

2장. 데이터 모델(Model)



데이터베이스와 모델

장고는 모델(Model)로 데이터를 관리한다. 데이터를 저장하고 조회, 삭제하는 등의 관리를 기존의 SQL 언어로 하지 않고 클래스(객체)를 사용한다.

◆ 장고 서버 구동시 경고 메시지

```
(mysite) c:\Webproject\Wpolls>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).

You have 18 unapplied migration(s). Your project may not work properly
without migrations.
Run 'python manage.py migrate' to apply them.
```

※ 아직 적용되지 않은 18개의 migration이 있다.

데이터베이스와 모델

◆ migrate 명령으로 앱 설정

```
(mysite)C:\projects\polls>python manage.py migrate
```

```
(mysite) c:\Webproject\polls>python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
```

◆ config/settings.py – 기본 설치 앱 확인

```
INSTALLED_APPS = [
    'poll.apps.PollConfig',
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
```

데이터베이스와 모델

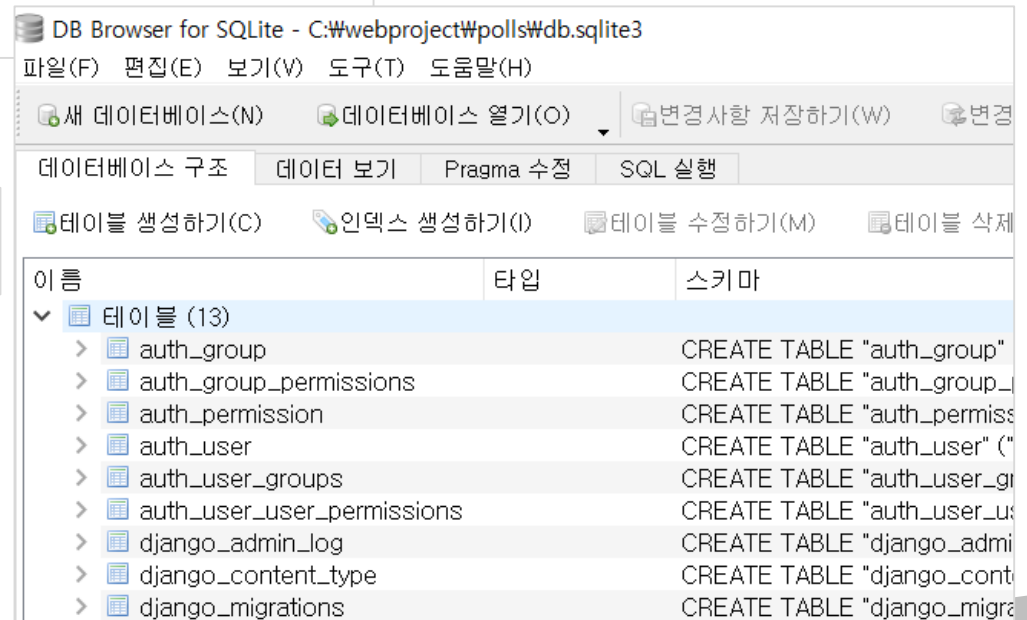
◆ DB Browser for SQLite로 데이터베이스 열기

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.sqlite3',  
        'NAME': BASE_DIR / 'db.sqlite3',  
    }  
}
```

config/setting.py

db 위치

C:\projects\polls\db.sqlite3



ORM(Object Relational Mapping)

◆ ORM(Object Relational Mapping) – 객체 관계 매핑 방식

파이썬은 현재 가장 많이 사용하고 있는 관계형 데이터베이스 방식을 개선한 ORM 방식을 사용한다.

테이블을 모델화(클래스의 객체)하여 DB를 획기적으로 관리할 수 있으며, 특히 SQL 언어의 문법을 사용하지 않는다.

SQL을 사용했을 경우 단점은

- SQL 쿼리문은 개발자마다 다양하게 작성되므로 일관성이나 유지 보수 측면에서 심플하지 않고 복잡할 수 있다.
- 또한, 개발자가 쿼리문 자체를 잘못 작성하여 시스템의 성능이 저하 될 수 있다.

ORM(Object Relational Mapping)

◆ Django Documentation > The model layer > introduction

This example model defines a **Person**, which has a **first_name** and **last_name**:

```
from django.db import models

class Person(models.Model):
    first_name = models.CharField(max_length=30)
    last_name = models.CharField(max_length=30)
```

first_name and **last_name** are **fields** of the model. Each field is specified as a class column.

The above **Person** model would create a database table like this:

