

## CPSC 304 Project Cover Page

Milestone #: 4

Date: November 25, 2022

Group Number: 21

Name	Student Number	CS Alias (Userid)	Email Address
Simon Zhou	29392750	r0b7x	simonzhou3@hotmail.com
Crispin Liu	8656292	n7z5j	crispinliu168@gmail.com
Kevin Qi	24105819	x7g3b	kevinqi413@gmail.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

## Section 1 -- Project Overview

Github Link: [https://github.students.cs.ubc.ca/CPSC304-2022W-T1/project\\_n7z5j\\_r0b7x\\_x7g3b](https://github.students.cs.ubc.ca/CPSC304-2022W-T1/project_n7z5j_r0b7x_x7g3b)

Our group has created a pokemon lookup web application project. Using PostgreSQL, Python, and JavaScript as our technology stack. The data is fetched and compiled using Python from an API called [PokeAPI](#). In the web application, we are able to search for a pokemon using a name. Amidst for a search, the level of when the pokemon can learn a move, stats, sprite, etc is shown. Additionally, users are able to create, update, or delete 'trainers' where trainers have 'trained' pokemons. The user is also able to add/delete a trained\_pokemon from a specific version.

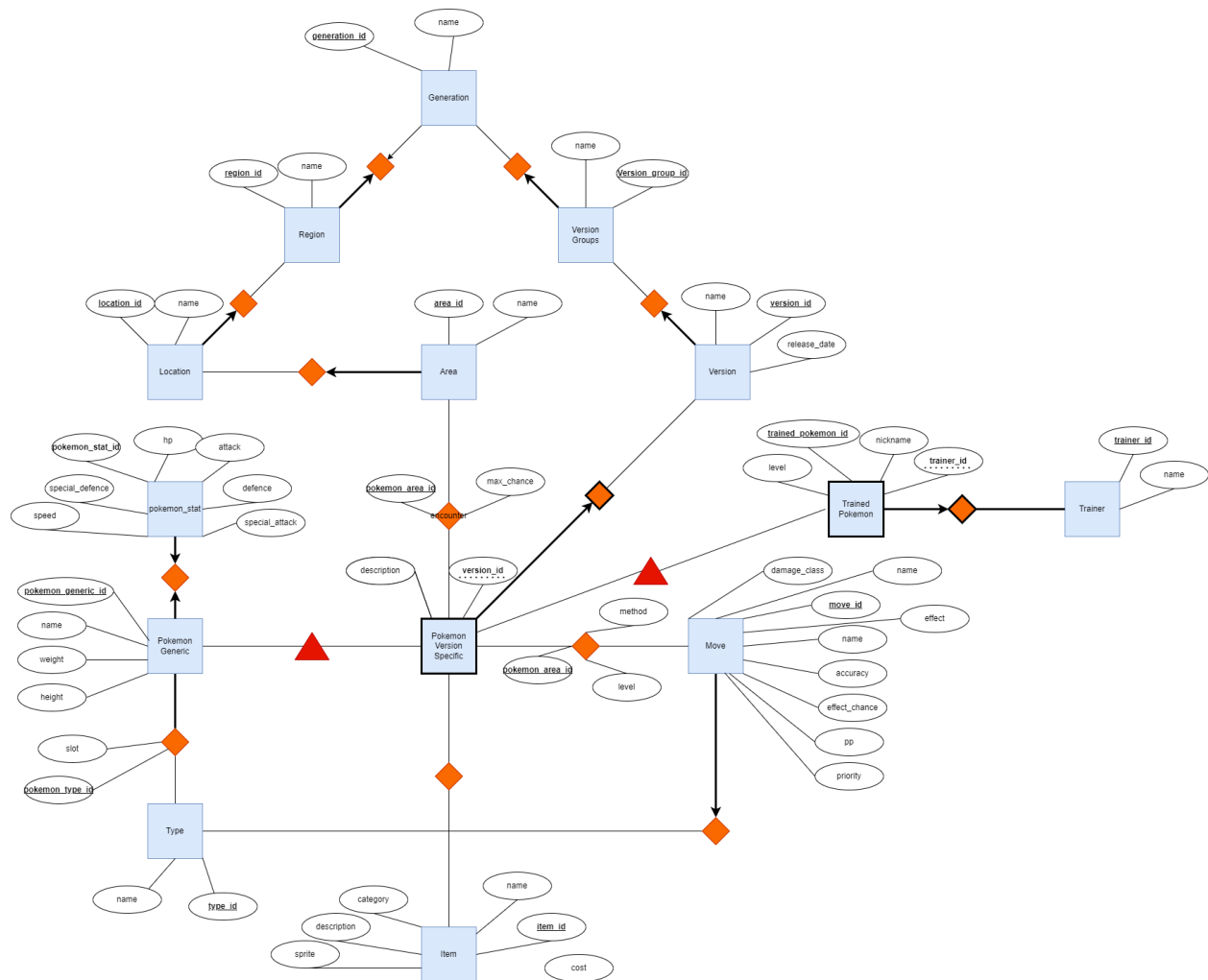
References for CSS styling and HTML code for Frontend:

Table Codes: [Vant 3 - Lightweight Mobile UI Components built on Vue](#)

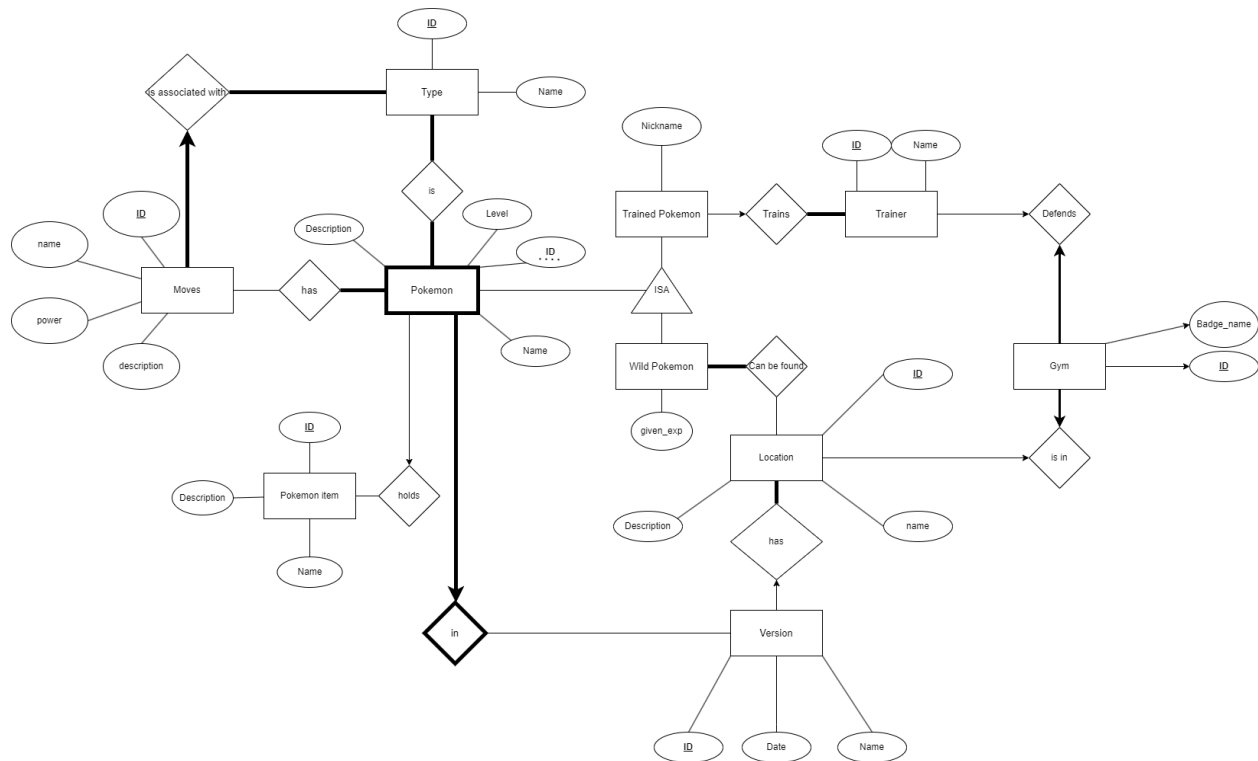
Css styling: [font-palette demo](#)

Bootstrap: [Bootstrap](#)

**NEW ER DIAGRAM:**



## OLD ER DIAGRAM:



## **Differences between Proposed ER Diagram and New ER Diagram:**

As we used an API (<https://pokeapi.co/>) , many entities were changed/added:

- Broke up the pokemon entity into: pokemon\_generic and pokemon\_specific where pokemon\_generic would be a super class to pokemon\_specific and pokemon\_specific would have a weak entity to the version.
- Removed Gym entity (entity table was meaningless)
- Added Generation, Region, Version Groups, and Areas.
  - Region, Version Groups, and Areas depend on Generation.
- Added many new attributes to existing tables.
- Changed dependencies on relationships.

## Section 3 -- Schema and Screenshots

The single SQL Script will be attached with this pdf file: ExportedDatabase.sql.

It contains a creation of Tables and Triggers as well as the concatenation of dynamic data fetched by the api. The SQL Script was created by using: pg\_dump.

(<https://www.postgresql.org/docs/current/app-pgdump.html>)

### List of Schema:

List of relations			
Schema	Name	Type	Owner
public	area	table	psqladmin
public	generation	table	psqladmin
public	item	table	psqladmin
public	location	table	psqladmin
public	move	table	psqladmin
public	pokemon_area	table	psqladmin
public	pokemon_generic	table	psqladmin
public	pokemon_item	table	psqladmin
public	pokemon_move	table	psqladmin
public	pokemon_specific	table	psqladmin
public	pokemon_stat	table	psqladmin
public	pokemon_type	table	psqladmin
public	region	table	psqladmin
public	trained_pokemon	table	psqladmin
public	trainer	table	psqladmin
public	type	table	psqladmin
public	version	table	psqladmin
public	version_group	table	psqladmin
(18 rows)			

**Note that some of these tables have over 500 entities which screenshots may not be able to show all entities from the table. In cases of tables that cannot show all the data, I have created another screenshot that shows the size of the table.**

i) Area Table:

```
cpssc304=> select * from area;
```

area_id	name	location_id
1	celadon-city-area	1
2	celadon-city-celadon-mansion	1
3	cerulean-city-area	2
4	cinnabar-island-area	3
5	cinnabar-island-cinnabar-lab	3
6	digletts-cave-area	4
7	fuchsia-city-area	5
8	mt-moon-mt-moon-square	6
9	mt-moon-1f	6
10	mt-moon-2f	6
11	mt-moon-b1f	6
12	mt-moon-b2f	6
13	pallet-town-area	7
14	rock-tunnel-1f	8
15	rock-tunnel-b1f	8
16	rock-tunnel-b2f	8
17	kanto-route-1-area	9
18	kanto-route-10-area	10
19	kanto-route-11-area	11
20	kanto-route-12-area	12
21	kanto-route-13-area	13
22	kanto-route-14-area	14
23	kanto-route-15-area	15
24	kanto-route-16-area	16
25	kanto-route-17-area	17
26	kanto-route-18-area	18
27	kanto-sea-route-19-area	19
28	kanto-route-2-south-towards-viridian-city	20
29	kanto-route-2-north-towards-pewter-city	20
30	kanto-sea-route-20-area	21
31	kanto-sea-route-21-area	22
32	kanto-route-22-area	23
33	kanto-route-24-area	24

## ii) Generation Table:

```
cpssc304=> select * from generation;
generation_id |      name
-----+-----
            1 | generation i
            2 | generation ii
            3 | generation iii
            4 | generation iv
            5 | generation v
            6 | generation vi
            7 | generation vii
            8 | generation viii
(8 rows)
```

