

고임김둘 팀

MySQL DB 스키마 모델 및 예제 데이터

팀장 김태랑

| 팀원 고다은

| 팀원 임

| 팀원 김서현

목차 구성

1

1차 ER 모델

2

2차 ER 모델

3

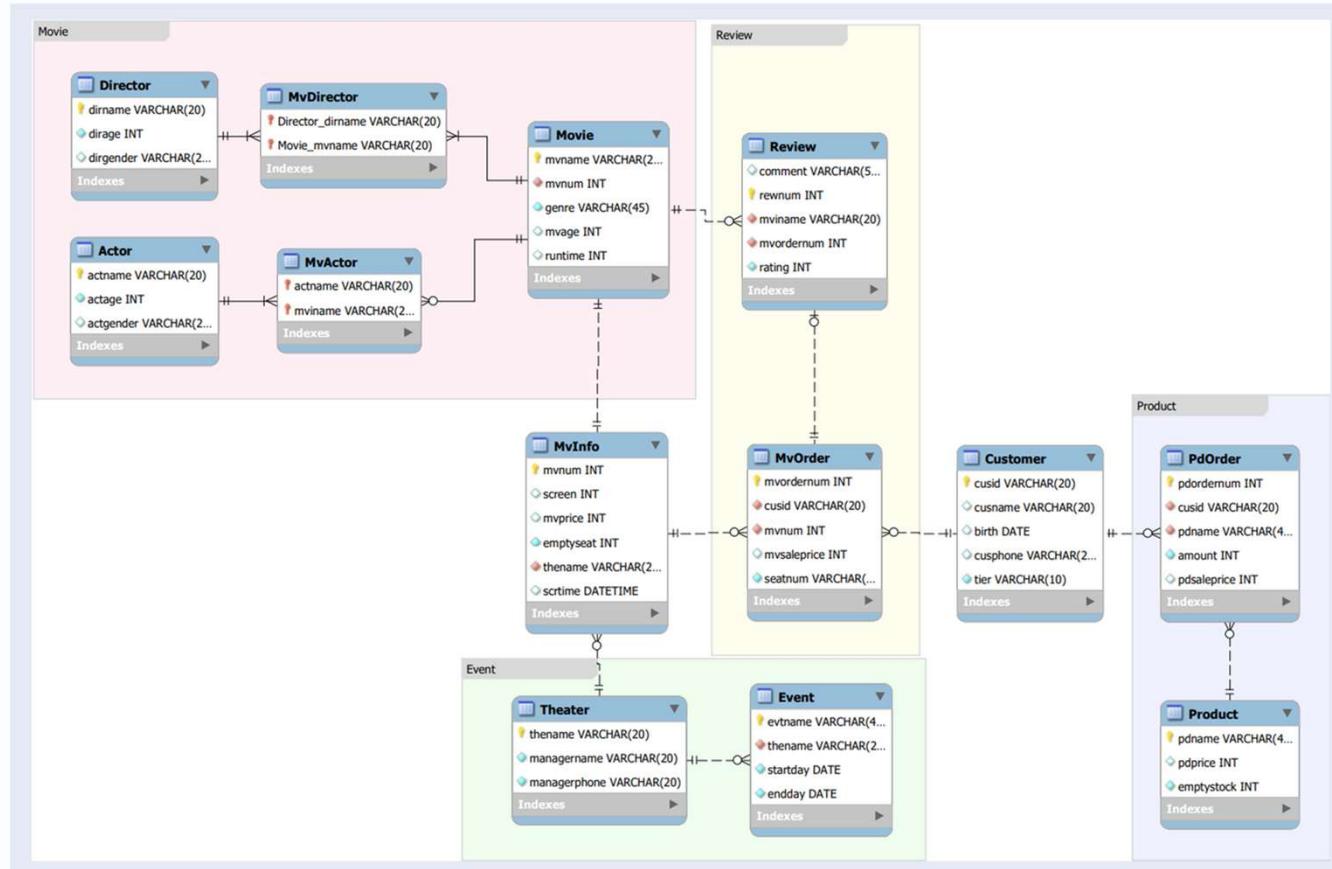
MySQL DB
스키마 모델 및
예제 데이터 분석

4

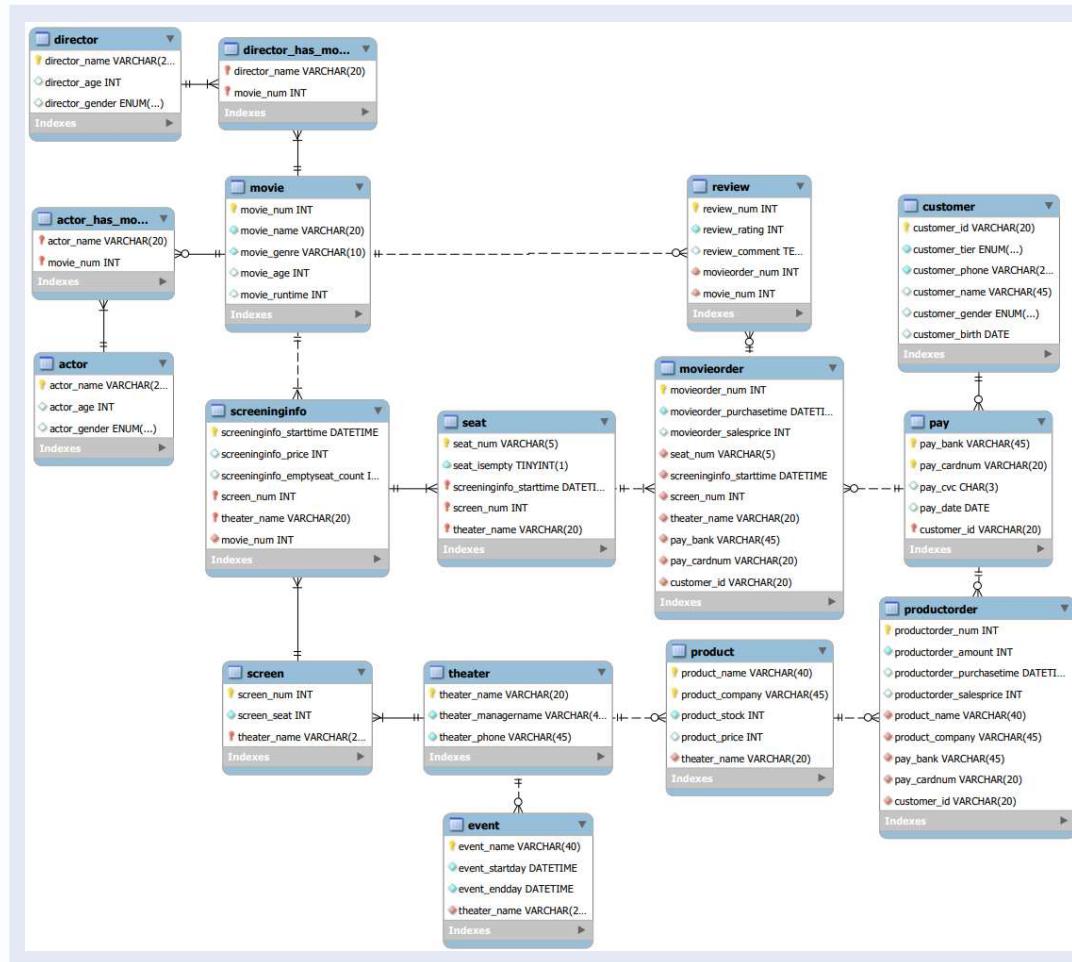
Q&A



1차 ER 모델 (수정 전)



2차 ER 모델 (수정 후)



MYSQL DB 스키마 모델 및 예제 데이터

데이터베이스 생성

```
CREATE DATABASE MovieDB;  
USE MovieDB;
```



CUSTOMER

SCHEMA

```
CREATE TABLE Customer (
    customer_id VARCHAR(20),
    customer_tier ENUM('Unrank', 'Bronze', 'Silver', 'Gold') DEFAULT 'Unrank' NOT NULL,
    customer_phone VARCHAR(20) NOT NULL UNIQUE,
    customer_name VARCHAR(45),
    customer_gender ENUM('Male', 'Female'),
    customer_birth DATE,
    PRIMARY KEY (customer_id)
);
```

DATA

```
INSERT INTO Customer VALUES
('user001', 'Silver', '010-1111-1111', '김태랑', 'Male', '2004-04-15'),
('user002', 'Gold', '010-2222-2222', '김서현', 'Female', '2003-05-03'),
('user003', 'Bronze', '010-3333-3333', '임성민', 'Male', '2005-04-09'),
('user004', 'Unrank', '010-4444-4444', '고다은', 'Female', '1999-07-07'),
('user005', 'Silver', '010-5555-5555', '임성민', 'Male', '2006-04-09');
```

