# BookMyShow Ticketing System

P1: Entity Listing, Attributes, and Table Structures

Entities and Their Attributes:

1. Theatre
   * Theatre\_ID (Primary Key)
   * Name
   * Location
2. Movie
   * Movie\_ID (Primary Key)
   * Title
   * Genre
   * Duration (in minutes)
   * Language
3. Show
   * Show\_ID (Primary Key)
   * Movie\_ID (Foreign Key referencing Movie\_ID)
   * Theatre\_ID (Foreign Key referencing Theatre\_ID)
   * Show\_Date (Date of the show)
   * Show\_Time (Time of the show)
4. Booking
   * Booking\_ID (Primary Key)
   * Show\_ID (Foreign Key referencing Show\_ID)
   * User\_Name
   * Seats\_Booked
   * Booking\_Date
5. Seat
   * Seat\_ID (Primary Key)
   * Theatre\_ID (Foreign Key referencing Theatre\_ID)
   * Seat\_Number
6. Show\_Seat
   * Show\_Seat\_ID (Primary Key)
   * Seat\_ID (Foreign Key referencing Seat\_ID)
   * Show\_ID (Foreign Key referencing Show\_ID)
   * Status (Available, Booked, etc.)

ER Diagram

The ER Diagram will show relationships such as:

* Theatre has a one-to-many relationship with Show
* Movie has a one-to-many relationship with Show
* Show has a many-to-many relationship with Seat through Show\_Seat
* Show has a one-to-many relationship with Booking

SQL Table Creation

1. Theatre Table

CREATE TABLE Theatre (

Theatre\_ID INT AUTO\_INCREMENT PRIMARY KEY,

Name VARCHAR(100) NOT NULL,

Location VARCHAR(255) NOT NULL

);

2. Movie Table

CREATE TABLE Movie (

Movie\_ID INT AUTO\_INCREMENT PRIMARY KEY,

Title VARCHAR(255) NOT NULL,

Genre VARCHAR(100),

Duration INT,

Language VARCHAR(100)

);

3. Show Table

CREATE TABLE Show (

Show\_ID INT AUTO\_INCREMENT PRIMARY KEY,

Movie\_ID INT,

Theatre\_ID INT,

Show\_Date DATE,

Show\_Time TIME,

FOREIGN KEY (Movie\_ID) REFERENCES Movie(Movie\_ID),

FOREIGN KEY (Theatre\_ID) REFERENCES Theatre(Theatre\_ID)

);

4. Booking Table

CREATE TABLE Booking (

Booking\_ID INT AUTO\_INCREMENT PRIMARY KEY,

Show\_ID INT,

User\_Name VARCHAR(100),

Seats\_Booked INT,

Booking\_Date DATETIME,

FOREIGN KEY (Show\_ID) REFERENCES Show(Show\_ID)

);

5. Seat Table

CREATE TABLE Seat (

Seat\_ID INT AUTO\_INCREMENT PRIMARY KEY,

Theatre\_ID INT,

Seat\_Number VARCHAR(10),

FOREIGN KEY (Theatre\_ID) REFERENCES Theatre(Theatre\_ID)

);

6. Show\_Seat Table (to maintain many-to-many relationship between Show and Seat)

CREATE TABLE Show\_Seat (

Show\_Seat\_ID INT AUTO\_INCREMENT PRIMARY KEY,

Seat\_ID INT,

Show\_ID INT,

Status ENUM('Available', 'Booked') DEFAULT 'Available',

FOREIGN KEY (Seat\_ID) REFERENCES Seat(Seat\_ID),

FOREIGN KEY (Show\_ID) REFERENCES Show(Show\_ID)

);

Sample Data Insertion

1. Sample data for Theatre

INSERT INTO theatres (name, location) VALUES ('PVR', 'Nexus (Forum Mall)');

2. Sample data for Movie

INSERT INTO movies (title, genre, duration, language) VALUES ('Dasara', 'Drama', 148, 'Telugu');

INSERT INTO movies (title, genre, duration, language) VALUES ('Kisi Ka Bhai Kisi Ki Jaan', 'Drama', 161, 'Hindi');

INSERT INTO movies (title, genre, duration, language) VALUES ('Tu Jhoothi Main Makkaar', 'Romantic Comedy', 164, 'Hindi');

INSERT INTO movies (title, genre, duration, language) VALUES ('Avatar: The Way of Water', 'Action', 192, 'English');

3. Sample data for Show

INSERT INTO shows (movie\_id, theatre\_id, show\_date, show\_time) VALUES (1, 1, '2024-04-25', '12:15:00');

-- Shows for 'Kisi Ka Bhai Kisi Ki Jaan'

INSERT INTO shows (movie\_id, theatre\_id, show\_date, show\_time) VALUES (2, 1, '2024-04-25', '01:00:00');

INSERT INTO shows (movie\_id, theatre\_id, show\_date, show\_time) VALUES (2, 1, '2024-04-25', '04:10:00');

INSERT INTO shows (movie\_id, theatre\_id, show\_date, show\_time) VALUES (2, 1, '2024-04-25', '06:20:00');

INSERT INTO shows (movie\_id, theatre\_id, show\_date, show\_time) VALUES (2, 1, '2024-04-25', '07:30:00');

INSERT INTO shows (movie\_id, theatre\_id, show\_date, show\_time) VALUES (2, 1, '2024-04-25', '10:30:00');

-- Shows for 'Tu Jhoothi Main Makkaar'

INSERT INTO shows (movie\_id, theatre\_id, show\_date, show\_time) VALUES (3, 1, '2024-04-25', '01:15:00');

-- Shows for 'Avatar: The Way of Water'

INSERT INTO shows (movie\_id, theatre\_id, show\_date, show\_time) VALUES (4, 1, '2024-04-25', '01:20:00');

4. Sample data for Seat

INSERT INTO seats (theatre\_id, seat\_number) VALUES (1, 'A1');

INSERT INTO seats (theatre\_id, seat\_number) VALUES (1, 'A2');

INSERT INTO seats (theatre\_id, seat\_number) VALUES (1, 'A3');

5. Sample data for Booking

INSERT INTO bookings (show\_id, user\_name, seats\_booked, booking\_date) VALUES (1, 'John Doe', 2, '2024-04-24 10:00:00');

6. Sample data for Show\_Seat

INSERT INTO show\_seats (seat\_id, show\_id, status) VALUES (1, 1, 'Booked');

INSERT INTO show\_seats (seat\_id, show\_id, status) VALUES (2, 1, 'Available');

INSERT INTO show\_seats (seat\_id, show\_id, status) VALUES (3, 2, 'Available');

This schema satisfies all the normalization forms:

* 1NF: Each column has atomic values.
* 2NF: All non-key attributes are fully functionally dependent on the primary key.
* 3NF: There is no transitive dependency.
* BCNF: Each functional dependency involves a superkey.

P2: SQL Query to List All Shows on a Given Date at a Given Theatre

Given the date and theatre, this query will list all shows with their respective timings:

SELECT

    s.show\_id,

    m.title AS Movie\_Title,

    s.show\_time

FROM

    shows s

JOIN

    movies m ON s.movie\_id = m.movie\_id

JOIN

    theatres t ON s.theatre\_id = t.theatre\_id

WHERE

    s.show\_date = '2024-04-25'

    AND t.name = 'PVR';

Explanation:

* This query joins the Show, Movie, and Theatre tables to fetch show details.
* It filters by the specified Show\_Date and Theatre Name to display relevant shows and timings.