

# Database

## Project 1-2: Implementing DDL

컴퓨터공학부  
2017-18538  
황선영

### 1. 핵심 모듈과 알고리즘

Project 1-2는 run.py, grammar.lack, exception\_handling.py로 이루어져있다. run.py는 create, show, desc, drop query를 받아 exception handling을 하면서 query 내용을 manipulate하여 DB에 해당 내용을 반영한다.

입력된 query는 DB에 저장되기 전 error check를 거치는데, 문서에 명시된 error의 우선순위를 정해 차례대로 exception handling을 한 후 이상이 없으면 DB에 저장하는 알고리즘을 사용했다.

### 2. 구현한 내용에 대한 간략한 설명

입력된 query는 transformer의 여러 def 함수를 거쳐 <key, value> 형태의 dictionary로 변환된다. Key는 attribute의 이름이거나 primary key, foreign key를 나타낸다. Value는 유형에 따른 detail을 dictionary 형태로 담고있다. Attribute일 경우 내부에 attribute 이름, data type, null 여부, key 여부의 정보를 담고 있다. Primary key의 경우 primary key의 list를 가지고 있다. Foreign key의 경우 foreign key인 attribute 이름 list와 referenced table 이름과 그 table의 attribute 이름 list를 담고 있다. 이러한 dictionary를 이용하여 data에 접근하면서 error를 handling하였다.

### 3. 구현하지 못한 내용

Error message를 출력할 때 대문자로 입력된 query의 경우에도 소문자로 출력하는 issue가 있다. 그 외의 경우에는 잘 구현하였다.

### 4. 가정한 것들

Tree 형태로 주어진 query를 어떻게 변형할까 고민하다가 schema를 <key, value> 형태로 저장하라고 명시되어 있어서 dictionary를 이용하여 query의 data에 효과적으로 접근하고 DB에 저장이 용이하도록 하였다.

### 5. 프로젝트를 하면서 느낀 점

Tree를 manipulate하는 과정에서 어려움을 느꼈다. 하지만 적절한 함수를 쓰며 dictionary로 변형할 수 있었고 DB에 schema를 저장할 때 dictionary 형태가 굉장히 효율적이라는 사실을 깨달았다. Exception handling의 우선순위를 결정하는 과정에서 query 요소들의 중요도를 정할 수 있었다. 이번 실습을 통해 DDL이 어떤 식으로 동작하는지 알 수 있어서 흥미로운 project였다.