

|  |
| --- |
| **데이터베이스 프로젝트 최종보고서** |
| **- 온라인 영화 예약 -** |





|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **제출일** | 2024.06.07 |  | **학과** | 컴퓨터공학과 |
| **강의명** | 데이터베이스 |  | **분반** | 001 |
| **교수님** | 신동일 교수님 |  | **팀원** | 20011675 류태호  20011556 송현준 |

**목차**

1. 개요
   * 1. 프로젝트 소개
     2. 프로젝트 목표
2. 요구사항 분석
   * 1. 로그인 화면
     2. 관리자 기능
     3. 회원 기능
     4. 오류 처리
3. 스키마 정의
   * 1. 테이블 설명
     2. EER Diagram
4. 프로그램 구조
5. 테스팅
6. 개요
   * 1. 프로젝트 소개

이 프로젝트는 온라인 영화 예약 시스템의 데이터베이스를 구축하여 사용자가 영화를 예매하고 관리자가 시스템을 관리할 수 있는 효율적인 환경을 제공하는 것을 목표로 합니다.

* + 1. 프로젝트 목표
* 영화 예약 시스템의 데이터베이스를 구축하여 영화, 상영 일정, 상영관, 티켓, 회원 등의 정보를 효율적으로 관리합니다.
* 관리자와 회원을 구분하여 각각의 기능을 제공하고 GUI를 통해 사용자가 편리하게 시스템을 이용할 수 있도록 합니다.
* 데이터베이스 초기화, 입력 기능, 오류 처리 등을 구현하여 시스템의 안정성을 보장하고 사용자 경험을 향상시킵니다.

1. 요구사항 분석
   * 1. 로그인 기능

* 최초의 로그인 화면에서 관리자 / 회원을 선택해서 로그인하며 로그인 시 관리자는 root/1234 계정으로, 회원은 user1/user1 계정으로 접속합니다.
* 회원은 user1으로만 로그인한다고 가정합니다.
* 접속 시 현재날짜는 2024년 1월 1일로 가정합니다.
  + 1. 관리자 기능
* 관리자로 로그인 시 연결 성공 여부를 표시하며 데이터베이스 초기화, 전체 테이블 조회, 테이블에 대한 입력/삭제/변경 기능을 구현합니다.
* 데이터베이스 초기화 시 테이블을 생성하고 샘플데이터를 삽입합니다.
* 테이블에 대한 입력 기능 수행 시 입력할 테이블을 선택 후 해당 테이블의 속성을 입력합니다.
* 입력은 각 속성의 자료형에 맞게 입력된다고 가정합니다.
* 테이블에 대한 삭제 기능 수행 시 삭제할 테이블을 선택 후 해당 테이블에서 삭제할 조건을 입력합니다.
* 테이블에 대한 변경 기능 수행 시 변경할 테이블을 선택 후 해당 테이블에서 변경할 값과, 조건을 입력합니다.
* 관리자는 데이터베이스의 모든 값을 임의로 수정할 수 있다고 가정합니다.
  + 1. 회원 기능
* 회원으로 로그인 시 연결 성공 여부를 표시하며 모든 영화 조회, 예매한 영화 조회 기능을 구현합니다.
* 영화 조회 기능 수행 시 영화명, 감독명, 배우명, 장르 중 하나 이상을 입력받고 해당 영화를 검색합니다.
* 조회된 영화를 선택 후 예매버튼을 누르면 해당 영화의 상영일정을 조회합니다.
* 조회된 상영일정 중 선택 후 예매버튼을 누르면 해당 상영일정의 좌석을 보여줍니다.
* 좌석을 선택하고 확인버튼을 누르면 결제창이 나옵니다.
* 결제창에서 결제방법과 결제상태 입력 후 티켓 발권 여부를 선택 후 확인버튼을 누르면 예매가 완료됩니다.
* 예매한 영화 조회 버튼을 누르면 예매내역을 출력합니다.
* 예매내역을 선택 후 삭제/영화변경/일정변경/추가정보 조회가 가능합니다.
* 삭제버튼을 누를 시 해당 예매내역은 삭제됩니다.
* 영화변경을 누를 시 상영중인 영화 목록을 조회하며, 예매 과정과 동일하게 영화 변경을 진행합니다.
* 일정변경을 누를 시 해당 영화의 상영일정을 조회하며, 예매 과정과 동일하게 일정 변경을 진행합니다.
* 추가정보조회를 누를 시 해당 예매의 영화에 해당하는 모든 상영일정과 상영관을 보이고, 해당 예매의 티켓에 해당하는 추가정보를 볼 수 있습니다.
  + 1. 오류 처리
* 진행 중 오류 발생 시 Utils 클래스의 showMessage메소드를 사용하여 사용자에게 새 창으로 띄워 출력합니다.

1. 스키마 정의
   * 1. 테이블 설명

*영화 요구사항에 따른 데이터베이스 테이블*

제약조건 : 하나의 영화에 여러명의 배우가 출연, 한명의 배우가 여러개의 영화에 출연. 따라서 Actor테이블을 추가로 만들어서 Movie테이블과 연관관계를 만들어준다.

**영화 테이블 (Movies)**

* **설명**: 영화 정보를 저장하는 테이블입니다.
* **컬럼**:
  + MovieID (INT, Primary Key, Auto Increment): 각 영화의 고유 식별자입니다.
  + Title (VARCHAR(255), Not Null): 영화 제목입니다.
  + MovieTime (INT, Not Null): 영화의 상영 시간(분)입니다.
  + Rating (VARCHAR(255)): 영화의 등급입니다.
  + Director (VARCHAR(255)): 영화 감독입니다.
  + Genre (VARCHAR(255)): 영화 장르입니다.
  + Introduction (VARCHAR(255)): 영화의 소개글입니다.
  + ReleaseDate (DATE): 영화의 개봉일입니다.
  + Score (DECIMAL(2,1)): 영화의 평균 점수입니다.

**배우 테이블 (Actors)**

* **설명**: 배우 정보를 저장하는 테이블입니다.
* **컬럼**:
  + ActorID (INT, Primary Key, Auto Increment): 각 배우의 고유 식별자입니다.
  + ActorName (VARCHAR(255)): 배우의 이름입니다.

**영화배우 테이블 (MovieActors)**

* **설명**: 영화와 배우 간의 다대다 관계를 관리하는 테이블입니다.
* **컬럼**:
  + MovieID (INT, Foreign Key): 영화의 ID입니다. Movies(MovieID)를 참조합니다.
  + ActorID (INT, Foreign Key): 배우의 ID입니다. Actors(ActorID)를 참조합니다.
  + **Primary Key**: 복합 키 (MovieID, ActorID).

*상영일정 요구사항에 따른 데이터베이스 테이블*

제약조건 : 모든 영화는 1개~4개 이하의 상영일정을 가짐. 같은 MovieId를 가지는 상영일정이 테이블에 최대 4개까지만 존재하게 제한

**상영일정 테이블 (Screenings)**

* **설명**: 영화 상영 정보를 저장하는 테이블입니다.
* **컬럼**:
  + ScreeningID (INT, Primary Key, Auto Increment): 각 상영의 고유 식별자입니다.
  + MovieID (INT, Foreign Key): 상영하는 영화의 ID입니다. Movies(MovieID)를 참조합니다.
  + TheaterID (INT, Foreign Key): 상영이 이루어지는 극장의 ID입니다. Theaters(TheaterID)를 참조합니다.
  + ScreeningStartDate (DATE): 상영 시작 날짜입니다.
  + ScreeningDate (DATE): 특정 상영 날짜입니다.
  + SessionNumber (INT): 상영 세션 번호입니다.
  + StartTime (TIME): 상영 시작 시간입니다.

*상영관 요구사항에 따른 데이터베이스 테이블*

제약조건 : 각 상영관에는 1개 이상의 상영일정 배정은 데이터베이스 초기화 후 샘플데이터 삽입할 때 상영관마다 상영일정 한 개씩 가지게 배정, 이후 관리자가 상영일정을 삭제하면 상영일정이 없는 상영관이 존재 가능. 상영관 사용 여부는 상영일정에 해당 상영관이 존재하는지에 따라 사용여부 True False 결정.

**상영관 테이블 (Theaters)**

* **설명**: 상영관 정보를 저장하는 테이블입니다.
* **컬럼**:
  + TheaterID (INT, Primary Key): 각 상영관의 고유 식별자입니다.
  + NumberOfSeats (INT): 상영관의 총 좌석 수입니다.
  + HorizontalSeats (INT): 가로 좌석 수입니다.
  + VerticalSeats (INT): 세로 좌석 수입니다.

**상영관 사용 테이블 (TheaterUse)**

* **설명**: 상영관과 상영일정 간의 관계를 관리하는 테이블입니다.
* **컬럼**:
  + TheaterID (INT, Foreign Key): 극장의 ID입니다. Theaters(TheaterID)를 참조합니다.
  + ScreeningID (INT, Foreign Key): 상영의 ID입니다. Screenings(ScreeningID)를 참조합니다.
  + TheaterUse (BOOLEAN): 특정 상영을 위해 극장이 사용되는지 여부를 나타냅니다.
  + **Primary Key**: 복합 키 (TheaterID, ScreeningID).

