Ryan Howerton

11/21/2018

CS 340

Project Step 5 Final

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

Note that you must be on the OSU network.

Feedback by the reviewers:

"Hi Ryan,
Your READ and DELETE functionalities for pokemon, types, and moves are working well.
Also, good job getting your select options for pokemon types and move types to dynamically load.
I wasn't able to add any moves, there seems to be some syntax error for your sql query to insert a move.
I tried searching for pokemon in your pokedex page, but I'm not sure if it is working because the list of pokemon isn't changing.
Your navigation links list (at the top) seem to change depending on what page you're on, which is fine, but when I'm in the moves page there is not link to the pokedex page.
Also, I'm assuming that your region maps page is still under construction.
Other than those things, your project is looking good. Keep up the good work." – Joseph Shin

"Hey Ryan,

The pokedex page implements the read and delete functions well, and the search works but only if you click "submit". If you try and search with the enter key, the page just reloads. Add, read, and delete all work for types, but only read and delete work for moves. Adding a move results in a page error. Things line up with your pdf, and you seem to be on the way to update functionality. Looks like the region map page still needs to be implemented, but you are on the right path!" – David Sahni

"Hi Ryan,

This website is coming along nicely. There are some minor issues, that should be easy to fix. I would go over your SQL, just because it will not allow the user to create a new move. Also, as someone else noted, the search bar functionality could be improved by allowing for the user to just press enter. The overall navigation could be improved, and there should be an easier way for the user to jump from page to page. Besides those minor issues, it looks good." – Forrest Hellard

"I also get a sql error when trying to add a new move. The sql error message just mentions the problem is near your insert query, but it looks fine as far as I can tell, so I'm betting it's something adjacent to it. The searches for a move by name and pokemon by name work well. Add and delete works for type as well as pokemon. The links for the region maps isn't working for me, giving me a 404 not found error. Same thing happens if you click search with nothing in the field probably since it just goes to "/search" and not "/search/:id". Also, view moves page has a link to view moves page and not pokedex so you have to go to type chart to get back to pokedex.” – Benjamin Alley

“I also am getting an error when adding a new move. Other than that, I think your site could benefit from a little more css, to make it slightly less cluttered and more readable. I'm thinking particularly of the form fields for adding new entities. A bit of margin between them would make it more attractive and usable. The columns with large numbers are also somewhat hard to read. Other than that, everything seems to work as expected and it seems to match your original plan!” – Sean Moss

“Hi Ryan,

So one thing I should note is you can make pokemon have negative numbers for stats and, as some of the data already in the table shows, very high stats. I'm pretty sure pokemon stats are capped at 0 to 255, so you may want to make a Javascript check for that. I also noticed that your Moves page doesn't support adding or updating, but it does support deleting (I deleted Pound to see if it would work, it does). You can't access the pokedex from the move page, so you should readd that.” – Le-Chuan Chang

Actions based on the feedback:

1. I fixed the issue with inserting a move into the moves list, it should now be working correctly.
2. I added error checking for adding a Pokémon, it should now no longer allow adding a pokemon with stats above 255 or below 1.
3. I fixed the link on the moves page to link back to the Pokémon page.
4. I am going to be working on adding some formatting to the page over the course of the next couple weeks.

Updates to the Draft version:

N/A

Draft PDF:

a. **Fixes based on Feedback from Step 4:**
   - I added all of the missing pages (except regions), and gave them full insert, delete, and search functionality.
   - I added a null box for Secondary Type that submits null to the database.

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:

