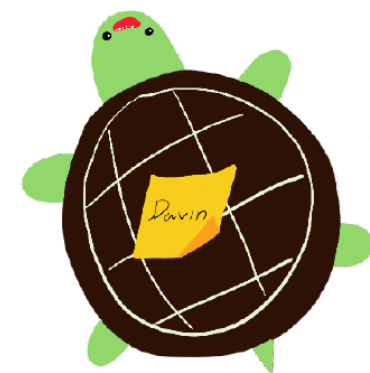




와플스튜디오 Backend Seminar[2]

변다빈 @davin111



2020.09.12.(토) 13:00 - Zoom

Results of Assignment 1

평가 요소와 피드백

- Django model 및 migration을 이해해 올바른 과정과 내용으로 migrate 했는가?
- CSRF 방어와 Django auth system을 이해한 채 회원가입, 로그인, 로그아웃 등을 수행하고 개발했는가?
- 기본적인 API 스펙을 만족하는 로그아웃, 자신 정보 조회, 설문조사 참여 API를 개발했는가?
- 불필요한 query 또는 코드 구조 없이 전반적인 개발이 이뤄졌는가?
- 전반적인 과제 내용뿐 아니라 directory 구조 등 제시 내용을 정확히 지켰는가?

Contents

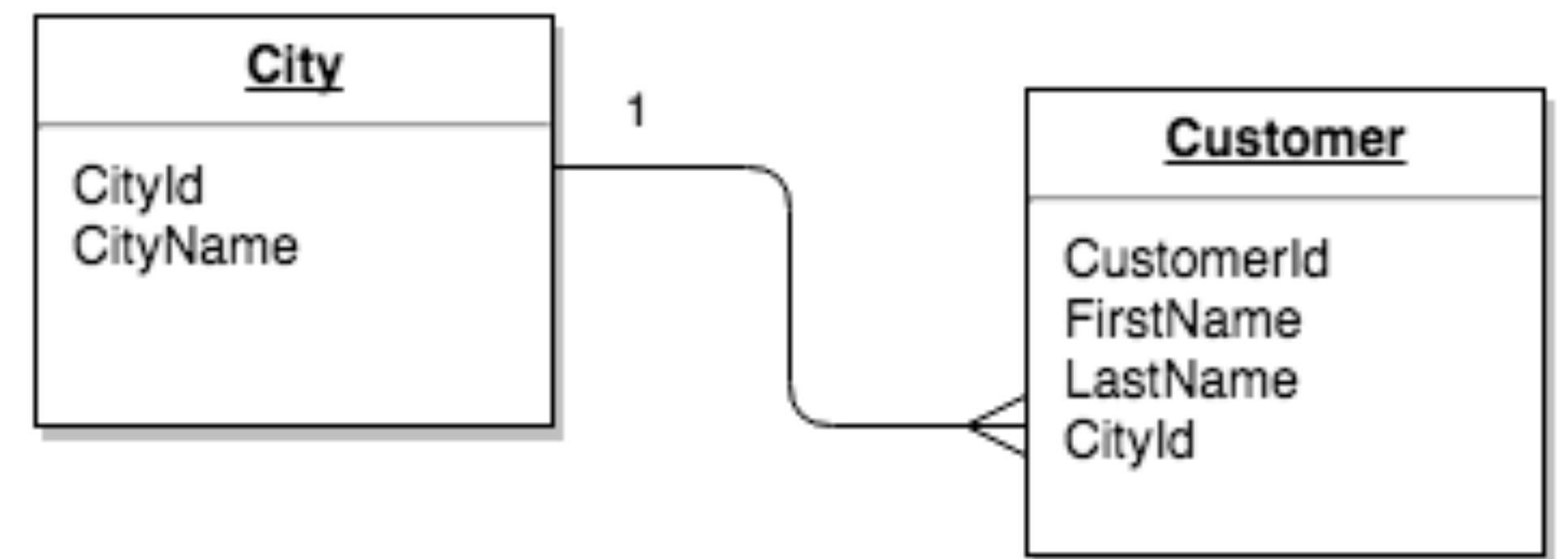
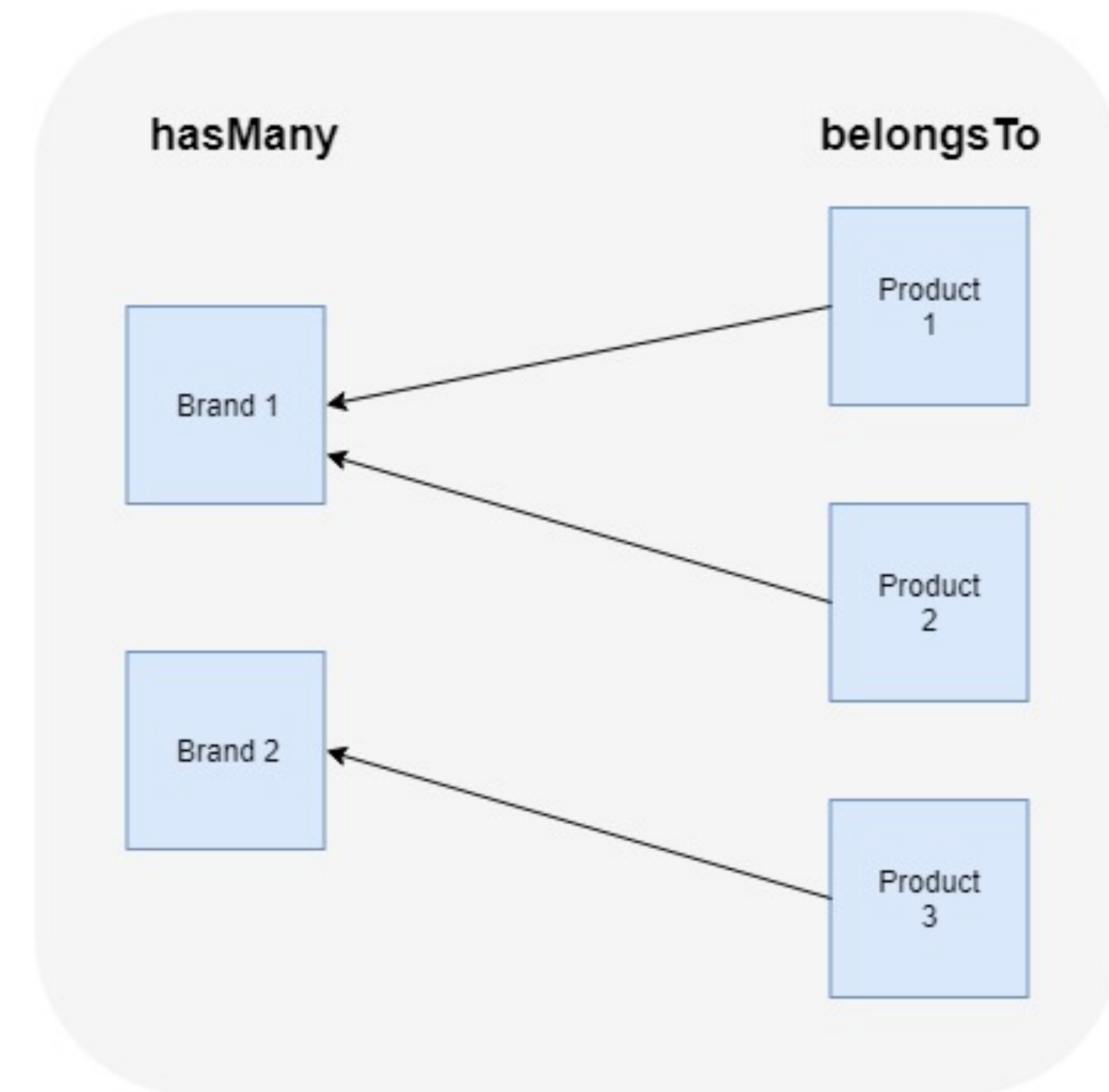
오늘 이야기할 주제들

