

Project: Implementing a Simple Database Application

Due: 2020/06/22 (Mon), 11:59 PM

이번 프로젝트의 목표는 데이터베이스를 이용하는 간단한 어플리케이션을 설계하고 구현하는 것이다.

Python 및 pymysql을 이용하여 공연 티켓 예매를 시뮬레이션 하는 간단한 어플리케이션을 만들어 본다. 이번 프로젝트를 통해 학생들은 어플리케이션과 데이터베이스를 연동하는 방법을 배우게 된다.

1 요구 사항

- 데이터는 관계형 데이터베이스에 저장되어야 한다.
- 각 컬럼의 타입은 2장에 명시된 대로 정의하여야 한다.
- 공연장, 공연 데이터 및 관객 데이터를 추가/삭제할 수 있어야 한다.
- 공연장, 공연 데이터 및 관객 데이터를 출력할 수 있어야 한다.
- 공연을 공연장에 배정할 수 있어야 한다.
- 관객이 공연을 예매할 수 있어야 한다.
- 공연장 별로 공연장에 배정된 공연의 정보를 출력할 수 있어야 한다.
- 공연 별로 공연을 예매한 관객의 명단을 출력할 수 있어야 한다.
- 공연 별로 좌석 별 예매 현황을 출력할 수 있어야 한다.

2 데이터 타입

1. 공연장
 - 공연장 ID: 정수
 - 이름: 문자열 (최대 200자)
 - 위치: 문자열 (최대 200자)
 - 정원: 정수 (1 이상의 값)
2. 공연
 - 공연 ID: 정수
 - 이름: 문자열 (최대 200자)
 - 종류: 문자열 (최대 200자)

- 가격: 정수 (0 이상의 값)
- 3. 관객
 - 관객 ID: 정수
 - 이름: 문자열 (최대 200자)
 - 성별: 문자열 ('M', 'F')
 - 나이: 정수 (1 이상의 값)

3 세부 사항

구현한 프로그램은 다음과 같은 기능을 할 수 있어야 한다.

1. 모든 공연장 정보 출력
 - 모든 공연장의 정보를 출력한다.
 - 공연장 ID, 이름, 위치, 정원, 열리는 공연의 수 순으로 출력
2. 모든 공연 정보 출력
 - 모든 공연의 정보를 출력한다.
 - 공연 ID, 이름, 종류, 가격, 예매자 수 순으로 출력
3. 모든 관객 정보 출력
 - 모든 관객의 정보를 출력한다.
 - 관객 ID, 이름, 성별, 나이 순으로 출력
4. 공연장 추가
 - 새로운 공연장을 추가한다.
 - 이름(String), 위치(String), 정원(int)을 입력 받음
5. 공연장 삭제
 - 공연장을 삭제한다.
 - 공연장 ID(int)를 입력 받음
 - 공연장을 삭제하면 그 공연장의 공연 배정 정보와 해당 공연에 관련된 예매 정보도 모두 삭제되어야 함
 - 해당 공연장이 없다면 에러 메시지 출력
6. 공연 추가

- 새로운 공연을 추가한다.
 - 이름(String), 종류(String), 가격(int)를 입력 받음

7. 공연 삭제

- 공연을 삭제한다.
 - 공연 ID(int)를 입력 받음
 - 공연을 삭제하면 관련된 예매 정보도 모두 삭제되어야 함
 - 해당 공연이 없다면 에러 메시지 출력

8. 관객 추가

- 새로운 관객을 추가한다.
 - 이름(String), 성별(String), 나이(int)를 입력 받음

9. 관객 삭제

- 관객을 삭제한다.
 - 관객 ID(int)를 입력 받음
 - 관객을 삭제하면 관련된 예매 정보도 모두 삭제되어야 함
 - 해당 관객이 없다면 에러 메시지 출력

10. 공연 배정

- 공연장에 공연을 배정한다.
 - 공연장 ID(int) 및 공연 ID(int)를 입력 받음
 - 하나의 공연장에 대해서 여러 개의 공연을 배정할 수 있음
 - 공연이 이미 다른 공연장에 배정되었다면 에러 메시지 출력

11. 공연 예매

- 관객을 공연에 예매 시킨다.
 - 공연 ID(int), 관객 ID(int), 좌석 번호 리스트(int list)를 입력 받음
 - 한 관객은 여러 개의 공연을 예매할 수 있음
 - 단, 공연 예매에 영향을 미치는 조건(예. 나이)은 예매자의 조건을 따른다.
 - 한 관객은 한 공연의 여러 좌석을 예매할 수 있음
 - 공연의 좌석 번호는 정수(int)이며, 1에서부터 해당 공연이 배정된 공연장의 정원 값 까지를 가질 수 있음 (예: 공연이 배정된 공연장의 정원이 100이라면, 좌석 번호는 1부터 100까지의 정수 값을 가질 수 있음)

