

Project 1-1: SQL Parser

Due: 2023/4/4 (Tues), 11:59 PM

0. Project Overview

이번 프로젝트의 목표는 간단한 SQL 파서(parser)를 구현하는 것이다.

SQL 파서란, 파싱(parsing)을 수행해서 SQL 구문의 구조를 이해하고 분석하는 도구를 뜻한다.

파싱(parsing)이란 입력 받은 구문을 분석하는 과정을 뜻한다. 입력 받은 구문이 정의된 문법 구조를 따르는 오류 없는 완전한 구문인지 분석하는 작업을 뜻한다. 파싱을 거쳐야 구문이 정확하게 해석되고 올바르게 명령을 실행할 수 있게 된다.

본 프로젝트에서 구현할 파서는 입력 받은 구문이 오류 없는 완전한 구문이면 아래와 같은 결과값을 출력한다.

```
DB_2023-12345> select ID from student;  
DB_2023-12345> 'SELECT' requested  
DB_2023-12345>
```

파서를 구현하는 과정에서, 입력 받은 구문에 대한 파싱 결과를 다음과 같이 받아볼 수 있다. 파싱이 완료되면 입력 받은 구문이 분석되어 Tree 형태로 저장되어 있고 이를 출력해서 확인할 수 있다.

파싱 결과물 출력 예시)

Code:

```
output = sql_parser.parse('select ID from student;')  
print('-----Parsed result-----')  
print(output.pretty())
```

Output:

```
-----Parsed result-----  
command  
  query_list  
    query  
      select_query  
        select  
          select_list  
            selected_column  
              None  
              column_name ID  
              None  
              None  
            table_expression  
              from_clause  
                from  
                  table_reference_list  
                    referred_table  
                      table_name      student
```

None
None
None

본 프로젝트는 Python 언어로 진행하며, Python용 파싱 라이브러리 Lark를 사용하도록 한다. Lark는 EBNF (Extended Backus-Naur Form) 형식으로 정의된 grammar를 기준으로 입력 받은 구문을 파싱해주는 API를 지원하는 라이브러리이다.

Lark가 제공하는 다양한 API를 활용하여 정의된 grammar를 기준으로 하는 파서를 생성할 수 있고, 생성된 파서를 사용해서 입력된 구문에 대한 파싱 결과를 tree 형태로 저장 및 출력할 수도 있다. 7장에 제공된 reference를 통해 다양한 API 명세와 구현 예시를 참고할 수 있다.

따라서, 본 프로젝트에서는 grammar가 일부 제공된 grammar.lark 파일을 수정, 완성해야 하고, run.py에 Lark API를 활용하여 SQL 파서를 구현한다. 그리고 run.py를 실행하여 SQL 구문을 입력했을 때 파서가 이를 올바르게 해석하고 올바른 종류의 구문이 요청되었다는 결과를 출력할 수 있어야 한다.

프로젝트 1-1에서 구현한 SQL 파서는 추후 프로젝트 1-2 및 1-3에서도 계속해서 사용될 예정이니 본 문서에서 제공되는 요구사항, 구현해야 하는 구문 Definition, 개발 환경 조건, 제출 및 성적 기준 안내 등을 숙지하여 프로젝트 1-1을 진행하면 된다.

1. 요구 사항

- 본 문서에 명시되어 있는 모든 SQL 구문들을 파싱할 수 있어야 한다.
- SQL 구문이 여러 줄로 이루어진 경우에도 파싱할 수 있어야 한다.
 - 구문의 끝에는 항상 세미콜론(';')이 존재한다고 가정한다.
 - 구문의 중간에 개행 문자('\r', '\n' 등)가 입력되어도 구문 입력이 종료되지 않아야 한다.
- 프롬프트는 "DB_학번>"의 형태이다.
 - DB_2023-12345> create table ...
- 2장에 명시된 스펙에 맞게 입력 값을 받고, 입력 값에 따른 결과를 출력한다.
 - DB_2023-12345> 'CREATE TABLE' requested
 - DB_2023-12345> 'DROP TABLE' requested
 - ...
- 에러가 발생한 경우, 에러를 출력한다.
 - DB_2023-12345> Syntax error

2. SQL

본 장에서는 각 구문에 대한 Definition과 이를 따르는 입출력 예제들이 제공된다. Definition과 예제를

참고하여 각 구문이 파싱될 수 있도록 구현해야 한다.

