

# 데이터베이스 프로젝트 최종보고서

21010914 김서현

22013463 부창오

데이터베이스001

## 1 요구사항분석내용

관리자 기능

해당 기능들은 모두 관리자 페이지에서 구현가능하다.

|                          |                     |
|--------------------------|---------------------|
| 데이터베이스 초기화 기능            | [테이블 초기화]버튼으로 기능    |
| 데이터베이스의 모든 테이블에 대한 입력 기능 | [테이블 입력] 버튼으로 기능    |
| 데이터베이스의 모든 테이블에 대한 삭제 기능 | [테이블 삭제] 버튼으로 기능    |
| 데이터베이스의 모든 테이블에 대한 변경 기능 | [테이블 변경] 버튼으로 기능    |
| 전체 테이블 조회                | [전체 테이블 보기] 버튼으로 기능 |

회원기능

초기화면에서 바로 회원로그인이 가능하며, 로그인후 회원 계정으로

윈도우에서 해당기능들이 구현된다.

|  |                             |
|--|-----------------------------|
| 회원 로그인 기능  | 최초윈도우에서 바로 가능               |
| 조건에 맞는 모든 영화에 대한 조회 기능   | (제목, 감독, 배우, 장르) 형식으로 검색 가능 |
| 위에서 조회한 영화에 대한 예매 기능   | 영화이름 선택후 예매가능               |
| 본인이 예매한 영화에 대해서 영화명, 상영일, 상영관번호, 좌석번호 및 판매가격 정보를 보여주는 기능                 | 조회 가능                       |
| 위에서 표시된 예매 정보 중에서 하나를 "클릭"하면 해당 예매에 대해서 모든 상영일정, 상영관, 티켓에 대한 정보를 보여주는 기능 | 조회 가능                       |
| 본인이 예매한 영화에 대하여 조회하고 한 개 이상의 예매 정보를 "삭제"하는 기능                            | 삭제 가능                       |

|   |          |
|---|----------|
| 본인이 예매한 영화에 대하여 조회하여 다른 영화로 예매를 변경하는 기능 | 예매 변경 가능 |
| 본인이 예매한 영화에 대하여 조회하여 다른 상영 일정으로 변경하는 기능 | 일정 변경 가능 |

## 기타구현기능

|                        |        |
|------------------------|--------|
| 프로그램 시작 시 구동당시의 날짜 초기화 | 초기화 가능 |
|------------------------|--------|

