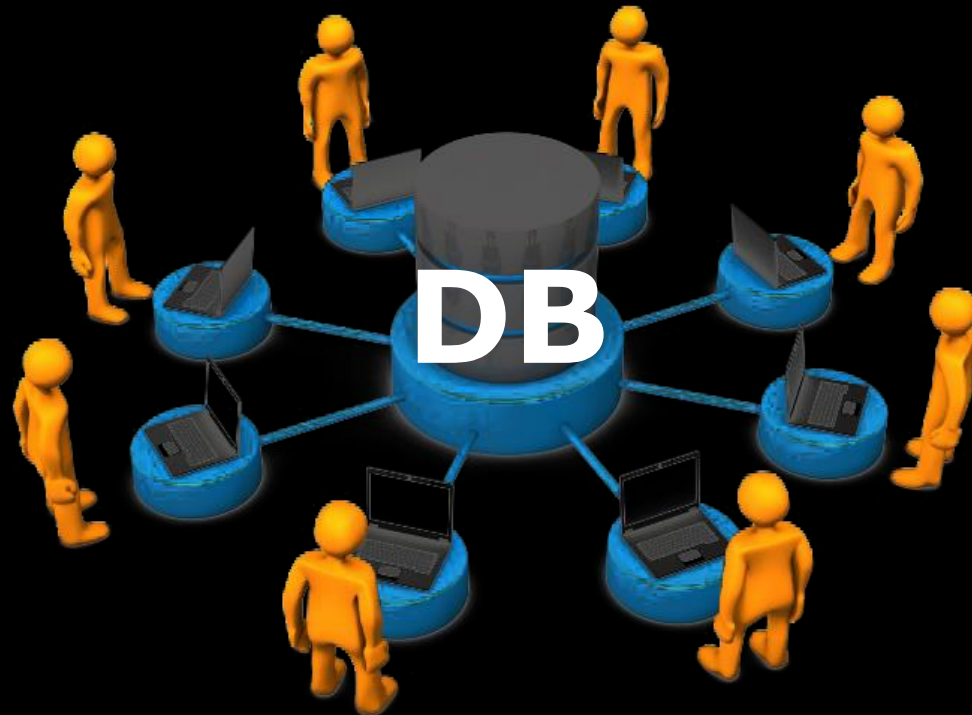


# model, 검색, admin

멋쟁이사자처럼 11기 운영진 강경아

# 데이터베이스란?

- 통합하여 관리되는 데이터의 **집합체**
- 하나의 데이터베이스에 여러 값을 입/출력합니다.



# SQL(Structured Query Language)

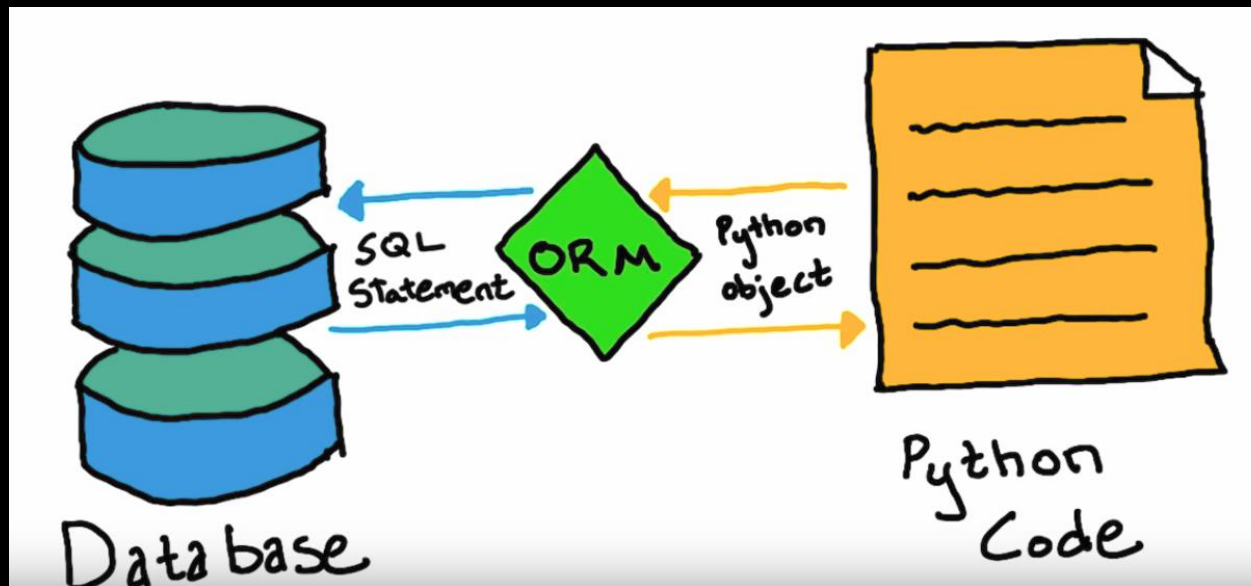
- 데이터를 정의, 조작, 제어하는 언어
- 다음과 같은 데이터 테이블이 만들어진다고 상상하면 좋습니다.
- SQL에 대해 다루지는 않을 거지만, 기본적인 개념을 알아보는 걸 권장합니다.