PAY

SCHEMA

```
CREATE TABLE Pay (
    pay_bank VARCHAR(45),
    pay_cardnum VARCHAR(20),
    pay_cvc CHAR(3),
    pay_date DATE,
    customer_id VARCHAR(20),
    PRIMARY KEY (pay_bank, pay_cardnum, customer_id),
    FOREIGN KEY (customer_id) REFERENCES Customer(customer_id)
);
```

DATA

```
INSERT INTO Pay VALUES
('KB국민', '1234567812345678', '123', '2027-05-01', 'user001'),
('신한', '9876543210987654', '456', '2025-12-31', 'user002'),
('우리', '4567890123456789', '789', '2026-01-15', 'user003'),
('카카오', '5555444433332222', '111', '2028-10-10', 'user002'),
('NH농협', '1111222233334444', '321', '2029-07-21', 'user005');
```

THEATER

SCHEMA

```
CREATE TABLE Theater (
    theater_name VARCHAR(20) NOT NULL,
    theater_managername VARCHAR(45) NOT NULL,
    theater_phone VARCHAR(45) CHECK (theater_phone REGEXP '^[0-9-]+$') NOT NULL,
    PRIMARY KEY (theater_name)
);
```

```
INSERT INTO Theater VALUES
('CGV평광', '이민호', '02-1234-5678'),
('메가박스신촌을못가', '박보영', '02-2345-6789'),
('롯데시네마목대', '송중기', '02-3456-7890'),
('메가박스코엑스라자', '김연아', '02-4567-8901'),
('CGV공대', '유재석', '02-5678-9012');
```

DATA



SCREEN

SCHEMA

```
CREATE TABLE Screen (
    screen_num INT,
    screen_seat INT CHECK (screen_seat >= 0) NOT NULL,
    theater_name VARCHAR(20),
    PRIMARY KEY (screen_num, theater_name),
    FOREIGN KEY (theater_name) REFERENCES Theater(theater_name)
);
```

DATA

```
INSERT INTO Screen VALUES
(1, 50, 'CGV평광'),
(2, 60, 'CGV평광'),
(1, 55, '메가박스신촌을못가'),
(1, 48, '롯데시네마목대'),
(2, 70, '메가박스코엑스라지');
```



PRODUCT

SCHEMA

```
CREATE TABLE Product (
    product_name VARCHAR(40),
    product_company VARCHAR(45),
    product_stock INT CHECK (product_stock >= 0) NOT NULL,
    product_price INT CHECK (product_price >= 0),
    theater_name VARCHAR(20) NOT NULL,
    PRIMARY KEY (product_name, product_company),
    FOREIGN KEY (theater_name) REFERENCES Theater(theater_name)
);
```

```
INSERT INTO Product VALUES
('팝콘', '롯데푸드', 100, 5000, 'CGV평광'),
('콜라', '코카콜라', 120, 3000, 'CGV평광'),
('핫도그', 'CJ푸드', 50, 4000, '메가박스신촌을못가'),
('오징어', '바다식품', 30, 4500, '롯데시네마목대'),
('나쵸', '스낵월드', 80, 3500, '메가박스코엑스라지');
```

DATA



PRODUCT ORDER

SCHEMA

```
CREATE TABLE ProductOrder (
    productorder_num INT,
    productorder_amount INT CHECK (productorder_amount > 0) NOT NULL,
    productorder_purchasetime DATETIME,
    productorder_salesprice INT CHECK (productorder_salesprice >= 0),
    product_name VARCHAR(40) NOT NULL,
    product_company VARCHAR(45) NOT NULL,
    pay_bank VARCHAR(45) NOT NULL,
    pay_cardnum VARCHAR(20) NOT NULL,
    customer_id VARCHAR(20) NOT NULL,
    PRIMARY KEY (productorder_num),
    FOREIGN KEY (product_name, product_company)
        REFERENCES Product(product_name, product_company),
    FOREIGN KEY (pay_bank, pay_cardnum, customer_id)
        REFERENCES Pay(pay_bank, pay_cardnum, customer_id)
);
```

DATA

```
INSERT INTO ProductOrder VALUES
(1001, 2, '2025-04-06 10:30:00', 10000, '팝콘', '롯데푸드', '신한',
'9876543210987654', 'user002'),
(1002, 1, '2025-04-06 11:00:00', 3000, '콜라', '코카콜라', 'KB국민',
'1234567812345678', 'user001'),
(1003, 3, '2025-04-06 13:00:00', 12000, '핫도그', 'CJ푸드', '우리',
'4567890123456789', 'user003'),
(1004, 2, '2025-04-06 15:00:00', 9000, '오징어', '바다식품', '카카오',
'5555444433332222', 'user002'),
(1005, 1, '2025-04-06 16:30:00', 3500, '나쵸', '스낵월드', 'NH농협',
'1111222233334444', 'user005');
```