## 예외기능

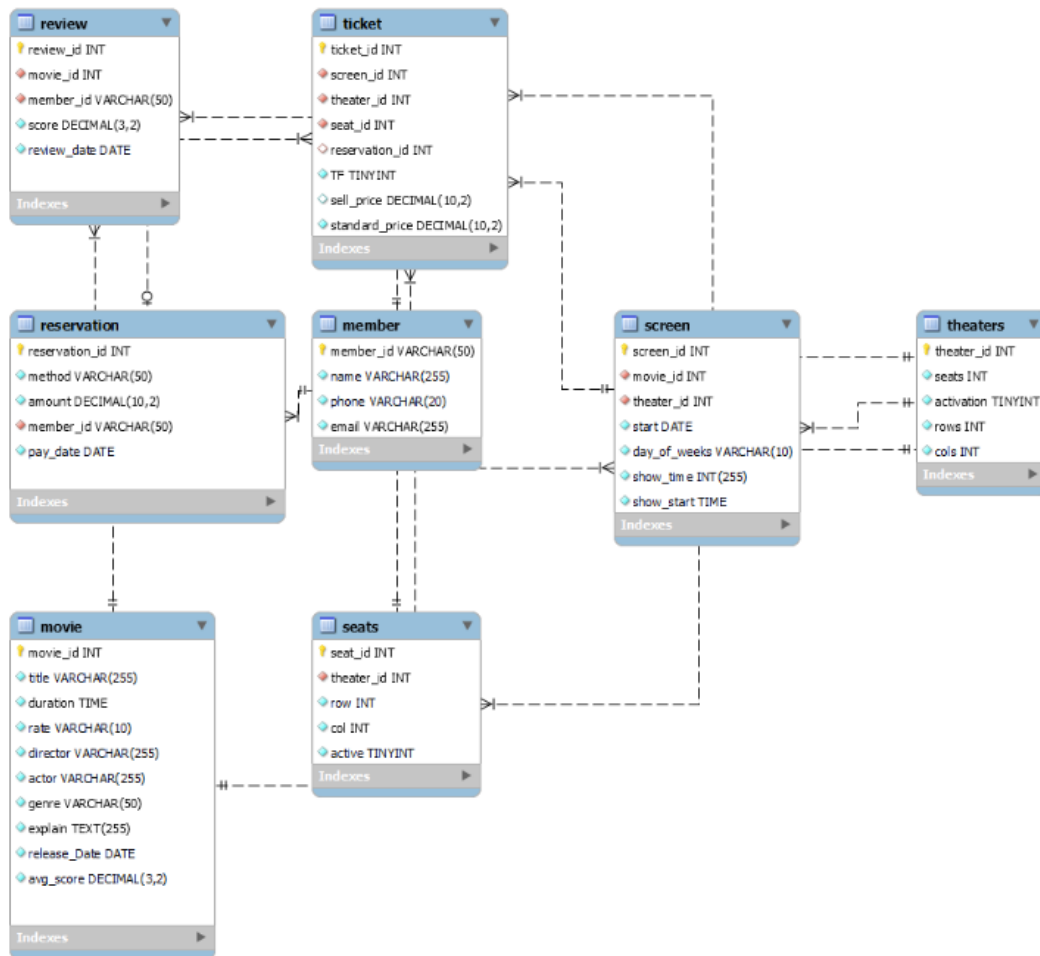
### 관리자

|               |                    |                |
|---------------|--------------------|----------------|
| 테이블 검색 시 예외처리 | 존재하지 않는 테이블 명 입력   | 해당 데이터가 없음을 알림 |
| 테이블 입력 시 예외처리 | 테이블 기본 키 정책 위배     | 오류 알림창 팝업      |
| 테이블 입력 시 예외처리 | 테이블 외래 키 정책 위배     | 오류 알림창 팝업      |
| 테이블 입력 시 예외처리 | 빈 입력칸 생길 시         | 오류 알림창 팝업      |
| 테이블 입력 시 예외처리 | 예매일자가 개봉일자보다 빠른 경우 | 유효날짜 요청 알림창 팝업 |

### 회원

|                               |                   |        |
|-------------------------------|-------------------|--------|
| 선택 기능 시 예외처리 (예매 삭제/ 예매 변경 시) | 선택을 하지 않고 버튼 클릭 시 | 주의창 팝업 |
|-------------------------------|-------------------|--------|

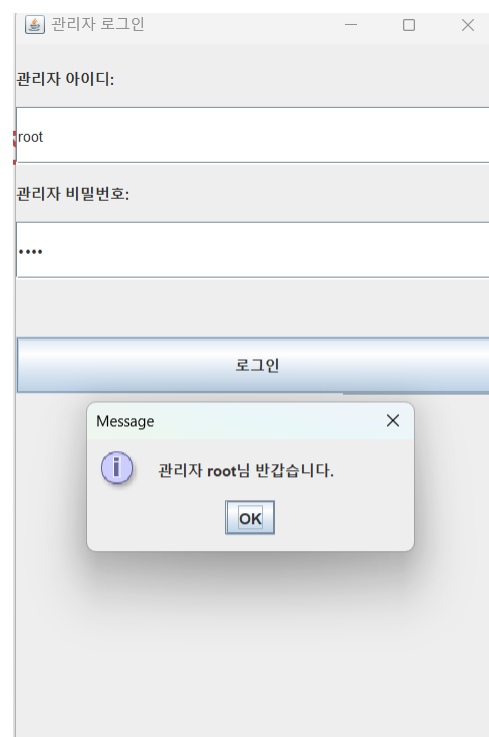
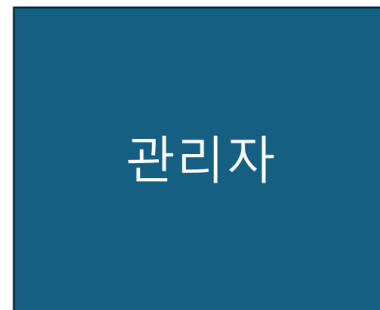
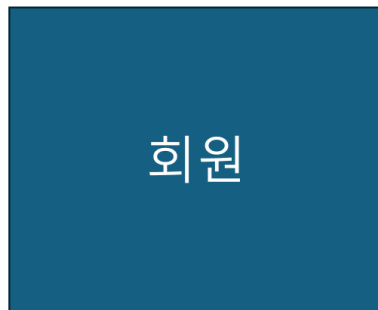
## 2 스키마 설계