id	이름	나이	학과
1	김멋사	20	컴공과
2	김멋쟁	21	토목과
3	김사자	22	컴과
4	김처럼	24	국문과

- <https://youtu.be/z9chRID1tec> (sql 기초에 대한 설명 추천 영상)

# ORM(Object Relational Mapping)

- 우리가 직접 SQL 언어를 배우려면 시간이 걸리기 때문에, Python 코드로 SQL 내용을 전달할 수 있도록 **ORM**이 존재합니다.
- 간단히 말해, Python으로 코드를 짜면, ORM이 그 코드를 SQL문으로 **변환**해 줍니다.



# model 작성

- 앞에서 보았던 데이터베이스의 모양을 정의해보겠습니다.

Id	제목	시간	본문
자동생성			

```
blog > 🐍 models.py > 📄 Post
1  from django.db import models
2
3  # Create your models here.
4  class Post(models.Model):
5      title = models.CharField(max_length=50)
6      pub_date = models.DateTimeField()
7      body = models.TextField()
```

# Field

Field는 데이터베이스에 데이터를 저장하는 데 사용되는 유형을 결정합니다.

이름	설명	사용처
CharField	짧은 글 저장	게시글 제목
TextField	긴 글 저장	게시글 본문
IntegerField	숫자 저장	게시글 순서
DateField	날짜 저장	날짜

- 그 외에도 여러가지 종류의 field가 있습니다. 장고 공식 사이트 참고!  
<https://docs.djangoproject.com/en/4.1/ref/models/fields/>


# 모델을 데이터베이스로 매핑해보자!

- `python manage.py makemigrations`  
migrations 파일을 만들어서 모델의 변경 내역을 저장합니다.  
터미널에 입력해보겠습니다.

```
$ python manage.py makemigrations
Migrations for 'blog':
  blog\migrations\0001_initial.py
    - Create model Post
(myvenv)
```

# Admin Page

- 우리가 만들었던 클래스의 모델을 직접 눈으로 확인해봅시다.
- **Admin** 페이지는 데이터베이스를 웹에서 쉽게 **확인, 관리**할 수 있게 해주는 페이지입니다.
- 다른 프레임워크들과 다르게 Django에서는 admin 페이지를 **지원합니다.**

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

models.py에서 Post라는 클래스를 가져오겠다!

불러온 post 모델을 admin page에 등록하겠다!



# 모델을 데이터베이스로 매핑해보자!

- `python manage.py migrate`

아까 만들었던 migrations 파일에 담긴 변경 내역을 실제로 SQL에 적용시킵니다.

```
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying sessions.0001_initial... OK
(myenv)
```

# 관리자 계정 생성

- Admin 페이지에 접근하기 위해 관리자 계정을 생성해줍니다.

