Designing a database for Pokemon with GenAI

Pankhuri Asthana

May 2024

1 Introduction

We're going to simplify Pokemon to just a couple of mechanics that are used in the game. Pokemon can have one or two 'types,' which decide whether they're more effective or less effective against other Pokemon types. Every pokemon has a primay type; some also have a secondary type.

The game involves using moves to attack other Pokemon, and each move has a certain power and type. Every move has a set of Pokemon who are capable of learning it; and every Pokemon has a set of moves it can learn.

2 Problem Statement

In this assignment, I was tasked with designing a database system for managing Pokémon data via two different approaches: SQL and NoSQL. SQL databases are relational databases that use structured query language (SQL) to manage data, while NoSQL databases provide a non-relational data model and offer flexibility in data storage and retrieval.

1. Create all the tables needed.
2. With the following details, populate the tables:

   - Bulbasaur is a pokemon of Grass type.
   - Charmander is a pokemon of Fire type.
   - Squirtle is a pokemon of Water type.
   - Eevee is a pokemon of Normal type
   - Pidgey is a pokemon of the Normal/Flying type.
   - Bulbasaur can learn Tackle, Vine Whip, and Return.
   - Charmander can learn Tackle, Ember, and Return.
   - Squirtle can learn Tackle, Water Gun, and Return.

- Eevee can learn Tackle, Headbutt, and Return.
- Pidgey can learn Tackle, Wing Attack, and Return.
- Tackle has 35 power and is Normal type.
- Water Gun has 40 power and is Water type.
- Ember has 40 power and is Fire type.
- Vine Whip has 40 power and is Grass type.
- Wing attack has 65 power and is Flying type.
- Headbutt has 70 power and is Normal type.
- Return has 100 power and is Normal type.
- Fire is powerful against Grass but weak to Water.
- Grass is powerful against Water but weak to both Fire and Flying.
- Water is powerful against Fire but weak to Grass.
- Normal is not weak to anything but not powerful against anything either.
- Flying is powerful against Grass and has no weaknesses.

3. Write a query that returns all the pokemon who can learn 'Return'.
4. Write a query that returns all the moves in the game that are powerful against Grass.

3 Database Design

3.1 SQL Approach:

I designed a relational database schema consisting of four tables: Pokemons, Types, Moves, and PokemonMoves. Each table represents a different entity in the system, and relationships between entities are managed through foreign key constraints(unique ids).

3.2 NoSQL Approach:

In the NoSQL approach, I opted for a document-based data model using MongoDB. I defined three collections: Pokemons, Types, and Moves. Each document in the collections represents a Pokémon, type, or move, and relationships between entities are handled through embedding Arrays of references (i.e. using object ids).

4 Populating Querying Data

4.1 SQL Approach:

Data was inserted into the SQL database using SQL INSERT statements.SQL queries were used to retrieve data from the relational database. For example, to find all Pokémon that can learn a certain move, I used JOIN operations to connect the Pokemons and Moves tables.

4.2 NoSQL Approach:

Data was inserted into MongoDB collections via CSV files which seemed an easier option to me. I made use of pipeline aggregation option which involved stages as tags to filter data as per need. The tag I needed to execute the queries based on return move and grass type (3 and 4) was '$match'.

5 Conclusion

As someone more familiar with SQL, I found it to be an intuitive and comfortable choice, given my familiarity with its workings and commands. However, delving into the realm of NoSQL, particularly with MongoDB, was a valuable learning experience. Despite initial challenges with data population and querying, I found that NoSQL offered a fresh perspective on data retrieval, leveraging built-in filters and flexible document structures. Through the use of ChatGPT and documentation, I was able to successfully design the database schema and execute queries in MongoDB.

Overall, while SQL remains my go-to choice for its familiarity and ease of use, the exploration of NoSQL provided me with a new set of tools and approaches for managing data.