

Introduction to Data Mining

Mini Project 1

BOOK YOUR SHOW

Our Iteration on What BookMyShow's Database Would Look Like

By Amog Rao, Mudit Surana, and Nimrat Kaur



SCOPE

Purpose

Recreating the BookMyShow Database with a **User-Centric Focus** for **Movie Bookings**.

The Scope of Our Database is For:

- Movie-Lovers/Goers
- ~~Multiplex owners~~
- ~~The Cast and Crew~~

And the Purpose is to Book/Order:

- Tickets
- ~~Beverages~~



FUNCTIONAL REQUIREMENTS



List of Locations

You first select a city, don't you? Or do you also just book hawa mein like my friends :(



Users

There's no hiding to your bookings, mummy ko pata chaljaega lol



Languages & Genres

"South" should be translated to Hindi na?

Horror = Comedy 😏



Theatres, Audis & Seats

Oppenheimer = IMAX (center seats)

Barbie = Phone pe chalega



Shows

Late Night Shows OP

Disclaimer: Do this only if you have prior permission from the warden (or take him along lol)

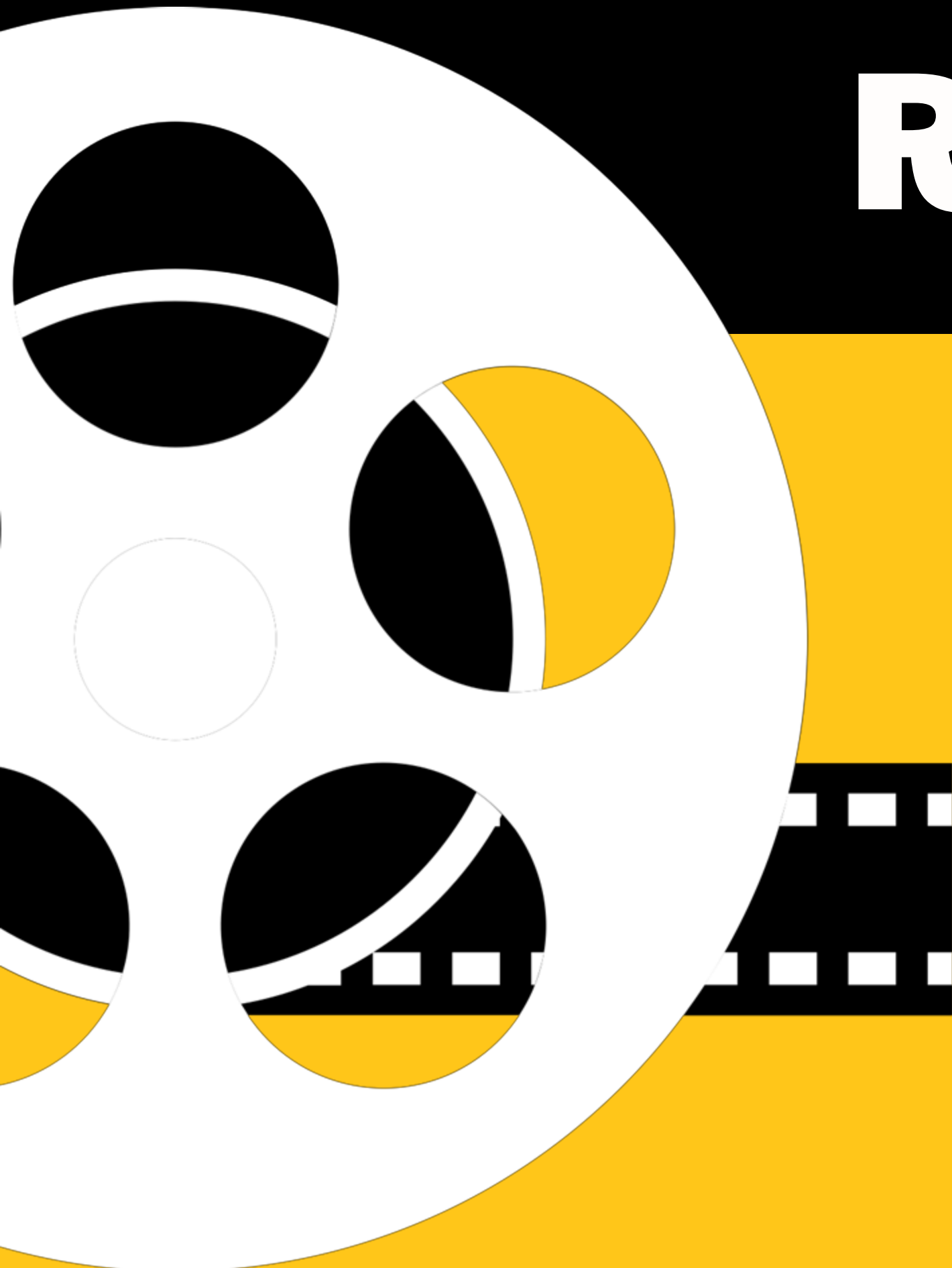


Tickets & Bookings

How many tickets? Booking History

Cancellations? (only if it's Adipurush or SOTY 2 lol)

