Ryan Howerton

10/14/2018

CS 340

Project Step 2 Final

Feedback by Peers:

1. ERD: 1. The outline and ERD agree with one another. 2. I don't believe the pokemon HAS region participation is correct. (I could be wrong) but shouldn't a pokemon have at least one region, and a region have at least one pokemon? I think both should be total participation, not partial. Also review Types HAS A pokemon. Can a type exist without a matching pokemon? As drawn, it can. 3. Cardinality agrees with outline 4. I think the ER and outline are adequate SCHEMA: 1. Tables are defined and are present correctly. 2. Foregin keys are defined and put in place correctly. 3. Yes, attributes match 4. I think the database design is adequate and any improvements that I can think of don't add anything to the projects requirements. DDQ: The data types, tables, and keys all make sense and follow the outline as given. The SQL syntax also appears correct. Great work!
2. I don't see locations in the ERD, how is it related to the other entities. Since Pokemon and type have a many to many relationship in the schema create a table for this relationship. Same goes for the Pokemon and region relationship. Create those additional tables tables in SQL.

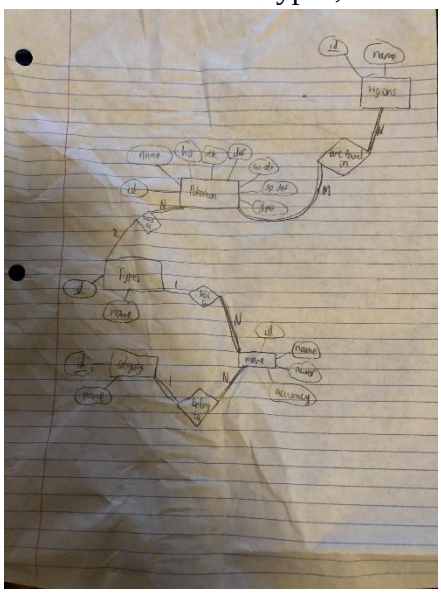
Actions based on Feedback:

I changed the participation between Pokémon and regions to be total in both directions as suggested by the first review. I also changed the relationship between Pokémon and types to be one-to-many/many-to-one as suggested by the second review.

I did not change the Pokémon:type relationship, as types can exist without Pokémon (moves can have types) but Pokémon cannot exist without types. I also did not create an extra table for locations, because it already exists in the “are found in” relationship. I could change one name or the other, but the table does exist.

Upgrades to Draft:

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.



← Double lines to denote total participation (pardon the crumpling, I didn’t realize I dropped the paper and when I move my chair back I rolled over it).

a. **Fixes based on Feedback from Step 1:**
   - I updated the Project Outline by adding more information about the source material.
   - I took away the category table and absorbed that into the move table.
   - I updated the "Pokemon are from regions" category to explain why the Locations table is composed only of two ID numbers.
   - I changed the attribute 'type' in the Types table to 'name' to avoid conflicts.
   - I added the attribute 'type1' to moves because I forgot it previously.
   - I added the attribute 'accuracy' to moves because that matters to Pokémon.

b. **Project Outline and Database Outline – Updated Version:**

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:

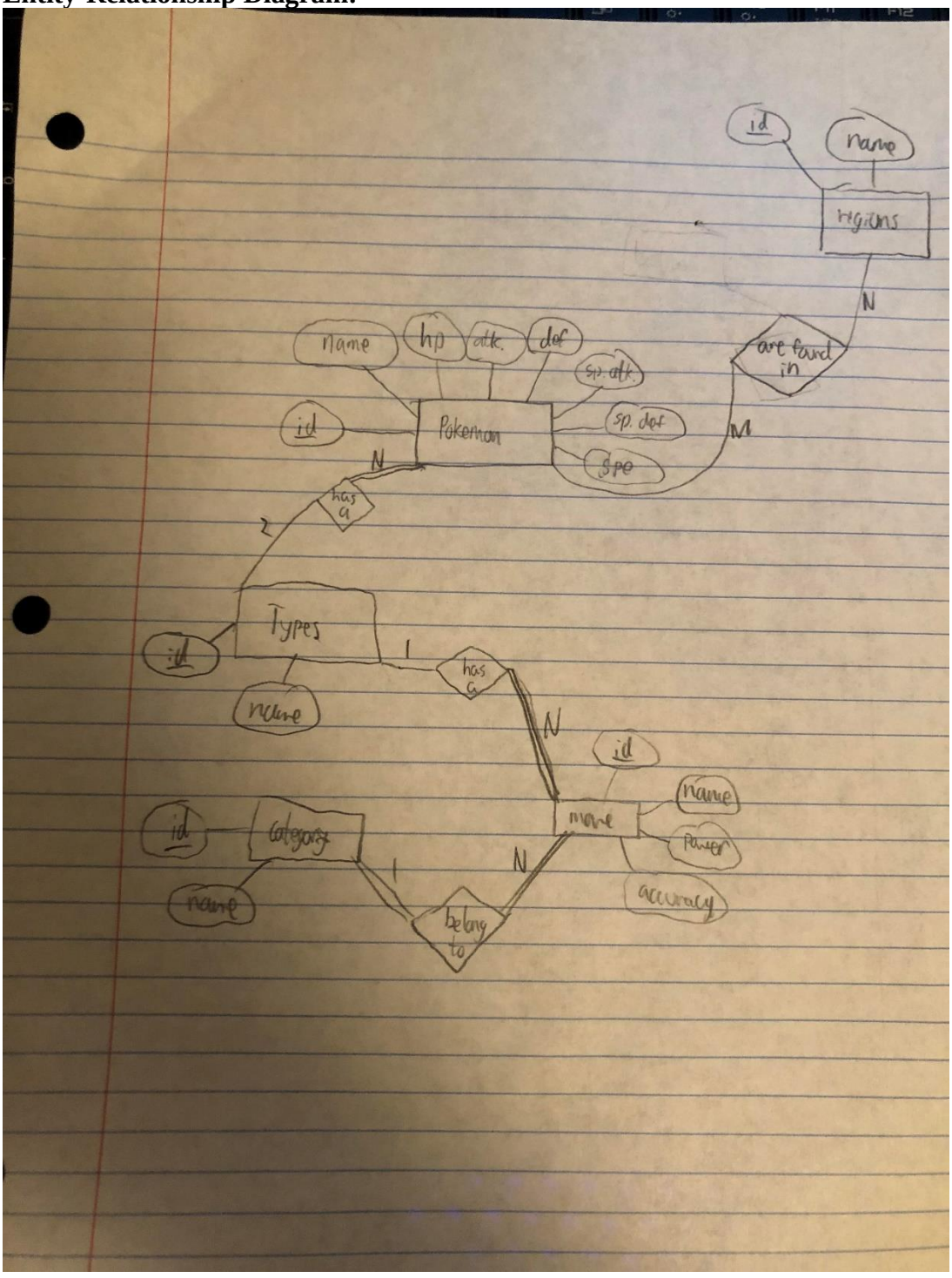
Pokemon have types: a Pokémon can be one or two types, but many Pokémon are the same type (if not the same combination). This is a one-to-many relationship, as a Pokémon can be one or two different types, but many Pokémon can be many different combinations of types.

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.

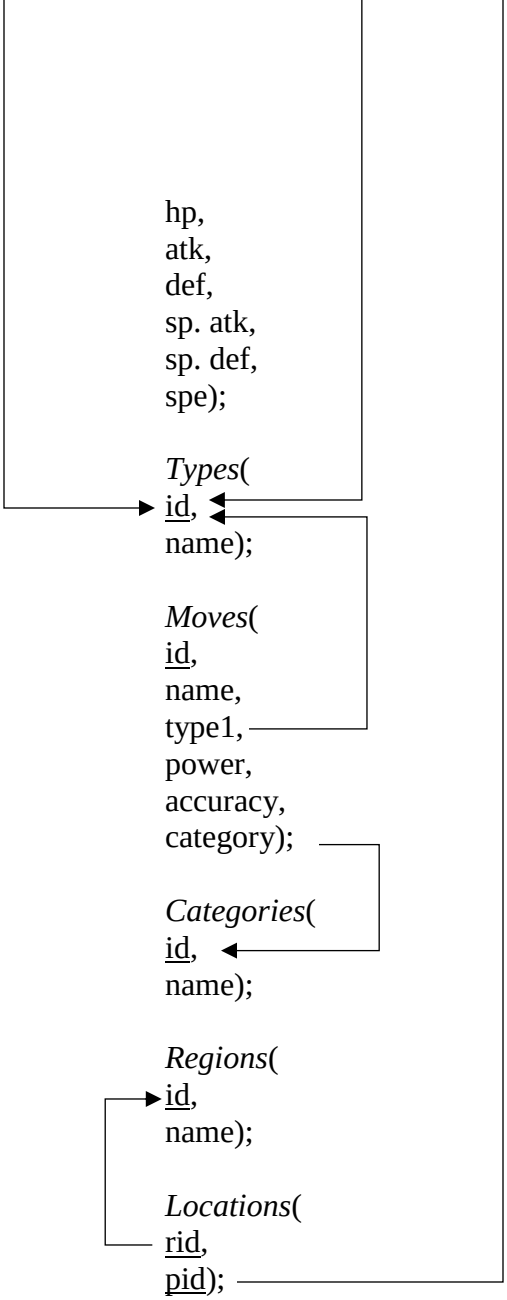
Moves have categories: a move can be one of three categories. This is a one-to-many relationship as all moves have to share the same categories.

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);