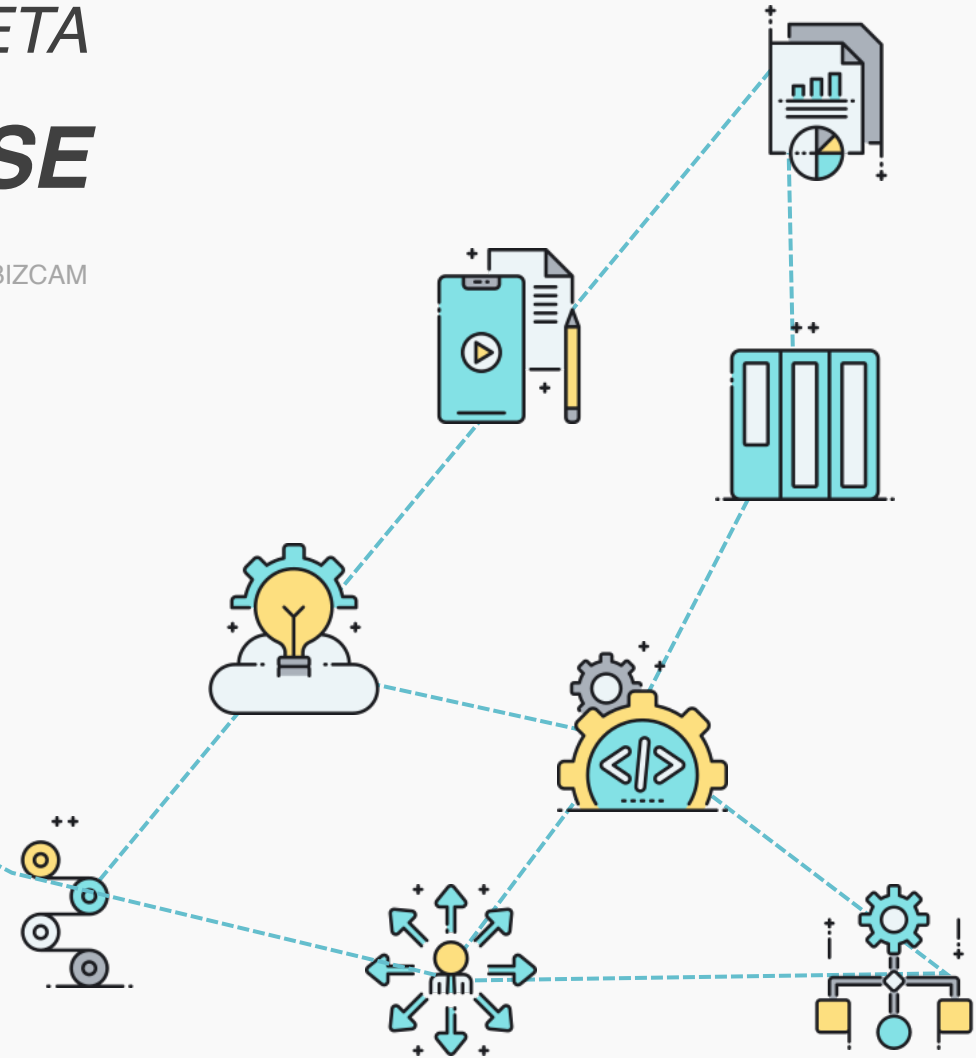


# *DIETA* **DATA BASE**

Enjoy your stylish business and campus life with BIZCAM





## Step. 1 .역할

Django database



## Step. 2 .조사

Database investigate



## Step. 3 .기능

Django database function



## Step. 4 .과정

Django database experiment



## Step. 5 .향후계획

Django database extra plane



++



## 1. Django Data-base의 역할

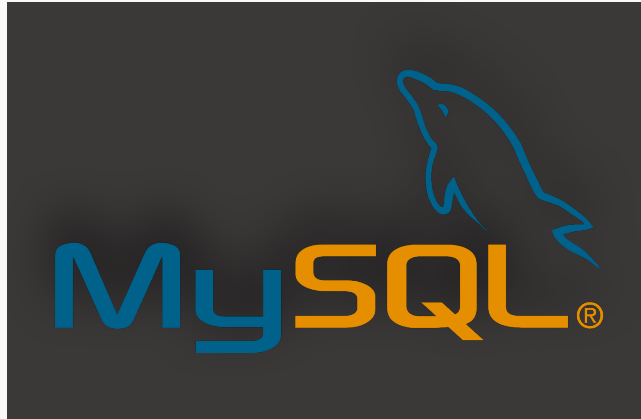
- 모든 애플리케이션에서 Data base 의 역할은 애플리케이션의 데이터창고 또는 심장부의 역할이다.
- Django는 SQLite3 데이터베이스 엔진을 사용하며 무료 서비스를 제공하고있다.
- MySQL or Oracle 등 다른 데이터베이스로 변경 하고 싶다면 **settings.py에서 변경이 가능하다.**

### 1-1. SQLite3는 무엇인가 ?



- 안드로이드와 아이폰에서 채택되면서 **모바일 환경에서 가장 영향력있는 DB**이다.
- 상대적으로 **라이브러리의 크기가 작고 메모리가 적은 환경에서도 성능은 뛰어나다.**
- 오픈 소스 프로젝트이고 특별한 라이선스 없이 이용이 가능하다.

## 2. Data-base 조사 - 웹 애플리케이션들은 보통 어떤 DB를 사용하는가?



### MySQL

가장 많이 사용하고  
오픈소스가 많은  
무료DBMS. Oracle에 병합흡수.  
(커뮤니티 버전)



