

# DB, MODEL, ADMIN

멋사 7기 신해철

# 지난 시간

```
from django.contrib import admin
from django.urls import path
import blog.views

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', blog.views.layout, name='layout'),
]
```

Urls.py

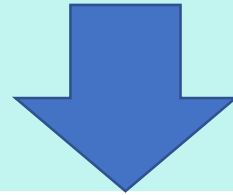
```
<h1>Hello World</h1>
```

Home.html

```
from django.shortcuts import render

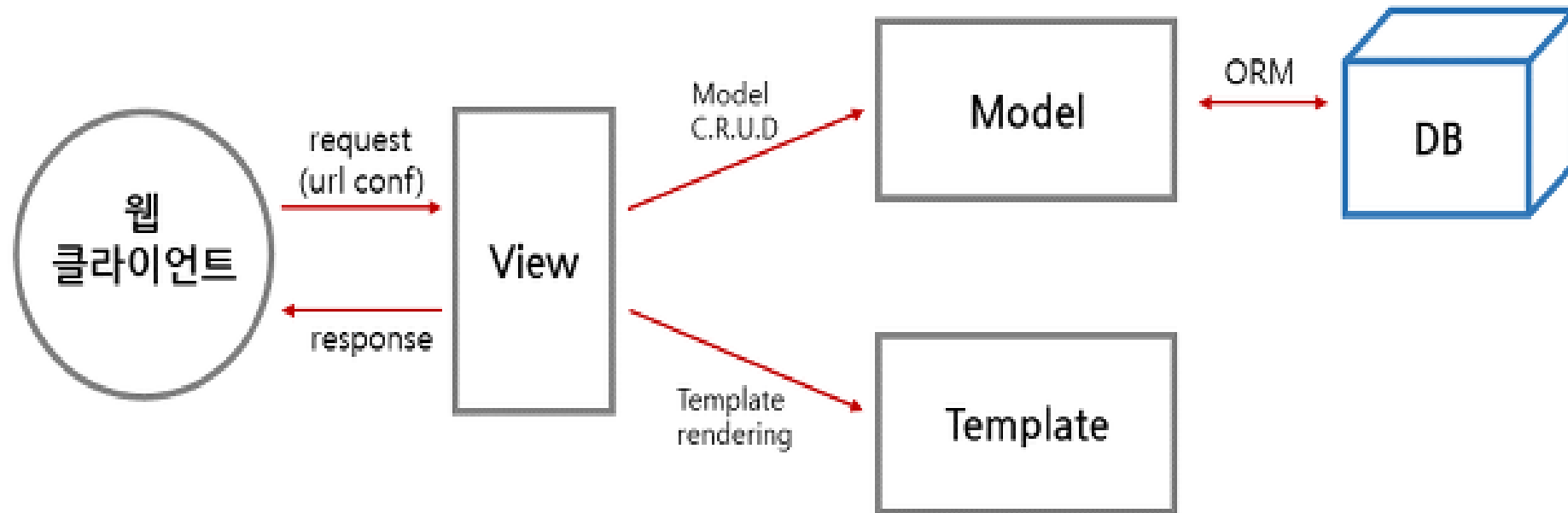
# Create your views here.
def layout(request):
    return render(request, 'blog/layout.html')
```

