Project: Implementing a Simple Database Application

Due: 2023/6/14 (Wed), 11:59 PM

이번 프로젝트의 목표는 데이터베이스를 이용하는 간단한 어플리케이션을 설계하고 구현하는 것이다. **Python** 및 **MySQL**을 이용하여 영화 예매를 시뮬레이션 하는 간단한 어플리케이션을 만들어 본다. 이번 프로젝트를 통해 학생들은 어플리케이션과 데이터베이스를 연동하는 방법을 배우게 된다.

1 요구 사항

- 데이터는 관계형 데이터베이스에 저장되어야 한다.
- 주어진 데이터로 데이터베이스를 초기화 할 수 있어야 한다.
- 영화 데이터 및 고객 데이터를 추가/삭제할 수 있어야 한다.
- 영화 데이터 및 고객 데이터를 출력할 수 있어야 한다.
- 고객이 영화를 예매할 수 있어야 한다.
- 고객이 영화에 평점을 부여할 수 있어야 한다.
- 영화 별로 영화를 예매한 고객의 명단 및 평점 정보를 출력할 수 있어야 한다.
- 영화 별로 부여된 평점을 출력할 수 있어야 한다.
- 특정 고객에게 영화를 추천할 수 있어야 한다.

2 세부 사항

구현한 프로그램은 다음과 같은 기능을 할 수 있어야 한다.

1. 데이터베이스 초기화 (5점)
    - 주어진 raw data를 데이터베이스에 추가한다.
        - 주어진 **data.csv** 파일을 읽어와 구성한 데이터베이스에 **추가**한다.
            - 각 레코드를 추가하는 과정에서 오류가 날 경우, 해당 레코드를 무시한다.
        - 기존 테이블이 없다면, 테이블을 새롭게 생성한다.
        - 데이터베이스 스키마 및 구성 방식은 프로그램의 기능들을 고려해 자유롭게 설계한다.
            - 데이터베이스 구성에 대한 설명은 보고서에 작성한다.
2. 모든 영화 정보 출력 (4점)

- 모든 영화의 정보를 출력한다.
    - 영화 ID, 제목, 감독, 원래 가격, 평균 예매 가격, 예매자 수, 평균 평점 순으로 출력한다.
    - 영화 ID 오름차순으로 출력한다.
- 영화에 대한 평점이 존재하지 않는다면 '*None*' 으로 출력한다.

3. 모든 고객 정보 출력 (4점)

- 모든 고객의 정보를 출력한다.
    - 고객 ID, 이름, 나이, 등급 순으로 출력한다.

    고객 ID 오름차순으로 출력한다

4. 영화 추가 (4점)

- 새로운 영화를 추가한다.
    - 제목, 감독, 가격을 입력으로 받는다.
    - 가격은 0원부터 100000원까지의 정수인 제한 조건을 가진다
        - 가격 제한 조건에 맞지 않는 값을 입력 받으면 *에러 메시지*를 출력한다
    - 모든 영화는 고유한 제목을 갖는다.
        - 중복되는 영화 제목이 들어올 경우 오류 메시지를 출력한다
    - 편의를 위해 각 영화에 별도의 ID 값을 부여한다.
        - ID는 새로운 영화가 추가될 때마다 자동 부여한다.
        - ID는 1부터 시작하는 *AUTO_INCREMENT* 옵션을 사용한다.

5. 고객 추가 (4점)

- 새로운 고객을 추가한다.
    - 이름, 나이, 등급을 입력으로 받는다.
    - 나이는 12~110세까지의 제한 조건을 가진다
        - 나이 제한 조건에 맞지 않는 값을 입력 받으면 *에러 메시지*를 출력한다
    - 모든 고객은 고유한 조건(이름, 나이)을 갖는다.
        - (홍길동, 22살)과 (홍길동, 21살)은 서로 다른 사람이지만, 또 다른 (홍길동, 22살)이 존재하지 않는다고 가정한다.
        - 중복되는 고유 조건이 들어올 경우, *에러 메시지*를 출력한다
    - 편의를 위해 각 고객에 별도의 ID 값을 부여한다.
        - ID는 새로운 고객이 추가될 때마다 자동 부여한다.