## iii) Item Table:

```
cpssc304=> select * from item;
item_id |      name      | cost | category |      description      |      sprite
-----+-----+-----+-----+-----+-----
      1 | master-ball    |    0 | standard-balls | The best Poké Ball with the ultimate level of performance. With it, you will catch any wild Pokémon without fail. | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/items/master-ball.png
      2 | ultra-ball     |   800 | standard-balls | An ultra-high-performance Poké Ball that provides a higher success rate for catching Pokémon than a Great Ball. | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/items/ultra-ball.png
      3 | great-ball     |   600 | standard-balls | A good, high-performance Poké Ball that provides a higher success rate for catching Pokémon than a standard Poké Ball. | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/items/great-ball.png
      4 | poke-ball      |   200 | standard-balls | A device for catching wild Pokémon. It's thrown like a ball at a Pokémon, comfortably encapsulating its target. | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/items/poke-ball.png
      5 | safari-ball    |    0 | standard-balls | A special Poké Ball that was used in the Safari Zone in the Kanto region and in the Great Marsh in the Sinnoh region. | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/items/safari-ball.png
      6 | net-ball       |  1000 | special-balls  | A somewhat different Poké Ball that is more effective when attempting to catch Water- or Bug-type Pokémon. | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/items/net-ball.png
      7 | dive-ball      |  1000 | special-balls  | A somewhat different Poké Ball that works especially well when catching Pokémon that live underwater. | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/items/dive-ball.png
      8 | nest-ball      |  1000 | special-balls  | A somewhat different Poké Ball that becomes more effective the lower the level of the wild Pokémon. | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/items/nest-ball.png
      9 | repeat-ball    |  1000 | special-balls  | A somewhat different Poké Ball that works especially well on a Pokémon species that has been caught before. | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/items/repeat-ball.png
     10 | timer-ball     |  1000 | special-balls  | A somewhat different Poké Ball that becomes progressively more effective at catching Pokémon the more turns that are taken in battle. | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/items/timer-ball.png
     11 | luxury-ball    |  3000 | special-balls  | A particularly comfortable Poké Ball that makes a wild Pokémon quickly grow friendlier after
```

```
cpssc304=> select count(*) from item;
count
-----
  1607
(1 row)
```

iv) location table:

```
cpssc304=> select * From location;
```

location_id	name	region_id
1	celadon-city	1
2	cerulean-city	1
3	cinnabar-island	1
4	digletts-cave	1
5	fuchsia-city	1
6	mt-moon	1
7	pallet-town	1
8	rock-tunnel	1
9	kanto-route-1	1
10	kanto-route-10	1
11	kanto-route-11	1
12	kanto-route-12	1
13	kanto-route-13	1
14	kanto-route-14	1
15	kanto-route-15	1
16	kanto-route-16	1
17	kanto-route-17	1
18	kanto-route-18	1
19	kanto-sea-route-19	1
20	kanto-route-2	1
21	kanto-sea-route-20	1
22	kanto-sea-route-21	1
23	kanto-route-22	1
24	kanto-route-24	1
25	kanto-route-25	1
26	kanto-route-26	1
27	kanto-route-27	1
28	kanto-route-28	1
29	kanto-route-3	1
30	kanto-route-4	1
31	kanto-route-5	1
32	kanto-route-6	1
33	kanto-route-7	1



```

cpsc304=> select count(*) From location;
count
-----
      706
(1 row)

```

#### v) Move Table:

```

cpsc304=> select * from move limit 1;
move_id | type_id | name  | accuracy | effect_chance | pp | priority | power | damage_class | effect
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----
      1 |      1 | pound |      100 |              | 35 |         0 |    40 | physical    | Inflicts regular damage.
(1 row)

```

#### vi) Pokemon\_Area Table

```

cpsc304=> select * from pokemon_area;
pokemon_area_id | pokemon_specific_id | area_id | max_chance
-----+-----+-----+-----
          1 |          3 |      3 |         100
          2 |          1 |     13 |         100
          3 |          2 |     13 |         100
          4 |         10 |     13 |         100
          5 |         11 |     13 |         100
          6 |         15 |     13 |         100
          7 |         16 |     13 |         100
          8 |         61 |     13 |         100
          9 |         62 |     13 |         100
         10 |         70 |     13 |         100
         11 |         71 |     13 |         100
         12 |         75 |     13 |         100
         13 |         76 |     13 |         100
         14 |         63 |     33 |         100
         15 |        123 |     55 |         100
         16 |        121 |     13 |         100
         17 |        122 |     13 |         100
         18 |        130 |     13 |         100
         19 |        131 |     13 |         100
         20 |        135 |     13 |         100
         21 |        136 |     13 |         100
         22 |        194 |    393 |            4
         23 |        192 |    490 |            8
         24 |        193 |    490 |            8
         25 |        194 |    490 |            8
         26 |        194 |    491 |            8

```

```

cpsc304=> select count(*) from pokemon_area;
count
-----
16665
(1 row)

```

## vii) Pokemon\_Generic Table:

```

cpsc304=> select * from pokemon_generic;
pokemon_generic_id | name | height | weight | sprite
-----
1 | bulbasaur | 7 | 69 | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/1.png
2 | ivysaur | 10 | 130 | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/2.png
3 | venusaur | 20 | 1000 | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/3.png
4 | charmander | 6 | 85 | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/4.png
5 | charmeleon | 11 | 190 | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/5.png
6 | charizard | 17 | 905 | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/6.png
7 | squirtle | 5 | 90 | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/7.png
8 | wartortle | 10 | 225 | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/8.png
9 | blastoise | 16 | 855 | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/9.png
10 | caterpie | 3 | 29 | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/10.png
11 | metapod | 7 | 99 | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/11.png
12 | butterfly | 11 | 320 | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/12.png
13 | weedle | 3 | 32 | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/13.png
14 | kakuna | 6 | 100 | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/14.png
15 | beedrill | 10 | 295 | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/15.png
16 | pidgey | 3 | 18 | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/16.png
17 | pidgeotto | 11 | 300 | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/17.png
18 | pidgeot | 15 | 395 | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/18.png
19 | rattata | 3 | 35 | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/19.png
20 | raticate | 7 | 185 | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/20.png
21 | spearow | 3 | 20 | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/21.png
22 | fearow | 12 | 380 | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/22.png
23 | ekans | 20 | 69 | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/23.png
24 | arbok | 35 | 650 | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/24.png
25 | pikachu | 4 | 60 | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/25.png
26 | raichu | 8 | 300 | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/26.png
27 | sandshrew | 6 | 120 | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/27.png
28 | sandslash | 10 | 295 | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/28.png
29 | nidoran-f | 4 | 70 | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/29.png
30 | nidorina | 8 | 200 | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/30.png
31 | nidoqueen | 13 | 600 | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/31.png
32 | nidoran-m | 5 | 90 | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/32.png
33 | nidorino | 9 | 195 | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/33.png

```

```

cpsc304=> select count(*) from pokemon_generic;
count
-----
905
(1 row)

```

viii) Pokemon\_Item Table:

pokemon_item_id	pokemon_specific_id	item_id	rarity
1	227	199	5
2	228	199	5
3	229	199	5
4	230	199	5
5	231	199	5
6	232	199	5
7	233	199	5
8	234	199	5
9	235	199	5
10	236	199	5
11	237	199	5
12	238	199	5
13	239	199	5
14	240	199	5
15	287	222	5
16	288	222	5
17	289	222	5
18	290	222	5
19	291	222	5
20	292	222	5
21	293	222	5
22	294	222	5
23	295	222	5
24	296	222	5
25	297	222	5
26	298	222	5
27	299	222	5

```
cpssc304=> select count(*) from pokemon_item;
count
-----
3106
(1 row)
```

**IX) pokemon\_move Table:**

```
cpssc304=> select * from pokemon_move LIMIT 10;
```

pokemon_move_id	pokemon_specific_id	move_id	method	level
1	4	9	egg	0
2	5	9	egg	0
3	6	9	egg	0
4	1	10	machine	0
5	2	10	machine	0
6	3	10	machine	0
7	9	10	tutor	0
8	10	10	tutor	0
9	11	10	tutor	0
10	12	10	machine	0

(10 rows)

```
cpssc304=> select count(*) from pokemon_move;
```

count

-----

444224

(1 row)

## X) Pokemon\_Specific Table:

```
cpsc304=> select * from pokemon_specific LIMIT 5;
pokemon_specific_id | pokemon_generic_id | version_id | description
-----+-----+-----+-----
1 | 1 | 1 | A strange seed was +
| | | planted on its +
| | | back at birth.\x0CThe plant sprouts +
| | | and grows with +
| | | this POKÉMON. +
2 | 1 | 2 | A strange seed was +
| | | planted on its +
| | | back at birth.\x0CThe plant sprouts +
| | | and grows with +
| | | this POKÉMON. +
3 | 1 | 3 | It can go for days +
| | | without eating a +
| | | single morsel.\x0CIn the bulb on +
| | | its back, it +
| | | stores energy. +
4 | 1 | 4 | The seed on its +
| | | back is filled +
| | | with nutrients.\x0CThe seed grows +
| | | steadily larger as +
| | | its body grows. +
5 | 1 | 5 | It carries a seed +
| | | on its back right +
| | | from birth. As it\x0Cgrows older, the+
| | | seed also grows +
| | | larger. +

(5 rows)
```

```
cpsc304=> select count(*) from pokemon_specific;
count
-----
8882
(1 row)
```

## XI) Pokemon\_Stat Table:

```
cpsc304=> select * from pokemon_stat LIMIT 20;
pokemon_stat_id | pokemon_generic_id | hp | attack | defence | special_attack | special_defence | speed
-----+-----+-----+-----+-----+-----+-----+-----
1 | 1 | 45 | 49 | 49 | 65 | 65 | 45
2 | 2 | 60 | 62 | 63 | 80 | 80 | 60
3 | 3 | 80 | 82 | 83 | 100 | 100 | 80
4 | 4 | 39 | 52 | 43 | 60 | 50 | 65
5 | 5 | 58 | 64 | 58 | 80 | 65 | 80
6 | 6 | 78 | 84 | 78 | 109 | 85 | 100
7 | 7 | 44 | 48 | 65 | 50 | 64 | 43
8 | 8 | 59 | 63 | 80 | 65 | 80 | 58
9 | 9 | 79 | 83 | 100 | 85 | 105 | 78
10 | 10 | 45 | 30 | 35 | 20 | 20 | 45
11 | 11 | 50 | 20 | 55 | 25 | 25 | 30
12 | 12 | 60 | 45 | 50 | 90 | 80 | 70
13 | 13 | 40 | 35 | 30 | 20 | 20 | 50
14 | 14 | 45 | 25 | 50 | 25 | 25 | 35
15 | 15 | 65 | 90 | 40 | 45 | 80 | 75
16 | 16 | 40 | 45 | 40 | 35 | 35 | 56
17 | 17 | 63 | 60 | 55 | 50 | 50 | 71
18 | 18 | 83 | 80 | 75 | 70 | 70 | 101
19 | 19 | 30 | 56 | 35 | 25 | 35 | 72
20 | 20 | 55 | 81 | 60 | 50 | 70 | 97
(20 rows)
```

```
cpsc304=> select count(*) from pokemon_stat;
count
-----
905
(1 row)
```

## XII) Region Table:

```
cpsc304=> select * from region;
region_id | name | generation_id
-----+-----+-----
1 | kanto | 1
2 | johto | 2
3 | hoenn | 3
4 | sinnoh | 4
5 | unova | 5
6 | kalos | 6
7 | alola | 7
8 | galar | 8
(8 rows)
```

### XIII) Trained\_Pokemon Table:

```
cpssc304=> select * from trained_pokemon;
trained_pokemon_id | pokemon_specific_id | trainer_id |      nickname      | level
-----+-----+-----+-----+-----
                1 |           7130 |         1 |                    |    66
                2 |           481 |         2 |                    |   100
                3 |           101 |         3 |                    |    36
                4 |           169 |         4 |                    |    99
                5 |           376 |         5 | Shiny Metagross |    82
(5 rows)
```

