

Database Project Final Report: Pokedex

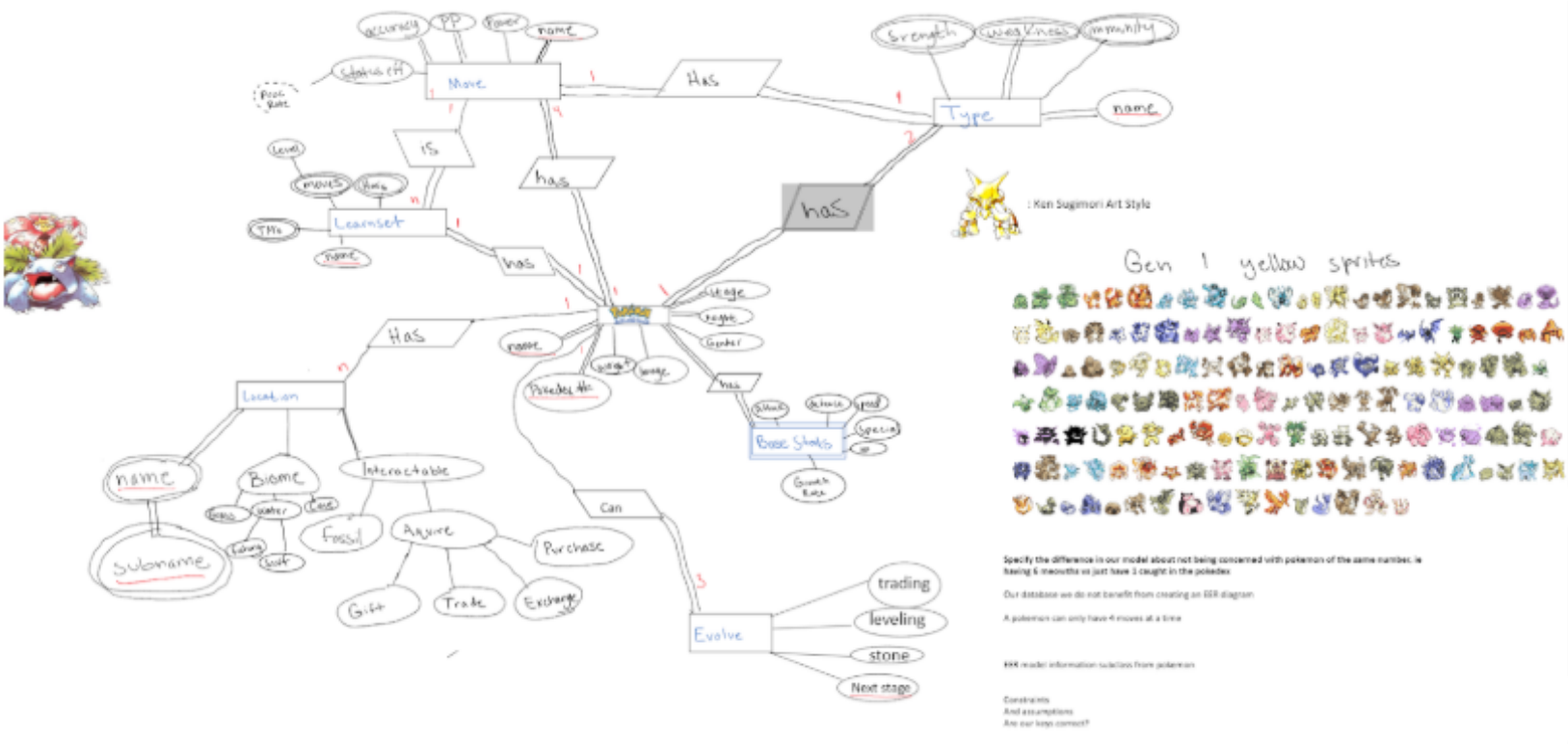
6 May 2019

Jacob Alongi, Andrew Kirkham, Samuel Bishop

Revised Problem Statement: Our proposed database application is an application that would catalog and sort generation one pokemon by multiple criteria, including: types, moves, evolutions, locations, learnsets. The target audience for this application would be anyone who is currently playing through a pokemon game (all the original gen 1 games including Red, Yellow, Blue, Green and the remakes Fire Red and Leaf Green as well as Pokemon Go or a plethora of spin off games), our application would make almost any situation regarding pokemon easier and more navigable.

