

1. 데이터베이스 스키마 설명

1.1. DVD

이 table은 DVD의 정보를 저장한다.

id, 제목, 감독 이름, 제한 나이, 누적 대출 횟수, 현재 남은 재고, 총 재고

1.2. User

이 table은 회원의 정보를 저장한다.

id, 이름, 나이, 누적 대출 횟수, 현재 대출 권수, 연체 횟수, 남은 페널티 수

1.3. Rating

이 table은 회원이 DVD에 남긴 평점을 저장한다.

DVD id, 회원 id, 평점

1.4. Loan

이 table은 현재 대출된 DVD와 대출한 회원의 정보를 저장한다.

회원 id, DVD id

2. 핵심 모듈 및 알고리즘 설명

기본적으로 스켈레톤을 충실히 따름

2.1. initialize_database

데이터베이스를 주어진 data.csv로 초기화함. 이 때 table이 존재한다면 모두 지우고 다시 만든다.

2.2. reset

유저에게 reset여부를 물어본 후 2.1을 실행시킴

2.3. print_DVDs/print_users

모든 DVD/user를 id 오름차순으로 출력

2.4. insert_DVD/user

주어진 정보를 통해 새로운 DVD/user를 생성

2.5. remove_DVD/user

주어진 id의 DVD/user를 제거. 이때 연관된 정보(rating)도 제거

2.6. checkout_DVD

DVD를 user가 대출. 연체 페널티에 대한 처리도 여기서 함.

2.7. return_and_rate_DVD

user가 DVD를 반납 및 평점을 매김

2.8. print_borrowing_status_for_user

주어진 id의 user가 빌리고 있는 DVD의 목록 출력. Loan table을 사용함

2.9. search

주어진 query가 포함된 제목의 DVD를 찾음. like문 사용

2.10. recommend_popularity

해당 유저가 보지 않은 DVD중 평균 평점이 가장 높은것과 가장 많이 대출해간 것을 추천

2.11. recommend_user_based

해당 유저가 보지 않은 DVD중 이 유저가 가장 높은 평점을 매길것으로 예상되는 것을 추천.

3. 구현 내용 요약

실행하면 명령어 집합이 먼저 출력되고 유저가 해당 명령어를 선택하면 그에 맞는 행동을 하는 함수로 이어진다. sql connection은 처음 실행 시 한 번만 만들어지며 모든 함수들간에 공유된다. 중간에 connection이 끊어지는 경우를 방지하기 위해 모든 함수의 시작에 connection의 연결을 보장할 수 있는 함수를 실행시킨다. 이를 통해 connection을 자주하여 발생하는 overhead를 줄인다.

4. 컴파일과 실행 방법

pip install 명령어를 통해 외부 라이브러리(mysql, pandas, numpy)를 다운로드.

그 후 python명령어를 통해 컴파일 및 실행

5. 느낀 점

mysql을 이용한 데이터 저장 및 조작을 실습해보았다. 매우 흥미로운 과제였고 내가 처리할 수 있는 작업도 최대한 DBMS에게 맡겨야 일이 쉬워진다는 것을 배웠다.

6. 기타사항

data.csv에 감독 이름이 32자인 경우가 있어 db schema상에서는 d_name이 varchar(32)로 되어있다. 이를 통해 초기화시 해당 정보가 제대로 저장되고 그 이후 조작으로 새로운 DVD를 넣는 경우는 감독 이름이 30자가 넘는 입력은 제한하고 있다.