### Oracle

가장 유명한 유료 DBMS.  
체계적인 관리와 시스템  
외국 DB시장의 50%를 차지.



### MongoDB

NOSQL 중 가장 많이 사용하는 DBMS.  
다양한 인덱싱으로 시장을 점유하고 있다.

### 3. Dieata Data-base의 기능. - 테이블 구성도 (추상)

- DIETA의 Application은 다양한 정보들을 User에게 제공해야 하므로 **DB의 관리와 API는 매우 중요하다.**

DATA		DATA	
회원의 정보	사는곳 , 나이, 이미지	판매제품 정보	가격, 다이어트 보조제의 데이터
그룹의 정보	브론즈, 실버, 골드, 플래티넘	게임 종류	매칭 시스템 게임의 정보
날씨의 데이터	맑음, 흐림, 비, 눈	포인트 점수	포인트를 관리할 수 있어야 하는 창
한국 지도	GPS를 표시	출석	출석체크 관리하는 창
운동별 칼로리정보	운동 칼로리 정보를 로드	달력	달력을 통해 다이어트일기
음식	음식 종류에대한 칼로리 데이터 셋		

### 3. Dieata Data-base의 기능. - 테이블 구성도 (세부)

ID	이름	사는곳	e - mail	등급	포인트	그룹	포인트
casuello	최진용	성남시 중원구 성남동	<a href="mailto:absd@gamil.com">absd@gamil.com</a>	Gold	90	Small	90

등급	포인트	(예시)등급	브론즈	실버	골드	플레티넘
Gold	90	Point	1-30	31-60	61-90	91-120

사는곳	GPS	위치정보	날씨	근처 그룹	날씨	날씨
성남시 중원구 성남동		표시	표기	검색	표기	맑음 흐림 비 눈

달력	출석	다이어트 일기	일일 목표(개인)	일일 목표(개인)	점수
날짜	체크	입력	85	일일 운동량 600kcal	10
				운동장소 찾아가기	5
				....	....
				추후입력	계100

