

Project 1-3 DML Parser

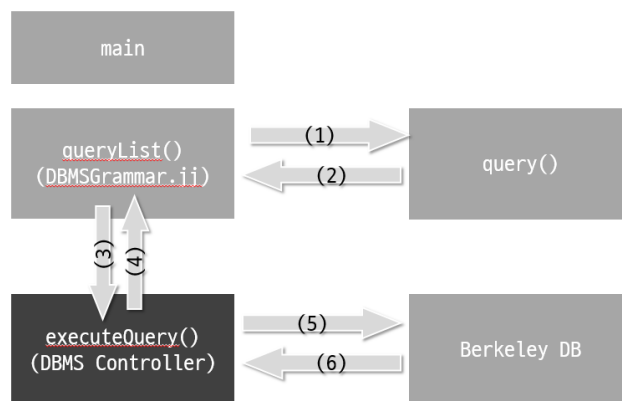
2015-10033 김다윤

■ 프로젝트 개요

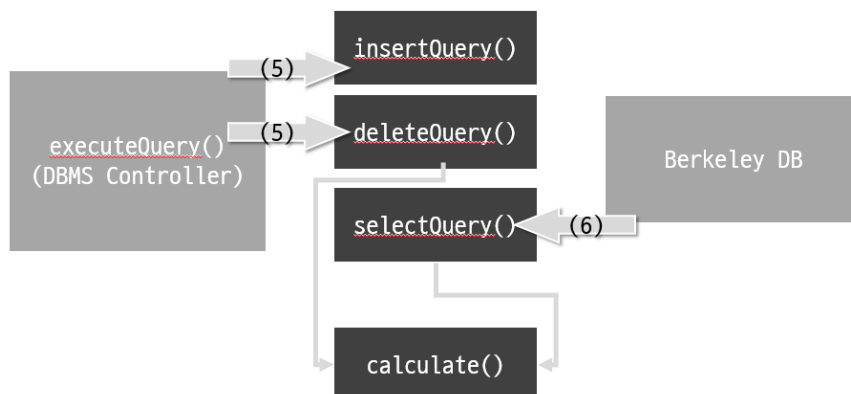
지금까지 SQL parser를 만들고 DDL manipulate 기능을 구현하였다. 여기에 실질적인 데이터 레코드를 저장할 수 있도록 DML manipulate 기능을 추가한다. 이를 위해 대표적인 쿼리 구문인 insert, delete, select를 구현한다. 정의한 스펙에 알맞게 각 경우에 대한 여러 케이스를 처리하여 정상적으로 작동하면서 데이터베이스가 의도한대로 삽입, 삭제를 할 수 있도록 구현한다. 또한 쿼리에 알맞은 데이터를 정확히 가져오도록 구현한다.

■ 핵심 모듈과 알고리즘

지난 프로젝트까지 만든 구조는 다음과 같다.

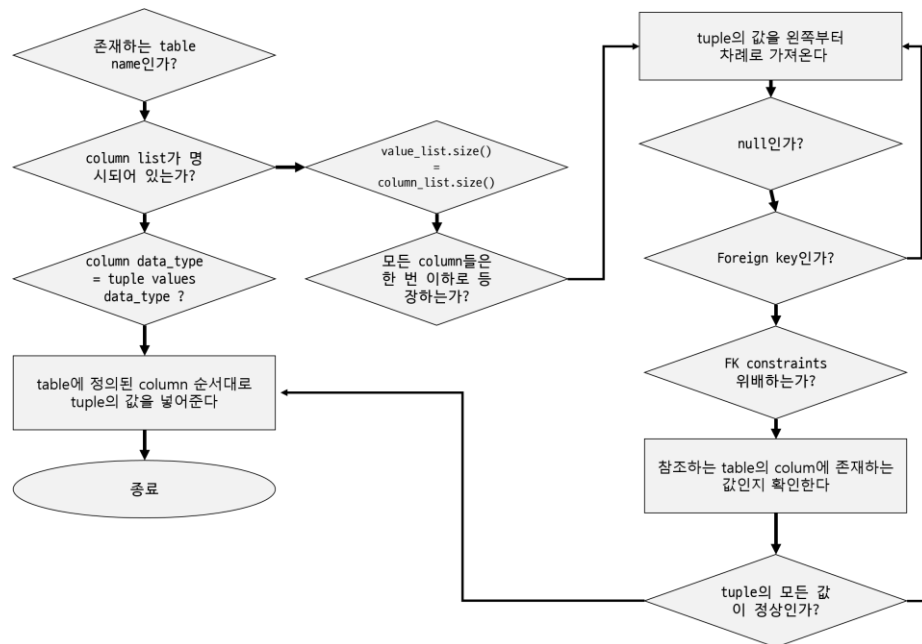


프로그램의 작동을 위한 전체적인 모듈을 만들었으며, 이번에는 이 구조에서 DB에 값을 저장 또는 삭제하는 (5)와 데이터를 가져오는 (6)의 기능을 구현한다.