### XIV) Trainer Table:

```
cpssc304=> select * from trainer;
trainer_id | gender | name
-----+-----+-----
          1 | Female | Cynthia
          2 | Male  | Ash
          3 | Male  | Crispin
          4 | Male  | Kevin
          5 | Male  | Simon
(5 rows)
```

## XV) Type Table:

```
cpssc304=> select * from type;
type_id | name
-----+-----
      1 | normal
      2 | fighting
      3 | flying
      4 | poison
      5 | ground
      6 | rock
      7 | bug
      8 | ghost
      9 | steel
     10 | fire
     11 | water
     12 | grass
     13 | electric
     14 | psychic
     15 | ice
     16 | dragon
     17 | dark
     18 | fairy
     19 | unknown
     20 | shadow
(20 rows)
```

## XVI) Version Table:

```
cpssc304=> select * from version limit 10;
version_id | name | release_date | version_group_id
-----+-----+-----+-----
      1 | red | 1996-02-27 | 1
      2 | blue | 1996-10-15 | 1
      3 | yellow | 1998-10-12 | 2
      4 | gold | 1999-11-21 | 3
      5 | silver | 1999-11-21 | 3
      6 | crystal | 2000-12-14 | 4
      7 | ruby | 2002-11-21 | 5
      8 | sapphire | 2002-11-21 | 5
      9 | emerald | 2004-09-16 | 6
     10 | firered | 2004-01-29 | 7
(10 rows)
```



**XVII) Version\_Group Table:**

```
cpsc304=> select * from version_group;
```

version_group_id	name	generation_id
1	red-blue	1
2	yellow	1
3	gold-silver	2
4	crystal	2
5	ruby-sapphire	3
6	emerald	3
7	firered-leafgreen	3
8	colosseum	3
9	xd	3
10	diamond-pearl	4
11	platinum	4
12	heartgold-soulsilver	4
13	black-white	5
14	black-2-white-2	5
15	x-y	6
16	omega-ruby-alpha-sapphire	6
17	sun-moon	7
18	ultra-sun-ultra-moon	7
19	lets-go-pikachu-lets-go-eevee	7
20	sword-shield	8
21	the-isle-of-armor	8
22	the-crown-tundra	8
23	brilliant-diamond-and-shining-pearl	8
24	legends-arceus	8

(24 rows)

## Section 4 -- SQL Queries

This section will be a compilation of all SQL Queries used in the project. We have sorted each SQL Queries by classes/tables. SQL Queries can be found with the variable "SQL =" on the screenshots

**NOTE: (%S) and {<variable\_name>} are ways we inject variables.**

Area SQL Queries:

```
        return arr

    @staticmethod
    async def listPokemonArea(pokemon_specific_id):
        SQL = (f"SELECT area.area_id, area.name, area.location_id, location.name "
              f"FROM pokemon_area AS pa, area, location "
              f"WHERE pa.pokemon_specific_id =(%)s AND pa.area_id = area.area_id AND area.location_id = location.location_id")
        query = await Database.execute(SQL,[pokemon_specific_id])
        return Area.listPokemonAreaFormat(query)

    def __init__(self, area_id):
        self.area_id = area_id
```

Item SQL Queries:

```
    @staticmethod
    async def listItem():
        SQL = f"SELECT item_id,name FROM item"
        query = await Database.execute(SQL,None)
        return query

    @staticmethod
    async def listPokemonItem(pokemon_specific_id):
        SQL = (f"SELECT item.item_id,name,rarity FROM item,pokemon_item as pi "
              f"WHERE pi.pokemon_specific_id = (%s) AND item.item_id = pi.item_id")
        query = await Database.execute(SQL,[pokemon_specific_id])
        print(query)
        return query

    def itemFormat(self,item):
        self.name = item[0]
```

```
        return

    async def load(self):
        SQL = (f"SELECT name,cost,category,description,sprite FROM item WHERE item_id=(%)s LIMIT 1")
        query = await Database.execute(SQL,[self.item_id])
        self.name = query[0][0]
        self.cost = query[0][1]
        self.category = query[0][2]
        self.description = query[0][3]
```

Location SQL Queries:

```

# Add location to database return id and all information on location
@staticmethod
async def addLocation(name,version_id):
    SQL = f"INSERT INTO location (name,region_id) VALUES (%s, %s) RETURNING location_id"
    query = await Database.execute(SQL,(name,version_id))
    return query[0][0]

def __init__(self, location_id):
    self.location_id = location_id
    self.name = None
    self.location = None

async def load(self):
    SQL = f"SELECT * FROM location WHERE location_id=({self.location_id})"
    query = await Database.execute(SQL,None)
    self.name = query[0][1]

```

Move SQL Queries:

```

@staticmethod
async def listPokemonMoves(pokemon_specific_id):
    SQL = (f"SELECT move.move_id, name "
    f"FROM pokemon_move AS pm, move "
    f"WHERE pm.pokemon_specific_id =(%)s) AND pm.move_id = move.move_id")
    query = await Database.execute(SQL,[pokemon_specific_id])
    return query

@staticmethod
async def listMoves():
    SQL = f"SELECT move_id,name FROM move"
    query = await Database.execute(SQL,None)
    return query

```

```

async def load(self):
    SQL = (f"SELECT * "
    f"FROM move,type "
    f"WHERE move.move_id=(%)s) AND type.type_id = move.type_id LIMIT 1")
    move = await Database.execute(SQL,[self.moveId])
    print(move)
    self.name = move[0][2]
    self.accuracy = move[0][3]
    self.effectChance = move[0][4]
    self.pp = move[0][5]
    self.priority = move[0][6]
    self.power = move[0][7]
    self.damageClass = move[0][8]
    self.typeId = move[0][9]

```

## Pokemon SQL Queries:

```
class Pokemon:

    @staticmethod
    async def listPokemon():
        SQL = f"SELECT pokemon_generic_id,name FROM pokemon_generic"
        query = await Database.execute(SQL,None)
        return query

    @staticmethod
    async def getPokemonVersions(generic_id):
        SQL = (f"SELECT version.version_id,name FROM pokemon_specific,version "
              f"WHERE pokemon_generic_id=(%) AND version.version_id = pokemon_specific.version_id")
        query = await Database.execute(SQL,[generic_id])
        print(query)
        return query

    @staticmethod
    async def getPokemonAreaCountPerRegion(generic_id):

        @staticmethod
        async def getPokemonAreaCountPerRegion(generic_id):
            innerSQL = (f"SELECT ps.pokemon_specific_id, ps.version_id, COUNT(*) AS areaCount "
                       f"FROM pokemon_specific as ps "
                       f"INNER JOIN pokemon_area as pa ON ps.pokemon_specific_id = pa.pokemon_specific_id "
                       f"INNER JOIN pokemon_generic as pg ON pg.pokemon_generic_id = ps.pokemon_generic_id "
                       f"WHERE pg.pokemon_generic_id=(%) GROUP BY ps.pokemon_specific_id, ps.version_id")

            SQL = (f"SELECT r.region_id, r.name, SUM(ps.areaCount) "
                  f"FROM ({innerSQL}) AS ps, version AS v, version_group AS vg, region AS r "
                  f"WHERE vg.version_group_id = v.version_group_id AND ps.version_id = v.version_id AND r.generation_id = vg.generation_id "
                  f"GROUP BY r.region_id")
            query = await Database.execute(SQL,[generic_id])
            print(query)
            return query

        def initPokemon(self):
            self.versionIdList = []

MS    OUTPUT    DEBUG CONSOLE    TERMINAL    GIT LENS    JUPYTER

    async def load(self):
        # Get general data
        SQL = (f"SELECT name,height,weight,sprite,pokemon_specific_id,description "
              f"FROM pokemon_generic AS pg INNER JOIN pokemon_specific AS ps ON ps.pokemon_generic_id = pg.pokemon_generic_id "
              f"WHERE pg.pokemon_generic_id =(%) AND ps.version_id =(%)")
        query = await Database.execute(SQL,[self.pokemon_generic_id,self.version_id])
        self.name = query[0][0]
        self.height = query[0][1]
        self.weight = query[0][2]
        self.sprite = query[0][3]
        self.pokemon_specific_id = query[0][4]

    async def findPokemonThatAllTrainer(gender):
        if gender is not None:
            SQL = f"SELECT DISTINCT pg.pokemon_generic_id, pg.name FROM pokemon_specific ps \"
                  f"INNER JOIN pokemon_generic pg ON pg.pokemon_generic_id = ps.pokemon_generic_id \"
                  f"WHERE NOT EXISTS((SELECT t.trainer_id FROM trainer t WHERE t.gender = {gender}) EXCEPT \"
                  f"(SELECT t.trainer_id FROM trained_pokemon tp INNER JOIN trainer t ON t.trainer_id = tp.trainer_id WHERE ps.pokemon_specific_id = tp.pokemon_specific_id)) \"
        else:
            SQL = f"SELECT DISTINCT pg.pokemon_generic_id, pg.name FROM pokemon_specific ps \"
                  f"INNER JOIN pokemon_generic pg ON pg.pokemon_generic_id = ps.pokemon_generic_id \"
                  f"WHERE NOT EXISTS((SELECT t.trainer_id FROM trainer t) EXCEPT \"
                  f"(SELECT t.trainer_id FROM trained_pokemon tp INNER JOIN trainer t ON t.trainer_id = tp.trainer_id WHERE ps.pokemon_specific_id = tp.pokemon_specific_id)) \"
        query = await Database.execute(SQL, [])
        return query
```

## Region SQL Queries:

```
pokemon_generic_id = None

@staticmethod
async def listRegions():
    SQL = (f"SELECT r.region_id, r.name, COUNT(DISTINCT ps.pokemon_generic_id) "
          f"FROM pokemon_specific AS ps, version AS v, version_group AS vg, region AS r "
          f"WHERE vg.version_group_id = v.version_group_id AND v.version_id = ps.version_id AND r.generation_id = vg.generation_id "
          f"GROUP BY r.region_id")
    query = await Database.execute(SQL, None)
    return query

def __init__(self, region_id):
    self.region_id = region_id
```

```
# Load all region statistic and everything
async def load(self):
    SQL = (f"SELECT r.region_id, r.name, COUNT(DISTINCT ps.pokemon_generic_id) "
          f"FROM pokemon_specific AS ps, version AS v, version_group AS vg, region AS r "
          f"WHERE vg.version_group_id = v.version_group_id AND v.version_id = ps.version_id AND r.generation_id = vg.generation_id AND region="
          f"GROUP BY r.region_id")
    # query = await Database.execute(SQL, None)
    # self.name = query[0]
    return None
```

## Stat SQL Queries:

```
async def load(self):
    SQL = (f"SELECT *"
          f"FROM pokemon_stat "
          f"WHERE pokemon_stat.pokemon_generic_id= (%s) LIMIT 1")
    stats = await Database.execute(SQL, [self.pokemon_generic_id])
    self.formatStats(stats[0])
    return

def getStats(self):
    return {
        "hp": self.hp,
```

