



1. Considere a query a seguir e responda. É um SQL funcional ou possui algum erro? Caso possua algum erro, comente e corrija.

```
select count(distinct origin_registration_id), new_status,  
old_status,  
from registrations_fact  
where new_status == (1, 2, 3, 4, 5, 6)  
order by new_status  
group by new_status;
```

2. Conecte-se à PokeAPI através da biblioteca pokebase, e retorne em um gráfico de barras, os dados de tipo dominante dos 100 primeiros pokémons disponíveis, de forma a agrupá-los e mostrá-los em ordem decrescente.

DICAS:

```
pip install pokebase ##Instalação da biblioteca  
import pokebase as pb  
pokemon = pb.pokemon(1)
```

Você pode chamar os pokémons pelo nome, mas a não ser que saiba o nome de todos os 100 primeiros, chame pelo id. Como na dica acima.

```
pokemon.types[0].type.name ##Para acessar o tipo do pokemon
```

3. Ainda usando a PokeAPI, crie e retorne um dataframe com os dados dos 50 primeiros pokémons, incluindo, nome, peso, altura e tipo.

4. Dado o código a seguir, transforme para o framework FastAPI.

```
from flask import Flask  
  
app = Flask(__name__)  
  
@app.route("/")  
def hello_world():  
    return "<p>Hello, World!</p>"
```