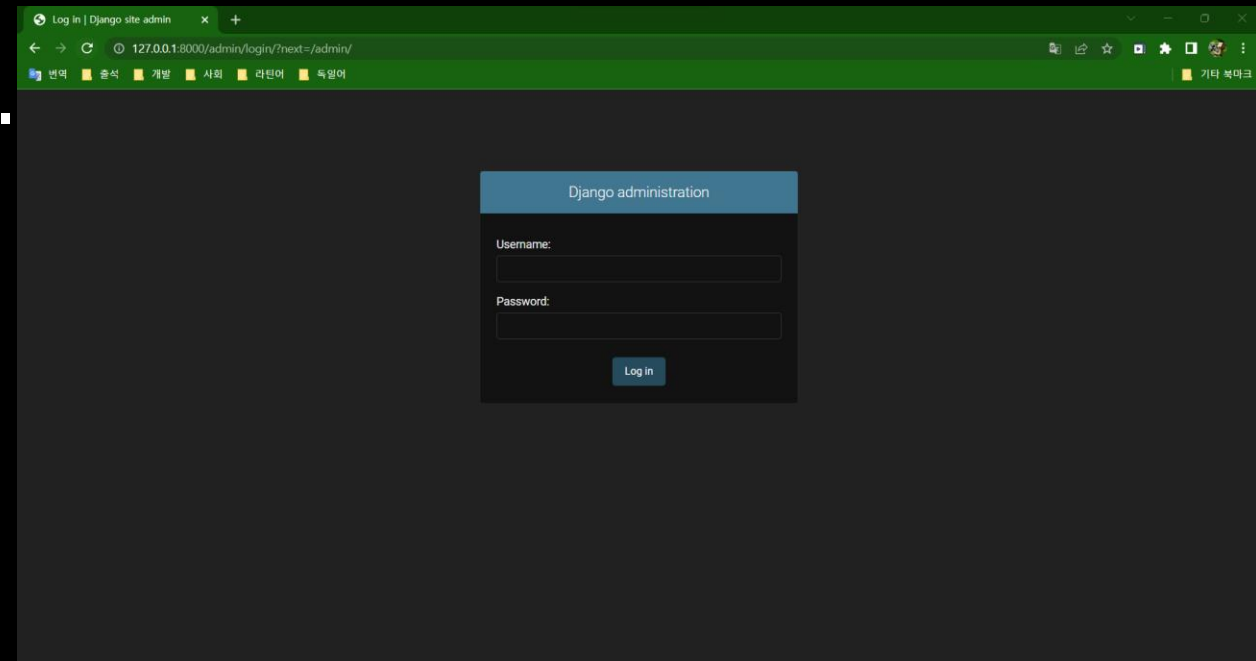
`python manage.py createsuperuser`

최고 관리자 계정 생성 명령어

```
$ python manage.py createsuperuser
Username (leave blank to use 'kga09'):
Email address: 공란 가능
Password: 원래 타이핑이 보이지 않아요!
Password (again):
Superuser created successfully.
(myvenv)
```