## Trained Pokemon Queries:

```
@staticmethod
async def listTrainedPokemon(trainer_id):
    SQL = f"SELECT tp.trainer_id, tp.trained_pokemon_id, ps.pokemon_specific_id,pg.name, tp.nickname, tp.level, pg.sprite, v.name " \
        f"FROM trained_pokemon tp INNER JOIN pokemon_specific ps ON ps.pokemon_specific_id = tp.pokemon_specific_id " \
        f"INNER JOIN pokemon_generic pg ON pg.pokemon_generic_id = ps.pokemon_generic_id " \
        f"INNER JOIN version v ON v.version_id = ps.version_id "\
        f"WHERE tp.trainer_id = (%s) "
    query = await Database.execute(SQL,[trainer_id])
    return TrainedPokemon.listTrainedPokemonFormat(query)

@staticmethod
async def create(trainer_id,data):
    pokemon_specific_id,nickname,level = itemgetter('pokemon_specific_id','nickname','level')(data)
    SQL = (f"INSERT INTO trained_pokemon (pokemon_specific_id,trainer_id,nickname,level) "
        f"VALUES (%s,%s,%s,%s) RETURNING trained_pokemon_id")
    query = await Database.execute(SQL,[pokemon_specific_id,trainer_id,nickname,level])
    return query[0][0]

def __init__(self, trained_pokemon_id):
    self.trained_pokemon_id = trained_pokemon_id
```

```
async def load(self):
    SQL = f"SELECT pokemon_specific_id,trainer_id,nickname,level FROM trained_pokemon WHERE trained_pokemon_id=(%s)"
    query = await Database.execute(SQL,[self.trained_pokemon_id])
    self.pokemon_specific_id = query[0][0]
    self.trainer_id = query[0][1]
    self.nickname = query[0][2]
    self.level = query[0][3]
    print("Finish loading trained pokemon data")
    return

async def update(self):
    SQL = f"UPDATE trained_pokemon SET nickname=(%s), level=(%s) WHERE trained_pokemon_id=(%s) RETURNING true"
    await Database.execute(SQL,[self.nickname,self.level,self.trained_pokemon_id])

async def delete(self):
    SQL = f"DELETE FROM trained_pokemon WHERE trained_pokemon_id=(%s) RETURNING true"
    await Database.execute(SQL,[self.trained_pokemon_id])

async def getLeaderboard(range, operator):
    SQL = f"SELECT tr.trainer_id, tr.name, COUNT(*) FROM trained_pokemon tp INNER JOIN trainer AS tr ON tp.trainer_id = tr.trainer_id "\
        f"INNER JOIN pokemon_specific AS ps ON ps.pokemon_specific_id = tp.pokemon_specific_id "\
        f"INNER JOIN pokemon_generic pg ON pg.pokemon_generic_id = ps.pokemon_generic_id "\
        f"INNER JOIN pokemon_stat pst ON pst.pokemon_generic_id = pg.pokemon_generic_id "\
        f"GROUP BY tr.trainer_id, tr.name HAVING COUNT(*) {operator} {range} ORDER BY COUNT(*) DESC, tr.name"

    query = await Database.execute(SQL, [])
    return query
```

## Trainer SQL Queries:

```
@staticmethod
async def listTrainers():
    SQL = f"SELECT trainer_id,name,gender FROM trainer"
    query = await Database.execute(SQL,None)
    return query

# Add location to database return id and all information on location
@staticmethod
async def create(data):
    name,gender = itemgetter('name','gender')(data)
    SQL = f"INSERT INTO trainer (name,gender) VALUES (%s,%s) RETURNING trainer_id"
    query = await Database.execute(SQL,[name,gender])
    return query[0][0]

def __init__(self, trainer_id):
```

```
    async def load(self):
        SQL = f"SELECT name,gender FROM trainer WHERE trainer_id=(%s)"
        query = await Database.execute(SQL,[self.trainer_id])
        self.name = query[0][0]
        self.gender = query[0][1]
        print("Finish loading trainer data")
        self.pokemon = await TrainedPokemon.listTrainedPokemon(self.trainer_id)
        print("Finish loading trained pokemons")

    async def update(self,data):
        self.name = data.get('name') or self.name
        self.gender = data.get('gender') or self.gender
        SQL = f"UPDATE trainer SET name=(%s), gender=(%s) WHERE trainer_id=(%s) RETURNING true"
        await Database.execute(SQL,[self.name,self.gender,self.trainer_id])
```

```
    async def delete(self):
        SQL = f"DELETE FROM trainer WHERE trainer_id=(%s) RETURNING true"
        await Database.execute(SQL,[self.trainer_id])

    async def getPokemonOwnedCount(self):
        SQL = (f"SELECT tp.pokemon_specific_id, pg.name, COUNT(tp.pokemon_specific_id) FROM trained_pokemon tp "
        f"INNER JOIN trainer AS tr ON tp.trainer_id = tr.trainer_id "
        f"INNER JOIN pokemon_specific AS ps ON tp.pokemon_specific_id = ps.pokemon_specific_id "
        f"INNER JOIN pokemon_generic AS pg ON pg.pokemon_generic_id = ps.pokemon_generic_id "
        f"WHERE tr.trainer_id = (%s) GROUP BY tp.pokemon_specific_id, pg.name")
        query = await Database.execute(SQL, [self.trainer_id])
        return query
```

Type SQL Queries:

```
        self.types.append(vals)
    return

    async def load(self):
        SQL = (f"SELECT name,slot "
              f"FROM type, pokemon_type "
              f"WHERE pokemon_type.pokemon_generic_id = (%s) AND pokemon_type.type_id = type.type_id")
        types = await Database.execute(SQL,[self.pokemon_generic_id])
        self.formatTypes(types)
        return

    def getTypes(self):
        return self.types
```

OUTPUT DEBUG CONSOLE TERMINAL GITLens Jupyter

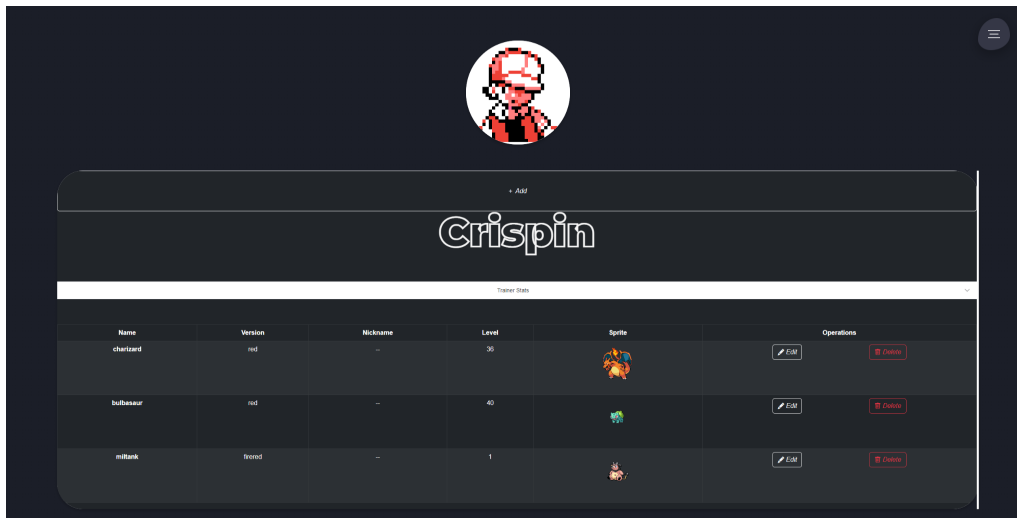


## Section 5 -- Screenshots of Required Queries

### INSERT

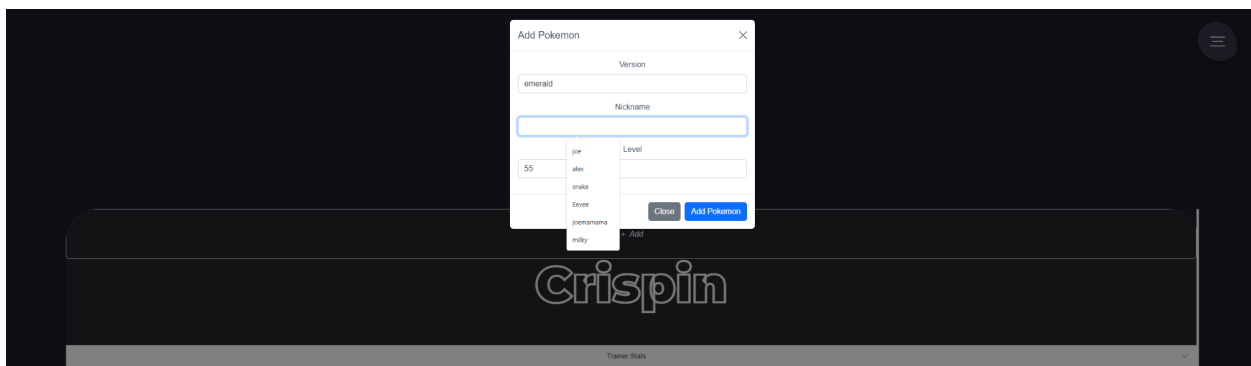
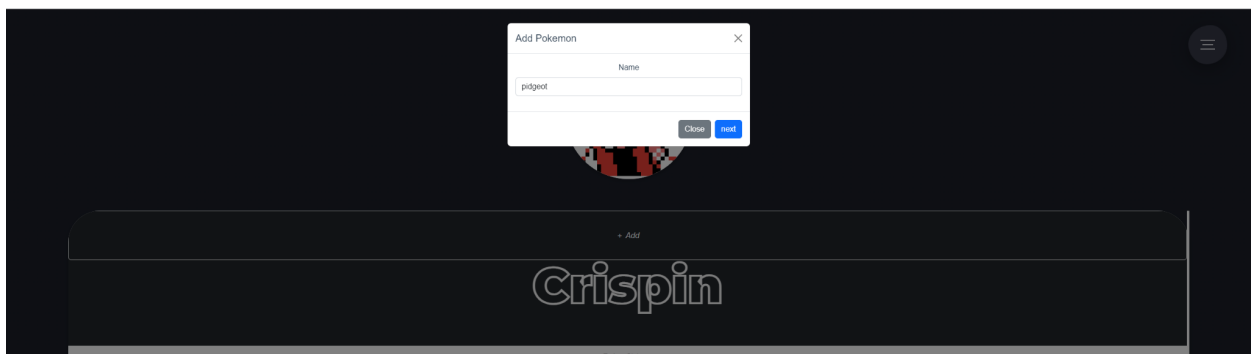
```
@staticmethod
async def create(trainer_id,data):
    pokemon_specific_id,nickname,level = itemgetter('pokemon_specific_id','nickname','level')(data)
    SQL = (f"INSERT INTO trained_pokemon (pokemon_specific_id,trainer_id,nickname,level) "
    f"VALUES (%s,%s,%s,%s) RETURNING trained_pokemon_id")
    query = await Database.execute(SQL,[pokemon_specific_id,trainer_id,nickname,level])
    return query[0][0]
```





### Before INSERT:



```
cpssc304=> select * From trained_pokemon WHERE trainer_id=3;
  trained_pokemon_id | pokemon_specific_id | trainer_id | nickname | level
-----+-----+-----+-----+-----
                3 |           101 |         3 |          |    36
               18 |              1 |         3 |          |    40
               25 |          4540 |         3 |          |     1
(3 rows)
```

During/After (clicking Add Pokemon button):

A screenshot of the Crispin application showing a list of Pokemon. At the top, there is a circular profile picture of a character and the name 'Crispin'. Below this is a table with the following columns: Name, Version, Nickname, Level, Sprite, and Operations. The table contains four rows of data.