- RDB의 database 설계의 기초
 - N-to-M relationship
 - Join
- Django와 DRF 좀 더 들어가기
 - Model, again
 - QuerySet, again
 - User Authentication System

N-to-M relationship

one-to-many or many-to-one

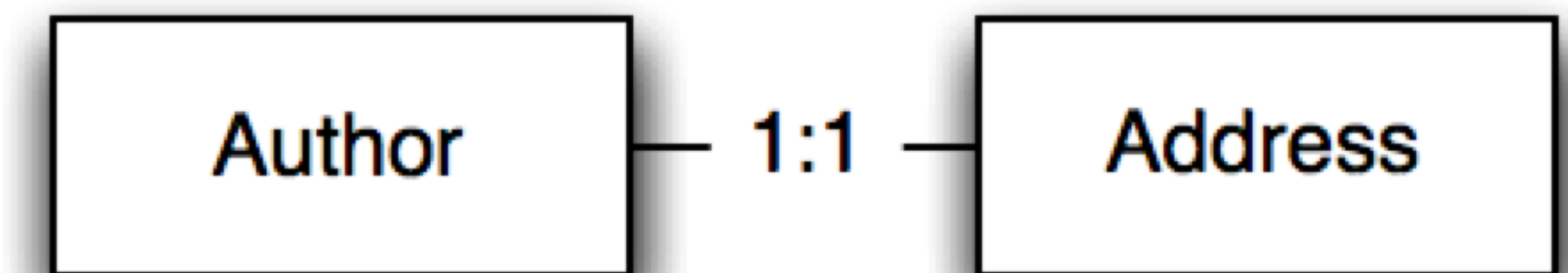
- 무엇이 어디에 속하는가?
- 어느 쪽에 ForeignKey를 달 것인가?
 - Reporters & Articles
 - Books & Categories
 - Survey results & OSs
 - Action logs & Users
- Django의 related_name, related_query_name