- 좌석 번호 리스트의 좌석 번호 값들 중 하나라도 범위를 벗어난다면 에러 메시지 출력
- 좌석 번호 리스트에서 각 좌석 번호는 콤마(,)로 구분됨 (예: 1, 3, 5)
- 좌석 번호 리스트의 좌석 번호 중 하나라도 이미 예매되어 있다면 예매에 실패함
- 예매에 성공했다면 총 티켓 가격 출력
 - 총 티켓 가격은 티켓 수 X 티켓 하나당 가격으로 계산됨
 - 티켓 하나당 가격은 나이에 따라 달라짐
 - 1~7세: 무료
 - 8~12세: 50% 할인
 - 13세~18세: 20% 할인
 - 19세 이상: 할인 없음
 - 계산된 총 티켓 가격은 소수점 첫째자리에서 반올림
- 공연이 공연장에 배정되지 않았다면 에러 메시지 출력

12. 공연장에 배정된 공연 목록 출력

- 공연장에 배정된 모든 공연의 정보를 출력한다.
 - 공연장 ID(int)를 입력 받음
 - 해당 공연장에 배정된 공연의 정보를 출력한다
 - 공연 ID, 이름, 종류, 가격, 예매 좌석수 순으로 출력
 - 해당 공연장이 없다면 에러 메시지 출력

13. 공연을 예매한 관객 정보 출력

- 공연을 예매한 모든 관객의 정보를 출력한다.
 - 공연 ID(int)를 입력 받음
 - 해당 공연을 예매한 관객의 정보를 중복 없이 출력한다.
 - 관객 ID, 이름, 성별, 나이 순으로 출력
 - 해당 공연이 없다면 에러 메시지 출력

14. 공연의 좌석 별 예매 상황 출력

- 공연의 각 좌석 별로 예매한 관객의 정보를 출력한다.
 - 공연 ID(int)를 입력 받음
 - 해당 공연의 좌석 번호와 그 좌석을 예매한 관객의 ID를 출력한다.

- 예매되지 않은 좌석 번호도 결과에 나와야 함
- 해당 공연이 없다면 에러 메시지 출력
- 공연이 공연장에 배정되지 않았다면 에러 메시지 출력

15. 프로그램 종료

16. 데이터베이스 리셋 및 생성

- 기존 모든 테이블 및 데이터가 존재할 경우 삭제
- 2장에서 정의한 데이터 타입에 해당하는 새로운 테이블 생성
- 삭제 실시 전 확인 메시지를 띄우고 y/n 사용자 입력을 받아야 함

기타 사항

- 입력 시 각 입력의 데이터 타입은 지켜진다고 가정 (예: 정원의 입력으로 String이 들어오지 않음)
- 공연장 ID, 공연 ID, 관객 ID는 공연장, 공연, 관객이 삽입될 때마다 자동 부여
- 1부터 시작
- AUTO_INCREMENT 옵션을 사용할 것
- 입력이 데이터 타입에 정의된 값의 범위를 벗어난다면 에러 메시지 출력
- 단, 문자열의 경우 최대 길이(200자)보다 길다면 200자로 truncate할 것
- 여러 개의 입력을 받는 경우, 입력을 차례로 받다가 특정 입력에 대해 에러가 있을 시 그 즉시 에러 메시지를 출력하고 해당 명령을 종료함
- 모든 출력은 첫 번째 컬럼에 대해 오름차순으로 정렬되어야 함

4 실행 예시

```
=====
1. print all buildings
2. print all performances
3. print all audiences
4. insert a new building
5. remove a building
6. insert a new performance
7. remove a performance
8. insert a new audience
9. remove an audience
10. assign a performance to a building
11. book a performance
12. print all performances which assigned at a building
13. print all audiences who booked for a performance
```

14. print ticket booking status of a performance

15. exit

16. reset database

=====

Select your action: 1

id	name	location	capacity	assigned
1	Seoul Arts Center	Seoul	5	1
2	Grand Peace Palace	Seoul	3	1

Select your action: 2

id	name	type	price	booked
1	Coldplay Concert	Concert	100000	3
2	Jekyll & Hyde	Musical	70000	2

Select your action: 3

id	name	gender	age
1	Bae Sangwhan	M	15
2	Kim Jihoon	M	30
3	Choi Byungseo	M	56

Select your action: 4

Building name: Sejong Art Center

Building location: Seoul

Building capacity: 6

A building is successfully inserted

Select your action: 6

Performance name: Romeo and Juliet

Performance type: Drama

Performance price: 50000

A performance is successfully inserted

Select your action: 8

Audience name: Shin Yoohyun

Audience gender: F

Audience age: 19

An audience is successfully inserted

Select your action: 1

id	name	location	capacity	assigned
1	Seoul Arts Center	Seoul	5	1
2	Grand Peace Palace	Seoul	3	1
3	Sejong Art Center	Seoul	6	0

