### **Book My Show Ticketing System**

### **P1: Database Design**

#### **Entities and Attributes:**

1. **Theatre**
   * **Attributes**: Theatre\_ID (PK), Theatre\_Name, Theatre\_Location
   * **Description**: Represents a theatre where movies are shown.
2. **Movie**
   * **Attributes**: Movie\_ID (PK), Movie\_Title, Genre, Duration (in minutes), Language
   * **Description**: Represents a movie that is available for screening.
3. **Movie\_Show**
   * **Attributes**: Show\_ID (PK), Theatre\_ID (FK), Movie\_ID (FK), Show\_Date, Show\_Time, Price
   * **Description**: Represents a particular showtime of a movie at a given theatre on a specific date.
4. **Ticket**
   * **Attributes**: Ticket\_ID (PK), Show\_ID (FK), Customer\_Name, Customer\_Email, Ticket\_Quantity, Total\_Price
   * **Description**: Represents a ticket purchased by a customer for a specific show.
5. **Booking**
   * **Attributes**: Booking\_ID (PK), Ticket\_ID (FK), Booking\_Date
   * **Description**: Represents a booking made for a particular ticket.

#### **Relationships:**

* A **theatre** can have multiple **shows**.
* A **movie** can have multiple **shows**.
* A **show** can have multiple **tickets** (since multiple customers can book tickets for the same show).
* A **ticket** can be linked to a **booking** (a booking can contain multiple tickets).

#### **Normalization:**

* The tables above are designed to follow 1NF (Atomic values), 2NF (No partial dependency), 3NF (No transitive dependency), and BCNF (Every determinant is a candidate key).
  + **1NF**: All columns have atomic values (no repeating groups).
  + **2NF**: No partial dependency, meaning non-key attributes are fully dependent on the primary key.
  + **3NF**: No transitive dependency; non-key attributes do not depend on other non-key attributes.
  + **BCNF**: All non-trivial functional dependencies have a candidate key as the determinant.

#### **Table Structure and Sample Entries:**

##### **1. Theatre Table**

CREATE TABLE Theatre (

Theatre\_ID INT AUTO\_INCREMENT PRIMARY KEY,

Theatre\_Name VARCHAR(255) NOT NULL,

Theatre\_Location VARCHAR(255) NOT NULL

);

-- Sample Data

INSERT INTO Theatre (Theatre\_Name, Theatre\_Location) VALUES

('PVR Cinemas', 'Dehradun'),

('Cinepolis', 'Gurugram');

##### **2. Movie Table**

CREATE TABLE Movie (

Movie\_ID INT AUTO\_INCREMENT PRIMARY KEY,

Movie\_Title VARCHAR(255) NOT NULL,

Genre VARCHAR(100),

Duration INT,

Language VARCHAR(100)

);

-- Sample Data

INSERT INTO Movie (Movie\_Title, Genre, Duration, Language) VALUES

('La La Land', 'Romance', 181, 'English'),

('Pushpa 2', Action, 210, 'Hindi');

##### **3. Show Table**

CREATE TABLE Movie\_Show(

Show\_ID INT AUTO\_INCREMENT PRIMARY KEY,

Theatre\_ID INT,

Movie\_ID INT,

Show\_Date DATE,

Show\_Time TIME,

Price DECIMAL(10, 2),

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

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

);

-- Sample Data

INSERT INTO Movie\_Show(Theatre\_ID, Movie\_ID, Show\_Date, Show\_Time, Price) VALUES

(1, 1, '2024-12-25', '14:00:00', 12.50),

(1, 2, '2024-12-25', '16:30:00', 10.00),

(2, 1, '2024-12-25', '13:00:00', 15.00);

##### **4. Ticket Table**

CREATE TABLE Ticket (

Ticket\_ID INT AUTO\_INCREMENT PRIMARY KEY,

Show\_ID INT,

Customer\_Name VARCHAR(255),

Customer\_Email VARCHAR(255),

Ticket\_Quantity INT,

Total\_Price DECIMAL(10, 2),

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

);

-- Sample Data

INSERT INTO Ticket (Show\_ID, Customer\_Name, Customer\_Email, Ticket\_Quantity, Total\_Price) VALUES

(1, 'Sachin Deoli', 'sachin@example.com', 2, 25.00),

(2, 'Ayush', 'ayush@example.com', 1, 10.00);

##### **5. Booking Table**

CREATE TABLE Booking (

Booking\_ID INT AUTO\_INCREMENT PRIMARY KEY,

Ticket\_ID INT,

Booking\_Date DATETIME,

FOREIGN KEY (Ticket\_ID) REFERENCES Ticket(Ticket\_ID)

);

-- Sample Data

INSERT INTO Booking (Ticket\_ID, Booking\_Date) VALUES

(1, '2024-12-24 10:30:00'),

(2, '2024-12-24 11:00:00');

### **P2: Query to List All Shows on a Given Date at a Given Theatre Along with Show Timings**

#### **Query:**

SELECT

S.Show\_ID,

M.Movie\_Title,

S.Show\_Date,

S.Show\_Time,

S.Price

FROM

Movie\_Show S

JOIN

Movie M ON S.Movie\_ID = M.Movie\_ID

JOIN

Theatre T ON S.Theatre\_ID = T.Theatre\_ID

WHERE

S.Show\_Date = '2024-12-25' -- Example date

AND T.Theatre\_Name = 'PVR Cinemas'; -- Example theatre

ORDER BY

S.Show\_Time;

#### **Explanation:**

* The query joins the Show, Movie, and Theatre tables.
* It filters the shows based on the given Show\_Date and Theatre\_Name.
* The results include the Show\_ID, Movie\_Title, Show\_Date, Show\_Time, and Price of each show.
* The output is ordered by Show\_Time.

### **Expected Output:**

For the example inputs (2024-12-25 date and 'PVR Cinemas' theatre), the query should return a list like:

| **Show\_ID** | **Movie\_Title** | **Show\_Date** | **Show\_Time** | **Price** |
| --- | --- | --- | --- | --- |
| 1 | La la land | 2024-12-25 | 14:00:00 | 12.50 |
| 2 | Pushpa 2 | 2024-12-25 | 16:30:00 | 10.00 |