([] 안의 내용은 생략될 수 있는 부분이다)

2.1 CREATE TABLE

- Definition

```
create table table_name (  
    column_name data_type [not null],  
    ...  
    [primary key(column_name1, column_name2, ...),]  
    [foreign key(column_name3) references table_name1(column_name4),]  
    [foreign key(column_name5) references table_name2(column_name6)]  
    ...  
)
```

※ *table_name*, *column_name*은 알파벳 및 '_'으로만 구성

- 예시

```
DB_2023-12345> create table account (  
    account_number int not null,  
    branch_name char(15),  
    primary key(account_number)  
);  
DB_2023-12345> 'CREATE TABLE' requested  
DB_2023-12345>
```

2.2 DROP TABLE

- Definition

```
drop table table_name;
```

- 예시

```
DB_2023-12345> drop table account;  
DB_2023-12345> 'DROP TABLE' requested  
DB_2023-12345>
```

2.3 EXPLAIN / DESCRIBE / DESC

- Definition

```
explain table_name;  
describe table_name;  
desc table_name;
```

- EXPLAIN 예시

```
DB_2023-12345> explain account;  
DB_2023-12345> 'EXPLAIN' requested  
DB_2023-12345>
```

- DESCRIBE 예시

```
DB_2023-12345> describe account;  
DB_2023-12345> 'DESCRIBE' requested  
DB_2023-12345>
```

- DESC 예시

```
DB_2023-12345> desc account;  
DB_2023-12345> 'DESC' requested  
DB_2023-12345>
```

2.4 INSERT

- Definition

```
insert into table_name [(col_name1, col_name2, ... )] values(value1, value2, ...);
```

- 예시

```
DB_2023-12345> insert into account values(9732, 'Perryridge');  
DB_2023-12345> 'INSERT' requested  
DB_2023-12345>
```

2.5 DELETE

- Definition

```
delete from table_name [where clause];
```

- 예시

```
DB_2023-12345> delete from account where branch_name = 'Perryridge';  
DB_2023-12345> 'DELETE' requested  
DB_2023-12345>
```

2.6 SELECT

- Definition

```
select [table_name.]column_name [as name], ...  
from table_name [as name], ...  
[where clause];
```

- 예시1

```
DB_2023-12345> select * from account;  
DB_2023-12345> 'SELECT' requested  
DB_2023-12345>
```

- 예시2

```
DB_2023-12345> select customer_name, borrower.loan_number, amount  
from borrower, loan  
where borrower.loan_number = loan.loan_number and branch_name = 'Perryridge';  
DB_2023-12345> 'SELECT' requested  
DB_2023-12345>
```

2.7 SHOW TABLES

- Definition

```
show tables;
```

- SHOW 예시

```
DB_2023-12345> show tables;  
DB_2023-12345> 'SHOW TABLES' requested  
DB_2023-12345>
```

2.8 UPDATE

- Definition

```
update table_name set column_name = comparable_value [where clause];
```

- 예시

```
DB_2023-12345> update student set id = 5 where name="susan";  
DB_2023-12345> 'UPDATE' requested  
DB_2023-12345>
```

2.9 EXIT

- Definition

```
exit;
```

- 예시

```
DB_2023-12345> exit;  
database@server:~$
```

3. Query Sequence

여러 개의 query가 세미콜론(';')으로 구분되어 입력 값으로 들어온 경우, query는 순차적으로 처리되어야 한다.

```
query1;query2;query3;...;queryN;
```

※ 만약 K번째 쿼리인 queryK에 에러가 있을 경우, query1부터 queryK-1까지는 처리해야 함.

- Query Sequence 예시

```
DB_2023-12345> insert into account values(9732, 'Perryridge'); show tables; insert  
into account; desc account;  
DB_2023-12345> 'INSERT' requested  
DB_2023-12345> 'SHOW TABLES' requested  
DB_2023-12345> Syntax error  
DB_2023-12345>
```

4. 개발 환경

- Python 3.6 ~ 3.9
- Lark API

5. 제출

1. grammar.lark 파일
 2. run.py 파일 (반드시 주석이 있어야 함)
 - run.py와 grammar.lark가 있는 directory에서 해당 python 프로그램을 실행 시 바로 SQL parser가 동작해야 함
 3. 리포트 (아래 내용 중 해당되는 내용을 작성)
 - 핵심 모듈과 알고리즘에 대한 설명
 - 구현한 내용에 대한 간략한 설명
 - (제시된 요구사항 중 구현하지 못한 부분이 있다면) 구현하지 못한 내용
 - 프로젝트를 하면서 느낀 점 및 기타사항
 - 리포트는 1장 recommend, 2장 이내로 작성
- 위의 3가지 파일을 압축하여 ETL에 제출
 - 파일명: PRJ1-1_학번.zip (e.g., PRJ1-1_2023-12345.zip)

6. 성적 관련 사항

- 제출 기한 이후 24시간 이내 제출시 10% 감점
- 제출 기한 이후 24시간 이후 48시간 이내 제출시 20% 감점
- 제출 기한 48시간 이후에는 점수 없음
- 부정 행위는 0점 처리
 - ◆ 다른 사람의 코드를 참조하는 행위
 - ◆ 이전에 수강한 사람의 코드를 참조하는 행위
 - ◆ 제출한 소스코드에 대해 표절 방지 프로그램을 돌릴 예정
- 본 문서에 명시되어 있는 출력 양식을 지키지 않았거나 주석이 없는 경우 감점

7. References

- Lark
 - <https://lark-parser.readthedocs.io/en/latest/index.html>