Mod 9 Reflection Note.

Zachary T Meert

I chose to implement an SQL-based character database because I wanted to learn how SQL works in practice. I also wanted my bot to save information in a reliable way, and I was getting tired of having to re-enter character details every time. This felt like a feature that was useful, educational, and honestly something I wanted to try just for fun.

Adding a database improves the bot in several ways. It gives the bot the ability to store characters permanently, which makes it easier for players to keep track of their information across sessions. It also creates a starting point for future features, such as checking which feats or weapons a character qualifies for based on the data they have already provided. This should save time during play because the bot can provide answers instantly instead of relying on manual lookups. The bot simply becomes more organized and more practical for regular use.

There were several challenges while implementing this feature. I had to learn basic SQL, including how to design a table, insert new rows, update them, and query only the data that belongs to a specific user. While doing this, I broke my bot more than once, usually because of mistakes with SQLite connections or by misunderstanding how Discord.py handles asynchronous code. I also ran into errors caused by small typos or by accidentally creating multiple bot instances. Fixing these problems taught me how the bot interacts with the database and helped me understand the structure of the system more clearly.

Overall, building the SQL character database improved the bot and helped me build confidence with SQL. It works consistently now, and it gives me room to add more advanced features in the future.