

멋쟁이사자처럼 심화 스터디 가이드라인

4주차

LIKE LION

01

4주차 목표

Active Record - Model

Migration

Database - Table

01

4주차 목표



이제는 Rails Console 이다.

- 콘솔의 위대함을 모르는 당신이 불쌍해..



\$ rails g model

\$ rails g migration

- g 의 의지는 계속된다

LIKE LION

02

Ruby Gem John 잼



Gem의 위대함을 모르는 그대에게

- Ruby Gem John 잼 에서 소개된 “**pry**”, “**awesome_print**”, “**faker**”

LIKE LION

02

Ruby Gem John 잼

pry

<https://rubygems.org/gems/pry>

<https://github.com/pry/pry>

pry 는 irb 와 rails console 의 업그레이드 버전!

rails 에서 사용하고 싶으면 Gemfile에

gem 'pry-rails' 를 넣고 \$ bundle 을 입력하세요!

그리고 터미널에서 \$ rails c 를 입력하면!

pry(main) 으로 나옵니다!

02

Ruby Gem John 잼

awesome_print

https://rubygems.org/gems/awesome_print

https://github.com/awesome-print/awesome_print

awesome_print 는 irb 와 rails console 에서 출력되는 화면을

획어어얼씬 어썸하게 바꿔주는 gem!

Gemlist에 gem 'awesome_print' => \$ bundle

LIKE LION

02

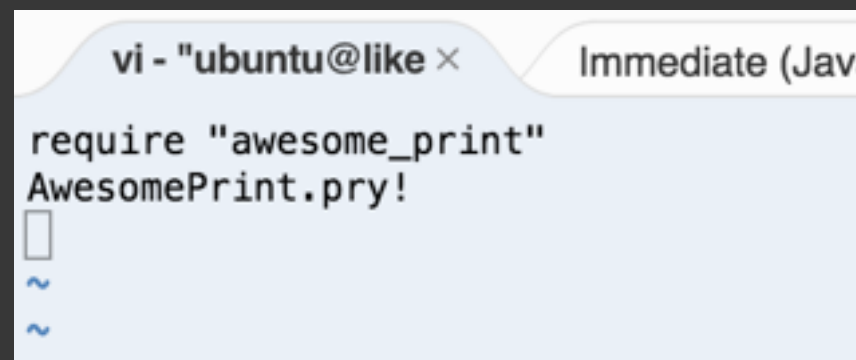
Ruby Gem John 잼

awesome_print

https://rubygems.org/gems/awesome_print

https://github.com/awesome-print/awesome_print

그리고 \$ vi ~/.pryrc

A screenshot of a terminal window showing a vi editor session. The window title is "vi - 'ubuntu@like x". The editor content shows the following lines: "require 'awesome_print'", "AwesomePrint.pry!", and a cursor on the third line. Below the cursor are two tilde symbols (~) indicating the end of the file. The window also shows a tab labeled "Immediate (Java)".

```
vi - "ubuntu@like x    Immediate (Java
require "awesome_print"
AwesomePrint.pry!
~
~
```

텅텅 빈 파일에 이렇게 작성하고 저장! (vi 기초를 배워볼 시간..)

이제부터 \$ rails c 가 즐거워질 일들만 남았다!

LIKE LION

02

Ruby Gem John 잼

faker

<https://rubygems.org/gems/faker>

<https://github.com/stympey/faker>

faker 는.. 직접 사용해 보면 바로 감이 옵니다!