*티켓 요구사항에 따른 데이터베이스 테이블*

**티켓 테이블 (Tickets)**

* **설명**: 상영을 위한 발권된 티켓 정보를 저장하는 테이블입니다.
* **컬럼**:
  + TicketID (INT, Primary Key, Auto Increment): 각 티켓의 고유 식별자입니다.
  + ScreeningID (INT, Foreign Key): 티켓이 발권된 상영의 ID입니다. Screenings(ScreeningID)를 참조합니다.
  + SeatID (INT, Foreign Key): 티켓이 배정된 좌석의 ID입니다. Seats(SeatID)를 참조합니다.
  + BookingID (INT, Foreign Key): 티켓과 관련된 예약의 ID입니다. Bookings(BookingID)를 참조합니다.
  + IsTicketing (BOOLEAN): 티켓 발권 여부를 나타냅니다.
  + StandardPrice (INT): 티켓의 표준 가격입니다.
  + SalePrice (INT): 티켓의 할인 가격입니다.

좌석 요구사항에 따른 데이터베이스 테이블

**좌석 테이블 (Seats)**

* **설명**: 특정 상영을 위한 극장 좌석 정보를 저장하는 테이블입니다.
* **컬럼**:
  + SeatID (INT, Primary Key, Auto Increment): 각 좌석의 고유 식별자입니다.
  + TheaterID (INT, Foreign Key): 좌석이 위치한 극장의 ID입니다. Theaters(TheaterID)를 참조합니다.
  + ScreeningID (INT, Foreign Key): 상영의 ID입니다. Screenings(ScreeningID)를 참조합니다.
  + IsActive (BOOLEAN): 좌석의 활성화 여부를 나타냅니다.

*회원정보 요구사항에 따른 데이터베이스 테이블*

제약조건 : 밑의 기능에서 사용자로 로그인 하는경우, 사용자의 CustomerID가 user1인 계정으로 로그인 했다고 가정.

**회원정보 테이블 (Customers)**

* **설명**: 회원 정보를 저장하는 테이블입니다.
* **컬럼**:
  + CustomerID (VARCHAR(255), Primary Key): 각 고객의 고유 식별자입니다.
  + CustomerName (VARCHAR(255)): 고객의 이름입니다.
  + PhoneNumber (VARCHAR(15)): 고객의 전화번호입니다.
  + Email (VARCHAR(255)): 고객의 이메일 주소입니다.

*예매정보 요구사항에 따른 데이터베이스 테이블*

제약조건 : Payment 결제방법의 예시로는 신한카드, 국민카드, 무통장입금 등…  
PaymentStatus의 예시로는 일시불, 3개월 할부 등… 으로 정함

**예약 테이블 (Bookings)**

* **설명**: 고객이 한 예약 정보를 저장하는 테이블입니다.
* **컬럼**:
  + BookingID (INT, Primary Key, Auto Increment): 각 예약의 고유 식별자입니다.
  + Payment (VARCHAR(50)): 예약에 사용된 결제 방법입니다.
  + PaymentStatus (VARCHAR(50)): 결제 상태입니다.
  + Amount (INT): 예약 금액입니다.
  + CustomerID (VARCHAR(255), Foreign Key): 예약한 고객의 ID입니다. Customers(CustomerID)를 참조합니다.
  + PaymentDate (DATE): 결제 날짜입니다
    1. EER Diagram

텍스트, 폰트, 평행, 도표이(가) 표시된 사진

자동 생성된 설명

1. 프로그램 구조

1. 메인 모듈

* Main.java
  + 프로그램의 진입점으로, 초기 로그인 화면을 표시합니다.
  + 관리자와 회원 로그인 버튼을 제공하며, 각 버튼 클릭 시 해당 역할에 맞는 기능으로 연결합니다.

2. 로그인 모듈

* MainPanel.java
  + 로그인 화면을 표시하며, 사용자가 관리자 또는 회원으로 로그인할 수 있게 합니다.
  + 로그인 성공 시 각 역할에 맞는 메인 화면으로 이동합니다.
  + 로그인 실패 시 오류 메시지를 출력합니다.

3. 관리자 기능 모듈

* AdminPanel.java
  + 관리자 메인 화면을 표시하며, 데이터베이스 초기화, 테이블 조회, 데이터 입력/삭제/변경 기능을 제공합니다.
  + initializeDatabase();
    - 데이터베이스 초기화 기능을 구현하며, 테이블 생성 및 샘플 데이터를 삽입합니다.
  + viewAllTables();
    - 데이터베이스 테이블 조회 기능을 제공하며, 모든 테이블의 데이터를 확인할 수 있습니다.
  + MovieInputComponent.java / MovieDeleteComponent.java / MovieUpdateComponent.java
    - 데이터베이스 테이블에 데이터 입력, 삭제, 변경 기능을 구현합니다.

4. 회원 기능 모듈

* UserPanel.java
  + 회원 메인 화면을 표시하며, 영화 조회 및 예매 수정 기능을 제공합니다.
  + MovieSearchComponent.java
    - 영화 조회 기능을 구현하며, 영화명, 감독명, 배우명, 장르를 기준으로 검색합니다.
  + BookingManager.java
    - 예매 기능을 제공하며, 상영 일정 조회 및 좌석 선택, 결제 기능을 포함합니다.
  + ViewUserBookings.java
    - 예매 내역 조회 기능을 제공하며, 예매 내역 삭제, 영화 변경, 일정 변경, 추가 정보 조회 기능을 포함합니다.
  + TheaterReservationComponent.java
    - 사용자에게 좌석을 GUI로 보여주고 선택후 예매하는 기능을 제공합니다.

5. 유틸리티 모듈

* Utils.java
  + showMessage()
    - 메시지를 화면에 출력합니다.
  + connectToDatabase()
    - 데이터베이스와 연결합니다.
  + switchToPanel()
    - 패널을 전환합니다.

프로그램 흐름

1. 초기화 및 로그인
   * Main.java 실행 시 MainPanel.java를 호출하여 로그인 화면을 표시합니다.
   * 사용자가 관리자 또는 회원 계정을 입력한 후 로그인 버튼을 클릭합니다.
   * Login.java에서 입력된 계정 정보를 검증합니다.
     + 로그인 성공 시 switchToPanel()을 통해 AdminPanel.java 또는 UserPanel.java로 전환합니다.
     + 로그인 실패 시 showMessage()를 통해 오류 메시지를 출력합니다.
2. 관리자 기능
   * 관리자 로그인 성공 시 AdminPanel.java로 이동합니다.
   * 관리자 메인 화면에서 데이터베이스 초기화, 테이블 조회, 데이터 입력/삭제/변경 기능을 선택합니다.
   * 데이터베이스 초기화: initializeDatabase()를 호출하여 테이블 생성 및 샘플 데이터를 삽입합니다.
   * 테이블 조회: viewAllTables()를 호출하여 모든 테이블 데이터를 확인합니다.
   * 데이터 입력: MovieInputComponent.java를 사용합니다.
   * 데이터 삭제: MovieDeleteComponent.java를 사용합니다.
   * 데이터 변경: MovieUpdateComponent.java를 사용합니다.
3. 회원 기능
   * 회원 로그인 성공 시 UserPanel.java로 이동합니다.
   * 회원 메인 화면에서 영화 조회 및 예매 기능을 선택합니다.
   * 영화 조회: MovieSearcher.java의 MovieSearchComponent를 호출하여 영화명, 감독명, 배우명, 장르를 기준으로 검색합니다.
   * 예매 기능: BookingManager.java를 사용하여 상영 일정 조회, 좌석 선택, 결제 기능을 처리합니다.
     + TheaterReservationComponent.java를 통해 좌석을 선택하고 예매를 진행합니다.
   * 예매 내역 조회 및 관리: BookingViewer.java의 ViewUserBookings를 호출하여 예매 내역 삭제, 영화 변경, 일정 변경, 추가 정보 조회 기능을 제공합니다.
4. 데이터베이스 연결 및 유틸리티
   * DatabaseConnector.java의 connectToDatabase()를 통해 데이터베이스와 연결합니다.
   * 공통 유틸리티 기능은 Utils.java에서 제공합니다:
     + showMessage()를 통해 메시지를 출력합니다.
     + switchToPanel()을 통해 패널을 전환합니다**.**
5. 테스팅

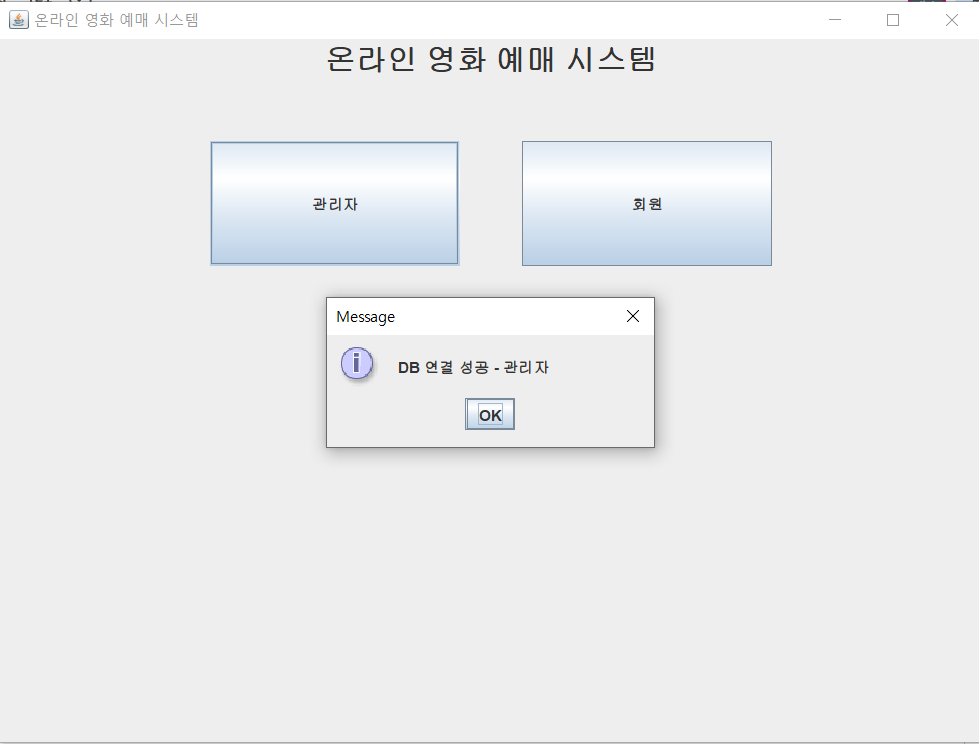
*기능 테스트 (테스트 화면은 윈도우 환경 / 맥 환경에서 진행 된 화면이라 UI는 다르게 보이지만 기능은 모두 같습니다.)*

