DBMS PROJECT

TOPIC: MOVIE TICKET BOOKING SYSTEM

PROBLEM STATEMENT:

Movie Ticket Booking Database Design is basically aimed to provide complete information of the movie and schedule to the customer according to which he or she can easily book tickets for their favourite movies. The database administrator can insert and delete data like movie schedules, shows timings, genre, cast and crew details, etc.

One can have wholesome cinematic experience by just booking tickets from their mobiles which saves their time and reduces effort. The details of the customer, tickets, payments, etc are stored. Customer can select the movies of their choice by taking the rating of movie into consideration.

ASSUMPTIONS:

- A customer can book any number of tickets with a single customer id.
- Only one seat can be booked for a single ticket.
- One theatre can have multiple seat type names with different costs.
- Multiple theatres can have seat types with same name.
- One customer can make many payments.
- Any number of tickets can be booked in a single payment.
- One movie can be screened in any number of shows.
- This database is applicable to single screen theatres.
- Any number of shows can be screened in a theatre in a single day.

TABLES:

1. CUSTOMER:

Attributes	Datatypes	Constraints
Customer_id	Int	Primary key
First_name	Varchar (30)	NOT NULL
Last_name	Varchar (30)	NOT NULL
age	Int	NOT NULL
Gender	Varchar (1)	NOT NULL
Mobile	Varchar (10)	UNIQUE, NOT NULL

email	Varchar (20)	UNIQUE

2.MOVIE:

Attributes	Datatype	constraints
Movie_id	Int	Primary key
Movie_name	Varchar (30)	NOT NULL
Genre	Varchar (30)	NOT NULL
Releasing_date	Date	NOT NULL
Rating	Decimal (2,1)	-
Cast	Varchar (50)	NOT NULL
Duration	Time	NOT NULL

3.THEATRE:

Attributes	Datatype	Constraints
Theatre_id	Int	Primary key
Theatre_name	Varchar (20)	NOT NULL
Location	Varchar (20)	NOT NULL
Capacity	Int	-

4.SHOWS:

Attributes	Datatype	Constraints
Show_name	Varchar (20)	Primary key (1)
Theatre_id	Int	Primary key (2), Foreign key (1)
Show_date	Date	Primary key (3)
Show_time	Time	NOT NULL
Movie_language	Varchar (20)	NOT NULL
Movie_id	Int	Foreign key (2)

5.PAYMENT:

Attributes	Datatype	Constraints
Payment_id	Int	Primary key
Price	Int	NOT NULL

6.SEAT_TYPE:

Attributes	Datatype	Constraints
Seat_type_name	Varchar (20)	Primary key (1)
Theatre_id	Int	Primary key (2), Foreign key
Cost	Int	NOT NULL
Seat_type_capacity	Int	NOT NULL

7.TICKET:

Attributes	Datatype	Constraints
Ticket_id	Int	Primary key
Customer_id	Int	Foreign key (1)
Theatre_id	Int	Foreign key (2)
Payment_id	Int	Foreign key (3)
Show_id	Int	Foreign key (4)
Show_data	Int	Foreign key (5)

8.SEAT:

Attributes	Datatype	Constraints
Seat_number	Varchar (5)	NOT NULL
Seat_type_name	Varchar (20)	Foreign key (1)
Theatre_id	Int	Foreign key (2)
Ticket_id	Int	Primary key Foreign key (3)

FUNTIONAL DEPENCENCIES AND PRIMARY KEY:

1. CUSTOMER:

Customer_id-> {First_name, Last_name, age, gender, mobile, email} Since all the fields depend on customer_id, (customer_id) + -> R. Hence, Customer_id is a primary key.

2.MOVIE:

Movie_id-> {Movie_name, genre, release_date, rating, cast, duration} Since all the fields depend on Movie_id, (Movie_id) + -> R.

Hence, Movie_id is a primary key.

3.THEATRE:

Theatre_id-> {Theatre_name, location, capacity}

Since all the fields depend on Theatre_id, (Theatre_id) + -> R. Hence, Theatre_id is a primary key.

4.SHOWS:

{Show_name, Theatre_id, Show_date}-> {Show_time, Movie_language, Movie_id}

Since all the fields depend on (Theatre_id, Show_name, Show_date) + -> R.

Hence, (Show_name, Show_date, Theatre_id) are combinedly a composite primary key.

5.PAYMENT:

Payment_id-> {Price, Payment_mode, Payment_date, Customer_id} Since all the fields depend on Payment_id, (Payment_id) + -> R. Hence, Payment_id is a primary key.

6.SEAT_TYPE:

(Seat_type_name, Theatre_id)-> {Cost, Seat_type_capacity} Since all the fields depend on Seat_type_name and Theatre_id, (Seat_type_name, Theatre_id) +->R

Hence, (Seat_type_name, Theatre_id) combinedly becomes a composite primary key.

7.TICKET:

Ticket_id-> {Customer_id, Theatre_id, Payment_id, Show_name, Show_date} Since all fields depends on Ticket_id, (Ticket_id) + ->R

Hence, Ticket_id is a primary key.

8.SEAT:

Ticket_id-> (Seat_number, Seat_type_name, Theatre_id) Since all fields depends on Ticket_id, (Ticket_id) +->R Hence, Ticket id is a primary key

NORMALISATION:

1. CUSTOMER:

Primary key: customer_id

All attributes depend on the Customer_id, hence the table is in 2NF.

All attributes depend directly on Customer_id hence the table is in 3NF. All determinants (customer_id) is Super key, hence the table is in BCNF.

2.MOVIE:

Primary key: Movie_id

All attributes depend on the Movie_id, hence the table is in 2NF.

All attributes depend directly on Movie_id hence the table is in 3NF. All determinants (Movie_id) is Super key, hence the table is in BCNF.

3.THEATRE:

Primary key: Theatre_id

All attributes depend on the Theatre _id, hence the table is in 2NF.

All attributes depend directly on Theatre _id hence the table is in 3NF. All determinants (Theatre _id) is Super key, hence the table is in BCNF.

4.SHOWS:

Primary key: Theatre_id, Show_name, Show_date

All attributes depend on the Theatre_id, Show_name, Show_date hence the table is in 2NF.

All attributes depend directly on Theatre_id, Show_name, Show_date hence the table is in 3NF.

All determinants (Theatre_id, Show_name, Show_date) is Super key, hence the table is in BCNF.

5.PAYMENT:

Primary key: Payment_id

All attributes depend on the Payment_id hence the table is in 2NF.

All attributes depend directly on Payment_id hence the table is in 3NF. All determinants (Payment_id) is Super key, hence the table is in BCNF.

6.SEAT_TYPE:

Primary key: Theatre_id, Seat_type_name

All attributes depend on the Theatre_id, Seat_type_name hence the table is in 2NF.

All attributes depend directly on Theatre_id, Seat_type_name hence the table is in 3NF.

All determinants (Theatre_id, Seat_type_name) is Super key, hence the table is in BCNF.

7.TICKET:

Primary key: Ticket_id

All attributes depend on the Ticket_id hence the table is in 2NF.

All attributes depend directly on Ticket_id hence the table is in 3NF. All determinants (Ticket_id) is Super key, hence the table is in BCNF.

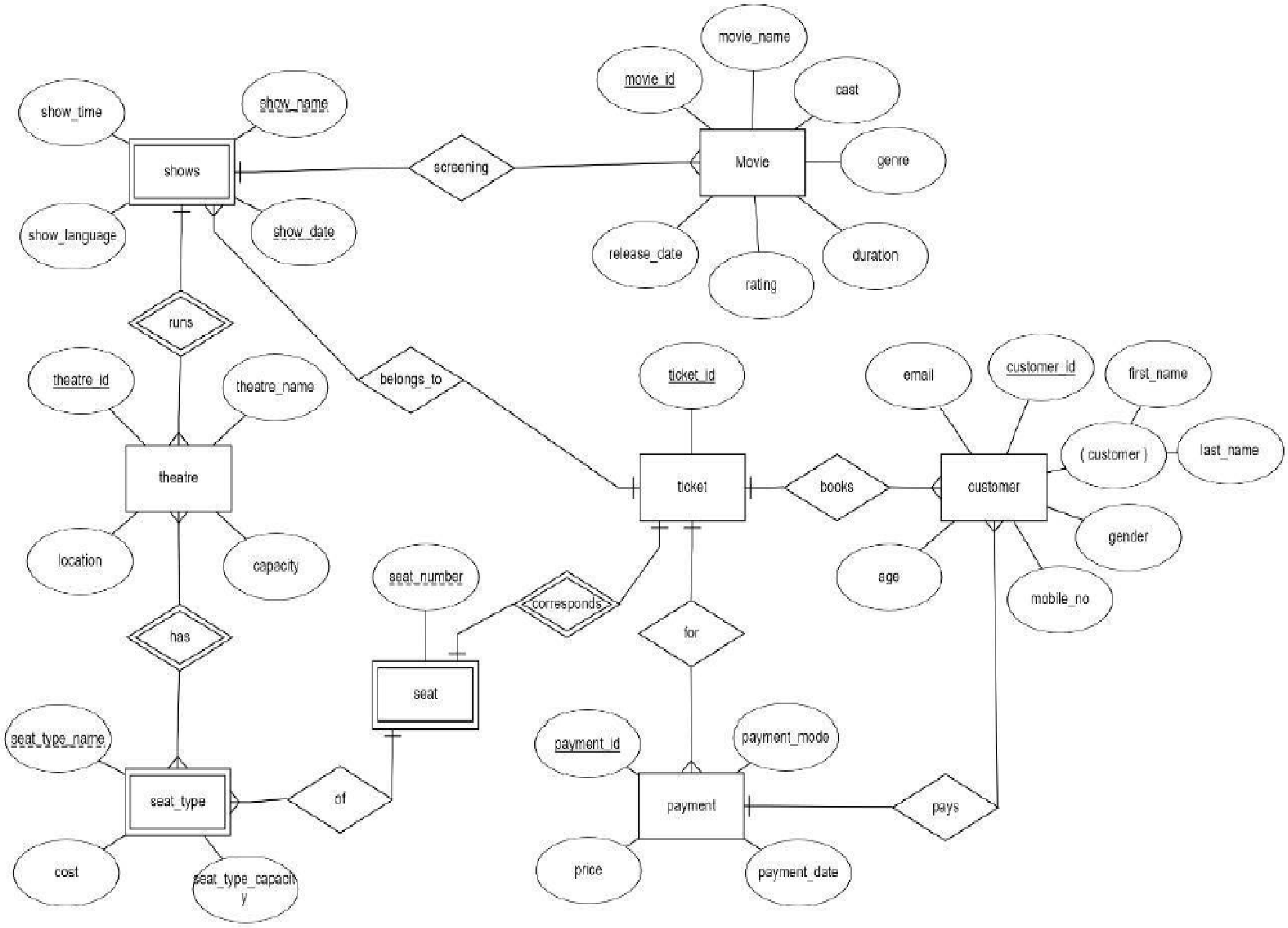
8.SEAT:

Primary key: Ticket_id

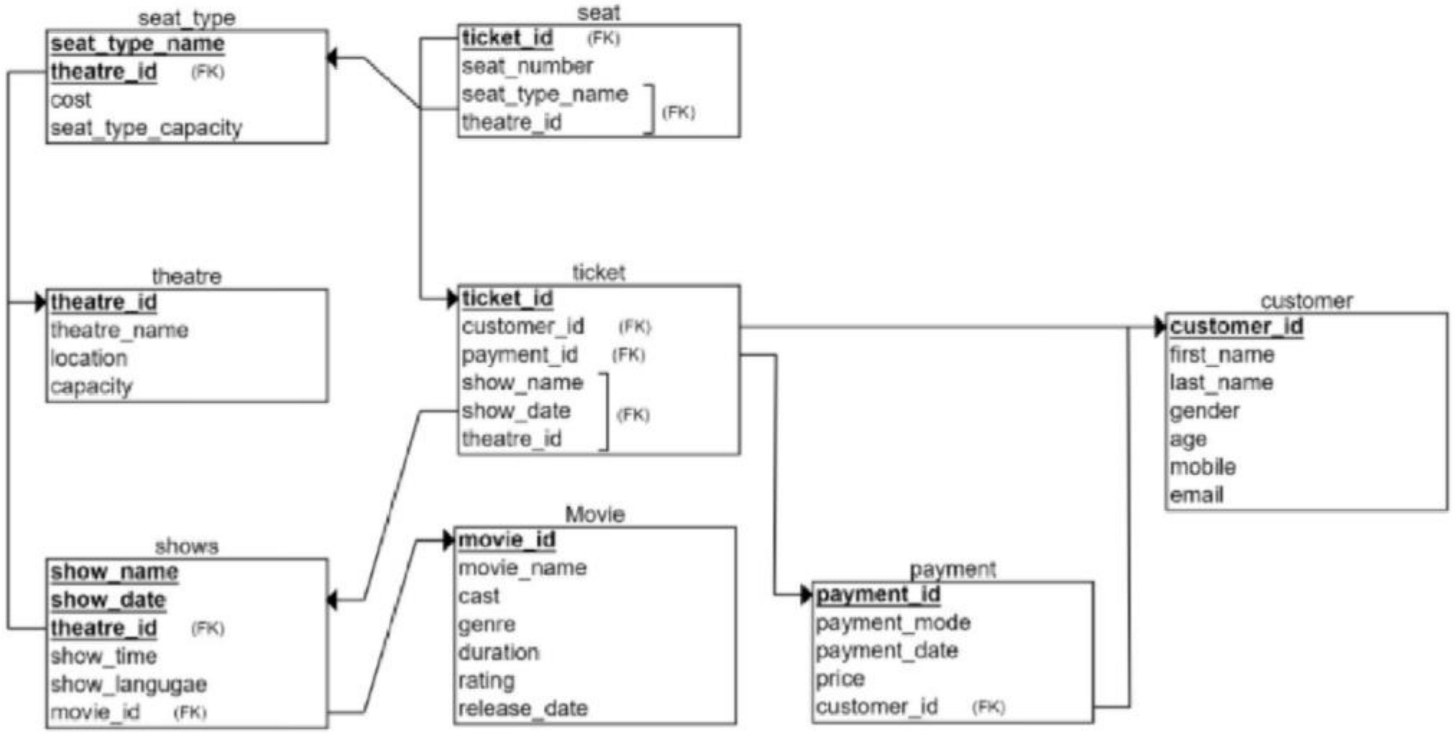
All attributes depend on the Ticket_id hence the table is in 2NF.

All attributes depend directly on Ticket_id hence the table is in 3NF. All determinants (Ticket_id) is Super key, hence the table is in BCNF.

ER DIAGRAM:

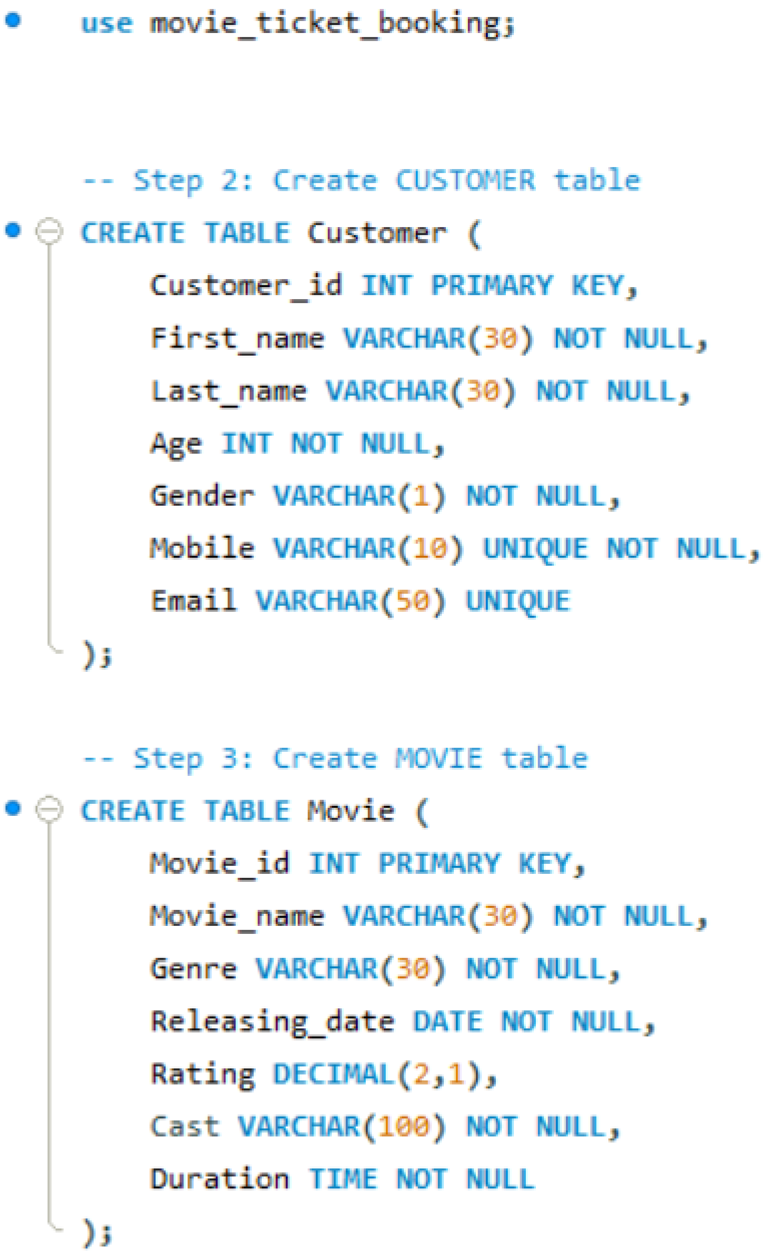

movie_name
movie_id
cast
show_time
show_name
shows
screening
Movie
genre
show_language
show_date
release_date
duration
rating
runs
theatre_id
theatre_name
belongs_to
ticket_id
email
customer_id
first_name
(customer)
last_name
theatre
ticket
books
customer
gender
location
capacity
seat_number
corresponds
age
mobile_no
has
for
seat
seat_type_name
payment_id
payment_mode
seat_type
of
payment
pays
cost
seat_type_capacit
y
price
payment_date

RELATIONAL SCHEMA:

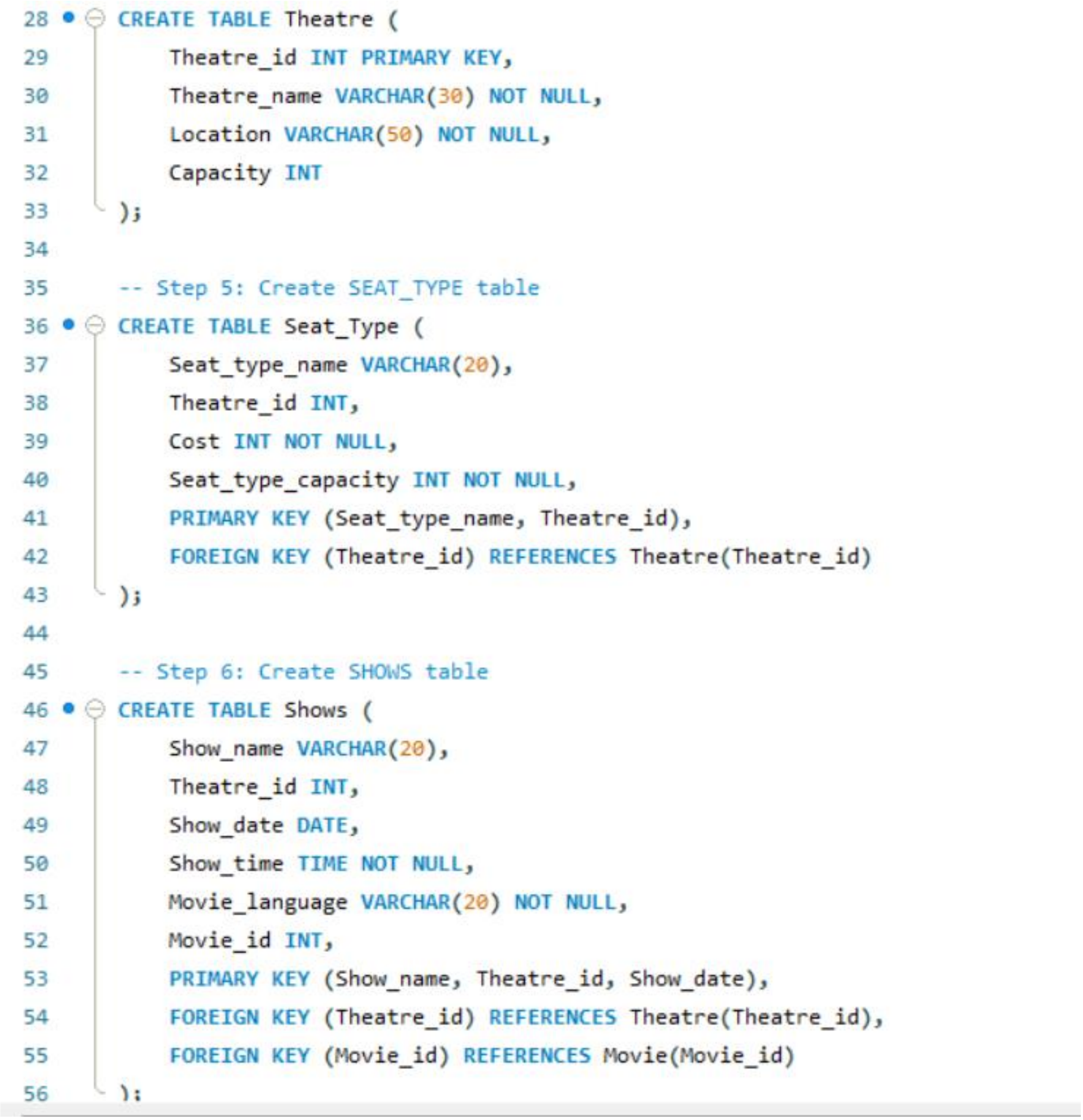


MYSQL CODE:

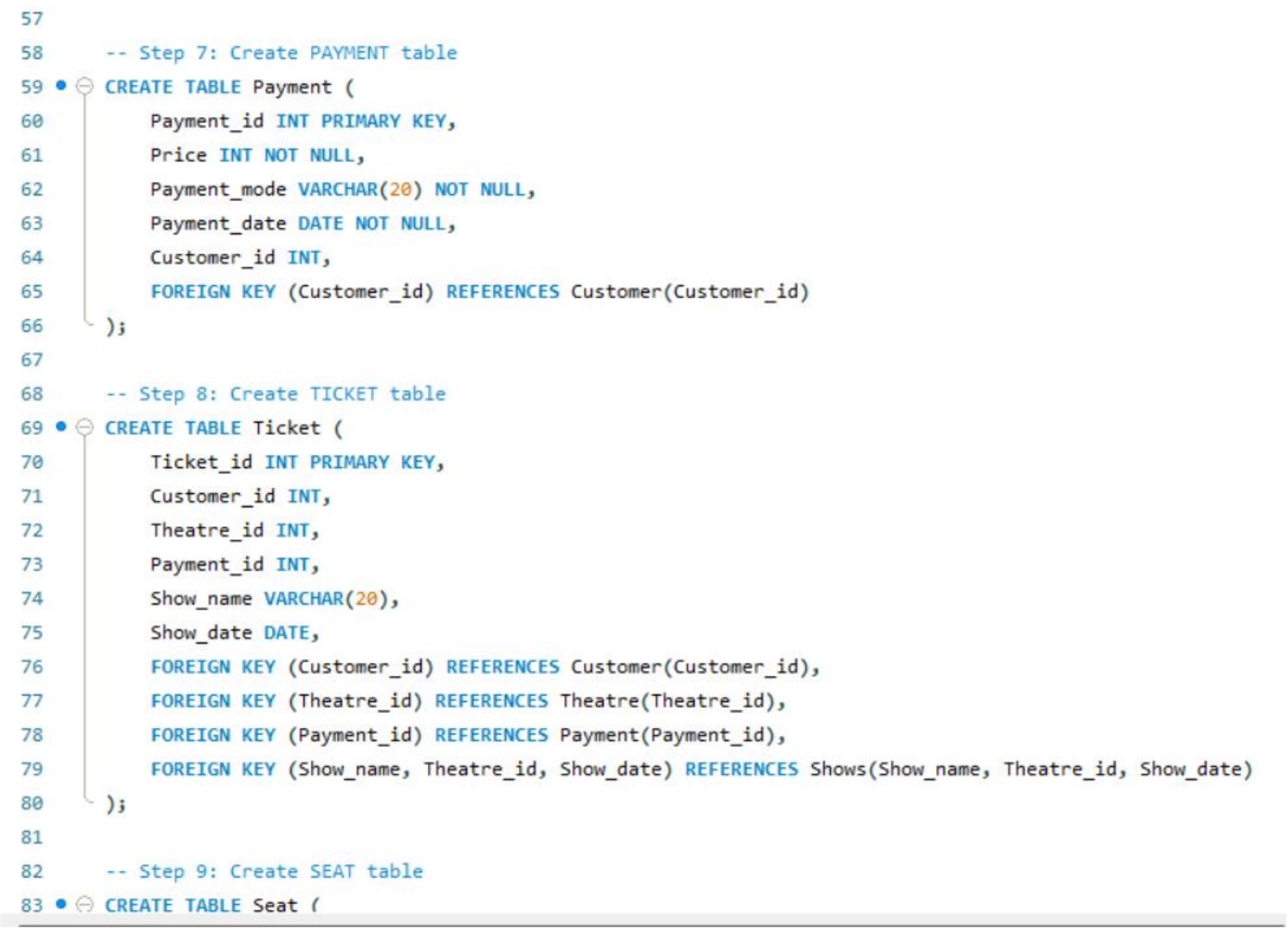
CREATING TABLES:



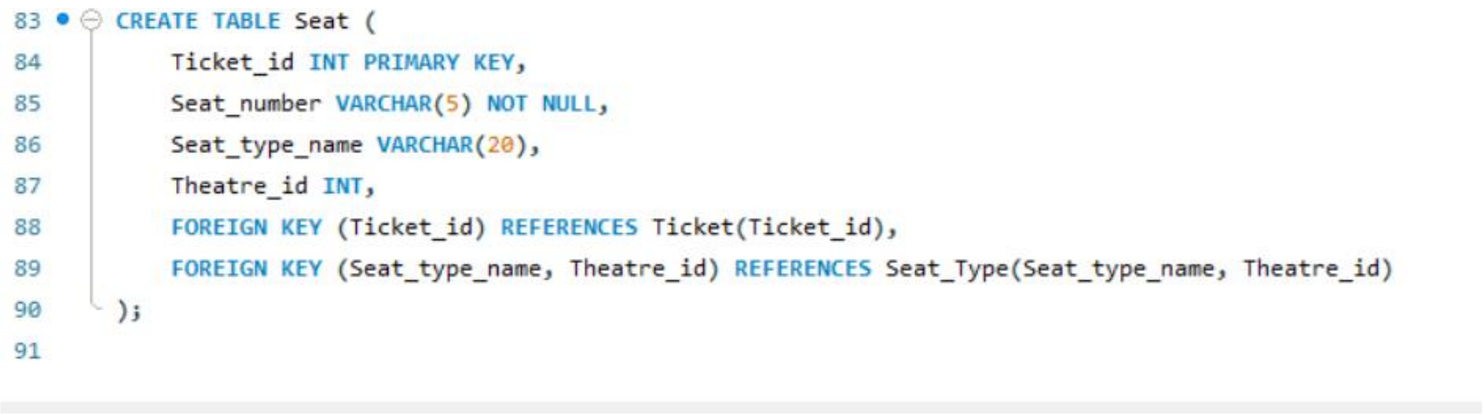
```
use movie_ticket_booking;


-- Step 2: Create CUSTOMER table
CREATE TABLE Customer (
    Customer_id INT PRIMARY KEY,
    First_name VARCHAR(30) NOT NULL,
    Last_name VARCHAR(30) NOT NULL,
    Age INT NOT NULL,
    Gender VARCHAR(1) NOT NULL,
    Mobile VARCHAR(10) UNIQUE NOT NULL,
    Email VARCHAR(50) UNIQUE
);

-- Step 3: Create MOVIE table
CREATE TABLE Movie (
    Movie_id INT PRIMARY KEY,
    Movie_name VARCHAR(30) NOT NULL,
    Genre VARCHAR(30) NOT NULL,
    Releasing_date DATE NOT NULL,
    Rating DECIMAL(2,1),
    Cast VARCHAR(100) NOT NULL,
    Duration TIME NOT NULL
);
```



```sql
CREATE TABLE Theatre (
    Theatre_id INT PRIMARY KEY,
    Theatre_name VARCHAR(30) NOT NULL,
    Location VARCHAR(50) NOT NULL,
    Capacity INT
);

-- Step 5: Create SEAT_TYPE table
CREATE TABLE Seat_Type (
    Seat_type_name VARCHAR(20),
    Theatre_id INT,
    Cost INT NOT NULL,
    Seat_type_capacity INT NOT NULL,
    PRIMARY KEY (Seat_type_name, Theatre_id),
    FOREIGN KEY (Theatre_id) REFERENCES Theatre(Theatre_id)
);

-- Step 6: Create SHOWS table
CREATE TABLE Shows (
    Show_name VARCHAR(20),
    Theatre_id INT,
    Show_date DATE,
    Show_time TIME NOT NULL,
    Movie_language VARCHAR(20) NOT NULL,
    Movie_id INT,
    PRIMARY KEY (Show_name, Theatre_id, Show_date),
    FOREIGN KEY (Theatre_id) REFERENCES Theatre(Theatre_id),
    FOREIGN KEY (Movie_id) REFERENCES Movie(Movie_id)
);
```

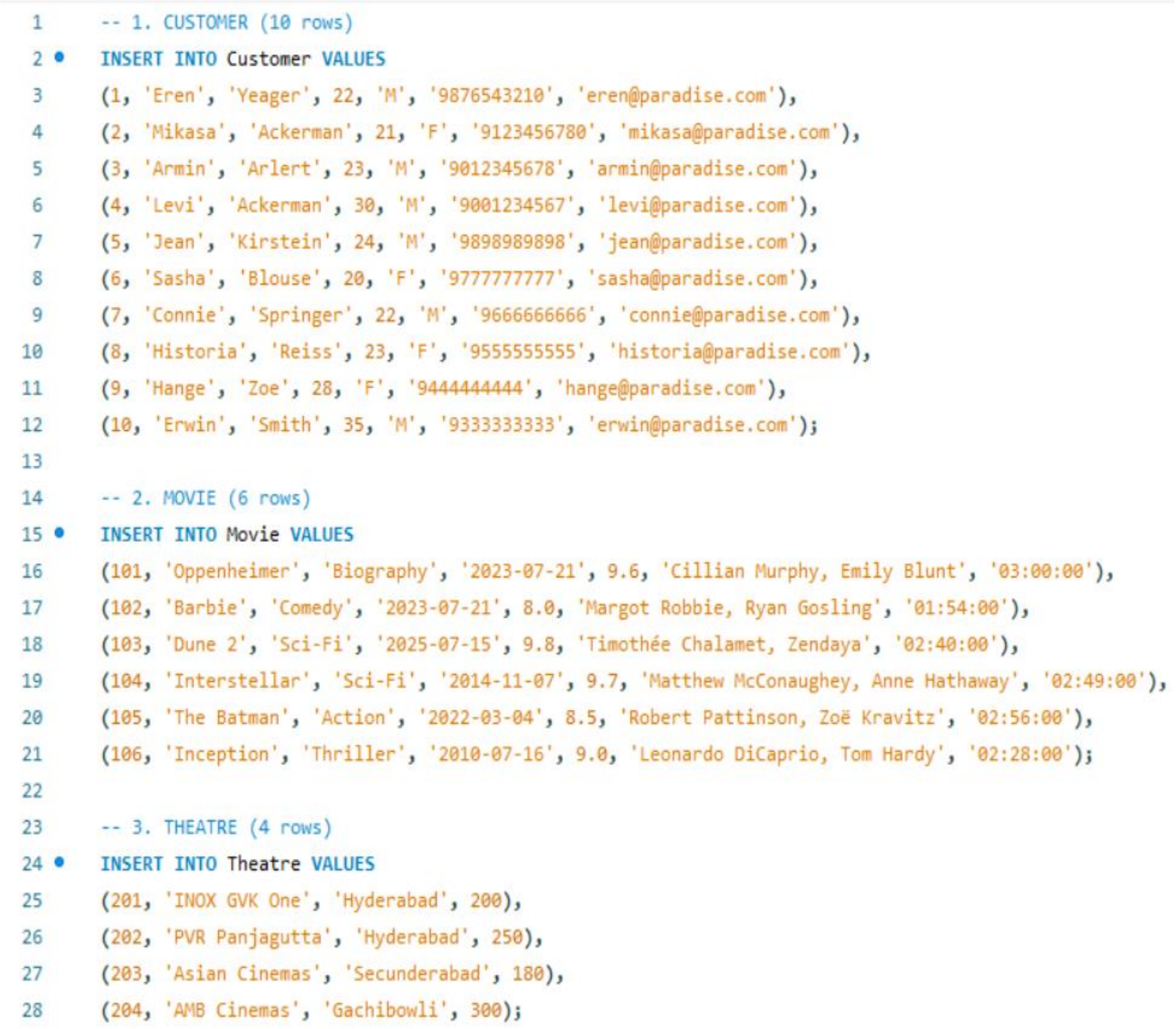


```sql

-- Step 7: Create PAYMENT table
CREATE TABLE Payment (
    Payment_id INT PRIMARY KEY,
    Price INT NOT NULL,
    Payment_mode VARCHAR(20) NOT NULL,
    Payment_date DATE NOT NULL,
    Customer_id INT,
    FOREIGN KEY (Customer_id) REFERENCES Customer(Customer_id)
);

-- Step 8: Create TICKET table
CREATE TABLE Ticket (
    Ticket_id INT PRIMARY KEY,
    Customer_id INT,
    Theatre_id INT,
    Payment_id INT,
    Show_name VARCHAR(20),
    Show_date DATE,
    FOREIGN KEY (Customer_id) REFERENCES Customer(Customer_id),
    FOREIGN KEY (Theatre_id) REFERENCES Theatre(Theatre_id),
    FOREIGN KEY (Payment_id) REFERENCES Payment(Payment_id),
    FOREIGN KEY (Show_name, Theatre_id, Show_date) REFERENCES Shows(Show_name, Theatre_id, Show_date)
);

-- Step 9: Create SEAT table
CREATE TABLE Seat (
```

