Electronics and Communication Engineering Department

Semester VII

(Academic Year 2023-24)

2CS0E54: Database Management Systems

Special Assignment Report

Submitted by:

Sanchit Sharma (20BEC108) | Sagar Sorathiya (20BEC122)

Submitted to: Prof. Ashwin Verma

Online Movie Booking System

Introduction:

The Online Movie Ticket Booking System is a cutting-edge platform that revolutionizes the way customers interact with the world of cinema. It offers a user-friendly and accessible solution for booking cinema tickets and accessing vital information about movies and their schedules from the comfort of one's home. This project stands as a pivotal tool for administrators, offering them the capability to efficiently manage and update movie descriptions, schedules, and other related data, which seamlessly reflects on the webpages accessible to customers.

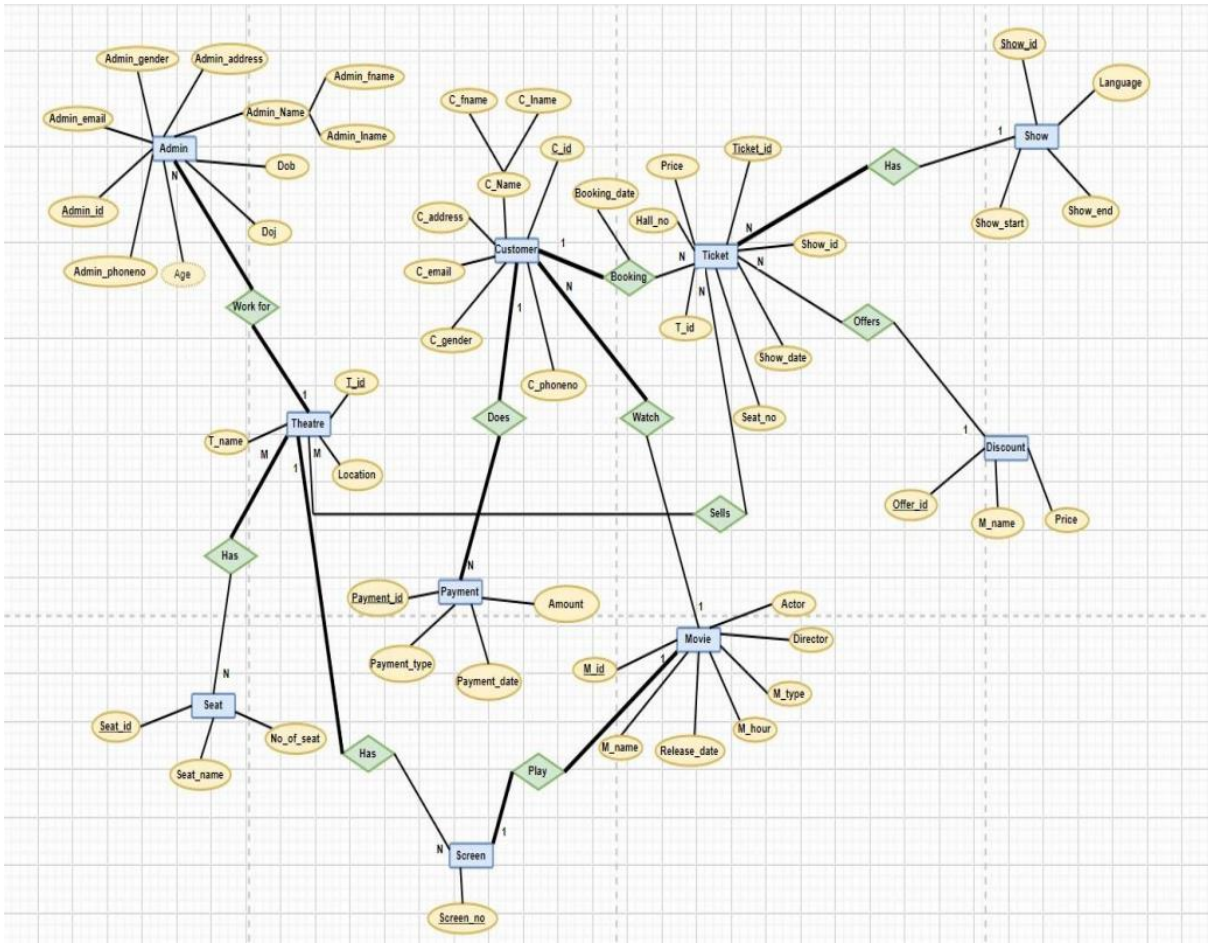
This innovative system provides a new avenue for customers to purchase cinema tickets, eliminating the need for manual ticketing processes and reducing the burden on patrons. By simplifying and automating the ticket booking process, this system not only enhances the customer experience but also offers a more efficient and streamlined approach to securing movie tickets.

At its core, the Online Movie Ticket Booking System is a customer-centric platform. It aims to provide customers with comprehensive and up-to-date information about movies and their screening schedules. By offering a wealth of movie-related data, including descriptions, genres, schedules, and available seats, this system empowers customers to make well-informed decisions when reserving their tickets. The ultimate goal is to offer an all-encompassing service that enhances the movie-going experience and simplifies the ticket booking process, all within the convenience of an online platform.

