

DCDSL PROJECT AY 2025-26

Phase 1 Presentation

Title of the Project:
Data-Driven Insights for BookMyShow

Outline of the Presentation

1. Names of all the group members and their work distribution
2. Introduction
3. Problem Statement
4. Objectives of the Project
5. Methodology
6. ER Diagram
7. ER Diagram To Relational Schema Conversion
8. Database Table Creation and Data Insertion
9. Basic and Advanced Query Implementation
10. Project Outcome
11. Conclusion
12. References
13. A photograph of the presentation.

Group Members



KASHISH CHELWANI

24070126082

ER Diagrams, insertion and queries



KAVISH NAG

24070126085

Scheme creation, insertion and queries



KUNAL JHINDAL

24070126098

Tables creation , insertion and queries

Introduction

BookMyShow is one of India's largest online ticket booking platforms, enabling users to book tickets for movies, events, and live shows. With millions of users, the platform generates vast amounts of data on user behavior, movie performance, theater occupancy, payment transactions, and customer reviews.

This project uses SQL-based analysis to explore the BookMyShow dataset and uncover meaningful insights that can help enhance platform efficiency, improve customer experience, and maximize revenue generation.





Problem Statement

BookMyShow, as a leading online ticketing platform, manages millions of users, theaters, movies, and transactions daily. However, the platform faces challenges in effectively understanding user behavior, movie performance, theater efficiency, seat utilization, revenue management, and payment reliability. While vast amounts of data are generated, critical issues remain:

- Identifying patterns in user engagement and retention
- Measuring and improving movie success metrics and theater performance
- Optimizing seat utilization and dynamic pricing strategies
- Detecting bottlenecks in revenue generation and transaction failures
- Leveraging reviews and sentiment analysis to improve customer satisfaction

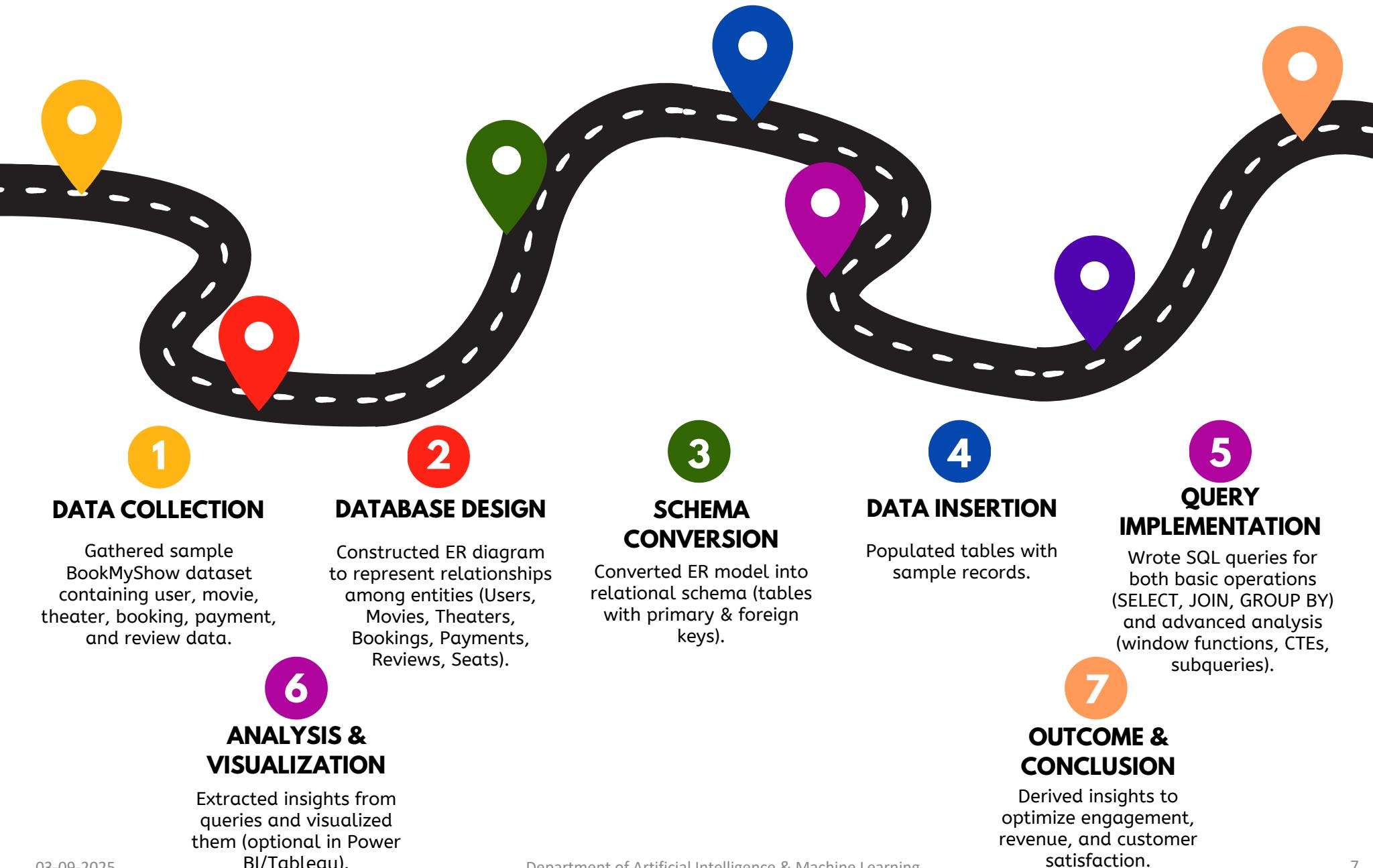
Addressing these challenges requires SQL-based analytical insights to uncover trends, highlight inefficiencies, and provide data-driven strategies. The ultimate goal is to enhance platform performance, boost revenue, and deliver a seamless user experience.



