

DB 프로젝트 1-3 보고서

2015-11923 최한결

본 프로젝트에서는 1-2에서 만든 DBMS parser에 DML 명령어를 수행할 수 있도록 기능을 추가하였다.

개발 환경

OS: MacOS catalina
IDE: Eclipse 4.15.0
Java version: 1.7

핵심 모듈과 알고리즘에 대한 설명

각 데이터는 각 테이블의 필드로 존재하고 리스트로 관리된다. Where 절에서 각 record가 where 절을 만족시키는지 여부를 확인하기 위해 condition package에 여러 클래스들을 만들었다. 우선 Boolean 클래스는 UNKNOWN을 고려해 and or not의 비트연산이 가능하도록 구현했고 Operand 클래스는 value 값이면 value를 반환하고 reference 하는 값일 경우 각 where절의 column이 어느 번째 위치에 있는지 기억했다가 record가 들어올 때마다 그 위치의 값을 반환한다. 이 외에 BooleanExpressionList, BooleanFactor, Predicate, ComparisonPredicate, NullPredicate는 모두 BooleanExpression 인터페이스를 상속받는 클래스이다. BooleanExpression 인터페이스는 checkBooleanValue라는 method를 갖고 제일 상단에 있는 BooleanExpressionList에서 checkBooleanValue를 호출하면 이차원 배열로 저장된 내부의 BooleanExpression이 평가되어 and연산으로 묶여 계산된 후 or연산으로 묶여 계산되어 Boolean값이 반환된다. 이를 통해 select, delete에서 각 record별로 where절을 만족하는지 평가한다.

구현 내용

Insert 시에는 table을 참조해 각 column Name의 index를 Array로 저장, 각 index에 해당되는 value를 삽입 후, 제약 조건 확인, 삽입한다. 참조 무결성을 확인하기 위해 Column에 저장된 reference table을 불러와 foreign key 들을 확인한다. Delete에서는 우선 where절을 통해 records를 filtering 한 후 참조 무결성을 유지하기 위해 해당 table을 참조하고 있는 모든 테이블의 records를 비교한다. 이 때 삭제하려는 record를 참조하고 있는 record의 column이 전부 nullable일 경우 참조하고 있는 record의 foreign key 값을 null로 변경하고 삭제한다. 전부

nullable이 아닐 경우엔 삭제를 취소한다.

가정한 것들

1. Select시에 출력될 record가 없을 때는 “Empty Set” 메시지를 출력한다. (mysql과 동일)
2. “select 뒤에 나오는 컬럼 이름을 해석하는 데에 문제가 있다면, SelectColumnResolveError(#colName)에 해당하는 메시지를 출력 “ 에서 select 뒤에 테이블명이 들어가 있을 경우 colName 을 tableName.colName 으로 출력한다고 가정했다.

Ex) DB_2015-11923> select a.name from account;

>Selection has failed: fail to resolve 'a.name'

3. Select절에서 from 절 뒤에 같은 table이 rename없이 두번 이상 나오면 SelectColumnResolveError(#colName) 에러 출력

Ex) DB_2015-11923> select * from student, student;

>Selection has failed: fail to resolve 'student.~'

DB_2015-11923> select * from student as a, student;

> 정상 출력

4. Error 가 중복될 때 Error 의 표시 순서는 고려하지 않아도 된다고 가정했다.

컴파일과 실행 방법

PRJ1-3_2015-11923.jar가 들어있는 폴더에 db라는 하위 폴더를 생성한 후 java -jar PRJ1-3_2015-11923.jar 명령어를 통해 실행할 수 있다.

느낀 점

데이터 무결성을 유지하기 위해 엄청 많은 조건 확인이 필요했고 힘든 작업이었다. 하지만 위 조건 확인을 안했을 때는 더욱 알 수 없는 버그들이 나올 수 있다는 것을 확인했고 DBMS에 필수적인 부분이라고 생각했다. 또한 각 기능(문장 하나하나)에 대해서도 모듈화를 제대로 해놓아야 한다고 생각했다.