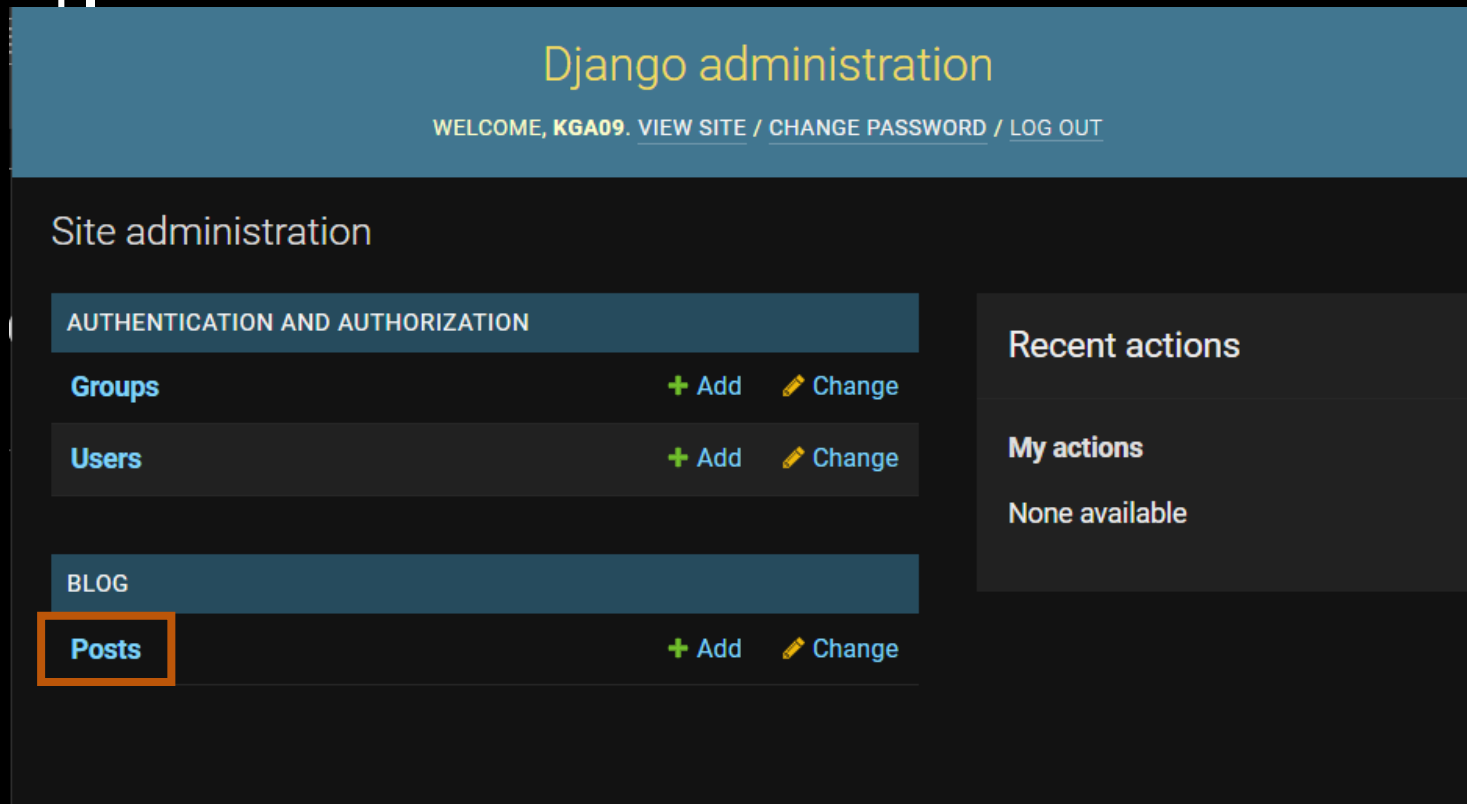
# Admin Page

- 다시 `python manage.py runserver`를 해줍니다.
- 페이지 주소에 admin을 붙여줍니다.  
`http://127.0.0.1:8000/admin`
- 그러면 아래와 같은 창이 나옵니다.
- 로그인을 해주세요!



# Admin page

- 로그인하면 우리가 만들었던 모델인 **Post 모델**이 있는 걸 볼 수 있습니다.



# Post 추가

Django administration
WELCOME, KGA09. [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

Home > Blog > Posts > Add post

Start typing to filter...

AUTHENTICATION AND AUTHORIZATION

Groups + Add
Users + Add

BLOG
Posts + Add

Add post

Title:
입력

Pub date:

Date: 2023-02-03
Today |

Time: 00:36:40
Now |

선택

Note: You are 9 hours ahead of server time.

Body:

입력

Save and add another
Save and continue editing
SAVE

# Post 추가

Django administration
WELCOME, KGA09. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Blog > Posts

Start typing to filter...

AUTHENTICATION AND AUTHORIZATION

Groups + Add

Users + Add

BLOG

Posts + Add

The post "Post object (1)" was added successfully.

Select post to change
ADD POST +

Action: Go 0 of 1 selected

☐ POST

☐ Post object (1)

1 post

Post object(1)에서 글 제목으로 수정하고  
시다며?

```

blog > models.py > ...
1 from django.db import models
2
3 # Create your models here.
4 class Post(models.Model):
5     title = models.CharField(max_length=50)
6     pub_date = models.DateTimeField()
7     body = models.TextField()
8
9     def __str__(self):
10         return self.title

```

Magic method 사용!