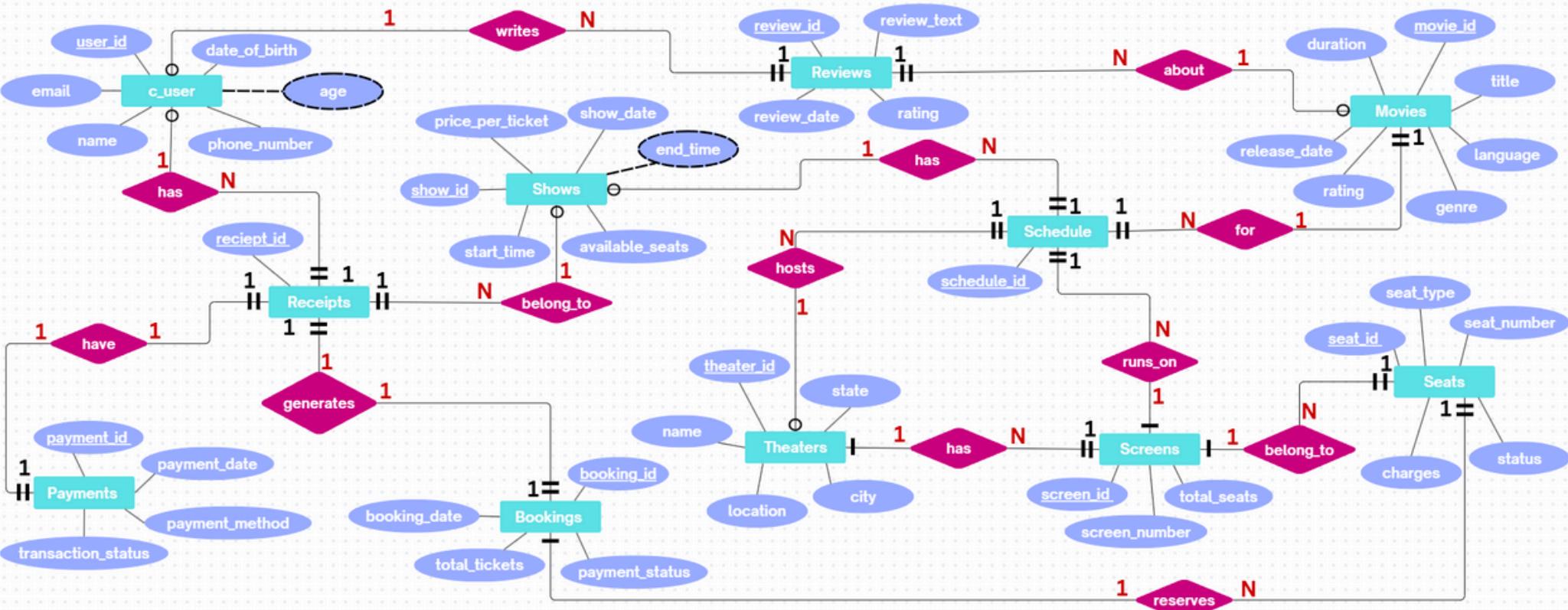
Objectives

1. Analyze user engagement to identify most active users and improve retention.
2. Evaluate movie and theater performance for better scheduling and marketing.
3. Track revenue generation across months, users, and payment methods.
4. Study seat utilization to align pricing strategy with demand.
5. Assess payment system reliability by identifying transaction failures.
6. Examine customer reviews and ratings to enhance content offerings.

