Homework 7: Databases

In this homework, you will be populating a database table with information about Pokemon (we have provided the cache data in pokemon.txt) and write code to fetch data from the table.

Pokemon is a video game series from Nintendo with a corresponding card game, television show, and several movies. Pokemon (short for pocket monsters) are creatures with special abilities that can be caught and taught to battle by Pokemon Trainers. Each pokemon has a unique a level of stats that affect their fighting ability, including hit points (HP), speed, attack power, and defense power. Many Pokemon can evolve into new forms from gaining battle experience. Every pokemon has a type, some have more than one, that defines what type of abilities they can use (such as water, fire, or grass). Pokemon are most often caught by players to battle other pokemon, but other activities exist for players and pokemon to take part in such as fashion shows.

We have provided the code for the following:

1. To read the cache data (**readDataFromFile()** function)
2. To create the database and set up the connection and cursor (**setUpDatabase()** function)
3. To set up one of the tables, called Types, in the database (**setUpTypesTable()** function): Run the starter code and then check the structure of the Types table in the DB Browser.

When done with the assignment, your database will have two tables, including the one we have provided and the one that you will write code to create and fill.

We have also provided test cases that will pass if the functions are written correctly. You may not edit the test cases in any way.

NOTE: It is okay for the extra credit test case to fail if you do not attempt the extra credit (**test_pokemon_of_type**)

Tasks

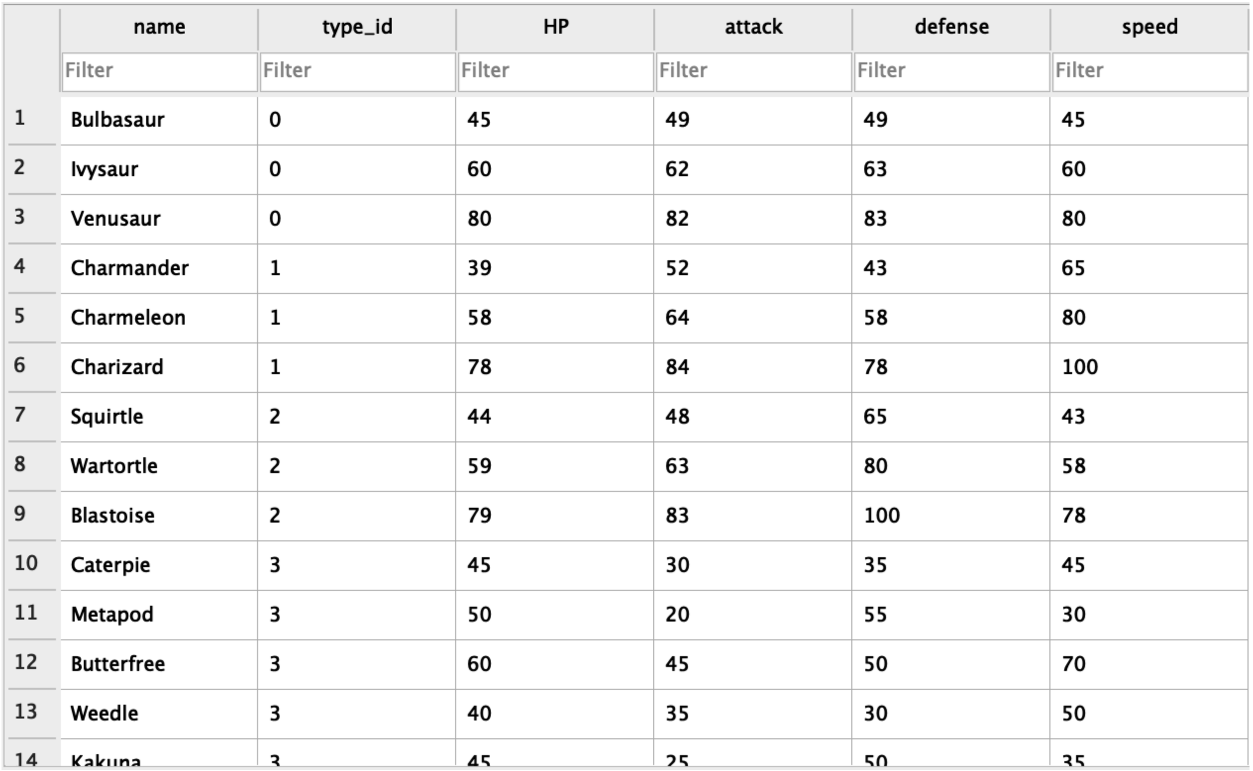
1. **setUpPokemonTable() function:** The function takes three arguments as input: the data (written in JSON format), the database cursor, and database connection object. It loads all of the pokemon in the data into a table called Pokemon. The function does not return anything. Some Pokemon have multiple types, we only want the first type. You want to make sure all the names are unique (hint: use a constraint).