Select your action: 2

id	name	type	price	booked
----	------	------	-------	--------

1	Coldplay Concert	Concert	100000	3
2	Jekyll & Hyde	Musical	70000	2
3	Romeo and Juliet	Drama	50000	0

Select your action: 3

id	name	gender	age
1	Bae Sangwhan	M	15
2	Kim Jihoon	M	30
3	Choi Byungseo	M	56
4	Shin Yoohyun	F	19

Select your action: 10
 Building ID: 3
 Performance ID: 3
 Successfully assign a performance

Select your action: 12
 Building ID: 3

id	name	type	price	booked
3	Romeo and Juliet	Drama	50000	0

Select your action: 11
 Performance ID: 3
 Audience ID: 4
 Seat number: 2, 4
 Successfully book a performance
 Total ticket price is 100,000

Select your action: 11
 Performance ID: 3
 Audience ID: 3
 Seat number: 1, 2
 The seat is already taken

Select your action: 13
 Performance ID: 3

id	name	gender	age
4	Shin Yoohyun	F	19

Select your action: 14
 Performance ID: 3

seat_number	audience_id
1	
2	4
3	
4	4
5	
6	

Select your action: 15
Bye!

5 개발 환경

- Python 3.5+
- MariaDB
 - ◆ DB 접속 정보
 - Host: astronaut.snu.ac.kr
 - Port: 3306
 - ID: 20DB_학번(예:20DB_2020_12345)
 - Password: ID와 동일
 - Default Schema: ID와 동일

6 채점

1. 구현 완성도 (80%)
 - 기능 완성도 (60%)
 - **1단계: 데이터베이스 리셋 및 스키마 생성 (10%)**
 - **해당 문제: 15**
 - 데이터베이스에 있는 데이터와 테이블이 모두 삭제되었는지
 - 설계한 스키마대로 테이블에 create되는지
 - **2단계: 데이터의 단순 입출력 (20%)**
 - **해당 문제: 1 ~ 9**
 - 데이터를 DB로부터 정확하게 retrieve하여 표시할 수 있는지
 - 주어진 인풋대로 데이터 레코드를 생성하여 DB에 저장하고 있는지
 - 지정한 데이터 레코드를 삭제할 수 있는지
 - **3단계: DB와 Application의 복합적 연동 (30%)**
 - **해당 문제: 10 ~ 14**
 - 잘 설계된 데이터베이스 스키마와 integrity constraint를 활용하여

명시된 기능을 잘 수행하도록 구현하였는지

- 적합한 데이터베이스 스키마 설계 (10%)
- 청결한 코드 (10%)
 - 적절한 모듈화 및 주석 첨가로 코드 가독성이 확보되었는지

2. 레포트 (20%)

- 하루 Delay마다 -10%, 3일 Delay 시 0점 처리

7 제출

1. Runnable python script 파일

- 파일명: main.py
- 다수의 스크립트 파일을 생성하는 것은 무관하나, 반드시 “python main.py” 커맨드로 프로그램이 수행 가능해야함.

2. DB

- (중요!) 프로젝트 제출시 채점을 위해 자신의 DB 계정 안의 테이블 스키마만 남기고 레코드는 모두 truncate해야 함
- 이를 지키지 않았을 시 채점이 제대로 되지 않아 0점 처리 될 수 있으니 주의!

3. 리포트

- 파일명: PRJ2_학번.docx (예: PRJ2_2020-12345.docx)
- 포함되어야 하는 내용
 - ◆ 핵심 모듈과 알고리즘에 대한 설명
 - ◆ 구현한 내용에 대한 간략한 설명
 - ◆ (제시된 요구사항 중 구현하지 못한 부분이 있다면) 구현하지 못한 내용
 - ◆ 가정한 것들
 - ◆ 컴파일과 실행 방법
 - ◆ 프로젝트를 하면서 느낀 점

- 제출 파일을 압축하여 etl로 제출

- 파일명: PRJ2_학번.zip (예: PRJ2_2020-12345.zip)

8 기타 사항

1. 프로젝트 관련 문의는 6/19 금요일까지만 가능

9 References

- MariaDB
 - <https://mariadb.org/>
- pymysql
 - <https://readthedocs.org/projects/pymysql/>
- MySQL Workbench
 - <https://dev.mysql.com/downloads/workbench/>

* 접속 후 password 변경 방법

```
set password = password('비밀번호')
```