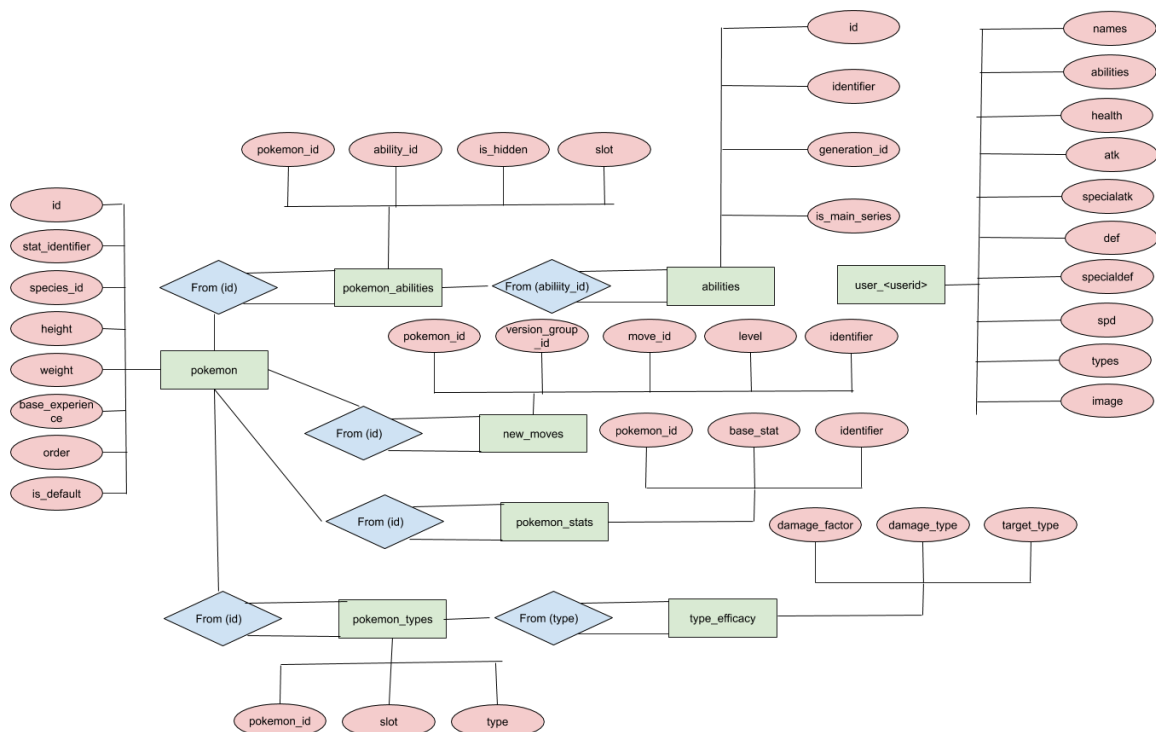


Our project was to create a pokedex bot for discord using discord.js, a node library for accessing the discord api, and MariaDB. The project had one main focus, which was the displaying of the pokemon from the series with various information, and one sort of side focus, allowing users to create their own custom pokemon. Our main goal was to make sure to show users pokemons stats, abilities, types, and type efficiencies. Here is the table layout: (if too small you can go to this link <https://docs.google.com/drawings/d/1obZTm2ROOaOKEsyCuaZ5coJWG4fsZHIYToPCaIAHpsQ/edit?usp=sharing>)



We met most of our goals even if some were rather rushed and could have been implemented better. We gave users almost all the information they would normally need when playing a pokemon game. The only thing that could have been better with the normal pokemon was including evolution lines. We just lacked the time overall to implement evolution lines. For us to

implement this we would have had to recursively query our database until we reached the end of the evolution line. As for the custom pokemon we did succeed in everything we wanted for them but we cut a lot of corners to get to that point. For time reasons we decided to implement individual tables for each user rather than just managing two tables, one with userids and the other with the custom pokemon in it.

I think the first goal of future work would be to refactor a lot of the code and adjust the procedure for custom pokemon. Right now the code base for this is 1000 lines long in a single javascript file, ideally this would be broken up by each command, there are also a few functions which could be generalized more. As for the procedure for custom pokemon, we would want to change it to the format which was mentioned in the goals portion, using 2 tables instead of individual tables. After those two parts are changed the next most important thing would be to figure out how to secure the custom pokemon from sql injection more. As it stands we do have some protection from sql injection but there is definitely more that could be done. After all these general changes I think future work would focus on just adding a bit more information to the pokemon and shiny command overall as well as overhauling the moves command to make use of embeds and paging using discord reactions. If there was more to be done beyond this, it would best be done using the api we got the csv files from instead of using the database at all.

Overall, it was nice being able to use the discord api and do something different from the rest of the class in general. For many of the members in our group this was a new experience but we quickly adjusted to using the api as we coded. The library we used to connect to the discord api, discord.js, has good documentation which makes it easy for even beginners to understand. As for MariaDB, we were all pretty used to it at this point in the class after using it for the entire semester. I, Ethan, will say that it probably would have been better to use

MongoDB in hindsight as it would have made it easier to get things such as evolution lines.

Overall the result of our project seemed to be good and it was enjoyable to make.