Gemlist에 gem 'awesome_print' => \$ bundle
그리고 \$rails c 로 콘솔을 열고,

02

Ruby Gem John 잼

faker

```
ruby - "likelionty" × Immediate × ruby - "ubuntu@l" × +
Using turbolinks 5.0.1
[10] pry(main)> Faker::Name.name
"Molly Stiedemann"
[11] pry(main)> Faker::Internet.email
"jovani@reichelstrosin.info"
[12] pry(main)> Faker::Lorem.paragraph
"Aut nihil sequi ratione ipsa. Autem dolor ut est harum tempore. Est et harum quas consequatur architecto. Consectetur qui accusantium explicabo nostrum."
[13] pry(main)> Faker::Lorem.paragraph(10)
"Eligendi repellendus atque et facilis ullam. Qui asperiores eligendi. Possimus omnis ut ipsa sapiente quia. Labore ducimus pariatur expedita facere rerum. Dol or sed debitis vel ratione eius in. Similique cupiditate sunt qui consequatur possimus suscipit ea. Optio sed est sit quo dolor. Accusantium laudantium omnis. Ipsa saepe facere eius impedit autem dolore enim. Quidem voluptates aut et."
[14] pry(main)> Faker::PhoneNumber.cell_phone
"856.125.8395"
[15] pry(main)> Faker::Book.title
"Lilies of the Field"
[16] pry(main)> Faker::Address.street_address
"6673 Quitzon Pine"
[17] pry(main)> Faker::Pokemon.name
"Weedle"
[18] pry(main)> Faker::Company.name
"Ankunding LLC"
[19] pry(main)> Faker::Company.logo
"https://pigment.github.io/fake-logos/logos/medium/color/10.png"
[20] pry(main)> █
```

02

Ruby Gem John 잼

faker

<https://rubygems.org/gems/faker>

<https://github.com/stympey/faker>

더 이상의 더미 데이터 고민은 없다!

더 알아보고 싶으면?

위의 github 주소로 들어가서 찾아보세요!

LIKE LION

03

Ruby on Rails

한 층 업그레이드 된 콘솔로 User 모델을 만들어 보자!

```
$ rails g model User
```

03

Ruby on Rails

우리의 **User** 는
name과 mail 을 가지고 있다!

db/migrate/[timestamp]_create_users.rb 에

t.string :name

t.string :mail

이 정도는 다 알고 계시죠?

LIKE LION

03

Ruby on Rails

\$ rake db:migrate

(rails -v >= 5.0) 은 rake 를 모두 rails로 대체 가능!

\$ rails db:migrate 도 가능하다는 것!

자 이제 \$ rails c 로 콘솔을 열고!

새로운 **User** (객체)를 만들어 봅시다!

이번 주에는 View/Controller 전혀 없이
rails console 로만 작업합니다!

이름과 메일은 **Faker**를 통해 생성해 봅시다.

03

Ruby on Rails

지금까지 우리 DB에서 **User** Table의 모습

User				
id	name	email	created_at	updated_at
1	lion	<u>like@lion.org</u>	↑애랑	↑애는
2	tiger	<u>like@tiger.org</u>	이제부터	안쓸거예요
3	dog	<u>like@dog.org</u>	왜냐하면	자동생성
4	cat	<u>like@cat.org</u>	되기	때문이죠!
...		

LIKE LION

03

Ruby on Rails

\$ rails db 콘솔로도 볼 수 있어요!

굳이 보고싶다면 말이에요.. 명령어는 다음장에!

```
likelionty:~/workspace $ rails db
SQLite version 3.8.2 2013-12-06 14:53:30
Enter ".help" for instructions
Enter SQL statements terminated with a ";"
sqlite> .headers on
sqlite> .mode columns
sqlite> select * from users;
```

우리 User 테이블에 있는
Data 들이에요!

id	name	mail	created_at	updated_at
1	Lion	like@lion.org	2017-02-01 10:32:28.773213	2017-02-01 10:32:28.773213
2	Tiger	like@tiger.or	2017-02-01 10:32:28.787131	2017-02-01 10:32:28.787131
3	Dog	like@dog.org	2017-02-01 10:32:28.796037	2017-02-01 10:32:28.796037
4	Cat	like@cat.org	2017-02-01 10:32:28.805865	2017-02-01 10:32:28.805865

```
sqlite> .exit
likelionty:~/workspace $ █
```

03

Ruby on Rails

\$ rails db 명령어로도 볼 수 있어요!