Methodology



Entity Relationship Diagram



Relational Scheme

Payment
<u>payment_id</u> payment_method payment_date transaction_status

Theaters
<u>theater_id</u> name location city state

Reviews
<u>review_id</u> user_id movie_id rating review_text review_date

Bookings
<u>booking_id</u> booking_date total_tickets payment_status

shows
<u>show_id</u> show_date start_time price_per_ticket available_seats

c_user
<u>user_id</u> c_name email phone_number DOB

screens
<u>screen_id</u> screen_number total_Seats

Seats
<u>seat_id</u> booking_id screen_id seat_number seat_type charger status

Schedule
<u>schedule_id</u> show_id movie_id theater_id screen_id

Movies
<u>movie_id</u> title genre m_lang duration rating release_date

Receipts
<u>Receipt_id</u> Booking_id user_id show_id payment_id

Creation

- **USER_ID**

```
CREATE TABLE c_user (user_id VARCHAR(20) PRIMARY KEY, name VARCHAR(100), email VARCHAR(100) UNIQUE, phone_number  
BIGINT, date_of_birth DATE);
```

- **Movies**

```
CREATE TABLE Movies (movie_id VARCHAR(20) PRIMARY KEY, title VARCHAR(200), genre VARCHAR(50), language VARCHAR(50),  
duration INT, rating  
DECIMAL(3,1), release_date DATE);
```

- **Theater_id**

```
CREATE TABLE Theaters (theater_id VARCHAR(20) PRIMARY KEY, name VARCHAR(100), location VARCHAR(100), city VARCHAR(100),  
state VARCHAR(100));
```

- **Screens**

```
CREATE TABLE Screens (screen_id VARCHAR(20) PRIMARY KEY, screen_number INT, total_seats INT);
```

- **Shows**

```
CREATE TABLE Shows (show_id VARCHAR(20) PRIMARY KEY, show_date DATE, start_time TIME, price_per_ticket DECIMAL(8,2),  
available_seats INT);
```

- **Schedule**

```
CREATE TABLE Schedule (schedule_id VARCHAR(20) PRIMARY KEY, show_id VARCHAR(20), movie_id VARCHAR(20), theater_id  
VARCHAR(20), screen_id  
VARCHAR(20), FOREIGN KEY (show_id) REFERENCES Shows(show_id),  
FOREIGN KEY (movie_id) REFERENCES Movies(movie_id), FOREIGN KEY (theater_id) REFERENCES Theaters(theater_id), FOREIGN KEY  
(screen_id) REFERENCES Screens(screen_id));
```

Creation

- **Bookings**

```
CREATE TABLE Bookings (booking_id VARCHAR(20) PRIMARY KEY, booking_date DATETIME, total_tickets INT, payment_status  
VARCHAR(50));
```

- **Seats**

```
CREATE TABLE Seats (seat_id VARCHAR(20) PRIMARY KEY, booking_id VARCHAR(20), screen_id VARCHAR(20), seat_number  
VARCHAR(10), seat_type VARCHAR(20), charges DECIMAL(8,2), status VARCHAR(20), FOREIGN KEY (booking_id) REFERENCES  
Bookings(booking_id), FOREIGN KEY (screen_id) REFERENCES Screens(screen_id));
```

- **Payments**

```
CREATE TABLE Payments (payment_id VARCHAR(20) PRIMARY KEY, payment_method VARCHAR(50), payment_date DATETIME,  
transaction_status VARCHAR(50));
```

- **Receipts**

```
CREATE TABLE Receipts (receipt_id VARCHAR(20) PRIMARY KEY, booking_id VARCHAR(20), user_id VARCHAR(20), show_id  
VARCHAR(20), payment_id VARCHAR(20), FOREIGN KEY (booking_id) REFERENCES Bookings(booking_id), FOREIGN KEY (user_id)  
REFERENCES c_user(user_id), FOREIGN KEY (show_id) REFERENCES Shows(show_id), FOREIGN KEY (payment_id) REFERENCES  
Payments(payment_id));
```

- **Reviews**

```
CREATE TABLE Reviews (review_id VARCHAR(20) PRIMARY KEY, user_id VARCHAR(20), movie_id VARCHAR(20), rating DECIMAL(3,1),  
review_text TEXT, review_date DATETIME, FOREIGN KEY (user_id) REFERENCES c_user(user_id), FOREIGN KEY (movie_id) REFERENCES  
Movies(movie_id));
```

Insertion

- **Bookings**

```
mysql> show tables;
+-----+
| Tables_in_bookmyshow |
+-----+
| bookings
| c_user
| movies
| payments
| receipts
| reviews
| schedule
| screens
| seats
| shows
| theaters
+-----+
11 rows in set (0.00 sec)
```

booking_id	booking_date	total_tickets	payment_status
BK_1	2025-07-26 19:40:18	2	Completed
BK_10	2025-12-28 13:57:17	1	Failed
BK_100	2025-12-20 15:05:03	1	Completed
BK_1000	2025-01-08 02:54:08	4	Pending
BK_10000	2024-03-05 13:33:43	1	Pending
BK_1001	2025-02-04 14:33:39	3	Completed
BK_1002	2025-03-30 23:33:12	2	Completed
BK_1003	2024-02-09 14:03:16	1	Completed
BK_1004	2025-07-06 16:57:11	4	Failed
BK_1005	2025-07-23 23:01:57	2	Completed
BK_1006	2024-05-19 16:12:49	3	Completed
BK_1007	2024-04-16 00:28:46	1	Completed
BK_1008	2025-11-29 11:18:23	3	Pending
BK_1009	2024-10-22 06:54:25	2	Completed
BK_101	2024-09-15 21:10:12	1	Completed