        - ID는 1부터 시작하는 *AUTO_INCREMENT* 옵션을 사용한다.

6. 영화 삭제 (4점)
    - 영화를 삭제한다.
        - 영화 ID를 입력 받는다.
        - 영화를 삭제하면 해당 영화의 예매 정보도 모두 삭제되어야 한다.
        - 영화를 삭제하면 해당 영화의 평점 정보도 모두 삭제되어야 한다.
        - 해당 ID의 영화가 존재하지 않는다면 *에러 메시지*를 출력한다.
7. 고객 삭제 (4점)
    - 고객을 삭제한다.
        - 고객 ID를 입력 받는다.
        - 고객을 삭제하면 관련된 예매 정보도 모두 삭제되어야 한다.
        - 고객을 삭제하면 관련된 평점 정보도 모두 삭제되어야 한다.
        - 해당 ID의 고객이 존재하지 않는다면 *에러 메시지*를 출력한다.
8. 영화 예매 (5점)
    - 고객이 영화를 예매한다.
        - 영화 ID, 고객 ID를 입력 받는다.
        - 고객은 등급에 따라 원래 영화 가격에 할인을 받는다.
        - 등급은 basic, premium, vip 중 하나이다.
        - Basic은 할인 없음, Premium은 25%, Vip는 50% 할인을 받는다.
            - 예시) Vip 고객의 경우, 원래 영화 가격 price가 10000이면, 예매 가격 reserve price는 5000이다.
        - 한 고객은 동일한 영화를 한 번만 예매할 수 있다고 가정한다.
            - 한 고객이 동일한 영화를 2번 이상 예매하는 경우 *에러 메시지*를 출력한다.
        - 한 영화는 최대 10명의 고객만 예매할 수 있다.
            - 10명 초과 고객이 예매할 경우 *에러 메시지*를 출력한다
        - 해당 ID의 영화가 존재하지 않는다면 *에러 메시지*를 출력한다.
        - 해당 ID의 고객이 존재하지 않는다면 *에러 메시지*를 출력한다.
9. 평점 부여 (5점)
    - 고객이 영화에 평점을 부여한다.

- 영화 ID, 고객 ID, 평점(1~5 사이의 정수)을 입력 받는다.
- 해당 ID의 영화가 존재하지 않는다면 *에러 메시지*를 출력한다.
- 해당 ID의 고객이 존재하지 않는다면 *에러 메시지*를 출력한다.
- 고객은 본인이 예매했던 영화에만 평점을 부여할 수 있다.
  - 고객이 영화를 예매한 기록이 없다면 *에러 메시지*를 출력한다.
- 이미 평점을 부여 했을 경우 *에러 메시지*를 출력한다.
- 평점이 유효한 범위(1~5 사이의 정수)를 벗어나면 *에러 메시지*를 출력한다.

10. 영화를 예매한 고객 정보 출력 (5점)

- 영화를 예매한 모든 고객의 정보를 출력한다.
  - 영화 ID를 입력 받는다.
  - 해당 영화를 예매한 고객의 정보를 중복 없이 출력한다.
    - 고객 ID, 이름, 나이, 예매 가격, 고객이 부여한 평점 순으로 출력한다.
    - 고객 ID 오름차순으로 출력한다.
    - 고객이 부여한 평점이 없다면 '*None*' 으로 출력한다.
  - 해당 ID의 영화가 존재하지 않는다면 *에러 메시지*를 출력한다.

