In [ ]:

```python
import requests
import json
from IPython.display import Image, display
import plotly.express as px
import pandas as pd
from pandas import read_csv
import numpy as np
import plotly.graph_objects as go
import sqlite3
from pandas import json_normalize
```

Table 1 General Info pokemon

In [ ]:

```python
# Define column names
column_names = ['id', 'name', 'height', 'order', 'weight', 'types']

# Initialize an empty list to store data
pokemon_data = []

# Iterate through Pokémon IDs
for id in range(1,1026 ):
    # Make the request to the PokeAPI
    request_url = f"https://pokeapi.co/api/v2/pokemon/{id}/"
    response = requests.get(request_url)
    data = json.loads(response.text)
        # Extract relevant data
    pokemon_info = {
        'id': data['id'],
        'name': data['name'],
        'height': data['height'],
        'order': data['order'],
        'weight': data['weight'],
        'types': data['types']
    }

    # Append the Pokémon info to the list
    pokemon_data.append(pokemon_info)

# Convert the list of dictionaries into a DataFrame
poke_df = pd.DataFrame(pokemon_data, columns=column_names)

# Display the DataFrame
poke_df
```

Out[ ]:

	id	name	height	order	weight	types
0	1	bulbasaur	7	1	69	[{'slot': 1, 'type': {'name': 'grass', 'url': ...
1	2	ivysaur	10	2	130	[{'slot': 1, 'type': {'name': 'grass', 'url': ...
2	3	venusaur	20	3	1000	[{'slot': 1, 'type': {'name': 'grass', 'url': ...
3	4	charmander	6	5	85	[{'slot': 1, 'type': {'name': 'fire', 'url': '...
4	5	charmeleon	11	6	190	[{'slot': 1, 'type': {'name': 'fire', 'url': '...
...	...	...	...	...	...	...
1020	1021	raging-bolt	52	1105	4800	[{'slot': 1, 'type': {'name': 'electric', 'url...
1021	1022	iron-boulder	15	1106	1625	[{'slot': 1, 'type': {'name': 'rock', 'url': '...
1022	1023	iron-crown	16	1107	1560	[{'slot': 1, 'type': {'name': 'steel', 'url': ...
1023	1024	terapagos	2	1108	65	[{'slot': 1, 'type': {'name': 'normal', 'url':...
1024	1025	pecharunt	3	1109	3	[{'slot': 1, 'type': {'name': 'poison', 'url':...

1025 rows × 6 columns

In [ ]:

```
# Function to extract type names
def extract_type_name(types_data, slot):
    for t in types_data:
        if t['slot'] == slot:
            return t['type']['name']
    return None

# Apply function to create primary_type and secondary_type columns
poke_df['primary_type'] = poke_df['types'].apply(lambda x: extract_type_name(x, 1))
poke_df['secondary_type'] = poke_df['types'].apply(lambda x: extract_type_name(x, 2))

# Drop the original 'types' column
poke_df.drop(columns=['types'], inplace=True)
```

In [ ]:

```
# Connect to SQLite database (create if it doesn't exist)
conn = sqlite3.connect('pokemon_data.db')

# Convert DataFrame to SQLite table
poke_df.to_sql('pokedex', conn, if_exists='replace', index=False)

# Commit changes and close connection
conn.commit()
conn.close()
```

Table 2 : Poke-Stats

In [ ]:

```
# Function to fetch data from the PokéAPI for a given Pokémon ID or name
def fetch_pokemon_data(id):
    url = f"https://pokeapi.co/api/v2/pokemon/{id}/"
    response = requests.get(url)
    if response.status_code == 200:
        return response.json()
    else:
        print("Error:", response.status_code)
        return None
pokemon_stats_info = []
# Example: Fetching data for Pokémon with ID 1 (Bulbasaur)
for i in range(1,1026):
    pokemon_data = fetch_pokemon_data(i)
    # Extract 'id', 'name', and 'stats' from the fetched data
    pokemon_stats = {
        'id': pokemon_data['id'],
        'name': pokemon_data['name'],
        'stats': pokemon_data['stats']
    }
    pokemon_stats_info.append(pokemon_stats)

# Create DataFrame for Pokémon stats
poke_stats_df = pd.DataFrame(pokemon_stats_info)

# Display the DataFrame
poke_stats_df
```