Functional Requirements:

- Ticket booking portal should be able to list the cinemas located in your city.
- Once customer select the city the movie and its showtimes are visible.
- Customer should log in before booking the ticket.
- The customer shall be shown a 2D image of the seats from which the desired seats are selected.
- For customer there are multiple option for payment and their variety of offers.
- The system has featured that admin can be able to add movie and their details.
- The system should have feature that admin can delete the movie and their detail.
- After booking, the system can generate the portable document file (.pdf) and then sent one copy to the customer’s Email-address and booking confirmation SMS to customer's phone.
- Waiting customers should be serviced in a fair, first come, first serve manner.

ER Diagram:



Relational Model:

- admin (admin_id, admin_fname, admin_lname, admin_gender, admin_email, admin_phoneno, admin_address, dob, doj, t_id)
- theatre (t_id, t_name, location)
- seat (seat_id, seat_name, no_of_seat)
- theatre_has_seat (t_id, seat_id)
- screen (screen_no, t_id, m_id)
- movie (m_id, m_name, m_type, m_hour, release_date, director,actor)
- customer (c_id, c_fname, c_lname, c_gender, c_email, c_phoneno, c_address, m_id)
- payment (payment_id, payment_type, payment_date, amount, c_id)
- ticket (ticket_id, show_date, seat_no, t_id, hall_no, price, c_id, offer_id, show_id, booking_date)
- sells (ticket_id, t_id)
- show (show_id, language, show_start, show_end)
- discount (offer_id, m_name, price)

Creating Database:

```
mysql> CREATE DATABASE movie_booking_dbms;
Query OK, 1 row affected (0.00 sec)
```

Admin Table:

Constraints:

- Primary key – admin_id
- Not Null – admin_fname, admin_lname, admin_gender
- Check constraint - admin_gender has a check constraint that ensures that the value is either ‘M’ or ‘F’. admin_email has a check constraint that ensure that the email should end with ‘@bookmymovie.com’.
- Foreign key – admin.t_id references theatre.t_id

```
mysql> CREATE TABLE admin (
    ->     admin_id VARCHAR(6) PRIMARY KEY CHECK(admin_id LIKE 'A%'),
    ->     admin_fname VARCHAR(20) NOT NULL,
    ->     admin_lname VARCHAR(20) NOT NULL,
    ->     admin_gender CHAR(1) NOT NULL CHECK(admin_gender IN ('M', 'F')),
    ->     admin_email VARCHAR(30) CHECK(admin_email LIKE '%@bookmymovie.com'),
    ->     admin_phoneno BIGINT(10),
    ->     admin_address VARCHAR(30),
    ->     dob DATE,
    ->     doj DATE,
    ->     t_id VARCHAR(6)
    -> );
Query OK, 0 rows affected, 1 warning (0.01 sec)
```

```
mysql> ALTER TABLE admin
    -> ADD CONSTRAINT fk_admin_theatre
    -> FOREIGN KEY (t_id) REFERENCES theatre(t_id);
Query OK, 5 rows affected (0.04 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

```
mysql> INSERT INTO admin (admin_id, admin_fname, admin_lname, admin_gender, admin_email, admin_phoneno, admin_address, dob, doj, t_id)
    -> VALUES
    -> ('A01', 'Yash', 'Mehta', 'M', 'yash@bookmymovie.com', 6747986463, 'Rajkot', STR_TO_DATE('12-03-1989', '%d-%m-%Y'), STR_TO_DATE('14-04-2012', '%d-%m-%Y'), 'T01'),
    -> ('A02', 'Manav', 'Joshi', 'M', 'manav1@bookmymovie.com', 4368903463, 'Surat', STR_TO_DATE('17-06-1979', '%d-%m-%Y'), STR_TO_DATE('23-09-2011', '%d-%m-%Y'), 'T01'),
    -> ('A03', 'Smit', 'Jariwala', 'M', 'smitjari@bookmymovie.com', 8256024896, 'Pune', STR_TO_DATE('01-12-1980', '%d-%m-%Y'), STR_TO_DATE('29-05-2005', '%d-%m-%Y'), 'T04'),
    -> ('A04', 'Disha', 'Patel', 'F', 'disha@bookmymovie.com', 9175366396, 'Delhi', STR_TO_DATE('24-07-1990', '%d-%m-%Y'), STR_TO_DATE('20-08-2015', '%d-%m-%Y'), 'T02'),
    -> ('A05', 'Zeel', 'Desai', 'F', 'zeel@bookmymovie.com', 8645789903, 'Mumbai', STR_TO_DATE('18-09-1970', '%d-%m-%Y'), STR_TO_DATE('11-04-2013', '%d-%m-%Y'), 'T03');
Query OK, 5 rows affected (0.03 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM admin;
+----------+-------------+-------------+--------------+-------------------------+---------------+---------------+------------+------------+------+
| admin_id | admin_fname | admin_lname | admin_gender | admin_email             | admin_phoneno | admin_address | dob        | doj        | t_id |
+----------+-------------+-------------+--------------+-------------------------+---------------+---------------+------------+------------+------+
| A01      | Yash        | Mehta       | M            | yash@bookmymovie.com    |    6747986463 | Rajkot        | 1989-03-12 | 2012-04-14 | T01  |
| A02      | Manav       | Joshi       | M            | manav1@bookmymovie.com  |    4368903463 | Surat         | 1979-06-17 | 2011-09-23 | T01  |
| A03      | Smit        | Jariwala    | M            | smitjari@bookmymovie.com|    8256024896 | Pune          | 1980-12-01 | 2005-05-29 | T04  |
| A04      | Disha       | Patel       | F            | disha@bookmymovie.com   |    9175366396 | Delhi         | 1990-07-24 | 2015-08-20 | T02  |
| A05      | Zeel        | Desai       | F            | zeel@bookmymovie.com    |    8645789903 | Mumbai        | 1970-09-18 | 2013-04-11 | T03  |
+----------+-------------+-------------+--------------+-------------------------+---------------+---------------+------------+------------+------+
5 rows in set (0.00 sec)
```

Theatre Table:

Constraints:

- Primary key - t_id
- Check constraint – t_id should start with ‘T’
- Not Null – t_name, location