N-to-M relationship

one-to-one

- Django의 ForeignKey(unique=True)와 무엇이 다른가?
 - 항상 reverse side에 접근하는 경우를 고려하자



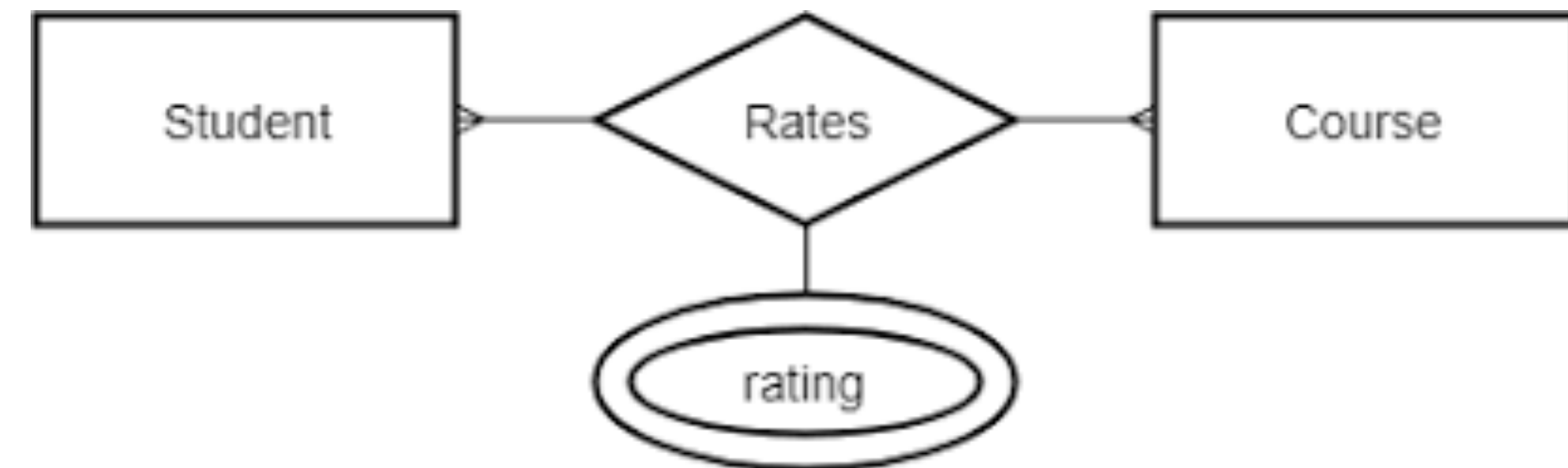
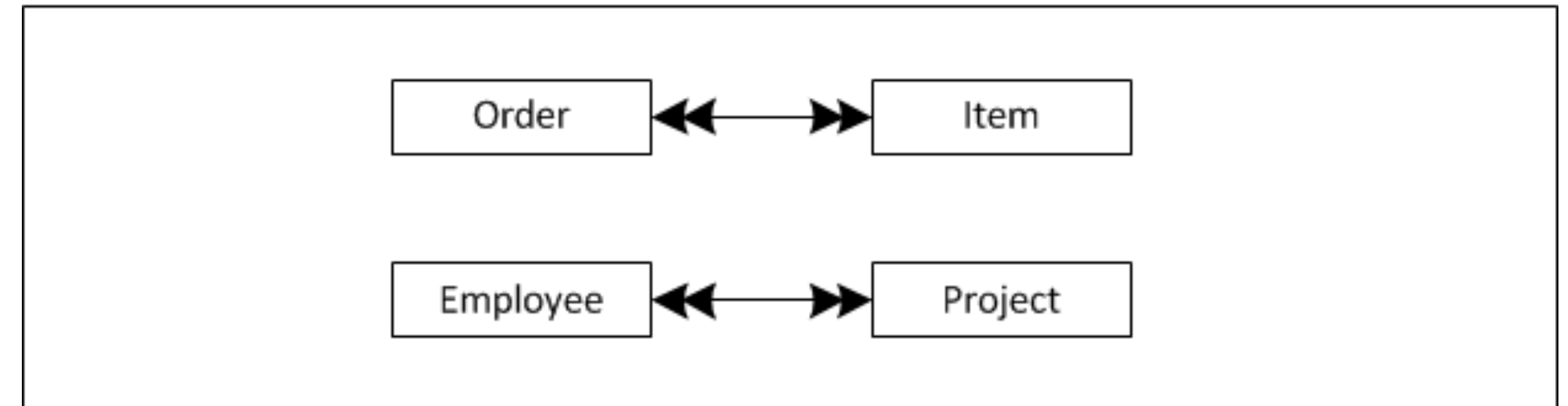
People		Drivers Licenses	
ID	Name	UserID	Number
1	Alice	1	F25532
2	Bob	2	S43212
3	Cathy	3	B98364

One-to-One

N-to-M relationship

many-to-many

- 어떤 관계에 해당하는가?
 - Pizzas & Toppings
 - Students & Courses
- ORM에서 지원하는 ManyToMany?
 - mapping table 직접 정의하기
 - 관계 속의 정보 관리



Do not just write codes!

- Don't write codes!

