**COMP-2702 – Data Management  
Final Project Proposal**

**Project Name: Pokémon Viability Analysis**

**Project Date: 16th of April 2025**

**Student Name: Cedric Parent**

# Project Description

Pokémon has always fascinated me with how vast a seemingly simple and childish game can truly be. As a kid I never got the chance to fully understand all the complexities hidden beneath the games’ simple looking exterior and that changes now, I will analyze all the intricacies and between Pokémon’s typing, the generation they originally came from as well as analyzing what happens when you run a test that emulates 50,000 Pokémon going head-to-head. I plan to make various charts and graphics that are not only pleasing to the eye, but ones that also capture the essence of what makes Pokémon the game it always should be recognized as, a statistical masterpiece.

This project is especially “needed” for the competitive audience of Pokémon as, I’ve found that, over the years, Pokémon games have continued to increase the complexity of game mechanics which makes it harder and harder every year to introduce new players to start playing competitively as, if they had an option between Pokémon, a very complex game with too many options for gameplay, and a simpler competitive game such as Marvel Rivals, I guarantee, without a doubt, that most people would choose to play the simpler game. Therefore, as this project can be used to identify key team members and which Pokémon seem to rise above others, it can easily be used by players who are interested in getting into the competitive scene of Pokémon and will help them in understanding what goes into the process of figuring out who are viable team members for competitive play.

As I mentioned above, I’ve always had a soft spot for Pokémon (no matter how bad the game may be from an objective viewpoint) mainly because of how many different combinations of possible events occurring even in something as simple as a trainer battle. There’s an enormous number of factors that contribute to this: stats; move pools; typing; abilities; level; generation; different forms or variants; status effects, and the list just goes on and on. In this project, I will be focusing on four main factors:

1. Pokémon’s effectiveness and win percentage when going head-to-head with another Pokémon.
2. Pokémon’s types and which types are more (or less) advantageous
3. Pokémon’s Stats: such as which stats are most (or least) advantageous
4. Pokémon’s generation of origin

# Link to Dataset: <https://github.com/rpdieego/IndigoTortoise-pkmn-notebooks/blob/master/datasets_2619_4359_tests.csv>

# Business Rules/Assumptions

* + Multiple Pokémon can have the same type and/or type combination
  + Multiple Pokémon can have the same stats in some areas (ex: Magcargo and Slugma’s Speed can both be 50)
  + Multiple types can be super effective (x2) , not very effective (0.5x) or ineffective/unusable (x0) against the same type
  + Multipliers on damaging moves do stack, for example a grass type move against a ground and water would be super effective against both types and thus make it do x4 damage, the same can be said for resistances, a grass move against a water and fire type would be x1 damage (0.5 \* 2)
  + There are multiple instances of the same Pokémon fighting each other and sometimes result in different winners

# A computer screen shot of a computer AI-generated content may be incorrect.Entity Relationship Diagram (ERD)

ERD

The Poke\_id PK is simply the ID number associated with each unique Pokémon, and as a result, is unique by nature, it has three relationships all going to the combat\_matchups table (which essentially gives a name to all of the IDs in the 50,000 simulated 1v1 battles) it is one to many as one Pokémon ID can show up many times as the first, second and winner of each battle.

The type1\_id and type2\_id are foreign keys inside of the pokemon\_info table originally from the type\_id PK from the types table. They work so well because each Pokémon has one primary type, and one optional secondary type. The reason the relationship between type\_id and type1\_id and type2\_id is one to many is because there is only one of each type, but, multiple Pokémon can have the same primary, secondary, or even share both types with a multitude of other Pokémon.

Similarly, from type\_id to attacking and defending type\_id, there is yet again a one-to-many relationship, as fire type for example, can appear as a defending type as well as an attacking type, however, each attacking\_type\_id or defending\_type\_id, only refers to one type.

The battle\_id PK is also quite simple yet important, it assigns a number to each battle, identifying each battle uniquely, as if it didn’t exist and I were to remove duplicates, it would potentially remove a set of records such as:

25(Pikachu) 1(Bulbasaur) 25(Pikachu as the winner)

25(“”) vs 1(“”) = 25(“”)

And removing these records would unknowingly change the final value of each Pokemon’s win rate and skew my analysis. However, with the battle\_id PK:

1(battle\_id) 25(Pikachu) 1(Bulbasaur) 25(Pikachu as the winner)

2(battle\_id) 25(“”) 1(“”) 25(“”)

These two records are in fact distinct from one another and would not be deleted upon deleting duplicate records.

# Power BI Dashboard

Line Chart: Show the average stat total by Pokémon type.

