

|  |
| --- |
| 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. 입력 기능
     5. 오류 처리 및 초기화
3. 스키마 설계
   * 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 화면

* 사용자가 시스템에 접근할 때 필요한 로그인 화면을 설계합니다.
* 화면은 관리자와 회원으로 구분되며, 사용자는 해당 권한에 따라 다른 접근 권한을 가집니다.
* 로그인 시 관리자는 root/1234 계정으로, 회원은 user1/user1 계정으로 접속합니다.
* 현재 날짜를 입력 받거나 소스코드에 미리 설정하여 사용합니다.
  + 1. 관리자 기능
* 데이터베이스 초기화 기능을 구현하여 시스템을 초기 상태로 되돌립니다.
* 모든 테이블에 대한 입력/삭제/변경 기능을 구현합니다. 이때, 조건식을 입력 받아 삭제/변경을 진행합니다.
* 전체 테이블을 조회하는 기능을 구현하여 관리자가 데이터베이스의 상태를 확인할 수 있도록 합니다.
  + 1. 회원 기능
* 모든 영화를 조회하고 예약하는 기능을 구현합니다. 영화명, 감독명, 배우명, 장르를 기준으로 검색할 수 있어야 합니다.
* 선택한 영화에 대한 예매 기능을 구현하여 상영관의 좌석을 GUI로 표시하고 선택할 수 있게 합니다.
* 예약한 영화에 대한 정보를 조회하는 기능을 제공합니다. 영화명, 상영일, 상영관번호, 좌석번호, 판매가격을 표시합니다.
* 예약 정보 중 하나를 선택하면 해당 예약에 대한 모든 상영일정, 상영관, 티켓 정보를 표시하는 기능을 구현합니다.
* 예약한 영화를 조회하여 취소하거나 다른 영화나 상영 일정으로 변경할 수 있는 기능을 제공합니다.
  + 1. 입력 기능
* 입력 기능은 모든 데이터를 한 번에 입력할 수 있는 GUI를 구현합니다.
* 예를 들어, 여러 속성을 입력해야 할 때 이를 한 번에 입력할 수 있는 UI를 설계합니다.
* + 1. 오류 처리 및 초기화
* 사용자가 시스템에 접근할 때 필요한 로그인 화면을 설계합니다.
* 화면은 관리자와 회원으로 구분되며, 사용자는 해당 권한에 따라 다른 접근 권한을 가집니다.
* 로그인 시 관리자는 root/1234 계정으로, 회원은 user1/user1 계정으로 접속합니다.
* 현재 날짜를 입력 받거나 소스코드에 미리 설정하여 사용합니다.

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

EER Diagram

텍스트, 폰트, 평행, 도표이(가) 표시된 사진

자동 생성된 설명