Views.py



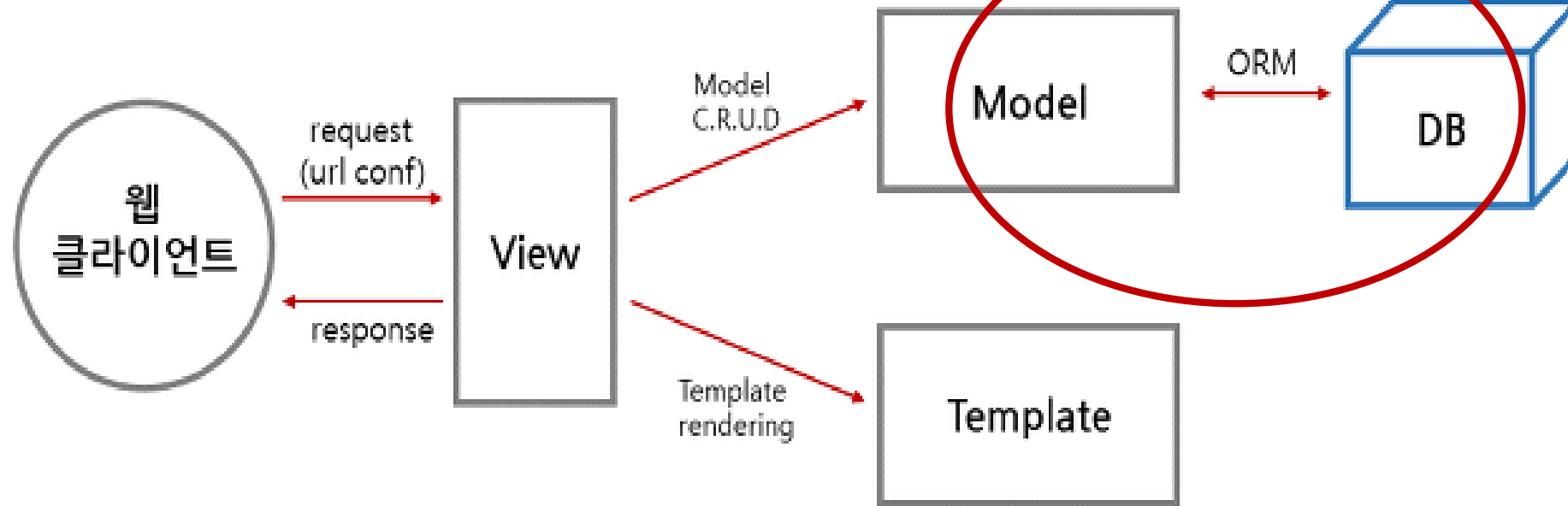
127.0.0.1:8000

# 지난 시간



# 지난 시간

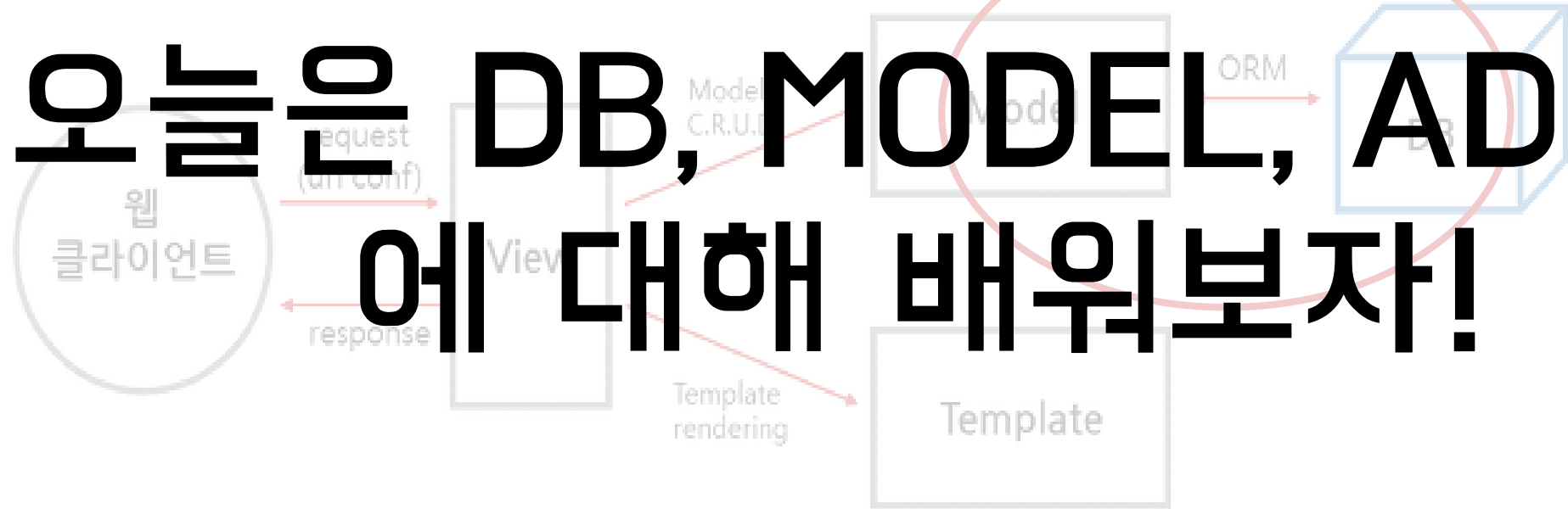
?