```
mysql> CREATE TABLE theatre(
    ->     t_id VARCHAR(6) PRIMARY KEY CHECK(t_id LIKE 'T%'),
    ->     t_name VARCHAR(10) NOT NULL,
    ->     location VARCHAR(12) NOT NULL
    -> );
Query OK, 0 rows affected (0.03 sec)
```

```
mysql> INSERT INTO theatre (t_id, t_name, location)
    -> VALUES
    -> ('T01', 'Rajhans', 'Surat'),
    -> ('T02', 'Rupali', 'Vadodara'),
    -> ('T03', 'Valentine', 'Delhi'),
    -> ('T04', 'Cinepolis', 'Mumbai'),
    -> ('T05', 'Cinemax', 'Surat');
Query OK, 5 rows affected (0.01 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM theatre;
+------+-----------+----------+
| t_id | t_name    | location |
+------+-----------+----------+
| T01  | Rajhans   | Surat    |
| T02  | Rupali    | Vadodara |
| T03  | Valentine | Delhi    |
| T04  | Cinepolis | Mumbai   |
| T05  | Cinemax   | Surat    |
+------+-----------+----------+
5 rows in set (0.00 sec)
```

Seat Table:

Constraints:

- Primary key – seat_id
- Check constraint – seat_name has a check constraint that ensures that the value is one of ‘Classic’, ‘Executive’, ‘Royal’ . no_of_seats has a check constraint that ensure that the ni. Of seats is greater than 0.

```
mysql> CREATE TABLE seat (
    ->    seat_id VARCHAR(6) PRIMARY KEY CHECK (SEAT_ID LIKE 'S%'),
    ->    seat_name VARCHAR(10) CHECK (seat_name IN ('classic','executive,'royal')),
    ->    no_of_seats TINYINT(10) CHECK (no_of seats > 0)
    -> );
```

```
mysql> INSERT INTO seat (seat_id, seat_name, no_of_seats)
    -> VALUES
    -> ('S01', 'Royal', 10),
    -> ('S02', 'Executive', 100),
    -> ('S03', 'Classic', 80);
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM seat;
+---------+-----------+-------------+
| seat_id | seat_name | no_of_seats |
+---------+-----------+-------------+
| S01     | Royal     |          10 |
| S02     | Executive |         100 |
| S03     | Classic   |          80 |
+---------+-----------+-------------+
3 rows in set (0.00 sec)
```

Theatre_has_seat Table:

Constraints:

- Primary key – t_id and seat_id
- Foreign key – theatre_has_seat.t_id references theatre.t_id and theatre_has_seat.seat_id references seat.seat_id

```
mysql> CREATE TABLE theatre_has_seat (
    ->     t_id VARCHAR(6),
    ->     seat_id VARCHAR(6),
    ->     PRIMARY KEY (t_id, seat_id),
    ->     FOREIGN KEY (t_id) REFERENCES theatre(t_id),
    ->     FOREIGN KEY (seat_id) REFERENCES seat(seat_id)
    -> );
Query OK, 0 rows affected (0.44 sec)
```

```
mysql> INSERT INTO theatre_has_seat (t_id, seat_id)
    -> VALUES
    -> ('T03', 'S01'),
    -> ('T03', 'S02'),
    -> ('T04', 'S03'),
    -> ('T05', 'S01'),
    -> ('T05', 'S02'),
    -> ('T05', 'S03'),
    -> ('T02', 'S01'),
    -> ('T01', 'S03');
Query OK, 8 rows affected (0.04 sec)
Records: 8  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM theatre_has_seat;
+------+---------+
| t_id | seat_id |
+------+---------+
| T02  | S01     |
| T03  | S01     |
| T05  | S01     |
| T03  | S02     |
| T05  | S02     |
| T01  | S03     |
| T04  | S03     |
| T05  | S03     |
+------+---------+
8 rows in set (0.00 sec)
```

Screen Table:

Constraints:

- Primary key – screen_id
- Check constraint - screen_id should start with 'SC'
- Foreign key – screen.t_id references theatre.t_id and screen.m_id references movie.m_id

```
mysql> CREATE TABLE screen (
    ->     screen_id VARCHAR(6) PRIMARY KEY CHECK (screen_id LIKE 'SC%'),
    ->     screen_no TINYINT(2),
    ->     t_id VARCHAR(6),
    ->     m_id VARCHAR(6)
    -> );
Query OK, 0 rows affected, 1 warning (0.01 sec)
```

