Database Final Project Phase 1

Katie Chang

Alec Tabatchnick

1. Target Domain
   1. Pokemon database regarding pokemon, their abilities, statistics, locations in game, evolutions, the items available and a variety of other things.
   2. Allow a user to become a trainer and view or add pokemon to a roster
   3. Do this using Java, JDBC, MySql, and spark to create a web app or a swing gui.
2. Comprehensive + representative list of questions this system should answer (15+)
   1. How many different pokemons have a given ability, grouped by generation?
   2. How does a given pokemon evolve, if at all?
   3. In which region can I catch a given pokemon?
   4. Which pokemon from each region has the best total base stats?
   5. Which pokemon is appears in all regions, if there is one?
   6. Select pokemon that is both a fire and ground type.
   7. Select the name and number of all pokemon with an encounter rate below 5%.
   8. Select the name and type of pokemon with more than 1 evolution.
   9. Select the name and type of pokemon who weigh more than 20 pounds and are of the fire type.
   10. Select the type of all pokemon who can use the move tackle.
   11. Select the region that has the most unique pokemon.
   12. Select all third-evolution pokemon that are of type Water AND Dark.
   13. Which pokemon type is the most popular? (the type with the most pokemon belonging to it)
   14. Pick all water pokemon, sort by pokedex number.
   15. Select the pokemons that require a held item in order to evolve.
3. Design and show a relational data model that you plan to use for your system.
   1. Tables:
      1. abilities
         1. CREATE TABLE abilities   
            (  
            id INT NOT NULL,  
            identifier varchar(32) NOT NULL,  
            generationId INT NOT NULL,  
            is\_main\_series INT NOT NULL,  
            PRIMARY KEY(id)  
            );
      2. ability\_prose (description of ability)
         1. CREATE TABLE ability\_prose   
            (  
            ability\_id INT NOT NULL,  
            short\_effect varchar(256) NOT NULL,  
            PRIMARY KEY(aiblity\_id)  
            );
      3. encounters
         1. CREATE TABLE encounters (

id INT NOT NULL,

version\_id INT NOT NULL,

location\_area\_id INT NOT NULL,

encounter\_slot\_id INT NOT NULL,

pokemon\_id INT NOT NULL,

min\_level INT NOT NULL,

max\_level INT NOT NULL

);

* + 1. pokemon\_evolution
       1. CREATE TABLE pokemon\_evolution(

id INT NOT NULL,  
 evolved\_species\_id INT NOT NULL,

evolution\_trigger\_id INT NOT NULL,

trigger\_item\_id INT ,

minimum\_level INT ,

gender\_id INT ,

location\_id INT ,

held\_item\_id INT ,

time\_of\_day VARCHAR(5) ,

known\_move\_id INT ,

known\_move\_type\_id INT ,

minimum\_happiness INT ,

minimum\_beauty INT ,

minimum\_affection INT ,

relative\_physical\_statsINT ,

party\_species\_id INT ,

party\_type\_id INT ,

trade\_species\_id INT ,

needs\_overworld\_rainINT NOT NULL,

turn\_upside\_down INT NOT NULL

);

* + 1. evolution\_triggers
       1. CREATE TABLE evolution\_triggers (  
          id INT NOT NULL,  
          identifier varchar(9) NOT NULL  
          );
    2. evolution\_chains
       1. CREATE TABLE evolution\_chains (  
          id INT NOT NULL,  
          baby\_trigger\_item\_id INT   
          );
    3. generations
       1. CREATE TABLE generations (  
          id INT NOT NULL,  
          main\_region\_id INT NOT NULL,  
          identifier varchar(32) NOT NULL  
          );
    4. moves
       1. CREATE TABLE moves (  
          id INT NOT NULL,  
          identifier varchar(32) NOT NULL,  
          generation\_id INT NOT NULL,  
          type\_id INT NOT NULL,  
          power INT NOT NULL,  
          pp INT NOT NULL,  
          accuracy INT NOT NULL,  
          priority INT NOT NULL,  
          target\_id INT NOT NULL,  
          damage\_class\_id INT NOT NULL,  
          effect\_id INT NOT NULL,  
          effect\_chance INT NOT NULL,  
          contest\_type\_id INT NOT NULL,  
          contest\_effect\_id INT NOT NULL,  
          super\_contest\_effect\_id INT NOT NULL  
          );
    5. pokemon
       1. ID - primary key
       2. CREATE TABLE pokemon (  
          ID INT NOT NULL,  
          identifier varchar(32) NOT NULL,  
          species\_id INT NOT NULL,  
          height INT NOT NULL,  
          weight INT NOT NULL,  
          base\_experience INT NOT NULL,  
          order INT NOT NULL,  
          is\_default INT NOT NULL  
          );
    6. pokemon\_colors
       1. CREATE TABLE pokemon\_colors(  
          id INT NOT NULL,  
          identifier varchar(10) NOT NULL  
          );
    7. pokemon\_evolution
       1. CREATE TABLE pokemon\_evolution (

id INT NOT NULL,  
evolved\_species\_id INT NOT NULL,  
evolution\_trigger\_id INT NOT NULL,  
trigger\_item\_id INT ,  
minimum\_level INT ,  
gender\_id INT ,  
location\_id INT ,  
held\_item\_id INT ,  
time\_of\_day varchar(9) ,  
known\_move\_id INT ,  
minimum\_happiness INT ,  
minimum\_beauty INT ,  
minimum\_affection INT ,  
relative\_physical\_stats INT ,  
party\_species\_id INT ,  
party\_type\_id INT ,

trade\_species\_id INT ,

needs\_overworld\_rain INT NOT NULL,  
 turn\_upside\_down INT NOT NULL  
 );