Out[ ]:

	id	name	stats
0	1	bulbasaur	[{'base_stat': 45, 'effort': 0, 'stat': {'name...
1	2	ivysaur	[{'base_stat': 60, 'effort': 0, 'stat': {'name...
2	3	venusaur	[{'base_stat': 80, 'effort': 0, 'stat': {'name...
3	4	charmander	[{'base_stat': 39, 'effort': 0, 'stat': {'name...
4	5	charmeleon	[{'base_stat': 58, 'effort': 0, 'stat': {'name...
...	...	...	...
1020	1021	raging-bolt	[{'base_stat': 125, 'effort': 0, 'stat': {'nam...
1021	1022	iron-boulder	[{'base_stat': 90, 'effort': 0, 'stat': {'name...
1022	1023	iron-crown	[{'base_stat': 90, 'effort': 0, 'stat': {'name...
1023	1024	terapagos	[{'base_stat': 90, 'effort': 0, 'stat': {'name...
1024	1025	pecharunt	[{'base_stat': 88, 'effort': 0, 'stat': {'name...

1025 rows × 3 columns

In [ ]:

```
# Extract data from 'stats' column
for row in poke_stats_df.itertuples():
    stats_data = row.stats
    stat_values = {}
    for stat in stats_data:
        stat_name = stat['stat']['name']
        base_stat = stat['base_stat']
        stat_values[stat_name] = base_stat

    # Update the DataFrame with the extracted values
    poke_stats_df.loc[row.Index, stat_values.keys()] = stat_values.values()

# Display the updated DataFrame
poke_stats_df
```

Out[ ]:

	id	name	stats	hp	attack	defense	special-attack	special-defense	speed
0	1	bulbasaur	[{'base_stat': 45, 'effort': 0, 'stat': {'name...	45.0	49.0	49.0	65.0	65.0	45.0
1	2	ivysaur	[{'base_stat': 60, 'effort': 0, 'stat': {'name...	60.0	62.0	63.0	80.0	80.0	60.0
2	3	venusaur	[{'base_stat': 80, 'effort': 0, 'stat': {'name...	80.0	82.0	83.0	100.0	100.0	80.0
3	4	charmander	[{'base_stat': 39, 'effort': 0, 'stat': {'name...	39.0	52.0	43.0	60.0	50.0	65.0
4	5	charmeleon	[{'base_stat': 58, 'effort': 0, 'stat': {'name...	58.0	64.0	58.0	80.0	65.0	80.0
...	...	...	...	...	...	...	...	...	...
1020	1021	raging-bolt	[{'base_stat': 125, 'effort': 0, 'stat': {'nam...	125.0	73.0	91.0	137.0	89.0	75.0
1021	1022	iron-boulder	[{'base_stat': 90, 'effort': 0, 'stat': {'name...	90.0	120.0	80.0	68.0	108.0	124.0
1022	1023	iron-crown	[{'base_stat': 90, 'effort': 0, 'stat': {'name...	90.0	72.0	100.0	122.0	108.0	98.0
1023	1024	terapagos	[{'base_stat': 90, 'effort': 0, 'stat': {'name...	90.0	65.0	85.0	65.0	85.0	60.0
1024	1025	pecharunt	[{'base_stat': 88, 'effort': 0, 'stat': {'name...	88.0	88.0	160.0	88.0	88.0	88.0

1025 rows × 9 columns

In [ ]:

```
# Connect to SQLite database (create if it doesn't exist)
conn = sqlite3.connect('pokemon_data.db')

# Convert list-like columns to JSON strings
poke_stats_df['stats'] = poke_stats_df['stats'].apply(json.dumps)  # Convert stats lis

# Convert DataFrame to SQLite table
poke_stats_df.to_sql('stats', conn, if_exists='replace', index=False)

# Commit changes and close connection
conn.commit()
conn.close()
```

