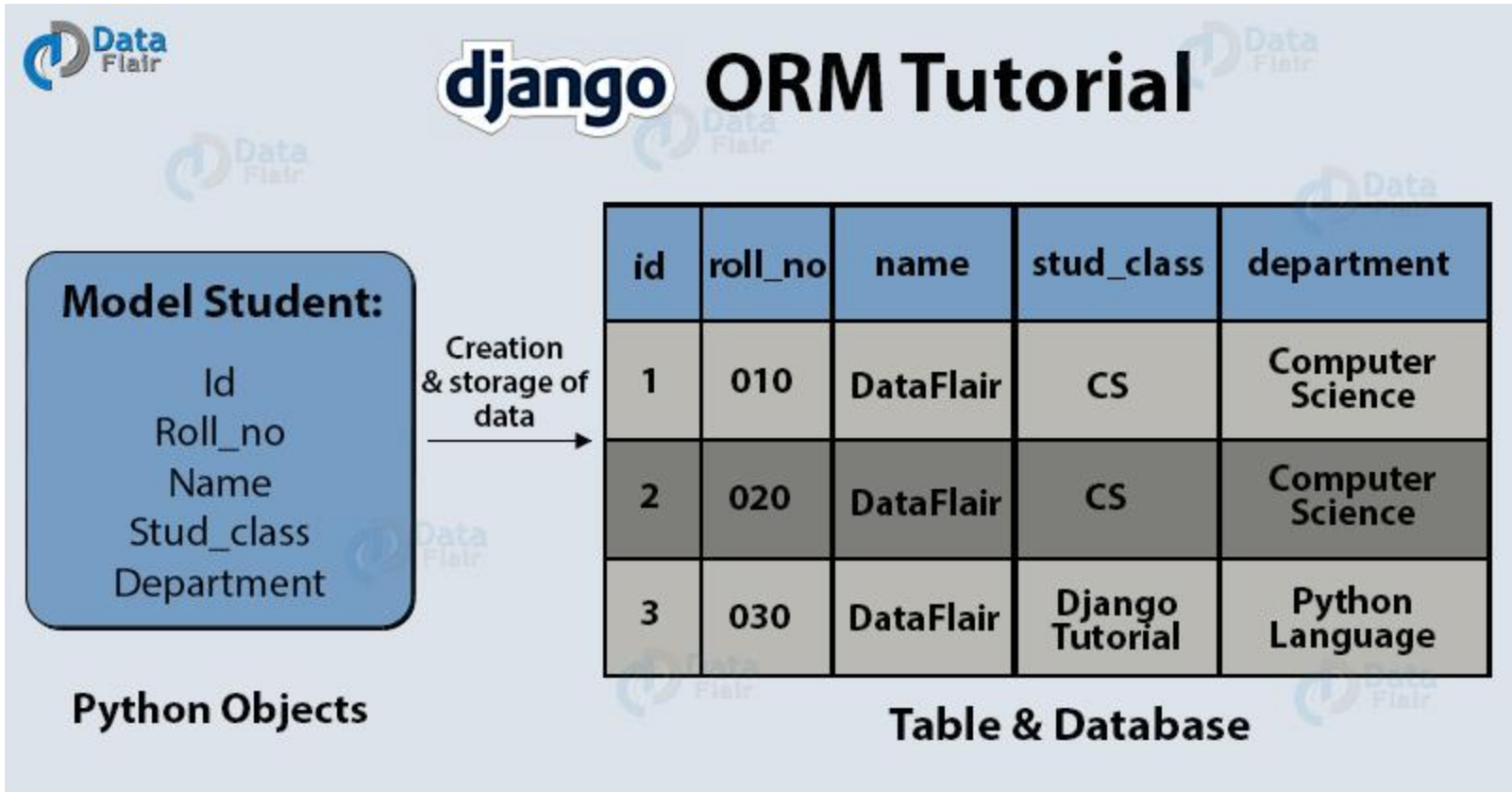


DJANGO DB 프로그래밍



ORM(Object Relation Model)