* + 1. pokemon\_habitats
       1. CREATE TABLE pokemon\_habitats (  
          id INT NOT NULL,  
          identifier varchar(32) NOT NULL  
          );
    2. pokemon\_abilities
       1. CREATE TABLE pokemon\_abilities (  
          pokemon\_id INT NOT NULL,  
          ability\_id INT NOT NULL,  
          is\_hidden INT NOT NULL,  
          slot INT NOT NULL  
          );
    3. pokemon\_types
       1. CREATE TABLE pokemon\_type (  
          pokemon\_id INT NOT NULL,  
          type\_id INT NOT NULL,  
          slot INT NOT NULL  
          );
    4. regions
       1. CREATE TABLE regions (  
          id INT NOT NULL,  
          identifier varchar(8) NOT NULL  
          );
    5. types
       1. CREATE TABLE types (  
          id INT NOT NULL,  
          identifier varchar(16) NOT NULL,  
          generation\_id INT NOT NULL,  
          damage\_class\_id INT   
          );
    6. type\_efficacy
       1. CREATE TABLE type\_efficacy (  
          damage\_type\_id INT NOT NULL,  
          target\_type\_id INT NOT NULL,  
          damage\_factor INT NOT NULL  
          );

1. Write SQL statements that will implement a representative sample of your target queries
   1. *Pick all water pokemon, sort by pokedex number.*
      1. SELECT pokemon.ID, pokemon.name FROM type\_names, pokemon, pokemon\_types WHERE type\_names = “water” AND type\_names.id = pokemon\_types.type\_id AND pokemon\_types.ID = pokemon.dexNum ORDER BY pokemon.ID ASC;
   2. *How many different pokemons have a given ability (say levitate), grouped by generation?*
      1. SELECT gen.num, count(\*) FROM abilities, pokemon\_abilities, generation, pokemon\_generation, pokemon WHERE abilities.name = “levitate” AND abilities.id = pokemon\_abilities.ability\_id AND pokemon\_abilities.poke\_ID = pokemon.ID AND pokemon.ID = pokemon\_generation.pokemon\_id AND pokemon\_generation.generation\_id = generation.id GROUP BY generation.id;
   3. *In which region can I catch a given pokemon? (say Pikachu)*
      1. SELECT region.identifier FROM regions, locations, encounters, pokemon WHERE pokemon.name = “Pikachu” AND pokemon.ID = encounters.pokemon\_id AND encounters.location\_area\_id = locations.id AND locations.region\_id = regions.id;
   4. *How does a given pokemon evolve, if at all? (say Pikachu)*
      1. SELECT \* FROM pokemon, pokemon\_evolution WHERE pokemon.identifier = “Pikachu” AND pokemon.id = pokemon\_evolution.id;
   5. *Select the types of all pokemon who can use the move tackle.*
      1. SELECT types.identifier FROM pokemon, pokemon\_moves, moves, pokemon\_types, types WHERE moves.identifier = “tackle” AND moves.id = pokemon\_moves.move\_id AND pokemon\_moves.pokemon\_id = pokemon.id AND pokemon.id = pokemon\_types.pokemon\_id AND pokemon\_types.type\_id = types.id GROUP BY types.identifier;
2. Provide a proposal of how you will load the database with values / data
   1. Using available csv’s:
      1. <https://github.com/veekun/pokedex/tree/master/pokedex/data/csv>
3. Potential conversion issues:
   1. Converting from CSV to SQL may require some research
   2. Use SQL Server Management Studio
   3. We might have to delete some values from the csv’s because there is too much info in the ones we found. In this case we will have to find a way to either remove them or ignore them when we import the tables.
4. Briefly describe a type of reports you plan to generate or any special user interface issues (eg views) that you plan to implement
   1. View of all pokemon’s names within a user’s (trainer’s) roster
   2. View of a given pokemon’s stats
5. What are the specialized / advanced topics you plan to focus on in your database design?
   1. Java and JDBC to connect to database
   2. Spark and Jquery to connect to a bootstrap webapp
   3. Essentially creating a full stack web app that allows for queries and creating a roster of pokemon.
6. Describe the DB platform you plan to use, including any relevant implementation details or challenging issues
   1. MySql connected to Java using JDBC with a swing gui interface all run on one computer.