실전학습 COP (Collaboration of Practices)
songpanet@naver.com

# GraphQL API Example – part 3

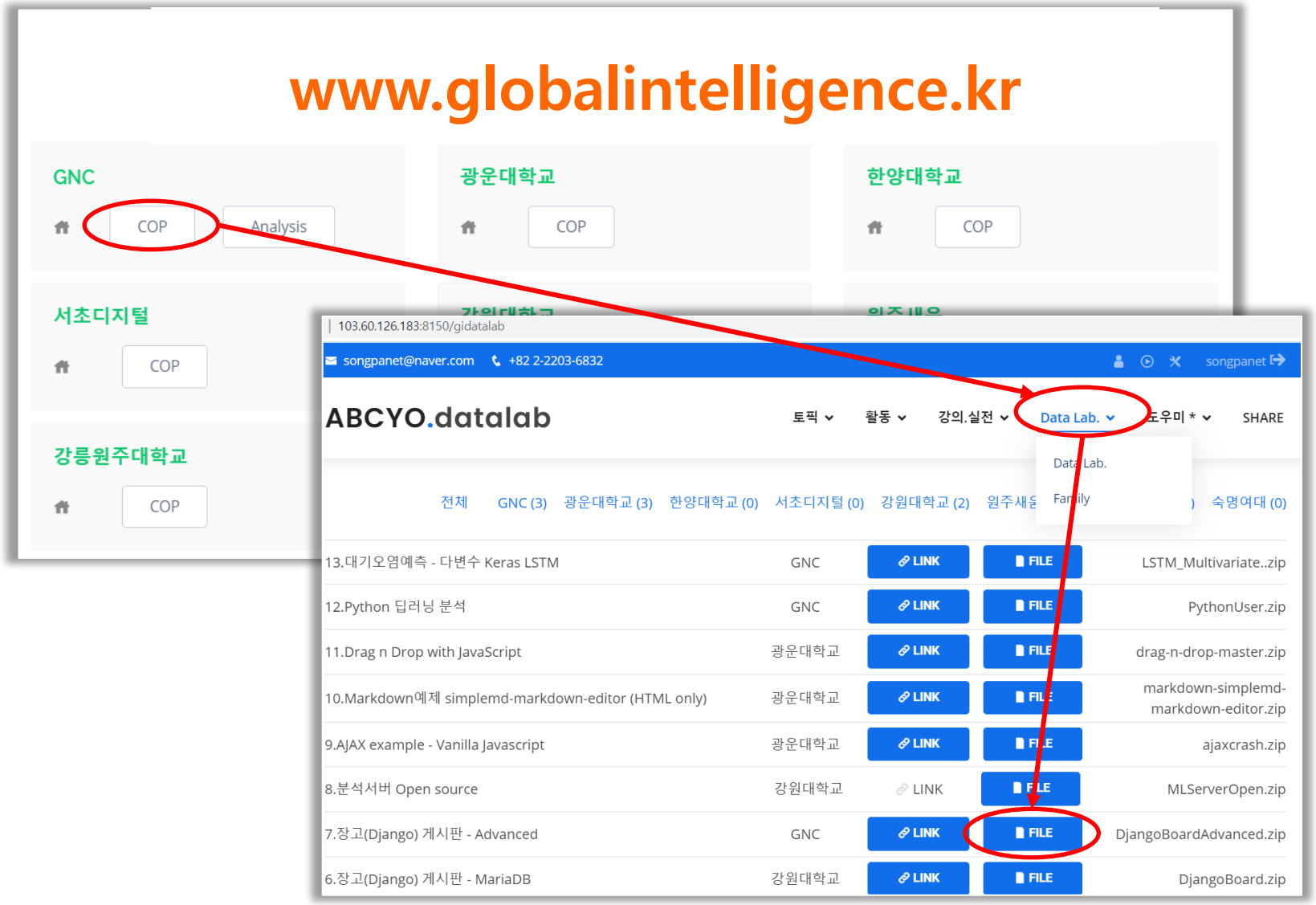
## Django graphene – 조회,등록,수정,삭제 graphql

