프로젝트 공지

406.426B 데이터관리와 분석

2023년 1학기

본 과목의 프로젝트는 총 2회로 이루어져 있다. 1차 프로젝트에서는 주어진 requirement들을 만족하는 데이터베이스를 설계 및 구현, 2차 프로젝트에서는 이를 기반으로 한 DB 마이닝과 추천 시스템 구현을 목적으로 한다.

프로젝트는 다음과 같다.

Project #1) Conceptual DB design & DB implementation

팀 구성일: 4월 6일, 발표일: 4월 20일

Project #2) DB mining & Recommendation system

팀 구성일: 5월 11일, 발표일: 5월 25일

본 프로젝트는 팀별로 진행되며, 각 팀은 매 프로젝트마다 자율적으로 3~5명으로 구성한다. 단, 팀원을 구하지 못하는 경우 충원을 희망하는 팀에 임의 배정하거나 팀을 구성하지 못한 인원으로 팀을 임의로 구성한다.

프로젝트 발표는 5분 이하의 발표 동영상으로 갈음한다. 결과물 제출은 발표일 전날 23시 59분까지 보고서와 발표 자료 및 동영상을 ETL에 업로드해야 한다.

Project #1: Conceptual DB design & DB implementation

사이트 A는 애니메이션 전문 OTT 서비스로, 사용자가 해당 사이트를 통해 애니메이션을 시청하고, 평점을 매기는 등의 서비스를 제공한다. 본 프로젝트는 사이트 A가 사용하는 DB의 ER diagram 도식화와 DB 구현을 목적으로 하며, 크게 두 부분으로 나뉜다.

PART Ⅰ. ER diagram 도식화

PART Ⅱ. DB 구현 및 데이터 입력

PART Ⅰ. ER diagram 도식화

PART Ⅰ는 사이트 A가 사용하는 DB에 대한 ER diagram을 도식화하는 것을 목표로 한다. 사이트 A의 DB는 아래의 requirement들을 만족해야 한다.

Relationship Constraints의 경우 (min, max) notation이 아닌 cardinality ratio/participation constraints notation을 사용하여 표시하여야 한다.

(R1-1) 사이트 A는 사이트에 가입된 사용자에 대한 정보를 저장하고 있다. 사용자가 사이트 A에서 활동한 기록들은 모두 저장되며, 사용자는 애니메이션을 시청하거나 평점을 매길 수 있다. 사용자에 대한 정보로는 사이트에서 부여한 고유번호, 사용자 이름이 있으며, 사용자가 보낸 메일의 수, 사용자가 시청한 애니메이션의 수, 사용자가 평점을 매긴 애니메이션의 수가 파생 정보로 관리되어야 한다.

(R1-2) 사이트 A는 사용자의 접근 이력에 따라 각 애니메이션에 대한 시청 상태를 결정하며, 총 6가지 상태가 존재한다. 이 중 6번째의 상태는 시청 예정인 상태로 아직 시청을 하지 않은 상태이다. 사용자의 애니메이션 평점 이력에 대한 정보는 사용자의 고유번호, 애니메이션의 고유번호, 평점이 저장되며, 사용자의 애니메이션 접근 이력에 대한 정보는 사용자의 고유번호, 애니메이션의 고유번호, 사용자의 시청 상태, 사용자의 애니메이션 시청 회차수가 저장된다.

(R1-3) 사이트 A는 사이트에 등록된 애니메이션에 대한 정보를 저장하고 있다. 애니메이션에 대한 정보로는 사이트에서 부여한 고유번호, 애니메이션 이름, 애니메이션의 원작에 따른 분류, 애니메이션의 방영 시기, 애니메이션의 제작 스튜디오가 저장되며, 방영 시기로부터 primiered(첫 방영 연도와 계절)가 파생 정보로 관리되어야 한다. 각 시청 상태에 따른 사용자들의 수와 해당 애니메이션을 시청 또는 시청 예정인 사용자 수의 총합, 사용자들이 매긴 평점의 평균이 파생 정보로 관리되어야 한다.

(R1-4) 사이트 A는 각 애니메이션을 프로듀싱한 프로듀서에 대한 정보를 저장하고 있다. 프로듀서에 대한 정보로는 사이트에서 부여한 고유번호, 프로듀서 이름, 프로듀싱한 애니메이션 고유번호가 저장되며, 프로듀서가 프로듀싱한 애니메이션의 수가 파생 정보로 관리되어야 한다.

(R1-5) 사이트 A는 프로듀서에 소속된 디렉터에 대한 정보를 저장하고 있다. 디렉터에 저장된 정보로는 프로듀서의 고유번호, 디렉터 이름, 성별, 연령이 저장되며, 한 프로듀서에 소속된 디렉터의 이름은 서로 다르다.

(R1-6) 사이트 A는 사용자가 프로듀서에게 메일을 보낼 수 있는 시스템을 지원한다. 따라서, 사용자가 프로듀서에게 보낸 메일에 대한 정보를 저장하고 있다. 메일에 저장된 정보로는 사이트에서 부여한 고유번호, 사용자 고유번호, 프로듀서 고유번호, 내용이 저장되며, 메일 내용의 길이가 파생 정보로 관리되어야 한다.

(R1-7) 사이트 A는 각 애니메이션을 제작한 스튜디오에 대한 정보를 저장하고 있다. 스튜디오에 대한 정보로는 사이트에서 부여한 고유번호, 스튜디오 이름, 스튜디오 소속 인원, 스튜디오에서 주력하는 장르의 고유번호가 저장되며, 스튜디오가 제작한 애니메이션 수가 파생 정보로 관리되어야 한다.