설계는 머리를 먼저 쓰기
일단 온갖 경우를 고민하기
의사소통 성실히 하기
래에게 항상 생각보다 더 구체적이고 친절하기
확장을 고려하기
하기

Join

RDB의 필요악

- join의 기본 유형들
 - inner join
 - natural join
 - left (outer) join
 - right (outer) join

course

<i>course_id</i>	<i>title</i>	<i>dept_name</i>	<i>credits</i>
BIO-301	Genetics	Biology	4
CS-190	Game Design	Comp. Sci.	4
CS-315	Robotics	Comp. Sci.	3

prereq

<i>course_id</i>	<i>prereq_id</i>
BIO-301	BIO-101
CS-190	CS-101
CS-347	CS-101

■ *course* **natural join** *prereq*

<i>course_id</i>	<i>title</i>	<i>dept_name</i>	<i>credits</i>	<i>prere_id</i>	<i>course_id</i>
BIO-301	Genetics	Biology	4	BIO-101	BIO-301
CS-190	Game Design	Comp. Sci.	4	CS-101	CS-190

Note: prereq information missing for CS-315 and
course information missing for CS-437.

Join

특히 주의할 점

- join은 성능상 무거운 연산
- 멍청한 natural join을 믿지 말 것
- 의도치 않게 중복 생성되는 row 주의
 - left outer join 등도 해당됨

가려 테이블 ①			불러온 테이블 ②			결과		
A	10		A	Apple		A	10	Apple
B	20		C	Code		B	20	Null
C	30		D	Denth		C	30	Code
D	40		D	Dos		D	40	Denth
D	50					D	50	Denth
						D	40	Dos
						D	50	Dos

↑ select * ①, ②, ③
from 가려 테이블
left outer join 불러온 테이블
on 가. ① = 불러. ①

의도치 않은 뺑뚱기

Django Model

사실 이 정도는 되어야 model 정의의 기본

- 그 'objects'가 뭐길래? - Manager

- **index**

- `__str__()` 또는 `__repr__()`
- class Meta:
 - `index_together`
 - `unique_together`