지난 시간

?

오늘은 DB, MODEL, ADMIN  
에 대해 배워보자!



# 데이터베이스 (DATABASE)

# DB(DataBase)

## 데이터베이스란?

### 1) 데이터베이스의 개념 및 정의

기존의 파일시스템에서는 중복된 정보를 효율적으로 처리하는 것이 관점 연구 분야였다. 이 중복을 피하여 정보를 일원화(一元化)하여 처리를 효율적으로 수행하기 위해서 서로 관련성을 가지며 중복이 없는 데이터의 집합을 유지하는 것을 데이터베이스(-, database)라고 한다. 그리고 데이터베이스를 관리하는 시스템을 데이터베이스 관리시스템(DBMS: Database Management System)이라 한다. 데이터베이스의 대표적인 구조는 트리구조(tree structure)인 구조적 데이터베이스 외에 관계형 데이터베이스(relational database)와 객체지향 데이터베이스 구조로 대별한다.

→ 여러 사람에 의해 공유되어 사용될 목적으로 통합하여 관리되는 **데이터의 집합**을 말한다

# DB(DataBase)

데이터베이스 종류에는 여러가지가 있지만 우리가 사용하는 장고 내장 DB는 SQLite이다. SQLite는 가장 흔히 사용되고 있는

## 관계형 데이터베이스!

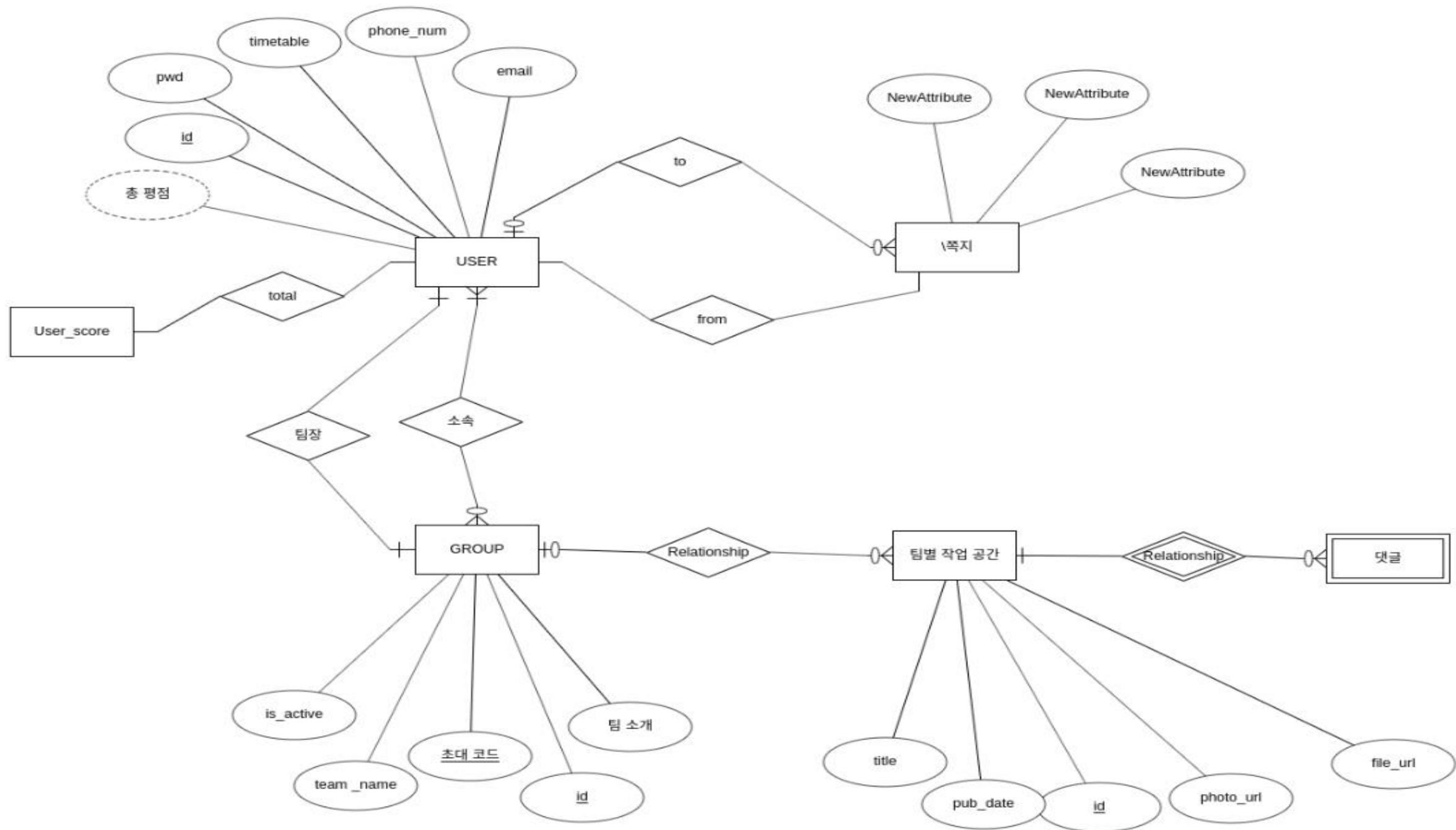


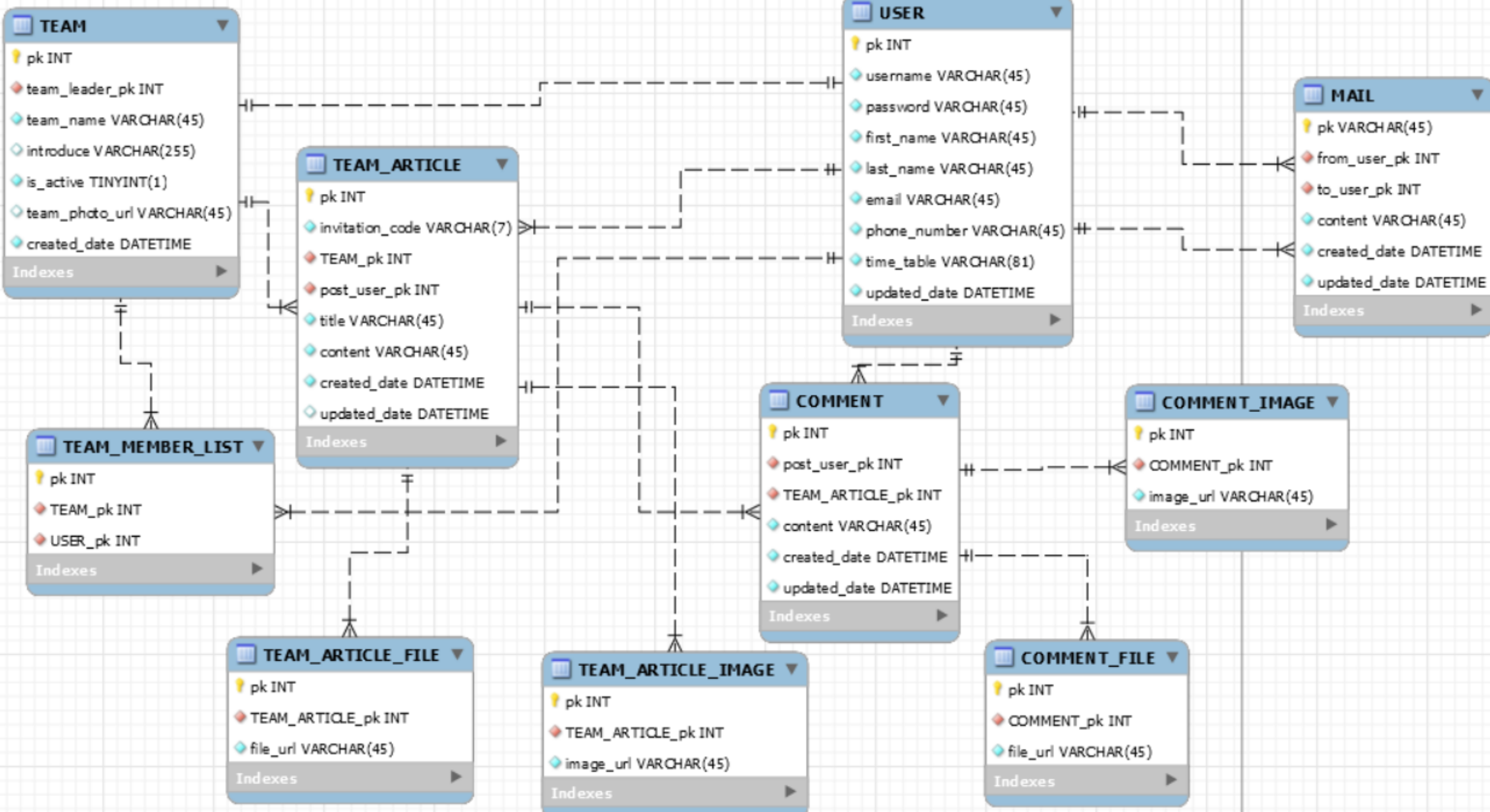
# 관계형 데이터베이스

## 간단요약!

행과 열이 있는 **테이블**에 데이터가 저장되고 그 **테이블들의 관계**로써 데이터베이스가 만들어진다.