### 3 프로그램 구조

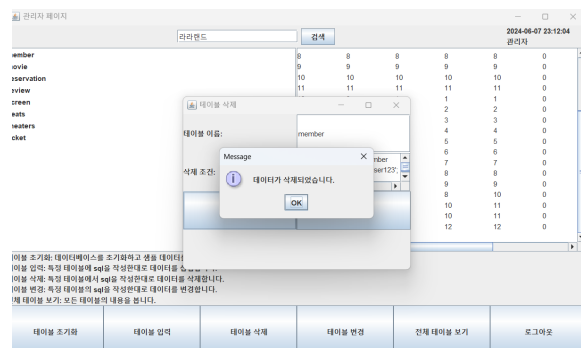
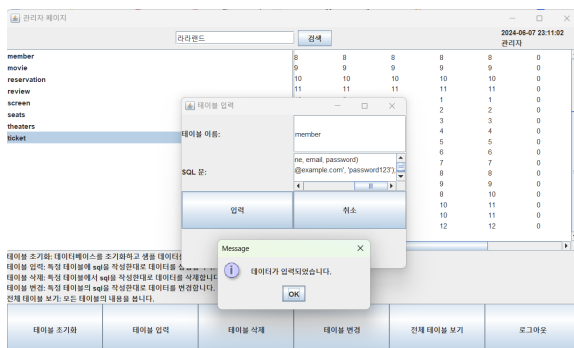
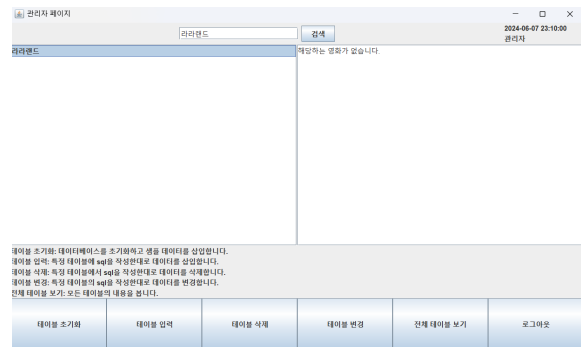
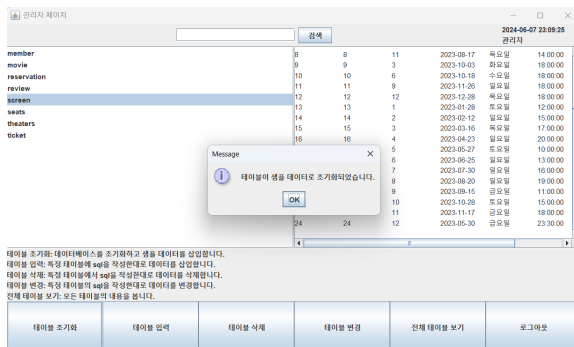
#### 초기화면 (로그인 페이지)

[별도의 관리자 로그인 윈도우]



