이태민
2017-15892

Project 1-3 Report

핵심 모듈과 알고리즘에 대한 설명 :

이번 프로젝트는, DML 기능들을 DBMS 에 추가하는 작업을 진행하였다. Insert, Delete, Select, Update 쿼리들을 처리하는 작업을 진행하면서 개인적으로 매우 혼란스러웠다. 데이터를 데이터베이스에 저장하는 것을 직관적이였지만, 이를 쿼리의 condition 에 따라 알맞게 작업들(delete,select,update)을 수행하는 로직을 직접 생각해서 구현해야 되었기 때문에, 여러 방법으로 로직을 구현할 수 있었다. 하지만, 1-2 에서 데이터베이스에 어떤 형식으로 schema 와 constraint 들을 저장 했는가에 따라 1-3을 구현해야 했기 때문에 1-2 코드를 뜯어 고치지 않는 이상, 구현 방법은 나름 제한적 이였다. 구현한 방식은 이러하였다

Database directory 안에, 1-2 에서 구현한 create query 를 통해 오른쪽과 같은 key-value pair 들이 저장된다. Referenced 된 테이블의 이름, schema 는 attribute 과 constraint 를 key-value 로 묶어서 저장, primary key 에 대한 정보 등을 저장하였다. Attribute 들이 테이블 안에서 몇번째에 위치했는지도 DML 에서 필요하기 때문에, 해당 정보도 '위치 - column' 의 key-value pair 로 저장하였다.

```
(b'2', b'name')
(b'test', b'referenced')
(b'1', b'age')
(b'age', b'int')
(b'name', b'char(10)')
(b'primary key', b'age, name')
```

Insert

Insert 에 대한 쿼리를 처리하는데 내가 채택한 방법은 이러했다

1. Table 이 존재하는지 확인한다
2. 쿼리 자체를 먼저 확인한다
    1. 쿼리 안에 들어온 column 과 값의 개 수와 타입을 비교하였다
3. Table 의 schema 와 쿼리를 비교한다
    1. 쿼리에서 제시하는 value 들의 수와 schema 안에 있는 attribute 의 수 비교
    2. Type 비교
4. Null 값 처리에 대한 함수도 구현하였다
    1. 테이블의 definition 에서, 해당 attribute 가 null 값이 가능한지 여부를 판단
5. 여기까지 error 가 없다면, record 를 database에 추가
    1. 추가하는 방법은 key 에 primary key 값들을 저장
        1. 예) primary key(age, name) 이면 key = '30, Jason"
    2. Data 에는 primary key 가 아닌 값들을 저장
        1. 예) other attributes : lastname, height 이면 data = "Kim, 180"

Delete, Select, Update 에 대한 쿼리 처리 알고리즘은 여러모로 비슷했다. 마지막 데이터를 처리하는 부분만 다르고, database 에서 record by record 방식으로 query 의 condition 들을 record 와 비교 후, condition 이 satisfied 된다면 알맞게 처리하는 방식으로 진행하였다. 아쉽게도 시간이 없어서 update 와 select 문은 구현하지 못하였다. Delete 의 알고리즘은 이렇게 진행하였다

Delete

1. Table 이 존재하는지 확인한다
2. Where clause 가 없는 쿼리라면, 모든 record 들을 지운다
    (a) Cursor 를 활용하여, 'foreignkey', 'primary key', 'references' 등, record 가 아닌 key-value pair 들을 빼고 모두 delete 해준다.