EVENT

SCHEMA

```
CREATE TABLE Event (
    event_name VARCHAR(40),
    event_startday DATETIME NOT NULL,
    event_endday DATETIME NOT NULL,
    theater_name VARCHAR(20) NOT NULL,
    PRIMARY KEY (event_name),
    FOREIGN KEY (theater_name) REFERENCES theater(theater_name),
    CHECK (event_endday >= event_startday)
);
```

DATA

```
INSERT INTO Event VALUES
('봄맞이할인', '2025-03-01 00:00:00', '2025-03-31 23:59:59', 'CGV평광'),
('신입회원이벤트', '2025-01-01 00:00:00', '2025-12-31 23:59:59', 'CGV공대'),
('야간할인', '2025-04-01 22:00:00', '2025-04-30 23:59:59', '메가박스코엑스라지'),
('멤버십데이', '2025-02-15 00:00:00', '2025-02-15 23:59:59', '메가박스신촌을못가'),
('발렌타인데이', '2025-02-14 00:00:00', '2025-02-14 23:59:59', '롯데시네마목대');
```

MOVIE

SCHEMA

```
CREATE TABLE Movie (
    movie_num INT,
    movie_name VARCHAR(20) NOT NULL,
    movie_genre VARCHAR(10) NOT NULL,
    movie_age INT CHECK (movie_age IN (0, 12, 15, 19)),
    movie_runtime INT CHECK (movie_runtime > 0),
    PRIMARY KEY (movie_num)
);
```

DATA

```
INSERT INTO Movie VALUES
(501, '인셉션', 'SF', 15, 148),
(502, '겨울왕국', 'Animation', 0, 102),
(503, '기생충', 'Drama', 15, 132),
(504, '범죄도시', 'Action', 19, 106),
(505, '라라랜드', 'Romance', 12, 128);
```



DIRECTOR

SCHEMA

```
CREATE TABLE Director (
    director_name VARCHAR(20),
    director_age INT CHECK (director_age >= 0),
    director_gender ENUM('Male', 'Female'),
    PRIMARY KEY (director_name)
);
```

DATA

```
INSERT INTO Director VALUES
('크리스토퍼', 53, 'Male'),
('봉준호', 55, 'Male'),
('김한민', 50, 'Male'),
('박찬욱', 58, 'Male'),
('드니빌뇌브', 56, 'Male');
```

DIRECTOR HAS MOVIE

SCHEMA

```
CREATE TABLE Director_has_Movie (
    director_name VARCHAR(20),
    movie_num INT,
    PRIMARY KEY (director_name, movie_num),
    FOREIGN KEY (director_name) REFERENCES Director(director_name),
    FOREIGN KEY (movie_num) REFERENCES Movie(movie_num)
);
```

DATA

```
INSERT INTO Director_has_Movie
VALUES
('크리스토퍼', 501),
('봉준호', 503),
('김한민', 504),
('드니빌뇌브', 505),
('박찬욱', 502);
```

ACTOR

SCHEMA

```
CREATE TABLE Actor (
    actor_name VARCHAR(20),
    actor_age INT CHECK (actor_age >= 0),
    actor_gender ENUM('Male', 'Female'),
    PRIMARY KEY (actor_name)
);
```

DATA

```
INSERT INTO Actor VALUES
('레오나르도', 49, 'Male'),
('조여정', 42, 'Female'),
('마동석', 51, 'Male'),
('에마스톤', 35, 'Female'),
('안나', 30, 'Female');
```

ACTOR HAS MOVIE

SCHEMA

```
CREATE TABLE Actor_has_Movie (
    actor_name VARCHAR(20),
    movie_num INT,
    PRIMARY KEY (actor_name, movie_num),
    FOREIGN KEY (actor_name) REFERENCES Actor(actor_name),
    FOREIGN KEY (movie_num) REFERENCES Movie(movie_num)
);
```

DATA

```
INSERT INTO Actor_has_Movie VALUES
('레오나르도', 501),
('조여정', 503),
('마동석', 504),
('에마스톤', 505),
('안나', 502);
```