This database would be essential to the new pokemon player, an experienced pokemon player, or aspiring competitive pokemon RPG players. This database is necessary because although pokemon was designed to be a kids game, the mechanics surrounding it are surprisingly complex. Whether you are a trying to finish the game as fast as possible for a popular pokemon speedrun competition, or you're trying to make the most out of the surprisingly convoluted battle system, our database would have functionality that enables the user to not only complete a pokemon generation one game, but play it at a high level.

Revised ER Diagram



Logical database design:

Pokedex: Individual owners Pokedex info

Primary Key: Owner_ID: if this becomes a database of multiple pokedexes then the Owner_ID

would uniquely identify the pokedex

Caught: how many pokemon the trainer has caught

Seen: how many have been seen, pokemon that are caught have been seen

Owner_ID	Caught	Seen
Jacob56789	4	20
(String)	(Integer)	(Integer)

Pokemon: Table of all the data displayed about a pokemon in a pokedex entry

Primary Key: Pokedex#: each pokemon only has one pokedex number and can be uniquely identified by it

Pokemon_Name: the name of the pokemon, can also be uniquely identified by its name but not necessary

Type1/2: Each pokemon can have up to two types but no less than one. The type can be used by the TYPE table below to calculate the damage multipliers of attacks

Stage: pokemon can be either Basic, Stage 1 or Stage2.

Gender Ratio: the number is the ratio of female pokemon to male pokemon. This can be derived from Gender_Ratio

Height: the height of a pokemon

Weight: the Weight of a pokemon

Description: when a pokemon is caught the description is populated in the pokedex with the corresponding description

Category: may be irrelevant but was in the original games. May have later been implemented to decide the pokemon that could breed with each other

Pokedex#	Pokemon_Name	Type1	Type2	Stage	Gender_Ratio	Height	Weight	Category
1 (Integer)	Bulbasaur (String)	Grass (String)	Poison (String)	Basic (String)	1:7 (String)	.7M (Double)	6.9Kg (Double)	Seed (String)

Evolution: Evolution table of all pokemon

Primary Key: Pokedex#: each pokemon only has one pokedex number and can be uniquely identified by it

Pokemon_Name: the name of the pokemon, can also be uniquely identified by its name but not necessary

Trading: boolean to determine if the pokemon evolves from trading

Stone: String that if not null tells that the pokemon can evolve to the next stage with an evolution stone

Next Stage: The pokemon that the current pokemon will evolve into. A pokemon may have more than one next stage and have another row to indicate it.

Pokemon#	Pokemon_Name	Trading	Stone	Evolve_Level	Next_Stage
1 (Integer)	Bulbasaur (String)	False (String)	Null (String)	16 (Integer)	Ivysaur (String)

Base_Stats: Individual stats of each pokemon

Primary Key: Pokedex#: each pokemon only has one pokedex number and can be uniquely identified by it

Pokemon_Name: the name of the pokemon, can also be uniquely identified by its name but not necessary

HP: Hitpoints of the pokemon if zero, faints

Attack: Physical base damage that the pokemon contributes to the total attack

Sp_Attack: Special base damage that the pokemon contributes to the total attack

Defense: Physical base defense that the pokemon contributes to the total defense

Speed: The pokemon with the higher speed get turn priority

Growth_Rate: how long it takes a pokemon to level up

Pokemon#	Pokemon_Name	HP	Attack	Sp_Atk	Defense	Sp_Def	Speed	Growth_Rate
1 (Integer)	Bulbasaur (String)	45 (Integer)	49 (Integer)	65 (Integer)	49 (Integer)	65 (Integer)	45 (Integer)	Medium_Slow (String)

Location: List of all locations of each pokemon and how they are obtained

Foreign Key: Pokedex#: [Pokemon] each pokemon only has one pokedex number and can be uniquely identified by it

Pokemon_Name: the name of the pokemon, can also be uniquely identified by its name but not necessary

Primary Key: Location: Tells the region that the pokemon is located, like a route or a town

Primary Key: Location_Subname: the specific area in the region that the pokemon may be found,

A pokemon may be found in 2 Location_subnames, but all Location_Subnames + Location in the table are unique meaning there is no duplicate combination of the two.