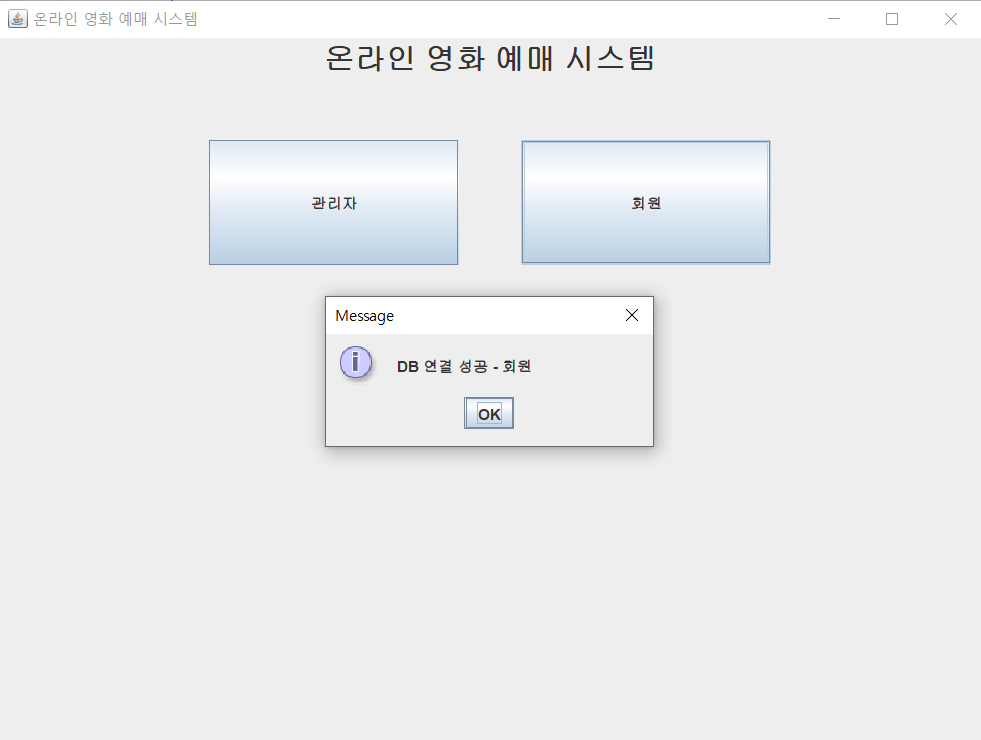
1. **메인페이지**



로그인 기능 : 관리자 / 회원 버튼 클릭

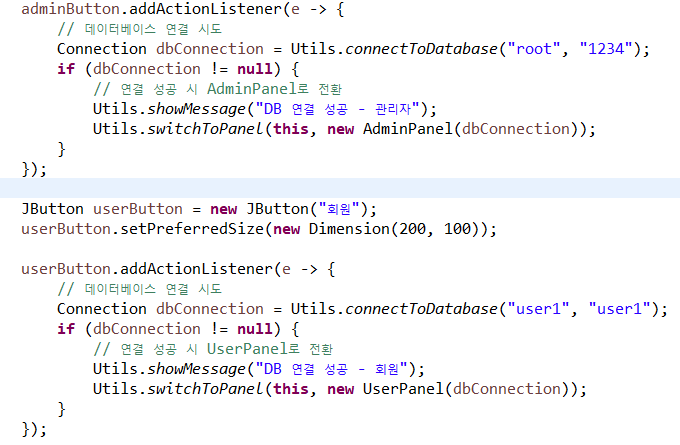
관리자 버튼 클릭 시 : ID = root / PW = root 계정으로 db1 데이터베이스에 접속 진행. 연결 성공 시 DB 연결 성공 - 관리자 메시지 출력

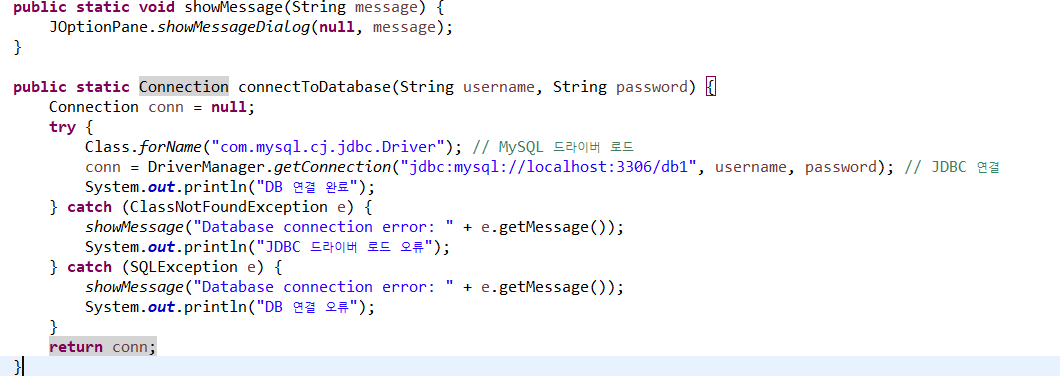


회원 버튼 클릭 시 : ID = user1 / pw = user1 계정으로 db1 데이터베이스에 접속 진행. 연결 성공 시 DB연결 성공 - 회원 메시지 출력

db연결 오류 발생 :

제약조건 : 로그인 버튼에서 계정 ID와 비밀번호를 코드상에 이미 적어둠.



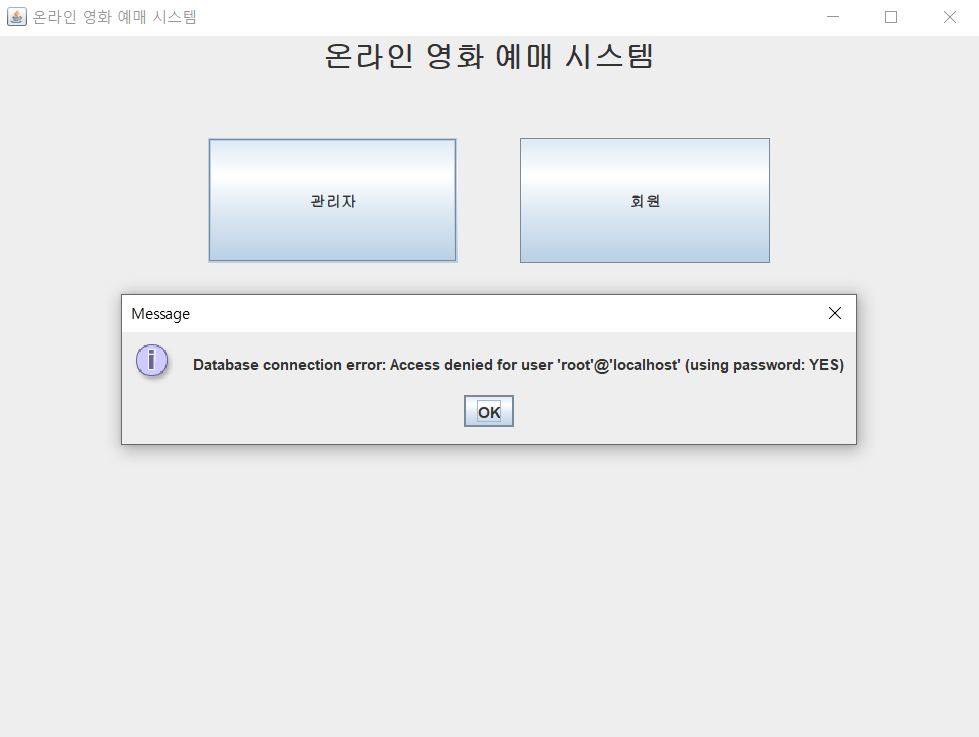


root, 1234 / user1, user을 이미 적어두어 그 값대로 로그인 진행.