잘 따라치셔야 해요.. 틀리면 카오스에 빠지게 되요.. 여러분 잘못이 아니라 db콘솔 잘못이에요..

레일즈 DB 콘솔을 켭니다. ⇒ `$ rails db`

결과 값을 이쁘게 보이도록 -1. ⇒ `sqlite> .headers on`

결과 값을 이쁘게 보이도록 -2. ⇒ `sqlite> .mode columns`

user 에 있는 모든 값들을 보여줘! ⇒ `sqlite> select * from users;`

결과 값 ⇒ 결과 값

DB 콘솔 종료! ⇒ `sqlite> .exit`

03

Ruby on Rails db/schema.rb

이 파일을 보면 현재 지금까지 rake db:migrate를 통해
반영한 모든 마이그레이션들의 최종 결과물을 볼 수 있어요

```
ActiveRecord::Schema.define(version: 20170201113242) do

  create_table "users", force: :cascade do |t|
    t.string "name"
    t.string "mail"
    t.datetime "created_at", null: false
    t.datetime "updated_at", null: false
  end

end
```

03

Ruby on Rails

그런데.. 아차!

생각해 보니 우리의 **User** 모델은 나이(**age**)도 있어야 해!

어떻게 하지..? 지금은 이름과 메일만 받았는데..

03

Ruby on Rails

지금까지라면..

db/migrate/[timestamp]_create_users.rb 에
t.integer :age 를 추가하고,

LIKE LION

03

Ruby on Rails

`$ rake db:drop / $ rails db:drop(>5.0)`

그래.. 마이그레이션 파일을 직접 수정했으면

무.조.건 db:drop 이지

그런데 만약.. 실제로 중요한 데이터를 많이 저장해 뒀으면.. 어찌지?

다 날아가잖아..

03

Ruby on Rails

`$ rake db:seed / $ rails db:seed(>5.0)`

그럴 때를 대비해서!

db/seeds.rb에 코드를 작성해 놓는거 아니겠어?!

03

Ruby on Rails

여기까지가

멋사 4기 강의에서 가르쳐준 내용

하지만

정말 이게 최선일까?

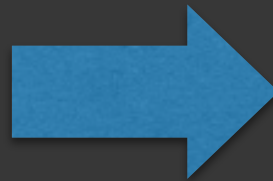
LIKE LION

03

Ruby on Rails

그냥 새로운 컬럼(column)을
추가해 주면 되잖아!

User		
id	name	email
1	lion	<u>like@lion.org</u>
2	tiger	<u>like@tiger.org</u>
...@...



User			
id	name	email	age
1	lion	<u>like@lion.org</u>	
2	tiger	<u>like@tiger.org</u>	
...@...	

LIKE LION

03

Ruby on Rails

오 좋다! 근데.. 어떻게?

`t.integer :age`

이 한 줄을 마이그레이션 파일에 직접 추가하는거 말고
다른 방법이 있나?

LIKE LION

03

Ruby on Rails

\$ rails g migration AddAgeToUser age:integer

upper camel case는 꼭 지켜야 합니다! upper camel case가 뭔지 모르겠으면 구글링!

뭔지 잘은 모르겠지만..

마이그레이션(migration)이라는 걸 생성했어!

그리고 이름이 AddAgeToUser 구나. 해석해 보면,

Age를 User에 추가하고, age는 정수(integer)다 라는 뜻이군..

근데 Age는 추가하려는 컬럼 이름이고

User는 우리 모델 이름이잖아?!

뭐가 바뀌었지?

db/migrate/[타임스탬프]_add_age_to_user.rb 라는 파일이 생겼다!