Habitat: If the pokemon can be found wandering in the wild it wells what kind of terrain the pokemon spawns in, ie grass, sea, cave

Gift: indicates if a pokemon can be received as a gift in that area

Trade: indicates if a pokemon can be traded for in that area

Exchange: indicates if a pokemon can be recived in exchange for an item in that area

Purchase: indicates if a pokemon can be bought and the amount in that area

Fossi: indicates the fossil that the pokemon can be obtained from in that area

Pokemon#	Pokemon_ Name	Location	Location_Subn ame	Habita t	Gift	Trad e	Exchan ge	Purcha se	Fossil
1 (Integer)	Bulbasaur (String)	Pallet_To wn (String)	Prof_Oak_Lab (String)	Null (String)	Tru e (Bo olean)	Null (Boo lean)	Null (Boolea n)	Null (Intege r)	Null (String)
1	Bulbasaur	Cerulean	Melanie's_Hous e	Null	Tru e	Null	Null	Null	Null

Learnset: List of all moves obtainable by each individual pokemon. They can be obtained by level, TM, or HM

So all three attributes can be null. The table contains the entire list of Pokemon moves for each pokemon. That is over 4000 moves. WOW such size much pride.

Foreign Key: Pokedex#: [Pokemon] each pokemon only has one pokedex number and can be uniquely identified by it

Primary Key: Move_Name: Name of the move in the list, may be multiple of the same move name if it can be learned in multiple ways

Level: if the pokemon can learn the move through leveling then the level is indicated

TM: Technical machine: if the pokemon can learn the move through a tm then the tm # is indicated

HM: Hold Machine: if the pokemon can learn the move through a hm then the hm # is indicated

Type: each move has a type that is used to calculate total damage

Power: base power of the move used to calculate total damage

Accuracy: % chance the ability has to hit, if null then it never misses

PP: Power Point: amount of times the move can be used by the pokemon

Learnset

Pokemon#	Level/TM/HM	Move_Name	Type	Power	Accuracy	PP
1	1	Growl	Normal	35	.95	35
1	1	Tackle	Normal	Null	1	40
1	7	Leech Seed	Grass	Null	.90	10
1	13	Vine Whip	Grass	35	1	10
1	20	Poison Powder	Poison	Null	.75	35

1	HM01	Cut	Normal	50	.95	30
1 (Integer)	TM06 (String)	Toxic (String)	Poison (String)	Null (Integer)	.85 (Double)	10 (Integer)

Moves: Entire move list in the game

Primary Key: Move_Name: name of the move, there is only one of each in the table and can be uniquely identified

Power: base power of the move used to calculate total damage

PP: Power Point: amount of times the move can be used by the pokemon

Accuracy: % chance the ability has to hit, if null then it never misses

Status_effect: description if the moves does more than just physical damage

Proc_Chance: chance the status effect will happen during an attack

Category: dictates if the Power value will be applied to special or physical damage

Move_Name	Power	PP	Accuracy	Status_eff	Proc_Chance	Category
Absorb	20	25	100	Absorb Half of Damage	100	Special
Acid	40	30	100	Lowers Opponents Defense	10	Special
Acid Armor	Null	20	Null	Raises Users Defense	100	Null

Agility (String)	Null (Integer)	30 (Integer)	Null (Integer)	Raises Users Speed (String)	100 (Integer)	Null (String)
---------------------	-------------------	-----------------	-------------------	--------------------------------	------------------	------------------

Type_Def: Each types Power of the attack is effected by the typing of the pokemon.

Multiplied by the values that correspond to the type defending in the table.

Primary Key: Type_Name: the name of the type of move attacking, moves only have one type

Fire - Normal: Types of the defending pokemon, which may have up to two types, the values are what the total damage is multiplied by

Type_Name	Fire	Water	Grass	Electric	Ice	Fighting	Poison	Ground	Flying	Psychic	Bug	Rock	Ghost	Dragon	Normal
Fire	.5	.5	2	1	2	1	1	1	1	1	2	.5	1	.5	1
Water	2	.5	.5	1	1	1	1	2	1	1	1	2	1	.5	1
Grass (String)	.5 (double)	2 (double)	.5 (double)	1 (double)	1 (double)	1 (double)	.5 (double)	2 (double)	.5 (double)	1 (double)	.5 (double)	2 (double)	1 (double)	.5 (double)	1 (double)

(4) Application program design (revised from Phase 2)

Application program design:

NOTE: Because of the limited knowledge of the SQL language all sudo code was done in C, however when the project is finished it will all be implemented in SQL.

