SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

01 PJT

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

Python 을 활용한 데이터 수집과 전처리 및 CSV 파일

SAMSUNG SW ACADEMY FOR YOUTH

INDEX

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

개요

프로젝트 개요

- 커뮤니티 서비스 개발을 위한 데이터 구성 단계로,
  필요한 데이터를 직접 추출하고 재구성하는 과정

프로젝트 목표

- 데이터 구조에 대한 분석과 이해
- 데이터 관리를 위한 DB 구성
- 데이터를 가공하고 CSV 형태로 구성
- 각 테이블에 데이터 삽입

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

준비사항

개발도구 및 라이브러리

- 개발도구
  - Visual Studio Code
  - Python 3.9+
- 활용 데이터
  - TMDB API

제공사항

- examples 폴더
  - 이번 프로젝트 해결을 위해 알아야 하는 혹은 직접적인 도움이 될 수 있는 코드

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

요구사항

필수 요구사항

A. **기본 영화 정보** 테이블 생성 및 데이터 수집

B. **영화 상세 정보** 테이블 생성 및 데이터 수집

C. **영화 리뷰 정보** 테이블 생성 및 데이터 수집

D. **영화 배우 정보** 테이블 생성 및 데이터 수집과 전처리

E. **영화 평점 통계** 테이블 생성 및 복합 데이터 수집과 전처리

A. 기본 영화 정보 테이블 생성 및 데이터 수집 - 요구사항

- problem_a.py 풀이
- movies 테이블 생성
  - 필요한 정보
    - 영화 ID(id), 영화 제목(title), 개봉일(release_date), 인기 점수(popularity)
- TMDB API에서 인기 영화 데이터를 수집
- 각 영화의 ID, 제목, 개봉일, 인기 점수를 추출하고 적절히 처리
- 처리된 데이터를 CSV 파일(movies.csv)로 저장
- 테이블에 수집한 데이터 삽입

A. 기본 영화 정보 테이블 생성 및 데이터 수집 - schema

Column name	Type	Description
id	INT, PRIMARY KEY	영화 ID
title	VARCHAR(255)	영화 제목
release_date	DATE	개봉일
popularity	FLOT	인기 점수

A. 제공되는 도서 데이터의 주요내용 수집 - 결과

- 데이터 수집 완료 예시

	* id int	* title varchar(255)	release_date date	popularity float
	Filter	Filter	Filter	Filter
1	38700	Bad Boys for Life	2020-01-15	791.696
2	150540	Inside Out	2015-06-17	1529.96
3	437342	The First Omen	2024-04-03	627.181
4	573435	Bad Boys: Ride or Die	2024-06-05	2520.87
5	614933	Atlas	2024-05-23	1014.93

❖ 주의) 수집 데이터는 요청 시기에 따라 다를 수 있음

B. 영화 상세 정보 테이블 생성 및 데이터 수집 - 요구사항

- problem_b.py 풀이
- movie_details 테이블 생성
  - 필요한 정보
    - 영화 ID(movie_id), 예산(budget), 수익(revenue), 상영 시간(runtime), 장르(genres)
- TMDB API에서 특정 영화의 상세 정보 수집
  - 특정 영화 ID는 movies.csv 데이터를 활용
- 각 영화의 ID, 예산, 수익, 상영 시간, 장르를 추출하고 적절히 처리
- 처리된 데이터를 CSV 파일(movie_details.csv)로 저장
- 테이블에 수집한 데이터 삽입

B. 영화 상세 정보 테이블 생성 및 데이터 수집 - schema

Column name	Type	Description
movie_id	INT, PRIMARY KEY	영화 ID
budget	INT	예산
revenue	INT	수익
runtime	INT	상영 시간 (분)
genres	VARCHAR(255)	장르 (‘,’로 구분된 문자열)

