

Django 14. M:N 데이터베이스 관계



- ❖ 학습해야 할 내용
 - ✓ Django

1. media 파일 경로

사용자가 업로드한 파일이 저장되는 위치를 Django 프로젝트 폴더(crud) 내부의 /uploaded_files로 지정하고자 한다. 이 때, settings.py에 작성해야 하는 설정 2가지를 작성하시오.

2. DB True or False

각 문항을 읽고 맞으면 T, 틀리면 F를 작성하시오.

- 1) RDBMS를 조작하기 위해서 SQL문을 사용한다.
- 2) SQL에서 명령어는 반드시 대문자로 작성해야 동작한다.
- 3) 일반적인 SQL문에서는 세미콜론(;)까지를 하나의 명령어로 간주한다.
- 4) SQLite에서 .tables, .headers on과 같은 dot(.)로 시작하는 명령어는 SQL문이 아니다.
- 5) 하나의 데이터베이스 안에는 반드시 한 개의 테이블만 존재해야 한다.

3. on_delete

게시글과 댓글의 관계에서 댓글이 존재하는 게시글은 삭제할 수 없도록 __a__에 들어갈 코드를 작성하시오. 그리고 이러한 설정이 되어있는 상황에서 Article 객체를 삭제하려고 할 때 발생하는 오류를 작성하시오.

```
class Article(models.Model):
    title = models.CharField(max_length=100)
    content = models.TextField()

class Comment(models.Model):
    content = models.TextField()
    article = models.ForeignKey(Article, on_delete=models.__a__)
```

Django 14. M:N 데이터베이스 관계



4. like

Article 모델과 User 모델을 M:N 관계로 설정하여 '좋아요' 기능을 구현하려고 한다. __a__와 __b__에 들어갈 내용을 작성하시오. 추가적으로 아래의 상황에서 __b__를 반드시 작성 해야 하는 이유를 함께 작성하시오.

```
from django.db import models
from django.conf import settings

class Article(models.Model):
    content = models.TextField()
    user = models.ForeignKey(settings.AUTH_USER_MODEL, on_delete=models.CASCADE)
    like_users = models.__(a)__(settings.AUTH_USER_MODEL, __b__)='like_articles')
```

5. follow

follow 기능을 구현하기 위해 accounts app의 models.py에 아래와 같은 모델을 작성하였다. Migration 작업 이후에 Database에 만들어지는 중개 테이블의 이름을 작성하고 이 테이블의 id를 제외한 필드 이름을 각각 작성하시오.

```
from django.db import models
from django.contrib.auth.models import AbstractUser

class User(AbstractUser):
    followings = models.ManyToManyField('self', symmetrical=False, related_name='followers')
```