11. 고객이 예매한 영화 정보 출력 (5점)

- 고객이 예매한 모든 영화의 정보를 출력한다.
  - 고객 ID를 입력 받는다.
  - 해당 고객이 예매한 영화의 정보를 중복 없이 출력한다.
    - 영화 ID, 제목, 감독, 예매 가격, 고객이 부여한 평점 순으로 출력한다.
    - 영화 ID 오름차순으로 출력한다
    - 고객이 부여한 평점이 없다면 '*None*' 으로 출력한다.
  - 해당 ID의 고객이 존재하지 않는다면 *에러 메시지*를 출력한다.

12. 고객을 위한 영화 추천 1 (6점)

- 고객이 보지 않은 영화 중 평균 평점이 가장 높은 영화와 가장 많은 고객이 본 영화를 추천한다.
- 고객 ID를 입력 받는다.
- 평균 평점이 가장 높은 영화와 가장 많은 고객이 본 영화를 출력한다.
  - 영화 ID, 제목, 예매 가격, 관객 수, 평균 평점 순으로 출력한다.

- 평균 평점이 가장 높은 영화에 한해 예상된 평점이 같은 경우 ID값이 작은 영화를 추천한다.
    - 모든 영화가 평점이 없을 경우, ID값이 작은 영화를 추천한다.
- 가장 많은 고객이 본 영화에 한해 고객 수가 같을 경우 ID값이 작은 영화를 추천한다.
- **이미 고객이 본 영화는 추천에서 제외한다.**

- 해당 ID의 고객이 존재하지 않는다면 *에러 메시지*를 출력한다.

13. 고객을 위한 영화 추천 2 (10점)

- 고객이 가장 높은 평점을 부여할 것이라고 예상되는 영화를 추천한다.
    - **Item-based Collaborative Filtering**을 사용해서 영화를 추천한다.
        - CF 에도 다양한 방법들이 존재하지만, 이번 과제에서는 다음과 같은 알고리즘을 사용해 고객을 위한 영화를 추천한다.

1. 고객이 영화에 부여한 평점을 기반으로 user-item matrix를 구축한다.
    - 고객의 수가 n, 영화의 수가 m 인 경우 n×m의 matrix가 생성된다.
    - 어떤 영화에도 평점을 매기지 않은 고객은 matrix를 생성 시 제외한다.
    - 예시 : User 4가 아직 평점을 부여하지 않은 item 2, item 3에 대한 평점을 예측해서 User 4에게 가장 적합한 item을 추천하는 것을 목표로 한다.

	Item 1	Item 2	Item 3	Item 4
User 1	5	4	4	
User 2	1	3		
User 3		2	1	4
User 4	2	?	?	5

2. 고객이 아직 평점을 부여하지 않은 영화에 대해서는 영화의 평균 평점으로 평점을 부여한다.
    - 한 개의 평점도 부여되지 않은 영화의 경우 모두 0으로 초기화 한다.
    - 예시 : User1의 경우 Item4에 부여한 평점 정보가 없기 때문에, 기존에 부여한 평점들의 평균인 $\frac{4+5}{2} = 4.5$ 을 Item4의 임시 평점으로 부여한다.

	Item 1	Item 2	Item 3	Item 4
User 1	5	4	4	4.5
User 2	1	3	2.5	4.5
User 3	2.67	2	1	4
User 4	2	3	2.5	5

3. 각 영화에 부여된 평점의 유사도를 측정한다.
    - 두 영화에 부여한 평점의 유사도가 높은 경우 두 영화의 성향이 유사하다고 판단한다.

- 각 유사도는 소수점 4자리까지만 반영한다.

- 다양한 유사도 측정 방식 중 adjusted cosine similarity를 사용한다.

$$Adjusted\ cosine\ similarity\ = \frac{\sum_{i=1}^{n}(A_i-\mu)(B_i-\mu)}{\sqrt{\sum_{i=1}^{n}(A_i-\mu)^2}\sqrt{\sum_{i=1}^{n}(B_i-\mu)^2}}$$

- μ = 전체 평균

- 모든 아이템에 대해 유사도를 구해서 n×n의 similarity matrix를 생성한다.