B. 영화 상세 정보 테이블 생성 및 데이터 수집 - 결과

- 데이터 수집 완료 예시

	* movie_id int	budget int	revenue int	runtime int	genres varchar(255)
	Filter	Filter	Filter	Filter	Filter
1	38700	90000000	426505244	124	Thriller, Action, Crime
2	150540	175000000	857611174	95	Animation, Family, Adventure, Drama, Comedy
3	437342	30000000	53689531	119	Horror
4	573435	100000000	130151244	115	Action, Crime, Thriller, Comedy
5	614933	100	0	120	Science Fiction, Action
6	626412	0	9800000	122	Science Fiction, Action, Fantasy, Adventure

❖ 주의) 수집 데이터는 요청 시기에 따라 다를 수 있음

C. 영화 리뷰 정보 테이블 생성 및 데이터 수집 - 요구사항

- problem_c.py 풀이
- movie_reviews 테이블 생성
  - 필요한 정보
    - 리뷰 ID(review_id), 영화 ID(movie_id), 작성자(author), 리뷰 내용(content), 평점(rating)
    - 영화 ID(movie_id)는 movies 테이블의 ID를 참조
- TMDB API에서 각 영화의 리뷰 데이터 수집
  - 특정 영화 ID는 movies.csv 데이터를 활용
- 리뷰의 평점이 5점 이상인 경우만 선택하고, 리뷰 내용이 없는 경우 '내용 없음'으로 대체
- 처리된 데이터를 CSV 파일(movie_reviews.csv)로 저장
- 테이블에 수집한 데이터 삽입

C. 영화 리뷰 정보 테이블 생성 및 데이터 수집 - schema

Column name	Type	Description
review_id	VARCHAR(255), PRIMARY KEY	리뷰 ID
movie_id	INT	영화 ID
author	VARCHAR(255)	작성자
content	TEXT	리뷰 내용
rating	FLOAT	평점

C. 영화 리뷰 정보 테이블 생성 및 데이터 수집 - 결과

- 데이터 수집 완료 예시

	* review_id varchar(255)	movie_id int	author varchar(255)	content text	rating float
	Filter	Filter	Filter	Filter	Filter
1	5611c3d992514178990 02fc	150540	Fatota	This is the most incredible n	10
2	56127371c3a368680b0152 9	150540	Andres Gomez	Another great movie from F	8
3	564d7a06c3a368602b009af	150540	Sxerks3	A powerfully moving story,	8
4	5e2099dc0102c900163d107	38700	Manuel São Bento	If you enjoy reading my Spc	5
5	5e87d446b84f940014c8f32 c	150540	Peter McGinn	I think this is one of the bes	10
6	5e8bbac63e09f30012a33ee	38700	itsogs	Another action packed mov	9

❖ 주의) 수집 데이터는 요청 시기에 따라 다를 수 있음

D. 영화 배우 정보 테이블 생성 및 데이터 수집과 전처리 - 요구사항 (1/2)

- problem_d.py 풀이
- movie_cast 테이블 생성
  - 필요한 정보
    - 배우 ID(cast_id), 영화 ID(movie_id), 배우 이름(name), 배역 이름(character), 출연 순서(order)
    - 영화 ID(movie_id)는 movies 테이블의 ID를 참조
- TMDB API에서 영화의 배우 정보를 수집
  - 특정 영화 ID는 movies.csv 데이터를 활용

D. 영화 배우 정보 테이블 생성 및 데이터 수집과 전처리 - 요구사항 (2/2)

- 배우 이름과 배역 이름에서 줄바꿈 문자를 공백으로 변경
- 배우 이름이 없는 경우 '이름 없음'으로 대체
- 출연 순서(order)가 10 이하인 배우들만 선택
- 동일 배우가 여러 번 출연한 경우 첫 번째 항목만 사용
- 처리된 데이터를 CSV 파일(movie_cast.csv)로 저장
- 테이블에 수집한 데이터 삽입