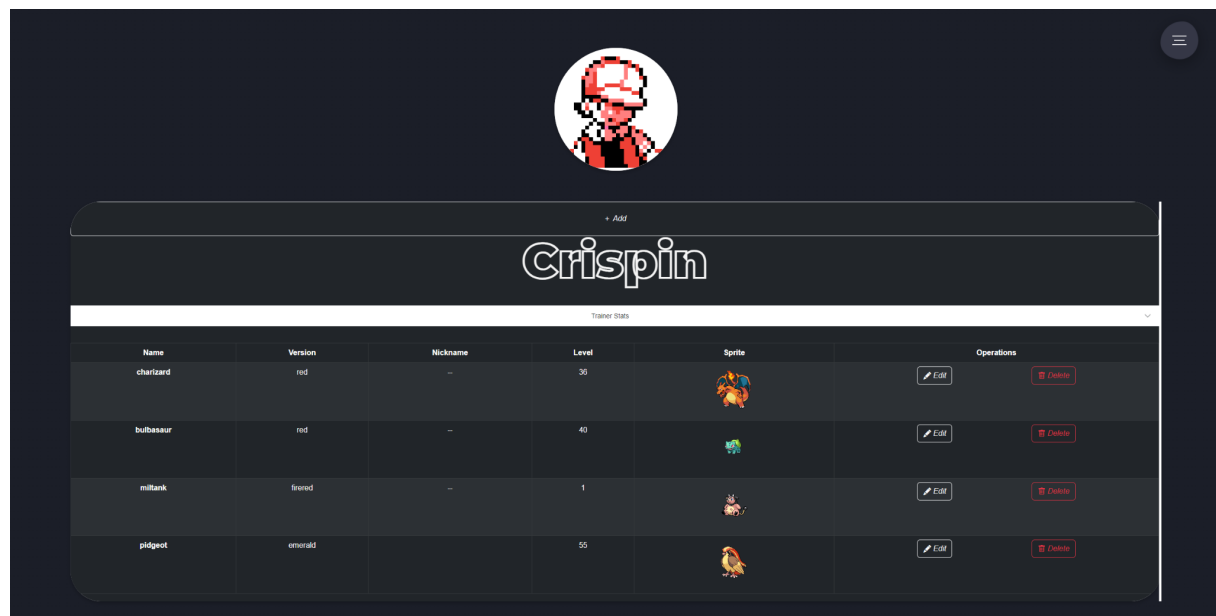
Name	Version	Nickname	Level	Sprite	Operations
charizard	red	--	30		<a href="#">Edit</a> <a href="#">Delete</a>
bulbasaur	red	--	40		<a href="#">Edit</a> <a href="#">Delete</a>
miltank	fire	--	1		<a href="#">Edit</a> <a href="#">Delete</a>
pidgeot	emerald	--	55		<a href="#">Edit</a> <a href="#">Delete</a>

```
cpssc304=> select * From trained_pokemon WHERE trainer_id=3;
  trained_pokemon_id | pokemon_specific_id | trainer_id | nickname | level
-----+-----+-----+-----+-----
                3 |             101 |         3 |          |    36
               18 |              1 |         3 |          |    40
               25 |          4540 |         3 |          |     1
               116 |           349 |         3 |          |    55
(4 rows)
```





## DELETE

```
async def delete(self):
    SQL = f"DELETE FROM trained_pokemon WHERE trained_pokemon_id=(%s) RETURNING true"
    await Database.execute(SQL, [self.trained_pokemon_id])
```

## Before DELETE:



The screenshot shows a web application interface for a Pokémon trainer. At the top, there is a circular profile picture of a character wearing a red cap and a red and white shirt. Below the profile picture, the name "Crispin" is displayed in a large, stylized font. Underneath the name, there is a section titled "Trainer Stats" with a dropdown arrow. Below this, there is a table listing the trainer's Pokémon team. The table has columns for Name, Version, Nickname, Level, Sprite, and Operations. The Operations column contains "Edit" and "Delete" buttons for each Pokémon.

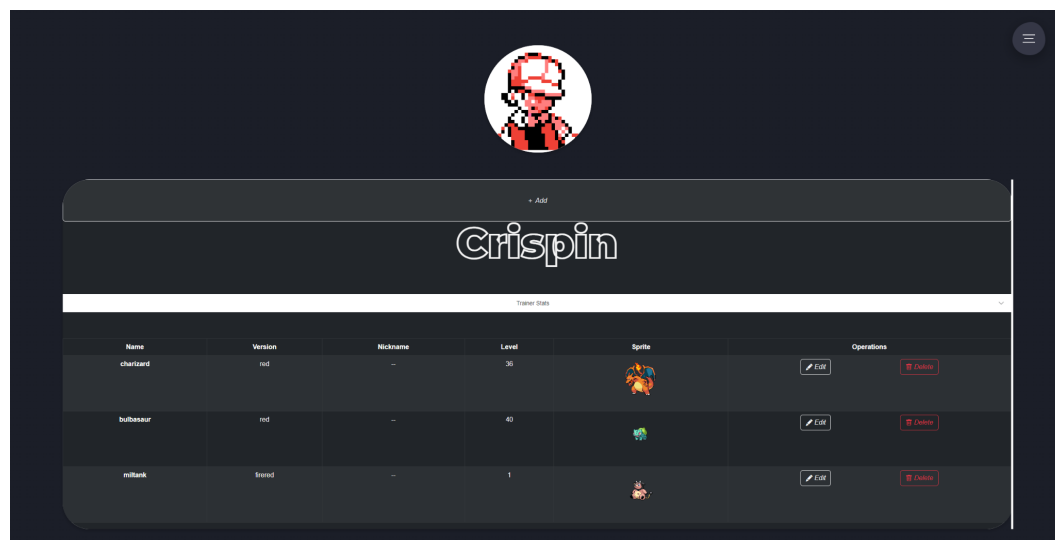
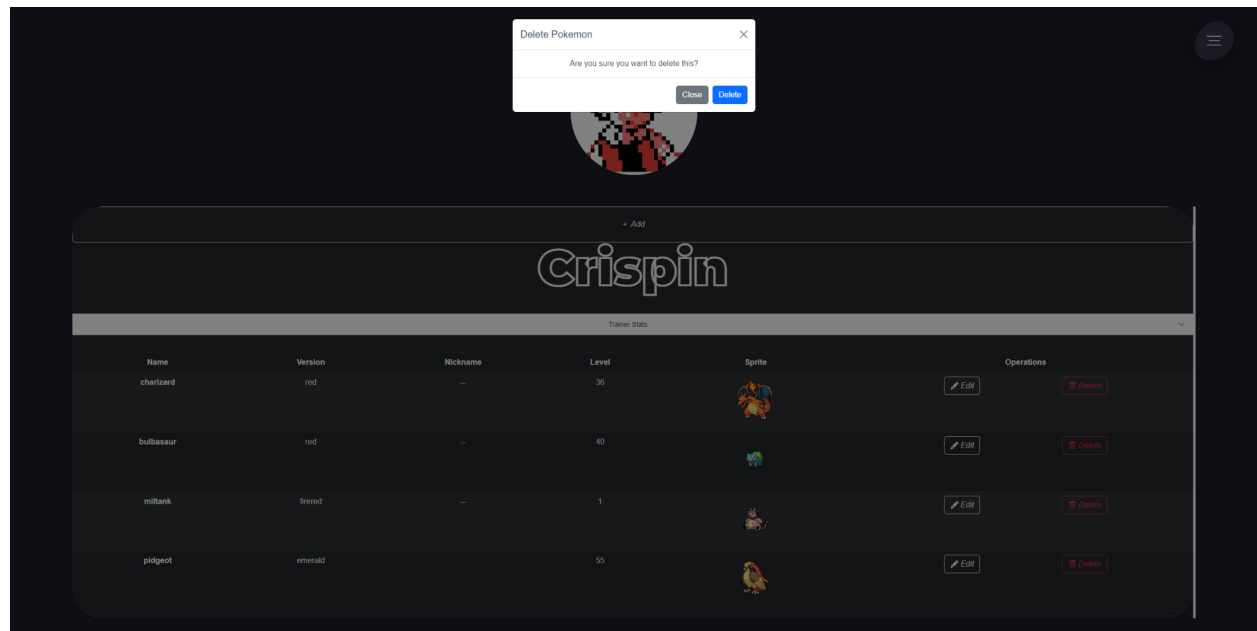
Name	Version	Nickname	Level	Sprite	Operations
charizard	red	--	36		<a href="#">Edit</a> <a href="#">Delete</a>
bulbasaur	red	--	40		<a href="#">Edit</a> <a href="#">Delete</a>
miltank	fire	--	1		<a href="#">Edit</a> <a href="#">Delete</a>
pidgeot	emerald	--	55		<a href="#">Edit</a> <a href="#">Delete</a>

```

cpssc304=> select * From trained_pokemon WHERE trainer_id=3;
  trained_pokemon_id | pokemon_specific_id | trainer_id | nickname | level
-----+-----+-----+-----+-----
                3 |                101 |          3 |          |    36
               18 |                  1 |          3 |          |    40
               25 |             4540 |          3 |          |     1
               116 |             349 |          3 |          |    55
(4 rows)

```

**After DELETE (clicking Delete button):**

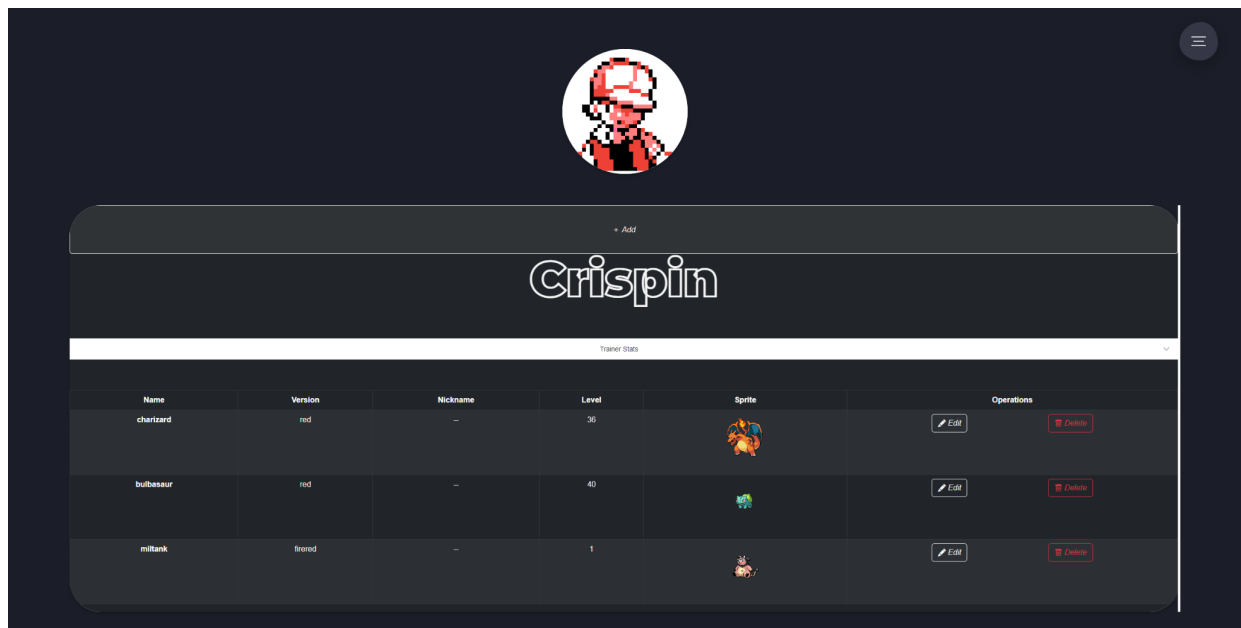


```
cpsc304=> select * From trained_pokemon WHERE trainer_id=3;
  trained_pokemon_id | pokemon_specific_id | trainer_id | nickname | level
-----+-----+-----+-----+-----
                3 |           101 |         3 |          |    36
               18 |              1 |         3 |          |    40
               25 |          4540 |         3 |          |     1
(3 rows)
```

## UPDATE Query:

```
async def update(self):
    SQL = f"UPDATE trained_pokemon SET nickname=(%s), level=(%s) WHERE trained_pokemon_id=(%s) RETURNING true"
    await Database.execute(SQL, [self.nickname, self.level, self.trained_pokemon_id])
```

