G·CAMP

G-CAMP WORKSHOP

BUILD THE RIGHT THING
BUILD THING RIGHT
SHIP FAST

Lecturer 블스
blogstudy@naver.com

G·CAMP

4강_데이터베이스(ORM)

- 테이블 생성
- 레코드 다루기(create, read, update, delete)

models.py에 테이블 클래스 정의 후 admin.py에 등록

```
from django.db import models

class Student(models.Model):
        s_name = models.CharField(max_length=100)
        s_major = models.CharField(max_length=100)
        s_age = models.IntegerField(default=0)
        s_grade = models.IntegerField(default=0)
        s_gender = models.CharField(max_length=30)

        def __str__(self):
                return self.s_name
```

student 테이블 생성을 위한 Student 클래스 정의

student table

id(PK)	s_name	s_major	s_age	s_grade	s_gender

```
from django.contrib import admin
from students.models import Student

admin.site.register(Student)
```

admin.py에 Student 등록

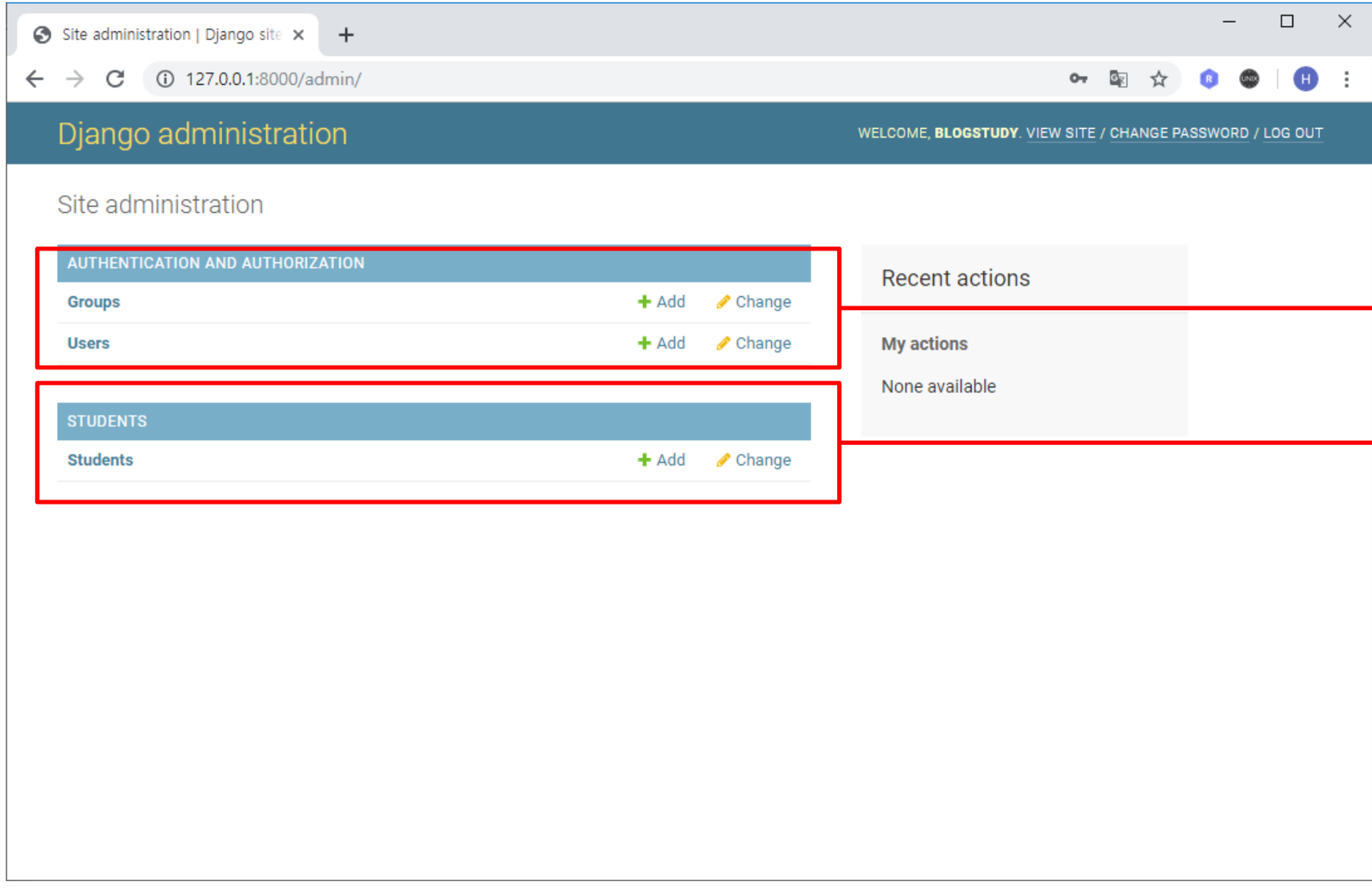
```
C:₩Django₩pjt₩tempProject>python manage.py makemigrations
Migrations for 'students':
  students₩migrations₩0001_initial.py
    - Create model Student

C:₩Django₩pjt₩tempProject>python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions, students
Running migrations:
  Applying students.0001_initial... OK

C:₩Django₩pjt₩tempProject>
```