(R1-8) 사이트 A는 장르에 대한 정보를 저장하고 있다. 장르에 대한 정보로는 사이트에서 부여한 고유번호, 장르 이름, 장르에 포함되는 애니메이션 고유번호가 저장되며, 각 장르에 해당되는 애니메이션의 수가 파생 정보로 관리되어야 한다.

(R1-9) 사이트 A는 라이센서에 대한 정보를 저장하고 있다. 라이센서에 대한 정보로는 사이트에서 부여한 고유번호, 라이센서 이름, 라이센서의 목적에 따른 타입, 라이센서가 라이선스를 소유하고 있는 애니메이션 고유번호가 저장된다. 라이센서 간에 라이센스를 공유할 수 있어서, 라이선스 공유에 대한 정보에는 라이선스 제공자 고유번호, 라이선스 피제공자 고유번호, 공유의 타입이 저장된다. 라이센서에 대한 정보에는 라이선스를 보유하고 있는 애니메이션의 수, 제공하고 있는 라이선스의 수, 제공받고 있는 라이선스의 수가 파생 정보로 관리되어야 한다.

PART Ⅱ. DB 구현 및 데이터 입력

PART Ⅱ는 이 사이트 A의 데이터에 적합한 데이터베이스 스키마를 설계하여 데이터베이스 테이블을 실제로 생성한 후 데이터 입력까지를 목표로 한다. 해당 프로그램은 Python과 MySQL을 사용하여 구현하여야 하며, 다음의 요구 조건들을 만족하여야 한다. Python에서 mysql-connector-python 외의 **별도 라이브러리는 사용할 수 없다.**

(R2-1) 사이트 A의 데이터를 활용하기에 앞서 이를 MySQL 상에 저장해야 한다. 이를 위해 먼저 DMA_team##의 이름을 가지는 schema를 생성해야 한다. 예를 들면, 1조의 schema 명은 DMA_team01이다. 이 때 schema가 존재할 경우 생성 과정을 다시 수행하지 않아야 한다.

(R2-2) schema를 설계한 이후에는 데이터를 저장하기 위한 table을 생성해야 한다. 생성하는 table과 column 이름과 순서는 주어진 데이터셋의 table 및 column과 일치해야 한다. 0 또는 1의 값을 가지는 column은 TINYINT(1)로, INTEGER type는 'INT(11)'로, 범위가 큰 INTEGER type은 'BIGINT(20)'로, STRING type는 'VARCHAR(255)'를 이용하여 생성한다. 그 외 날짜는 'DATE'를 통해 생성한다. 이 때 table이 존재할 경우 생성 과정을 다시 수행하지 않아야 한다. (R2-2)에서는 foreign key 조건을 작성하지 않고 (R2-3)에서 데이터 입력 후 foreign key 조건을 추가한다.

(R2-3) 생성된 table에 데이터를 저장해야 한다. 데이터는 csv파일로 주어지며 이를 직접 변형해서는 안된다.

(R2-4) 해당 데이터베이스 schema에 foreign key 조건들을 반영해주어야 한다.

채점 기준(절대평가)

- PART Ⅰ ER diagram의 requirement 만족 여부(50 %)
- PART Ⅱ 설계한 데이터베이스 스키마와 constraints(25%), requirement 만족 여부(15%)
- 보고서 품질(5%), 발표(5%)

결과물들을 'DMA_project1_team##.zip' 파일로 압축하여 발표일 전날인 **4월 19일 23:59까지** ETL에 업로드해야 한다. ETL 상에 문제가 생겼을 경우 alswo5131@snu.ac.kr 로 오류 증명 파일과 함께 제출 기한 전에 보내야 한다. 제출해야 할 결과물과 파일명, 파일 확장자는 다음과 같다.

- 보고서

  - 파일명: DMA_project1_team##_보고서.pdf
  - 보고서에는 PART Ⅰ에서의 문제 정의, 도식화한 ER diagram의 도식화 과정과 최종 ER diagram, PART Ⅱ에서의 문제 정의, 설계한 스키마(Relation Schema)와 코드에 대한 설명이 포함되어야 한다. 이 때 스키마 설계 시 constraints들과 이들의 설정 근거가 포함되어야 한다.

- 발표 자료 및 발표 동영상

  - 발표 자료 파일명: DMA_project1_team##_발표자료.pdf
  - 발표 동영상 파일명: DMA_project1_team##_발표동영상.mp4
  - 발표 동영상은 팀 당 5분 이내로 제작되어야 하며 powerpoint의 녹화 기능을 사용한다.

- Python 프로그램 코드

  - Python 코드 파일명: DMA_project1_team##.py
  - 함수들의 입력 값들의 의미는 다음과 같다.

    host, user, password: MySQL에 접근하기 위한 계정 정보

    directory: 데이터가 저장된 주소 (ex. C:/dir/user.txt → 'C:/dir/')

  - 뼈대 코드의 주석에 작성된 TODO들에 따라 팀의 번호, MySQL 계정 정보, 데이터(csv)들이 저장된 주소 등을 바꿔야 한다.
  - mysql.connector 외의 다른 패키지를 import하여 사용하는 것은 허용되지 않는다.
  - PART Ⅱ의 각 requirement(R2-1~R2-4)에 해당하는 Python 코드는 주어진 뼈대 코드의 requirement# 함수로 구현되어야 한다. 예를 들어 R2-1은 requirement1 함수에 구현되어야 한다.