PJT명	Python을 활용한 데이터 수집과 전처리 및 CSV 파일 생성	
단계	[Python PJT]	
진행일자	2025.07.25	
예상 구현 시간	필수기능	5H
	추가기능	2H
	심화기능	1H

1. 목표

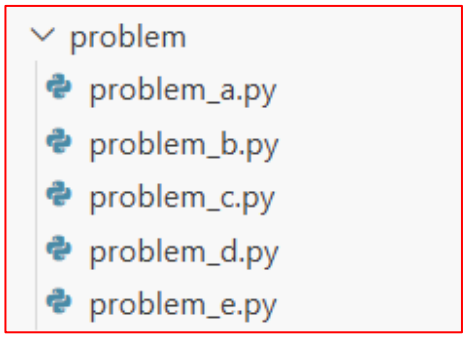
- 데이터 구조에 대한 분석과 이해 능력을 함양한다.
- Python을 활용하여 데이터를 가공하고 CSV 형태로 구성할 수 있다.
- 데이터베이스 스키마를 설계하고, 수집한 데이터를 테이블에 삽입할 수 있다.
- API를 활용하여 외부 데이터를 수집하고, 요구사항에 맞게 전처리할 수 있다.

2. 준비사항

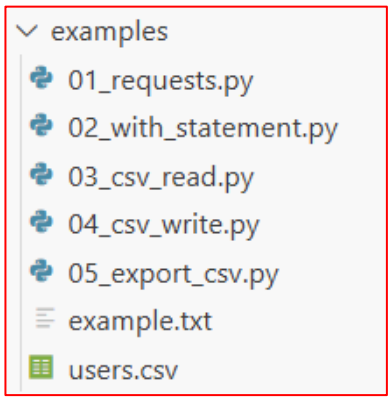
1) 프로젝트 구조

- 프로젝트는 문제 해결을 위한 스켈레톤 코드와 예시 코드를 포함하여 구성됩니다.
  1. problem 폴더
  2. examples 폴더

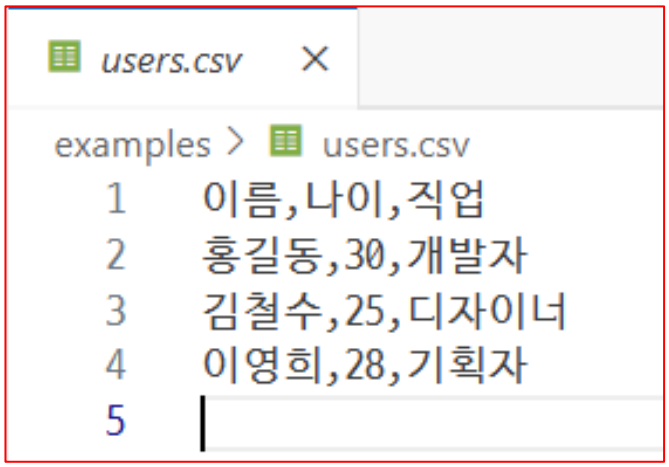
- problem 폴더
  - 요구사항 구현을 위한 스켈레톤 코드 파일 5개 (problem_a.py ~ problem_e.py)가 제공됩니다.



- examples 폴더
  - 프로젝트 해결에 도움이 될 수 있는 예시 코드가 포함되어 있습니다. 코드는 강의 시간에 함께 작성합니다.



1. users.csv: csv 데이터 예시



1) 사용 데이터

- TMDB API: 영화 데이터 수집을 위해 TMDB(The Movie Database) API를 활용합니다.

2) 개발언어 및 툴

- Python 3.11+ / Visual Studio Code

3) 필수 라이브러리 / 오픈소스

- Python 내장 모듈 json
- Python 패키지 requests
- TMDB API

3. 작업 순서

1) 팀원과 같이 요구사항(필수/도전)을 확인하고, GitLab에 프로젝트를 생성합니다.
   - 프로젝트 이름은 `01-pjt`로 지정합니다.
   - 각 반 담당 강사님을 Maintainer로 설정합니다.
2) 제공된 `examples` 폴더의 코드를 확인하고, 요구사항에 필요한 API 활용법 및 데이터 처리 방식을 파악합니다.
3) 각 문제 코드(`problem_*.py`)를 확인하고, 요구사항에 따라 TMDB API 데이터를 활용하여 필수 요구사항을 구현합니다.
4) 작성한 코드들을 정리하고, `README.md`를 작성합니다.
5) `README.md` 작성이 완료되면 도전 과제를 진행합니다.
6) 제출 기한에 맞춰 모든 산출물이 GitLab에 업로드 될 수 있도록 합니다.