```
mysql> ALTER TABLE screen
    -> ADD CONSTRAINT fk_screen_theatre
    -> FOREIGN KEY (t_id) REFERENCES theatre(t_id)
    -> ON DELETE CASCADE;
Query OK, 0 rows affected (0.06 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> ALTER TABLE screen
    -> ADD CONSTRAINT fk_screen_movie
    -> FOREIGN KEY (m_id) REFERENCES movie (m_id)
    -> ON DELETE CASCADE;
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
mysql> INSERT INTO screen (screen_id, screen_no, t_id, m_id)
    -> VALUES
    -> ('SC01', 1, 'T01', 'M01'),
    -> ('SC02', 2, 'T03', 'M04'),
    -> ('SC03', 1, 'T05', 'M02'),
    -> ('SC04', 3, 'T03', 'M01'),
    -> ('SC05', 1, 'T03', 'M03'),
    -> ('SC06', 3, 'T02', 'M04');
Query OK, 6 rows affected (0.01 sec)
Records: 6  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM screen;
+-----------+-----------+------+------+
| screen_id | screen_no | t_id | m_id |
+-----------+-----------+------+------+
| SC01      |         1 | T01  | M01  |
| SC02      |         2 | T03  | M04  |
| SC03      |         1 | T05  | M02  |
| SC04      |         3 | T03  | M01  |
| SC05      |         1 | T03  | M03  |
| SC06      |         3 | T02  | M04  |
+-----------+-----------+------+------+
6 rows in set (0.00 sec)
```

Movie Table:

Constraints:

- Primary key – m_id
- Not Null – m_name, m_type, m_hour, release_date, director, actor
- Check constraint – m_id should start with ‘M’

```
mysql> CREATE TABLE movie (
    ->     m_id VARCHAR(6) PRIMARY KEY CHECK (m_id LIKE 'M%'),
    ->     m_name VARCHAR(50) NOT NULL,
    ->     m_type VARCHAR(10) NOT NULL,
    ->     m_hour INT(6) NOT NULL,
    ->     release_date DATE NOT NULL,
    ->     director VARCHAR(20) NOT NULL,
    ->     actor VARCHAR(20) NOT NULL
    -> );
Query OK, 0 rows affected, 1 warning (0.04 sec)
```

```
mysql> INSERT INTO movie (m_id, m_name, m_type, m_hour, release_date, director, actor)
    -> VALUES
    -> ('M01', 'Don', 'Action', 2, STR_TO_DATE('12-02-2012', '%d-%m-%Y'), 'Farhan Akhtar', 'Shahrukh Khan'),
    -> ('M02', 'Radhe', 'Action', 3, STR_TO_DATE('14-05-2021', '%d-%m-%Y'), 'Prabhu Deva', 'Salman Khan'),
    -> ('M03', 'KabirSingh', 'Romance', 3, STR_TO_DATE('23-07-2019', '%d-%m-%Y'), 'Sandeep Reddy', 'Shahid Kapoor'),
    -> ('M04', 'NH10', 'Thriller', 2, STR_TO_DATE('29-10-2015', '%d-%m-%Y'), 'Navdeep Singh', 'Anushka Sharma');
Query OK, 4 rows affected (0.02 sec)
Records: 4  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM movie;
+------+------------+----------+--------+--------------+---------------+----------------+
| m_id | m_name     | m_type   | m_hour | release_date | director      | actor          |
+------+------------+----------+--------+--------------+---------------+----------------+
| M01  | Don        | Action   |      2 | 2012-02-12   | Farhan Akhtar | Shahrukh Khan  |
| M02  | Radhe      | Action   |      3 | 2021-05-14   | Prabhu Deva   | Salman Khan    |
| M03  | KabirSingh | Romance  |      3 | 2019-07-23   | Sandeep Reddy | Shahid Kapoor  |
| M04  | NH10       | Thriller |      2 | 2015-10-29   | Navdeep Singh | Anushka Sharma |
+------+------------+----------+--------+--------------+---------------+----------------+
4 rows in set (0.00 sec)
```

Customer Table:

Constraints:

- Primary key – c_id
- Not Null – c_fname, c_lname, c-phoneno
- Check constraint – c_id should start with ‘C’ and c_gender should be in ‘M’ or ‘F’

```
mysql> CREATE TABLE customer (
    ->     c_id VARCHAR(6) PRIMARY KEY CHECK (c_id LIKE 'C%'),
    ->     c_fname VARCHAR(20) NOT NULL,
    ->     c_lname VARCHAR(20) NOT NULL,
    ->     c_gender CHAR(1) CHECK (c_gender IN ('M', 'F')),
    ->     c_email VARCHAR(20),
    ->     c_phoneno BIGINT(10) NOT NULL,
    ->     c_address VARCHAR(30),
    ->     m_id VARCHAR(6)
    -> );
Query OK, 0 rows affected, 1 warning (0.03 sec)
```

