions, but a region has many Pokémon. Sometimes regions have Pokémon that were previously found elsewhere, so this is a many-to-many relationship, as defined by the Locations table.

Moves have types: a move can have one, and only one, type. This is a one-to-many relationship as many moves can be the same type.

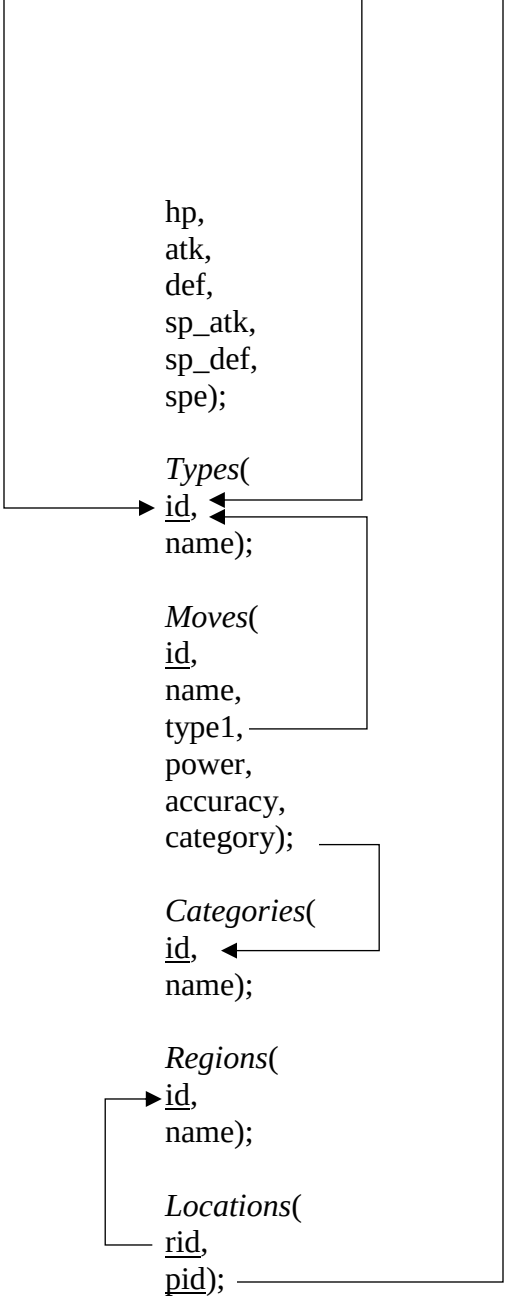
Moves have categories: a move can be one of three categories. This is a one-to-many relationship as all moves have to share the same categories.

c. Entity-Relationship Diagram:



d. Schema:

Pokemon(
<u>id</u>,
name,
type1,
type2,


hp,
atk,
def,
sp_atk,
sp_def,
spe);
Types(
id,
name);
Moves(
id,
name,
type1,
power,
accuracy,
category);
Categories(
id,
name);
Regions(
id,
name);
Locations(
rid,
pid);