## Before UPDATE:

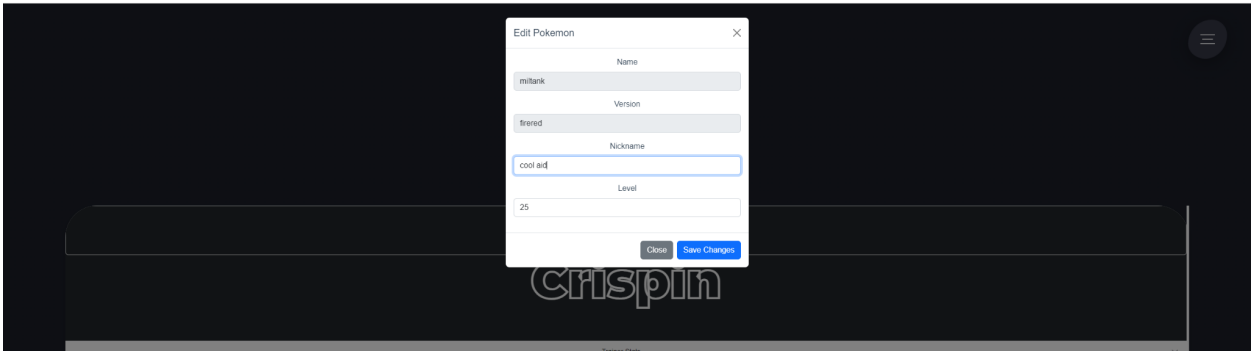


```

cpsc304=> select * From trained_pokemon WHERE trainer_id=3;
  trained_pokemon_id | pokemon_specific_id | trainer_id | nickname | level
-----+-----+-----+-----+-----
                3 |                101 |          3 |          |    36
               18 |                  1 |          3 |          |    40
               25 |               4540 |          3 |          |     1
(3 rows)

```

After UPDATE:



+ Add					
Crispin					
Trainer Stats					
Name	Version	Nickname	Level	Sprite	Operations
charizard	red	--	36		<a href="#">Edit</a> <a href="#">Delete</a>
bulbasaur	red	--	40		<a href="#">Edit</a> <a href="#">Delete</a>
miltank	fired	cool aid	25		<a href="#">Edit</a> <a href="#">Delete</a>

```

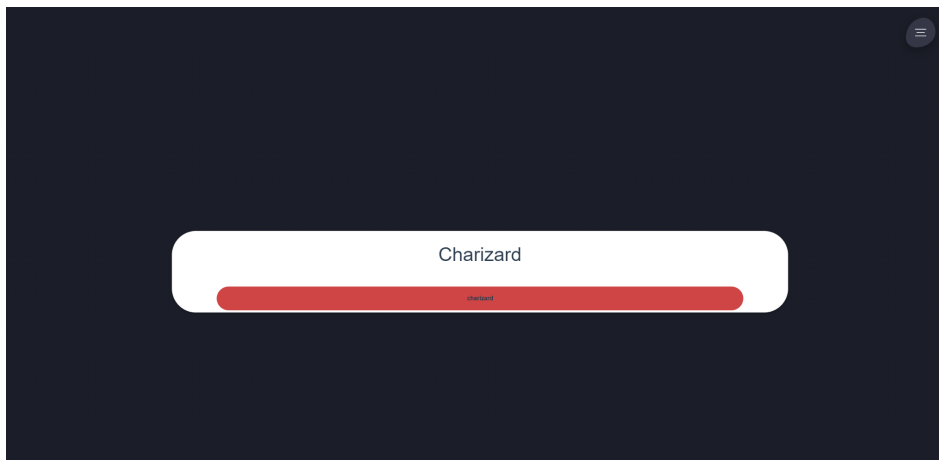
cpsc304=> select * From trained_pokemon WHERE trainer_id=3;
  trained_pokemon_id | pokemon_specific_id | trainer_id | nickname | level
-----+-----+-----+-----+-----
                3 |                101 |          3 |          |    36
               18 |                  1 |          3 |          |    40
               25 |               4540 |          3 | cool aid |    25
(3 rows)

```

## SELECTION

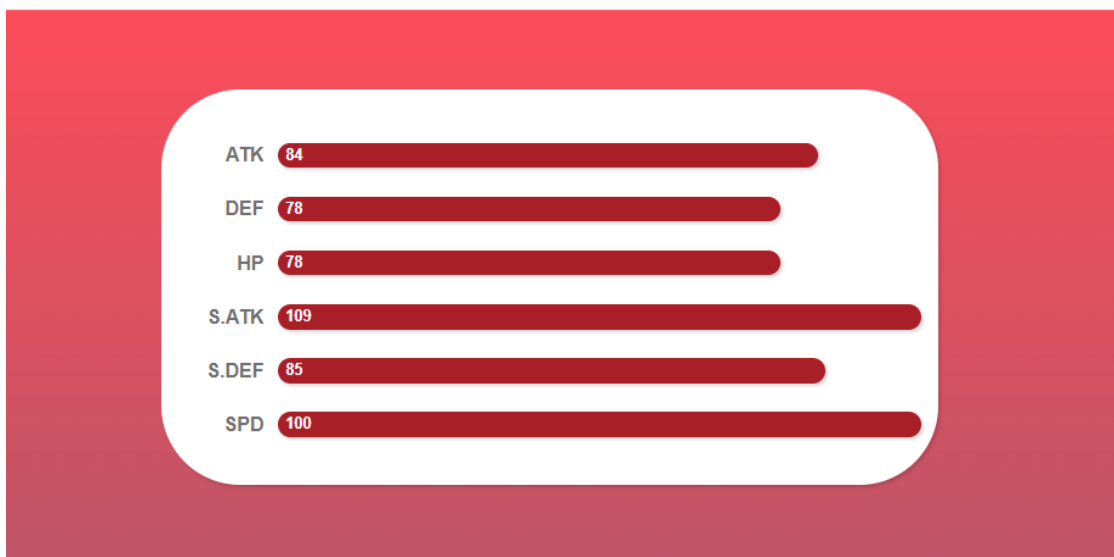
```
async def load(self):
    SQL = (f"SELECT *"
          f"FROM pokemon_stat "
          f"WHERE pokemon_stat.pokemon_generic_id= (%s) LIMIT 1")
    stats = await Database.execute(SQL,[self.pokemon_generic_id])
    self.formatStats(stats[0])
```

### Before Selection Query :



### After Selection Query:

- Note: The search bar queries multiple data of pokemon\_generic

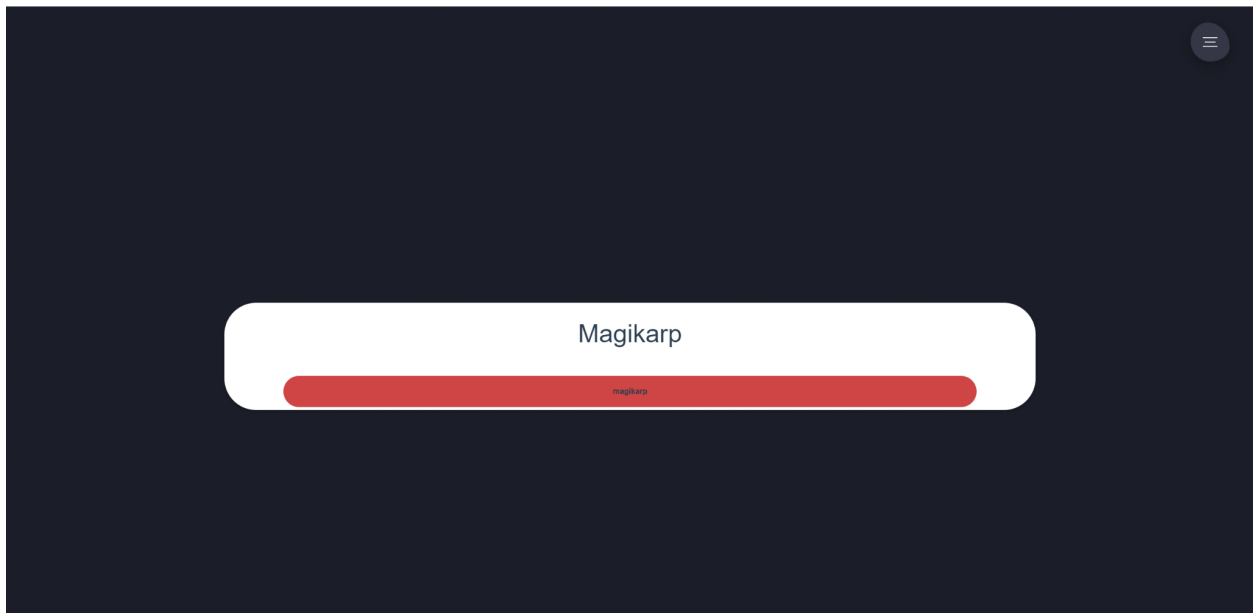


## PROJECTION

```
@staticmethod
async def listPokemonArea(pokemon_specific_id):
    SQL = (f"SELECT area.area_id, area.name, area.location_id, location.name "
          f"FROM pokemon_area AS pa, area, location "
          f"WHERE pa.pokemon_specific_id =(%)s) AND pa.area_id = area.area_id AND area.location_id = location.location_id")
    query = await Database.execute(SQL,[pokemon_specific_id])
    return Area.listPokemonAreaFormat(query)

def __init__(self, area_id):
    self.area_id = area_id
```

## Before Projection Query:



## After Projection Query:



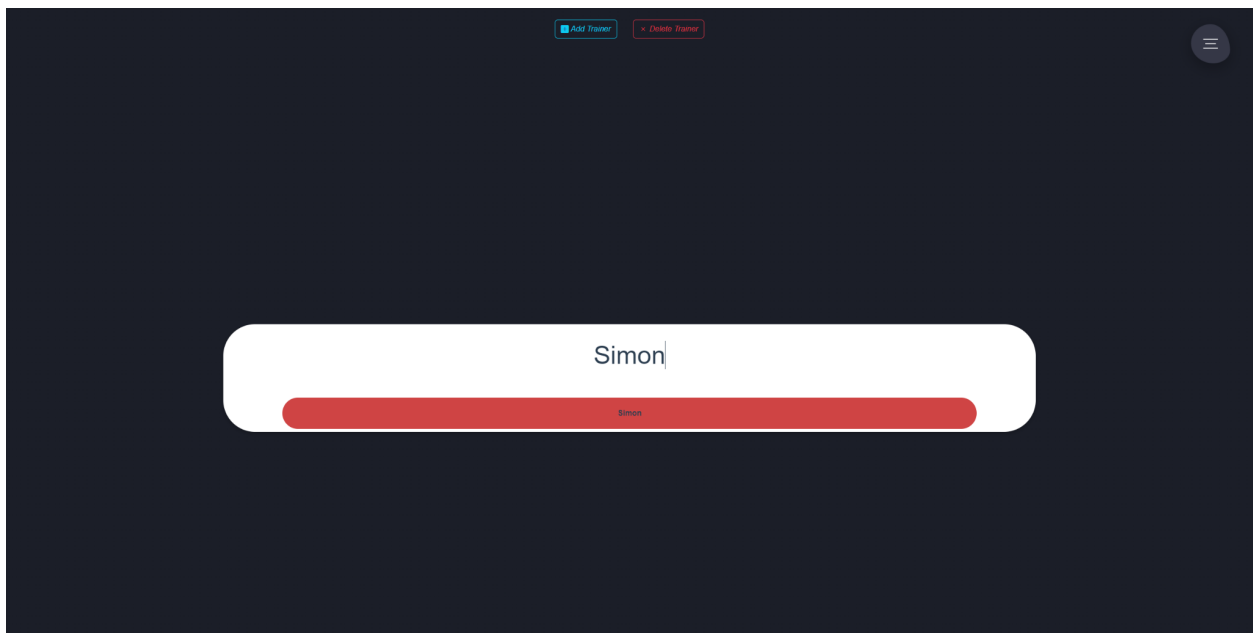


## JOIN


```
@staticmethod
async def listTrainedPokemon(trainer_id):
    SQL = f"SELECT tp.trainer_id, tp.trained_pokemon_id, ps.pokemon_specific_id,pg.name, tp.nickname, tp.level, pg.sprite, v.name " \
        f"FROM trained_pokemon tp INNER JOIN pokemon_specific ps ON ps.pokemon_specific_id = tp.pokemon_specific_id " \
        f"INNER JOIN pokemon_generic pg ON pg.pokemon_generic_id = ps.pokemon_generic_id " \
        f"INNER JOIN version v ON v.version_id = ps.version_id "\
        f"WHERE tp.trainer_id = (%s) "
    query = await Database.execute(SQL,[trainer_id])
    return TrainedPokemon.listTrainedPokemonFormat(query)

@staticmethod
async def create(trainer_id,data):
    pokemon_specific_id,nickname,level = itemgetter('pokemon_specific_id','nickname','level')(data)
    SQL = (f"INSERT INTO trained_pokemon (pokemon_specific_id,trainer_id,nickname,level) "
```