15 rows in set (0.00 sec)

Insertion

- **c_user**

```
mysql> select * from c_user limit 15;
+-----+-----+-----+-----+-----+
| user_id | name           | email            | phone_number | date_of_birth |
+-----+-----+-----+-----+-----+
| US_1    | Allison Hill   | AllisonHill@gmail.com | 8143513964  | 1978-10-06   |
| US_10   | Ryan Munoz     | RyanMunoz@gmail.com | 7718714954  | 1990-05-10   |
| US_100  | Stephen McKee   | StephenMcKee@gmail.com | 7684458507  | 2000-11-02   |
| US_1000 | Elizabeth Sanders | ElizabethSanders@gmail.com | 7500457480  | 2002-04-10   |
| US_10000 | David Hayes     | DavidHayes@gmail.com | 9925098696  | 1988-08-22   |
| US_1001 | Matthew Davis MD | MatthewDavis@gmail.com | 8738108435  | 1967-03-07   |
| US_1002 | Stephen Wood     | StephenWood@gmail.com | 9191689085  | 1981-04-09   |
| US_1003 | Dennis Marshall   | DennisMarshall@gmail.com | 8643247871  | 2005-06-18   |
| US_1004 | Derrick Brown     | DerrickBrown@gmail.com | 7960132258  | 1959-07-31   |
| US_1005 | Mary Miller       | MaryMiller@gmail.com | 9060839308  | 1997-07-03   |
| US_1006 | Joshua Arellano   | JoshuaArellano@gmail.com | 7233952833  | 1977-12-22   |
| US_1007 | Kristen Randall   | KristenRandall@gmail.com | 9809275822  | 1993-01-21   |
| US_1008 | Jennifer Nichols  | JenniferNichols@gmail.com | 9003282487  | 1971-08-29   |
| US_1009 | Karen Keith       | KarenKeith@gmail.com | 9604640487  | 2004-08-13   |
| US_101  | Sandra Aguilar    | SandraAguilar@gmail.com | 8070481556  | 1995-12-30   |
+-----+-----+-----+-----+-----+
15 rows in set (0.00 sec)
```

Insertion

- movies

movie_id	title	genre	language	duration	rating	release_date
MV_1	Address clear	Sci-Fi	Kannada	118	6.6	2022-03-05
MV_10	After there quickly matter	Romance	Hindi	130	4.3	2013-08-27
MV_100	Increase toward tell	Drama	Hindi	168	8.2	2020-02-17
MV_1000	Stay knowledge day	Sci-Fi	Tamil	139	8.8	2017-12-14
MV_10000	However sort billion	Horror	Kannada	176	5.6	2018-11-09
MV_1001	Because rise forget	Drama	Malayalam	133	3.7	2016-12-01
MV_1002	Develop reason	Sci-Fi	Kannada	98	5.6	2002-04-02
MV_1003	Term read	Comedy	Malayalam	147	8.2	2005-03-04
MV_1004	Call successful them	Comedy	English	117	3.1	2023-01-19
MV_1005	Address clear	Sci-Fi	Kannada	110	3.7	2020-11-30
MV_1006	Lot we pass	Romance	Hindi	155	8.9	2014-05-13
MV_1007	Degree voice challenge	Romance	Malayalam	104	8.3	2023-03-20
MV_1008	Nearly population give class	Action	Kannada	144	7.7	2001-08-16
MV_1009	Exactly turn traditional	Comedy	Hindi	108	5.6	2012-12-08
MV_101	Quite spring enter	Horror	English	177	5.7	2001-06-08

15 rows in set (0.01 sec)

Insertion

- payment

payment_id	payment_method	payment_date	transaction_status
PMT_1	Credit Card	2025-07-26 19:40:18	Success
PMT_10	Wallet	2025-12-28 13:57:17	Success
PMT_100	Credit Card	2025-12-20 15:05:03	Success
PMT_1000	Debit Card	2025-01-08 02:54:08	Success
PMT_10000	Credit Card	2024-03-05 13:33:43	Refunded
PMT_1001	Wallet	2025-02-04 14:33:39	Success
PMT_1002	Net Banking	2025-03-30 23:33:12	Success
PMT_1003	Credit Card	2024-02-09 14:03:16	Success
PMT_1004	Wallet	2025-07-06 16:57:11	Success
PMT_1005	UPI	2025-07-23 23:01:57	Refunded
PMT_1006	Credit Card	2024-05-19 16:12:49	Success
PMT_1007	Net Banking	2024-04-16 00:28:46	Refunded
PMT_1008	Credit Card	2025-11-29 11:18:23	Success
PMT_1009	UPI	2024-10-22 06:54:25	Success
PMT_101	UPI	2024-09-15 21:10:12	Success

15 rows in set (0.00 sec)