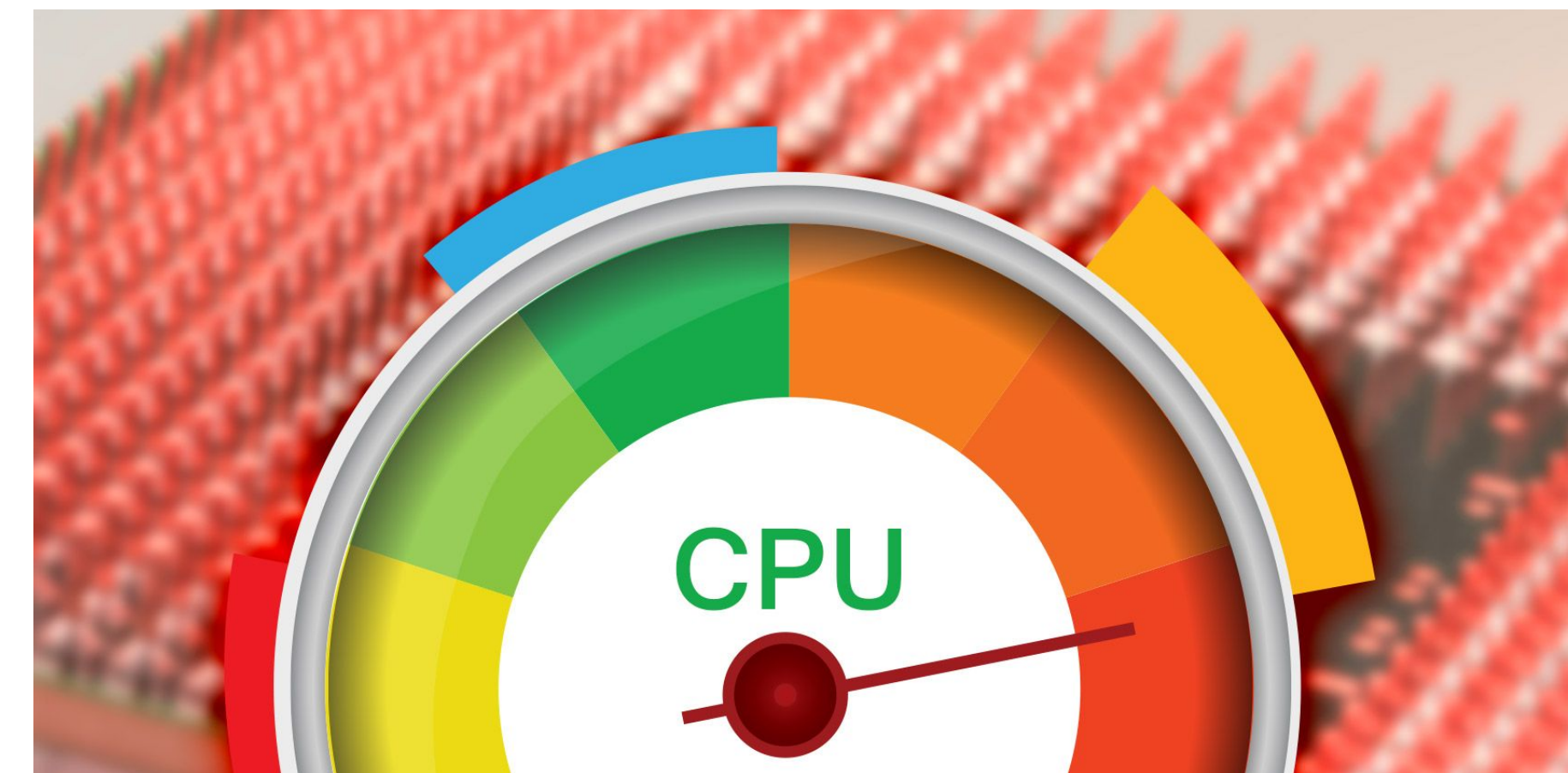


Django Reinhardt (1910~1953)

Index

모르면 반드시 터진다

- To Be Continued...



Django QuerySet

보고 또 봐도 전혀 아깝지 않은 Django의 핵심

- QuerySet의 method들
 - 다른 table과 join해 filter()하기
 - QuerySet의 update()와 Model object의 save()
 - first(), last()
 - values(), values_list()
- **Lazy evaluation**
- **QuerySet의 caching**

A diagram illustrating the Django ORM architecture. It features a stack of four horizontal layers, each with a label on the right side. The layers, from top to bottom, are: 'Model / Fields' (lightest green), 'Managers' (medium green), 'QuerySets' (darker green), and 'Backends' (darkest green). The text 'Django ORM' is prominently displayed in a large, dark green font across the middle of the stack.