```
mysql> ALTER TABLE customer
    -> ADD CONSTRAINT fk_cust_movie
    -> FOREIGN KEY (m_id)
    -> REFERENCES
    -> movie(m_id)
    -> ON DELETE CASCADE;
Query OK, 0 rows affected (0.04 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
mysql> INSERT INTO customer (c_id, c_fname, c_lname, c_gender, c_email, c_phoneno, c_address, m_id)
    -> VALUES
    -> ('C01', 'Neel', 'Patel', 'M', 'neel@gmail.com', 9656034521, 'Surat', 'M02'),
    -> ('C02', 'Hardik', 'Patel', 'M', 'hardik@gmail.com', 6455214572, 'Delhi', 'M04'),
    -> ('C03', 'Amisha', 'Mishra', 'F', 'amisham@gmail.com', 9034572245, 'Vadodara', 'M04'),
    -> ('C04', 'Pooja', 'Yadav', 'F', 'pooja@gmail.com', 9712373735, 'Surat', 'M01');
Query OK, 4 rows affected (0.01 sec)
Records: 4  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM customer;
+------+---------+---------+----------+-------------------+------------+-----------+------+
| c_id | c_fname | c_lname | c_gender | c_email           | c_phoneno  | c_address | m_id |
+------+---------+---------+----------+-------------------+------------+-----------+------+
| C01  | Neel    | Patel   | M        | neel@gmail.com    | 9656034521 | Surat     | M02  |
| C02  | Hardik  | Patel   | M        | hardik@gmail.com  | 6455214572 | Delhi     | M04  |
| C03  | Amisha  | Mishra  | F        | amisham@gmail.com | 9034572245 | Vadodara  | M04  |
| C04  | Pooja   | Yadav   | F        | pooja@gmail.com   | 9712373735 | Surat     | M01  |
+------+---------+---------+----------+-------------------+------------+-----------+------+
4 rows in set (0.01 sec)
```

Payment Table:

Constraints:

- Primary key – payment_id
- Check constraint – payment_id should strat with ‘P’ and payment_type should be in ‘Credit’, ‘Debit’, ‘Net Banking’
- Not Null – payment_type, payment_date, amount
- Foreign key – payment.c_id references customer.c_id

```
mysql> CREATE TABLE payment (
    ->     payment_id VARCHAR(6) PRIMARY KEY CHECK (payment_id LIKE 'P%'),
    ->     payment_type VARCHAR(15) CHECK (payment_type IN ('Credit', 'Debit', 'Net Banking')) NOT NULL,
    ->     payment_date DATE NOT NULL,
    ->     amount INT NOT NULL,
    ->     c_id VARCHAR(6)
    -> );
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> ALTER TABLE payment
    -> ADD CONSTRAINT fk_pay_cust
    -> FOREIGN KEY (c_id)
    -> REFERENCES
    -> customer (c_id)
    -> ON DELETE CASCADE;
Query OK, 0 rows affected (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
mysql> INSERT INTO payment (payment_id, payment_type, payment_date, amount, c_id)
    -> VALUES
    -> ('P01', 'Credit', STR_TO_DATE('30-10-2015', '%d-%m-%Y'), 550, 'C03'),
    -> ('P02', 'Net Banking', STR_TO_DATE('01-01-2016', '%d-%m-%Y'), 695, 'C02'),
    -> ('P03', 'Debit', STR_TO_DATE('17-05-2021', '%d-%m-%Y'), 1500, 'C01'),
    -> ('P04', 'Credit', STR_TO_DATE('26-02-2012', '%d-%m-%Y'), 1000, 'C04');
Query OK, 4 rows affected (0.01 sec)
Records: 4  Duplicates: 0  Warnings: 0
```

```
mysql> SELECT * FROM payment;
+------------+--------------+--------------+--------+------+
| payment_id | payment_type | payment_date | amount | c_id |
+------------+--------------+--------------+--------+------+
| P01        | Credit       | 2015-10-30   |    550 | C03  |
| P02        | Net Banking  | 2016-01-01   |    695 | C02  |
| P03        | Debit        | 2021-05-17   |   1500 | C01  |
| P04        | Credit       | 2012-02-26   |   1000 | C04  |
+------------+--------------+--------------+--------+------+
4 rows in set (0.00 sec)
```

Shows Table:

Constraints:

- Primary key – show_id
- Check constraint – show_id should start with ‘SH’. Language should be in ‘Hindi’, ‘English’
- Not Null – show_start, show_end

```
mysql> CREATE TABLE shows (
    ->     show_id VARCHAR(6) PRIMARY KEY  CHECK  (show_id LIKE 'SH%'),
    ->     language VARCHAR(8) CHECK (language IN ('Hindi', 'English')),
    ->     show_start VARCHAR(10) NOT NULL,
    ->     show_end VARCHAR(10) NOT NULL
    -> );
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> INSERT INTO shows (show_id, language, show_start, show_end)
    -> VALUES
    -> ('SH01', 'Hindi', '12:45pm', '2:45pm'),
    -> ('SH02', 'Hindi', '9:15am', '12:15pm'),
    -> ('SH03', 'Hindi', '8:45pm', '10:45pm'),
    -> ('SH04', 'Hindi', '7:30pm', '9:30pm'),
    -> ('SH05', 'Hindi', '3:30pm', '5:30pm');
Query OK, 5 rows affected (0.00 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM shows;
+---------+----------+------------+----------+
| show_id | language | show_start | show_end |
+---------+----------+------------+----------+
| SH01    | Hindi    | 12:45pm    | 2:45pm   |
| SH02    | Hindi    | 9:15am     | 12:15pm  |
| SH03    | Hindi    | 8:45pm     | 10:45pm  |
| SH04    | Hindi    | 7:30pm     | 9:30pm   |
| SH05    | Hindi    | 3:30pm     | 5:30pm   |
+---------+----------+------------+----------+
5 rows in set (0.00 sec)
```

Discount Table:

Constraints:

- Primary key – offer_id
- Check constraint – offer_id should start with ‘O’
- Not Null – m_name, price