D. 영화 배우 정보 테이블 생성 및 데이터 수집과 전처리 - schema

Column name	Type	Description
cast_id	INT, PRIMARY KEY	배우 ID
movie_id	INT	영화 ID
name	VARCHAR(255)	배우 이름
character	VARCHAR(255)	배역 이름
order	INT	출연 순서

D. 영화 배우 정보 테이블 생성 및 데이터 수집과 전처리 - 결과

- 데이터 수집 완료 예시

	* cast_id int	movie_id int	* name varchar(255)	character varchar(255)	order int
	Filter	Filter	Filter	Filter	Filter
1	0	38700	Will Smith	Mike Lowrey	0
2	1	955555	Ma Dong-seok	Ma Seok-do	0
3	2	823464	Dan Stevens	Trapper	2
4	3	626412	Kim Tae-ri	Ean	1
5	4	1022789	Amy Poehler	Joy (voice)	0

❖ 주의) 수집 데이터는 요청 시기에 따라 다를 수 있음

E. 영화 평점 통계 테이블 생성 및 복합 데이터 수집과 전처리 - 요구사항 (1/2)

- problem_e.py 풀이
- movie_ratings 테이블 생성
  - 필요한 정보
    - 영화 ID(movie_id), 평균 평점(average_rating),
      투표 수(vote_count), 평점 분포(rating_distribution)
- TMDB API에서 영화의 평점 정보를 수집
  - 평점 분포 계산을 위한 rating 정보는 movie_reviews.csv 데이터를 활용

E. 영화 평점 통계 테이블 생성 및 복합 데이터 수집과 전처리 - 요구사항 (2/2)

- API 데이터를 사용하여 평균 평점과 투표 수를 수집
- 평균 평점이나 투표 수가 없는 경우 0으로 대체
- movie_reviews.csv를 사용하여 평점 분포를 계산
  - 각 review의 rating 값의 소수점을 제한 값의 수를 누적
  - 각 평점의 개수를 JSON 형식으로 저장
- 처리된 데이터를 CSV 파일(movie_ratings.csv)로 저장
- 테이블에 수집한 데이터 삽입

E. 영화 평점 통계 테이블 생성 및 복합 데이터 수집과 전처리 - schema

Column name	Type	Description
movie_id	INT, PRIMARY KEY	영화 ID
average_rating	FLOAT	평균 점수
vote_count	INT	투표 수
rating_distribution	JSON	평점 분포 (1부터 10점까지)

E. 영화 평점 통계 테이블 생성 및 복합 데이터 수집과 전처리 - 결과

- 데이터 수집 완료 예시

	* movie_id int	average_rating float	vote_count int	rating_distribution json
	Filter	Filter	Filter	Filter
1	38700	7.125	7882	{"5": 2, "6": 1, "9": 1}
2	150540	7.915	20524	{"5": 1, "6": 1, "7": 2, "8": 3, "10": 2}
3	437342	6.777	501	{"6": 3}
4	573435	7.049	263	{"7": 1}
5	614933	6.746	745	{"8": 3}

❖ 주의) 수집 데이터는 요청 시기에 따라 다를 수 있음

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

제출

제출 시 주의사항

- 제출기한은 금일 18시까지 입니다. 제출기한을 지켜 주시기 바랍니다.
- 반드시 `README.md` 파일에 단계별로 구현 과정 중 학습한 내용, 어려웠던 부분, 새로 배운 것들 및 느낀 점을 등을 상세히 기록하여 제출합니다.
  - 단순히 완성된 코드만을 나열하지 않습니다.
- 위에 명시된 요구사항은 최소 조건이며, 추가 개발을 자유롭게 진행할 수 있습니다.
- https://lab.ssafy.com/에 프로젝트를 생성하고 제출합니다.
  - 프로젝트 이름은 '프로젝트 번호 + pjt'로 지정합니다. (ex. `01-pjt`)
- 반드시 각 반 담당 강사님을 Maintainer로 설정해야 합니다.