

22-f Database Project 1-1: SQL Parser

조경·지역시스템공학부

2017-19651 이혜민

1. 핵심 모듈과 알고리즘

SQL parser 를 구현하는 핵심 모듈은 lark 이다. Lark 모듈을 활용해 'grammar.lark' 파일을 작성하고, 이를 runs.py 파일에서 불러와 터미널로 받은 input 을 파싱한다. 'run.py' 파일에는 입력받은 input 을 처리하는 로직(세미콜론으로 끝나는 경우 처리, 여러 줄로 인풋을 받는 경우 처리, lark 로 정상적으로 처리된 쿼리의 종류를 프린트하기 등)을 구현하였으며, 'grammar.lark' 파일에는 실제 쿼리들의 문법을 작성하여 올바른 쿼리인지 확인할 수 있도록 하였다.

2. 구현 내용

2-1. 'grammar.lark' 파일에서 이미 정의되어 있는 create table query, select query 를 제외한 drop table, desc, insert, delete, show tables, update query 를 구현하였다.

drop, desc, show tables 의 경우 별도의 column 이나 value list, 그리고 where 절이 없기 때문에 비교적 간단하게 작성하였다.

```
// DROP TABLE, DESC, SHOW TABLES
drop_table_query : DROP TABLE table_name
desc_query : DESC table_name
show_tables_query : SHOW TABLES
```

Insert, delete, update tables 의 경우 최대한 이미 정의되어 있는 where_clause 와 comparable_value, column_name_list 를 활용하고 UPDATE, SET 등의 키워드를 새로 추가하여 구현하였다. value_list 의 경우 column_name_list 와 유사하게 작성하였다. 이번 과제에서는 not null 인 column 에 실제 NULL 값이 들어가는지 유효성을 체크하지 않기 때문에 항상 NULL 로 insert 할 경우도 고려하였다.

```
// INSERT
insert_query : INSERT INTO table_name [column_name_list] VALUES value_list
value_list : LP available_value ("," available_value)* RP
available_value : comparable_value | NULL

// DELETE
delete_query : DELETE FROM table_name [where_clause]

// UPDATE TABLES
update_query : UPDATE table_name SET column_name "=" comparable_value [where_clause]
```

2-2. 'run.py' 파일의 경우 프롬프트를 표시하고, input 을 받은 뒤 이 input 이 한 번에 세미콜론으로 끝난 경우와 끝나지 않은 경우를 나누어 처리하였다.

한 line 이 세미콜론으로 끝난 경우에는 세미콜론 기준으로 input 을 split 하여 split 된 query 를 각각 sql_parser 로 처리한다.

한 line 이 세미콜론으로 끝나지 않은 경우에는 다음 line 을 받아서 이전 query list 에 붙인 후 다시 세미콜론으로 끝나는지 확인한다.

```
while True:
    with open('grammar.lark') as file:
        sql_parser = Lark(file.read(), start="command", lexer="basic")
    query_str = input(PROMPT_NAME)
    query_list = query_str.split(';')

    # query finished with semicolon in one line
    if (query_list[-1] == ""):
        # pop last blank by split
        query_list.pop()

        while len(query_list) > 0:
            query = query_list.pop(0)
            query_with_semicolon = query + ";"
            try:
                result = sql_parser.parse(query_with_semicolon)
                parse_result(query_with_semicolon)
            except SystemExit:
                sys.exit(0)
            except:
                print(PROMPT_NAME + ERROR_STRING)
                break;

    # query didn't finish with semicolon in one line
    # multi-line query/queries
    else:
        while query_list[-1] != "":
            query_list[-1] += " "
            new_query_str = input()
            new_query_list = new_query_str.split(';')
            # continue unfinished last query
            if len(new_query_list) > 0:
                query_list[-1] += new_query_list.pop(0)
            query_list = query_list + new_query_list

        # pop last blank by split
        query_list.pop()

        while len(query_list) > 0:
            query_with_semicolon = query_list.pop(0) + ";"
            try:
                result = sql_parser.parse(query_with_semicolon)
                parse_result(query_with_semicolon)
            except SystemExit:
                sys.exit(0)
            except:
                print(PROMPT_NAME + ERROR_STRING)
                break;
```

그리고 각각의 query 가 정상적으로 처리되면, 'parser_result()' 메서드를 통해 어떤 종류의 query 가 request 되었는지 프린트한다. 이 때, 세미콜론 뒤 공백을 무시하기 위해 lstrip()처리를 한 후 query 의 앞글자를 비교하여 처리한다.

```
def parse_result(query):
    query = query.lstrip()
    if query[:12] == 'create table':
        print(PROMPT_NAME + "'CREATE TABLE' requested")
    elif query[:10] == 'drop table':
        print(PROMPT_NAME + "'DROP TABLE' requested")
    elif query[:4] == 'desc':
        print(PROMPT_NAME + "'DESC' requested")
    elif query[:6] == 'insert':
        print(PROMPT_NAME + "'INSERT' requested")
    elif query[:6] == 'delete':
        print(PROMPT_NAME + "'DELETE' requested")
    elif query[:6] == 'select':
        print(PROMPT_NAME + "'SELECT' requested")
    elif query[:4] == 'show':
        print(PROMPT_NAME + "'SHOW TABLES' requested")
    elif query[:6] == 'update':
        print(PROMPT_NAME + "'UPDATE' requested")
    elif query[:4] == 'exit':
        raise SystemExit
```

3. 가정한 것

- 유효성을 체크하지 않으니 모든 column 에 들어가는 value 가 comparable_value (INT | STR | DATE) 뿐만 아니라 NULL 도 가능하다고 가정하였다.
- IDENTIFIER 에 해당하는 " 없는 문자열은 column 의 value 로 들어오지 않는다고 가정하였다.