느낀 점들

<ALPHABET>과 <LEGAL IDENTIFIER> 순서. -> 한 글자인 경우.

마찬가지로 <DIGIT>와 <INT\_VALUE>의 순서. -> 한 글자.

KEYWORD 토큰들과 <LEGAL IDENTIFIER> 간의 순

한 글자 특수문자들을 < NON\_QUOTE\_SPECIAL\_CHARACTERS> 보다 먼저 정의해야 함.

<SPACE = “ “>와 같이 토큰을 정의하면 SKIP과 충돌이 일어나서, 그냥 “ “로 해줌.

EBNF 문법을 JavaCC 문법에 맞춰 옮겨 적는 작업이 대부분의 시간을 소요했기 때문에 크게 난해한 부분은 없었음. 컴파일러 수업 때 while과 for문을 이용해 파싱했던 것을 생각하면 편한 라이브러리인 것 같다.

columnInTable() 함수에서 tableName()과 columnName()이 정의상 모두 <LEGAL\_IDENTIFIER> 토큰이므로 2번의 LOOKAHEAD는 필수. 따로 정의를 둔 이유가 있을 것 같아 고치지 않음.

Predicate() 함수는 4번의 LOOKAHEAD가 필요한데 지나치게 많음. 따라서 문법을 다음과 같이 수정해 2번의 LOOKAHEAD만 발생하도록 고침.

뼈대 코드의 프린팅 방식이 비효율적인 것 같아 수정.

Option의 DEBUG\_PARSER는 문서를 찾아보니 default가 false가 불필요한 것 같아 삭제.

QA 7번은 틀린 토큰을 입력하고 enter를 누르면 바로 에러 메시지를 띄우는 것으로 함. Enter를 입력한 순간 이전 줄의 입력 값은 바꿀 수가 없으므로 에러가 입력되었다면 유저가 다음 줄에 무엇을 적든 에러. 따라서 에러 처리.

Ctrl+Z 누르면 에러가 무한 생성. <EOF> 처리 해줌.

Eclipse 가 불편해 JetBrain으로 환경 설정을 진행했으나 JetBrain의 Extension은 .jj파일로 .java파일 생성하는 기능을 지원하지 않아 Eclipse에서 진행. Git 사용은 불편해 CLI 사용.