SCREENINGINFO

SCHEMA

```
CREATE TABLE ScreeningInfo (
    screeninginfo_starttime DATETIME,
    screeninginfo_price INT CHECK (screeninginfo_price >= 0),
    screeninginfo_emptyseat_count INT CHECK
        (screeninginfo_emptyseat_count >= 0),
    screen_num INT,
    theater_name VARCHAR(20),
    movie_num INT NOT NULL,
    PRIMARY KEY (screeninginfo_starttime, screen_num,
                  theater_name),
    FOREIGN KEY (screen_num, theater_name) REFERENCES
        Screen(screen_num, theater_name),
    FOREIGN KEY (movie_num) REFERENCES Movie(movie_num)
);
```

DATA

```
INSERT INTO ScreeningInfo VALUES
('2025-04-06 10:00:00', 12000, 50, 1, 'CGV평광', 501),
('2025-04-06 13:30:00', 13000, 60, 2, 'CGV평광', 504),
('2025-04-06 16:00:00', 10000, 55, 1, '메가박스신촌을못가', 503),
('2025-04-06 19:00:00', 11000, 48, 1, '롯데시네마목대', 502),
('2025-04-06 21:30:00', 14000, 70, 2, '메가박스코엑스라지', 505);
```

SEAT

SCHEMA

```
CREATE TABLE Seat (
    seat_num VARCHAR(5),
    seat_isempty TINYINT(1) CHECK (seat_isempty IN (0, 1)) NOT NULL,
    screeninginfo_starttime DATETIME,
    screen_num INT,
    theater_name VARCHAR(20),
    PRIMARY KEY (seat_num, screeninginfo_starttime, screen_num, theater_name),
    FOREIGN KEY (screeninginfo_starttime, screen_num, theater_name)
        REFERENCES ScreeningInfo(screeninginfo_starttime, screen_num, theater_name)
);
```

```
INSERT INTO Seat VALUES
('A1', 1, '2025-04-06 10:00:00', 1, 'CGV평광'),
('A2', 1, '2025-04-06 13:30:00', 2, 'CGV평광'),
('B3', 1, '2025-04-06 13:30:00', 2, 'CGV평광'),
('C4', 1, '2025-04-06 16:00:00', 1, '메가박스신촌을못가'),
('D5', 1, '2025-04-06 19:00:00', 1, '롯데시네마목대');
```

DATA



SCHEMA

```
CREATE TABLE MovieOrder (
    movieorder_num INT,
    movieorder_purchasetime DATETIME NOT NULL,
    movieorder_salesprice INT CHECK (movieorder_salesprice >= 0),
    seat_num VARCHAR(5) NOT NULL,
    screeninginfo_starttime DATETIME NOT NULL,
    screen_num INT NOT NULL,
    theater_name VARCHAR(20) NOT NULL,
    pay_bank VARCHAR(45) NOT NULL,
    pay_cardnum VARCHAR(20) NOT NULL,
    customer_id VARCHAR(20) NOT NULL,
    PRIMARY KEY (movieorder_num),
    FOREIGN KEY (seat_num, screeninginfo_starttime, screen_num,
    theater_name)
        REFERENCES Seat(seat_num, screeninginfo_starttime,
    screen_num, theater_name),
    FOREIGN KEY (pay_bank, pay_cardnum, customer_id)
        REFERENCES Pay(pay_bank, pay_cardnum, customer_id)
);
```

MOVIE ORDER

DATA

```
INSERT INTO MovieOrder VALUES
(9001, '2025-04-05 22:00:00', 12000, 'A1', '2025-04-06 10:00:00', 1,
'CGV평광', '신한', '9876543210987654', 'user002'),
(9002, '2025-04-06 11:30:00', 13000, 'B3', '2025-04-06 13:30:00', 2,
'CGV평광', 'KB국민', '1234567812345678', 'user001'),
(9003, '2025-04-06 14:20:00', 10000, 'C4', '2025-04-06 16:00:00', 1,
'메가박스신촌을못가', '우리', '4567890123456789', 'user003'),
(9004, '2025-04-06 17:30:00', 11000, 'D5', '2025-04-06 19:00:00', 1,
'롯데시네마목대', '카카오', '5555444433332222', 'user002'),
(9005, '2025-04-06 20:00:00', 14000, 'A2', '2025-04-06 13:30:00',
2, 'CGV평광', 'NH농협', '1111222233334444', 'user005');
```