## 관리자 페이지

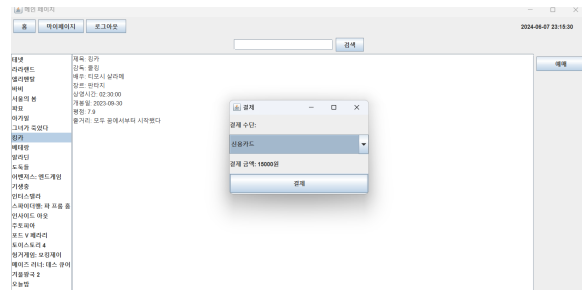
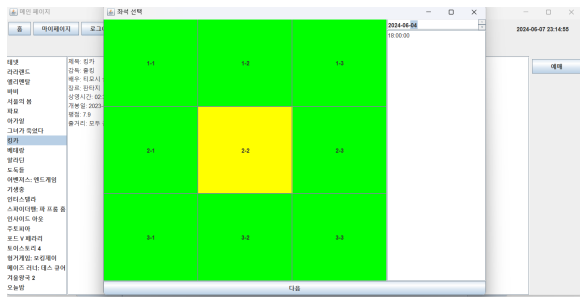
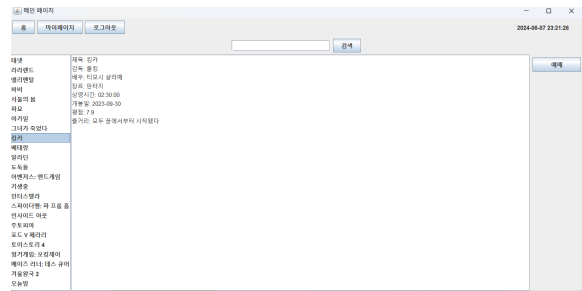
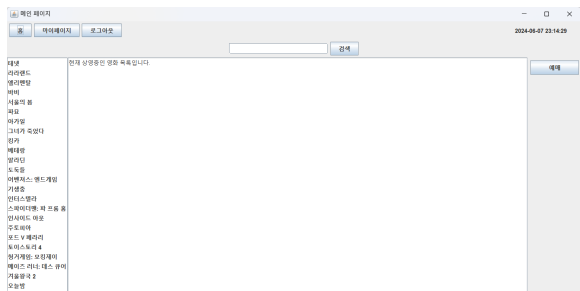
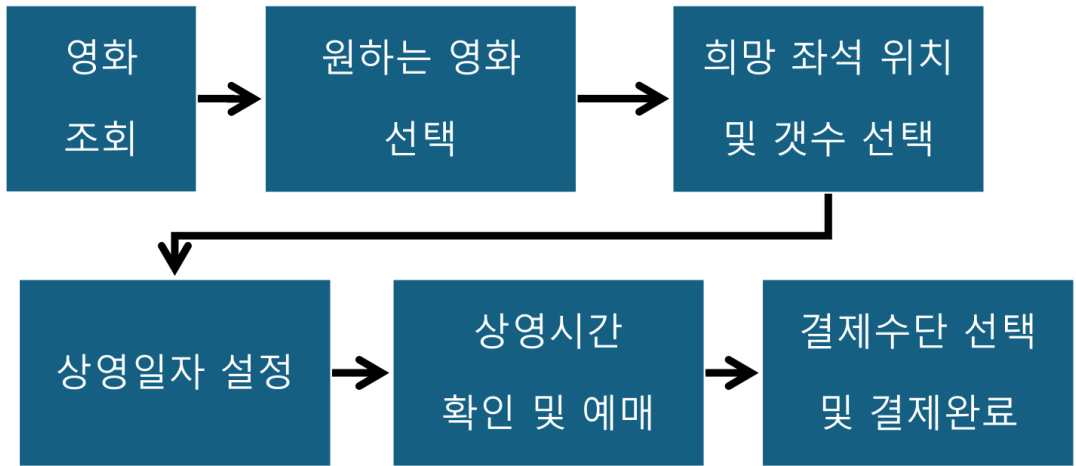
|           |  |
|-----------|--|
| 검색기능      | 영화 제목 및 형식으로 검색하면 조회 가능                  |
| 테이블 초기화   | 초기화 알림창 뜬 후 초기화 데이터셋트 확인가능               |
| 테이블 입력    | MySQL INSERT 쿼리문 입력시 데이터 추가가능            |
| 테이블 삭제    | MySQL DELETE 쿼리문 입력시 데이터 삭제가능            |
| 테이블 변경    | MySQL ALTER 쿼리문 입력시 데이터 수정가능             |
| 전체 테이블 조회 | 테이블 이름 목록 조회 가능 및 이름 클릭 후 해당 테이블 상세조회 가능 |