DB변경사항 반영

4-1. 테이블 생성



기본 사용자 및 그룹 테이블

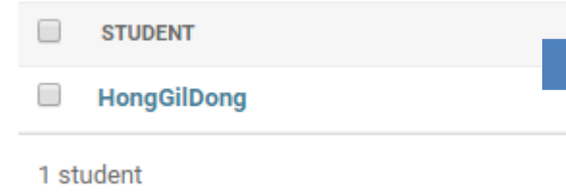
Student 테이블

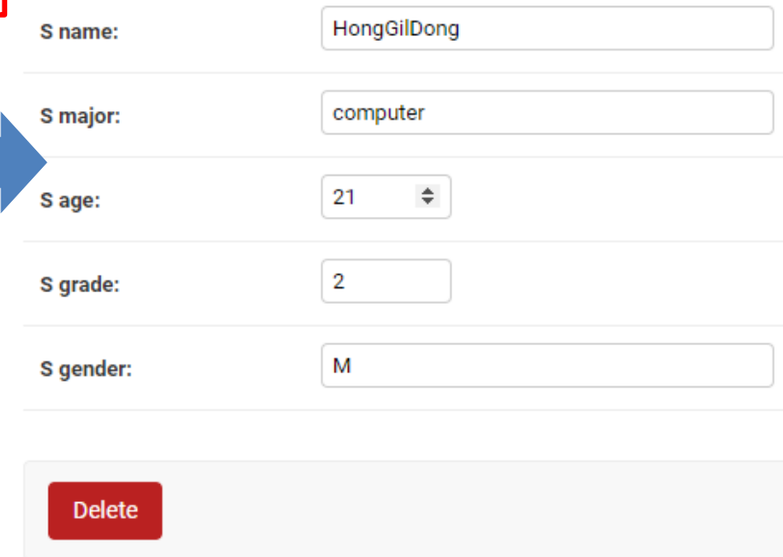
장고 shell모드 실행

```
C:₩Django₩pjt₩tempProject>python manage.py shell
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
(InteractiveConsole)
>>>
```

레코드 추가(create) – 데이터 생성

```
>>> from students.models import Student
>>> qs = Student(s_name='HongGilDong', s_major='computer', s_age=21, s_grade=2, s_gender='M')
>>> qs.save()
>>>
```





레코드 읽기(read) – 데이터 검색

```
>>> Student.objects.all()
<QuerySet [<Student: HongGilDong>, <Student: HongGilJa>, <Student: HongGilJaSoon>]>
>>> qs = Student.objects.all()
>>> print(qs)
<QuerySet [<Student: HongGilDong>, <Student: HongGilJa>, <Student: HongGilJaSoon>]>
>>> type(qs)
<class 'django.db.models.query.QuerySet'>
>>>
```

→ 데이터 전체 : QuerySet 타입으로 반환

```
>>> qs = Student.objects.get(s_name='HongGilDong')
>>> qs
<Student: HongGilDong>
>>> type(qs)
<class 'students.models.Student'>
>>>
```