로그인 오류 발생 : 로그인 ID / PW 을 코드상에서 임의로 수정한 경우 발생 이 경우엔 에러메시지 출력하게 설정

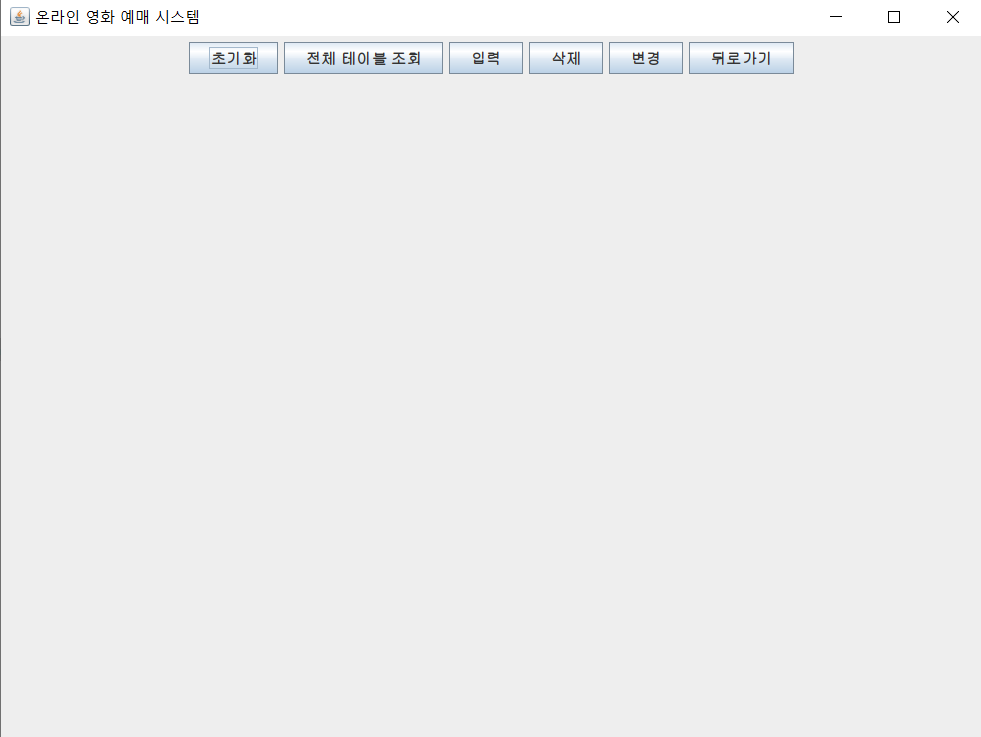


텍스트, 스크린샷, 소프트웨어, 컴퓨터 아이콘이(가) 표시된 사진

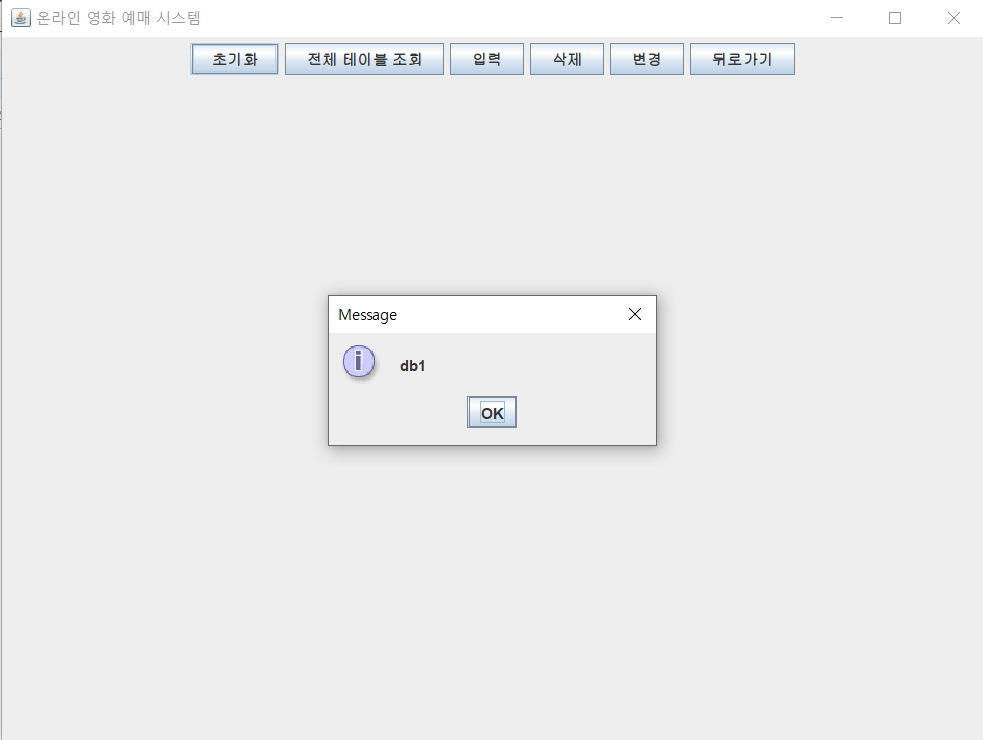
자동 생성된 설명

1. **관리자 (root / root)**

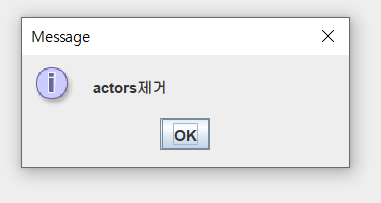
관리자 페이지



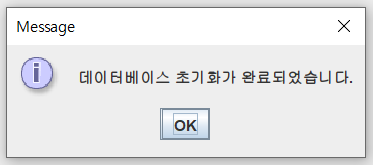
초기화 : 초기화 버튼 클릭 시 메소드 작동 (선택된 DB 이름 알려줌)



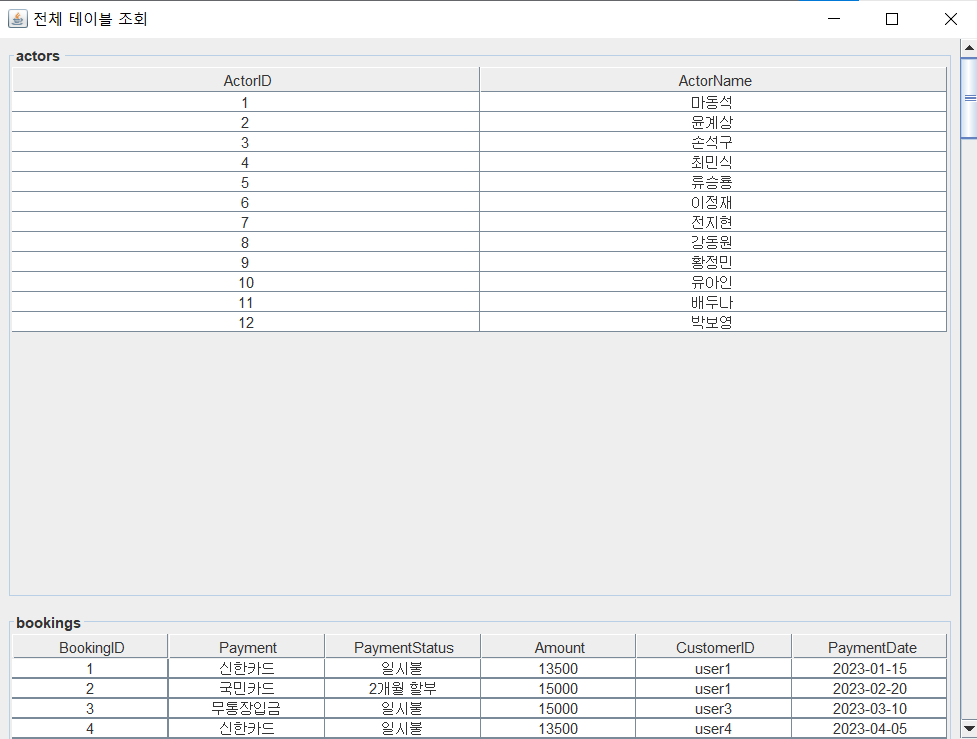
OK 누를 시 선택된 DB에 존재하는 테이블마다 “테이블이름”제거 란 메시지 팝업 생성 및 테이블 제거 찾아낸 모든 테이블에 대해 진행. (예시: actor란 테이블이 있었다면 actors제거란 팝업 생성하면서 actors 테이블 제거)



모든 테이블을 다 제거하면 테이블 생성 및 샘플 데이터 삽입 후 초기화 완료 팝업



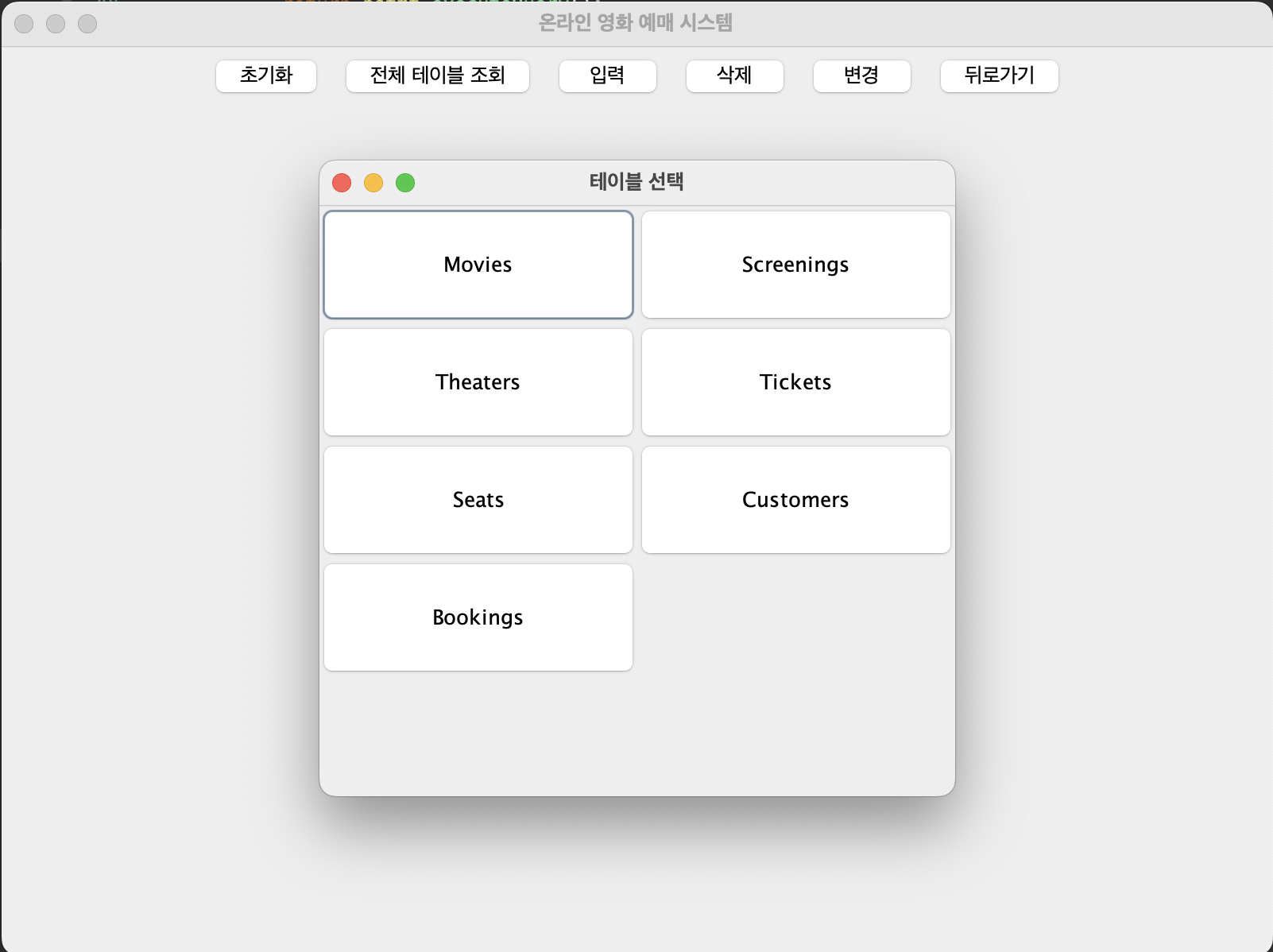
전체 테이블 조회 : 전체 테이블 조회 시 새로운 창에 현재 db1에 존재하는 모든 테이블 제공