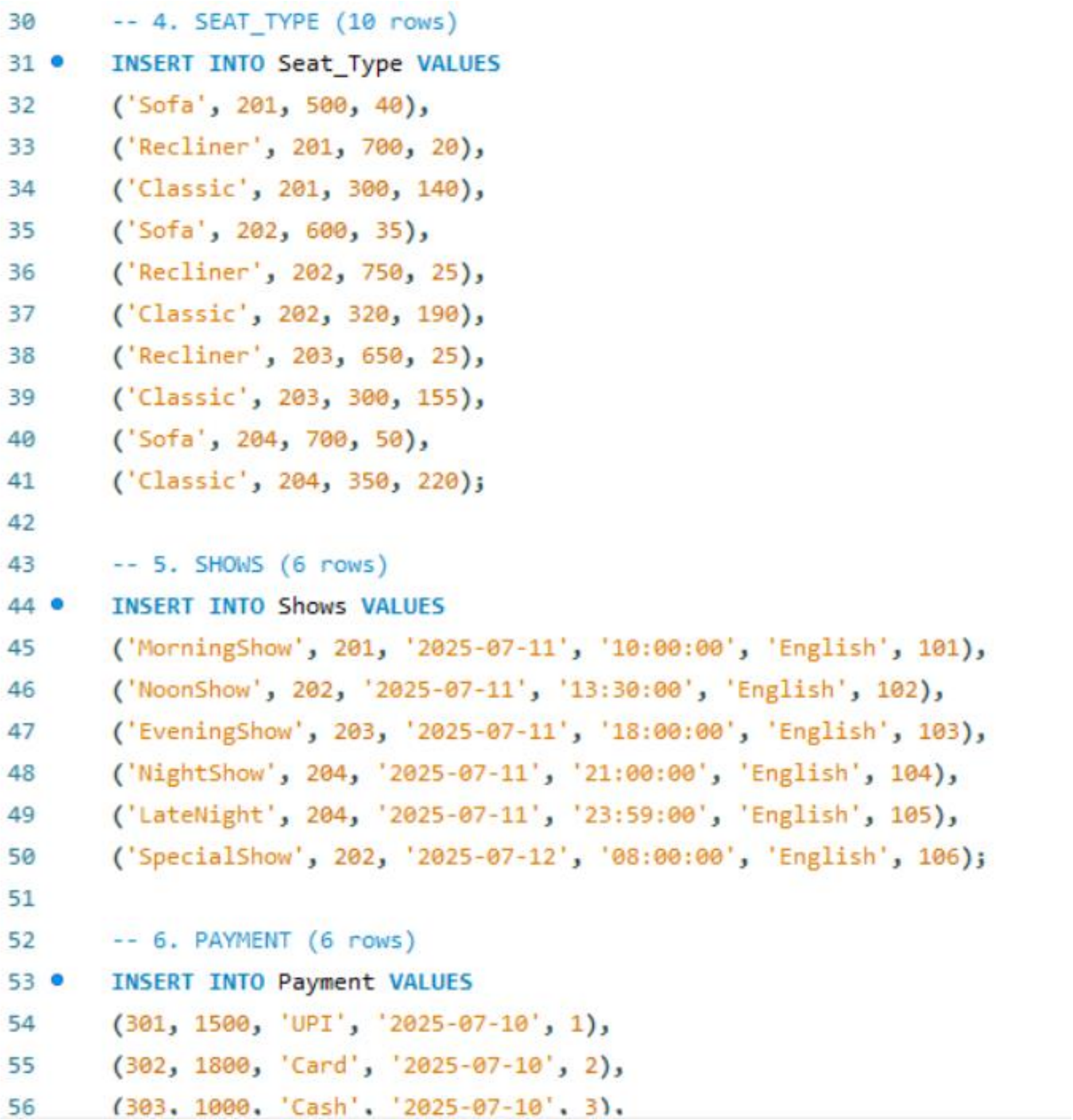


```sql
CREATE TABLE Seat (
    Ticket_id INT PRIMARY KEY,
    Seat_number VARCHAR(5) NOT NULL,
    Seat_type_name VARCHAR(20),
    Theatre_id INT,
    FOREIGN KEY (Ticket_id) REFERENCES Ticket(Ticket_id),
    FOREIGN KEY (Seat_type_name, Theatre_id) REFERENCES Seat_Type(Seat_type_name, Theatre_id)
);

```

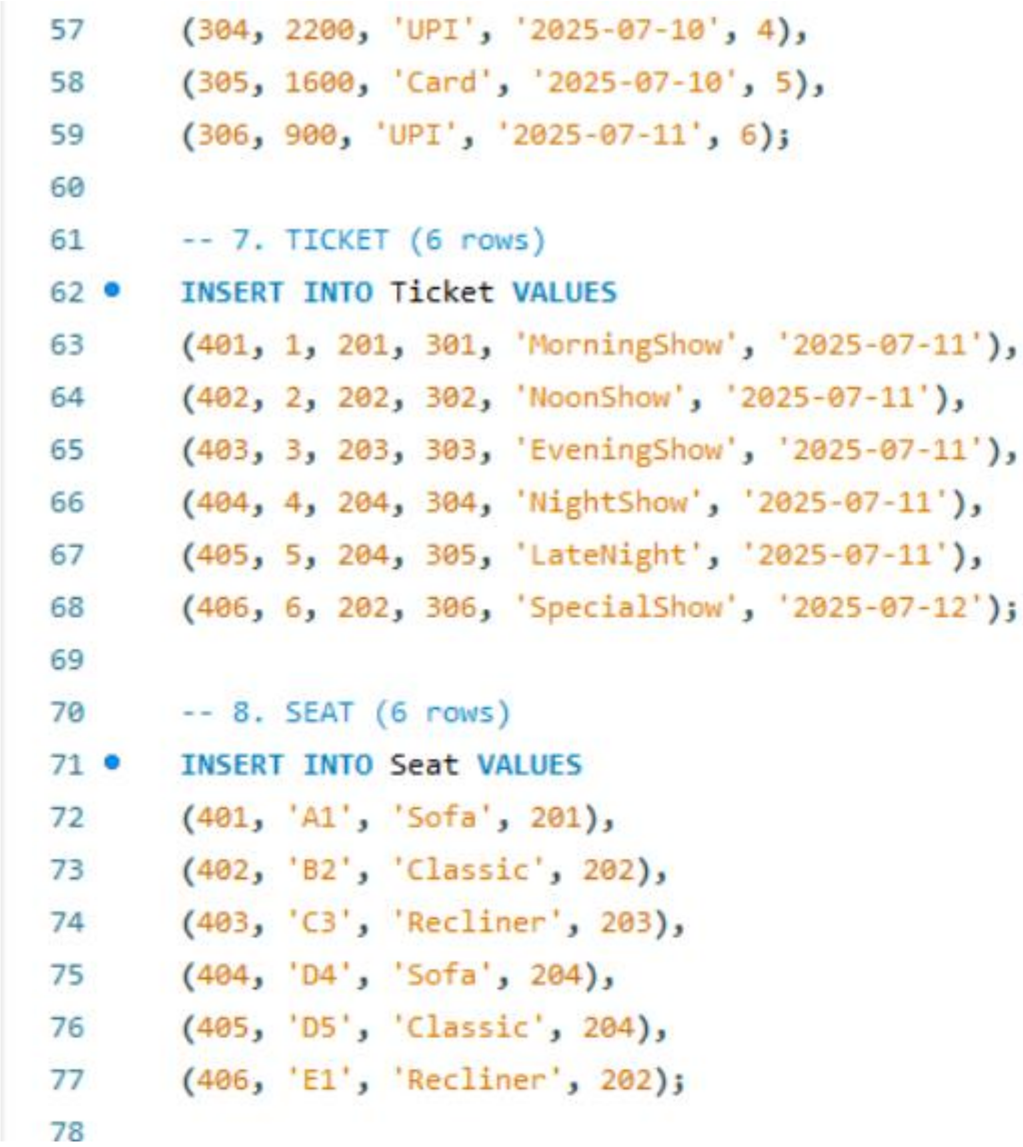
INSERTING DATA:



```sql
-- 1. CUSTOMER (10 rows)
INSERT INTO Customer VALUES
(1, 'Eren', 'Yeager', 22, 'M', '9876543210', 'eren@paradise.com'),
(2, 'Mikasa', 'Ackerman', 21, 'F', '9123456780', 'mikasa@paradise.com'),
(3, 'Armin', 'Arlert', 23, 'M', '9012345678', 'armin@paradise.com'),
(4, 'Levi', 'Ackerman', 30, 'M', '9001234567', 'levi@paradise.com'),
(5, 'Jean', 'Kirstein', 24, 'M', '9898989898', 'jean@paradise.com'),
(6, 'Sasha', 'Blouse', 20, 'F', '9777777777', 'sasha@paradise.com'),
(7, 'Connie', 'Springer', 22, 'M', '9666666666', 'connie@paradise.com'),
(8, 'Historia', 'Reiss', 23, 'F', '9555555555', 'historia@paradise.com'),
(9, 'Hange', 'Zoe', 28, 'F', '9444444444', 'hange@paradise.com'),
(10, 'Erwin', 'Smith', 35, 'M', '9333333333', 'erwin@paradise.com');

-- 2. MOVIE (6 rows)
INSERT INTO Movie VALUES
(101, 'Oppenheimer', 'Biography', '2023-07-21', 9.6, 'Cillian Murphy, Emily Blunt', '03:00:00'),
(102, 'Barbie', 'Comedy', '2023-07-21', 8.0, 'Margot Robbie, Ryan Gosling', '01:54:00'),
(103, 'Dune 2', 'Sci-Fi', '2025-07-15', 9.8, 'Timothée Chalamet, Zendaya', '02:40:00'),
(104, 'Interstellar', 'Sci-Fi', '2014-11-07', 9.7, 'Matthew McConaughey, Anne Hathaway', '02:49:00'),
(105, 'The Batman', 'Action', '2022-03-04', 8.5, 'Robert Pattinson, Zoë Kravitz', '02:56:00'),
(106, 'Inception', 'Thriller', '2010-07-16', 9.0, 'Leonardo DiCaprio, Tom Hardy', '02:28:00');

-- 3. THEATRE (4 rows)
INSERT INTO Theatre VALUES
(201, 'INOX GVK One', 'Hyderabad', 200),
(202, 'PVR Panjagutta', 'Hyderabad', 250),
(203, 'Asian Cinemas', 'Secunderabad', 180),
(204, 'AMB Cinemas', 'Gachibowli', 300);
```