```
CREATE TABLE myapp_person (
    "id" serial NOT NULL PRIMARY KEY,
    "first_name" varchar(30) NOT NULL,
    "last_name" varchar(30) NOT NULL
);
```

모델(Model) – 필드 타입(type)

◆ 필드(속성) 타입

Field Type	설 명
CharField	제한된 문자열 타입, max length 옵션 사용
TextField	제한이 없는 가변 길이 대용량 문자열 타입
IntegerField	32bit 정수 타입 (-21억 ~ 21억)
BigIntegerField	64bit 정수 타입 (조, 경 이상)
DateTimeField	날짜와 시간을 갖는 타입
ImageField	이미지 업로드를 위한 필드
FileField	파일 업로드를 위한 필드

설문 투표 모델 – Question, Choice

◆ 설문 투표 앱에 사용할 Question, Choice 모델

Question 모델

칼럼 이름	자료형	설명
question_text	CharField(max_length=200)	질문내용
pub_date	DateTimeField()	출판일

Choice 모델

칼럼 이름	자료형	설명
choice_text	CharField(max_length=200)	답변내용
votes	IntegerField(default=0)	투표수
question	ForeignKey(Question, on_delete=models.CASCADE)	질문내용 (외래키)

Question 모델 만들기

- poll/models.py에 Question 모델 생성

```
from django.db import models
```

```
class Question(models.Model):
```

models.Model 을 상속받은 Question 클래스

```
    question_text = models.CharField(max_length=200)
```

```
    pub_date = models.DateTimeField()
```

```
    def __str__(self):
```

```
        return self.question_text
```

객체 정보를 문자열로 출력하는 함수

Question 모델 만들기

1. makemigrations 명령으로 테이블 관리 시작하기

```
(mysite)C:\projects\polls>python manage.py makemigrations
```

```
operations = [  
    migrations.CreateModel(  
        name='Question',  
        fields=[  
            ('id', models.BigAutoField(auto_created=True, primary_key=True,  
            ('subject', models.CharField(max_length=200)),  
            ('content', models.TextField()),  
            ('create_date', models.DateTimeField()),  
        ],  
    ),  
]
```

migrations/0001.initial.py

2. makemigrations 후 다시 migrate 명령 실행

```
(mysite)C:\projects\polls>python manage.py migrate
```

데이터 저장(삽입)

◆ 장고 셸 실행하기

```
(mysite)C:\projects\polls>python manage.py shell
```

◆ 자료 삽입 – Question 모델의 객체 생성

```
q = Question(question_text='', pub_date=timezone.now())
```

자료 삽입시 반드시 – `save()`, shell 종료 – `exit()`, `quit()`

```
>>> from poll.models import Question
>>> from django.utils import timezone
>>> Question.objects.all()
<QuerySet []>
>>> q = Question(question_text='당신이 접종받은 백신은 무엇인가요?', pub_date=timezone.now())
>>> q.save()
```

데이터 조회

◆ 자료 조회

전체 조회 : `Question.objects.all()`

1개 조회 : `Question.objects.get(id=1)`

```
>>> Question.objects.all()
<QuerySet [<Question: Question object (1)>]>
>>> q.id
1
>>> q.question_text
'당신이 접종받은 백신은 무엇인가요?'
>>> q.pub_date
datetime.datetime(2022, 2, 23, 21, 8, 7, 366727, tzinfo=datetime.timezone.utc)
>>> Question.objects.get(id=1)
<Question: Question object (1)>
```

데이터 수정 및 삭제하기

◆ 자료 수정

```
>>> q = Question.objects.get(id=1)
>>> q.question_text = "접종 받은 백신의 종류를 선택하세요"
>>> q.save()
```

◆ 자료 삭제

```
>>> q2 = Question.objects.get(id=2)
>>> q2.delete()
(4, {'poll.Choice': 3, 'poll.Question': 1})
>>> Question.objects.all()
<QuerySet [<Question: 접종 받은 백신의 종류를 선택하세요>]>
```

Choice 모델 만들기

- poll/models.py에 Choice모델 생성

```
from django.db import models
```

```
class Choice(models.Model):
```

```
    choice_text = models.CharField(max_length=200)
```

```
    votes = models.IntegerField(default=0)
```

```
    question = models.ForeignKey(Question, on_delete=models.CASCADE)
```

```
    #외래키(CASCADE-외래키에 연결된 데이터 삭제)
```

```
    def __str__(self):
```

```
        return self.choice_text
```

Choice 모델 만들기

◆ makemigrations 명령으로 테이블 관리 시작하기

```
(mysite)C:\projects\polls>python manage.py makemigrations
```

◆ makemigrations 후 다시 migrate 명령 실행

```
(mysite)C:\projects\polls>python manage.py migrate
```

Choice – 자료 삽입

◆ 자료 삽입

q= Question.objects.get(id=1)

c = Choice(choice_text="", votes=0, question=q)

