Yongun Seong (성용운)
2017-19937
DB

Project 1-3: Implementing DML Report

For this part of the project, I implemented DML queries on top of BerkeleyDB. As I had not submitted anything for the previous phase, I also implemented parts of phase 2.

My implementation is split into two parts, the parser and the engine. The parser handles parsing incoming queries. It also performs any validation steps that can be performed with just the query, without querying the database. It then constructs various objects that are passed on to the engine. The engine takes these pristine queries, performs additional validation that can only be done after querying the database, and persists any changes to the Berkley DB file.

I omitted large parts of phase 2 in my implementation. In fact, I only implemented the CREATE TABLE query, and even then, I did not implement most of the validation for table creation. However, given some extra time, I believe it would not be too challenging to add that functionality. Due to time constraints, I chose to omit such code for my submission.

While writing code for this project, I realized how important types are, even for small to medium projects. Thanks to the type annotations I had added to my code, I was able to easily check my code and confidently refactor any parts I had to change without being afraid about how it would affect every other part of the code, or missing code that was depending on the code being changed. The type checking in Python, although not perfect, was usually quite sufficient for my use, and there was a readily available escape hatch whenever the type system interfered with my code. Also, I learned how modular code improves code readability and maintainability. By separating out the parser from the rest of the DB engine, I could focus on a single feature in each part, without having to consider its complex interaction with the rest of the implementation. Again, types were very helpful, as I was able to extensively annotate the communication points between the two modules to make sure both parts would correctly when merged.