그리고 작성한 적 없는 코드도!

```
1 class AddAgeToUser < ActiveRecord::Migration
2   def change
3     add_column :users, :age, :integer
4   end
5 end
6
```

03

Ruby on Rails

다시 \$ rake db: **migrate**

과연 우리가 원하던 것처럼 되었을까..?

03

Ruby on Rails

\$ rails db

```
likelionty:~/workspace $ rails db
SQLite version 3.8.2 2013-12-06 14:53:30
Enter ".help" for instructions
Enter SQL statements terminated with a ";"
```

```
sqlite> select * from users;
1|Lion|like@lion.org|2017-02-01 10:32:28.773213|2017-02-01 10:32:28.773213|
2|Tiger|like@tiger.org|2017-02-01 10:32:28.787131|2017-02-01 10:32:28.787131|
3|Dog|like@dog.org|2017-02-01 10:32:28.796037|2017-02-01 10:32:28.796037|
4|Cat|like@cat.org|2017-02-01 10:32:28.805865|2017-02-01 10:32:28.805865|
```

```
sqlite> .headers on
sqlite> .mode columns
```

```
sqlite> select * from users;
id          name      mail          created_at          updated_at
-----
1          Lion     like@lion.org 2017-02-01 10:32:28.773213 2017-02-01 10:32:28.773213
2          Tiger    like@tiger.or 2017-02-01 10:32:28.787131 2017-02-01 10:32:28.787131
3          Dog      like@dog.org  2017-02-01 10:32:28.796037 2017-02-01 10:32:28.796037
4          Cat      like@cat.org  2017-02-01 10:32:28.805865 2017-02-01 10:32:28.805865
```

```
sqlite> .exit
```

```
likelionty:~/workspace $ █
```

그냥 하면 구려요!

Good!

예썬!

이빠져라!

03

Ruby on Rails db/schema.rb

이 파일을 보면 현재 지금까지 rake db:migrate를 통해
반영한 모든 마이그레이션들의 최종 결과물을 볼 수 있어요

```
ActiveRecord::Schema.define(version: 20170201105724) do

  create_table "users", force: :cascade do |t|
    t.string "name"
    t.string "mail"
    t.datetime "created_at", null: false
    t.datetime "updated_at", null: false
    t.integer "age"
  end

end
```

Ruby on Rails

주의사항

\$ rails g migration 을 통해 생성된

모든 마이그레이션 파일이 위와 같이 자동적인 코드를 제공하지 않아요!

직접 작성해야 하는 경우가 더 많을거예요!

이번같은 경우는 똑똑한 Rails가 마이그레이션 이름(AddAgeToUser)을 통해

우리가 뭘 할지 미리 알아서 이렇게 된거예요!

(Upper Camel Case 도 이것 때문에 필수였어요!)