REVIEW

SCHEMA

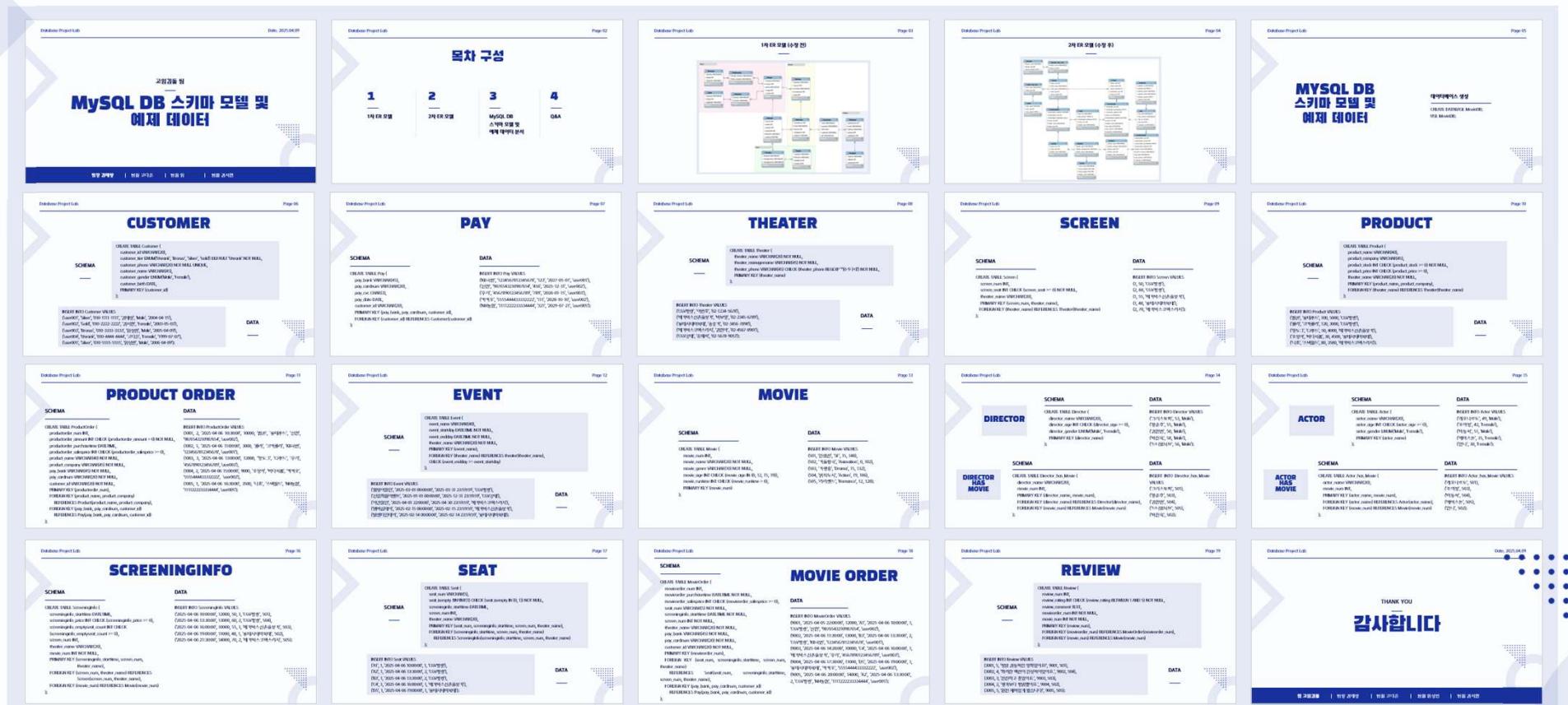
```
CREATE TABLE Review (
    review_num INT,
    review_rating INT CHECK (review_rating BETWEEN 1 AND 5) NOT NULL,
    review_comment TEXT,
    movieorder_num INT NOT NULL,
    movie_num INT NOT NULL,
    PRIMARY KEY (review_num),
    FOREIGN KEY (movieorder_num) REFERENCES MovieOrder(movieorder_num),
    FOREIGN KEY (movie_num) REFERENCES Movie(movie_num)
);
```

```
INSERT INTO Review VALUES
(3001, 5, '정말 감동적인 영화였어요!', 9001, 501),
(3002, 4, '화려한 액션이 인상적이었어요.', 9002, 504),
(3003, 3, '잔잔하고 좋았어요.', 9003, 503),
(3004, 2, '생각보다 평범했어요.', 9004, 502),
(3005, 5, '완전 재미있게 봤습니다!', 9005, 505);
```

DATA



Q&A



THANK YOU

감사합니다