데이터들은 스크롤바를 통해 모두 조회 가능

1. **입력**

입력 버튼 클릭 시 입력할 테이블 출력

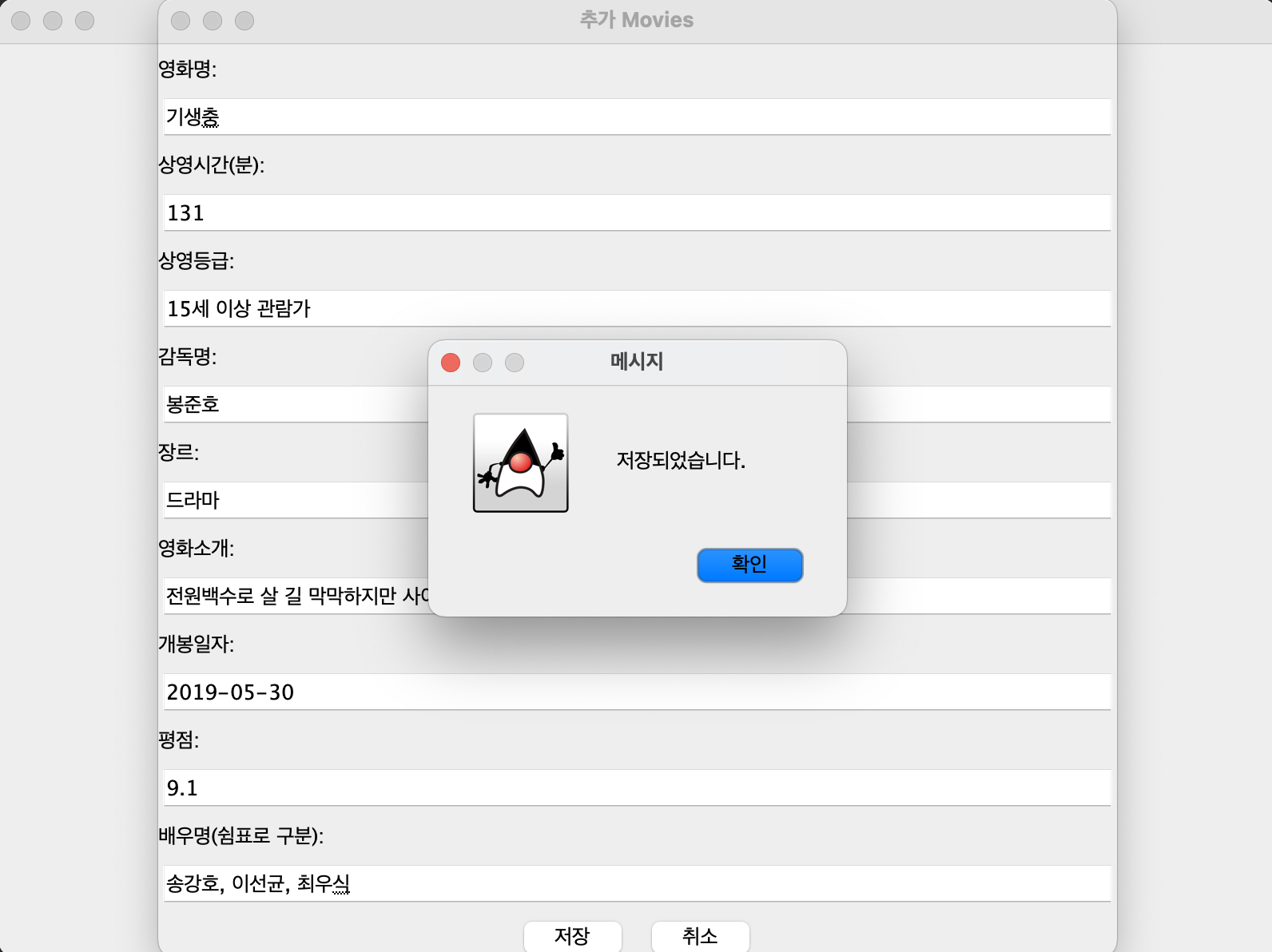
텍스트, 영수증, 스크린샷이(가) 표시된 사진

자동 생성된 설명

테이블 선택 시 해당 테이블에 대해 입력할 속성의 값을 입력할 창 생성

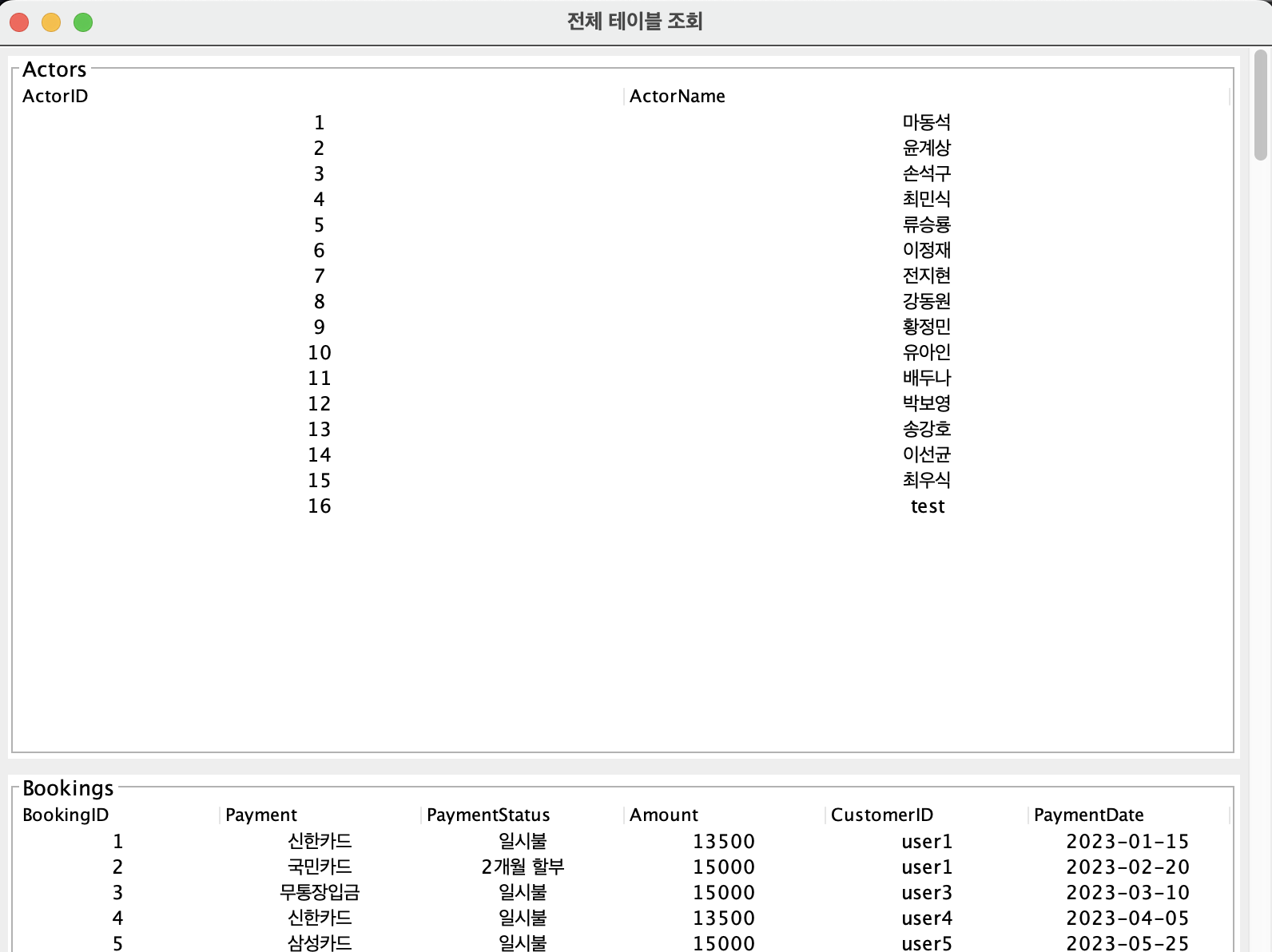
값을 입력 후 저장을 누르면 저장됨

해당 속성의 자료형에 맞는 값을 입력해야함

텍스트, 스크린샷, 소프트웨어, 번호이(가) 표시된 사진

자동 생성된 설명

영화명은 중복 가능하다고 가정



1. **삭제**

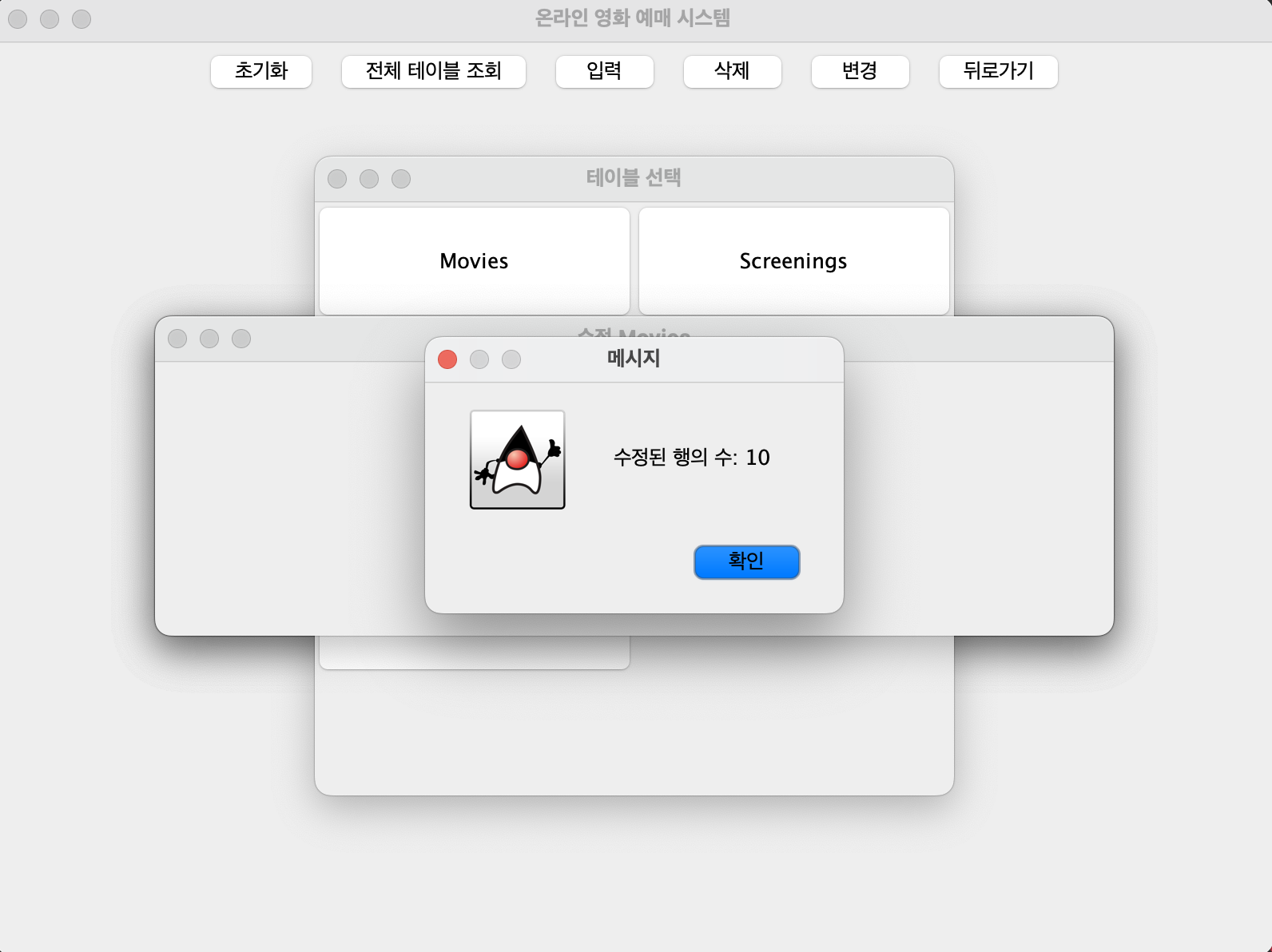