4. 요구사항

본격적인 영화 추천 커뮤니티 서비스 개발에 앞서, 서비스의 기반이 될 데이터를 구성하는 것을 목표로 합니다. 이를 위해 외부 API(TMDB)에서 제공되는 영화 관련 데이터를 수집 및 관리하고, 이를 분석 및 전처리하여 재사용 가능한 형태로 가공하는 프로그램을 구현해야 합니다.

사용자는 인기 영화 목록, 각 영화의 상세 정보, 리뷰, 출연진 등 다양한 데이터를 수집하고, 이를 정제하여 여러 개의 CSV 파일로 저장하게 됩니다. 이 과정은 데이터베이스 구축의 전 단계로, 체계적인 데이터 파이프라인의 기초를 다지는 작업입니다.

팀원과 상의하여 아래 요구사항을 만족할 수 있도록 요구사항 명세서를 작성 및 구현합니다.

영화 정보를 전부 보유하기는 어려우므로, 외부에서 영화 데이터를 가져와야 합니다. API를 활용하여 데이터를 받아오고, 필요한 형태로 정리하는 연습 해 봅시다.

- 요구사항 예시(참고용)
  - 아래의 내용을 참고하여 추가적인 아이디어에 대해 요구사항을 추가 또는 수정하여 기능을 구현한다. 단, **필수 기능은 반드시 구현해야 하며, 임의로 변경 할 수 없다**.

번호	분류	요구사항명	요구사항 상세	우선순위
기능적 요구사항				
F01	영화 데이터	기본 정보 수집	인기 영화 목록에서 ID, 제목, 개봉일, 인기 점수를 추출하여 CSV로 저장하는 기능	필수
F02	영화	상세 정보 수집	각 영화의 예산, 수익, 상영 시간, 장르	필수

	데이터		정보를 추출하여 CSV로 저장하는 기능	
F03	영화 데이터	리뷰 정보 수집	각 영화의 리뷰 정보를 조건에 맞게 필터링하고 처리하여 CSV로 저장하는 기능	필수
F04	영화 데이터	출연진 정보 수집 및 전처리	각 영화의 출연진 정보를 조건에 맞게 필터링하고 처리하여 CSV로 저장하는 기능	필수
F05	영화 데이터	평점 통계 정보 수집 및 전처리	API 데이터와 수집된 리뷰 데이터를 복합적으로 활용하여 평점 통계 정보를 생성하고 CSV로 저장하는 기능	필수
F06	영화 데이터	최고 수익률 영화 분석	수집된 데이터를 바탕으로 예산 대비 수익률이 가장 높은 영화를 찾는 기능	도전
F07	영화 데이터	다작 배우 목록 수집	수집된 출연진 정보에서 2편 이상의 영화에 출연한 배우 목록을 추출하는 기능	도전

1) 기본(필수) 기능

A. 기본 영화 정보 테이블 생성 및 데이터 수집

인기 있는 영화 목록을 API를 통해 조회하여, 모든 데이터 처리의 기준이 될 기본 영화 정보를 수집합니다. 각 영화의 고유 ID, 제목, 개봉일, 그리고 인기도 점수를 추출하여 movies.csv 파일로 저장해야 합니다.

- 요구사항 번호: F01
- 구현: `problem_a.py` 파일을 수정하여 구현합니다.
- 필요한 정보: 영화 ID(id), 영화 제목(title), 개봉일(release_date), 인기 점수(popularity)

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

- 구현해야 할 화면 (CSV 저장 결과 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

기본 정보만으로는 서비스에 활용하기 부족하므로, F01에서 수집한 개별 영화 ID를 바탕으로 각 영화의 상세 정보를 추가로 수집합니다. 영화의 예산, 수익, 총 상영 시간, 그리고 장르 정보를 추출하여 movie_details.csv 파일로 저장합니다..

- 요구사항 번호: F02
- 구현: `problem_b.py` 파일을 수정하여 구현합니다.
- 필요한 정보: 영화 ID(movie_id), 예산(budget), 수익(revenue), 상영 시간(runtime), 장르(genres).

