Pokémon Emerald Database

Jacob Janzen
janzenj2@myumanitoba.ca

Daniel La Rocque
larocq17@myumanitoba.ca

Jared Webber
webberj1@myumanitoba.ca

Colin Johnson
johns233@myumanitoba.ca

December 3, 2021

1 Data Summary

Pokémon was fresh on our minds when we began the project because our instructor, Adam Pazdor, had brought up his interest in the game several times during class. We very quickly realized that a database containing information about Pokémon would be a great way to have lots of data with very interesting relationships.

We ultimately decided to constrain our idea to one specific game in the franchise while building our data model because we wanted to track data about non-player characters and locations in the games which would quickly get very complicated when discussing multiple games. It would introduce even more complications when considering the fact that many relationships between entities would change depending on the specific game. To solve this, we chose one specific game rather arbitrarily: Pokémon Emerald.

The Data consists of the set of Pokémon that are present in Pokémon Emerald, the locations, moves, and non-player *trainers* in the game, as well as the *types* that Pokémon and moves can have. We tracked the methods in which each Pokémon can learn each move as well as the location of all non-player characters and the locations where each Pokémon can be found.

In the end, we had a 1.4 MB database. Our largest table, `Learns`, had 15280 records while our smallest table, `HM`, had only 8. In total, there were 24547 records in the database.

2 Data Model

We gathered data based on the data model we created, which was created from brainstorming every type of data we could gather about the game that we deemed interesting and then linked together in every way that we could think of.

Most of the challenges in creating the data model were in fine tuning the details after working out exactly which data we would use. For instance, originally the `Learns` table contained every move that every Pokémon could learn without any distinction of method. Later, we decided to add the method and split it into one relationship for each method in which a Pokémon could learn a move. This added a lot of unnecessary complications to the data set and made queries far more difficult than they needed to be. Thus, we combined them back into a single relationship with a method attribute being a primary key on the relationship. Unfortunately, learning a move by breeding is a little more complicated than other methods of learning a move. It is dependent not only upon the Pokémon learning the move and the move itself, but also the father of the Pokémon learning the move. Therefore, we separated that ternary relationship between two Pokémon and move from the rest of the learn methods.

Another challenge that came up in the project was the relationship between a trainer and the Pokémon that they used in their team. Originally, we had `Team` as a weak entity that depended on `Trainer` and `TeamMember` as a weak entity that depended on `Team`. This two-layer weak entity added a lot of unnecessary complication, so we changed `TeamMember` to be a relationship instead which has an ID as a part of its primary key.

A tricky participation ratio was surrounding the `HasTypes` relationship. The ??? type which existed in Pokémon Emerald was a type only used for the move *Curse*. Thus, we had only partial participation from `Type` to `Pokemon` and total participation from the other side because every Pokémon has at least one type and can have two.

A tricky cardinality ratio is through the `EvolvesFrom` relationship. While generally, when a Pokémon evolves, it can only evolve into one Pokémon, there are rare situations where one Pokémon can evolve into multiple different ones. For instance, *Eevee* can evolve into *Vaporeon*, *Jolteon*, *Flareon*, *Espeon*, or *Umbreon* depending on the specific circumstances that it evolves under. This relationship does not work in the opposite direction. There are no cases where one Pokémon evolves from two different Pokémon in Pokémon Emerald.

3 Database Summary

Table	Cardinality	Arity
Pokemon	386	17
Abilities	610	2
EggGroups	504	2
Location	106	1
Move	354	6
Trainer	493	3
Type	18	2
FoughtAt	520	2
FoundAt	647	6
HasTypes	557	2
HM	8	2
Learns	15280	3
LearnsByBreeding	2007	3
Team	887	3
TeamMember	1795	6
TM	50	2
Effectiveness	324	3

4 Queries

5 Interface

add details about how to set up