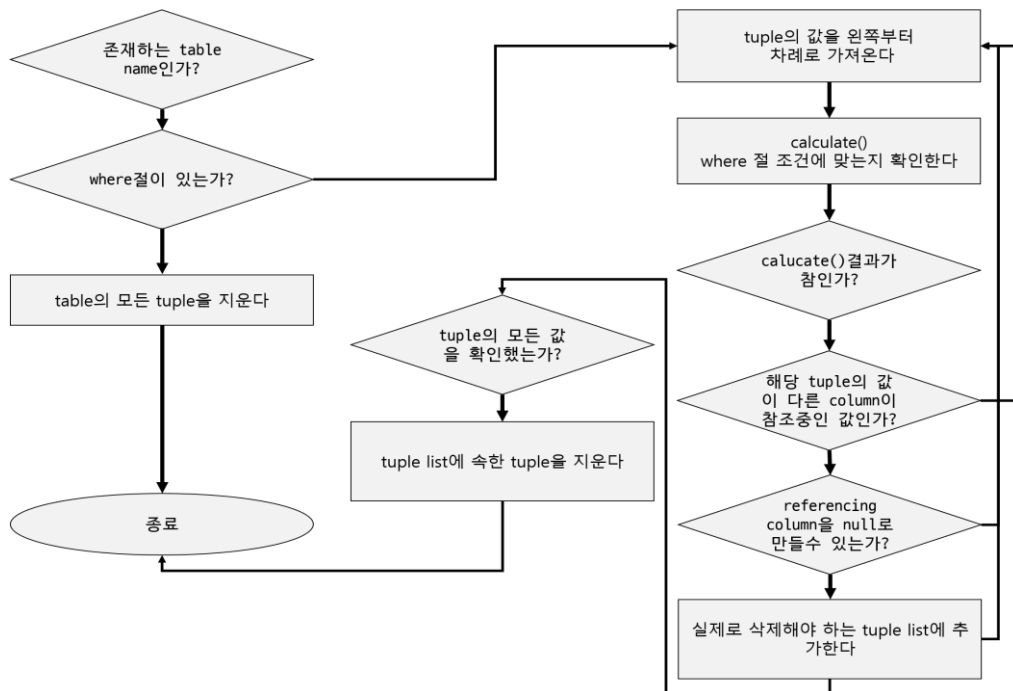
◎ tuple insert를 위한 알고리즘

입력 : insert into <table_name> [<column_list>] values (<values_list>)



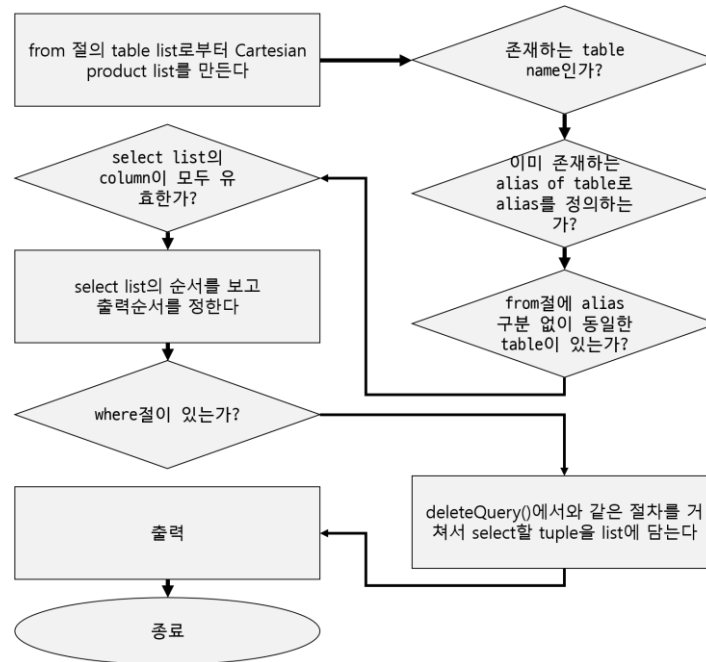
◎ delete를 위한 알고리즘

입력 : delete from <table_name> [where <where_clause>]



◎ tuple select를 위한 알고리즘

입력 : select <select_list> from <table_name_list> [where <where_clause>]



■ 구현 내용

◎메소드

insertQuery(), deleteQuery(), selectQuery() : 앞의 알고리즘으로 설명을 대신한다.

makeCartesian() : select query의 from <table_list>에서 table이 둘 이상 입력될 경우 이들을 join 하여 cartesian product를 만들기 위한 재귀함수이다. table이 하나이면 해당 table을 반환하고, 그렇지 않다면 재귀적으로 만들어진 joined table에 해당 table을 join한 후 만들어진 table을 반환한다.

calculate() : where절의 자료구조인 ExpTree를 계산하기 위한 메소드이다. 재귀적으로 호출되며 operator의 종류에 따라 경우를 나누어 처리한다.

◎자료구조

ExpTree : where절은 그 특성에 따라 predicate를 node로 가지는 이진트리로 구성한다.

predicate : where절의 node의 type으로, 해당 술어에 속하는 operator와 operands를 담고 있다.

refTabCol : operand의 type으로, 해당 operand가 table column을 가리키는 comparable value인지와 column이라면 테이블 이름과 컬럼 이름을 가진다.

■ 가정한 것들

- ◎ select 절의 컬럼의 테이블이 존재하지 않으면 selectColumnResolveError를 출력한다
- ◎ 명시된 스펙이 기대하는 동작과 일치하는 선에서 MYSQL의 작동을 참조하였다.

■ 컴파일과 실행 방법

1. 실행

```
$cd [jar file location]
```

```
$java -jar [filename].jar
```

2. 종료

```
$exit;
```

■ 프로젝트를 하면서 느낀 점

쿼리문 처리에 대한 다양한 케이스들이 존재 해서 어떤 경우에는 유효한 쿼리로 판단해서 추론을 하는 반면 유효하지 않은 경우에는 구체적으로 무엇이 틀렸는지 메시지를 출력하는 것이 굉장히 까다롭다고 느꼈다. 또한 한 record만 바뀌어도 해당 테이블을 완전히 지웠다가 다시 쓰게끔 구현을 해서 효율적이지 못한 구조가 된 것 같아 아쉬웠다.