Donut Chart: Show win rates from the test data by Pokémon type with slicers to fit for Pokémon typing

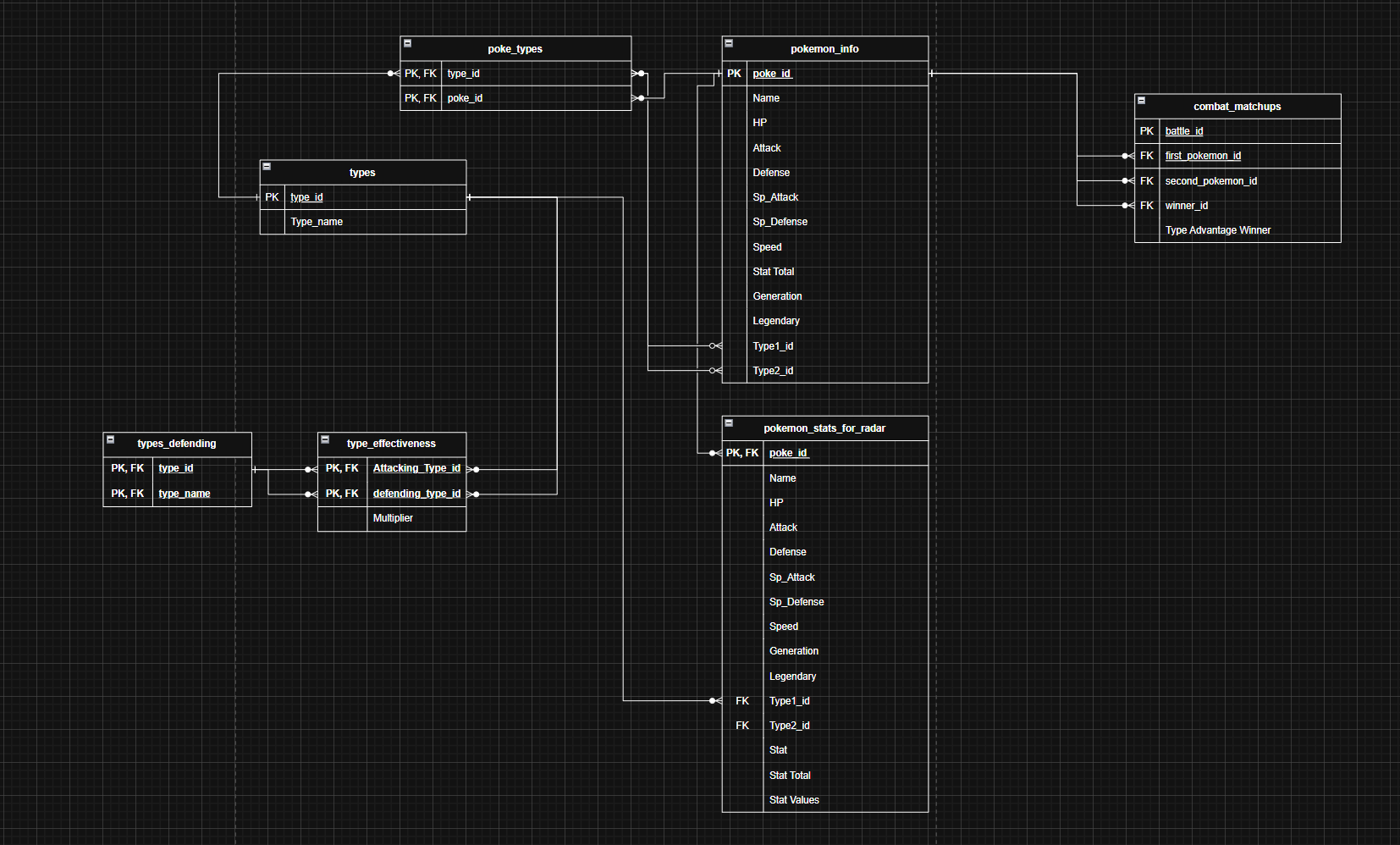
Type matchup matrix with rows as the attacking Pokémon type and columns as the defending Pokémon

A gauge that shows the win rates of a selected Pokémon using DAX to make it so that when you select a Pokémon, it selects their ID, and divides their wins by how much they appear

Decomposition Tree that shows what factors influence battle wins the most by using type\_name, legendary and generation columns

Radar Chart representing the average stats with slicers for Types, Legendary status and Generation of origin.

Annex:



I added quite a few measures as well as a helper table (with no relationship to the other tables in the model). However, the tables I did add to the model are the following:

Types Defending: I added this table to differentiate the type\_name used by the type\_effectiveness table through its relationship with the types table as, if I didn’t make a new way to fetch a type\_name through the type\_effectiveness table, the type effectiveness matrix would not be able to work. It is essentially an extra types table that refers to the defending type instead of the attacking type.

Pokémon Stats for Radar: This table is almost a carbon copy of the pokemon\_info table except for a few small changes. Firstly, I added the stat and stat values column which gives each pokemon’s stat name in the stat column and the value of that stat in the stat values column. This made it possible to create the radar chart by making measures that calculated the average stat by individual Pokémon, instead of just the average across the board like the original columns (HP, Attack, etc.). These columns also made it possible to filter results directly based off Pokemon’s features such as Typing, Legendary status and generation as the stats are individual row values, instead of just being columns next to pokemon.

Poke Types: This table serves as a bridge table between the Pokemon info and the types table. It has a composite primary (and foreign) key of Poke\_id and type\_id, which by extension made it possible to create the type1\_id and type2\_id columns which distinguish between Pokemon’s two types. This table also classifies Pokemon by their type and reduces the amount of redundant data that was originally in the pokemon\_info table (type1\_name and type2\_name).