Insertion

- review

review_id	user_id	movie_id	rating	review_text	review_date
REV_1	US_8392	MV_8805	1.2	Could be better.	2025-07-01 16:16:00
REV_10	US_2201	MV_861	9	Great movie!	2025-05-21 04:23:30
REV_100	US_546	MV_492	8.1	Could be better.	2024-05-10 11:22:00
REV_1000	US_850	MV_5745	5.4	Great movie!	2025-03-03 00:52:31
REV_10000	US_1709	MV_3579	8.8	Great movie!	2025-05-27 13:25:25
REV_1001	US_1166	MV_9656	6.8	Could be better.	2025-04-16 03:13:49
REV_1002	US_1550	MV_4545	9.3	Great movie!	2024-05-23 17:00:36
REV_1003	US_2938	MV_6564	6.6	Could be better.	2024-01-02 15:24:21
REV_1004	US_6253	MV_1479	9	Great movie!	2024-07-24 03:09:14
REV_1005	US_7082	MV_1683	9.4	Great movie!	2025-10-15 23:38:25
REV_1006	US_7844	MV_1179	9.4	Great movie!	2025-01-15 17:05:16
REV_1007	US_7589	MV_9780	4.7	Great movie!	2025-08-25 04:39:34
REV_1008	US_3214	MV_3214	1.9	Great movie!	2024-07-30 14:58:31
REV_1009	US_189	MV_7186	1.7	Could be better.	2024-03-15 02:53:18
REV_101	US_8871	MV_5948	1.7	Great movie!	2025-02-09 18:26:34

Insertion

- screen

screen_id	screen_number	total_seats
SC_1	2	266
SC_10	2	161
SC_100	8	116
SC_1000	3	228
SC_101	9	288
SC_102	3	163
SC_103	8	276
SC_104	4	300
SC_105	10	268
SC_106	2	201
SC_107	4	289
SC_108	7	291
SC_109	9	150
SC_11	3	154
SC_110	5	135

15 rows in set (0.00 sec)

Insertion

- **seat**

seat_id	booking_id	screen_id	seat_number	seat_type	charges	status
ST_1	BK_9384	SC_666	B18	VIP	50	Available
ST_10	BK_6810	SC_458	B9	Regular	10	Booked
ST_100	BK_5783	SC_392	C3	Regular	10	Booked
ST_1000	BK_8683	SC_588	C4	Premium	25	Available
ST_10000	BK_6864	SC_31	E15	Regular	10	Booked
ST_1001	BK_937	SC_11	E3	VIP	50	Booked
ST_1002	BK_9185	SC_776	A14	Premium	25	Booked
ST_1003	BK_335	SC_852	B11	Regular	10	Booked
ST_1004	BK_1666	SC_401	C9	Premium	25	Booked
ST_1005	BK_870	SC_80	E19	Premium	25	Booked
ST_1006	BK_4749	SC_781	C8	Premium	25	Available
ST_1007	BK_4794	SC_673	A17	Regular	10	Available
ST_1008	BK_3317	SC_302	C13	VIP	50	Booked
ST_1009	BK_6423	SC_405	C15	Premium	25	Booked
ST_101	BK_3945	SC_274	B5	Regular	10	Available

15 rows in set (0.00 sec)

Insertion

- show

show_id	show_date	start_time	price_per_ticket	available_seats
SH_1	2024-04-06	22:30:00	174.34	139
SH_10	2024-04-06	19:45:00	304.13	197
SH_100	2025-03-12	13:30:00	393.44	157
SH_1000	2025-01-25	15:30:00	413.34	130
SH_10000	2023-01-29	11:45:00	172.58	172
SH_1001	2023-12-31	20:00:00	488.08	140
SH_1002	2025-06-04	14:45:00	455.49	65
SH_1003	2025-07-10	16:00:00	302.75	92
SH_1004	2024-01-30	17:45:00	338.78	148
SH_1005	2024-08-02	19:00:00	163.77	133
SH_1006	2024-03-11	11:00:00	446.69	61
SH_1007	2023-11-05	18:15:00	170.98	96
SH_1008	2024-11-29	22:45:00	205.82	64
SH_1009	2023-12-01	20:00:00	219.59	60
SH_101	2023-05-04	13:15:00	226.24	185

15 rows in set (0.00 sec)

Insertion

- theater

theater_id	name	location	city	state
TH_1	PVR Cinemas	Mangalore 101	Mangalore	Karnataka
TH_10	Carnival Cinemas	Jalandhar 301	Jalandhar	Punjab
TH_100	INOX	Kota 18	Kota	Rajasthan
TH_1000	Miraj Cinemas	Tiruchirappalli 414	Tiruchirappalli	Tamil Nadu
TH_10000	Miraj Cinemas	Tiruchirappalli 342	Tiruchirappalli	Tamil Nadu
TH_1001	PVR Cinemas	Jalandhar 280	Jalandhar	Punjab
TH_1002	Carnival Cinemas	Durgapur 14	Durgapur	West Bengal
TH_1003	Cinepolis	Varanasi 94	Varanasi	Uttar Pradesh
TH_1004	Wave Cinemas	Ajmer 462	Ajmer	Rajasthan
TH_1005	Cinepolis	Madurai 183	Madurai	Tamil Nadu
TH_1006	INOX	Siliguri 147	Siliguri	West Bengal
TH_1007	Cinepolis	Bathinda 192	Bathinda	Punjab
TH_1008	Miraj Cinemas	Kota 213	Kota	Rajasthan
TH_1009	Wave Cinemas	Agra 23	Agra	Uttar Pradesh
TH_101	Wave Cinemas	Bhavnagar 329	Bhavnagar	Gujarat

