**🎬 Movie Booking System – Database Documentation**

**📌 1. Introduction**

The **Movie Booking System** is a database management solution designed to handle online movie ticket reservations efficiently.  
It centralizes and manages data related to **users, movies, theatres, shows, seats, bookings, payments, and reviews**, ensuring accuracy, consistency, and scalability.

This project demonstrates the use of **relational database concepts, normalization, relationships, and constraints** to simulate a real-world movie booking platform.

**📌 2. Objectives**

The primary objectives of this system are:

* To provide a structured way of storing and retrieving user, movie, and theatre information.
* To manage multiple shows across different theatres and screens.
* To ensure accurate seat allocation and avoid double booking.
* To track booking transactions with payment details.
* To allow users to review and rate movies.
* To generate useful insights like revenue, occupancy, and movie popularity.

**📌 3. System Overview**

**🔹 Entities Involved**

* **Users** → Customers who book tickets.
* **Movies** → Collection of films available for booking.
* **Theatres** → Venues where movies are shown.
* **Shows** → Scheduled screenings of movies in theatres.
* **Seats** → Seat mapping for each theatre and screen.
* **Bookings** → Ticket reservations made by users.
* **Payments** → Transaction details for each booking.
* **Reviews** → Ratings and comments given by users.

**🔹 Relationships**

* A **user** can make multiple **bookings**.
* A **movie** can be scheduled in multiple **theatres** via **shows**.
* A **booking** can include multiple **seats**.
* Each **booking** has an associated **payment**.
* A **user** can submit reviews for movies.

**📌 4. Key Features**

* **User Management** → Registration, login, and profile data.
* **Movie Management** → Genres, languages, duration, and release details.
* **Show Scheduling** → Linking movies to theatres with timing & pricing.
* **Seat Allocation** → Regular & premium seats with availability.
* **Booking System** → Confirmed / Cancelled booking records.
* **Payment Tracking** → Secure payments with multiple methods.
* **Review System** → User ratings and feedback on movies.

**📌 5. Sample Queries**

**🔹 User Queries**

-- Fetch all registered users

SELECT \* FROM users;

**🔹 Movie Queries**

-- List movies released after 2020

SELECT title, genre, release\_date

FROM movies

WHERE release\_date > '2020-01-01';

**🔹 Booking Queries**

-- Show bookings with user name and movie title

SELECT b.booking\_id, u.name, m.title, b.total\_amount, b.status

FROM bookings b

JOIN users u ON b.user\_id = u.user\_id

JOIN shows s ON b.show\_id = s.show\_id

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

**🔹 Insights**

-- Total revenue from confirmed bookings

SELECT SUM(total\_amount) AS total\_revenue

FROM bookings

WHERE status = 'CONFIRMED';

-- Find the most popular movie

SELECT m.title, COUNT(b.booking\_id) AS total\_bookings

FROM bookings b

JOIN shows s ON b.show\_id = s.show\_id

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

WHERE b.status = 'CONFIRMED'

GROUP BY m.title

ORDER BY total\_bookings DESC

LIMIT 1;

**📌 6. Advantages of the System**

* Ensures **data consistency** using relational integrity.
* Provides **scalability** for managing multiple theatres and shows.
* Tracks **payments and revenue** in real time.
* Improves **user experience** by avoiding seat conflicts.
* Generates **business insights** like occupancy and movie popularity.

**📌 7. Conclusion**

The **Movie Booking System Database** provides a robust and efficient way to manage ticket booking operations.  
It integrates multiple components — users, movies, theatres, shows, bookings, and payments — into a single cohesive platform.

With further enhancements such as **admin roles, discount management, and real-time seat tracking**, this project can be extended into a fully functional movie ticket booking application.