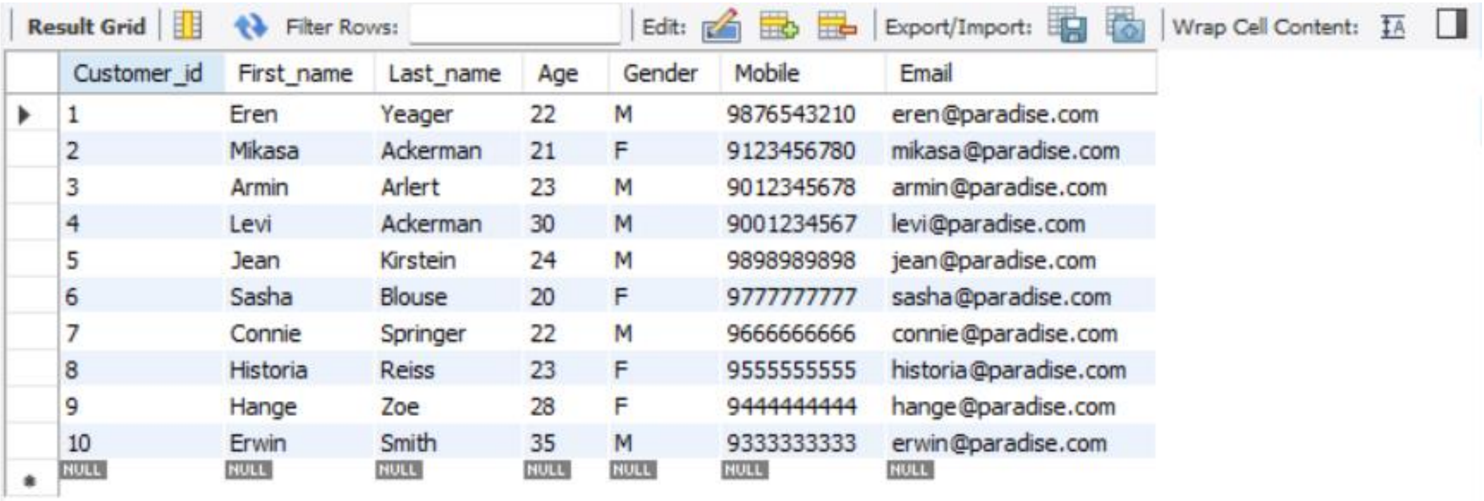


```
-- 4. SEAT_TYPE (10 rows)
INSERT INTO Seat_Type VALUES
('Sofa', 201, 500, 40),
('Recliner', 201, 700, 20),
('Classic', 201, 300, 140),
('Sofa', 202, 600, 35),
('Recliner', 202, 750, 25),
('Classic', 202, 320, 190),
('Recliner', 203, 650, 25),
('Classic', 203, 300, 155),
('Sofa', 204, 700, 50),
('Classic', 204, 350, 220);

-- 5. SHOWS (6 rows)
INSERT INTO Shows VALUES
('MorningShow', 201, '2025-07-11', '10:00:00', 'English', 101),
('NoonShow', 202, '2025-07-11', '13:30:00', 'English', 102),
('EveningShow', 203, '2025-07-11', '18:00:00', 'English', 103),
('NightShow', 204, '2025-07-11', '21:00:00', 'English', 104),
('LateNight', 204, '2025-07-11', '23:59:00', 'English', 105),
('SpecialShow', 202, '2025-07-12', '08:00:00', 'English', 106);

-- 6. PAYMENT (6 rows)
INSERT INTO Payment VALUES
(301, 1500, 'UPI', '2025-07-10', 1),
(302, 1800, 'Card', '2025-07-10', 2),
(303, 1000, 'Cash', '2025-07-10', 3),
```

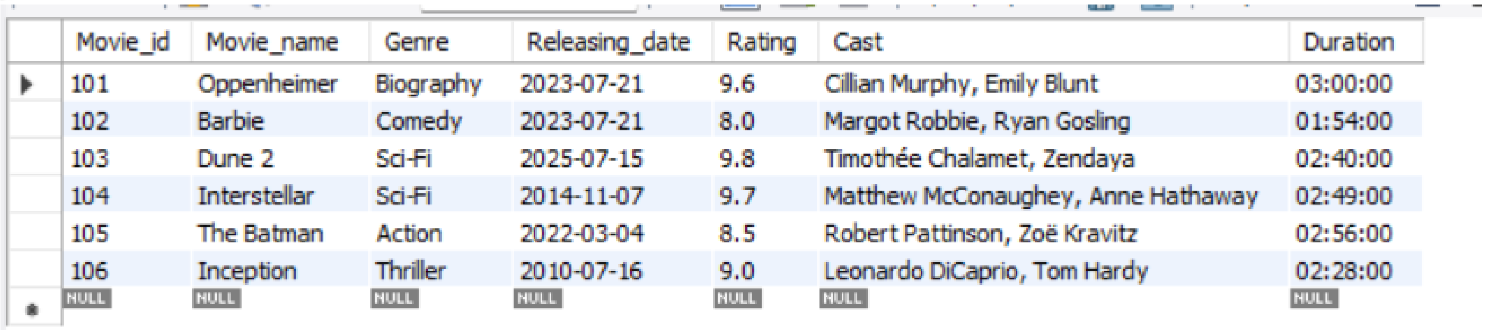


```sql
(304, 2200, 'UPI', '2025-07-10', 4),
(305, 1600, 'Card', '2025-07-10', 5),
(306, 900, 'UPI', '2025-07-11', 6);

-- 7. TICKET (6 rows)
INSERT INTO Ticket VALUES
(401, 1, 201, 301, 'MorningShow', '2025-07-11'),
(402, 2, 202, 302, 'NoonShow', '2025-07-11'),
(403, 3, 203, 303, 'EveningShow', '2025-07-11'),
(404, 4, 204, 304, 'NightShow', '2025-07-11'),
(405, 5, 204, 305, 'LateNight', '2025-07-11'),
(406, 6, 202, 306, 'SpecialShow', '2025-07-12');

-- 8. SEAT (6 rows)
INSERT INTO Seat VALUES
(401, 'A1', 'Sofa', 201),
(402, 'B2', 'Classic', 202),
(403, 'C3', 'Recliner', 203),
(404, 'D4', 'Sofa', 204),
(405, 'D5', 'Classic', 204),
(406, 'E1', 'Recliner', 202);
```

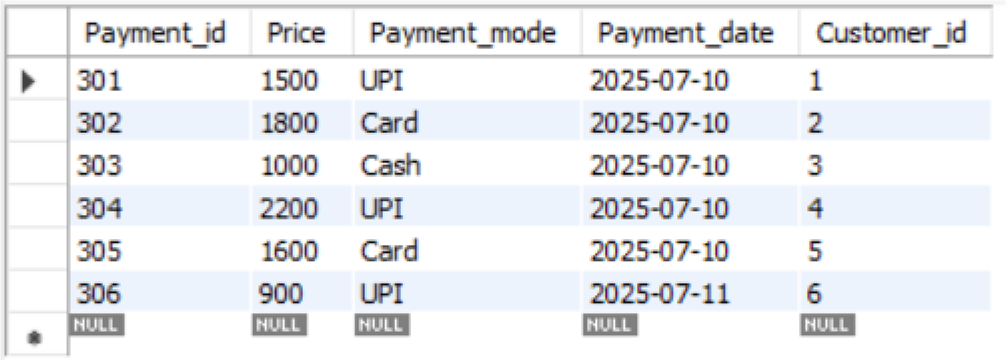
TABLES CREATED:



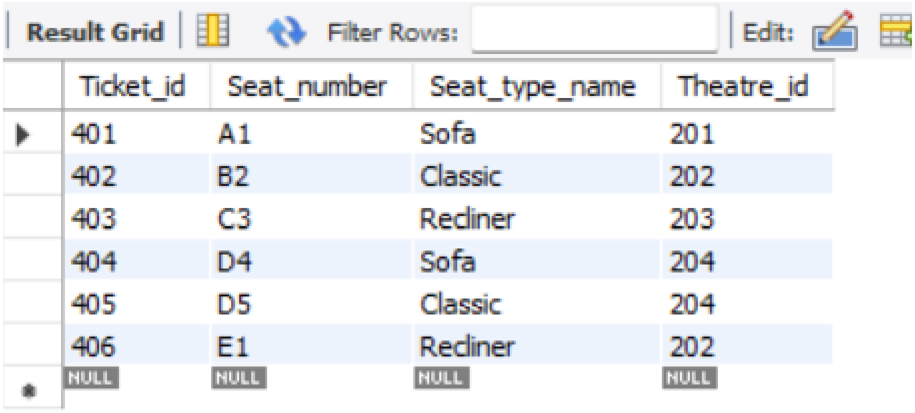
Customer_id	First_name	Last_name	Age	Gender	Mobile	Email
1	Eren	Yeager	22	M	9876543210	eren@paradise.com
2	Mikasa	Ackerman	21	F	9123456780	mikasa@paradise.com
3	Armin	Arlert	23	M	9012345678	armin@paradise.com
4	Levi	Ackerman	30	M	9001234567	levi@paradise.com
5	Jean	Kirstein	24	M	9898989898	jean@paradise.com
6	Sasha	Blouse	20	F	9777777777	sasha@paradise.com
7	Connie	Springer	22	M	9666666666	connie@paradise.com
8	Historia	Reiss	23	F	9555555555	historia@paradise.com
9	Hange	Zoe	28	F	9444444444	hange@paradise.com
10	Erwin	Smith	35	M	9333333333	erwin@paradise.com
NULL	NULL	NULL	NULL	NULL	NULL	NULL