03

Ruby on Rails

참고자료

퍼펙트 루비 온 레일즈

2.4장

5.1 ~ 5.4 장

&

http://rubykr.github.io/rails_guides/migrations.html

LIKE LION

04

4주차 개인 과제



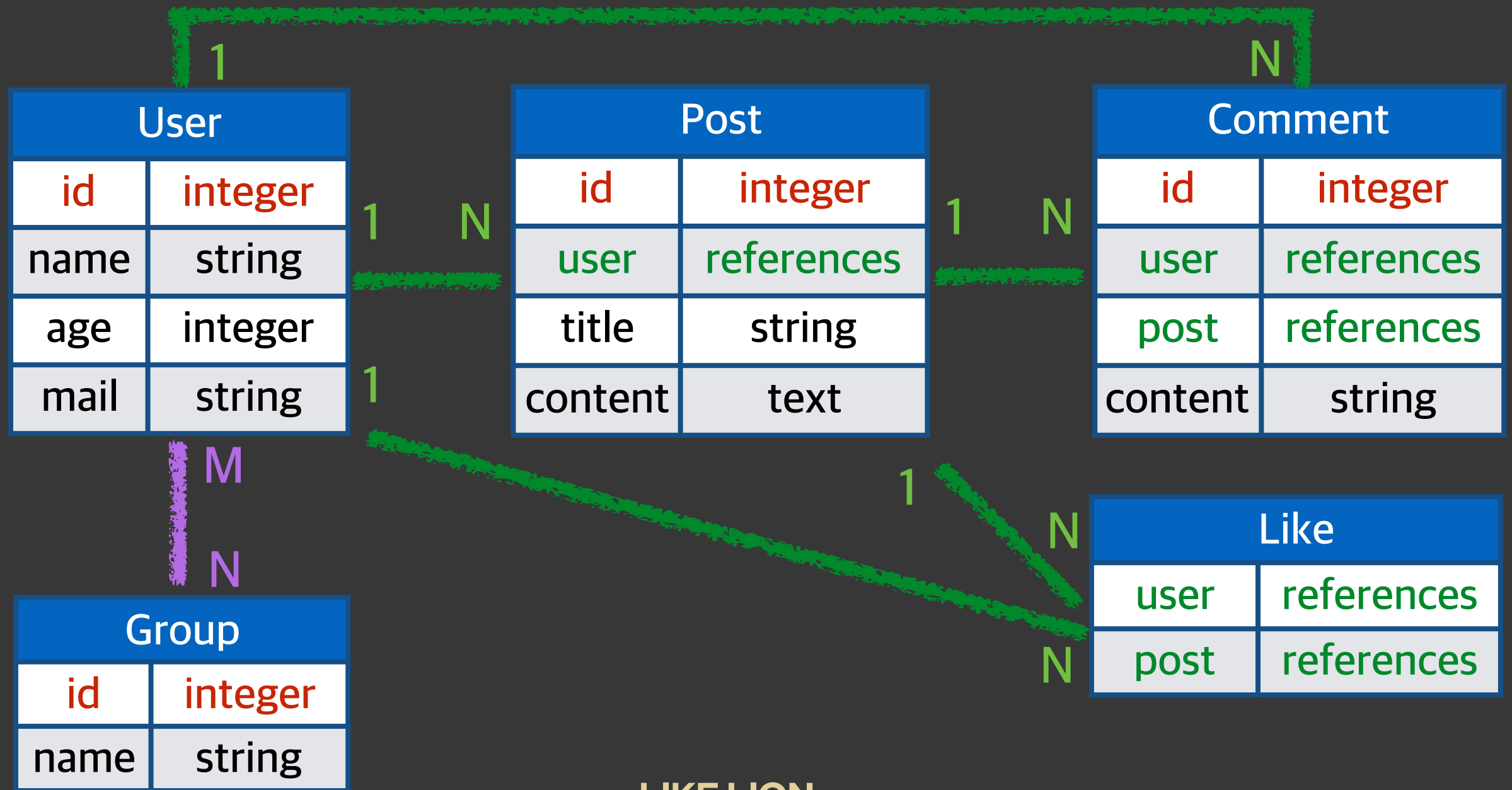
Basic 실제 게시판 모델링

LIKE LION

04

4주차 개인 과제

ERD(Entity Relation Diagram)



LIKE LION

04

4주차 개인 과제

가장 일반적인 게시판의 데이터 베이스 구조입니다!

위와 같은 구조로 이루어진
데이터 베이스 모델링을 해보자.

+

Faker 를 통해 더미 데이터를 삽입해 보자.

데이터의 구성

User

- 10명의 User가 존재합니다.
- name과 mail은 Faker로!
- 나이는 20~30대 5명, 40~50대 5명으로!

데이터의 구성

Group

- 2개의 Group이 존재합니다.
- Group들의 name은 young과 old
- 20~30대 들은 young Group에,
- 40~50대 들은 old Group에.