삭제도 마찬가지로 테이블 선택 후 삭제 조건 입력할 창 생성

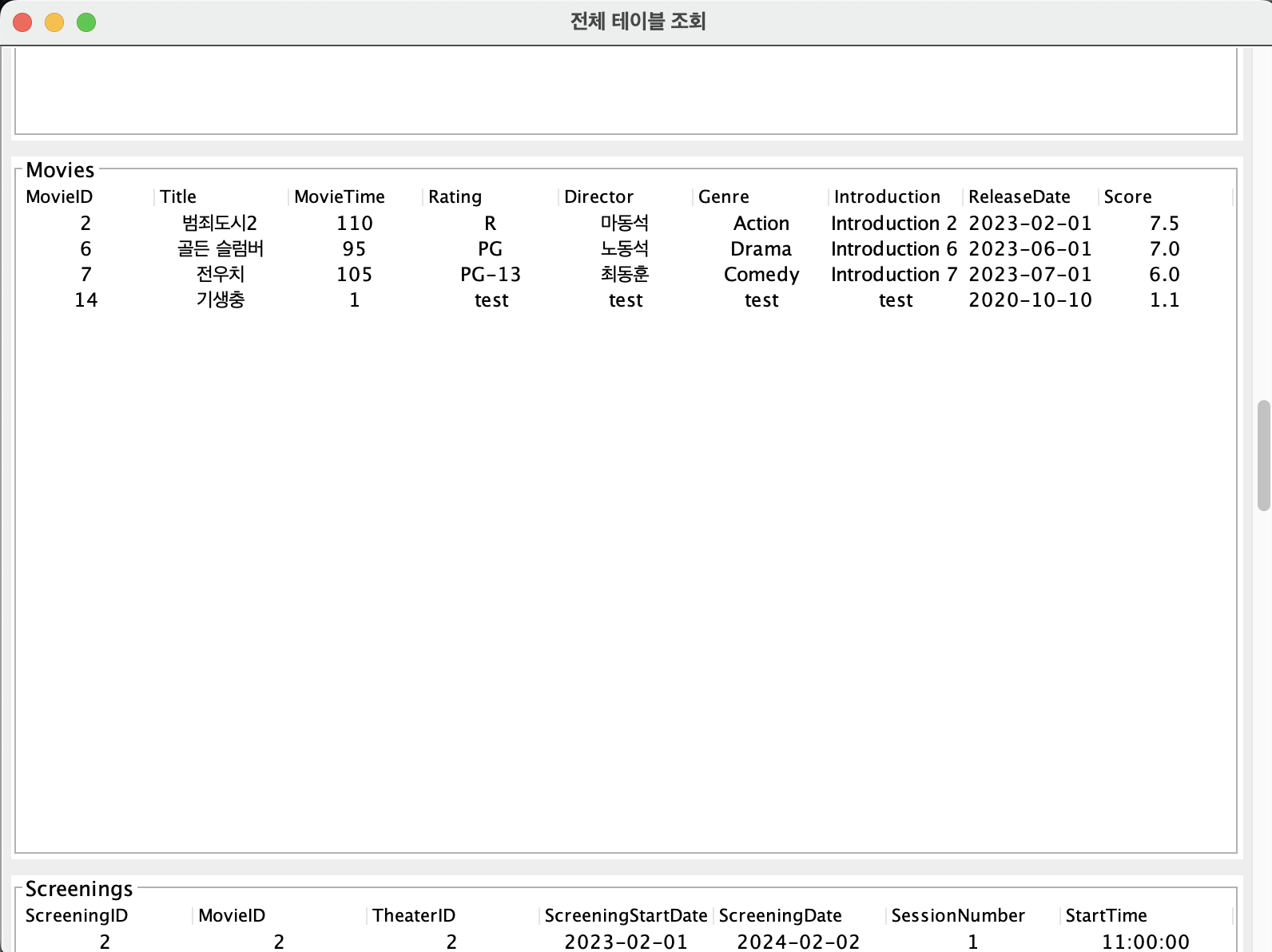
삭제할 조건 입력 후 저장을 누르면 삭제 진행

조건식은 where절에 들어갈 조건만 입력



삭제된 행의 개수 출력





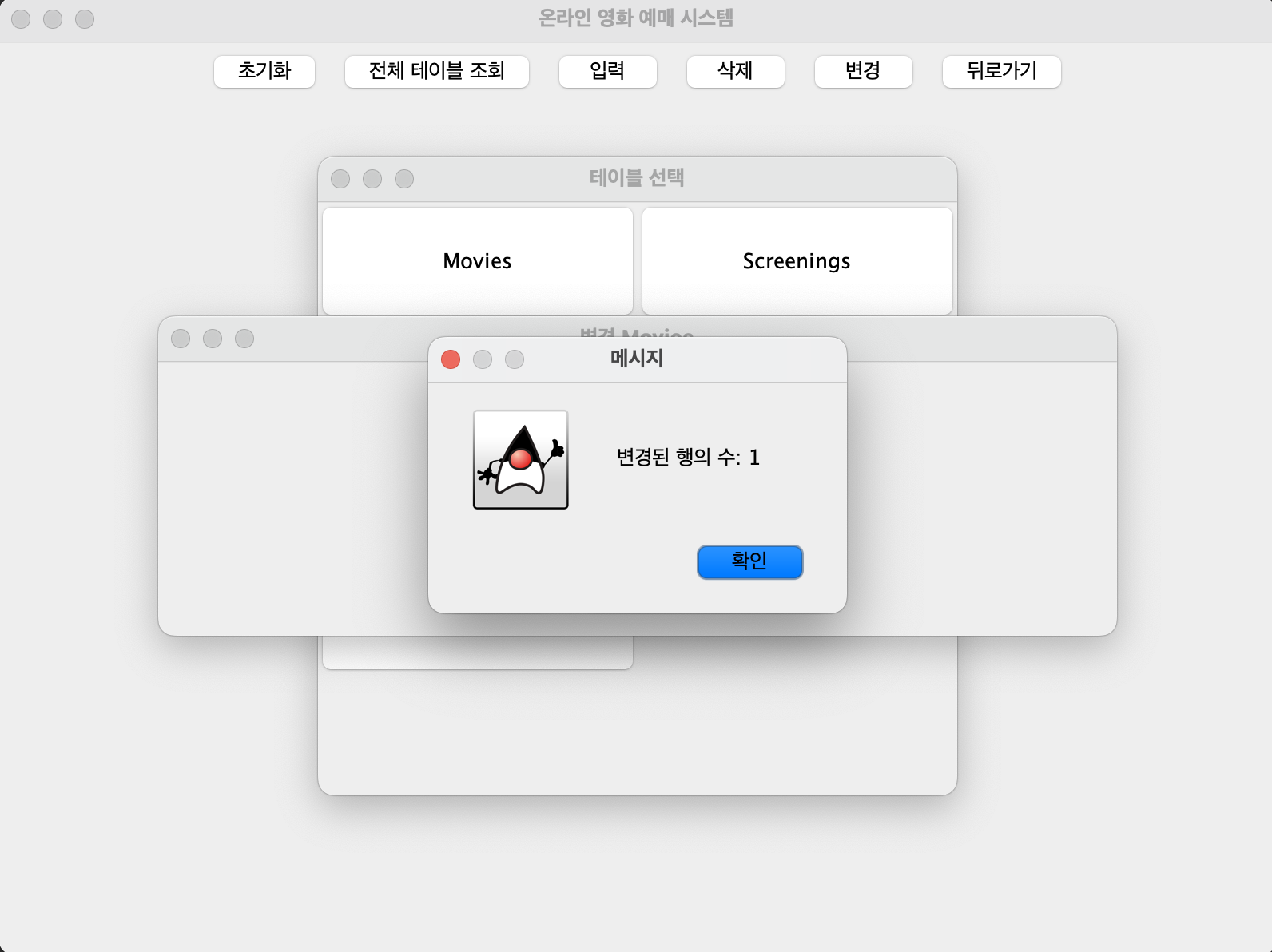
1. **변경**

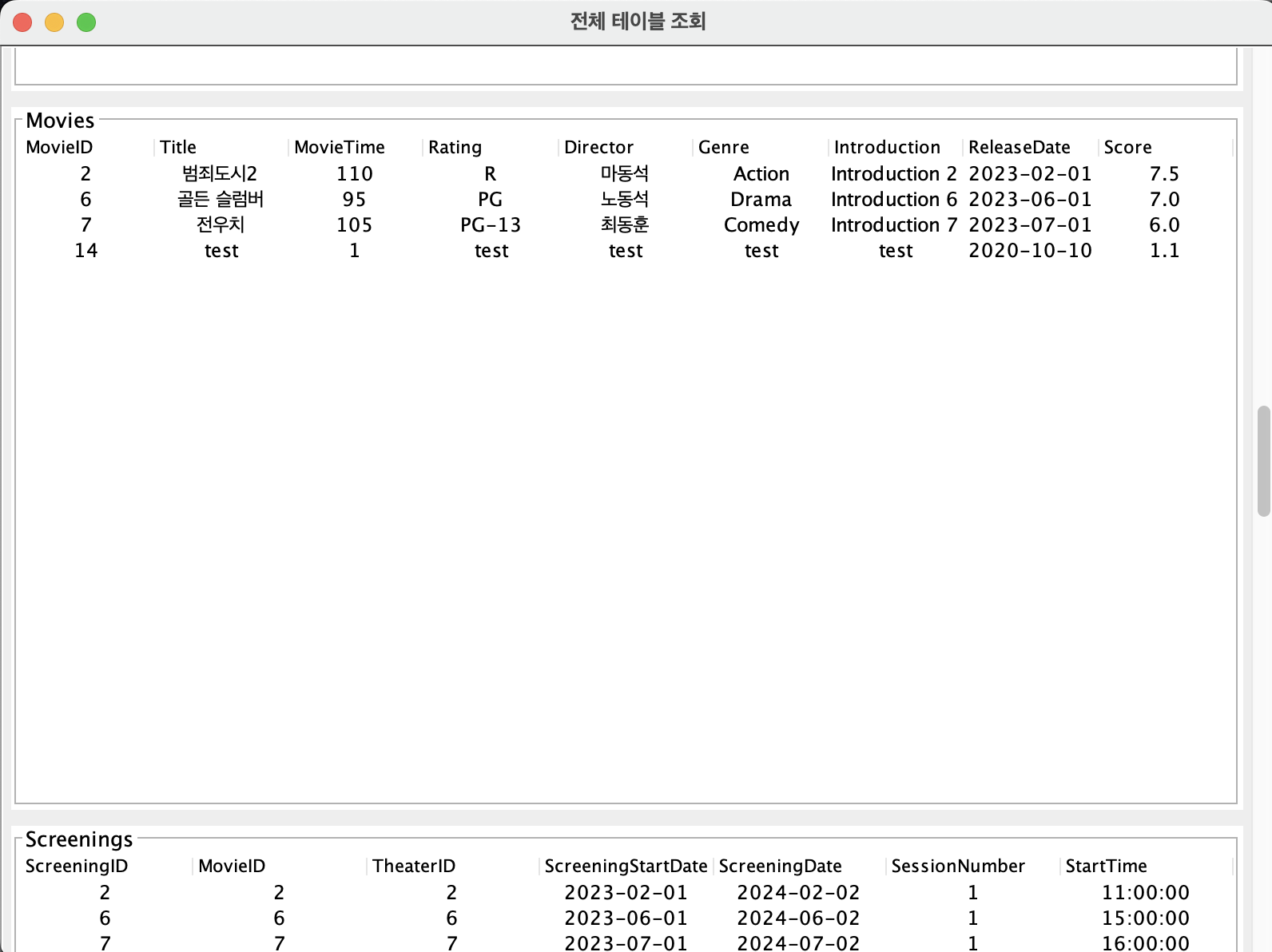
변경할 테이블을 선택 후 변경할 값과 조건을 입력할 창 생성

변경할 값과 변경할 조건을 차례로 입력해서 변경 진행



변경된 행의 개수 출력





1. **뒤로가기**

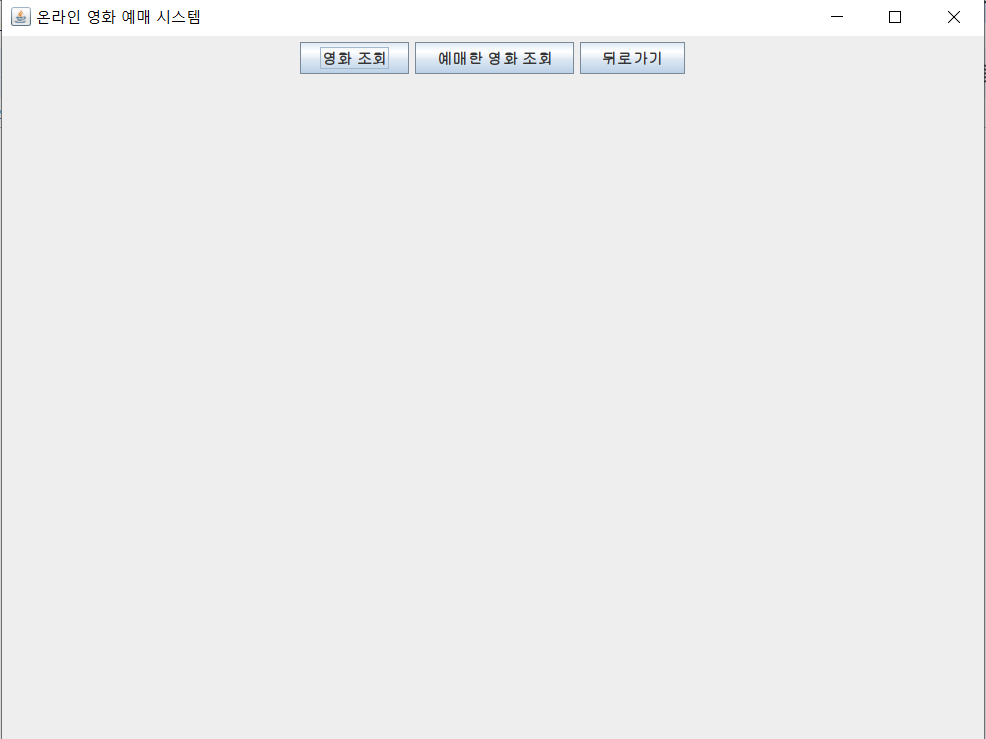
뒤로가기 클릭 시 이전 메인화면인 로그인창으로 다시 전환



이 화면으로 다시 이동

1. **회원 (user1 / user1, CustomerId = “user1”)**

회원 페이지

텍스트, 스크린샷, 번호, 평행이(가) 표시된 사진

자동 생성된 설명

영화 조회 : 영화 조회 클릭시 영화 검색 창을 띄움

검색 창 : 영화명, 감독명, 배우명, 장르를 입력 받을 수 있는 창

검색 제약조건

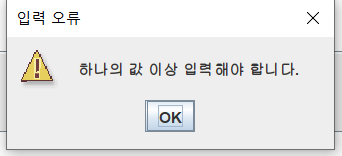
1. 각각 검색 창은 정확한 검색만 가능 (ex. 범죄도시1 을 검색하고 싶으면 영화명에 범죄도시1을 써야 나오지, 범죄도시만 쓰고 검색하면 범죄도시1 영화가 나오지 않는다. )
2. 공백도 검색어에 포함되므로 필요없는 공백을 입력해선 안된다.