Reviews



PERFECTION



GREAT



GOOD

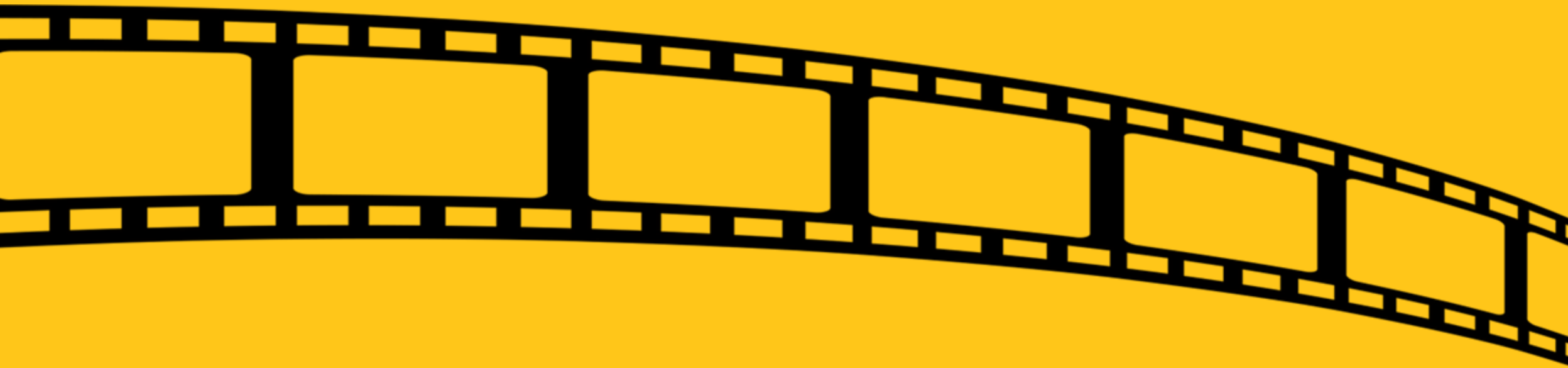


DISAPPOINTMENT



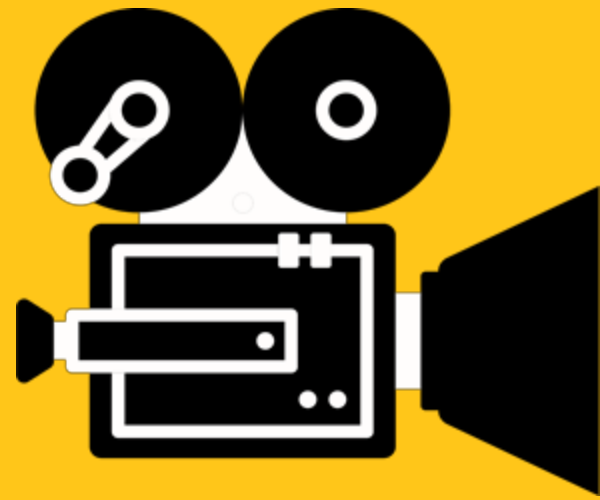
DISASTER





And
BOX OFFICE STATS

Kya baat hai kya cheez hai paisa

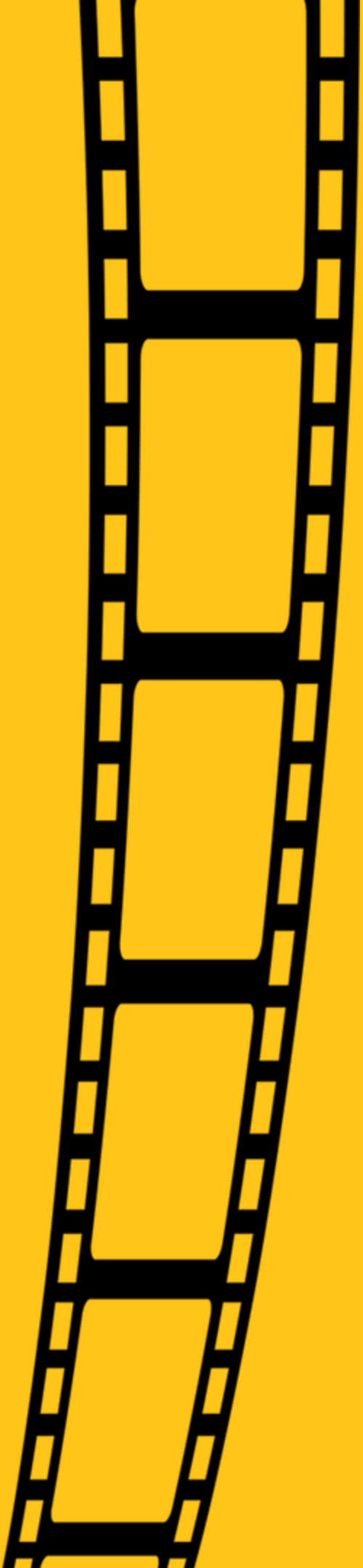


What's in Store?