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)

- 구현해야 할 화면 (CSV 저장 결과 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

사용자들의 평가를 분석하기 위해 각 영화에 달린 리뷰 데이터를 수집합니다. 모든 리뷰를 수집하는 것이 아니라, 평점이 5점 이상인 리뷰만 필터링해야 합니다. 또한, 리뷰 내용이 비어있는 경우 '내용 없음'으로 처리하는 전처리 과정을 포함하며, 최종 결과를 movie_reviews.csv 파일로 저장합니다.

- 요구사항 번호: F03
- 구현: `problem_c.py` 파일을 수정하여 구현합니다.
- 필요한 정보: 리뷰 ID(review_id), 영화 ID(movie_id), 작성자(author), 리뷰 내용(content), 평점(rating).

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

- 구현해야 할 화면 (CSV 저장 결과 예시)

	review_id varchar(255)	movie_id int	author varchar(255)	content text	rating float
	Filter	Filter	Filter	Filter	Filter
1	5611c3d99251417899002fc	150540	Fatota	This is the most incredible r	10
2	56127371c3a368680b0152	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	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

영화에 출연한 배우와 배역 정보를 수집합니다. 비중 있는 역할을 파악하기 위해 출연 순서가 10 이하인 배우들만 선택하며 , 데이터에 포함된 불필요한 줄바꿈 문자를 공백으로 변경하는 등의 데이터 정제 작업을 수행합니다. 처리된 정보는 movie_cast.csv 파일로 저장합니다.

- 요구사항 번호: F04
- 구현: `problem_d.py` 파일을 수정하여 구현합니다.
- 필요한 정보: 배우 ID(cast_id), 영화 ID(movie_id), 배우 이름(name), 배역 이름(character), 출연 순서(order).

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

- 구현해야 할 화면 (CSV 저장 결과 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

여러 데이터 소스를 복합적으로 활용하여 영화의 종합적인 평점 통계를 생성합니다. API에서는 평균 평점과 총투표 수를 가져오고, F03에서 수집한 movie_reviews.csv 파일을 분석하여 1점에서 10점까지의 평점 분포를 계산합니다. 이 두 정보를 결합하여 movie_ratings.csv 파일로 저장합니다.

- 요구사항 번호: F05
- 구현: `problem_e.py` 파일을 수정하여 구현합니다.
- 필요한 정보: 영화 ID(movie_id), 평균 평점(average_rating), 투표 수(vote_count), 평점 분포(rating_distribution).

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)

- 구현해야 할 화면 (CSV 저장 결과 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

2) 도전 과제

기본 기능 구현 후, 수집한 데이터를 바탕으로 다음 두 도전과제 요구사항을 해결합니다.

A. 최고 수익률 영화 분석

수집된 데이터를 바탕으로 실질적인 분석을 수행하는 도전 과제입니다. 각 영화의 예산(budget)과 수익(revenue) 정보를 활용하여, 투자 대비 가장 높은 성과를 낸 영화가 무엇인지 수익률을 계산하고 해당 영화를 찾아내는 기능을 구현합니다.

- 요구사항 번호: F06
- 구현: `problem_f_1.py` 파일을 생성하여 구현합니다.

B. 다작 배우 목록 수집

수집된 출연진 데이터를 집계하여 의미 있는 정보를 추출하는 도전 과제입니다. movie_cast.csv 파일의 정보를 분석하여, 이번에 수집된 영화 목록 중 2편 이상에 출연한 배우들의 목록을 찾아내는 기능을 구현합니다.

- 요구사항 번호: F07
- 구현: `problem_f_2.py` 파일을 생성하여 구현합니다.

5. 참고자료

- json – JSON encoder and decoder
  https://docs.python.org/3.11/library/json.html
- TMDB API Documentation
  https://developer.themoviedb.org/docs

6. 결과

제출 기한은 진행일 18시까지이므로 제출 기한을 지킬 수 있도록 합니다. 제출은 GitLab을 통해서 이루어집니다.

- **산출물과 제출**
  - 단계별로 구현 과정 중 학습한 내용, 어려웠던 부분, 새로 배운 것들 및 느낀 점을 상세히 기록한 README.md
  - 완성된 각 문제 별 소스코드 및 실행 화면 캡쳐본
  - 프로젝트 이름은 01-pjt로 지정, 각자의 계정에 생성할 것
  - 각 반 담당 강사님을 Maintainer로 설정

- 끝 -