다이어트 일기	음식 종류	음식 칼로리	운동 종류	운동칼로리
입력	입력		입력	

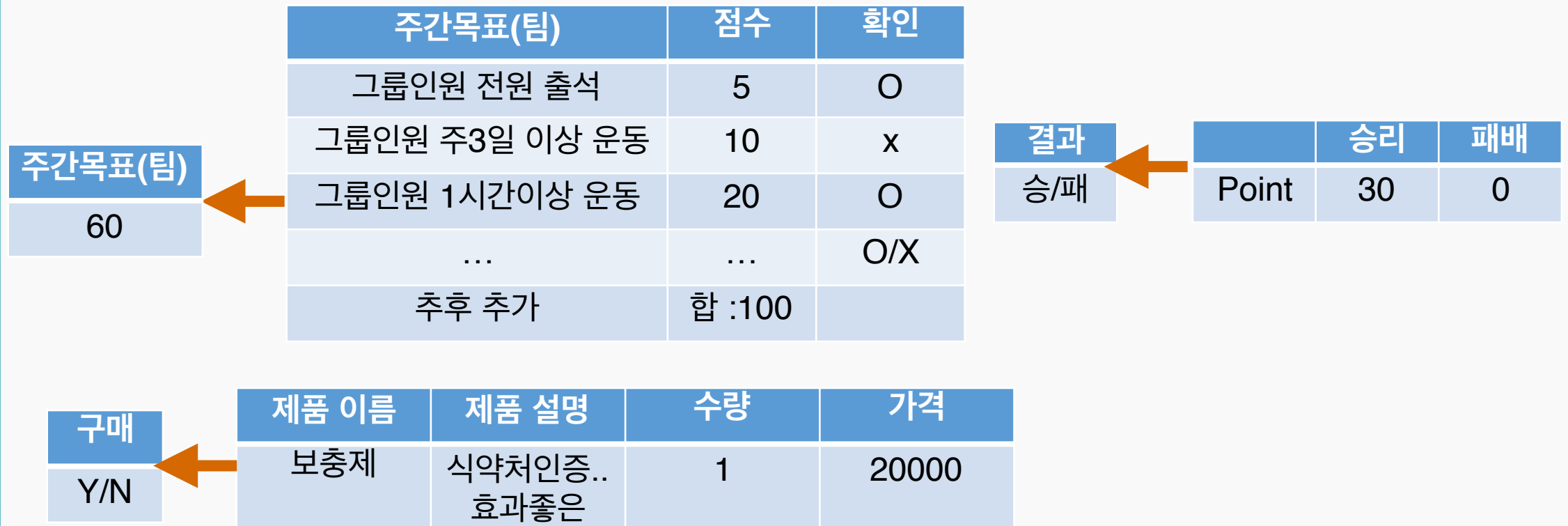
### 3. Dieata Data-base의 기능. - 테이블 구성도 (세부)

ID	이름	사는곳	e - mail	등급	포인트	그룹	포인트
casuello	최진용	성남시 중원구 성남동	<a href="mailto:absd@gamil.com">absd@gamil.com</a>	Gold	90	Small	90

		그룹이름	그룹	그룹인원	출석	운동 분류	운동 종목	주간목표(팀)	매칭	결과
<div>그룹</div> <div>HUFS</div>	HUFS	HUFS	Small	10명 이내	인원별 출석 인원별	다이어트			그룹간 매칭	승/패
						헬스				
						건강				
	HUFS	DUFS	Medium	11-20명	인원별 출석 인원별	다이어트			그룹간 매칭	승/패
						헬스				
						건강				
	HUFS	SCFS	Large	21-30명	인원별 출석 인원별	다이어트			그룹간 매칭	승/패
						헬스				
						건강				