회원 페이지는 영화 예매 및 예매내역 조회로 나뉜다.

1 영화 예매

# 회원 페이지 – 영화 예매



## 2 예매 내역 조회

## 회원 -마이페이지 (예매 내역 조회)

### 티켓정보조회

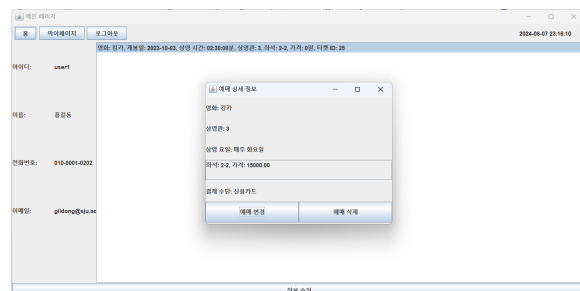
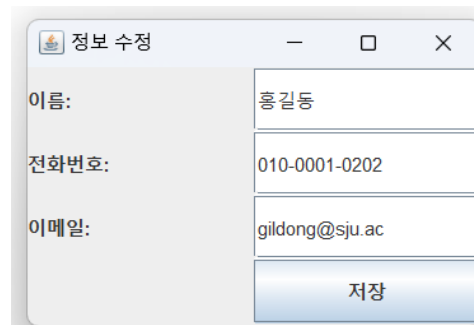
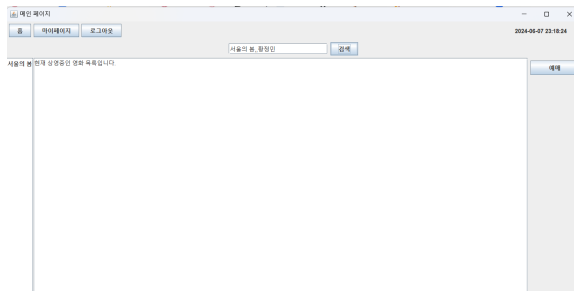
- GUI에서 예매된 내역의 영화명, 상영일, 상영관, 좌석 및 가격 조회 가능

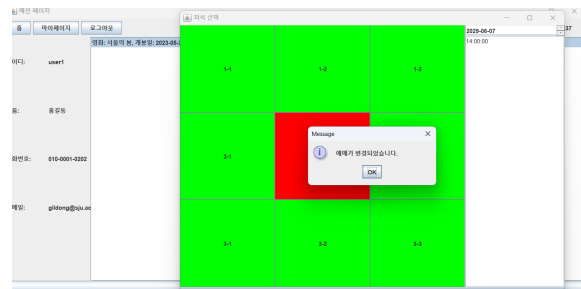
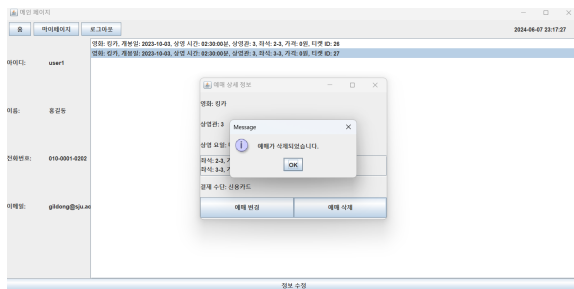
### 예매 정보 삭제