(ex. “골든 슬럼버” 영화가 있다면, “ 골든 슬럼버” / “골든슬럼버”,

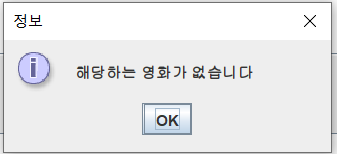
“골든 슬럼버 “ 등 쓸모없는 공백이 존재 시 영화검색이 안된다.)

1. 적어도 하나의 창에는 검색어를 작성해야 한다.
2. 여러 조건을 검색하면 AND 조건으로 검색 된다. (ex. 장르에 Action, 배우명에 마동석을 입력하면 장르가 Action이고 배우가 마동석인 영화들만 검색된다.)
3. 검색 결과는 MovieID와 Title만 출력한다.

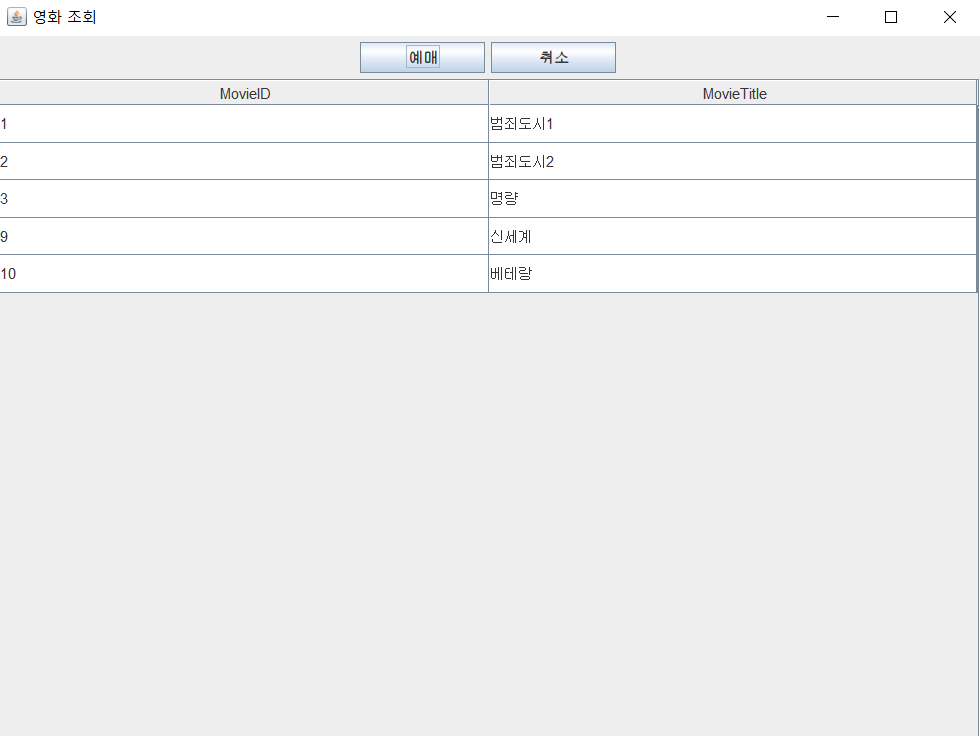
검색창에 아무것도 입력안하고 검색 시



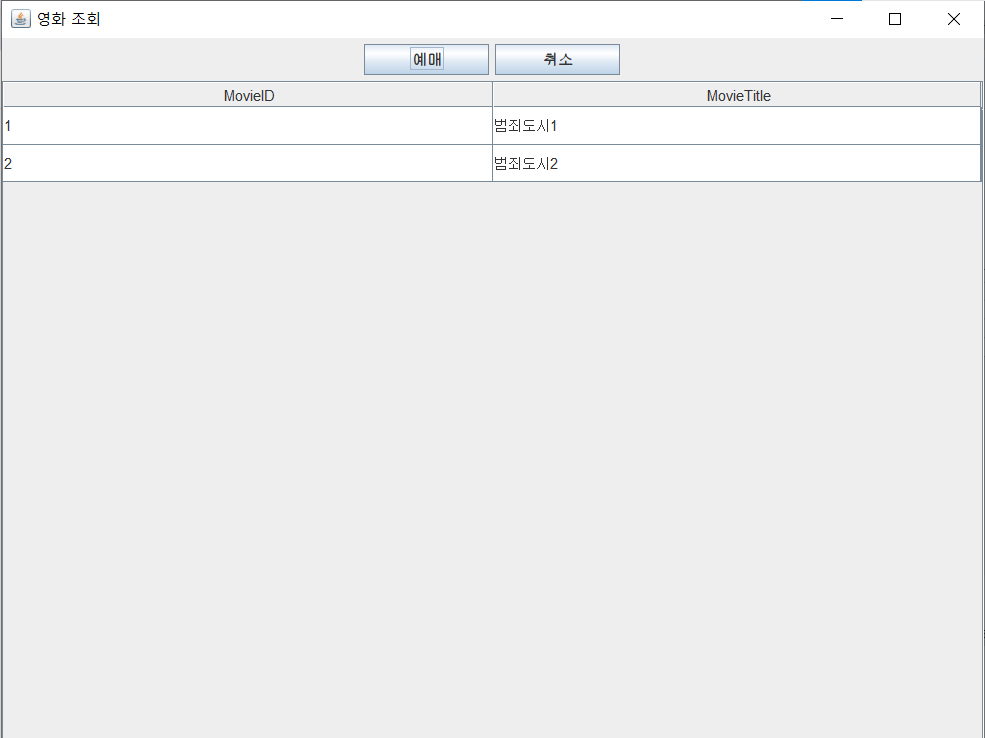
검색창에 정확한 값을 입력하지 않은 경우



검색창에 장르 : Action을 입력한 결과



검색창에 장르 : Action, 배우명 : 마동석 을 입력한 결과



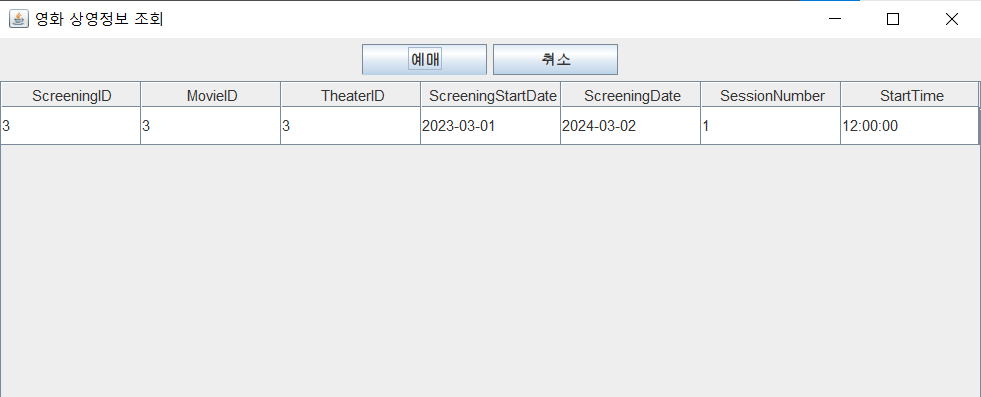
검색 후 영화목록 페이지에서

* 영화를 클릭 후 검색 버튼을 누를 시 해당영화의 모든 상영일정을 조회한다.

(장르 : Action 검색 예시)



영화를 클릭 시 파란색 네모로 표시, 이 후 예매 버튼을 클릭 시



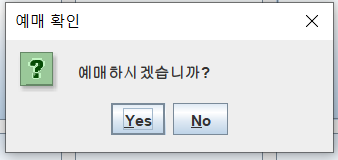
해당 영화의 상영일정들이 등장한다.

상영일정도 똑같이 마우스로 클릭 후 예매 버튼 클릭 시 영화관 좌석 gui가 나와 좌석 선택 후 예매하게 한다.

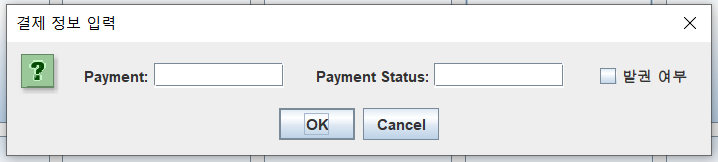


빨간 색 버튼, 예약됨 : 이미 예약 된 좌석으로 버튼을 클릭할 수 없다.

다른 버튼 클릭 시 예매하시겠습니까? 팝업



Yes 누를 시

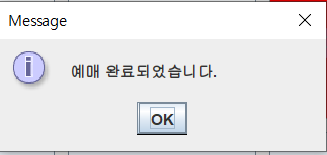


Payment : 신한카드, 국민카드 등… 결제방식 입력

Payment Status : 일시불, 3개월 할부 등… 입력

발권여부 : 체크시 발권, 체크 안할 시 미발권

결제 정보도 입력 완료 후 ok 누를 시 예매 완료 후 해당좌석 예매됨 표시, 버튼 비활성화





제약사항 : Payment에 “신한카드” 입력 시 금액 10퍼센트 할인된 금액으로 결제가 된다.

티켓 StandardPrice는 15000 으로 고정.

결제 정보 입력시 Payment, Payment Status 둘 중 하나라도 입력안하면 오류 메시지 발송