Table 3: Evolutions

In [ ]:

```
# Define column names
column_names = ['id', 'name', 'order', 'evolves_from_species', 'evolution_chain','gene

# Initialize an empty list to store data
pokemon_species_data = []

# Iterate through Pokémon IDs
for id in range(1,1026):
    # Make the request to the PokeAPI
    request_url = f"https://pokeapi.co/api/v2/pokemon-species/{id}/"
    response = requests.get(request_url)
    data = json.loads(response.text)
        # Extract relevant data
    pokemon_info = {
        'id': data['id'],
        'name': data['name'],
        'order': data['order'],
        'evolves_from_species': data['evolves_from_species'],
        'evolution_chain': data['evolution_chain'],
        'generation': data['generation']
    }

    # Append the Pokémon info to the list
    pokemon_species_data.append(pokemon_info)

# Convert the list of dictionaries into a DataFrame
poke_species_df = pd.DataFrame(pokemon_species_data, columns=column_names)

# Display the DataFrame
poke_species_df
```

Out[ ]:

	id	name	order	evolves_from_species	evolution_chain	generatior
0	1	bulbasaur	1	None	{'url': 'https://pokeapi.co/api/v2/evolution-c...	{'name' 'generation-i' 'url' 'https://pokea..
1	2	ivysaur	2	{'name': 'bulbasaur', 'url': 'https://pokeapi....	{'url': 'https://pokeapi.co/api/v2/evolution-c...	{'name' 'generation-i' 'url' 'https://pokea..
2	3	venusaur	3	{'name': 'ivysaur', 'url': 'https://pokeapi.co...	{'url': 'https://pokeapi.co/api/v2/evolution-c...	{'name' 'generation-i' 'url' 'https://pokea..
3	4	charmander	4	None	{'url': 'https://pokeapi.co/api/v2/evolution-c...	{'name' 'generation-i' 'url' 'https://pokea..
4	5	charmeleon	5	{'name': 'charmander', 'url': 'https://pokeapi...	{'url': 'https://pokeapi.co/api/v2/evolution-c...	{'name' 'generation-i' 'url' 'https://pokea..
...	...	...	...	...	...	..
1020	1021	raging-bolt	1023	None	{'url': 'https://pokeapi.co/api/v2/evolution-c...	{'name' 'generation-ix' 'url' 'https://poke..
1021	1022	iron-boulder	1024	None	{'url': 'https://pokeapi.co/api/v2/evolution-c...	{'name' 'generation-ix' 'url' 'https://poke..
1022	1023	iron-crown	1025	None	{'url': 'https://pokeapi.co/api/v2/evolution-c...	{'name' 'generation-ix' 'url' 'https://poke..
1023	1024	terapagos	1026	None	{'url': 'https://pokeapi.co/api/v2/evolution-c...	{'name' 'generation-ix' 'url' 'https://poke..
1024	1025	pecharunt	1027	None	{'url': 'https://pokeapi.co/api/v2/evolution-c...	{'name' 'generation-ix' 'url' 'https://poke..

1025 rows × 6 columns

In [ ]:

```
# Extract 'name' from 'generation' column
poke_species_df['generation'] = poke_species_df['generation'].apply(lambda x: x['name'
# Extract 'name' from 'evolves_from_species' column
poke_species_df['evolves_from_species'] = poke_species_df['evolves_from_species'].appl
```

In [ ]:

```
# Add 'evolves_to' column
poke_species_df['evolves_to'] = None

# Iterate over each row
for index, row in poke_species_df.iterrows():
    # Initialize an empty list to store potential evolutions
    evolves_to = []

    # Iterate over each row again to check for potential evolutions
    for _, next_row in poke_species_df.iterrows():
        if row['name'] == next_row['evolves_from_species']:
            evolves_to.append(next_row['name'])  # Add potential evolution to the list

    # If there are potential evolutions, assign them to the 'evolves_to' column
    if evolves_to:
        poke_species_df.at[index, 'evolves_to'] = evolves_to

# Display the DataFrame
poke_species_df
```