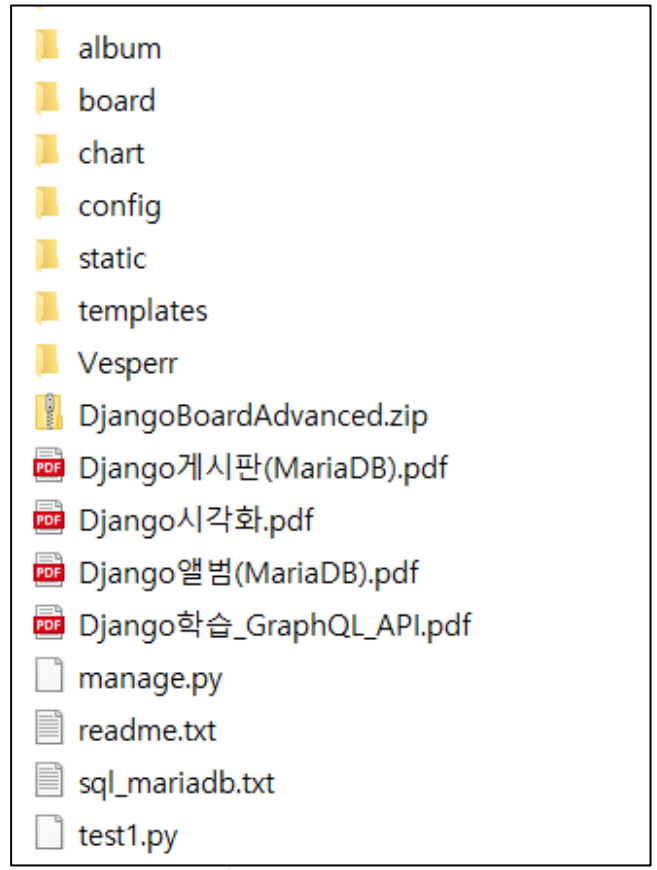
COP실전학습용 – www.globalintelligence.kr

실전학습 참여 문의 : songpanet@naver.com

# Source download

## COP ➔ Data Lab. ➔ 7번의 FILE을 다운로드 받는다.




- ✓ 설명
- ✓ DB source (MariaDB)

# Review – Part 1

## Data (게시판 테이블)

| # | 이름 | 데이터 유형 | 길이/설정 | 부호 ... | NU |
|---|---|---|---|---|---|
| 1 | b_no | INT | 11 | | |
| 2 | b_title | VARCHAR | 255 | | |
| 3 | b_note | VARCHAR | 4096 | | |
| 4 | parent_no | INT | 11 | | |
| 5 | b_writer | VARCHAR | 50 | | |
| 6 | b_count | INT | 11 | | |
| 7 | b_date | TIMESTAMP | | | |

ORM

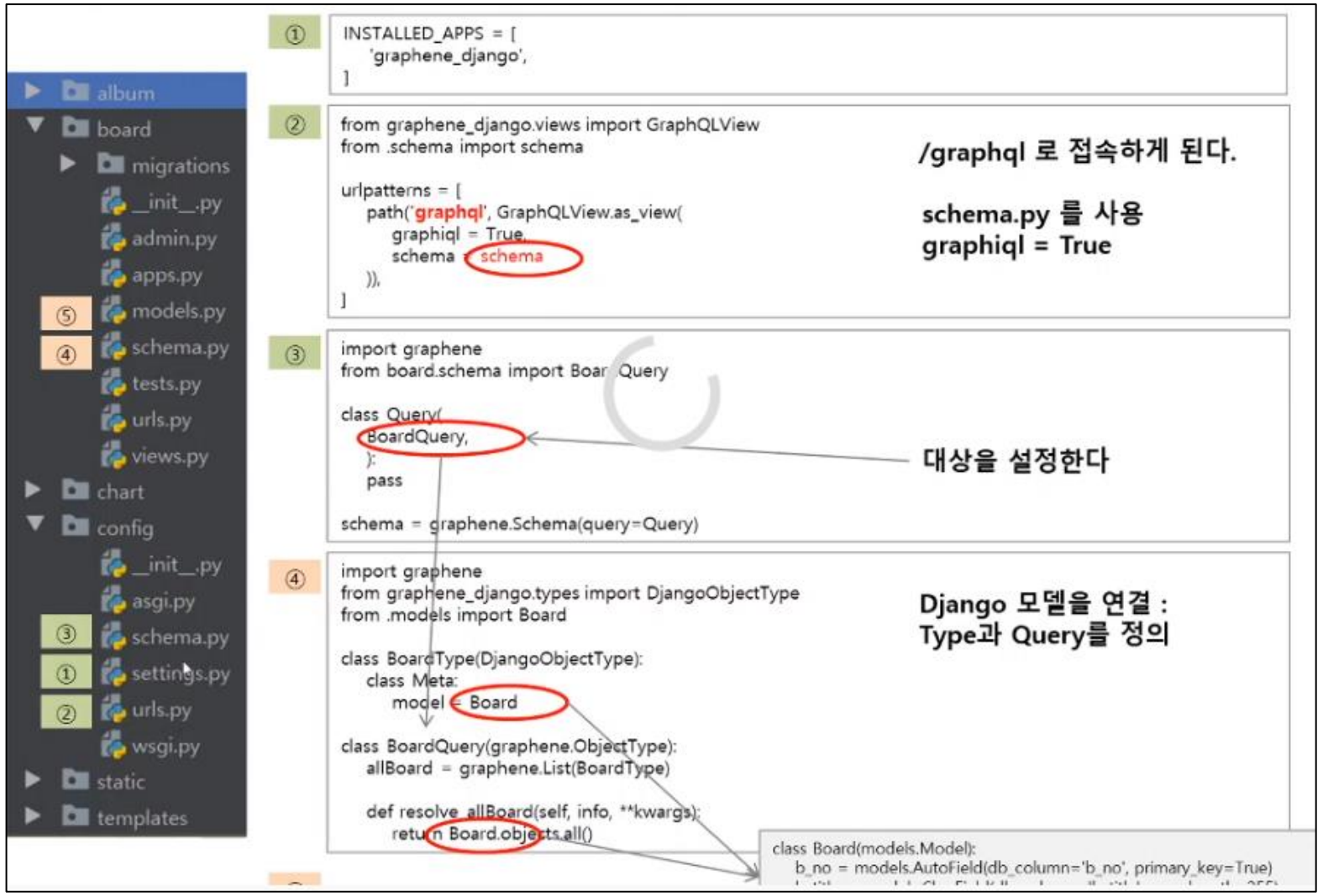
```python
from django.db import models

class Board(models.Model):
    b_no = models.AutoField(db_column='b_no', primary_key=True)
    b_title = models.CharField(db_column='b_title', max_length=255)
    b_note = models.TextField(db_column='b_note', )
    b_writer = models.CharField(db_column='b_writer', max_length=50)
    parent_no = models.IntegerField(db_column='parent_no', default=0)
    b_count = models.IntegerField(db_column='b_count', default=0)
    b_date = models.DateTimeField(db_column='b_date', )

    class Meta:
        managed = False
        db_table = 'board'

    def __str__(self):
        return "제목 : " + self.b_title + ", 작성자 : " + self.b_writer
```