MODEL

# Model

## Model이란?

Django 에서 모델은 데이터를 데이터베이스에 저장하고 데이터를 가공하는 작업을 하며 데이터 서비스를 제공하는 기능을 담당한다. Django의 Model은 각각의 Django App 안에 기본적으로 생성되는 models.py에 코드를 작성하여 정의한다. models.py 에서는 하나 이상의 모델을 정의 할 수 있으며 클래스 형태로 정의한다. 여기서 **하나의 모델 클래스는 데이터베이스에서 하나의 테이블에 해당된다.**

# Model

## 간단요약!

1. 데이터베이스를 담당
2. Models.py 에 코드를 작성
3. Class형태로 정의
4. 하나의 모델 클래스는 데이터베이스에서 하나의 테이블에 해당

실습으로 ㄱ ㄱ

github.com/Sinhaecheol/DbModelAdmin

Sinhaecheol / DbModelAdmin

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Security Insights Settings

No description, website, or topics provided. Edit

Manage topics

1 commit 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find File Clone or download

Sinhaecheol first commit Latest commit 20a72ca 3 hours ago

.vscode	first commit	3 hours ago
DbModelAdmin	first commit	3 hours ago
app	first commit	3 hours ago
venv	first commit	3 hours ago
db.sqlite3	first commit	3 hours ago
manage.py	first commit	3 hours ago

Help people interested in this repository understand your project by adding a README. Add a README

프로젝트 이름: DbModelAdmin

가상환경 이름: venv

app이름: app



# Model

```
app > models.py > ...
1  from django.db import models
2
3  # Create your models here.
4  class Blog(models.Model):
5      title = models.CharField(max_length=100)
6      body = models.TextField()
7      pub_date = models.DateTimeField('date published')
```

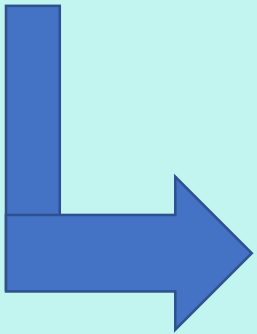
App/models.py

참고로 model, template, view를 코딩하는 순서는 따로 정해져 있지는 않으나 독립적으로 개발할 수 있는 model을 먼저 개발하고 view와 template을 같이 개발하는 것이 일반적이다.