데이터의 구성

Post

- 1개의 Post가 존재합니다.
- 해당 Post는 User 중
user_id 가 5인 사람이 작성합니다.
- Post 의 title과 content는 자유롭게!(Faker)

데이터의 구성

Comment & Like

- 1개의 Post에
- Group name : young 의 User들은 Like를,
- Gruop name : old 의 User들은 Comment를 한 줄씩(자유롭게)!

데이터의 구성

제시해 드린 것들은 가장 기본적인 데이터들입니다.

훨씬 자유롭게 많이 데이터를 입력 하시면서
최대한 Console 로 많은 Data 를 생성하고 골라보세요!
 pry 와 awesome_print 를 즐기세요!

04

4주차 개인 과제

제출물

마지막 migration을 끝낸 이후
db/schema.rb 파일을 제출해 주세요!

05

4주차 학습 목표

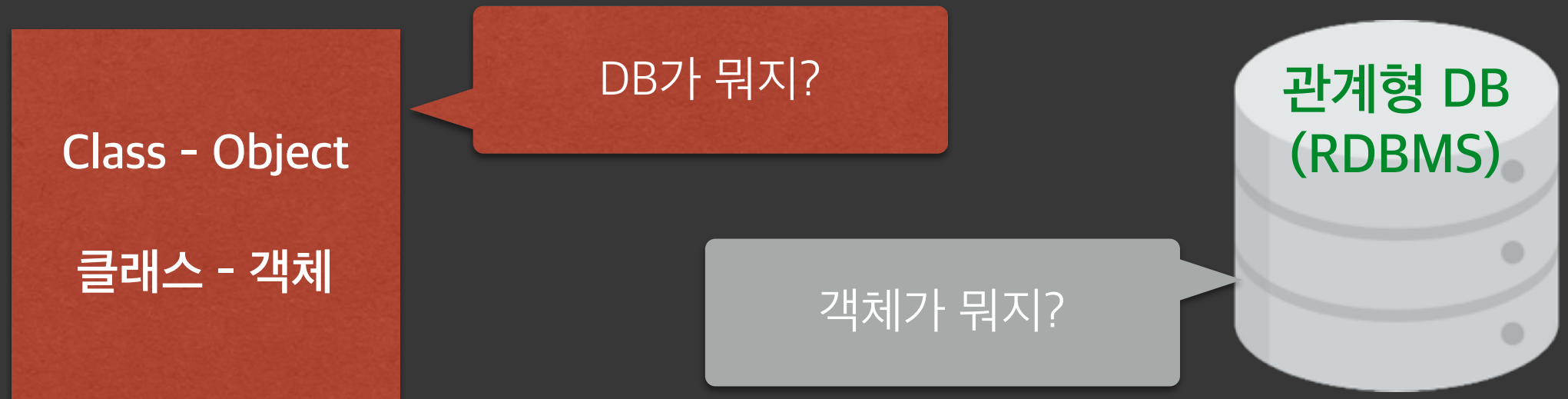
본격적으로 과제를 시작하기 전에..

나는 Model 을 얼마나 알고 있을까?