- [예매 상세정보]에서 [예매취소] 버튼 클릭으로 기능
- 예매내역삭제
- 마이페이지에서 예매내역 업데이트

### 예매 정보 변경

- [예매 상세정보]에서 [예매 변경]버튼 클릭으로 기능
- 예매내용 변경
- 마이페이지에서 예매내역 업데이트





## 4 테스트 방법

### 관리자 기능

#### 전체 테이블 조회 기능

- 전체 DB 조회 버튼을 클릭 → SELECT 쿼리문으로 실행되며, 테이블내용들이 표시됨

#### 데이터베이스의 모든 테이블에 대한 입력 기능

- 테이블명 입력 : { movie, theaters ,screen, ticket, seats , member, reservation, review }
- SQL 문 입력

# INSERT INTO movie\_database.[테이블이름] (속성1,속성2,속성3..... , )  
VALUES ('data1', 'data2', 'data3', .....);

- 입력 버튼 클릭
- 전체 DB 조회로 추가된 데이터 확인

#### 데이터베이스의 모든 테이블에 대한 삭제 기능

- 테이블명 입력 { movie, theaters ,screen, ticket, seats , member, reservation, review }
- 삭제 조건문 입력

예외 상황 { 외래키 참조시, 외래키로 참조되는 데이터가 있을시, 예약이나 티켓등 예매상황이라 지을 수 없을 시 }

- 삭제 버튼 클릭
- 전체 DB 조회로 삭제된 데이터 확인



## 데이터베이스의 모든 테이블에 대한 변경 기능

- SQL 문 입력
- 변경 버튼 클릭
- 전체 DB 조회로 변경된 데이터 확인

## 데이터베이스 초기화 기능 :

1. 전체 DB 조회로 기존 DB 확인
2. [테이블 초기화]버튼 클릭해 데이터 초기화
3. 전체 DB 조회로 변경된 DB 확인

## 회원기능

### 회원 로그인 기능

user1/user1 로그인시 회원페이지 접근 가능

로그인 버튼 클릭

부적절한 아이디 및 비밀번호 입력시 주의 메시지 출력

### 조건에 맞는 모든 영화에 대한 조회 기능

- 로그인 후 메인화면에 전체 영화 조회 가능
- (제목, 감독, 배우, 장르) 형식으로 검색 가능, 반점으로 구분하며 공백가능
- [검색] 버튼 클릭 → 원하는영화 조회 가능

### 위에서 조회한 영화에 대한 예매 기능

- 예매를 원하는 영화 선택 (단일 선택만 가능)
- 원하는 상영일자 선택(단일 선택만 가능), 상영하는 특정 요일만 예매 가능
- 원하는 좌석 및 갯수 선택
- 결제수단 선택 → 예매완료
- 마이페이지에서 회원의 영화 예매내역 조회 가능

### 회원의 예매한 영화에 대한 예매 정보 조회 기능

- [마이페이지] 버튼 클릭
- 회원이 예매한 영화의 영화명, 상영일, 상영관번호, 좌석번호 및 판매가격 확인

조회된 예매 정보 중에서 특정 예매내역 선택시 해당 내역에 대한 정보를 표시하는 기능

- 리스트에서 기본적으로 예매의 영화명, 상영일정, 상영관, 좌석 및 가격에 대한 정보 확인 가능
- 특정 내역 클릭시 [예매상세정보 창]으로 영화명, 상영관, 상영요일, 좌석, 가격 및 결제 수단 확인 가능

#### **회원의 영화예매내역중에서 특정 예매 정보를 삭제하는 기능**

- [마이 페이지] 버튼을 클릭하여 회원의 영화예매 내역 조회 가능
- 삭제를 원하는 예매내역 클릭하여 [예매상세정보] 창 접근 (한 좌석단위로만 변경가능)
- [예매 삭제] 버튼 클릭
- 새로고침된 마이페이지를 통해 예매가 삭제됐는지 확인

