

## 1. 핵심 모듈 설명

### 1.1. class MyTransformer(lark.Transformer)

`lark.Transformer.transform()`는 `lark parser`의 `parse` 결과로 나온 `tree`를 `leaf`부터 하나씩 읽으며 `token`의 `type`과 동일한 이름의 메소드를 호출해줌.

`lark.Transformer`를 상속받는 `MyTransformer`를 만들어 `transform()`을 이용함.

### 1.2. get\_valid\_query(query, e)

`query`가 잘못되어 (예약어를 식별자로 사용, 문법에 맞지 않는 입력 등) 에러가 발생하는 경우 `lark.exceptions.UnexpectedInput`이 발생함. 이 에러에는 `pos_in_stream`이라는 에러가 발생한 위치를 알 수 있는 변수가 있음.

이를 이용하여 `query`에서 에러가 발생한 위치 이전의 부분만 가져오고 거기서 완성된 `query`만 다시 정리하여 `return`함

### 1.3. 나머지 모듈들은 어렵지 않기 때문에 코드 내 주석 참고.

## 2. 알고리즘 설명

### 2.1. sql\_parser 생성 (line 123-124)

`grammar.lark`를 이용해 `sql_parser`를 생성

### 2.2. 사용자로부터 input (line 128)

`get_query()`함

### 2.3. parse 및 결과에 따른 적절한 처리 (line 132-133)

입력 받은 `query`를 `parse()`한 후 `transform()`함.

### 2.4. 에러 핸들링 (line 135-154)

`get_valid_query()`를 사용해 에러가 발생하기 직전 `query`까지 다시 구해 파싱하고 그 뒤 `Syntax error` 출력.

## 3. 기타 사항

### 3.1. 식별자로 예약어를 사용하는 경우에 대한 별도 로직

이러한 로직은 불필요하다. 왜냐하면 식별자로 예약어가 들어오는 경우 `sql_parser`가 그것을 `IDENTIFIER`로 인식하지 못하고 해당 예약어로 인식하여 문법에 맞지 않는 `input`이라고 생각하기 때문이다.

해당 로직은 line 35-46, line 105-115에 주석으로 남아있다.