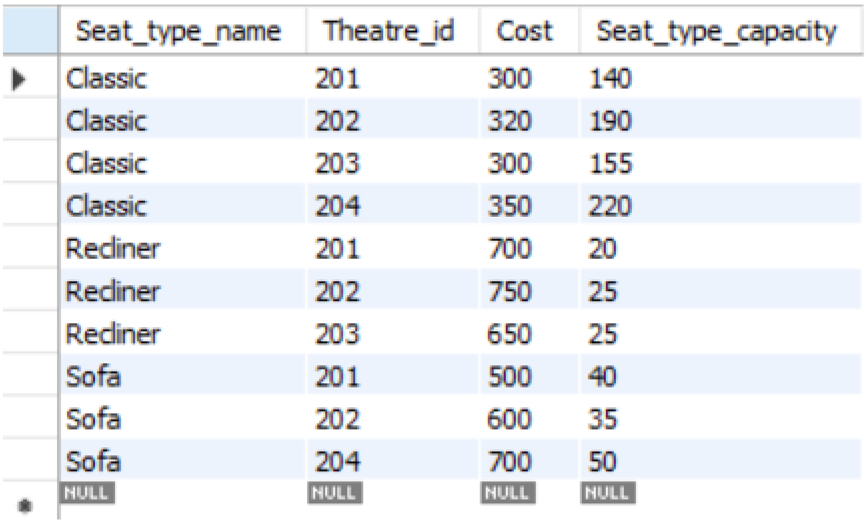
Movie_id	Movie_name	Genre	Releasing_date	Rating	Cast	Duration
101	Oppenheimer	Biography	2023-07-21	9.6	Cillian Murphy, Emily Blunt	03:00:00
102	Barbie	Comedy	2023-07-21	8.0	Margot Robbie, Ryan Gosling	01:54:00
103	Dune 2	Sci-Fi	2025-07-15	9.8	Timothée Chalamet, Zendaya	02:40:00
104	Interstellar	Sci-Fi	2014-11-07	9.7	Matthew McConaughey, Anne Hathaway	02:49:00
105	The Batman	Action	2022-03-04	8.5	Robert Pattinson, Zoë Kravitz	02:56:00
106	Inception	Thriller	2010-07-16	9.0	Leonardo DiCaprio, Tom Hardy	02:28:00
NULL	NULL	NULL	NULL	NULL	NULL	NULL



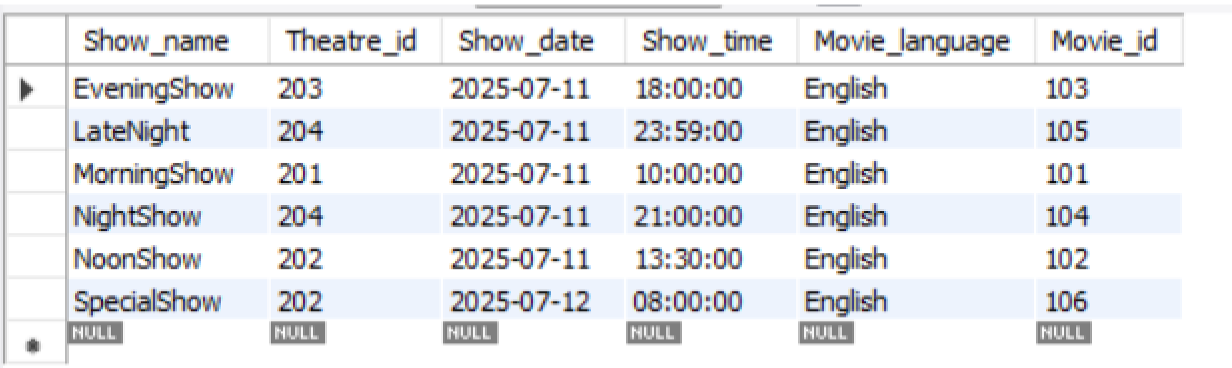
Payment_id	Price	Payment_mode	Payment_date	Customer_id
301	1500	UPI	2025-07-10	1
302	1800	Card	2025-07-10	2
303	1000	Cash	2025-07-10	3
304	2200	UPI	2025-07-10	4
305	1600	Card	2025-07-10	5
306	900	UPI	2025-07-11	6
NULL	NULL	NULL	NULL	NULL



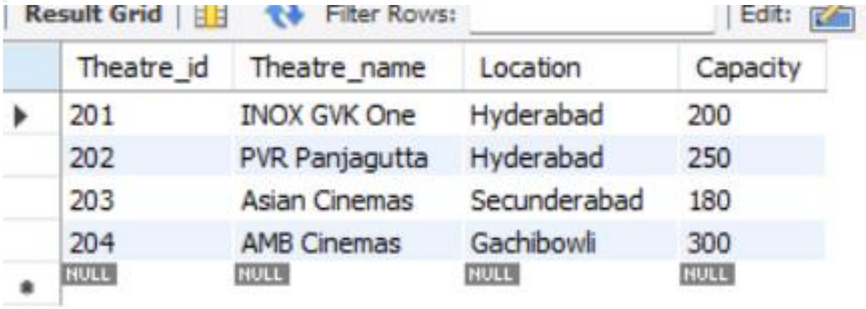
Ticket_id	Seat_number	Seat_type_name	Theatre_id
401	A1	Sofa	201
402	B2	Classic	202
403	C3	Recliner	203
404	D4	Sofa	204
405	D5	Classic	204
406	E1	Recliner	202
NULL	NULL	NULL	NULL



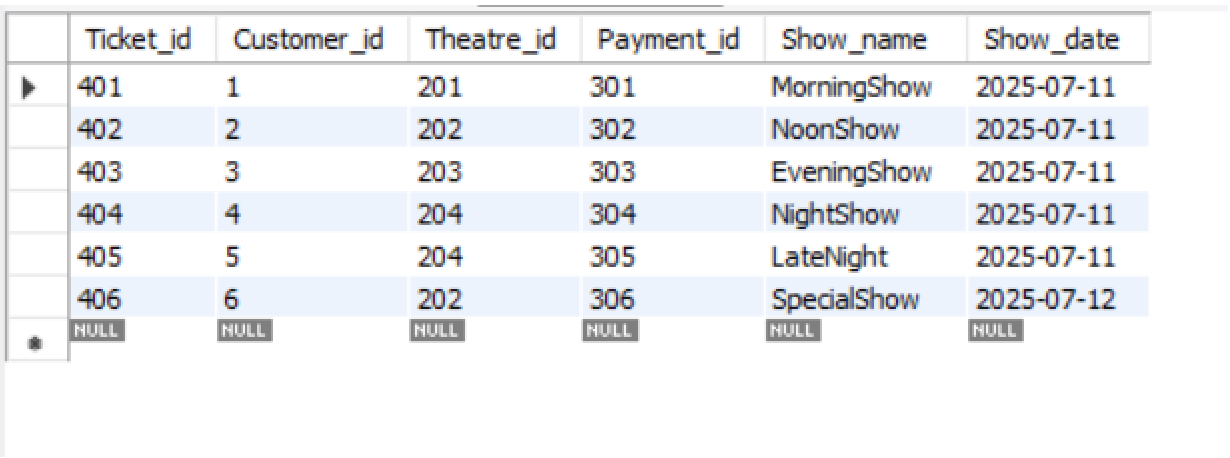
Seat_type_name	Theatre_id	Cost	Seat_type_capacity
Classic	201	300	140
Classic	202	320	190
Classic	203	300	155
Classic	204	350	220
Recliner	201	700	20
Recliner	202	750	25
Recliner	203	650	25
Sofa	201	500	40
Sofa	202	600	35
Sofa	204	700	50
NULL	NULL	NULL	NULL



Show_name	Theatre_id	Show_date	Show_time	Movie_language	Movie_id
EveningShow	203	2025-07-11	18:00:00	English	103
LateNight	204	2025-07-11	23:59:00	English	105
MorningShow	201	2025-07-11	10:00:00	English	101
NightShow	204	2025-07-11	21:00:00	English	104
NoonShow	202	2025-07-11	13:30:00	English	102
SpecialShow	202	2025-07-12	08:00:00	English	106
NULL	NULL	NULL	NULL	NULL	NULL