15 rows in set (0.00 sec)

Insertion

- Schedule

Schedule_id	show_id	movie_id	theater_id	screen_id
S_1	SH_1	MV_2044	TH_489	SC_148
S_10	SH_10	MV_859	TH_452	SC_699
S_100	SH_100	MV_4275	TH_179	SC_796
S_1000	SH_1000	MV_4094	TH_87	SC_971
S_10000	SH_10000	MV_7413	TH_211	SC_97
S_1001	SH_1001	MV_6543	TH_414	SC_904
S_1002	SH_1002	MV_65	TH_405	SC_191
S_1003	SH_1003	MV_8720	TH_449	SC_68
S_1004	SH_1004	MV_137	TH_498	SC_788
S_1005	SH_1005	MV_3468	TH_252	SC_108
S_1006	SH_1006	MV_7287	TH_441	SC_135
S_1007	SH_1007	MV_6975	TH_229	SC_339
S_1008	SH_1008	MV_115	TH_136	SC_315
S_1009	SH_1009	MV_9203	TH_208	SC_280
S_101	SH_101	MV_1277	TH_486	SC_137

15 rows in set (0.00 sec)

Insertion

- Receipt

Receipt_id	booking_id	user_id	show_id	payment_id
R_1	BK_1	US_543	SH_6385	PMT_1
R_10	BK_10	US_2851	SH_2314	PMT_10
R_100	BK_100	US_8701	SH_3176	PMT_100
R_1000	BK_1000	US_781	SH_6305	PMT_1000
R_10000	BK_10000	US_5835	SH_1559	PMT_10000
R_1001	BK_1001	US_617	SH_6568	PMT_1001
R_1002	BK_1002	US_9076	SH_2251	PMT_1002
R_1003	BK_1003	US_7708	SH_1683	PMT_1003
R_1004	BK_1004	US_5917	SH_9277	PMT_1004
R_1005	BK_1005	US_3549	SH_3126	PMT_1005
R_1006	BK_1006	US_449	SH_7934	PMT_1006
R_1007	BK_1007	US_8344	SH_4141	PMT_1007
R_1008	BK_1008	US_6053	SH_8958	PMT_1008
R_1009	BK_1009	US_7997	SH_7769	PMT_1009
R_101	BK_101	US_7789	SH_1179	PMT_101

15 rows in set (0.00 sec)

Sub Queries

Find movies that have average rating greater than overall average rating

```
mysql> SELECT title
-> FROM Movies
-> WHERE movie_id IN (
->   SELECT movie_id
->   FROM Reviews
->   GROUP BY movie_id
->   HAVING AVG(rating) > (SELECT AVG(rating) FROM Reviews)
-> )
-> LIMIT 15;
+-----+
| title |
+-----+
| Increase toward tell
| Stay knowledge day
| Term read
| Call successful them
| Address clear
| Nearly population give class
| Exactly turn traditional
| Quite spring enter
| Enter newspaper
| Stock nation
| Thank least look ago
| Look laugh suggest admit
| Himself
| Form base
| Yes add
+-----+
15 rows in set (0.05 sec)
```

List users who have made more than 1 booking

```
mysql> SELECT c.name
-> FROM c_user c
-> WHERE c.user_id IN (
->   SELECT r.user_id
->   FROM Reciepts r
->   GROUP BY r.user_id
->   HAVING COUNT(r.booking_id) > 1
-> )
-> LIMIT 15;
+-----+
| name |
+-----+
| David Hayes
| Matthew Davis MD
| Mary Miller
| Kristen Randall
| Jennifer Nichols
| Blake Orr
| Megan Oliver
| Peter Thompson
| Lori Flowers MD
| Kristine Garcia
| Joseph Cooper
| Steve Rivera
| Rebecca Valencia
| Debra Butler
| Chelsea Lewis
+-----+
15 rows in set (0.03 sec)
```