  - Tip 1 : 자기 자신과의 유사도는 항상 1이다. 그러므로 similarity matrix의 대각 성분들은 모두 1이다.

  - Tip 2: Item 1과 Item 2의 유사도와 Item 2와 Item 1의 유사도는 동일하기 때문에 similarity matrix는 symmetric하게 구성된다.

- 예시 : 전체 평균 $\mu = \frac{5+4+4+4.5+\cdots}{16} = 3.1667$

- Item 1과 Item 2 사이의 adjusted cosine similarity를 구하면,
$\frac{(5-\mu)\times(4-\mu)+(1-\mu)\times(3-\mu)+\cdots}{\sqrt{(5-\mu)^2+(1-\mu)^2+\cdots}\sqrt{(4-\mu)^2+(3-\mu)^2+\cdots}} \approx 0.5896$

	Item 1	Item 2	Item 3	Item 4
Item 1	1	0.5896	0.6197	-0.3496
Item 2	0.5896	1	0.9462	-0.0972
Item 3	0.6197	0.9462	1	-0.4060
Item 4	-0.3496	-0.0972	-0.4060	1

4. 예측하고자 하는 고객의 영화와 나머지 영화와의 similarity를 weight으로 해서 weighted sum 값으로 각 영화에 대한 평점을 예측한다.

   - 각 평점은 소수점 2자리까지 출력한다.

	Item 1	Item 2	Item 3	Item 4
User 1	5	4	4	4.5
User 2	1	3	2.5	4.5
User 3	2.67	2	1	4
User 4	2	ⓐ	ⓑ	5

   - 예시 : User 4가 아직 평점을 부여하지 않은 item 2(ⓐ)와 item 3(②)의 평점을 예측하면 다음과 같이 구할 수 있다.

   - ⓐ $= \frac{0.5896\times2+0.9462\times2.5+(-0.0972)\times5}{0.5896+0.9462+(-0.0972)} \approx 2.1261$

   - ⓑ $= \frac{0.6197\times2+0.9462\times3+(-0.4060)\times5}{0.6197+0.9462+(-0.4060)} \approx 1.7657$

   - 추천 대상 (Item 2, Item 3) 중 Item 2의 예상 평점이 다른 추천 대상들의 예상 평점보다 높기 때문에, User 4에게는 item 2를 우선 추천하게 된다.

	Item 1	Item 2	Item 3	Item 4
User 1	5	4	4	4.5
User 2	1	3	2.5	4.5
User 3	2.67	2	1	4
User 4	2	2.13	1.77	5

- 고객 ID와 추천 영화 개수를 입력 받는다.
- 해당 고객이 가장 높은 평점을 부여할 것으로 예상되는 영화 정보를 추천 영화 개수만큼 출력한다.
  - 영화 ID, 제목, 예매 가격, 평균 평점, 예상된 평점 순으로 출력한다.
  - 예상된 평점이 높은 순대로 내림차순으로 출력한다.
  - 예상된 평점이 같은 경우 ID값이 작은 영화를 추천한다.
  - **이미 고객이 본 영화는 추천에서 제외한다.**
  - 추천 영화 개수(입력값 N)가 고객이 보지 않은 영화 개수를 초과할 경우, 최대 추천 개수만큼만 출력한다.
  - 해당 고객의 평점 정보가 하나도 존재하지 않는다면 *에러 메시지*를 출력한다.
- 해당 ID의 고객이 존재하지 않는다면 *에러 메시지*를 출력한다.

14. 프로그램 종료
15. 데이터베이스 리셋 및 생성 (5점)
    - 기존 모든 테이블 및 데이터가 존재할 경우 삭제한다.
      - 삭제 실시 전 확인 메시지를 띄우고 사용자 입력(y/n)을 받아야 한다.
    - 모든 데이터가 삭제된 경우, 1번을 수행해 데이터베이스를 초기화한다.