```
mysql> CREATE TABLE discount (
    ->     offer_id VARCHAR(6) PRIMARY KEY CHECK (offer_id LIKE 'O%'),
    ->     m_name VARCHAR(30) NOT NULL,
    ->     price INT NOT NULL
    -> );
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> INSERT INTO discount (offer_id, m_name, price)
    -> VALUES
    -> ('O01', 'NH10', 50),
    -> ('O02', 'Don', 60),
    -> ('O03', 'Radhe', 90),
    -> ('O04', 'KabirSingh', 150);
Query OK, 4 rows affected (0.01 sec)
Records: 4  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM discount;
+----------+------------+-------+
| offer_id | m_name     | price |
+----------+------------+-------+
| O01      | NH10       |    50 |
| O02      | Don        |    60 |
| O03      | Radhe      |    90 |
| O04      | KabirSingh |   150 |
+----------+------------+-------+
4 rows in set (0.00 sec)
```

Ticket Table:

Consrtraints:

- Primary key – ticket_id
- Check constraint – ticket_id should start with ‘t’. seat_no, price should be greater than 0
- Not Null – show_date, screen_no
- Foreign key – ticket.t_id references theatre.t_id (ON DELETE CASCADE), ticket.c_id references customer.c_id (ON DELETE CASCADE), ticket.offer_id references discount.offer_id (ON DELETE CASCADE), ticket.show_id references shows.show_id (ON DELETE CASCADE)

```
mysql> CREATE TABLE ticket (
    ->     ticket_id VARCHAR(6) PRIMARY KEY CHECK (ticket_id LIKE 't%'),
    ->     show_date DATE NOT NULL,
    ->     seat_no INT CHECK (seat_no > 0),
    ->     t_id VARCHAR(6),
    ->     screen_no INT NOT NULL,
    ->     price INT CHECK (price > 0),
    ->     c_id VARCHAR(6),
    ->     offer_id VARCHAR(6),
    ->     show_id VARCHAR(6),
    ->     booking_date DATE
    -> );
Query OK, 0 rows affected (0.03 sec)
```

```
mysql> ALTER TABLE ticket
    -> ADD CONSTRAINT fk_tick_theatre
    -> FOREIGN KEY (t_id)
    -> REFERENCES
    -> theatre (t_id)
    -> ON DELETE CASCADE;
Query OK, 0 rows affected (0.06 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> ALTER TABLE ticket
    -> ADD CONSTRAINT fk_tick_cust
    -> FOREIGN KEY (c_id)
    -> REFERENCES
    -> customer (c_id)
    -> ON DELETE CASCADE;
Query OK, 0 rows affected (0.06 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
mysql> ALTER TABLE ticket
    -> ADD CONSTRAINT fk_tick_disc
    -> FOREIGN KEY (offer_id)
    -> REFERENCES
    -> discount (offer_id)
    -> ON DELETE CASCADE;
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> ALTER TABLE ticket
    -> ADD CONSTRAINT fk_tick_shows
    -> FOREIGN KEY (show_id)
    -> REFERENCES
    -> shows (show_id)
    -> ON DELETE CASCADE;
Query OK, 0 rows affected (0.04 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
mysql> INSERT INTO ticket (ticket_id, show_date, seat_no, t_id, screen_no, price, c_id, offer_id, show_id, booking_date)
    -> VALUES
    -> ('t01', STR_TO_DATE('28-02-2012', '%d-%m-%Y'), 4, 'T01', 1, 940, 'C04', 'O02', 'SH04', STR_TO_DATE('26-02-2012', '%d-%m-%Y')),
    -> ('t02', STR_TO_DATE('01-11-2015', '%d-%m-%Y'), 2, 'T02', 3, 500, 'C03', 'O01', 'SH01', STR_TO_DATE('30-10-2015', '%d-%m-%Y')),
    -> ('t03', STR_TO_DATE('18-05-2021', '%d-%m-%Y'), 5, 'T05', 1, 1410, 'C01', 'O03', 'SH02', STR_TO_DATE('17-05-2021', '%d-%m-%Y')),
    -> ('t04', STR_TO_DATE('03-01-2016', '%d-%m-%Y'), 3, 'T03', 2, 645, 'C02', 'O01', 'SH05', STR_TO_DATE('01-01-2016', '%d-%m-%Y'));
Query OK, 4 rows affected (0.00 sec)
Records: 4  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM ticket;
+-----------+------------+---------+------+-----------+-------+------+----------+---------+--------------+
| ticket_id | show_date  | seat_no | t_id | screen_no | price | c_id | offer_id | show_id | booking_date |
+-----------+------------+---------+------+-----------+-------+------+----------+---------+--------------+
| t01       | 2012-02-28 |       4 | T01  |         1 |   940 | C04  | O02      | SH04    | 2012-02-26   |
| t02       | 2015-11-01 |       2 | T02  |         3 |   500 | C03  | O01      | SH01    | 2015-10-30   |
| t03       | 2021-05-18 |       5 | T05  |         1 |  1410 | C01  | O03      | SH02    | 2021-05-17   |
| t04       | 2016-01-03 |       3 | T03  |         2 |   645 | C02  | O01      | SH05    | 2016-01-01   |
+-----------+------------+---------+------+-----------+-------+------+----------+---------+--------------+
4 rows in set (0.00 sec)
```

Sells Table:

Constraints:

- Primary key – ticket_id, t_id
- Check constraint - ticket_id should start with ‘t’, t_id should start with 'T’
- Foreign key – sells.ticket_id references ticket.ticket_id (ON DELETE CASCADE), sells.t_id references theatre.t_id (ON DELETE CASCADE)