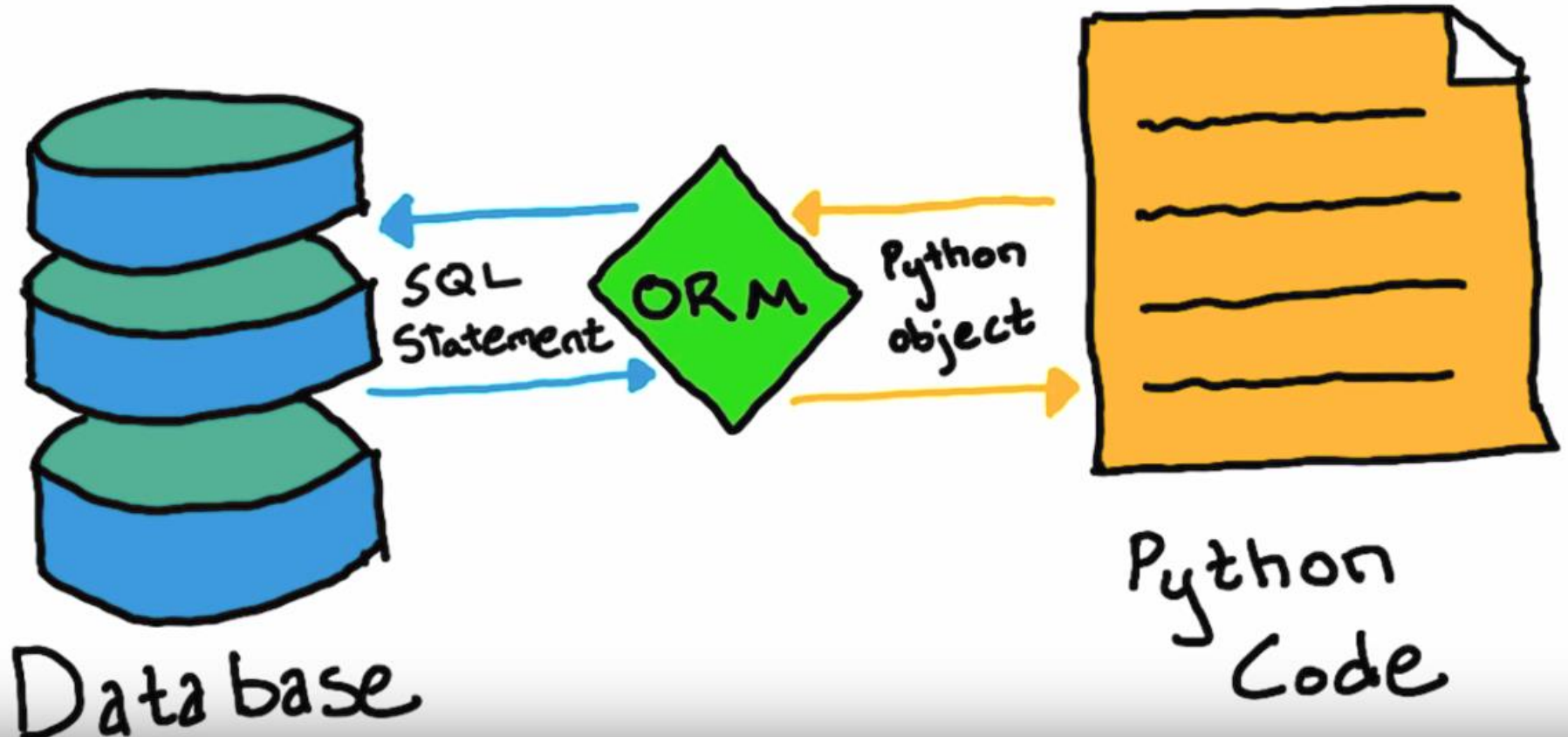
ORM(Object Relation Model)

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



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

ORM(Object Relation Mapping)