# Model

models안에 있는 Model상속

```
app > models.py > ...  
1  from django.db import models  
2  
3  # Create your models here.  
4  class Blog(models.Model):  
5      title = models.CharField(max_length=100)  
6      body = models.TextField()  
7      pub_date = models.DateTimeField('date published')
```



테이블 이름: Blog

id	title	body	pub_date
----	-------	------	----------

# Model

장고와 데이터베이스는 **별개!**

장고는 데이터베이스에게 어떤 데이터를 처리하기로 했는지 알려주어야 함!

```
python manage.py makemigrations
```

데이터베이스가 알아듣게끔 하는 과정

```
python manage.py migrate
```

우리가 만든 것을 적용하는 과정

model의 수정이 있을 때마다 해주어야 한다!

그럼 이제 데이터를 넣어볼까?

admin페이지 ㄱ ㄱ ㄱ ㄱ

Django administration

Username:

Password:

Log in

?

ADMIN

# Admin

admin페이지에 접속하기 위해선 admin계정이 필요하다!

`python manage.py createsuperuser`  
admin계정을 만들어주는 명령어

username: admin

password: admin123

까먹으면 안돼!!!!!!!!!!

# Admin

다시 admin페이지에 접속을 해보면...?



## Site administration

### AUTHENTICATION AND AUTHORIZATION

**Groups**

[+ Add](#) [✎ Change](#)

**Users**

[+ Add](#) [✎ Change](#)

### Recent actions

### My actions

None available

터터  
오오

# Admin

```
app > admin.py > ...  
1 from django.contrib import admin  
2 from .models import Blog  
3 # Register your models here.  
4 admin.site.register(Blog)
```

app/admin.py

원래는 app.models 라고 쓰는 것이 맞지만 같은 폴더안에 있을 때는 앱이름 생략해도 ㄱ

admin.site.register(Blog)

admin사이트에 Blog클래스를 등록 해주겠다는 의미


# Admin

## Site administration

APP

Blogs


+ Add

 Change

AUTHENTICATION AND AUTHORIZATION


Groups

+ Add

 Change

Users

+ Add

 Change

Recent actions

My actions

None available

# Admin

Home › App › Blogs › Add blog

Add blog

Title:

test1

Body:


1

Date published:

Date:

2019-09-20


Today



Time:

04:35:46

Now



Note: You are 9 hours ahead of server time.

Save and add another

Save and continue editing

SAVE

SAVE ㄣㄣ

# Admin

Select blog to change

Action:   0 of 1 selected

<input type="checkbox"/>	<b>BLOG</b>
<input type="checkbox"/>	<b>Blog object (2)</b>

---

1 blog

저장된 것을 확인할 수 있다!

# Admin

☐ Blog object (2)



☐ test1

```
class Blog(models.Model):  
    title = models.CharField(max_length=100)  
    body = models.TextField()  
    pub_date = models.DateTimeField('date published')  
  
    def __str__(self):  
        return self.title
```



목록이 title로 뜨게 해준다.  
참고!

# Admin

테이블 이름: Blog

id	title	body	pub_date
1	test1	안녕하세요	2019-09-20

■

■

■

테이블에 데이터가 추가되는 형식이다

데이터를 넣었으니  
보여줘야지!



# QuerySet&Method

template에 데이터를 띄워주기 위해서 view가 model에게 받은 데이터(객체)를 목록화 시킨것이 쿼리셋(queryset)이다.

**요약!** 전달받은 모델의 객체목록

메소드는 객체목록을 다루는 함수!

