

Django - 데이터베이스 설계

1. 목표

- 데이터를 생성, 조회, 삭제, 수정할 수 있는 Web Application 제작
- 데이터베이스 테이블간 관계 설정 (1:N)

2. 준비 사항

1. (필수) Python Web Framework

- Django 2.1.x
- Python 3.6.x

3. 요구 사항

1. 데이터베이스 설계

- `db.sqlite3` 에서 테이블 간의 관계는 아래와 같습니다.

■ 아래

Genre(1) : Movie(N)

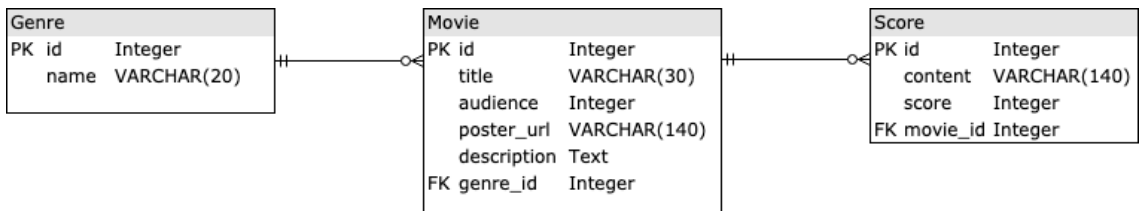
장르는 영화들을 가지고 있습니다.

예) 애니메이션 - 주먹왕랄프 2, 극장판 포켓몬스터 모두의 이야기

Movie(1) : Score(N)

영화는 유저들이 기입한 평점 정보를 가지고 있습니다.

예) 캡틴마블 - 4, 10, 5, 1, 10



◦ Genre

필드명	자료형	설명
id	Integer	Primary Key
name	String	장르 구분

○ Movie

필드명	자료형	설명
id	Interger	Primary Key
title	String	영화명
audience	Integer	누적 관객수
poster_url	Text	포스터 이미지 URL
description	Text	영화 소개
genre_id	Integer	Genre의 Primary Key(id 값)

○ Score

Name	자료형	설명
id	Integer	Primary Key
content	String	한줄평(평가 내용)
score	Integer	평점
movie_id	Integer	Movie의 Primary Key(id 값)

2. **movies** App

Genre는 CRUD를 만들지 않습니다.

1. 영화와 장르 정보는 제공된 csv 파일을 통해 데이터베이스에 반영합니다.

아래의 코드는 예시입니다. (프로젝트 디렉토리내에 csv파일이 있으며, movies app의 Genre 및 Movie 클래스인 경우)

반드시 migrate를 한 이후 실행할 것.

```
$ sqlite3 db.sqlite3
>> .mode csv
>> .import genre.csv movies_genre
>> .import movie.csv movies_movie
```

2. 영화 목록

1. (필수) 해당 페이지에 접근하는 URL은 `GET /movies/` 입니다.
2. (필수) 데이터베이스에 존재하는 모든 영화의 목록이 표시 되며, 각 영화의 `title`, 영화포스터가 표시됩니다.
3. (필수) `title`을 클릭 시, 해당 `영화 정보 조회` 페이지로 이동합니다.

3. 영화 정보 조회

1. (필수) 영화 정보 상세 조회 URL은 `GET /movies/1/`, `GET /movies/2/` 등이며, 동적으로

할당되는 부분이 존재합니다. 동적으로 할당되는 부분에는 데이터베이스에 저장된 영화 정보의 Primary Key가 들어갑니다.

2. **(필수)** 해당 Primary Key를 가진 영화의 모든 정보가 표시됩니다. **(장르 포함)**

3. **(필수)** 영화 정보의 최하단에는 **목록**, **수정**, **삭제** 링크가 있으며, 클릭 시 각각 **영화 목록**, **해당 영화 정보 수정 Form**, **해당 영화 정보 삭제** 페이지로 이동합니다.

4. 영화 정보 삭제

1. **(필수)** 영화 삭제 URL은 `POST /movies/1/delete/`, `POST /movies/2/delete/` 등이며, 동적으로 할당되는 부분이 존재합니다. 동적으로 할당되는 부분에는 데이터베이스에 저장된 영화 정보의 Primary Key가 들어갑니다.

2. **(필수)** 해당 Primary Key를 가진 영화 정보를 데이터베이스에서 삭제합니다.

3. **(필수)** **영화 정보 목록** 페이지로 Redirect 합니다.

5. 평점 생성

1. **(필수)** **영화 정보 조회** 페이지에서 평점을 작성할 수 있습니다. (form)

2. **(필수)** 사용자로부터 한줄평과 평점(숫자)를 받습니다.

3. **(필수)** 평점 생성 URL은 `POST /movies/1/scores/new`, `POST /movies/2/scores/new` 등이며, 동적으로 할당되는 부분이 존재합니다. 동적으로 할당되는 부분에는 데이터베이스에 저장된 영화 정보의 Primary Key가 들어갑니다.

4. **(필수)** 해당하는 영화의 **영화 정보 조회** 페이지로 Redirect 합니다.

6. 평점 목록

1. **(필수)** **영화 정보 조회** 페이지에 해당하는 영화의 평점 내역이 나타납니다.

2. **(필수)** 한줄평과 평점을 모두 출력합니다.

7. 평점 삭제

1. **(필수)** **영화 정보 조회** 페이지에 있는 **평점 목록** 에 삭제 버튼이 있습니다.

2. **(필수)** 평점 삭제 URL은 `POST /movies/1/scores/1/delete/`, `POST /movies/2/scores/3/delete/` 등이며, 동적으로 할당되는 부분이 존재합니다. 동적으로 할당되는 부분에는 데이터베이스에 저장된 영화 정보의 Primary Key와 평점 정보의 Primary Key가 각각 들어갑니다.

3. **(필수)** 해당 Primary Key를 가진 평점 정보를 데이터베이스에서 삭제합니다.

4. **(필수)** 해당하는 영화의 **영화 정보 조회** 페이지로 Redirect 합니다.