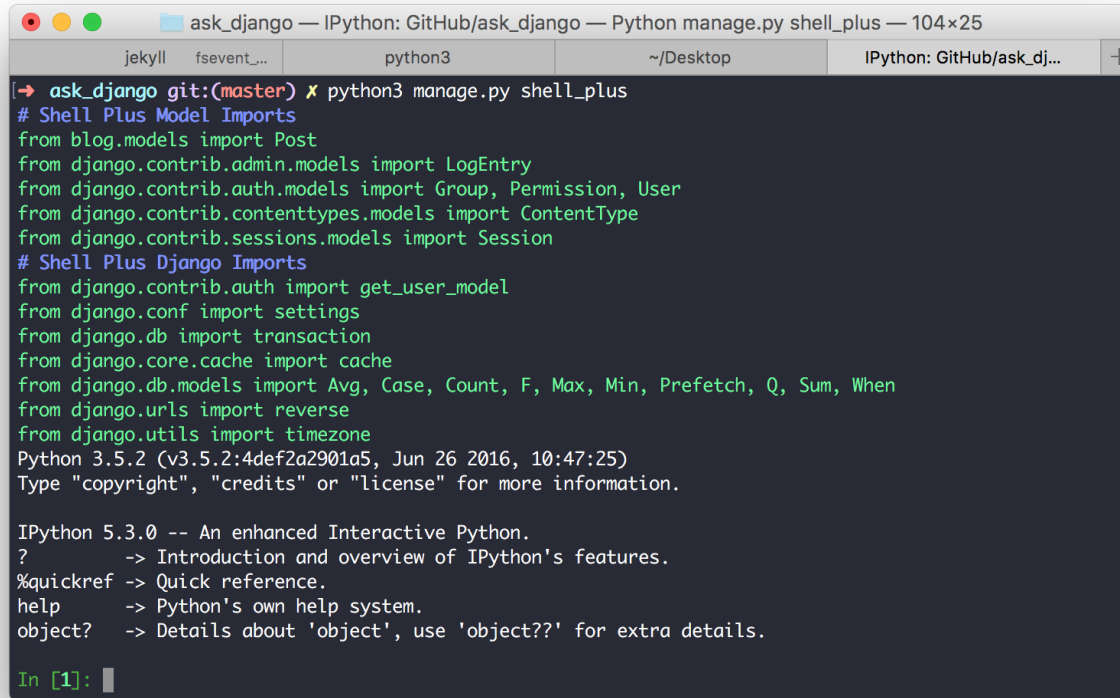


테이블 생성

- 관리자 및 user 생성
python manage.py createsuperuser
- Installed app 설정 (mysite/settings.py¹)
INSTALLED_APPS = ['polls.apps.PollsConfig ' , ... #앱명.apps.앱명Config
- db 관련 설정은 app 별로 설정
- Model 클래스 생성
class Post(models.Model)
- admin.py 수정
from django.contrib import admin
from .models import Post
admin.site.register(Post)
- DB 변경 사항 반영
python manage.py makemigrations myapp
python manage.py migrate

Shell

- Django 전용 python 해석기
- python manage.py shell



```
ask_django — IPython: GitHub/ask_django — Python manage.py shell_plus — 104x25
jekyll  fsevent...  python3  ~/Desktop  IPython: GitHub/ask_dj...  +
[→ ask_django git:(master) ✕ python3 manage.py shell_plus
# Shell Plus Model Imports
from blog.models import Post
from django.contrib.admin.models import LogEntry
from django.contrib.auth.models import Group, Permission, User
from django.contrib.contenttypes.models import ContentType
from django.contrib.sessions.models import Session
# Shell Plus Django Imports
from django.contrib.auth import get_user_model
from django.conf import settings
from django.db import transaction
from django.core.cache import cache
from django.db.models import Avg, Case, Count, F, Max, Min, Prefetch, Q, Sum, When
from django.urls import reverse
from django.utils import timezone
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 26 2016, 10:47:25)
Type "copyright", "credits" or "license" for more information.

IPython 5.3.0 -- An enhanced Interactive Python.
?          -> Introduction and overview of IPython's features.
%quickref  -> Quick reference.
help       -> Python's own help system.
object?    -> Details about 'object', use 'object??' for extra details.

In [1]:
```

```
pip install django-extensions
INSTALLED_APPS = [ 'django_extensions' , ]
python manage.py shell_plus --notebook
```