Join Queries

List all bookings with the user's name and show date

```
mysql> SELECT b.booking_id, c.name, s.show_date
-> FROM Bookings b
-> JOIN Receipts r ON b.booking_id = r.booking_id
-> JOIN c_user c ON r.user_id = c.user_id
-> JOIN Shows s ON r.show_id = s.show_id
-> LIMIT 15;
+-----+-----+-----+
| booking_id | name      | show_date |
+-----+-----+-----+
| BK_1        | Kenneth Edwards | 2023-04-06 |
| BK_10       | David Williams | 2024-03-30 |
| BK_100      | Suzanne Luna   | 2025-09-04 |
| BK_1000     | Christine Goodwin | 2023-01-15 |
| BK_10000    | Maria Sullivan | 2023-06-03 |
| BK_1001     | Bryan Parker   | 2023-03-31 |
| BK_1002     | Carolyn Smith  | 2023-01-25 |
| BK_1003     | Kimberly Adams | 2024-08-17 |
| BK_1004     | Diana Mullins | 2024-01-22 |
| BK_1005     | Amy Walsh      | 2024-06-07 |
| BK_1006     | Ryan Rosales   | 2024-10-07 |
| BK_1007     | Thomas Olson   | 2024-07-10 |
| BK_1008     | Sophia Richmond | 2025-03-17 |
| BK_1009     | Kathryn Perry  | 2024-05-17 |
| BK_101      | Michelle Schultz | 2025-07-22 |
+-----+-----+-----+
15 rows in set (0.00 sec)
```

Get all movies and the theaters they are scheduled in

```
mysql> SELECT m.title, t.name AS theater_name, sc.schedule_id
-> FROM Schedule sc
-> JOIN Movies m ON sc.movie_id = m.movie_id
-> JOIN Theaters t ON sc.theater_id = t.theater_id
-> LIMIT 15;
+-----+-----+-----+
| title          | theater_name | schedule_id |
+-----+-----+-----+
| Suggest want something | Wave Cinemas | S_1          |
| Popular business | Cinepolis    | S_10         |
| Certain road    | Wave Cinemas | S_100        |
| Language house  | Wave Cinemas | S_1000       |
| Piece factor score make | Carnival Cinemas | S_10000    |
| No night hand   | INOX         | S_1001       |
| Challenge know impact media | Wave Cinemas | S_1002       |
| Above someone something | Wave Cinemas | S_1003       |
| Alone full      | Carnival Cinemas | S_1004    |
| Career off      | Cinepolis    | S_1005       |
| End two         | INOX         | S_1006       |
| Choose carry teach | Cinepolis    | S_1007       |
| See recently nation | Cinepolis    | S_1008       |
| Season attack forward | Carnival Cinemas | S_1009    |
| Choose carry teach | Cinepolis    | S_101        |
+-----+-----+-----+
15 rows in set (0.00 sec)
```

Group by, Having by Queries

Count number of bookings per user, order by most bookings

```
15 rows in set (0.03 sec)

mysql> SELECT r.user_id, COUNT(b.booking_id) AS total_bookings
-> FROM Receipts r
-> JOIN Bookings b ON r.booking_id = b.booking_id
-> GROUP BY r.user_id
-> ORDER BY total_bookings DESC
-> LIMIT 15;
+-----+-----+
| user_id | total_bookings |
+-----+-----+
| US_5879 | 7 |
| US_7940 | 6 |
| US_4772 | 6 |
| US_4280 | 6 |
| US_287 | 6 |
| US_6605 | 6 |
| US_3368 | 6 |
| US_608 | 6 |
| US_16 | 6 |
| US_810 | 6 |
| US_4958 | 6 |
| US_9644 | 5 |
| US_2297 | 5 |
| US_8911 | 5 |
| US_9076 | 5 |
+-----+-----+
15 rows in set (0.03 sec)
```

Get average rating for each movie, order by highest first

```
mysql> SELECT m.title, AVG(rv.rating) AS avg_rating
-> FROM Reviews rv
-> JOIN Movies m ON rv.movie_id = m.movie_id
-> GROUP BY m.title
-> ORDER BY avg_rating DESC
-> LIMIT 15;
+-----+-----+
| title | avg_rating |
+-----+-----+
| Good brother throw prove | 7.17500010728836 |
| Research send consider | 6.900000061307635 |
| Involve fast treatment | 6.78666660785675 |
| After mention | 6.675000071525574 |
| Despite hot collection | 6.673913017563198 |
| Talk per whether also | 6.651999926567077 |
| Father scene | 6.5708333651224775 |
| Defense investment | 6.52222216129303 |
| Front service red | 6.511764708687277 |
| Project civil | 6.509375032037497 |
| Article whatever company | 6.445000052452087 |
| Four increase my | 6.357142870766776 |
| Green item | 6.329999963442485 |
| Choice watch | 6.295833284656207 |
| Not hear experience | 6.285714223271325 |
+-----+-----+
15 rows in set (0.09 sec)
```