#### **본인이 예매한 영화에 대하여 조회하여 다른 영화/ 다른 상영일정 으로 예매를 변경하는 기능**

- [마이 페이지]버튼을 클릭하면 회원의 영화예매 내역 조회 가능
- 변경을 원하는 예매내역 클릭하여 [예매상세정보] 창 접근 (동시 예매가 된 영화예매의 경우 예매내역의 영화 및 좌석위치 및 갯수가 동시에 변경됨)
- [예매 변경] 버튼클릭
- 영화선택 후 선택 버튼 클릭
- 원하는 갯수의 좌석 선택후 해당 영화의 상영일정에 맞춰 변경
- [마이페이지]버튼으로 윈도우 초기화후 예매가 변경 됐는지 확인

#### **기타 구현 기능**

- 회원이 [홈] 버튼 클릭 시 영화 조회 및 예매 화면으로 전환
- [마이페이지]에서 [정보수정]버튼 클릭으로 회원의 개인정보수정 가능
- 회원 및 관리자 [로그아웃]버튼 클릭시 초기 로그인화면으로 전환됨
- 프로그램 시작 시 날짜 및 시간은 구동당시의 표준시각을 사용함

#### **예외 처리 테스트 기능**

관리자 페이지에서 외래키 관련 데이터 임의로 삭제 불가

회원예매페이지에서 개봉일보다 이른 상영일자 예매 불가

## 5 초기 실행용 SQL 코드

```
CREATE DATABASE IF NOT EXISTS db1;
```

```
USE db1;
```

```
CREATE TABLE IF NOT EXISTS db1.theaters (  
    theater_id INT NOT NULL AUTO_INCREMENT COMMENT '상영관 아이디',  
    seats INT NOT NULL COMMENT '좌석 수',  
    activation TINYINT NOT NULL COMMENT '상영관 사용 여부',  
    `rows` INT NOT NULL COMMENT '가로',  
    `cols` INT NOT NULL COMMENT '세로',  
    PRIMARY KEY (theater_id)  
) ENGINE = InnoDB;
```

```
CREATE TABLE IF NOT EXISTS db1.movie (  
    movie_id INT NOT NULL AUTO_INCREMENT COMMENT '영화 ID',  
    title VARCHAR(255) NOT NULL COMMENT '영화제목',  
    duration TIME NOT NULL COMMENT '상영시간',  
    rate VARCHAR(10) NOT NULL COMMENT '상영 등급',  
    director VARCHAR(255) NOT NULL COMMENT '감독명',  
    actor VARCHAR(255) NOT NULL COMMENT '배우명',  
    genre VARCHAR(50) NOT NULL COMMENT '장르',  
    `explain` TEXT NOT NULL COMMENT '영화 설명',  
    release_date DATE NOT NULL COMMENT '개봉일',  
    avg_score DECIMAL(3,2) NOT NULL COMMENT '평균 평점',  
    PRIMARY KEY (movie_id)  
) ENGINE = InnoDB;
```

```
CREATE TABLE IF NOT EXISTS db1.member (  
    member_id VARCHAR(50) NOT NULL COMMENT '회원아이디',  
    name VARCHAR(255) NOT NULL COMMENT '이름',  
    phone VARCHAR(20) NOT NULL COMMENT '전화번호',  
    email VARCHAR(255) NOT NULL COMMENT '이메일',  
    PRIMARY KEY (member_id)  
) ENGINE = InnoDB;
```

```

CREATE TABLE IF NOT EXISTS db1.seats (
    seat_id INT NOT NULL AUTO_INCREMENT COMMENT '좌석 아이디',
    theater_id INT NOT NULL COMMENT '상영관 아이디',
    `row` INT NOT NULL COMMENT '가로',
    `col` INT NOT NULL COMMENT '세로',
    active TINYINT NOT NULL COMMENT '좌석 사용 여부',
    PRIMARY KEY (seat_id),
    INDEX theater_id_idx (theater_id ASC),
    CONSTRAINT fk_seat_theater FOREIGN KEY (theater_id) REFERENCES db1.theater (theater_id)
) ENGINE = InnoDB;