Function Implementation:

At the bottom of the file, the results have screenshots ordered by figures 1-5

Figure1:

Display the contents of all tables

```
124 #Display contents of all Tables
125 • Select *
126 From BASE_STATS
127 Order By PokemonNum ASC;
128 • Select *
129 From EVOLUTION
130 Order By PokemonNum ASC;
131 • Select *
132 From IMAGES
133 Order By image_id ASC;
134 • Select *
135 From LEARNSET
136 Order By PokemonNum ASC;
137 • Select *
138 From LOCATIONS
139 Order By PokemonNum ASC;
140 • Select *
141 From MOVES
142 Order By Namez ASC;
143 • Select *
144 From POKEMON
145 Order By PokedexNum ASC;
146 • Select *
147 From TRAINER;
148 • Select *
149 From TYPE_DEF;
```

	PokemonNum	Pokemon_Name	HP	Attack	Defense	Sp_Attack	Sp_Def	Speed
▶	1	Bulbasaur	45	49	49	65	65	45
	2	Ivysaur	60	62	63	80	80	60
	3	Venusaur	80	82	83	100	100	80
	4	Charmander	39	52	43	60	50	65
	5	Charmeleon	58	64	58	80	65	80
	6	Charizard	78	84	78	109	85	100
	7	Squirtle	44	48	65	50	64	43
	8	Wartortle	59	63	80	65	80	58
	9	Blastoise	79	83	100	85	105	78
	10	Caterpie	45	30	35	20	20	45
	11	Metapod	50	20	55	25	25	30
	12	Butterfree	60	45	50	90	80	70
	13	Weedle	40	35	30	20	20	50
	14	Kakuna	45	25	50	25	25	35
	15	Beedrill	65	90	40	45	80	75
	16	Pidgey	40	45	40	35	35	56
	17	Pidgeotto	63	60	55	50	50	71
	18	Pidgeot	83	80	75	70	70	101
	19	Rattata	30	55	35	25	35	77

BASE_STATS 47 × EVOLUTION 48 IMAGES 49 LEARNSET 50 LOCATIONS 51 MOVES 52 POKEMON 53 TRAINER 54 TYPE_DEF 55

Figure 2:

144 #How many pokemon of one type are in generation 1

145 • SELECT count(Type1) + count(Type2)

146 FROM POKEMON

147 WHERE POKEMON.Type1 = 'Grass' OR POKEMON.Type2 = 'Grass';

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
count(Type1) + count(Type2)			
28			

Figure 3:

149 #Average stats of chosen types

150 • SELECT avg(HP), avg(Attack), avg(Defense), avg(Sp_Attack), avg(Sp_Def), avg(Speed)

151 FROM BASE_STATS, POKEMON

152 WHERE POKEMON.Type1 = 'Fire' or POKEMON.Type2 = 'Fire';

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	avg(HP)	avg(Attack)	avg(Defense)	avg(Sp_Attack)	avg(Sp_Def)	avg(Speed)
	64.2119	72.5497	68.2252	67.1391	66.0199	68.9338

Figure 4:

```

5      #All Locations of Magikarp
6 •    select Location, Location_Surname, Habitat, Pokemon_Name
7      from POKEMON p, LOCATIONS l
8      where p.PokedexNum = '129' and l.PokemonNum = '129';
9

```

Location	Location_Surname	Habitat	Pokemon_Name
Route 6		Old Rod	Magikarp
Route 10		Old Rod	Magikarp
Route 11		Old Rod	Magikarp
Route 12		Old Rod	Magikarp
Route 13		Old Rod	Magikarp
Route 17		Old Rod	Magikarp
Route 17		Super Rod	Magikarp
Route 18		Old Rod	Magikarp
Route 18		Super Rod	Magikarp
Route 19		Old Rod	Magikarp
Route 20		Old Rod	Magikarp
Route 21		Old Rod	Magikarp
Route 22		Old Rod	Magikarp
Route 23		Old Rod	Magikarp
Route 24		Old Rod	Magikarp

Location	Location_Surname	Habitat	Pokemon_Name
Route 25		Old Rod	Magikarp
Celadon...		Old Rod	Magikarp
Cerulea...	1F	Old Rod	Magikarp
Cerulea...	B1F	Old Rod	Magikarp
Fuchsia ...		Old Rod	Magikarp
Fuchsia ...		Super Rod	Magikarp
Pallet To...		Old Rod	Magikarp
Seafoa...	B3F	Old Rod	Magikarp
Seafoa...	B4F	Old Rod	Magikarp
Vermilio...	City	Old Rod	Magikarp
Vermilio...	Harbor	Old Rod	Magikarp
Viridian ...		Old Rod	Magikarp
Safari Z...	East	Old Rod	Magikarp
Safari Z...	West	Old Rod	Magikarp
Safari Z...	North	Old Rod	Magikarp