# QuerySet&Method

```
app > views.py > home
1  from django.shortcuts import render
2  from .models import Blog
3
4  def home(request):
5      blogs = Blog.objects.all()
6      return render(request, 'app/home.html', {'blogs': blogs})
```

app/views.py

from .models import Blog

view에게 객체를 넘겨주는 코드

blogs = Blog.objects.all()

Blog는 클래스, objects는 쿼리셋, all()은 메소드

# QuerySet&Method

```
app > templates > app > home.html > h1
3 | <h1>hello world</h1>
4 | {{ blogs }}
```

app/templates/app/home.html



<Blog: test1>, <Blog: asd> .....

객체가 리스트 형식으로 넘어왔다!

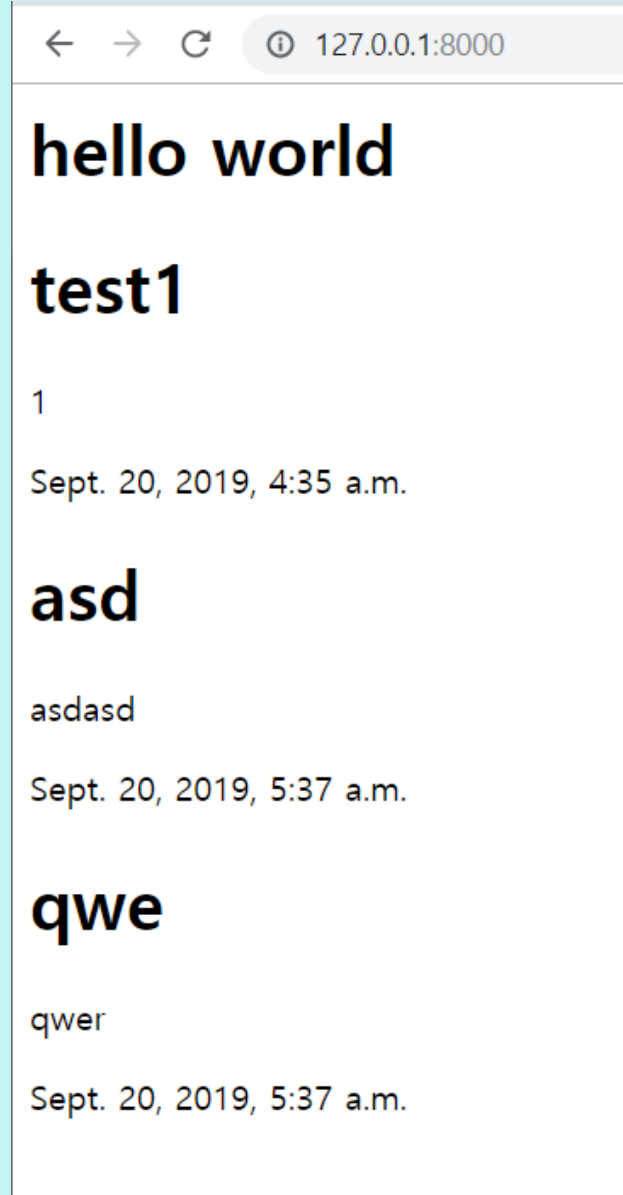
# QuerySet&Method

```
app > templates > app > <> home.html > ...  
1 <h1>hello world</h1>  
2 {% for blog in blogs.all %}  
3   <h1>{{ blog.title }}</h1>  
4   <p>{{ blog.body }}</p>  
5   <p>{{ blog.pub_date }}</p>  
6 {% endfor %}
```



app/templates/app/home.html

객체안의 정보들을 자세하게 보여주려면  
for문을 돌려서 객체안에 있는 정보들을  
차례대로 찍는다!



# QuerySet&Method

참고!

템플릿 안에 쓰는 {}. 이것을 템플릿 언어라고 한다.

{{ }} : 템플릿 변수 {% %}: 템플릿 태그

실습 때 썼던 all() 말고도 다양한 메소드들이 있다!

모를 땐 역시 구글링!

끝!