→ 데이터 한개 : Student 타입으로 반환

레코드 읽기(read) – 필드 데이터 검색

데이터 다수 : 첨자([ ])를 이용한 접근

```
>>> qs = Student.objects.all()
>>> qs
<QuerySet [<Student: HongGilDong>, <Student: HongGilJa>, <Student: HongGilJaSoon>]>
>>> qs[1]
<Student: HongGilJa>
>>> qs[1].s_name
'HongGilJa'
>>> qs[1].s_age
22
```

데이터 한개 : '.'를 이용한 속성 접근

```
>>> qs = Student.objects.get(s_name='HongGilDong')
>>> qs.s_name
'HongGilDong'
>>> qs.s_age
21
>>> qs.s_major
'computer'
>>>
```

레코드 읽기(read) – 데이터 필터(filter)

```
>>> qs = Student.objects.filter(s_age__lt=22)
>>> qs
<QuerySet [<Student: HongGilDong>]>
>>> qs = Student.objects.filter(s_age__gt=22)
>>> qs
<QuerySet [<Student: HongGilJaSoon>]>
>>> qs = Student.objects.filter(s_age__lte=22)
>>> qs
<QuerySet [<Student: HongGilDong>, <Student: HongGilJa>]>
>>> qs = Student.objects.filter(s_age__gte=22)
>>> qs
<QuerySet [<Student: HongGilJa>, <Student: HongGilJaSoon>]>
>>>
```

__lt	~보다 작다
__lte	~보다 작거나 같다
__gt	~보다 크다
__gte	~보다 크거나 같다
__isnull	~ null인 자료 검색
__contains	특정 문자열을 포함하는 자료 검색
__startwith	특정 문자열로 시작하는 자료 검색
__endwith	특정 문자열로 끝나는 자료 검색

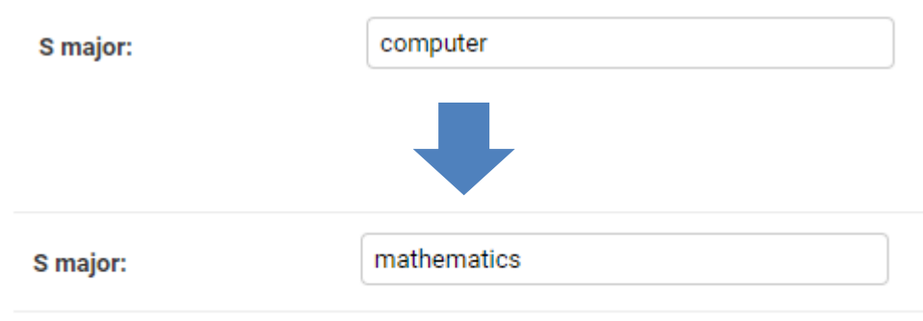
레코드 읽기(read) – 데이터 정렬

```
<QuerySet [<Student: HongGilDong>, <Student: HongGilJa>, <Student: HongGilJaSoon>]>
>>> qs = Student.objects.order_by('s_age')
>>> qs
<QuerySet [<Student: HongGilDong>, <Student: HongGilJa>, <Student: HongGilJaSoon>]>
>>> qs = Student.objects.order_by('-s_age')
>>> qs
<QuerySet [<Student: HongGilJaSoon>, <Student: HongGilJa>, <Student: HongGilDong>]>
>>>
```

order_by(' ')	오름차순
order_by('- ')	내림차순

레코드 업데이트(update) – 데이터 수정

```
>>> qs = Student.objects.get(s_name='HongGilDong')
>>> qs
<Student: HongGilDong>
>>> qs.s_major = 'mathematics'
>>> qs.save()
>>> qs.s_major
'mathematics'
```



레코드 삭제(delete) – 데이터 삭제

```
>>> qs = Student.objects.filter(s_age__gte=22)
>>> qs
<QuerySet [<Student: HongGilJa>, <Student: HongGilJaSoon>]>
>>> qs.delete()
(2, {'students.Student': 2})
>>>
```