## Before Join Query:









After Join Query:



+ Add

Simon

Trainer Stats

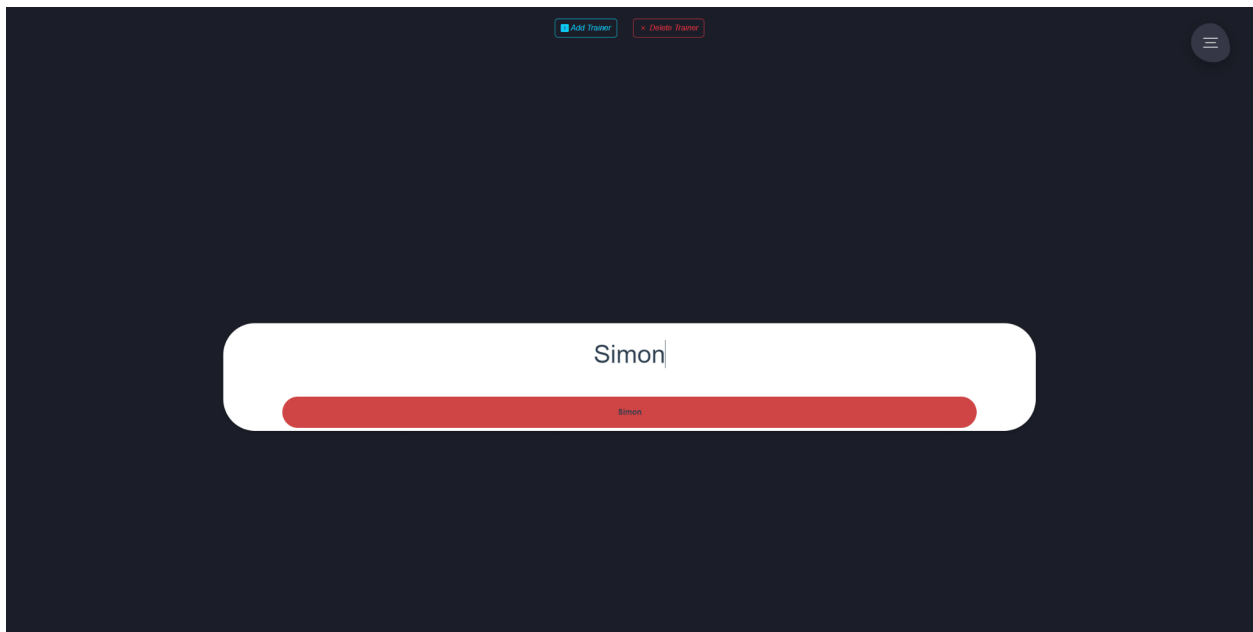
Name	Version	Nickname	Level	Sprite	Operations
reshiram	black 2	—	58		<a href="#">Edit</a> <a href="#">Delete</a>
greninja	y	—	44		<a href="#">Edit</a> <a href="#">Delete</a>
bulbasaur	red	—	60		<a href="#">Edit</a> <a href="#">Delete</a>
miltank	firered	—	1		<a href="#">Edit</a> <a href="#">Delete</a>
diglett	white 2	—	20		<a href="#">Edit</a> <a href="#">Delete</a>
rattata	soulsilver	Shiny Metagross	81		<a href="#">Edit</a> <a href="#">Delete</a>

```
cpsc304> SELECT tp.trainer_id, tp.trained_pokemon_id, ps.pokemon_specific_id,pg.name, tp.nickname, tp.level, pg.sprite, v.name
FROM trained_pokemon tp INNER JOIN pokemon_specific ps ON ps.pokemon_specific_id = tp.pokemon_specific_id
INNER JOIN pokemon_generic pg ON pg.pokemon_generic_id = ps.pokemon_generic_id
INNER JOIN version v ON v.version_id = ps.version_id
WHERE tp.trainer_id = 5;
trainer_id | trained_pokemon_id | pokemon_specific_id | name | nickname | level | sprite | name
-----
5 | 12 | 8169 | reshiram | | 58 | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/643.png | black 2
5 | 13 | 8212 | greninja | | 44 | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/658.png | y
5 | 20 | 1 | bulbasaur | | 60 | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/1.png | red
5 | 27 | 4540 | miltank | | 1 | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/241.png | firered
5 | 89 | 1000 | diglett | | 20 | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/50.png | white 2
5 | 5 | 376 | rattata | Shiny Metagross | 81 | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/19.png | soulsilver
5 | 90 | 2652 | eevee | Garcho0 | 77 | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/133.png | diamond
5 | 88 | 2652 | eevee | joe4 | 554 | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/133.png | diamond
5 | 91 | 5000 | lombre | joe | 50 | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/271.png | white 2
5 | 93 | 8860 | regieleki | aack | 52 | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/894.png | shield
5 | 94 | 5484 | aggron | | 44 | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/306.png | platinum
5 | 95 | 3003 | mew | | 87 | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/151.png | yellow
5 | 97 | 4812 | mudkip | | 45 | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/258.png | platinum
5 | 98 | 1612 | magnemite | | 45 | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/81.png | diamond
5 | 99 | 86 | chanmeleon | | 12 | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/5.png | crystal
5 | 105 | 4352 | kingdra | joenamama | 77 | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/230.png | emerald
5 | 101 | 4580 | raikou | | 60 | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/243.png | soulsilver
5 | 106 | 1454 | tentacruel | joe | 1212 | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/73.png | platinum
5 | 107 | 6762 | luxio | | 90 | https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/404.png | pearl
(19 rows)
```

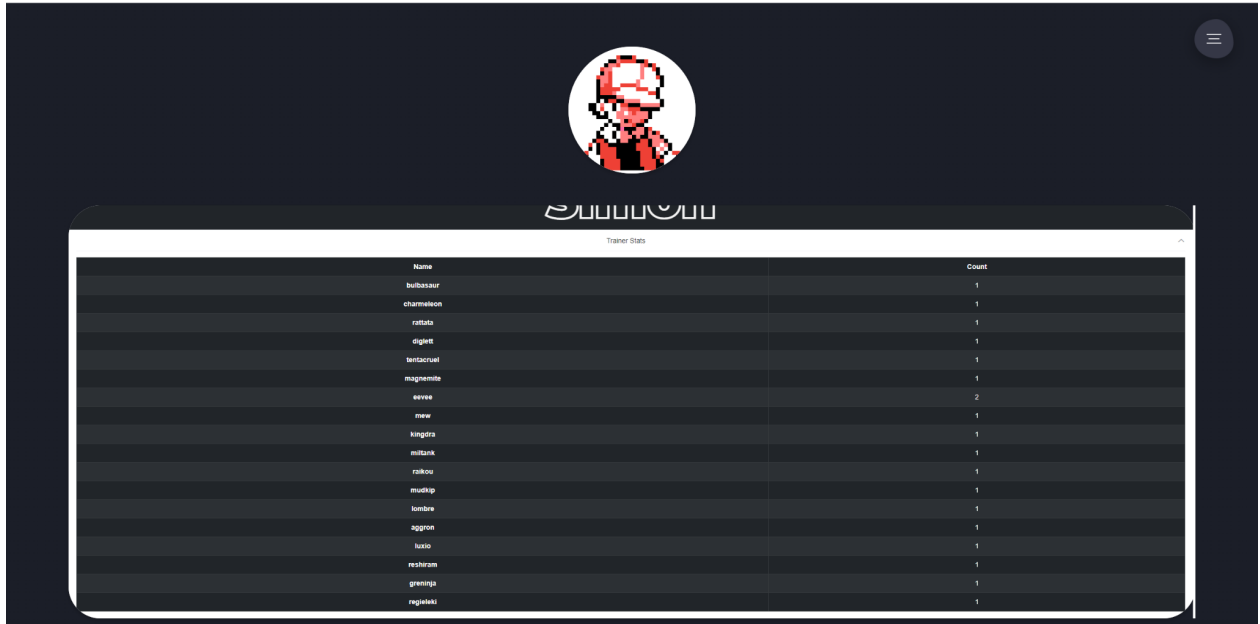
## AGGREGATION WITH GROUP BY

```
ync def getPokemonOwnedCount(self):
    SQL = (f"SELECT tp.pokemon_specific_id, pg.name, COUNT(tp.pokemon_specific_id) FROM trained_pokemon tp "
          f"INNER JOIN trainer AS tr ON tp.trainer_id = tr.trainer_id "
          f"INNER JOIN pokemon_specific AS ps ON tp.pokemon_specific_id = ps.pokemon_specific_id "
          f"INNER JOIN pokemon_generic AS pg ON pg.pokemon_generic_id = ps.pokemon_generic_id "
          f"WHERE tr.trainer_id = (%s) GROUP BY tp.pokemon_specific_id, pg.name")
    query = await Database.execute(SQL, [self.trainer_id])
    return query
```

## Before Aggregation Query:



## After Aggregation Query:



Name	Count
bulbasaur	1
charmeleon	1
rattata	1
diglett	1
tentacruel	1
magnemite	1
eevee	2
mew	1
kingdra	1
miltank	1
raikou	1
mudkip	1
lombre	1
aggron	1
luxio	1
reshiram	1
greninja	1
regieleki	1

```
cpssc304=> SELECT tp.pokemon_specific_id, pg.name, COUNT(tp.pokemon_specific_id) FROM trained_pokemon tp
INNER JOIN trainer AS tr ON tp.trainer_id = tr.trainer_id
INNER JOIN pokemon_specific AS ps ON tp.pokemon_specific_id = ps.pokemon_specific_id
INNER JOIN pokemon_generic AS pg ON pg.pokemon_generic_id = ps.pokemon_generic_id
WHERE tr.trainer_id = 5 GROUP BY tp.pokemon_specific_id, pg.name;
```

pokemon_specific_id	name	count
1	bulbasaur	1
86	charmeleon	1
376	rattata	1
1000	diglett	1
1454	tentacruel	1
1612	magnemite	1
2652	eevee	2
3003	mew	1
4352	kingdra	1
4540	miltank	1
4580	raikou	1
4812	mudkip	1
5000	lombre	1
5484	aggron	1
6762	luxio	1
8169	reshiram	1
8212	greninja	1
8860	regieleki	1

(18 rows)

