Project 1-1: SQL Parser

Due: 2019/4/1 (Mon), 11:59 PM

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

SQL 파서란 SQL 구문의 구조를 이해하고 분석하는 도구를 뜻한다.

프로젝트 1-1에서 구현한 SQL 파서는 추후 프로젝트 1-2 및 1-3에서도 계속해서 사용될 예정이다.

1. 요구 사항

- 2장에 나열된 모든 SQL 구문들을 파싱할 수 있어야 한다.
- SQL 구문이 여러 줄로 이루어진 경우에도 파싱할 수 있어야 한다.
  - 쿼리의 끝에는 항상 세미콜론(';')이 존재한다고 가정한다.
  - 쿼리의 중간에 개행 문자('₩r', '₩n' 등)가 나타나더라도 쿼리를 종료하지 않아야 한다.
- 프롬프트는 "DB_학번> "의 형태이다.
  - `DB_2019-12345> create table …`
- 안내된 스펙에 맞게 입력 값을 받고, 알맞은 결과를 출력한다.
  - `'CREATE TABLE' requested`
  - `'DROP TABLE' requested`
  - …
- 에러가 발생한 경우, 에러를 출력한다.
  - `Syntax error`

다음은 이해를 돕기 위한 예제들이다. 자세한 문법은 Grammar Definition 부분에 명시되어 있다.

([ ] 안의 내용은 생략될 수 있음)

2. SQL

2.1 CREATE TABLE

```
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*의 경우,

- 알파벳 및 '_'으로만 구성됨
- 자세한 사항은 Grammar Definition 참조

- CREATE TABLE 예시

```
DB_2019-12345> create table account (
      account_number int not null,
      branch_name char(15),
      primary key(account_number)
);
DB_2019-12345> 'CREATE TABLE' requested
DB_2019-12345>
```

2.2 DROP TABLE

```
drop table table_name;
```

- DROP TABLE 예시

```
DB_2019-12345> drop table account;
DB_2019-12345> 'DROP TABLE' requested
DB_2019-12345>
```

2.3 DESC

```
desc table_name;
```

- DESC 예시

```
DB_2019-12345> desc account;
DB_2019-12345> 'DESC' requested
DB_2019-12345>
```

2.4 INSERT

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

- INSERT 예시

```
DB_2019-12345> insert into account values(9732, 'Perryridge');
DB_2019-12345> 'INSERT' requested
DB_2019-12345>
```

2.5 DELETE

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

- DELETE 예시

```
DB_2019-12345> delete from account where branch_name = 'Perryridge';
DB_2019-12345> 'DELETE' requested
DB_2019-12345>
```

2.6 SELECT

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

- SELECT 예시1

```
DB_2019-12345> select * from account;
DB_2019-12345> 'SELECT' requested
DB_2019-12345>
```

- SELECT 예시2

```
DB_2019-12345> select customer_name, borrower.loan_number, amount
from borrower, loan
where borrower.loan_number = loan.loan_number and branch_name = 'Perryridge';
DB_2019-12345> 'SELECT' requested
DB_2019-12345>
```

2.7 SHOW TABLES

```
show tables;
```

- SHOW TABLES 예시

```
DB_2019-12345> show tables;
DB_2019-12345> 'SHOW TABLES' requested
DB_2019-12345>
```

3. Query Sequence

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

```
query1;query2;query3;…;queryN;
```

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

- Query Sequence 예시

```
DB_2015-12345> insert into account values(9732, 'Perryridge'); show tables; insert
into account; desc account;
DB_2019-12345> 'INSERT' requested
DB_2019-12345> 'SHOW TABLES' requested
DB_2019-12345> Syntax error
DB_2019-12345>
```

4. 개발 환경

- Java
- Eclipse
- JavaCC (for Java) API

5. 제출

1. Executable jar 파일
   - 파일명: PRJ1-1_학번.jar (e.g., PRJ1-1_2019-12345.jar)
2. 소스 파일 (반드시 주석이 있어야 함)
3. 리포트 (아래와 내용을 포함해야 함)
   - 핵심 모듈과 알고리즘에 대한 설명
   - 구현한 내용에 대한 간략한 설명
   - (제시된 요구사항 중 구현하지 못한 부분이 있다면) 구현하지 못한 내용
   - 가정한 것들
   - 컴파일과 실행 방법
   - 프로젝트를 하면서 느낀 점

- 위의 3가지 파일을 압축하여 lecture@europa.snu.ac.kr로 제출
  - 파일명: PRJ1-1_학번.zip (e.g., PRJ1-1_2019-12345.zip)

- 메일 제목: [DB Project1-1] your student ID, your name (e.g., [DB Project 1-1] 2019-12345 홍길동)

- 리포트의 Hard copy는 제출 기한 다음 날까지 301동 420호로 제출

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 않을 시 감점

7. References

- **JavaCC**
  - https://javacc.org/
  - http://www.engr.mun.ca/~theo/JavaCC-Tutorial/javacc-tutorial.pdf
  - http://www.javaworld.com/javaworld/jw-12-2000/jw-1229-cooltools.html
- **Regular expression**
  - http://gnosis.cx/publish/programming/regular_expressions.html
  - http://www.regular-expressions.info/reference.html