Figure 5:

```

9      #All moves that can be learned by Pikachu
0 •    select Move_Name, Pokemon_Name
1      from LEARNSET l, POKEMON p
2      where l.PokemonNum = '25' and p.Pokemon_Name = 'Pikachu';

```

sult Grid		Filter Rows:	Export:	Wrap Cell Content:
Move_Name	Pokemon_Name			
Growl	Pikachu			
ThunderShock	Pikachu			
Quick Attack	Pikachu			
Swift	Pikachu			
Agility	Pikachu			
Thunder	Pikachu			
Thunder Wave	Pikachu			
Flash	Pikachu			
Mega Punch	Pikachu			
Mega Kick	Pikachu			
Toxic	Pikachu			
Body Slam	Pikachu			
Take Down	Pikachu			
Double-Edge	Pikachu			
Pay Day	Pikachu			

Move_Name	Pokemon_Name
Submission	Pikachu
Seismic Toss	Pikachu
Rage	Pikachu
Thunderbolt	Pikachu
Thunder	Pikachu
Mimic	Pikachu
Double Team	Pikachu
Reflect	Pikachu
Bide	Pikachu
Swift	Pikachu
Skull Bash	Pikachu
Rest	Pikachu
Thunder Wave	Pikachu
Substitute	Pikachu

Figure 6:

```

4      #All types that do double damage to Ice Types
5 •    select Type_Name
6      from TYPE_DEF
7      where Ice = '2';

```

sult Grid		Filter Rows:	Edit:
Type_Name			
Fighting			
Fire			
Rock			

User manual:

This is for end-users who may not have database knowledge. Describe precisely how to use your system step by step with screenshots of your system interface and sample outputs.

Download and install MySQL workbench

Use the attached Key-pair: CS2830.PEM to connect with the server Using SSH



- To use the key pair you will need to run commands in bash
- If you are using Windows you will need something like GitBash to run commands
- Find the key pair where it is save and run the command in the pic below that says ssh -i

Use the following information to fill in the forms:

Connect To Your Instance



I would like to connect with

- ☒ A standalone SSH client 
- ☐ A Java SSH Client directly from my browser (Java required) 

To access your instance:

1. Open an SSH client. (find out how to [connect using PuTTY](#))
2. Locate your private key file (CS2830KeyPair.pem). The wizard automatically detects the key you used to launch the instance.
3. Your key must not be publicly viewable for SSH to work. Use this command if needed:

```
chmod 400 CS2830KeyPair.pem
```

4. Connect to your instance using its Public DNS:

```
ec2-18-222-191-67.us-east-2.compute.amazonaws.com
```

Example:

```
ssh -i "CS2830KeyPair.pem" ec2-user@ec2-18-222-191-67.us-east-2.compute.amazonaws.com
```

Please note that in most cases the username above will be correct, however please ensure that you read your AMI usage instructions to ensure that the AMI owner has not changed the default AMI username.

Connection Name:

Connection Remote Management System Profile

Connection Method: Method to use to connect to the RDBMS

Parameters SSL Advanced

SSH Hostname: SSH server hostname, with optional port number.

SSH Username: Name of the SSH user to connect with.

SSH Password: SSH user password to connect to the SSH tunnel.

SSH Key File: Path to SSH private key file.

MySQL Hostname: MySQL server host relative to the SSH server.

MySQL Server Port: TCP/IP port of the MySQL server.

Username: Name of the user to connect with.

Password: The MySQL user's password. Will be requested later if not set.

Default Schema: The schema to use as default schema. Leave blank to select it later.

The pic above should be similar to how it should be set up

Open the file PokemonEntries and the queries are saved inside

There are at least 1 query for each table to show what each table is capable of showing

Movepool: all moves a pokemon can't learn

Type_Def: Show all types that Ice is weak to

Pokedex: Get Caught and seen pokemon

Average stat of all types

Learnset: same as movepool

Location: Show all locations of a pokemon

Pokemon: #of each type of pokemon

Evolution: Show Evolutions of Eevee



Figure 7: #Squirtlesquad