```
mysql> CREATE TABLE sells (
    ->     ticket_id VARCHAR(6) CHECK (ticket_id LIKE 't%'),
    ->     t_id VARCHAR(6) CHECK (t_id LIKE 'T%'),
    ->     PRIMARY KEY (ticket_id, t_id),
    ->     FOREIGN KEY (ticket_id) REFERENCES ticket (ticket_id) ON DELETE CASCADE,
    ->     FOREIGN KEY (t_id) REFERENCES theatre (t_id) ON DELETE CASCADE
    -> );
Query OK, 0 rows affected (0.03 sec)
```

```
mysql> INSERT INTO sells (ticket_id, t_id)
    -> VALUES
    -> ('t01', 'T01'),
    -> ('t02', 'T02'),
    -> ('t03', 'T05'),
    -> ('t04', 'T03');
Query OK, 4 rows affected (0.01 sec)
Records: 4  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM sells;
+-----------+------+
| ticket_id | t_id |
+-----------+------+
| t01       | T01  |
| t02       | T02  |
| t04       | T03  |
| t03       | T05  |
+-----------+------+
4 rows in set (0.00 sec)
```

All the Tables inserted in the Database:

```
mysql> SHOW TABLES;
+-----------------------------+
| Tables_in_movie_booking_dbms |
+-----------------------------+
| admin                       |
| customer                    |
| discount                    |
| movie                       |
| payment                     |
| screen                      |
| seat                        |
| sells                       |
| shows                       |
| theatre                     |
| theatre_has_seat            |
| ticket                      |
+-----------------------------+
12 rows in set (0.02 sec)
```

Queries:

1. **Display theatre name and theatre location in which admins work.**

```
mysql> SELECT admin_id, CONCAT(admin_fname,' ', admin_lname) AS admin_name, t_name, location
    -> FROM admin
    -> INNER JOIN theatre ON admin.t_id = theatre.t_id;
+----------+--------------+-----------+----------+
| admin_id | admin_name   | t_name    | location |
+----------+--------------+-----------+----------+
| A01      | Yash Mehta   | Rajhans   | Surat    |
| A02      | Manav Joshi  | Rajhans   | Surat    |
| A03      | Smit Jariwala| Cinepolis | Mumbai   |
| A04      | Disha Patel  | Rupali    | Vadodara |
| A05      | Zeel Desai   | Valentine | Delhi    |
+----------+--------------+-----------+----------+
5 rows in set (0.01 sec)
```

2. **Display admin name, number of months he/she worked in the theatre and there experience.**

```
mysql> SELECT admin_fname,
    -> TIMESTAMPDIFF(MONTH, doj, NOW()) AS months_worked,
    -> TIMESTAMPDIFF(YEAR, doj, NOW()) AS experience
    -> FROM admin;
+-------------+---------------+------------+
| admin_fname | months_worked | experience |
+-------------+---------------+------------+
| Yash        |           138 |         11 |
| Manav       |           145 |         12 |
| Smit        |           221 |         18 |
| Disha       |            98 |          8 |
| Zeel        |           126 |         10 |
+-------------+---------------+------------+
5 rows in set (0.00 sec)
```

3. **Display theatre name and screen no having movie name as 'NH10'.**

```
mysql> SELECT t_name, screen_no
    -> FROM theatre
    -> INNER JOIN screen ON theatre.t_id = screen.t_id
    -> INNER JOIN movie ON screen.m_id = movie.m_id
    -> WHERE movie.m_name = 'NH10';
+-----------+-----------+
| t_name    | screen_no |
+-----------+-----------+
| Valentine |         2 |
| Rupali    |         3 |
+-----------+-----------+
2 rows in set (0.02 sec)
```

4. **Display customer first name, customer last name, customer gender and movie for which corresponding customer has booked the ticket.**

```
mysql> SELECT CONCAT(c_fname,' ',c_lname) AS customer_name, c_gender AS gender, m_name AS movie_name
    -> FROM customer
    -> INNER JOIN movie ON customer.m_id = movie.m_id;
+---------------+--------+------------+
| customer_name | gender | movie_name |
+---------------+--------+------------+
| Neel Patel    | M      | Radhe      |
| Hardik Patel  | M      | NH10       |
| Amisha Mishra | F      | NH10       |
| Pooja Yadav   | F      | Don        |
+---------------+--------+------------+
4 rows in set (0.02 sec)
```

5. **Display payment id, payment date, payment type and customer name who had paid more than 600.**

```
mysql> SELECT payment_id, payment_date, payment_type, CONCAT(c_fname,' ',c_lname) AS customer_name
    -> FROM customer
    -> INNER JOIN payment ON customer.c_id = payment.c_id
    -> WHERE payment.amount > 600;
+------------+--------------+--------------+---------------+
| payment_id | payment_date | payment_type | customer_name |
+------------+--------------+--------------+---------------+
| P02        | 2016-01-01   | Net Banking  | Hardik Patel  |
| P03        | 2021-05-17   | Debit        | Neel Patel    |
| P04        | 2012-02-26   | Credit       | Pooja Yadav   |
+------------+--------------+--------------+---------------+
3 rows in set (0.01 sec)
```

6. **Display ticket id, theatre name, customer name , language, show date and timing of movie.**

