

Project 2

2015-10033 김다운

■ 프로젝트 개요

maria DB 데이터베이스를 이용해 공연 티켓 예매라는 간단한 어플리케이션을 만든다. 구현을 위해 데이터베이스 스키마를 직접 설계하고 JDBC를 이용하여 작성한 프로그램이 데이터베이스에 연결되도록 한다. 이를 통해 어플리케이션과 데이터베이스 연동하는 방법을 배운다.

■ 핵심 모듈과 알고리즘

◎ 데이터베이스

relational database schema로 나타내면 다음과 같다.

Building	Matching	Performance	Reservation	Audience
<ul style="list-style-type: none">• <u>id</u>• name• location• capacity	<ul style="list-style-type: none">• <u>b.id</u>• <u>p.id</u>	<ul style="list-style-type: none">• <u>id</u>• name• type• price	<ul style="list-style-type: none">• <u>b.id</u>• <u>p.id</u>• <u>seat.id</u>• status• <u>a.id</u>	<ul style="list-style-type: none">• <u>id</u>• name• gender• age

밑줄과 기울어진 attribute는 각 table의 Primary key를 의미한다.

각 색깔은 동일한 외래키와 참조키 관계를 의미하며 진하게 표시된 attribute가 해당 관계에서의 참조키를, 그렇지 않은 attributes는 외래키를 의미한다.

◎ 어플리케이션

reserve.java : 커맨드를 선택하고 입출력을 수행하는 main 함수가 존재한다. 커맨드에 맞는 DAO 클래스의 메소드를 호출한다.

DAO.java : 데이터베이스에 접근하여 원하는 쿼리를 수행하는 클래스이다.

■ 구현 내용

DAO.java 내의 쿼리들이 어플리케이션과 데이터베이스를 연결하는데 있어서 반드시 필요한 요소이다. 14종류의 각 커맨드마다 필요한 쿼리를 작성하였고 이 중 공통으로 필요한 쿼리는 별도의 함수(buildingIsExist(), performanceIsExist(), AudienceIsExist())로 작성하였다.

■ 가정한 것들

◎ 삽입되는 공연장, 공연, 사람은 항상 다르다고 가정했다. 즉, 공연장의 이름과 장소, 수용인원이 같더라도 반드시 다른 공연장이라고 가정하였다. 공연의 이름과 종류, 가격이 같더라도 반드시 다른 공연이라고 가정하였다. 관객의 이름과 성별, 나이가 같더라도 반드시 다른 관객이라고 가정하였다.

■ 컴파일과 실행 방법

1. 실행

```
$cd [jar file location]
```

```
$java -jar [filename].jar
```

2. 종료

```
$exit;
```

■ 프로젝트를 하면서 느낀 점

지난 프로젝트들보다는 훨씬 간단하게 구현할 수 있었다. 그러나 정확하고 효율적인 쿼리 구문을 작성해야 하는데 여러 번 테스트를 해서 검증 해야 했기 때문에 시간이 필요했다. 지금까지 데이터베이스를 설계해보는 연습을 했는데 이것을 어플리케이션을 짜고 직접 연동해보는 과정을 연습해보고 나니 데이터베이스의 중요성을 깨달았다.