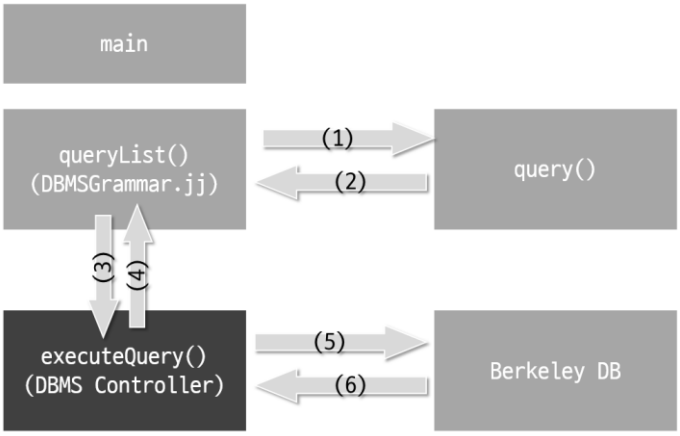
Project 1-2 Implementing DDL

2015-10033 김다윤

▣ 프로젝트 개요

지난 프로젝트 때 구현한 SQL parser에 스키마 관리 기능을 추가한다. 즉, 테이블의 생성, 삭제 및 테이블 목록과, 각 테이블의 구성 칼럼들을 보여준다. 이러한 기능은 SQL의 Data definition language의 역할로, `create table, drop table, desc, show tables` 구문으로 기능을 수행한다. 테이블 스키마는 Berkeley DB를 이용하여 프로그램이 종료되어도 없어지지 않도록 한다.

▣ 핵심 모듈과 알고리즘



지난 프로젝트에서 완성한 부분은 (1)과 (2)이다. queryList()함수는 입력문을 syntax 체크하고 파싱하는 query()로부터 파싱구문을 받는다.

파싱된 입력문을 DB에 저장하기 전 database constraints이 맞는지 확인하고 알맞게 가공하는 과정이 필요하다. queryList()는 파싱된 입력을 QueryTemplate 타입으로 바꾸어 executeQuery()로 전달한다. (3) 각 쿼리에 알맞은 전처리를 수행한 후 스키마를 파일에 저장 또는 삭제하거나(5) 파일로부터 원하는 스키마를 가져온다(6)

▣ 구현 내용

1. 자료구조

- QueryTemplate : 쿼리 종류에 관계없이 executeQuery()함수로 파싱된 쿼리를 보내기 위해 설정한 자료형이다. int q가 있어서 q의 값에 따라 쿼리 종류를 구분할 수 있다.

- Table : berkeley db에 저장하는 형태이다. 테이블이름과 각 테이블이 가진 칼럼의 리스트로 이루어진다. 테이블이름은 berkeley db에서 key의 역할을, 칼럼 리스트는 value로 기능한다. 따라서 각 테이블은 최종적으로 Table형태로 변환된다.

- colAttribute : 칼럼의 이름, 속성을 지니는 자료형이다. 속성에는 데이터타입, null여부, PK여부, FK여부, (FK라면) 참조하는 테이블명과 컬럼이름이 있다.

- fkConstraint : reference constraint를 처리한 후 얻게 되는 foreign key constraints이다.

- refConstraint : 파싱할 때 reference constraints를 임시적으로 저장하기 위한 자료형이다.

2. 동작제어

- DBMSController : 데이터베이스에 접근하는 함수들이 정의되어있다. 대표적으로 다음과 같은 함수들이 있다.

• create() : 테이블을 생성해서 저장한다. 이름이 같거나, 잘못된 참조를 하는 등, create 쿼리에서 일어날 수 있는 예외사항을 처리한다. 처리 후 문제가 없으면 Table 형태로 바꾸어 db file에 저장한다. 참고로 berkeley db에 byte로 변환해야하는데, arraylist of string인 컬럼을 변환하기 위해 각 칼럼마다 ';'라는 delimeter를 추가하여 변환한다.

• load(table_name): 테이블이름을 보고 해당 테이블을 꺼내온다. key들 간의 중복을 허용하지 않기 때문에 없거나, 있다면 하나만 존재한다.

• drop(table_name): 데이터베이스를 읽는 커서가 해당 테이블이름의 key를 가리키면 이를 삭제한다.

- createTableQuery : create table [table name] 쿼리가 처리해야 할 것이 가장 많기 때문에 별도의 클래스로 선언하여 에러처리 등의 전처리를 한 후 DMBSController에서 입출력을 한다.

3. 에러 처리

- ddlException : DDL구문을 처리하면서 발생하는 에러를 처리하는 클래스

-DBMSMessage : 에러 시 출력할 메시지

▣ 가정한 것들

DDL 구문을 처리할 때는 primary key와 foreign key에 관한 많은 제약이 있다. 문서에 명시되지 않았거나 명시되어있더라도 애매한 사항들을 다음과 같이 가정하였다.

- 하나의 컬럼이 여러 칼럼들을 참조할 때 : MultipleReference 에러 발생시킨다

- primary key(..., C_i, ..., C_j, ...) 에서 Ci와 Cj가 같은 칼럼일 때: duplicateColumnDefError 에러를 발생시킨다.

- 어떤 테이블을 삭제할 때, 해당 테이블을 참조하고 있는 칼럼들의 null가능 여부에 관계없이 삭제가 불가능 하여 DropReferencedTableError 에러로 처리한다.

- foreign key가 자기자신의 테이블을 참조할 경우 ReferenceColumnItself에러를 발생한다.

▣ 컴파일과 실행 방법

* jar file이 위치한 폴더에 'db' 폴더가 생성되어 있어야 한다.

1. 실행

$cd [jar file location]

$java –jar [filename].jar

2. 종료

$exit;

▣ 프로젝트를 하면서 느낀 점

파서를 구현하는 것보다 시간이 오래 걸렸고 까다롭게 느껴졌다. berkeley DB라는 낯선 툴을 이해해야 했기 때문이다. 또한 reference constraints를 지키기 위해서 테이블 생성시에 처리해 줄 조건이 정말 많았는데, 모든 경우들을 고려해서 에러 처리를 하는 것이 어려웠다.