샘플 모으기(전체 검색, insert)



<https://wayhome25.github.io/django/2017/03/04/django-06-poll-project-3-shell/>

<https://medium.com/@peteryun/python-django%EB%A1%9C-%EC%9B%B9-%EC%84%9C%EB%B9%84%EC%8A%A4-%EA%B0%9C%EB%B0%9C%ED%95%98%EA%B8%B0-9-shell%EC%9D%84-%EC%9D%B4%EC%9A%A9%ED%95%9C-%EC%9E%A5%EA%B3%A0-%EA%B4%80%EB%A6%AC-450558ec6ed2>



- **Model Manager**
 - .objects
- 데이터베이스 질의 인터페이스를 제공
- 디폴트 Manager로서 모델클래스.objects 가 제공된다.
- Model Manager를 통해 해당 모델 클래스의 DB 데이터를 추가, 조회, 수정, 삭제 (CRUD)가 가능하다.
- **QuerySet**
- SQL을 생성해주는 인터페이스
- .objects.all()
- objects.create() :
- Chaining을 지원
- connection 모듈을 통해 queryset으로 만들어진 실제 sql문을 shell에서 확인
 - .objects.filter(title__icontains='1').filter(title__endswith='3')

```
from django.db import connection
ModelCls.objects.all().order_by('-id')[:10]
connection.queries[-1]
```


Select / Filtering



```
queryset = 모델클래스명.objects.all()
queryset = queryset.filter(조건필드1=조건값1, 조건필드2=조건값2)
queryset = queryset.filter(조건필드3=조건값3)
```

```
qs2 = Post.objects.filter(title__icontains='1').filter(title__endswith='3') # 체이닝
```

```
# from django.db.models import Q
.objects.all().filter(Q(조건필드1=조건값1) | Q(조건필드2=조건값2)) # or 조건
```

```
queryset = queryset[:10] # 현재 queryset에서 처음10개만
queryset = queryset[10:20] # 현재 queryset에서 처음10번째부터 20번째까지
```

```
get은 1개 일때만 가능(나머지는예외발생)
model_instance = queryset.get(title='my title')
```



```
queryset = queryset.order_by('field1') # 지정 필드 오름차순 요청  
queryset = queryset.order_by('-field1') # 지정 필드 내림차순 요청  
queryset = queryset.order_by('field2', 'field3') # 1차기준, 2차기준
```



- 필수필드 모두 지정해야한다. IntegrityError 발생
- blank=True, null=True, 디폴트값이 지정된 필드는 제외
- Model instance의 save함수로 저장
- Model manager의 create 함수로 저장

- 1:n 관계 생성

```
model_instance = Post(author=User.objects.all()[0], title='title', text='content')  
model_instance.save()
```

```
new_post = Post.objects.create(author=User.objects.get(id=1), title='title',  
text='content')
```

- **Model Instance** 속성을 변경하고, **save** 함수를 통해 저장

```
post_instance = Post.objects.get(id=66)
post_instance.title = 'edit title' # title 수정
post_instance.save()
```

- **QuerySet의 update** 함수에 업데이트할 속성값을 지정하여 일괄 수정

```
queryset = Post.objects.all()
queryset.update(title='test title') # 일괄 update 요청
```



- **Model Instance** 속성을 변경하고, **save** 함수를 통해 저장

```
post_instance = Post.objects.get(id=66)  
post_instance.delete()
```

- **QuerySet의 delete** 함수에 업데이트할 속성값을 지정하여 일괄 삭제

```
queryset = Post.objects.all()  
queryset.delete() # 일괄 delete 요청
```