첫 번째 단계인 E/R Diagram 설계에 대해 설명하겠습니다.

먼저 조건대로 Movie, User를 각각 strong entity로 설정한 다음, 보시는 바와 같이 movie 테이블에서는 movie\_id를 기본키로 두고 그 외의 attribute들은 다음과 같이 설정했으며 user 테이블에서는 user\_id를 기본키로 두고 그 외의 attribute들을 설정하였습니다. 그리고 movie의 multivalued에 해당하는 genre 테이블을 weak entity로 설정하여 1대다 관계로 두었으며, genre에서는 movie\_id를 외래키로 가져와 기본키로 두고 그 외의 속성은 genre의 이름과 genre의 번호로 구성하였습니다. 다음은 rate 테이블인데, 이는 user 테이블과 movie 테이블 간의 Association 관계를 성립시켜주는 테이블입니다. 따라서 user table의 기본키와 movie table의 기본키를 외래키로 데려와 기본키로 설정한 뒤 그외의 속성은 다음과 같이 설정하였습니다. 여기서 user 와 rate의 관계를 설명하면, user는 적어도 20개 이상의 영화를 평가했고 한 rate 정보는 한 명의 user에게서만 나오므로 1대 다의 관계로 설정하였습니다. 두번째로 movie와 rate의 관계를 설명하면, movie 또한 최소 20명 이상의 별점 평점이 매겨져 있다는 조건이 있고 한 rate 정보는 한 영화의 많은 평점 정보 중 하나에 해당되므로 1대다 관계로 설정하였습니다. 이상 1단계 E/R Diagram 설계에 대한 설명을 마치겠습니다.

두번째 단계인 Relational Database 설계입니다.

첫 번째 단계에서 설명한 E/R 다이어그램에 따라 데이터베이스 및 테이블을 생성하였습니다.

먼저 create database 문으로 db\_201812839라는 데이터베이스를 생성하고 use 문으로 그 데이터베이스를 쓴다고 선언한 뒤, user, movie, genre, rate 테이블을 만들었습니다. 먼저 user 테이블부터 보면 user\_id는 int형으로 설정하고 기본키로 설정하였습니다. 그 외에 나이는 int, 그 외의 속성들은 다 varchar()버캐릭터형으로 선언하였습니다. 그 다음에 무비 테이블을 보면, 무비 아이디 또한 int형으로 설정하고 기본키로 선언하였으며, 그 외의 속성들은 모두 버캐릭터 형으로 선언하였습니다

그 다음엔 user와 movie의 association 관계를 맺어주는 rate 테이블입니다. 먼저 보시는 바와 같이 user\_id와 movie\_id를 가져와 기본키로 묶고 외래키 처리를 해주었습니다. 그 외의 속성들은 int형으로 선언하였습니다. 그리고 마지막에 rating, 즉 평점의 범위가 1 이상 5 이하인지 검사해주는 것도 넣었습니다.

마지막으로 genre 테이블입니다. 여기서는 무비 테이블의 무비 아이디와 장르 아이디를 기본키로 묶어 설정하고 그 외의 속성인 장르 이름은 버캐릭터 형으로 선언하였습니다. 여기서 무비 아이디는 외래키로 지정하였습니다. 이상 2단계 데이터베이스 설계에 대한 설명을 마치겠습니다

세번째 단계인 데이터베이스 구축입니다. 파이썬에서의 구현을 통해 데이터베이스에 정보들을 각각 입력했습니다. 영상에서는 MySQL에서 확인했을 때 값들이 잘 들어갔는지 확인해보겠습니다. 주어진 데이터 셋은 943명의 유저와 1682개의 영화, 그리고 100000개의 데이터가 있었습니다.

4번째 단계인 검색 프로그램입니다. 장르, 사용자의 직업, 평점의 최소평점과 최대평점, 정렬 조건을 입력하여 그에 대한 영화 검색 결과를 제공하도록 설계하였습니다. (V) 먼저 GUI를 활용하여 검색 창에서 장르, 사용자의 직업, 정렬 조건은 다음과 같이 리스트 박스에서 선택할 수 있도록 하였으며, 평점의 하한선과 상한선은 직접 입력할 수 있도록 하였습니다. 다음은 시연입니다. 먼저 장르를 선택하고, 직업은 none으로 해준 뒤 title 순으로 정렬을 시켜보겠습니다. 보시는 바와 같이 해당 장르의 영화들이 title의 알파벳 순으로 정렬되어 나오는 것을 볼 수 있습니다. 다음은 평균평점에 의한 정렬로 출력시켜보겠습니다. 보시는 바와 같이 평점이 높은 순으로 출력이 됨을 알 수 있습니다. 사실 이 외의 사용자의 직업과 평점의 하한선과 상한선은 최대한 구현하려 했으나 잘 구현되지 않았습니다… 이상 4번째 단계인 검색 프로그램의 설명을 마치겠습니다.