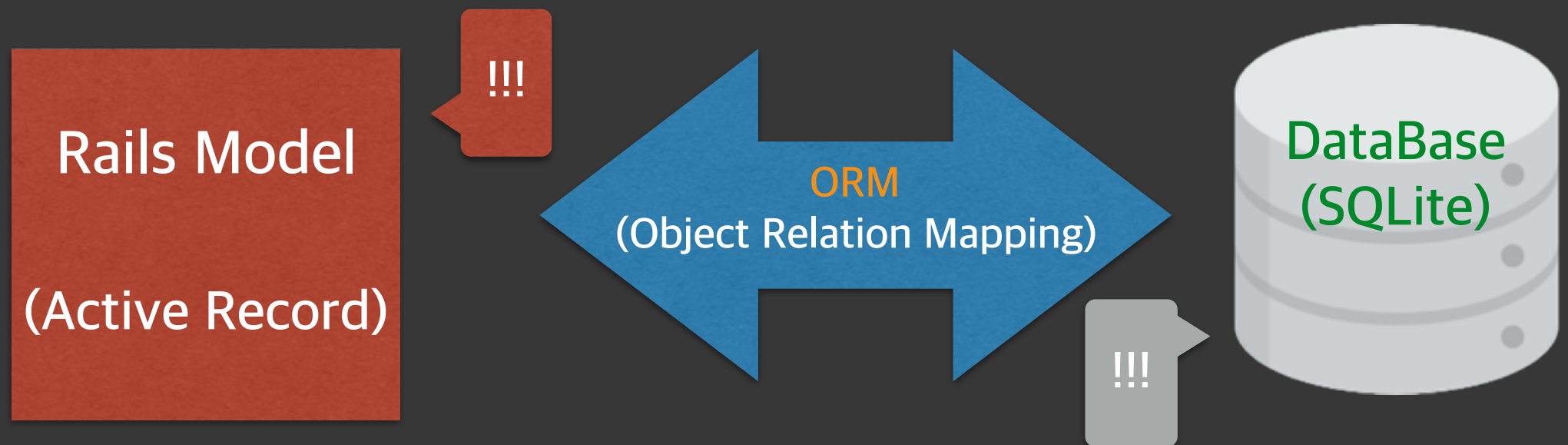
05

4주차 학습 목표

원래대로라면..



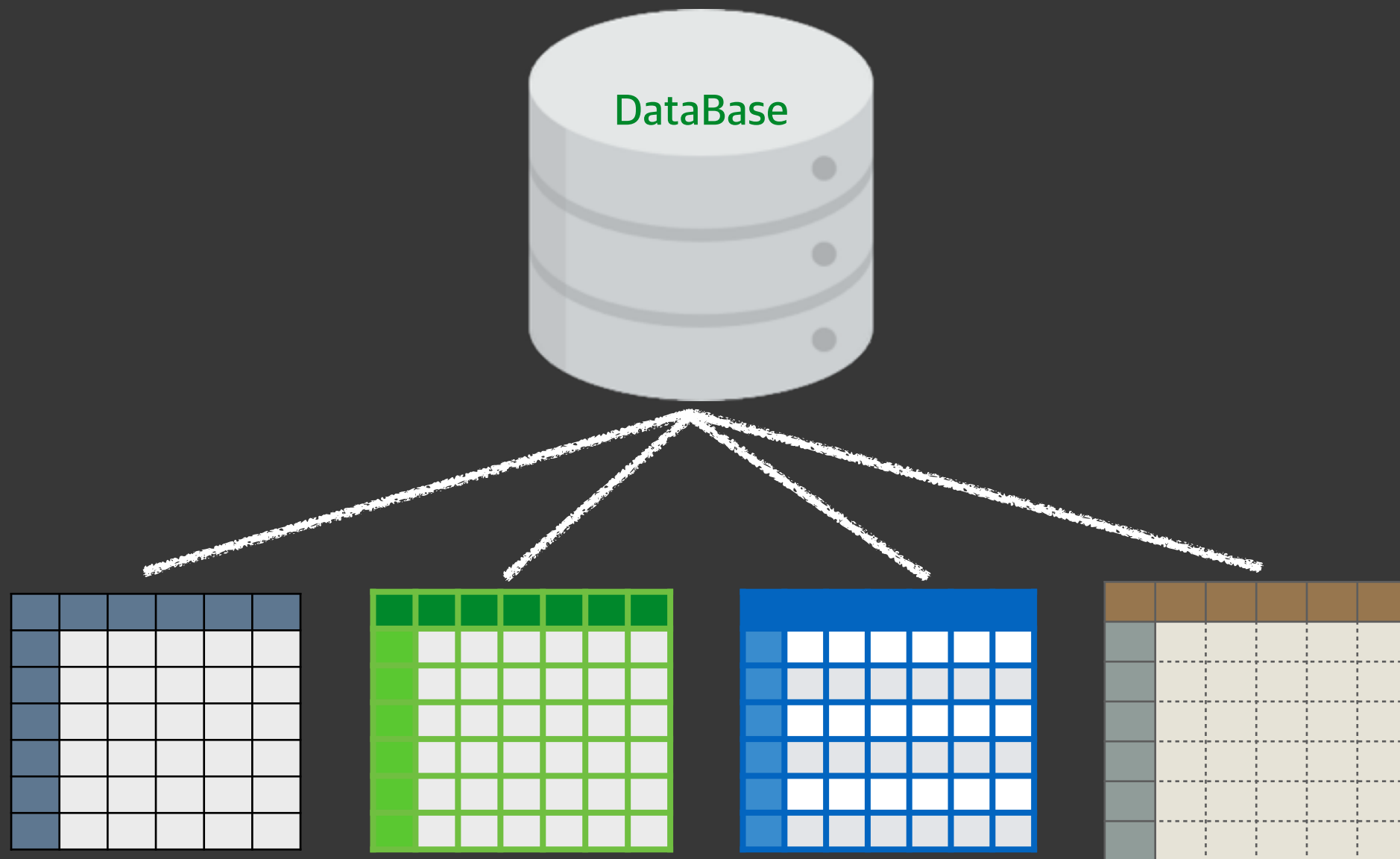
Rails 는..