Django ORM

Model / Fields

Managers

QuerySets

Backends

User

인증에 대해 더 살펴보기

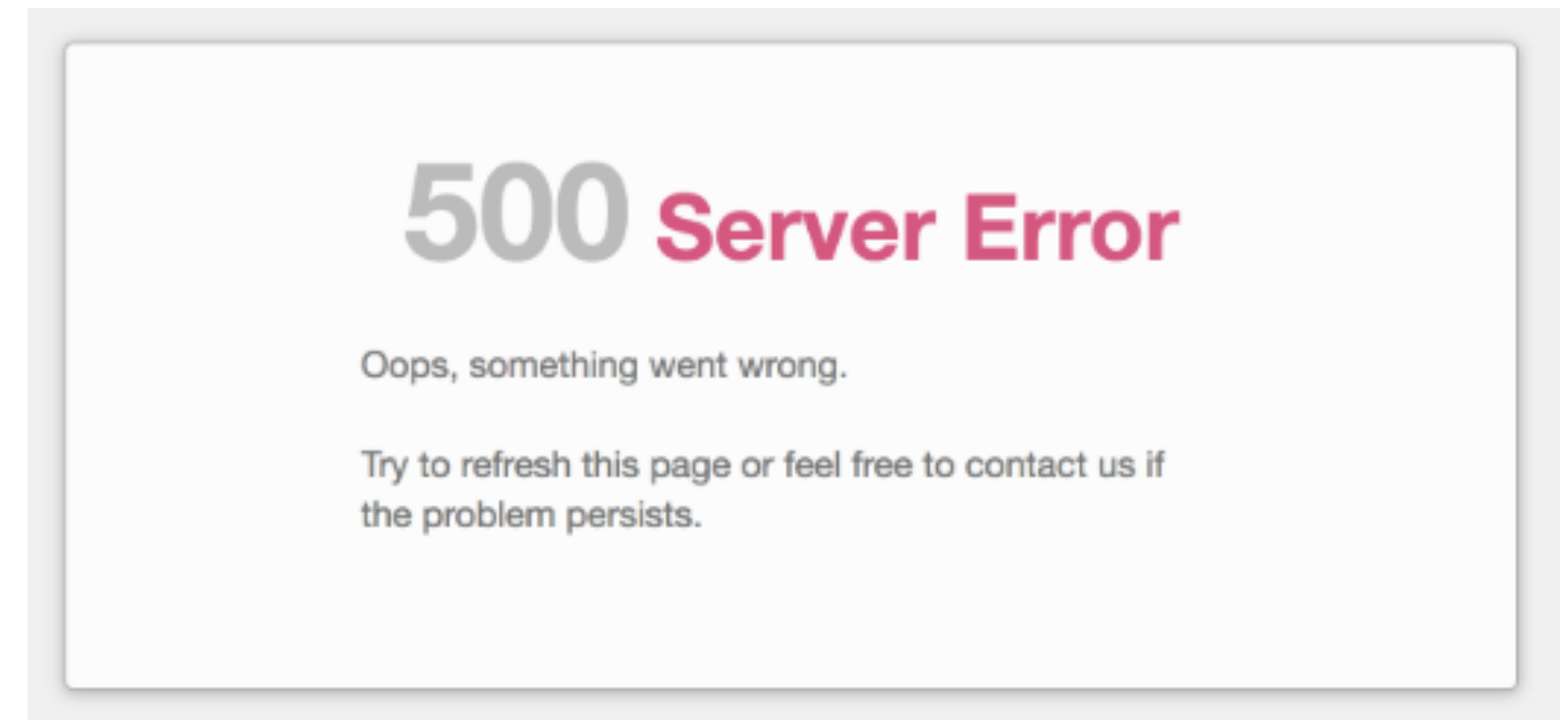


- CSRF 방어와는 무관. 그건 로그인 인증만으로 request들을 다 믿을 수 없다는 얘기.
- Session 기반 인증
 - 벌써 몇 번 간접적으로 접한 Middleware부터 알고 가기
 - SessionMiddleware와 AuthenticationMiddleware의 합작
- Token 기반 인증 - mobile client를 위해서 특히 필요
- DRF의 TokenAuthentication

500을 피하자 (revisited)

INTERNAL SERVER ERROR

- logic상 status code 500이 발생할 수 있는 여지는 최대한 없애기
- 모든 프로그래밍이 그렇지만, 서버 프로그래밍은 염려와 강박을 탑재할 것
- 안정적인 서버 자체가 서비스의 가치
- 유저가 서버의 존재를 느낄 일이 없어야
- 문제될 상황에 대해선 합리적인 처리가 중요
 - 예상되는 원인을 코드상에 드러나게 짚어서 대응
 - 문제의 책임과 원인을 찾고 외부에 알리는 프로그래밍



Cloud Computing 서비스 (if time allows, again)

Amazon Web Services revisited

- 왜 돈 주고 남의 컴퓨터를 쓸까?
 - 내 컴의 성능은 어느 정도?
 - 내 컴이 꺼지면?
 - 모니터링 부탁할 수 있을까?
 - 관리를 부탁할 수 있을까? (재부팅하기, 갑자기 늘리기, 줄이기)



kubernetes



Google Cloud Platform

실제 동작 확인하기 AWS EC2, RDS

Any Questions?

또는 세미나장이 하고 싶은 말, 못다한 말