기타 사항

- 특정 입력에 대해 에러가 있을 시 그 즉시 에러 메시지를 출력하고 해당 명령을 종료한다.
- 모든 출력은 **첫 번째 컬럼**에 대해 **오름차순**으로 정렬되어야 한다.
- **해당 문서에 명시되지 않은 상황에 대해서는 자유롭게 가정하고, 해당 사항은 보고서에 명시한다.**

3 실행 예시

- 프로그램의 출력은 검은색 글씨로 표시됨
- 유저의 입력은 **빨간색** 글씨로 표시됨

```
============================================================
1. initialize database
2. print all movies
3. print all users
4. insert a new movie
5. remove a movie
6. insert a new user
7. remove a user
8. book a movie
9. rate a movie
10. print all users who booked for a movie
11. print all movies booked by a user
12. recommend a movie for a user using popularity-based method
13. recommend a movie for a user using item-based collaborative filtering
14. exit
15. reset database
============================================================
Select your action: 1
Database successfully initialized

Select your action: 2
-------------------------------------------------------------------------------
id      title           director        price  avg. price  reservation  avg. rating
-------------------------------------------------------------------------------
1       The Matrix      Lana Wachowski  10000  10000       2            None
2       WALL-E          Andrew Stanton  10000  10000       3            None
-------------------------------------------------------------------------------

Select your action: 3
-------------------------------------------------------------------------------
id      name                                    age             class
-------------------------------------------------------------------------------
1       Bae Sangwhan                            15              basic
2       Kim Jihoon                              30              basic
3       Choi Byungseo                           56              basic
-------------------------------------------------------------------------------

Select your action: 4
Movie title: Alien
Movie director: Ridley Scott
Movie price: 10000
One movie successfully inserted

Select your action: 6
User name: My User Name
User age: 19
User class: basic
One user successfully inserted

Select your action: 2
-------------------------------------------------------------------------------
id      title           director        price  avg. price  reservation  avg. rating
-------------------------------------------------------------------------------
1       The Matrix      Lana Wachowski  10000  10000       2            None
2       WALL-E          Andrew Stanton  10000  10000       3            None
3       Alien           Ridley Scott    10000  None        0            None
-------------------------------------------------------------------------------

Select your action: 3
-------------------------------------------------------------------------------
id      name                                    age             class
-------------------------------------------------------------------------------
1       Bae Sangwhan                            15              basic
2       Kim Jihoon                              30              basic
```

```
3       Choi Byungseo                           56              basic
4       My User Name                            19              basic
-------------------------------------------------------------------------------

Select your action: 8
Movie ID: 3
User ID: 1
Movie successfully booked

Select your action: 9
Movie ID: 1
User ID: 1
Ratings (1~5): 4
Movie successfully rated

Select your action: 9
Movie ID: 2
User ID: 1
Ratings (1~5): 4
Movie successfully rated

Select your action: 9
Movie ID: 2
User ID: 2
Ratings (1~5): 4
Movie successfully rated

Select your action: 9
Movie ID: 3
User ID: 1
Ratings (1~5): 4
Movie successfully rated

Select your action: 10
Movie ID: 1
-------------------------------------------------------------------------------
id      name                            age     res. price      rating
-------------------------------------------------------------------------------
1       Bae Sangwhan                    15      10000           4
2       Kim Jihoon                      30      10000           None
-------------------------------------------------------------------------------

Select your action: 11
User ID: 1
-------------------------------------------------------------------------------
id      title                           director        res. price      rating
-------------------------------------------------------------------------------
1       The Matrix                      Lana Wachowski  10000           4
2       Wall-E                          Andrew Stanton  10000           4
3       Alien                           Ridley Scott    10000           4
-------------------------------------------------------------------------------

Select your action: 12
User ID: 2

-------------------------------------------------------------------------------
Rating-based
id      title       res. price  reservation  avg. rating
-------------------------------------------------------------------------------
3       Alien  10000        1            4
-------------------------------------------------------------------------------
Popularity-based
id      title       res. price  reservation  avg. rating
-------------------------------------------------------------------------------
```

```
3       Alien  10000        1            4
--------------------------------------------------------------------------------

Select your action: 13
User ID: 2
Recommend Count: 1
--------------------------------------------------------------------------------
id      title                   res. price      avg. rating    expected rating
--------------------------------------------------------------------------------
3       Alien              10000        4              4
--------------------------------------------------------------------------------

Select your action: 14
Bye!
```