The table should have the following columns:

a. name (datatype: text and primary key)
b. type_id (datatype: integer)
c. HP (datatype: integer)
d. attack (datatype: integer)
e. defense (datatype: integer)
f. speed (datatype: integer)

To find the type_id for each pokemon, you will have to look up the type of each pokemon in the type table we create for you. See **setUpTypesTable** for details.

Expected Table in DB Browser:



	name	type_id	HP	attack	defense	speed
	Filter	Filter	Filter	Filter	Filter	Filter
1	Bulbasaur	0	45	49	49	45
2	Ivysaur	0	60	62	63	60
3	Venusaur	0	80	82	83	80
4	Charmander	1	39	52	43	65
5	Charmeleon	1	58	64	58	80
6	Charizard	1	78	84	78	100
7	Squirtle	2	44	48	65	43
8	Wartortle	2	59	63	80	58
9	Blastoise	2	79	83	100	78
10	Caterpie	3	45	30	35	45
11	Metapod	3	50	20	55	30
12	Butterfree	3	60	45	50	70
13	Weedle	3	40	35	30	50
14	Kakuna	3	45	25	50	35

(To find the type_id for each pokemon, you will have to look up the type of each pokemon in the type table we create for you. See **setUpTypeTable** for details)

2. **getPokemonByHP() function:** The function takes three arguments as input: an HP, the database cursor, and database connection object. It selects all the pokemon for a particular HP and returns a list of tuples. Each tuple contains the pokemon name, type_id, and HP.

   Expected output for pokemon with HP of “`50`”:

```
[('Cloyster', 2, 50), ('Cubone', 7, 50), ('Hitmonchan', 12, 50),
('Hitmonlee', 12, 50), ('Magneton', 6, 50), ('Metapod', 3, 50),
('Sandshrew', 7, 50), ('Trubbish', 5, 50)]
```

3. **getPokemonByHPAboveSpeedAndAboveAttack() function:** The function takes five arguments as input: the HP, the speed, the attack, the database cursor, and database connection object. It selects all the pokemon at an HP, speed greater than the rating passed to the function, with an attack greater than the attack passed into the function, and returns a list of tuples. Each tuple in the list contains the pokemon name, speed, attack, and defense.

   Expected Output for pokemon with HP = 60, speed > 20 and attack > 85:

   ```
   [('Arbok', 80, 95, 69), ('Raichu', 110, 90, 55),
   ('Parasect', 30, 95, 80), ('Dodrio', 110, 110, 70),
   ('Zoroark', 105, 105, 60)]
   ```

4. **getPokemonAboveSpeedAboveDefenseOfType() function:** The function takes five arguments as input: a speed, a defense, a type (not type_id), the database cursor, and database connection object. It returns a list of tuples for all of the pokemon that match that type and have a speed greater than the speed passed to the function, and a defense greater than the defense passed to the function. Each tuple in the list should contain the pokemon name, type (not type_id), speed, and defense.
   Note: You have to use a JOIN for this task.

   Expected Output for speed > 50, defense > 60, and type “Grass”:

   ```
   [('Exeggutor', 'Grass', 55, 85), ('Ivysaur', 'Grass', 60, 63),
   ('Tangela', 'Grass', 60, 115), ('Venusaur', 'Grass', 80, 83)]
   ```

Grading Rubric

1. **setUpPokemonTable()** - 25 points
   a. 10 points for entering all 106 pokemon in the table
   b. 5 points for creating all 6 columns in the table
   c. 10 points for using the correct type for each column

2. **getPokemonByHP()** - 10 points
   a. 5 points for returning the correct number of pokemon by HP
   b. 5 points for returning the three columns: pokemon name, type_id, and HP

3. **getPokemonByHPAboveSpeedAndAboveAttack()** - 10 points

   a. 5 points for returning a list of tuples of pokemon at the HP and above or equal to a speed and attack.
   b. 5 points for returning all four columns: pokemon name, speed, attack, and defense.

4. **getPokemonAboveSpeedAboveAttackOfType()** - 15 points
   a. 10 points for correctly using a JOIN to get the rows
   b. 5 points for correctly outputting a list of tuples with pokemon name, type, speed, and defense.

Git Commits

Make at least 3 git commits before the deadline. Each commit is worth 5 points. Please upload a link to your GitHub repository URL to Canvas.

Extra Credit - 6 points

getPokemonOfType() -- this function takes in 5 parameters: type attack, defense, the database cursor, and database connection object. It returns a list of all of the pokemon names that match the type, are greater than or equal to that attack, and match that defense.

Write more tests for **getPokemonOfType()** in **test_pokemon_of_type_extra_credit**. Write at least 3 assert statements to receive full points.