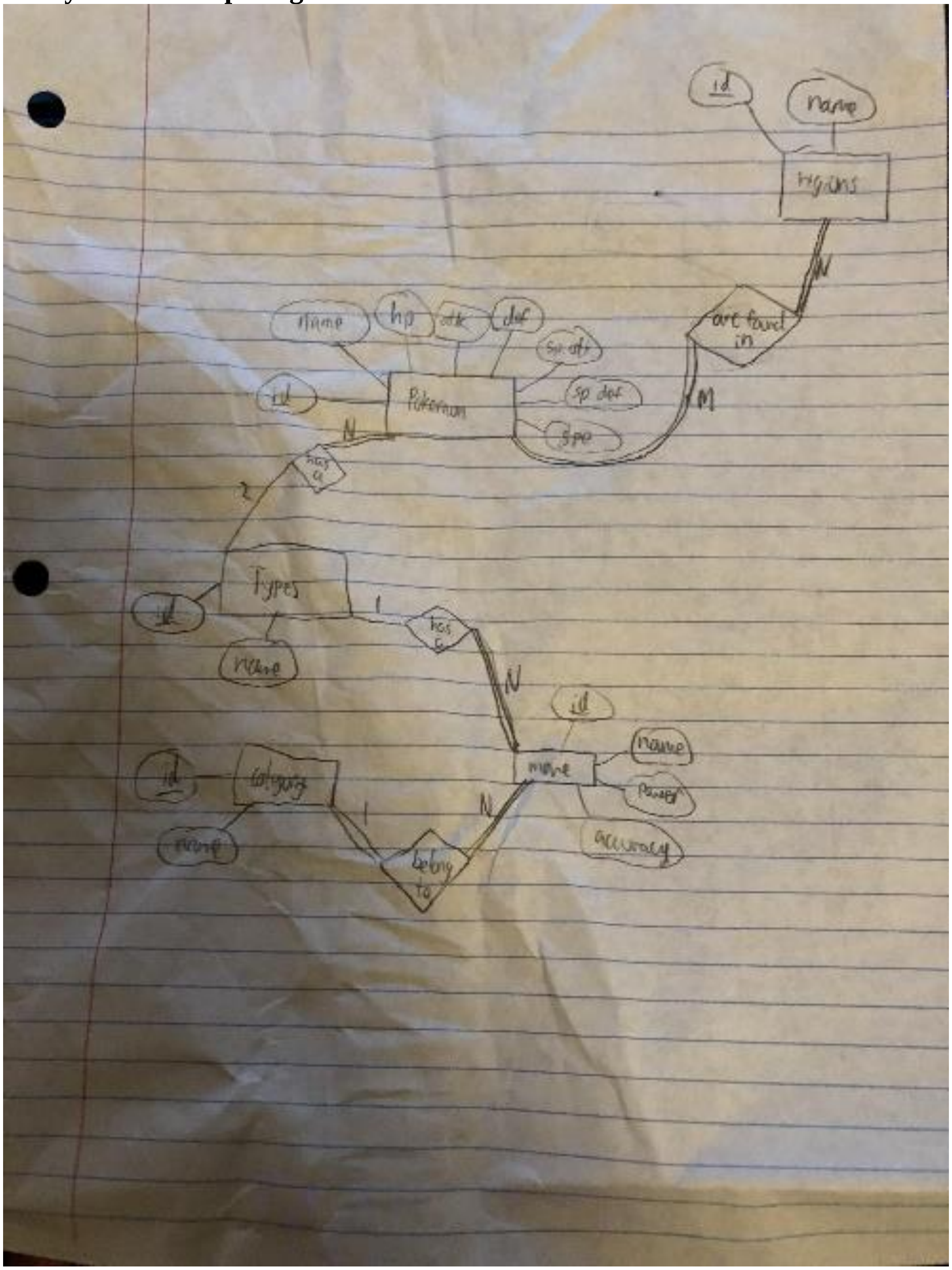
Pokemon have types: a Pokémon can be one or two types, but many Pokémon are the same type (if not the same combination). This is a one-to-many relationship, as a Pokémon can be one or two different types, but many Pokémon can be many different combinations of types.

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.

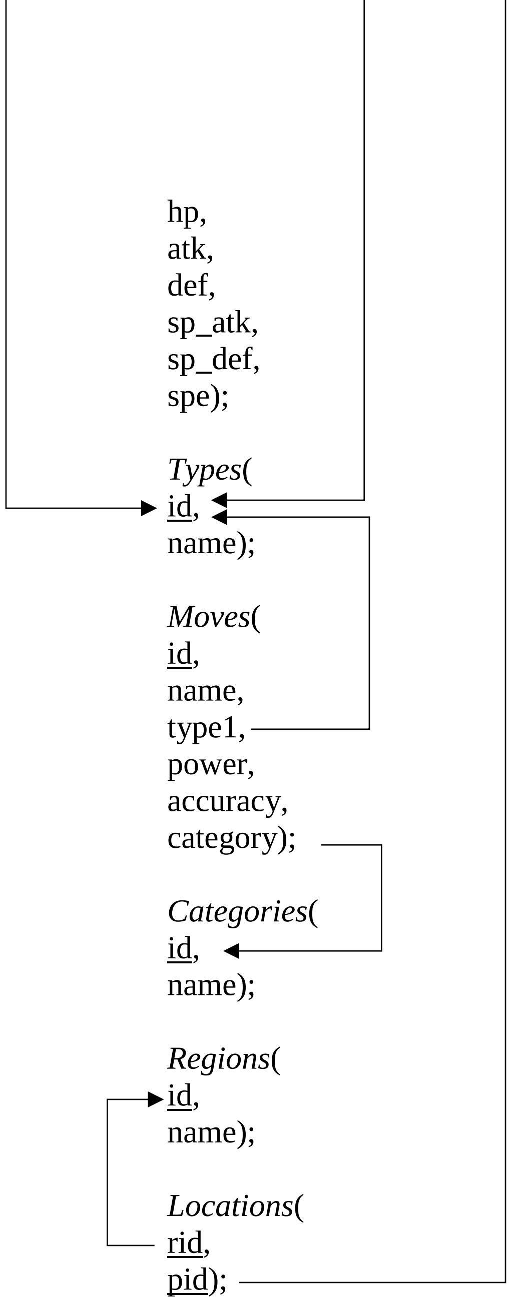
Moves have categories: a move can be one of three categories. This is a one-to-many relationship as all moves have to share the same categories.

c. **Entity-Relationship Diagram:**



d. **Schema:**

Pokemon(
id,
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);