Aggregate Queries

(Find total seats booked per booking)

```
mysql> SELECT r.user_id, COUNT(b.booking_id) AS total_bookings
-> FROM Receipts r
-> JOIN Bookings b ON r.booking_id = b.booking_id
-> GROUP BY r.user_id
-> ORDER BY total_bookings DESC
-> LIMIT 15;
+-----+-----+
| user_id | total_bookings |
+-----+-----+
| US_5879 | 7 |
| US_7940 | 6 |
| US_4772 | 6 |
| US_4280 | 6 |
| US_287 | 6 |
| US_6605 | 6 |
| US_3368 | 6 |
| US_608 | 6 |
| US_16 | 6 |
| US_810 | 6 |
| US_4958 | 6 |
| US_9644 | 5 |
| US_2297 | 5 |
| US_8911 | 5 |
| US_9076 | 5 |
+-----+-----+
15 rows in set (0.04 sec)
```

Calculate total revenue collected (sum of charges)

```
mysql> SELECT SUM(s.charges) AS total_revenue
-> FROM Seats s;
+-----+
| total_revenue |
+-----+
| 185295 |
+-----+
1 row in set (0.01 sec)
```

Master Queries

Find theaters where the average ticket price is above the overall average ticket price

```
mysql> SELECT t.name AS theater_name, AVG(s.price_per_ticket) AS avg_ticket_price
-> FROM Schedule sc
-> JOIN Shows s ON sc.show_id = s.show_id
-> JOIN Theaters t ON sc.theater_id = t.theater_id
-> GROUP BY t.theater_id, t.name
-> HAVING AVG(s.price_per_ticket) > (
->     SELECT AVG(price_per_ticket) FROM Shows
-> )
-> ORDER BY avg_ticket_price DESC
-> LIMIT 15;
+-----+-----+
| theater_name | avg_ticket_price |
+-----+-----+
| Miraj Cinemas | 400.16615295410156 |
| Wave Cinemas | 384.2200006338266 |
| PVR Cinemas | 384.0664708754596 |
| Carnival Cinemas | 381.2744995117188 |
| Miraj Cinemas | 381.21849899291993 |
| Carnival Cinemas | 380.95947426243833 |
| Cinepolis | 379.8586669921875 |
| Wave Cinemas | 379.04625129699707 |
| INOX | 379.0350003923689 |
| Miraj Cinemas | 377.68850173950193 |
| Miraj Cinemas | 376.99700088500975 |
| INOX | 375.8838899400499 |
| Cinepolis | 373.0599989457564 |
| Miraj Cinemas | 372.9327748616536 |
| INOX | 372.8472720059481 |
+-----+-----+
15 rows in set (0.14 sec)
```

Get users who booked more than 1 ticket total, and show their total spend (using subquery for charges)

```
mysql> SELECT *
-> FROM (
->     SELECT c.name, SUM(s.charges) AS total_spent, COUNT(b.booking_id) AS total_bookings
->     FROM c_user c
->     JOIN Receipts r ON c.user_id = r.user_id
->     JOIN Bookings b ON r.booking_id = b.booking_id
->     JOIN Seats s ON b.booking_id = s.booking_id
->     GROUP BY c.user_id, c.name
->     HAVING COUNT(b.booking_id) > 1
->     AND SUM(s.charges) > (
->         SELECT AVG(total_cost)
->         FROM (
->             SELECT SUM(charges) AS total_cost
->             FROM Seats
->             GROUP BY booking_id
->         ) AS booking_totals
->     )
-> ) AS temp
-> ORDER BY total_spent DESC
-> LIMIT 15;
+-----+-----+-----+
| name | total_spent | total_bookings |
+-----+-----+-----+
| Jason Thomas | 230 | 10 |
| Jaime O'Neill | 205 | 8 |
| Madison Stanton | 200 | 10 |
| Jacqueline Herman | 190 | 9 |
| Donald Ford | 185 | 9 |
| Kenneth Smith | 185 | 6 |
| Sara Hamilton | 185 | 6 |
| Kyle Johnson | 180 | 7 |
| Cesar Wilson | 180 | 6 |
| John Pierce | 180 | 8 |
| Jeremy Turner | 180 | 12 |
| Erin McKinney | 180 | 9 |
| Julie Wilson | 180 | 9 |
| Brian Ramirez | 175 | 9 |
| Stephanie Harris | 175 | 4 |
+-----+-----+-----+
15 rows in set (0.35 sec)
```

Project Outcome

- Identified most active users and repeat customers for loyalty programs.
- Discovered top-performing movies, languages, and genres for targeted marketing.
- Recognized theaters with low occupancy for better scheduling and pricing optimization.
- Improved payment efficiency by analyzing failures and refunds.
- Leveraged user reviews to understand content preferences and satisfaction.

Conclusion

The project successfully demonstrated how SQL-based analysis can transform raw transactional data into actionable insights for BookMyShow. By focusing on user behavior, theater efficiency, revenue optimization, and payment system reliability, the platform can enhance overall customer experience and improve profitability.

References

- BookMyShow official website –
<https://www.bookmyshow.com>
- Kaggle - <https://www.kaggle.com/soumenduray99>
- MySQL Documentation – <https://dev.mysql.com/doc>
- Dataset: Synthetic dataset created for project purposes (not official BookMyShow data)

Presentation



**THANK
YOU
FOR LISTENING TO
OUR PRESENTATION**