



Django 12. 1:N 데이터베이스 관계

❖ 학습해야 할 내용

- ✓ Django Model Relation

1. M:N True or False

각 문항을 읽고 맞으면 T, 틀리면 F를 작성하고 틀렸다면 그 이유도 함께 작성하시오.

- 1) Django에서 1:N 관계는 ForeignKeyField를 사용하고 M:N 관계는 ManyToManyField를 사용한다.
- 2) ManyToManyField를 설정하고 만들어지는 테이블 이름은 앱이름_클래스이름_지정한 필드이름의 형태로 만들어진다.
- 3) ManyToManyField의 첫번째 인자는 모델, 두번째 인자는 related_name이 들어가는 데 두 가지 모두 필수적으로 들어가야 한다.

2. like

아래 빈 칸에 들어갈 코드를 각각 작성하시오.

```
class Article(models.Model):
    ...
    user = models.ForeignKey(settings.AUTH_USER_MODEL, on_delete=models.CASCADE)
    like_users = models.ManyToManyField(settings.AUTH_USER_MODEL, related_name=
'like_articles')
```

```
<!-- articles/index.html -->

{% for article in articles %}
    ...
    <p>{{ article.title }}</p>
    <a href="{% url 'articles:like' article.pk %}">
        {% if __ (a) __ in __ (b) __ %}
            <i class="fas fa-heart fa-lg" style="color:crimson;"></i>
        {% else %}
            <i class="fas fa-heart fa-lg" style="color:black;"></i>
        {% endif %}
    </a>
    <span>{{ __ (b) __ |length }}명이 이 글을 좋아합니다.</span>
{% endfor %}
```

Django 12. 1:N 데이터베이스 관계



3. follow

모델 정보가 다음과 같을 때 빈칸에 들어갈 코드를 각각 작성하시오.

```
app_name = 'accounts'
urlpatterns = [
    ...,
    path('<int:user_pk>/follow/', views.follow, name='follow'),
]
```

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

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

```
# accounts/views.py
from django.contrib.auth import get_user_model

User = get_user_model()

@require_POST
def follow(request, __(a)__):
    person = get_object_or_404(get_user_model(), pk=__(a)__)
    user = request.user

    if person.__(b)__.__(c)__(pk=user.pk).exists():
        person.__(b)__.__(d)__(user)
    else:
        person.__(b)__.__(e)__(user)
    return redirect('accounts:profile', person.username)
```

4. User AttributeError

아래와 같은 에러 메시지가 발생하는 이유와 이를 해결하기 위한 방법을 작성하시오.

AttributeError at /accounts/signup/

Manager isn't available; 'auth.User' has been swapped for 'accounts.User'

```
Request Method: POST
Request URL: http://127.0.0.1:8000/accounts/signup/
Django Version: 2.1.15
Exception Type: AttributeError
Exception Value: Manager isn't available; 'auth.User' has been swapped for 'accounts.User'
```



Django 12. 1:N 데이터베이스 관계

5. related_name

아래의 경우 related_name을 필수적으로 설정해야 한다. 그 이유를 설명하시오.

```
class Article(models.Model):
    ...
    user = models.ForeignKey(settings.AUTH_USER_MODEL, on_delete=models.CASCADE)
    like_users = models.ManyToManyField(settings.AUTH_USER_MODEL, related_name=
'like_articles')
```

6. follow logic in template

Person에는 view에서 넘어온 유저 정보가 담겨 있고 모델 정보가 아래와 같을 때, 빈칸에 들어갈 코드를 각각 작성하시오.

```
app_name = 'accounts'
urlpatterns = [
    ...,
    path('<int:user_pk>/follow/', views.follow, name='follow'),
]
```

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

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

```
<!-- accounts/profile.html -->
<h3>작성자 : {{ person.username }}</h3>
<h4>
    팔로잉 : {{ __ (a) __ |length }}
    팔로워 : {{ __ (b) __ |length }}
</h4>
<p>
    {% if __ (c) __ != __ (d) __ %}
    {% if __ (c) __ in __ (b) __ %}
    <a href="{% url 'accounts:follow' __ (e) __ %}">Unfollow</a>
    {% else %}
    <a href="{% url 'accounts:follow' __ (e) __ %}">Follow</a>
    {% endif %}
    {% endif %}
</p>
```