LIKE LION

05

4주차 학습 목표



LIKE LION

05

4주차 학습 목표

이 명령어들은 뭘 만들어 주는거지?

\$ rails g model

\$ rails g migration

05

4주차 학습 목표

이 명령어들, 써 보기는 했는데 정확하게 뭐하는 거지?

```
$ rake db:migrate
```

```
$ rake db:seed
```

```
$ rake db:drop
```

05

4주차 학습 목표

이 파일들은 뭐지?

db/migrate/[마이그레이션_파일들]

app/model/[모델_클래스_파일들]

\$ rake db:migrate 이후 생성된 db/schema.rb

05

4주차 학습 목표

위의 그림들
다 이해 하시나요?

위의 질문들
모두 나름의 답을 내릴 수 있나요?

05

4주차 학습 목표

이번 과제는 위 질문들 + Rails 에서
Model 과 DB가 어떻게 상호작용하는지에 대한
나름의 답을 찾는것에
포커스를 맞추고 진행해 주세요!

05

4주차 학습 목표

추천 참고 자료

퍼펙트 루비 온 레일즈

2.4장

5.1 ~ 5.4 장

&

http://rubykr.github.io/rails_guides/migrations.html

Rails - 스터디 권장



위의 그림들과 질문들에 대한 나만의 설명과 대답을
스터디에서 공유하며, 과제에서 막힌 부분을 해결하고!
Model - DB 에 대한 이해와 설명자료를 만들어 보자.

4주차 스터디 정리

1. Ruby Gem John 잼
 - pry, awesome_print, faker
 - 설치하고, 사용하고, 즐겨라!
2. Model - ActiveRecord - Migration - DB - Table - Data

08

다음주 스포일러



LIKE LION

08

다음주 스포일러

Rails Model 의 모든것 part 2

아직까지 안나온 것

Association, Validation, CallBack, Scope, etc..

여러분이 먼저 책 보고 인터넷 보고 공부하셔도 됩니다..

웹툰작가의 심경을 느끼고 있습니다.. 늦었는데 분량조절도 실패라니..

죄송합니다ㅠㅠ

LIKE LION

09

정리

개인별로

2/8 (수) 까지

게시판 모델링
schema.rb 제출

모여서

스터디 하기

- Model
- ActiveRecord
- Migration
- DB(RDBMS)
- Table
- Data

정리해 보기

5기한테 어떻게
이 심오한
Model 의 세계를
설명할 수 있을까..
라는 고민을 담아
자료 제작 하기