

Pokemon Database using SQL and NoSQL database

May 9, 2024

Chapter 1

solution using GenAI

1.1 Problem Statement

We're going to simplify Pokemon to just a couple of mechanics that are used in the game, and thankfully, you don't need to know how it works to do this exercise.

Pokemon can have one or two 'types,' which decide whether they're more effective or less effective against other Pokemon types. Every Pokémon has a primary 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.

At the very least, we'd need database tables to store Pokemon, Type, and Move. However, 'Pokemon' and 'Move' have a classic many-to-many relationship. How do you deal with this?

1. Create all the tables needed. (5)
2. With the following details, populate the tables: (5)
 1. Bulbasaur is a pokemon of Grass type.
 2. Charmander is a pokemon of Fire type.
 3. Squirtle is a pokemon of Water type.
 4. Eevee is a pokemon of Normal type
 5. Pidgey is a pokemon of the Normal/Flying type.
 6. Bulbasaur can learn Tackle, Vine Whip, and Return.
 7. Charmander can learn Tackle, Ember, and Return.

8. Squirtle can learn Tackle, Water Gun, and Return.
9. Eevee can learn Tackle, Headbutt, and Return.
10. Pidgey can learn Tackle, Wing Attack, and Return.
11. Tackle has 35 power and is Normal type.
12. Water Gun has 40 power and is Water type.
13. Ember has 40 power and is Fire type.
14. Vine Whip has 40 power and is Grass type.
15. Wing attack has 65 power and is Flying type.
16. Headbutt has 70 power and is Normal type.
17. Return has 100 power and is Normal type.
18. Fire is powerful against Grass but weak to Water.
19. Grass is powerful against Water but weak to both Fire and Flying.
20. Water is powerful against Fire but weak to Grass.
21. Normal is not weak to anything but not powerful against anything either.
22. Flying is powerful against Grass and has no weaknesses.
3. Write a query that returns all the Pokemon who can learn 'Return'. (5)
4. Write a query that returns all the moves in the game that are powerful against Grass. (5)

1.2 prompts given

All the prompts given and the conversation are given in the hyperlink provided [Conversation with ChatGPT](#)

1.3 let's discuss the answers given by Chat-GPT

According to me the answers were just satisfactory as it kept missing on some points.

1.3.1 First point

it could only understand the query which is directly executable and for query which needed some abstract logic flow where it made no success even after i told it to change the number of tables

and its approach in some cases
(mainly query 4)

1.3.2 second point

changing the approach of a GenAI chat bot using our own prompts could only work for the superficial tasks
whereas the core problem solving remains the same. it is for that reason that it kept on insisting me to use
5 tables in total in my database whereas the task could be completed using four tables also.

1.4 Queries used

I populated the tables using the data given in both SQL and NoSQL database and the output is given here for your reference
(I was not sure if I shall upload pictures directly in report made by latex).
The document contains the queries and their output [Click here for doc](#)

Chapter 2

Difference between SQL and NOSQL database

2.1 Reflections

SQL databases offer simplicity and ease of maintenance, making them ideal for smaller-scale projects. In contrast, NoSQL databases provide scalability and flexibility, capable of managing large-scale databases. They go beyond simple data retrieval, enabling system management at a larger scale. While SQL databases excel in straightforward data operations, NoSQL databases cater to the complex needs of expansive systems with diverse data structures.

2.2 conclusion

SQL databases are like organized spreadsheets, perfect for structured data found in traditional business applications.

They excel at keeping things neat and orderly, ensuring that data is stored efficiently and can be easily accessed and queried.

They're reliable and guarantee that your data is always consistent and secure, making them ideal for financial systems and reporting.

On the other hand, NoSQL databases are like big boxes where you can throw in any kind of data without worrying too much about

how it's organized. They offer flexibility and scalability, making them well-suited for modern web and mobile applications, where data structures may evolve rapidly and need to handle high volumes of traffic. NoSQL databases prioritize speed and agility over strict organization, making them great for real-time analytics, IoT, and content management systems.

In summary, if your project requires structured data and complex queries with a focus on consistency and reliability, SQL databases are the way to go. But if you need flexibility, scalability, and speed for handling diverse and rapidly changing data, NoSQL databases offer the flexibility and performance you need.