Queries TODO:

Note: main data is ***pokemon\_species***, we probably won’t use ***pokemon*** directly because it contains mega and special forms of certain pokemon, thus introduce duplications.

1. Given national pokedex id, find the name of the pokemon (pokemon\_species)
2. Given national pokedex id, find the stats of the pokemon (pokemon\_stats & stats)
3. Given national pokedex id, find the type of pokemon (pokemon\_types & types)
4. Given national pokedex id, find the generation of that pokemon (generations & pokemon\_species)
5. Given national pokedex id, find what is the id of the pokemon it evolves from (pokemon\_species @ evolves\_from\_species\_id)
6. Given national pokedex id, find the id of all ancestor and descendant of that pokemon (pokemon\_species @ evolution\_chain\_id)
7. Given generation id, find the region (generations & regions)
8. Given ancestor id and pokemon id, find all evolve conditions (pokemon\_evolution)
9. Given generation id, find all pokemon of that generation (pokemon\_species)
10. Given condition stats, find all ids of qualified pokemon (pokemon\_stats & stats)

e.g. find all pokemons whose HP base stat >= 80

1. Given stats sum, find all ids of qualified pokemon (pokemon\_stats)

e.g. find all pokemons whose stats sum >= 600 THIS IS HARD

1. Given certain type, find ids of all pokemon of that type (pokemon\_types & types)

Note: some pokemon have 2 types, be careful when write query

1. Given Pokemon name, return [id, identifier, type] (pokemon\_species & pokemon\_types & types) detailed requirement see .sql file

**Certain query should be made to have formatted output: [id, identifier, type]. Duplications caused by two-type now can be handled by the program. MAKE LIFE EASIER**

TODO: Query which need reformat:

5; 6; 10; 11; 12;