Out[ ]:

	id	name	order	evolves_from_species	evolution_chain	generation	
0	1	bulbasaur	1	None	{'url': 'https://pokeapi.co/api/v2/evolution-c...	generation-i	
1	2	ivysaur	2	bulbasaur	{'url': 'https://pokeapi.co/api/v2/evolution-c...	generation-i	
2	3	venusaur	3	ivysaur	{'url': 'https://pokeapi.co/api/v2/evolution-c...	generation-i	
3	4	charmander	4	None	{'url': 'https://pokeapi.co/api/v2/evolution-c...	generation-i	[
4	5	charmeleon	5	charmander	{'url': 'https://pokeapi.co/api/v2/evolution-c...	generation-i	
...	...	...	...	...	...	...	
1020	1021	raging-bolt	1023	None	{'url': 'https://pokeapi.co/api/v2/evolution-c...	generation-ix	
1021	1022	iron-boulder	1024	None	{'url': 'https://pokeapi.co/api/v2/evolution-c...	generation-ix	
1022	1023	iron-crown	1025	None	{'url': 'https://pokeapi.co/api/v2/evolution-c...	generation-ix	
1023	1024	terapagos	1026	None	{'url': 'https://pokeapi.co/api/v2/evolution-c...	generation-ix	
1024	1025	pecharunt	1027	None	{'url': 'https://pokeapi.co/api/v2/evolution-c...	generation-ix	

1025 rows × 7 columns

In [ ]:

```
poke_species_df['generation'] = poke_species_df['generation'].str.replace('generation-
```

In [ ]:

```
poke_species_df['generation'] = poke_species_df['generation'].str.replace('ix', '9')
poke_species_df['generation'] = poke_species_df['generation'].str.replace('viii', '8')
poke_species_df['generation'] = poke_species_df['generation'].str.replace('vii', '7')
poke_species_df['generation'] = poke_species_df['generation'].str.replace('vi', '6')
poke_species_df['generation'] = poke_species_df['generation'].str.replace('iv', '4')
poke_species_df['generation'] = poke_species_df['generation'].str.replace('v', '5')
poke_species_df['generation'] = poke_species_df['generation'].str.replace('iii', '3')
poke_species_df['generation'] = poke_species_df['generation'].str.replace('ii', '2')
poke_species_df['generation'] = poke_species_df['generation'].str.replace('i', '1')
```

In [ ]:

```
poke_species_df
```

Out[ ]:

	id	name	order	evolves_from_species	evolution_chain	generation
0	1	bulbasaur	1	None	{'url': 'https://pokeapi.co/api/v2/evolution-c...	1
1	2	ivysaur	2	bulbasaur	{'url': 'https://pokeapi.co/api/v2/evolution-c...	1
2	3	venusaur	3	ivysaur	{'url': 'https://pokeapi.co/api/v2/evolution-c...	1
3	4	charmander	4	None	{'url': 'https://pokeapi.co/api/v2/evolution-c...	1
4	5	charmeleon	5	charmander	{'url': 'https://pokeapi.co/api/v2/evolution-c...	1
...	...	...	...	...	...	...
1020	1021	raging-bolt	1023	None	{'url': 'https://pokeapi.co/api/v2/evolution-c...	9
1021	1022	iron-boulder	1024	None	{'url': 'https://pokeapi.co/api/v2/evolution-c...	9
1022	1023	iron-crown	1025	None	{'url': 'https://pokeapi.co/api/v2/evolution-c...	9
1023	1024	terapagos	1026	None	{'url': 'https://pokeapi.co/api/v2/evolution-c...	9
1024	1025	pecharunt	1027	None	{'url': 'https://pokeapi.co/api/v2/evolution-c...	9

1025 rows × 7 columns

In [ ]:

```
# Convert the 'generation' column to integer type
poke_species_df['generation'] = poke_species_df['generation'].astype(int)
```

In [ ]:

```
# Connect to SQLite database (create if it doesn't exist)
conn = sqlite3.connect('pokemon_data.db')

# Convert list/dict-like columns to JSON strings
poke_species_df['evolution_chain'] = poke_species_df['evolution_chain'].apply(json.dum
poke_species_df['evolves_to'] = poke_species_df['evolves_to'].apply(json.dumps)

# Convert DataFrame to SQLite table
poke_species_df.to_sql('evolutions', conn, if_exists='replace', index=False)

# Commit changes and close connection
```

```
conn.commit()
conn.close()
```

In [ ]: