**Identify Key Entities:**

1. **Users** (or Customers)
2. **Movies**
3. **Theaters**
4. **Shows** (specific movie screenings at specific times)
5. **Tickets**
6. **Payments**
7. **Seats**

**Define Relationships:**

1. **Users** book **Tickets**.
2. **Movies** are shown in **Theaters**.
3. **Theaters** host **Shows**.
4. **Shows** have multiple **Seats**.
5. **Tickets** are associated with **Payments**.

**Attributes for Each Entity:**

1. **Users**
   * UserID (Primary Key)
   * Name
   * Email
   * PhoneNumber
2. **Movies**
   * MovieID (Primary Key)
   * Title
   * Genre
   * Language
   * Duration
   * Rating
3. **Theaters**
   * TheaterID (Primary Key)
   * Name
   * Location
   * TotalSeats
4. **Shows**
   * ShowID (Primary Key)
   * MovieID (Foreign Key)
   * TheaterID (Foreign Key)
   * StartTime
   * EndTime
5. **Tickets**
   * TicketID (Primary Key)
   * ShowID (Foreign Key)
   * UserID (Foreign Key)
   * SeatNumber
   * BookingTime
6. **Payments**
   * PaymentID (Primary Key)
   * TicketID (Foreign Key)
   * PaymentMode (Credit Card, UPI, etc.)
   * Amount
   * PaymentStatus (Success, Failure)
7. **Seats**
   * SeatID (Primary Key)
   * ShowID (Foreign Key)
   * Status (Available/Booked)
8. **Movie\_Rating**

* Review\_Id
* User\_Id
* Movie\_name
* Review\_Date
* Rating (Out of 5)

**Relationships Diagram**

1. **Users → Tickets:** One user can book multiple tickets.
2. **Movies → Shows:** One movie can have multiple shows.
3. **Theaters → Shows:** One theater can host multiple shows.
4. **Shows → Tickets:** One show can have multiple tickets booked.
5. **Tickets → Payments:** One ticket can have one payment.
6. **Shows → Seats:** One show has multiple seats.
7. **Movie → Review:** One Movie has One Review
8. **User → Review:** One User gives one movie Review

**Database Design**

**Tables and Attributes**

1. **Users**
   * User\_ID (Primary Key)
   * Email (Unique, Not Null)
   * Mobile\_No (Unique, Not Null)
   * Name (Not Null)
2. **Movies**
   * Movie\_ID (Primary Key)
   * Movie\_Name (Not Null)
   * Language (Not Null)
   * Duration (Not Null)
   * Genre (Not Null)
   * Rating (Not Null)
3. **Theatres**
   * Theatre\_ID (Primary Key)
   * Theatre\_Name (Not Null)
   * Seating\_Capacity (Not Null)
   * Location (Not Null)
4. **Seats**
   * Seat\_ID (Primary Key)
   * Theatre\_ID (Foreign Key referencing Theatres)
   * Status (Enum: 'Booked', 'Available')
   * Show\_Time (Not Null)
5. **Shows**
   * Show\_ID (Primary Key)
   * Theatre\_ID (Foreign Key referencing Theatres)
   * Movie\_ID (Foreign Key referencing Movies)
   * Show\_Type (Enum: 'Morning', '1st Show', '2nd Show', 'Matinee')
   * Show\_Time (Not Null)
6. **Tickets**
   * Ticket\_ID (Primary Key)
   * User\_ID (Foreign Key referencing Users)
   * Seat\_No (Foreign Key referencing Seats)
   * Show\_ID (Foreign Key referencing Shows)
7. **Payments**
   * Payment\_ID (Primary Key)
   * Payment\_Mode (Enum: 'Credit Card', 'Debit Card', 'UPI', 'Cash', etc.)
   * Amount (Not Null)
   * Ticket\_ID (Foreign Key referencing Tickets)
   * Status (Enum: 'Paid', 'Pending', 'Refunded')

**Relationships and Integrity Constraints:**

* **User to Tickets**: A user can buy many tickets; each ticket belongs to a single user (1:M).
* **Shows to Movies/Theatres**: A show must be associated with one movie and one theatre (1:M).
* **Tickets to Seats**: A ticket is linked to a specific seat and show (1:M).
* **Payments to Tickets**: A payment is linked to one ticket (1:M).

**Design Considerations:**

* Use **composite keys** if required for detailed reporting, like linking seats to specific times.
* Ensure **unique constraints** for seat availability to avoid double booking.
* Include **indexes** on fields such as Show\_Time and Theatre\_ID for performance.