4 개발 환경

- Python 3.6+
- MySQL
  - ◆ DB 접속 정보
    - ■ Host: astronaut.snu.ac.kr
    - ■ Port: 7000
    - ■ ID: DB학번(예:DB2022_12345)
    - ■ Password: ID와 동일
  - ◆ 접속 후 password 변경 방법
    - ■ set password = password('비밀번호')

5 배점 (100점)

1. 구현 완성도 (90점)
   - 기능 완성도 (70점)
     - 기능별 세부 배점은 위에 명시된 배점을 따른다.
     - **1단계:** 데이터베이스 리셋 및 스키마 생성 (10점)
       - **해당 문제: 1, 15**
       - Raw data를 기반으로 데이터베이스 initialization이 되었는가?
       - 데이터베이스에 있는 데이터가 모두 삭제 되었는가?
     - **2단계:** 데이터의 단순 입출력 (24점)
       - **해당 문제: 2 ~ 7**

        - 데이터를 DB로부터 정확하게 retrieve하여 표시할 수 있는가?
        - 주어진 input 대로 데이터 레코드를 생성하여 DB에 저장하고 있는가?
        - 지정한 데이터 레코드를 삭제할 수 있는가?
    - **3단계:** DB와 Application의 복합적 연동 (36점)
        - **해당 문제: 8 ~ 13**
        - 잘 설계된 데이터베이스 스키마와 integrity constraint를 활용하여 명시된 기능을 잘 수행하도록 구현하였는가?
- 적합한 데이터베이스 스키마 설계 (10점)
- 청결한 코드 (10점)
    - 적절한 모듈화 및 주석 첨가로 코드 가독성이 확보되었는가?

2. 보고서 (10점)

6 제출

1. DB
    - **(중요!) 프로젝트 제출시 채점을 위해 자신의 DB 계정 안의 테이블 스키마만 남기고 레코드는 모두 truncate해야 함**
    - **채점 과정에서 형식은 동일하지만 다른 테스트 데이터가 쓰일 수 있음**
    - **이를 지키지 않았을 시 채점이 제대로 되지 않아 0점 처리 될 수 있으니 주의!**
2. 프로젝트 코드 압축 파일
    - 프로젝트 코드 최상단에서 "**`python run.py`**" 명령어를 통해 실행시킬 수 있어야 함
    - **`run.py`**와 **`data.csv`** 파일은 같은 폴더 내에 위치한다.
    - 외부 라이브러리 **`numpy`**와 **`pandas`**를 사용해도 되며, 사용한 외부 라이브러리는 **`requirements.txt`** 파일에 명시한다.
3. 리포트
    - 파일명: PRJ2_학번.pdf (예: PRJ2_2023-12345.pdf)
    - 반드시 pdf 포맷으로 제출
    - 포함되어야 하는 내용
        - 핵심 모듈과 알고리즘에 대한 설명
        - 구현한 내용에 대한 간략한 설명

- ◆ (제시된 요구사항 중 구현하지 못한 부분이 있다면) 구현하지 못한 내용
- ◆ 가정한 것들 (추가적으로 처리한 예외처리 등)
- ◆ 컴파일과 실행 방법
- ◆ 프로젝트를 하면서 느낀 점

- 위의 두가지를 압축하여 ETL로 제출
  - ■ **하루 Delay마다 -10%, 3일 Delay 시 0점 처리**