### 3. Dieata Data-base의 기능. - 테이블 구성도 (세부)

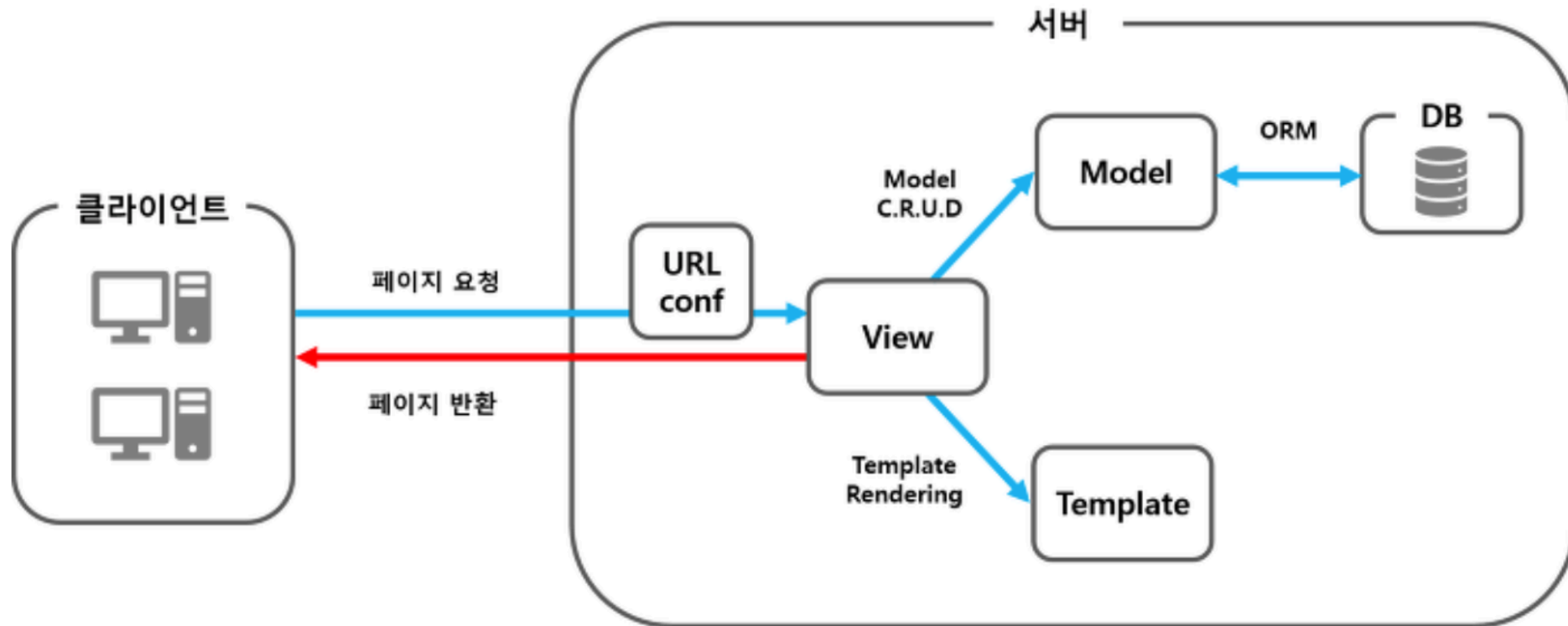
ID	이름	사는곳	e - mail	등급	포인트	그룹	포인트	구매
casuello	최진용	성남시 중원구 성남동	<a href="mailto:absd@gamil.com">absd@gamil.com</a>	Gold	90	Small	90	Y/N



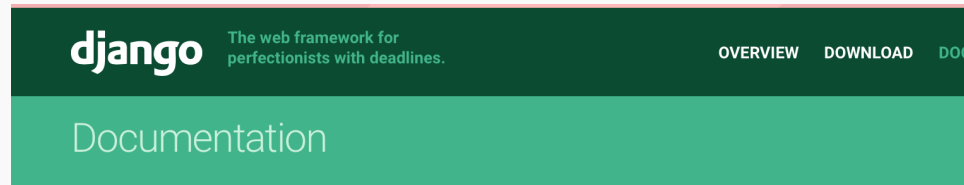


## 4. Django 과정.

- 장고의 데이터 흐름.



