프로젝트 #2: DB Implementation & Query Processing

본 프로젝트는 사이트 A의 데이터에 적합한 데이터베이스 스키마를 설계하여 데이터베이스 테이블을 실제로 생성한 후 데이터를 입력하고 활용하는 프로그램을 구현하는 것을 목적으로 한다. 해당 프로그램은 Python과 MySQL을 사용하여 구현하여야 하며, 다음의 요구 조건들을 만족하여야 한다.

R1)

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.

Schema를 설계한 이후에는 데이터를 저장하기 위한 table을 생성해야 한다. 생성하는 table의 column들의 순서, 자료형, 제약은 data_explanation.xlsx를 따라야 한다.

생성된 table에 데이터를 저장해야 한다. 주어진 csv 파일들의 파일 명은 table 명을 의미하며, csv 파일들의 첫 번째 줄은 column 명이고 그 외는 data이다. csv 파일들에 저장된 column의 순서는 위에서 생성한 table의 column의 순서와 일치한다(즉, data_explanation.xlsx의 순서와 일치함). table에 입력된 데이터는 csv 파일의 데이터와 일치해야 하며, csv 파일을 직접 변형해서는 안 된다.

프로그램 실행 시 schema 또는 table이 이미 존재할 경우 schema, table의 생성과 데이터 저장을 다시 수행하지 않아야 한다.

R2)

사이트 A는 연도별 사용자들의 ~~유입량과~~ 활동량의 추이를 확인하고자 한다. 이를 위해 연도별로 사이트 A에서 새로이 그룹에 가입한 사용자의 수, 생성된 그룹의 수, 개최된 오프라인 모임의 수를 출력하라. 출력된 결과는 연도에 따라 오름차순으로 정렬되어야 하며, column의 순서는 연도, 해당 연도에 새로이 그룹에 가입한 사용자의 수, 해당 연도에 생성된 그룹의 수, 해당 연도에 개최된 오프라인 모임의 수여야 한다.

R3)

사이트 A는 호평을 받은 것에 비해 오프라인 모임 개최 횟수가 적은 장소를 소개하는 이벤트를 가지고자 한다. 이를 위해 5회 이상의 리뷰를 받았고 평균 평점은 4.5이상이지만 오프라인 모임이 3회 이하로 개최된 장소들을 찾아 개최된 장소의 id, 도시 id, 장소 명, 주소를 출력하라. 출력된 결과는 id에 따라 오름차순으로 정렬되어야 하며, column의 순서는 id, 도시 id, 장소 명, 주소여야 한다.

R4)

사이트 A는 활동을 많이 하는 가입자들을 상대로 이벤트를 준비하고자 한다. 이를 위해 1년 이상 활동한 그룹이 5개 이상인 가입자들을 찾아 가입자의 id, 도시 id, 이름을 출력하라. 활동 기간은 마지막 방문 날짜와 가입한 날짜 차이이다. 출력된 결과는 id에 따라 오름차순으로 정렬되어야 하며, column의 순서는 id, 도시 id, 이름이어야 한다.

R5)

사이트 A는 등록은 되어 있지만 그룹원이 하나도 없는 그룹들을 정리하고자 한다. 그룹원이 하나도 없고 2014년 이전에 생성된 그룹은 삭제하고자 하지만, 2014년 이후에 생성된 그룹은 홍보하고자 한다. 이를 위해 R1)에서 생성한 table인 gatherings에 remarks라는 새로운 column을 추가한다. 이 때, remarks의 데이터형은 VARCHAR(20)이며 제약 사항은 없다. 만약 그룹원이 한 명이라도 있으면 해당 그룹의 remarks는 비어있다(null). 2014년 이전에 생성된 그룹의 remarks에는 “to be deleted”를, 2014년 이후에 생성된 그룹은 remarks에 “need promotion”을 저장한다. 이후, 그룹원이 없는 그룹들을 찾아 그룹의 id, 그룹명, 생성 날짜, remarks를 출력하라. 출력된 결과는 id에 따라 오름차순으로 정렬되어야 하며, column의 순서는 id, 그룹명, 생성 날짜, remarks여야 한다.

※csv 파일을 직접 변형해서는 안 된다.

R6)

(자유주제) R2~R5와 같이 사이트 A가 마주할 수 있는 문제를 하나 정의하라. 그 후, 문제를 해결하기 위한 쿼리를 제시하고 그 결과를 출력하라. 단, 정의된 문제는 R2 ~ R5와 중복되어서는 안 된다.

제출하는 Python 코드에 대한 제약은 다음과 같다.

- 각 조는 DMA_project2_team##.py 파일을 제출해야 한다. 예를 들면 1조의 파일 명은 DMA_project2_team01.py이다.
- 함수들의 입력 값들의 의미는 다음과 같다.

  host, user, password: MySQL에 접근하기 위한 계정 정보

  directory_in: 데이터가 저장된 주소 (ex. C:/dir/user.txt → 'C:/dir/')
- 뼈대 코드의 주석에 작성된 TODO들을 따라 팀의 번호, MySQL 계정 정보, 데이터(csv)들이 저장된 주소 등을 바꿔야 한다.
- mysql.connector 외의 다른 패키지를 import하여 사용하는 것은 허용되지 않는다.
- 각 requirement(R1~R6)에 해당하는 Python 코드는 주어진 뼈대 코드의 requirement# 함수로 구현되어야 한다. 예를 들면 R1는 requirement1 함수에 구현되어야 한다.
- R2~R6의 결과는 각각 project2_team##_req#.txt에 저장되어야 한다. 예를 들면 1조의 R1의 결과는 project2_team01_req1.txt에 저장되어야 한다.
- R2~R6의 결과를 저장할 때, column들 사이의 분리 기호는 세미콜론(;)이어야 하며, row들 사이에는 줄 넘김(\n)으로 분리되어야 한다.
- R2~R4, R6는 nested query를 이용하여 하나의 requirement 당 하나의 SQL 문장 만으로 작성되어야 한다.
- R5는 최대 3개의 SQL 문장으로 작성되어야 한다.
- requirement2~requirement6 함수는 각각 3분 이내로 결과를 출력해야 한다.

python 프로그램 코드, 보고서와 발표 자료를 압축하여 발표일(11월 14일) 00:00까지 ETL에 업로드 해야 한다. 발표 당일에 사용하는 발표 자료와 ETL에 제출된 내용은 정확히 같아야 한다.

제출하는 압축 파일, python 프로그램 코드, 보고서, 발표자료는 각각 파일명을 'DMA_project2_team##.zip', 'DMA_project2_team##.py' 'DMA_project2_team##_보고서', 'DMA_project2_team##_발표자료'으로 가져야 하며, 발표 자료는 pdf 파일이어야 한다.

※ 보고서 및 발표 자료 하드 카피는 제출하지 않으셔도 됩니다.

채점 기준(절대 평가):

- 설계된 DB 및 구현된 프로그램의 requirements 만족 여부: 90%

  R1~R6 각각 15%
- 보고서 품질: 5%
- 발표: 5%