   (b) globCounter 라는 변수를 increment 한다 (delete 가 일어날때마다). 이 변수는 총 row deleted 의 개수를 의미한다.
3. Query Parsing 작업을 해준다
   (a) Query 에서 명시된 column, operator, operand 를 구별하여 parsing 한 후, 프로그램 내부에서 이를 쓸 수 있게 나눈다.
   (b) Column[ ], operator [ ], operand [ ] 에서 각각 index 에 위치한 값들은 related 한 값들이다(from one condition).
      (a) 예) delete from table where age = 3 and name != 'John' 이면,
         (a) Column [age, name], operator [=, !=], operand [3, John] 과 같이 저장된다
4. 2에서 parse 된 값들을 type checking 을 한다
   (a) Column 이 존재하는지, operand 와 column 의 constraint 가 일치하는지 등 에 대한 여부는 이 단계(함수) 에서 모두 진행함
5. deleteQuery 함수를 통해 database 에서 해당 테이블 안에 있는 값들을 delete
   (a) 여기서 3번에서 parse 한 값들을 갖고 와서 적절하게 condition 들을 확인하는 작업을 진행하였다.
   (b) 어려웠던 부분은, where 절에서 여러 condition 들이 주어졌을때, 이를 처리하는 로직을 구현하는데 오래 걸렸다. Schema 에서 age,name 순서로 저장된 값들과, query 에서 where name = 'john' and age = 20 로 scheme 의 반대 순서로 들어온 query 에 대해 처리를 해주어야 됐기 때문에 for loop 을 통해 query 에 들어온 column 들이 table 에서 몇 번째 column 인지 뽑아내고, 그에 맞게 처리하였다.

구현하지 못한 내용 :

Select query 와 update query

Project 1-2 의 점수가 나왔을때, 너무 저조한 성적이 나와서 1-3을 구현하면서 많은 생각들이 들었다. 1-3 에서는 1-2 에서 구현한 create table 의 형식에 따라서 구현해야 되는 부분들이 너무나도 많았으며, 저조한 점수 때문에, 1-2 에서 구현한 내용들이 전부 틀렸다고 생각이 들었다. 이 때문에, 1-3 을 구현해도, 1-2 에서 잘못 구현한 모듈들과 방법 때문에, 1-3 또한 낮은 점수를 받을 것이라는 생각이 들었다. 또한, 1-2 에서 어느 부분이 틀렸는지 알 방법이 없었기 때문에, 1-2 를 계속 고쳐보려고 노력했던 것 같다. 결론적으로 claim session 때 체점을 다시 해주시겠다고 조교님께서 말씀해주셔서 너무 감사했지만, 1-3을 끝내기에는 시간이 부족하였다. 시간 관리를 못 한 내 잘못이다

하지만, select 와 update query 또한, delete query 를 처리하는 방법과 동일하게, 위의 알고리즘에서 1-4 step 을 통해 query 의 condition 에 맞는 record 들을 선별하고, select 는 이를 출력, update 는 record 값을 수정해주는 작업을 해주면 될 것 이라고 생각한다. 시간적 여유만 있다면 구현할 수 있을 것 같다.

프로젝트를 하면서 느낀 점:

프로젝트 1은, 코드 양을 봤을때는 다른 과목들에 비해 비교적으로 적은 편에 속한다는 생각이 들었지만, 난이도는 매우 높은 편에 들었다고 생각한다. 여러 corner case 때문에 고려해야될 부분들이 상당히 많았고, 대부분의 case 들을 스스로 찾아야 했던 것 같다. 추가적으로, 구현 방법에 대해 명확한 guideline 이 없다 보니, 학생이 원하는데로 자율적으로 로직을 짜는 것은 좋았으나, 이를 1-2 같은 경우에 잘못 짜게 되면, 1-3에도 어느정도 영향이 간다는 점이 아쉬웠다. Lark module 이나 berkeleydb 에 대한 지식을 가르쳐주지 않은 것도 조금 아쉬웠다. 마지막으로는, 개인적으로 데이터베이스 수업에서 기업에서 다루는 DBMS 를 사용하는 방법이나, query 들을 embedd 해서 프로그램을 만든다던지, 현업에서 사용하는 기술들을 배우고 싶었으며, 학문적이고 이론적인 DBMS 구현이라는 프로젝트가 과연 얼마나 학생에게 도움이 될지 의문이 들었다. 물론 DBMS 를 구현함으로서 많은 것을 배웠지만, 실용적이지 않다는 느낌을 받았다. 데이터베이스 과제가 아닌, 프로그래밍 과제라는 느낌이 더 강했다.