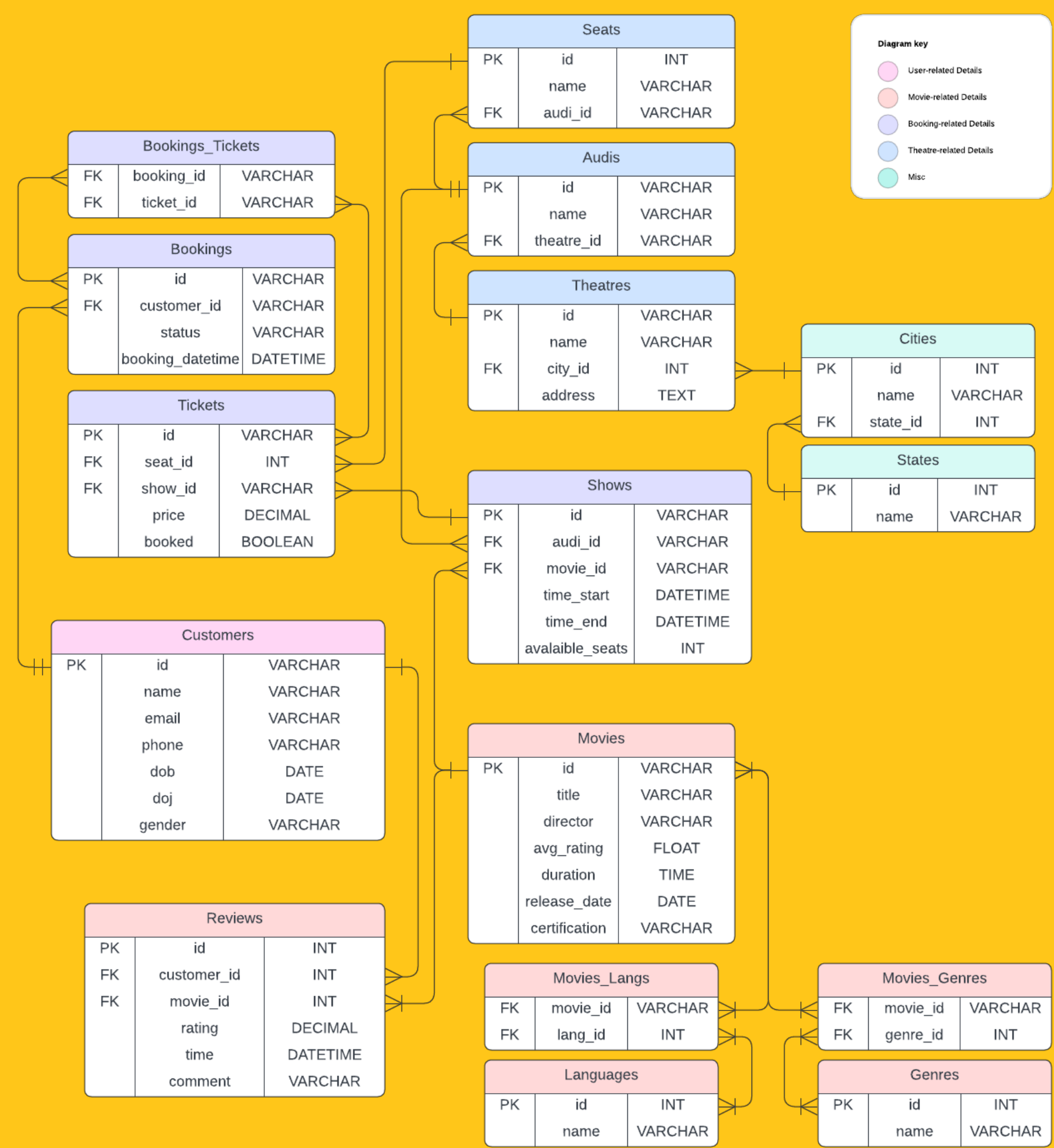
- 16 Tables
- 8 Views
- 4 Triggers
- 5 Indexes



**= 8 Major
Functions**



REPRESENTATION



OPTIMIZATIONS

INDEX

- For available seats in a show
- For a seat's auditorium
- For a ticket's seat
- For the booked status of a ticket
- For the price of a ticket

TRIGGERS

- Updating available seats for a show
- Updating movie ratings
- Automatically add seats for a show

```
sqlite> SELECT * FROM AllTimeBoxOffice;  
INDMV001,450  
INDMV002,360  
sqlite> UPDATE Bookings SET status="cancelled" WHERE id="BK002";  
sqlite> SELECT * FROM AllTimeBoxOffice;  
INDMV001,300  
INDMV002,180
```



