

본 프로젝트에서 Javacc를 통해 SQL파서를 구현하였다. 입력값들을 토큰화 시키고 들어올 수 있는 모든 토큰 조합들을 과제 스펙에 맞게 설정해주었다.

가정한 것들

1. query를 입력 도중 가능한 query 입력이 없을 경우 query가 완성되기 전에 syntax error를 출력시켰다. Query가 완성되어도 정상적인 입력으로 처리될 가능성이 없으므로 syntax error를 빠르게 출력하는 것도 상관없다고 생각하였다.
2. 모든 공백은 줄바꿈으로 대체 가능하다고 가정하였다.
Ex)
create
table
~ 와 같은 입력이 정상적으로 수행된다.
3. 각 query가 완성됨과 동시에 query가 수행되고 연속된 query에 대해서 결과의 출력은 모든 query가 완성된 후에 일어난다고 가정하였다.
Ex)
desc
a; show
tables; desc
b;
같은 입력이 들어올 경우 두 번째 줄, 세번째 줄까지 입력되었을 때 desc가 실행되지만
'DESC' requested
'SHOW TABLES' requested
'DESC' requested
의 세 output은 4번째 줄까지 입력 받은 후에 출력된다고 생각했다.

구현 내용

과제 SQL 문법 문서에 맞게 각각의 토큰 조합들을 명시해 주는 작업을 하였다.

가정 3번을 구현하기 위해서 query가 실행될 때 output을 출력해주는게 아니고 저장해두었다가 뱉어내는 방식이 필요했다. 이를 위해 command를 각 query마다 한번 실행되도록 바꾸고 각 query 실행 결과를 qs라는 string에 저장하도록 하였다.

Prompt 라인이 제대로 출력되기 위해서 prompt가 출력되는 조건을 생각해보았다.

semicolon 뒤 줄바꿈문자가 따라서 나올 때까지 입력이 계속 받아져야 한다.

이를 구현하기 위해 token QueryEnd라는 state를 만들고 semicolon이 들어올 때 QueryEnd state로 전환되게 하였다. 줄바꿈문자를 디폴트 상태에선 무시, QueryEnd state에서만 EOQ라는 Token으로 받게 하였다. Semicolon 이후 문자가 다시 들어오면 입력이 계속되는 default 상태로 전환하고 줄바꿈문

자가 들어오면 결과와 prompt 라인을 출력하도록 구현하였다.

느낀 점

실제로 SQL구문 분석을 만들어 보는 과정이 재미있었다. 하지만 모든 경우의 수를 따져 구현하는 부분이 조금 양이 많고 복잡했고 prompt 라인을 정상적으로 출력하기 위해 알고리즘을 바꿀 때 JavaCC에 대한 공부가 새로 필요해서 조금 힘들었다.