

데이터베이스 기말 프로젝트:

MovieLens DB 구축 및 DB검색 프로그램 개발

MovieLens dataset

- This data set consists of:
 - 100,000 ratings (1-5) from 943 users on 1682 movies.
 - Each user has rated at least 20 movies.
 - Simple demographic info for the users (age, gender, occupation, zipcode)
- The data was collected through the MovieLens web site (movielens.umn.edu) during the seven-month period from September 19th, 1997 through April 22nd, 1998.
- URL:
 - <https://grouplens.org/datasets/movielens/100k/>

데이터의 구성

- u.genre
 - A list of the genres.
- u.item
 - Information about the movies;
 - this is a tab separated list of movie **id** | **movie title** | **release date** | **video release date** | **IMDb URL** | unknown | Action | Adventure | Animation | Children's | Comedy | Crime | Documentary | Drama | Fantasy | Film-Noir | Horror | Musical | Mystery | Romance | Sci-Fi | Thriller | War | Western |
 - The last 19 fields are the **genres**, a 1 indicates the movie is of that genre, a 0 indicates it is not; movies can be in several genres at once. The movie ids are the ones used in the u.data data set.

데이터의 구성

- u.user
 - Demographic information about the users;
 - this is a tab separated list of user id | age | gender | occupation | zip code
 - The user ids are the ones used in the u.data data set.
- u.occupation
 - A list of the occupations.
- u.data
 - The full u data set, 100000 ratings by 943 users on 1682 items.
 - Each user has rated at least 20 movies. Users and items are numbered consecutively from 1. The data is randomly ordered.
 - This is a tab separated list of **user id | item id | rating | timestamp**. The time stamps are unix seconds since 1/1/1970 UTC

1. E/R Modeling

- Movie, User을 각각 Strong Entity로 설계할 것.
- Genre는 Movie에 대하여 multi-valued attribute로 보고, ID Dependent, weak entity로 설계할 것.
- Movie와 User 의 Association Relationship을 설계할 것.
- E/R Diagram을 그릴 것.
- 시연 방법: E/R Diagram을 설명할 것.

2. Relational Database 설계

- E/R Diagram 으로부터, Relational Table들을 설계하고, mySQL에 관계형 데이터베이스를 생성하고, create table SQL 문을 사용하여 테이블들을 생성할 것
- 각 테이블에 기본키, 외래키를 반드시 지정할 것.
- 각 애트리뷰트의 타입을 정하고, 필요한 경우 not null, unique 등을 지정할 것.
- 시연 방법: create table SQL 문들을 가지고 설명할 것.

3. 데이터베이스 구축

- Python 프로그램을 작성하여, 주어진 source 데이터 파일들을 읽어들이고, mySQL 안에 데이터베이스(dbname = db_자기학번)를 구축할 것.
- 시연방법
 - Mysql workbench에서 SQL 문을 실행하여, 각테이블의 tuple 수가 맞는지, 튜플들은 정상적으로 입력되어 있는지 시연할 것.

4. 검색 프로그램 작성

- DB 응용 프로그램을 작성할 것
- 2가지 옵션
 - 텍스트 기반 콘솔 프로그램
 - 웹기반 GUI 프로그램 (가점 있음)
- 프로그래밍 언어는 가급적 python을 활용하되, java, php, jsp, asp 등 자율선택 가능
- 검색 시나리오

각 영화는 최소 20명 이상의 별점 평점이 매겨져 있음.

검색 조건: 영화의 장르, 사용자의 직업, 평점의 평균 등의 조건에서 선택할 수 있도록 할 것.

검색결과: 영화ID, 제목, 장르, 평균평점, vote 수(별점을 준 사용자의 수)

검색 정렬: 영화 제목순, 평균평점순, vote수 순 중에서 선택할 수 있도록 할것

4. 검색 예

1. 평균 평점이 4.0 이상인 영화의 정보를 평점 순으로 검색하라.
2. 직업이 engineer 인 사용자들에 의한 평균 평점이 높은 순으로 영화 정보를 검색하되 영화 제목 순으로 검색하라.
3. Comedy 장르 영화 중에서 평균 평점이 4.5 이상인 영화 정보를 vote가 많은 순으로 검색하라.

- 검색결과: 영화 ID, 제목, 장르, 평균평점, Vote 수

4. 검색 인터페이스 화면 개념도

장르:

사용자의 직업:

평점: MIN MAX

Sorting by:

Search Movie

- 모든 조건을 항상 선택할 필요는 없고, 일부 조건만 가지고 검색이 가능하도록 만들 것.

- 시연방법
:최소한 5가지 입력조건(검색 조건과 정렬 조건의 조합)을 가지고, 검색결과를 시연할 것.

과제 제출

- 과제 결과는 문서 및 동영상 형식의 2가지로 제출하여야 함
- **문서 보고서**: 파워포인트 혹은 워드/한글 등으로 작성
 - 각 단계별로 E/R 다이어그램, SQL/Python 소스코드, 개발 내용 설명, 시연 결과 등 포함할 것
 - 동영상 시연을 볼때, 부가적으로 참고할 수 있는 자료를 넣을 것.
- **동영상 보고서**: 단계별 개발 내용 설명 및 시연을 포함하여 유튜브 동영상을 제작하고, 유튜브의 동영상 링크를 알려주면 됨.
- 제출 마감: 6월 30일 자정까지
- 팀과제가 아니고, 개인 과제로 진행