```
mysql> SELECT ticket_id, t_name, CONCAT(c_fname,' ',c_lname) AS customer_name, Language, show_date, show_start, show_end
    -> FROM ticket
    -> INNER JOIN theatre ON ticket.t_id = theatre.t_id
    -> INNER JOIN customer ON ticket.c_id = customer.c_id
    -> INNER JOIN shows ON ticket.show_id = shows.show_id;
+-----------+-----------+---------------+----------+------------+------------+----------+
| ticket_id | t_name    | customer_name | Language | show_date  | show_start | show_end |
+-----------+-----------+---------------+----------+------------+------------+----------+
| t02       | Rupali    | Amisha Mishra | Hindi    | 2015-11-01 | 12:45pm    | 2:45pm   |
| t03       | Cinemax   | Neel Patel    | Hindi    | 2021-05-18 | 9:15am     | 12:15pm  |
| t01       | Rajhans   | Pooja Yadav   | Hindi    | 2012-02-28 | 7:30pm     | 9:30pm   |
| t04       | Valentine | Hardik Patel  | Hindi    | 2016-01-03 | 3:30pm     | 5:30pm   |
+-----------+-----------+---------------+----------+------------+------------+----------+
4 rows in set (0.01 sec)
```

7. **Display customer name show date and booking date where customer has booked ticket two days prior to the show date.**

```
mysql> SELECT CONCAT(c_fname,' ',c_lname) AS customer_name, show_date, booking_date
    -> FROM customer
    -> INNER JOIN ticket ON customer.c_id = ticket.c_id
    -> WHERE show_date - booking_date = 2;
+---------------+------------+--------------+
| customer_name | show_date  | booking_date |
+---------------+------------+--------------+
| Hardik Patel  | 2016-01-03 | 2016-01-01   |
| Pooja Yadav   | 2012-02-28 | 2012-02-26   |
+---------------+------------+--------------+
2 rows in set (0.01 sec)
```

8. **Display ticket id, customer name, movie name and ticket price after applying corresponding discount.**

```
mysql> SELECT ticket_id, CONCAT(c_fname,' ',c_lname) AS customer_name, m_name AS movie_name, ticket.price - discount.price AS after_discount_price
    -> FROM ticket
    -> INNER JOIN discount ON ticket.offer_id = discount.offer_id
    -> INNER JOIN customer ON ticket.c_id = customer.c_id;
+-----------+---------------+------------+----------------------+
| ticket_id | customer_name | movie_name | after_discount_price |
+-----------+---------------+------------+----------------------+
| t03       | Neel Patel    | Radhe      |                 1320 |
| t04       | Hardik Patel  | NH10       |                  595 |
| t02       | Amisha Mishra | NH10       |                  450 |
| t01       | Pooja Yadav   | Don        |                  880 |
+-----------+---------------+------------+----------------------+
4 rows in set (0.02 sec)
```

9. **Display the show id, movie name, and the number of tickets booked for each show**

```
mysql> SELECT shows.show_id, m_name AS movie_name, COUNT(ticket_id) AS tickets_booked
    -> FROM shows
    -> LEFT JOIN ticket ON shows.show_id = ticket.show_id
    -> INNER JOIN movie ON movie.m_id = (SELECT m_id FROM customer WHERE customer.c_id = (SELECT c_id FROM ticket WHERE ticket.show_id = shows.show_id))
    -> GROUP BY show_id, m_name;
+---------+------------+---------------+
| show_id | movie_name | tickets_booked |
+---------+------------+---------------+
| SH01    | NH10       |             1 |
| SH02    | Radhe      |             1 |
| SH04    | Don        |             1 |
| SH05    | NH10       |             1 |
+---------+------------+---------------+
4 rows in set (0.01 sec)
```

10. **Display movie details which release after 2018.**

```
mysql> SELECT * FROM movie WHERE EXTRACT(YEAR FROM release_date)>2018;
+------+------------+---------+--------+--------------+---------------+---------------+
| m_id | m_name     | m_type  | m_hour | release_date | director      | actor         |
+------+------------+---------+--------+--------------+---------------+---------------+
| M02  | Radhe      | Action  |      3 | 2021-05-14   | Prabhu Deva   | Salman Khan   |
| M03  | KabirSingh | Romance |      3 | 2019-07-23   | Sandeep Reddy | Shahid Kapoor |
+------+------------+---------+--------+--------------+---------------+---------------+
2 rows in set (0.00 sec)
```

11. **Display payment details in ascending order by amount.**

```
mysql> SELECT * FROM payment ORDER BY amount ASC;
+------------+--------------+--------------+--------+------+
| payment_id | payment_type | payment_date | amount | c_id |
+------------+--------------+--------------+--------+------+
| P01        | Credit       | 2015-10-30   |    550 | C03  |
| P02        | Net Banking  | 2016-01-01   |    695 | C02  |
| P04        | Credit       | 2012-02-26   |   1000 | C04  |
| P03        | Debit        | 2021-05-17   |   1500 | C01  |
+------------+--------------+--------------+--------+------+
4 rows in set (0.00 sec)
```