## 4. Django 과정. - Django 회원가입 기능.



### Source code for django.contrib.auth.forms

```
from __future__ import unicode_literals

from collections import OrderedDict

from django import forms
from django.contrib.auth import authenticate, get_user_model
from django.contrib.auth.hashers import (
    UNUSABLE_PASSWORD_PREFIX, identify_hasher,
)
from django.contrib.auth.models import User
from django.contrib.auth.tokens import default_token_generator
from django.contrib.sites.shortcuts import get_current_site
from django.core.mail import EmailMultiAlternatives
from django.forms.utils import flatatt
from django.template import loader
from django.utils.encoding import force_bytes
from django.utils.html import format_html, format_html_join
from django.utils.http import urlsafe_base64_encode
from django.utils.safestring import mark_safe
from django.utils.text import capfirst
from django.utils.translation import ugettext, ugettext_lazy as _
```

- 회원가입, 코드&결과 화면

- django.contrib.auth. 를사용하고 앞쪽에서 본 표대로 수정.

[My Homepage](#)

어서오세요! [로그인](#) [회원가입](#)

### 회원가입

사용자 이름:	<input type="text"/>
비밀번호:	<input type="password"/>
비밀번호 확인:	<input type="password"/>
성:	<input type="text"/>
이름:	<input type="text"/>
이메일 주소:	<input type="text"/>

[회원가입](#)

## 4. Django 과정.

- 테이블 생성
- MVC(model-View-controller)
- model.py 편집

```
from django.db import models

class Post(models.Model):
    title = models.CharField(max_length=100) # 길이 제한이 있는 문자열
    content = models.TextField() # 길이 제한이 없는 문자열
    created_at = models.DateTimeField(auto_now_add=True) # 해당 레코드 생성시 현재 시간 자동저장
    updated_at = models.DateTimeField(auto_now=True) # 해당 레코드 갱신시 현재 시간 자동저장
    # DB에서는 길이제한 유무에 따라서 문자열 필드타입이 다르다.
    # 길이 제한이 없는 문자열을 많이 쓰면 성능이 좋지 않다.
```

## 4. Django 과정. - API 호출

```
curl -v -X GET "https://dapi.kakao.com/v2/search/blog" \
--data-urlencode "query=https://brunch.co.kr/@tourism 집짓기" \
-H "Authorization: KakaoAK kkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkk"
```

우선, 헤더와 파라미터를 사전형으로 아래와 같이 만든다.

```
headers = {
    'Content-Type': 'application/json; charset=utf-8',
    'Authorization': 'KakaoAK 자기가 생성한 앱의 앱을 여기에 넣는다'
}

params = {
    'sort': 'accuracy',
    'size': 50,
    'query': keyword
}
```

- 요청시 json응답.

```
result = requests.get("https://dapi.kakao.com/v2/search/blog", headers=headers, params=params)
```

## - 헤더에서 편집

Kakao

Developers

API 소개

개발가이드

개발자포럼

기술블로그

Search

정진원

내 애플리케이션

DIETA

개요

통계

API

실시간

활동

앱로그 분석

설정

일반

사용자 관리

카카오링크

음성

메시지 템플릿 v2

푸시 알림

팀 관리

고급

APP

DIETA

애플리케이션(앱) ID : 303096

개요

카카오 로그인

사용자 관리

사용 불가 - 카카오 로그인을 사용하기 위해서는 사용자 관리를 설정해야 합니다.

앱 정보

설정

앱키

앱키 표시

네이티브 앱키

ae3794d6e66bab06425a155c65061bf9

REST API 키

68f3c715616996e194d0b6394a842565

JavaScript 키

974026aa5c7236583b454e919b8468f7

Admin 키

118f82dc988c3a54e151d114c86b62f7

설정된 플랫폼

## 5. 향후 계획.

### CONTENTS. 1

좀 더 효율적인 DB의 관리와 세부기능을 더 추가해야 할 것.

### CONTENTS. 2

원활한 DB처리 속도를 위해 속도및 품질 개선.