```
>>> q = Question.objects.get(id=1)
>>> c = Choice(choice_text='화 이 자 ', votes=0, question=q)
>>> c.save()
>>> c = Choice(choice_text='모 더 나 ', votes=0, question=q)
>>> c.save()
```


Choice – 자료 조회

◆ 자료 조회

연결된 자료로 전체 조회하기 - **q.choice_set.all()**

```
>>> q1 = Question.objects.get(id=1)
>>> q1.choice_set.all()
<QuerySet [<Choice: 화 이 자 >, <Choice: 모 더 나 >, <Choice: 얀 센 >]>
```

1개 조회하기 - **q.choice_set.get(id=1)**

```
>>> q1.choice_set.get(id=2)
<Choice: 모 더 나 >
```

Choice – 자료 수정

◆ 자료 수정

- 자료 가져오기 `c = Choice.objects.get(id=2)`
- `c.choice_text = "수정내용"`

```
>>> c = Choice.objects.get(id=2)
>>> c.choice_text
'모 더 나 '
>>> c.choice_text = '아스 크 라 제 네 카 '
>>> c.save()
```

```
>>> c.votes
1
>>> c.votes = 3
>>> c.save()
>>> c.votes
3
```

Choice – 자료 삭제

◆ 자료 삭제

- **c.delete()**

```
>>> c.delete()
(1, {'poll.Choice': 1})
>>> Choice.objects.all()
<QuerySet [<Choice: 화 이 자 >, <Choice: 얀 센 >]>
```

장고 Admin

◆ 장고 Admin

- 장고는 Admin은 관리자 기능을 갖춘 app이다.
- 모델 관리를 shell 환경이 아닌 사이트 화면에서 할 수 있다.
- 127.0.0.1:8000/admin으로 접속.
- 먼저 슈퍼 유저를 생성해야 함.

```
(mysite)C:\projects\polls>python manage.py createsuperuser
```

```
사용자 이름 : admin
```

```
이메일 주소 :
```

```
Password:*****
```

```
Password(again):*****
```

장고 Admin

◆ localhost:8000/admin 에 접속하기

Django 관리

사용자 이름:

비밀번호:

로그인

admin / 12345

Django 관리

사이트 관리

POLL

Choices + 추가 ✎ 변경

Questions + 추가 ✎ 변경

인증 및 권한

그룹 + 추가 ✎ 변경

사용자(들) + 추가 ✎ 변경

장고 Admin

- ◆ 장고 Admin에서 모델 관리하기
 - poll/admin.py 에 등록

```
from django.contrib import admin
from .models import Question, Choice

admin.site.register(Question)
admin.site.register(Choice)
```

장고 Admin

◆ Question 추가하기

question 추가

Question text:	<input type="text" value="당신의 취미는 무엇입니까?"/>	
Pub date:	날짜: <input type="text" value="2021-12-12"/> 오늘 	
	시각: <input type="text" value="07:06:45"/> 현재 	



장고 Admin

◆ Choice 추가하기

choice 추가


Choice text:

Votes:

Question:  

✓ Choice "독서"가 성공적으로 추가되었습니다.

변경할 choice 선택

액션: 

<input type="checkbox"/>	CHOICE
<input type="checkbox"/>	독서
<input type="checkbox"/>	운동
<input type="checkbox"/>	영화 감상
<input type="checkbox"/>	컴퓨터 게임
<input type="checkbox"/>	모더나
<input type="checkbox"/>	안센
<input type="checkbox"/>	화이자

DB Browser for SQLite3

◆ DB Browser

DB Browser for SQLite - C:\webproject\polls\polls.db.sqlite3

파일(F) 편집(E) 보기(V) 도구(T) 도움말(H)

새 데이터베이스(N) 데이터베이스 열기(O) 변경사항 저장(S)

데이터베이스 구조 데이터 보기 Pragma 수정 SQL 실행

테이블(T): poll_question

	id	question_text	pub_date
...	필터		필터
1	1	어떤 백신을 접종했나요?	2021-12-11 13:29:46.879587
2	2	당신의 취미는 무엇입니까?	2021-12-11 22:04:04

새 데이터베이스(N) 데이터베이스 열기(O)

데이터베이스 구조 데이터 보기 Pragma 수정

테이블(T): poll_choice

	id	choice_text	votes	question_id
...	필터		필터	필터
1	1	화이자	3	1
2	2	얀센	2	1
3	3	모더나	1	1
4	4	컴퓨터 게임	0	2
5	5	영화 감상	0	2
6	6	운동	0	2
7	7	독서	0	2