```

```

CREATE TABLE IF NOT EXISTS db1.reservation (
    reservation_id INT NOT NULL AUTO_INCREMENT COMMENT '예매 ID',
    method VARCHAR(50) NOT NULL COMMENT '결제 수단',
    status VARCHAR(50) NOT NULL COMMENT '결제 상태',
    amount DECIMAL(10,2) NOT NULL COMMENT '결제 금액',
    member_id VARCHAR(50) NOT NULL COMMENT '회원 아이디',
    pay_date DATE NOT NULL COMMENT '결제 날짜',
    PRIMARY KEY (reservation_id),
    INDEX member_id_idx (member_id ASC),
    CONSTRAINT fk_reservation_member FOREIGN KEY (member_id) REFERENCES db1.member (member_id)
) ENGINE = InnoDB;

```

```

CREATE TABLE IF NOT EXISTS db1.screen (
    screen_id INT NOT NULL AUTO_INCREMENT COMMENT '상영 일정 아이디',
    movie_id INT NOT NULL COMMENT '영화 아이디',
    theater_id INT NOT NULL COMMENT '상영관 아이디',
    `start` DATE NOT NULL COMMENT '상영 시작일',
    day_of_weeks VARCHAR(10) NOT NULL COMMENT '상영요일',
    show_start TIME NOT NULL COMMENT '상영 시작 시간',
    show_time INT NOT NULL COMMENT '상영 회차',
    PRIMARY KEY (screen_id),
    INDEX movie_id_idx (movie_id ASC),
    INDEX theater_id_idx (theater_id ASC),
    CONSTRAINT fk_screen_movie FOREIGN KEY (movie_id) REFERENCES db1.movie (movie_id),
    CONSTRAINT fk_screen_theater FOREIGN KEY (theater_id) REFERENCES db1.theater (theater_id)
) ENGINE = InnoDB;

```

```

CREATE TABLE IF NOT EXISTS db1.ticket (
    ticket_id INT NOT NULL AUTO_INCREMENT COMMENT '티켓 ID',
    screen_id INT NOT NULL COMMENT '상영 일정 ID',
    theater_id INT NOT NULL COMMENT '상영관 ID',
    seat_id INT NOT NULL COMMENT '좌석 ID',
    reservation_id INT NULL COMMENT '예매 ID',
    TF TINYINT NOT NULL COMMENT '티켓 발권 여부',
    standard_price DECIMAL(10,2) NOT NULL COMMENT '티켓 표준 가격',
    sell_price DECIMAL(10,2) NULL COMMENT '티켓 판매 가격',
    PRIMARY KEY (ticket_id),
    INDEX screen_id_idx (screen_id ASC),
    INDEX theater_id_idx (theater_id ASC),
    INDEX seat_id_idx (seat_id ASC),
    INDEX reservation_id_idx (reservation_id ASC),
    CONSTRAINT fk_ticket_screen FOREIGN KEY (screen_id) REFERENCES db1.screen (screen_id),
    CONSTRAINT fk_ticket_theater FOREIGN KEY (theater_id) REFERENCES db1.theater (theater_id),
    CONSTRAINT fk_ticket_seat FOREIGN KEY (seat_id) REFERENCES db1.seat (seat_id),
    CONSTRAINT fk_ticket_reservation FOREIGN KEY (reservation_id) REFERENCES db1.reservation (reservation_id)
) ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS db1.review (
    review_id INT NOT NULL AUTO_INCREMENT COMMENT '리뷰 아이디',
    movie_id INT NOT NULL COMMENT '영화 아이디',
    member_id VARCHAR(50) NOT NULL COMMENT '회원아이디',
    score DECIMAL(3,2) NOT NULL COMMENT '평점',
    review_date DATE NOT NULL COMMENT '리뷰 날짜',
    PRIMARY KEY (review_id),
    INDEX movie_id_idx (movie_id ASC),
    INDEX member_id_idx (member_id ASC),
    CONSTRAINT fk_review_movie FOREIGN KEY (movie_id) REFERENCES db1.movie (movie_id),
    CONSTRAINT fk_review_member FOREIGN KEY (member_id) REFERENCES db1.member (member_id)
) ENGINE = InnoDB;

```