OPTIMIZATIONS

VIEWS

AllTimeBoxOffice	
Movies.title	VARCHAR
total_box_office	DECIMAL

LastMonthBoxOffice	
Movies.title	VARCHAR
monthly_box_office	DECIMAL

LastWeekBoxOffice	
Movies.title	VARCHAR
weekly_box_office	DECIMAL

WeeklyOccupancyRates	
Movies.title	VARCHAR
occupancy_rate	INT

DailyTicketsSoldPerPeriod	
show_date	DATE
tickets_sold	INT

WeeklyTicketsByTheatre	
Theatres.name	VARCHAR
tickets_sold	INT

WeeklyTicketsByCity	
Cities.name	VARCHAR
tickets_sold	INT

WeeklyTicketsByMovie	
Movie.title	VARCHAR
tickets_sold	INT



QUERIES

```
SELECT
    Theatres.name AS theatre_name,
    COUNT(*) AS tickets_sold
FROM Tickets
JOIN Shows ON Tickets.show_id = Shows.id
JOIN Audis ON Shows.audi_id = Audis.id
JOIN Theatres ON Audis.theatre_id = Theatres.id
WHERE Shows.time_start >= DATE('now', '-7 days')
GROUP BY Theatres.name;
```

```
Carnival Cinemas – Infiniti Mall|10
INOX – R-City Mall|20
PVR Cinemas – Phoenix Mall|20
```

```
SELECT
    Cities.name AS city_name,
    COUNT(*) AS tickets_sold
FROM Tickets
JOIN Shows ON Tickets.show_id = Shows.id
JOIN Audis ON Shows.audi_id = Audis.id
JOIN Theatres ON Audis.theatre_id = Theatres.id
JOIN Cities ON Theatres.city_id = Cities.id
WHERE Shows.time_start >= DATE('now', '-7 days')
GROUP BY Cities.name;
```

```
Mumbai|50
```



QUERIES

```
/*no. of tickets sold in the last week by movie*/  
SELECT  
    Movies.title AS movie_title,  
    COUNT(*) AS tickets_sold  
FROM Tickets  
JOIN Shows ON Tickets.show_id = Shows.id  
JOIN Movies ON Shows.movie_id = Movies.id  
WHERE Shows.time_start >= DATE('now', '-7 days')  
GROUP BY Movies.title;
```

```
/*total tickets sold for a movie in its run time*/  
SELECT  
    Movies.title,  
    COUNT(*) as tickets_sold  
FROM Tickets  
JOIN Shows ON Tickets.show_id = Shows.id  
JOIN Movies ON Shows.movie_id = Movies.id  
WHERE Movies.title = 'Baahubali: The Beginning' --  
GROUP BY Movies.title;
```

```
3 Idiots|10  
Baahubali: The Beginning|10  
Dangal|10  
PK|10  
Padmaavat|10
```

```
Baahubali: The Beginning|10
```



QUERIES

```
SELECT
  CASE
    WHEN time_start >= TIME('06:00:00') AND time_start < TIME('09:00:00') THEN '6AM-9AM'
    WHEN time_start >= TIME('09:00:00') AND time_start < TIME('12:00:00') THEN '9AM-12PM'
    WHEN time_start >= TIME('12:00:00') AND time_start < TIME('15:00:00') THEN '12PM-3PM'
    WHEN time_start >= TIME('15:00:00') AND time_start < TIME('18:00:00') THEN '3PM-6PM'
    WHEN time_start >= TIME('18:00:00') AND time_start < TIME('21:00:00') THEN '6PM-9PM'
    WHEN time_start >= TIME('21:00:00') AND time_start < TIME('24:00:00') THEN '9PM-12AM'
    ELSE 'Other'
  END AS time_period,
  COUNT(*) AS tickets_sold
FROM Tickets
JOIN Shows ON Tickets.show_id = Shows.id
WHERE Shows.time_start >= DATE('now', '-7 days')
GROUP BY time_period;
```

6PM-9PM | 50

/*net revenue by movie*/

```
SELECT
  Movies.title,
  SUM(Tickets.price) AS net_revenue
FROM Tickets
JOIN Shows ON Tickets.show_id = Shows.id
JOIN Movies ON Shows.movie_id = Movies.id
GROUP BY Movies.title;
```

```
3 Idiots|2170
Baahubali: The Beginning|2210
Dangal|1740
PK|1980
Padmaavat|2140
```



LIMITATIONS

- Limited Consideration for the **Owners** such as assessing profits
- Cannot be used to generate **recommendations** based on cast and crew
- Accounting for **Regional Differences** in the Theatre Systems
- Limited only to **movie shows** in theatres unlike BookMyShow
- Lacks configuration for the seats
- Cannot purchase **food combos**



**THANK
YOU!**