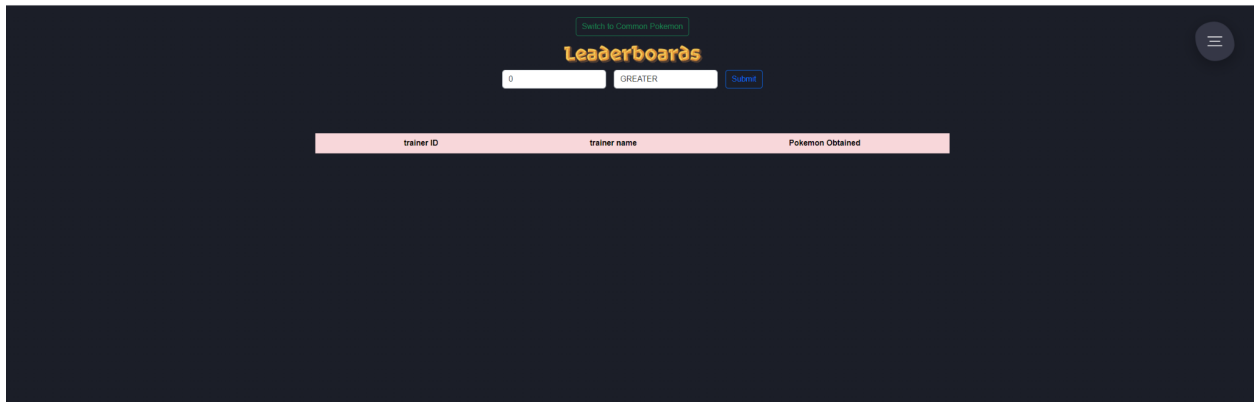
AGGREGATION WITH HAVING

```
Database.execute(SQL,[self.trained_pokemon_id])

def getLeaderboard(range, operator):
    f"SELECT tr.trainer_id, tr.name, COUNT(*) FROM trained_pokemon tp INNER JOIN trainer AS tr ON tp.trainer_id = tr.trainer_id "\
    f"INNER JOIN pokemon_specific AS ps ON ps.pokemon_specific_id = tp.pokemon_specific_id "\
    f"INNER JOIN pokemon_generic pg ON pg.pokemon_generic_id = ps.pokemon_generic_id "\
    f"INNER JOIN pokemon_stat pst ON pst.pokemon_generic_id = pg.pokemon_generic_id "\
    f"GROUP BY tr.trainer_id, tr.name HAVING COUNT(*) {operator} {range} ORDER BY COUNT(*) DESC, tr.name"

    = await Database.execute(SQL, [])
    query
```

Before AGGREGATION WITH HAVING:



After AGGREGATION WITH HAVING:

Switch to Common Pokemon

### Leaderboards

GREATER

Submit

trainer ID	trainer name	Pokemon Obtained
5	Simon	19
1	Cynthia	8

```
cp304=> SELECT tr.trainer_id, tr.name, COUNT(*) FROM trained_pokemon tp INNER JOIN trainer AS tr ON tp.trainer_id = tr.trainer_id
INNER JOIN pokemon_specific AS ps ON ps.pokemon_specific_id = tp.pokemon_specific_id
INNER JOIN pokemon_generic pg ON pg.pokemon_generic_id = ps.pokemon_generic_id
INNER JOIN pokemon_stat pst ON pst.pokemon_generic_id = pg.pokemon_generic_id
GROUP BY tr.trainer_id, tr.name HAVING COUNT(*) > 7 ORDER BY COUNT(*) DESC, tr.name;
trainer_id | name | count
-----+-----+-----
5 | Simon | 19
1 | Cynthia | 8
(2 rows)
```

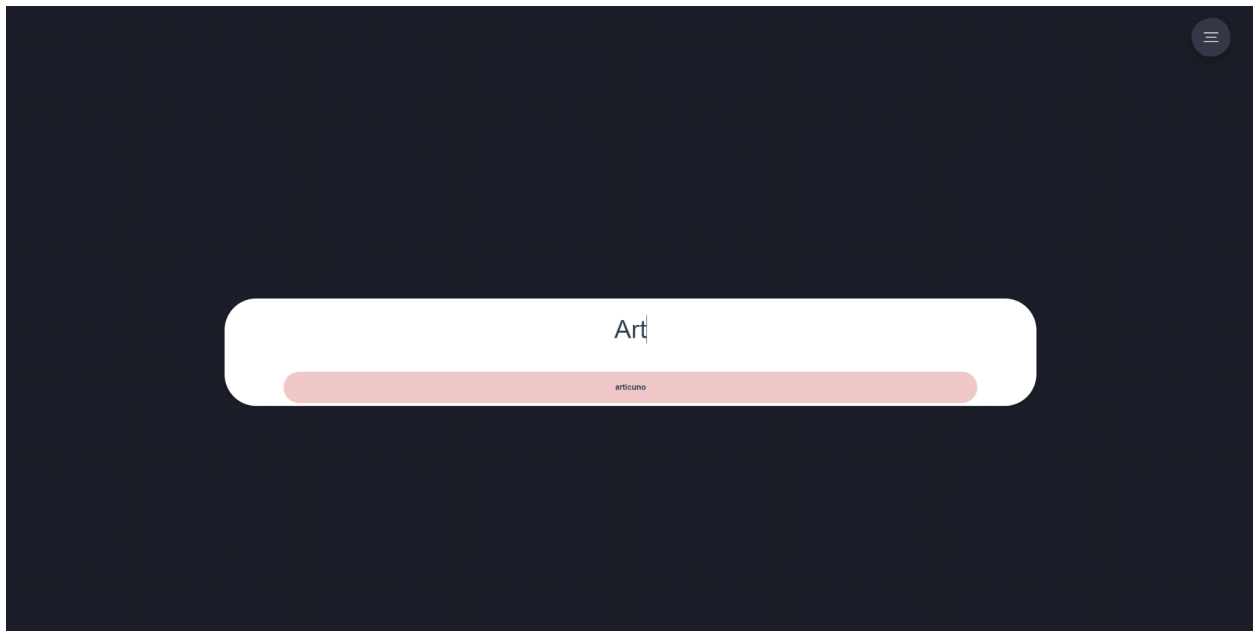
NESTED AGGREGATION GROUP BY

```
@staticmethod
    async def getPokemonAreaCountPerRegion(generic_id):
        innerSQL = (f"SELECT ps.pokemon_specific_id, ps.version_id, COUNT(*) AS areaCount "
        f"FROM pokemon_specific as ps "
        f"INNER JOIN pokemon_area as pa ON ps.pokemon_specific_id = pa.pokemon_specific_id "
        f"INNER JOIN pokemon_generic as pg ON pg.pokemon_generic_id = ps.pokemon_generic_id "
        f"WHERE pg.pokemon_generic_id=(%s) GROUP BY ps.pokemon_specific_id, ps.version_id")

        SQL = (f"SELECT r.region_id, r.name, SUM(ps.areaCount) "
        f"FROM ({innerSQL}) AS ps, version AS v, version_group AS vg, region AS r "
        f"WHERE vg.version_group_id = v.version_group_id AND ps.version_id = v.version_id AND r.generation_id = vg.generation_id "
        f"GROUP BY r.region_id")
        query = await Database.execute(SQL,[generic_id])
        print(query)
        return query

def initPokemon(self):
    self.versionIdList = []
```

BEFORE NESTED AGGREGATION GROUP BY:



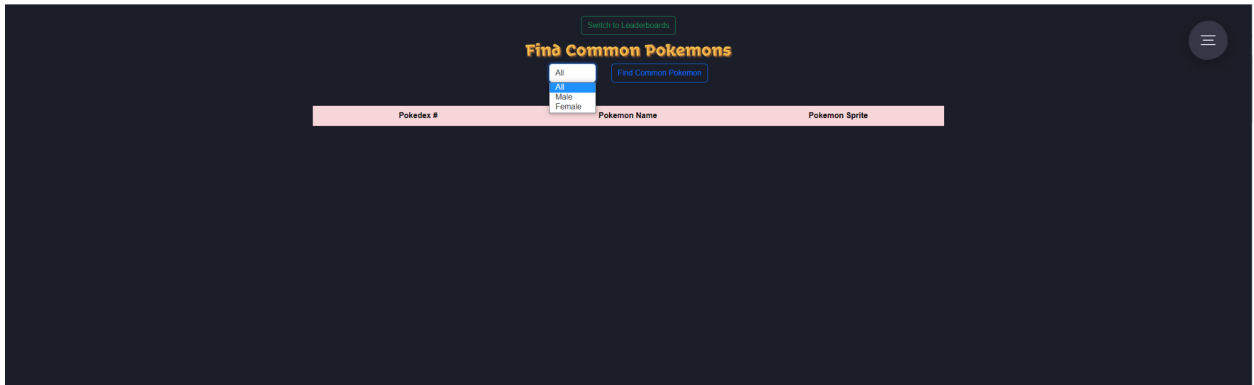
AFTER NESTED AGGREGATION GROUP BY:

Pokemon Area Count Location By Region	
Region Name	Area Count
kanto	3
hoenn	2
sinnoh	3

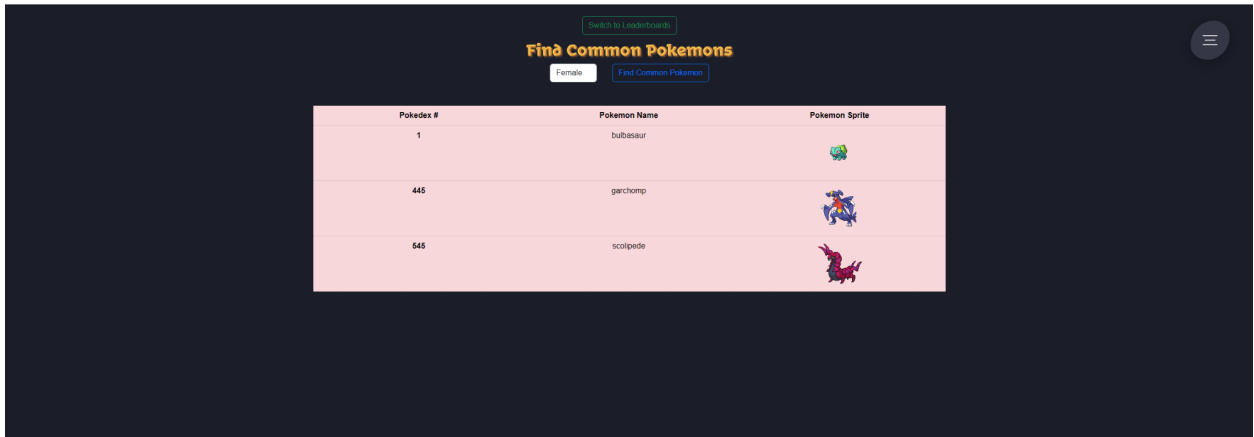
DIVISION

```
def findPokemonThatAllTrainer(gender):
    gender is not None:
        SQL = f"SELECT DISTINCT pg.pokemon_generic_id, pg.name FROM pokemon_specific ps "\
            f"INNER JOIN pokemon_generic pg ON pg.pokemon_generic_id = ps.pokemon_generic_id "\
            f"WHERE NOT EXISTS((SELECT t.trainer_id FROM trainer t WHERE t.gender = {gender}) EXCEPT "\
            f"(SELECT t.trainer_id FROM trained_pokemon tp INNER JOIN trainer t ON t.trainer_id = tp.trainer_id WHERE ps.pokemon_specific_id = tp.pokemon_specific_id)) "
    else:
        SQL = f"SELECT DISTINCT pg.pokemon_generic_id, pg.name FROM pokemon_specific ps " \
            f"INNER JOIN pokemon_generic pg ON pg.pokemon_generic_id = ps.pokemon_generic_id " \
            f"WHERE NOT EXISTS((SELECT t.trainer_id FROM trainer t) EXCEPT " \
            f"(SELECT t.trainer_id FROM trained_pokemon tp INNER JOIN trainer t ON t.trainer_id = tp.trainer_id WHERE ps.pokemon_specific_id = tp.pokemon_specific_id)) "
    query = await Database.execute(SQL, [])
    return query
```

BEFORE DIVISION QUERY:



AFTER DIVISION QUERY:



```
cpsc304=> SELECT DISTINCT pg.pokemon_generic_id, pg.name, pg.sprite FROM pokemon_specific ps
INNER JOIN pokemon_generic pg ON pg.pokemon_generic_id = ps.pokemon_generic_id
WHERE NOT EXISTS((SELECT t.trainer_id FROM trainer t WHERE t.gender = 'Female') EXCEPT
(SELECT t.trainer_id FROM trained_pokemon tp INNER JOIN trainer t ON t.trainer_id = tp.trainer_id WHERE ps.pokemon_specific_id = tp.pokemon_specific_id)) ORDER BY pg.pokemon_generic_id ASC;
```

pokemon_generic_id	name	sprite
1	bulbasaur	<a href="https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/1.png">https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/1.png</a>
445	garchomp	<a href="https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/445.png">https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/445.png</a>
545	scolipede	<a href="https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/545.png">https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/545.png</a>

(3 rows)