## Django source 구성



# Review – Part 2

## Query

```
class Board(graphene.ObjectType):
    b_no = graphene.ID()
    b_title = graphene.String()
    b_writer = graphene.String()
    b_note = graphene.String()
    d_count = graphene.Int()
    b_date = graphene.DateTime()

class Query(graphene.ObjectType):
    boards = graphene.List(Board)

    def resolve_boards(self, info):
        return [
            Board(b_title='Title입니다.1', b_writer='고길동', b_note='내용입니다
            Board(b_title='Title입니다.2', b_writer='홍길동', b_note='내용입니다
            Board(b_title='Title입니다.3', b_writer='타이손', b_note='내용입니다
        ]
                                    니다. 1', b_count=0, b_date=datetime.now()),
                                    니다. 2', b_count=0, b_date=datetime.now()),
                                    니다. 3', b_count=0, b_date=datetime.now())
schema = graphene.Schema(query=Query)

result = schema.execute(
    '''
    {
        boards {
            bTitle
        }
    }
    '''
)
```

## Mutation

```
class CreateBoard(graphene.Mutation):
    class Arguments:
        btitle = graphene.String()

    board = graphene.Field(Board)

    def mutate(self, info, btitle):
        if info.context.get ('is_vip'):
            btitle = btitle.upper()

        board = Board(btitle=btitle)
        return CreateBoard(board=board)

class Mutations(graphene.ObjectType):
    create_user = CreateBoard.Field()

schema = graphene.Schema(mutation=Mutations)

result = schema.execute(
    '''
    mutation createUser ($btitle: String) {
        createUser(btitle: $btitle) {
            board {
                btitle
            }
        }
    }
    ''',
    variable_values={'btitle':'제목45w3535322... Time is Gold'},
    context={'is_vip':True}
)
print(result)
```

# GraphQL API와 게시판

Board

Django 웹서버

| # | 이름 | 데이터 유형 |
|---|---|---|
| 1 | b_no | INT |
| 2 | b_title | VARCHAR |
| 3 | b_note | VARCHAR |
| 4 | parent_no | INT |
| 5 | b_writer | VARCHAR |
| 6 | b_count | INT |
| 7 | b_date | TIMESTAMP |
| 8 | usage_flag | VARCHAR |

게시판 목록 조회하기 - query
게시판 한 건 조회하기 - query
입력 - mutate
수정 - mutate
삭제 - mutate

# GraphQL – Query, Mutation, Subscription

## Query

## Mutations