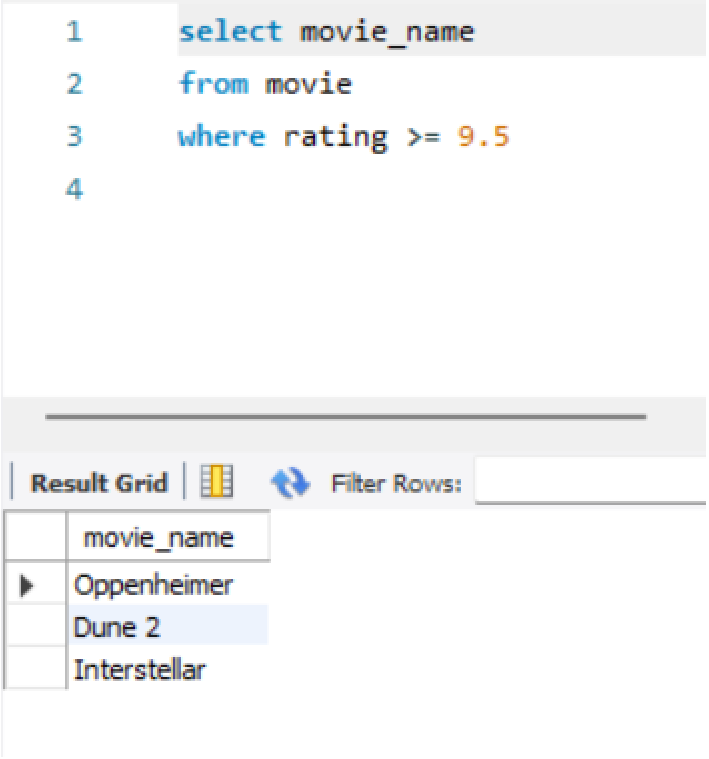
Theatre_id	Theatre_name	Location	Capacity
201	INOX GVK One	Hyderabad	200
202	PVR Panjagutta	Hyderabad	250
203	Asian Cinemas	Secunderabad	180
204	AMB Cinemas	Gachibowli	300
NULL	NULL	NULL	NULL



Ticket_id	Customer_id	Theatre_id	Payment_id	Show_name	Show_date
401	1	201	301	MorningShow	2025-07-11
402	2	202	302	NoonShow	2025-07-11
403	3	203	303	EveningShow	2025-07-11
404	4	204	304	NightShow	2025-07-11
405	5	204	305	LateNight	2025-07-11
406	6	202	306	SpecialShow	2025-07-12
NULL	NULL	NULL	NULL	NULL	NULL

QUERIES:

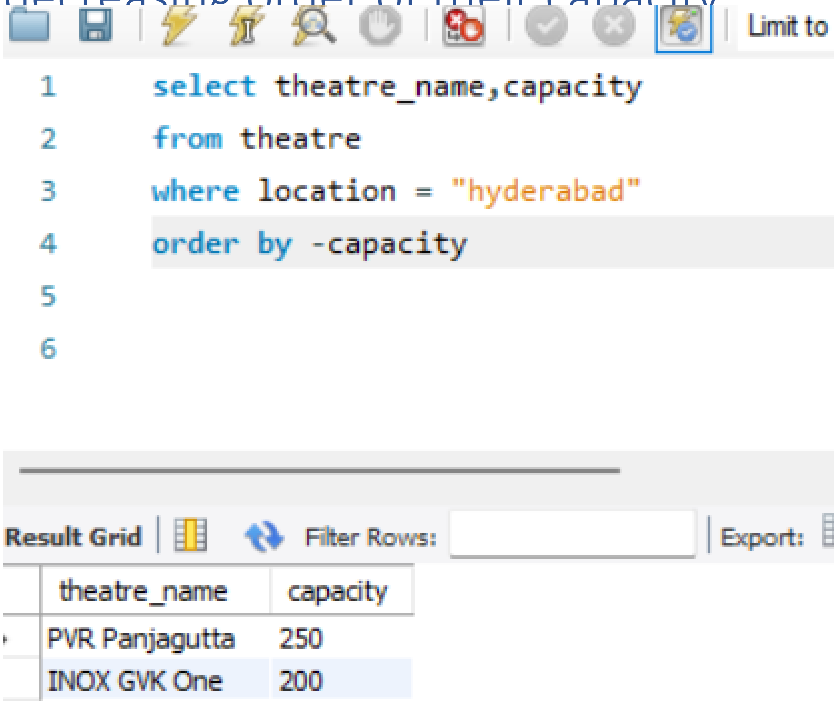
1.Display all movie names with rating greater than 9.5.



```sql
select movie_name
from movie
where rating >= 9.5
```

movie_name
Oppenheimer
Dune 2
Interstellar

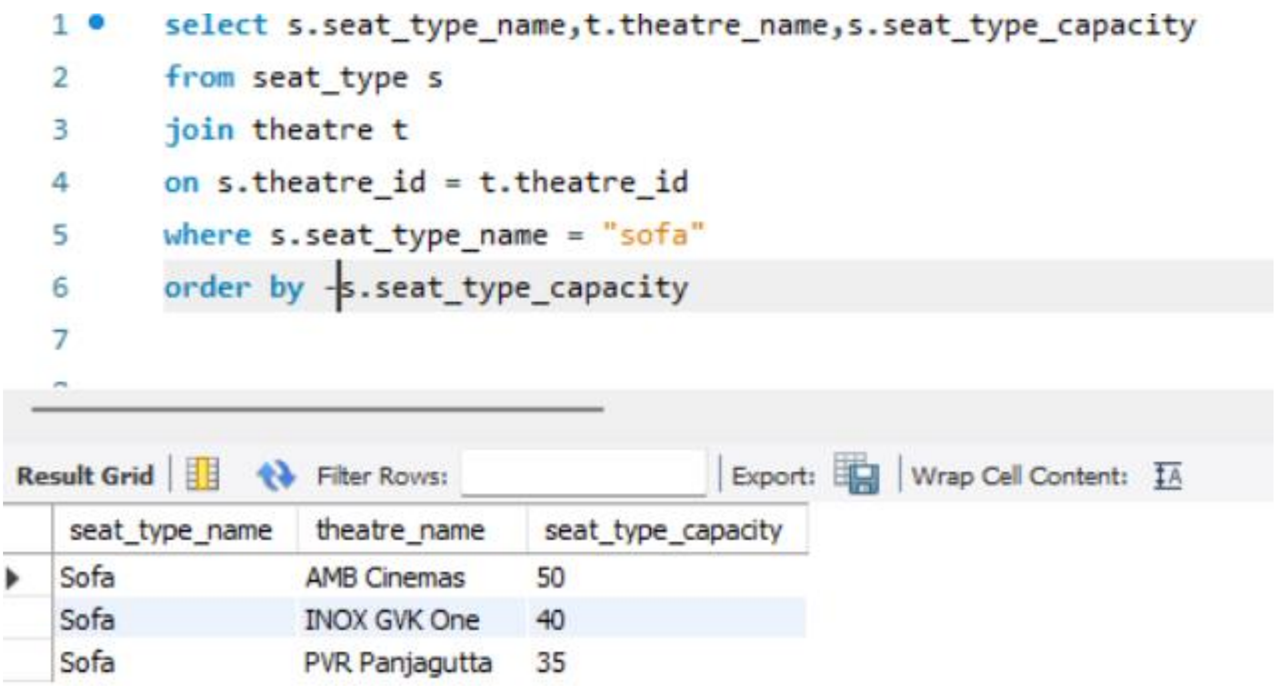
2.Display all theatre names of theatres located in Hyderabad in decreasing order of their capacity



```
select theatre_name,capacity
from theatre
where location = "hyderabad"
order by -capacity
```

theatre_name	capacity
PVR Panjagutta	250
INOX GVK One	200

3.Display all the theatre names who have ‘sofas’ in it ordered by number of sofas in each table and display number of sofas in each theatre.

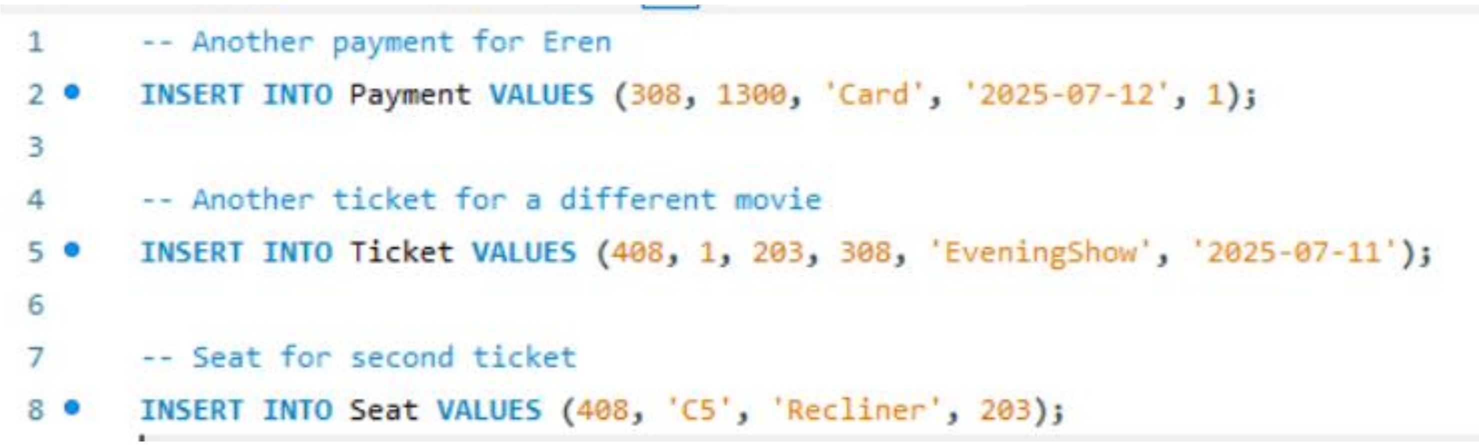


```sql
select s.seat_type_name,t.theatre_name,s.seat_type_capacity
from seat_type s
join theatre t
on s.theatre_id = t.theatre_id
where s.seat_type_name = "sofa"
order by -s.seat_type_capacity
```

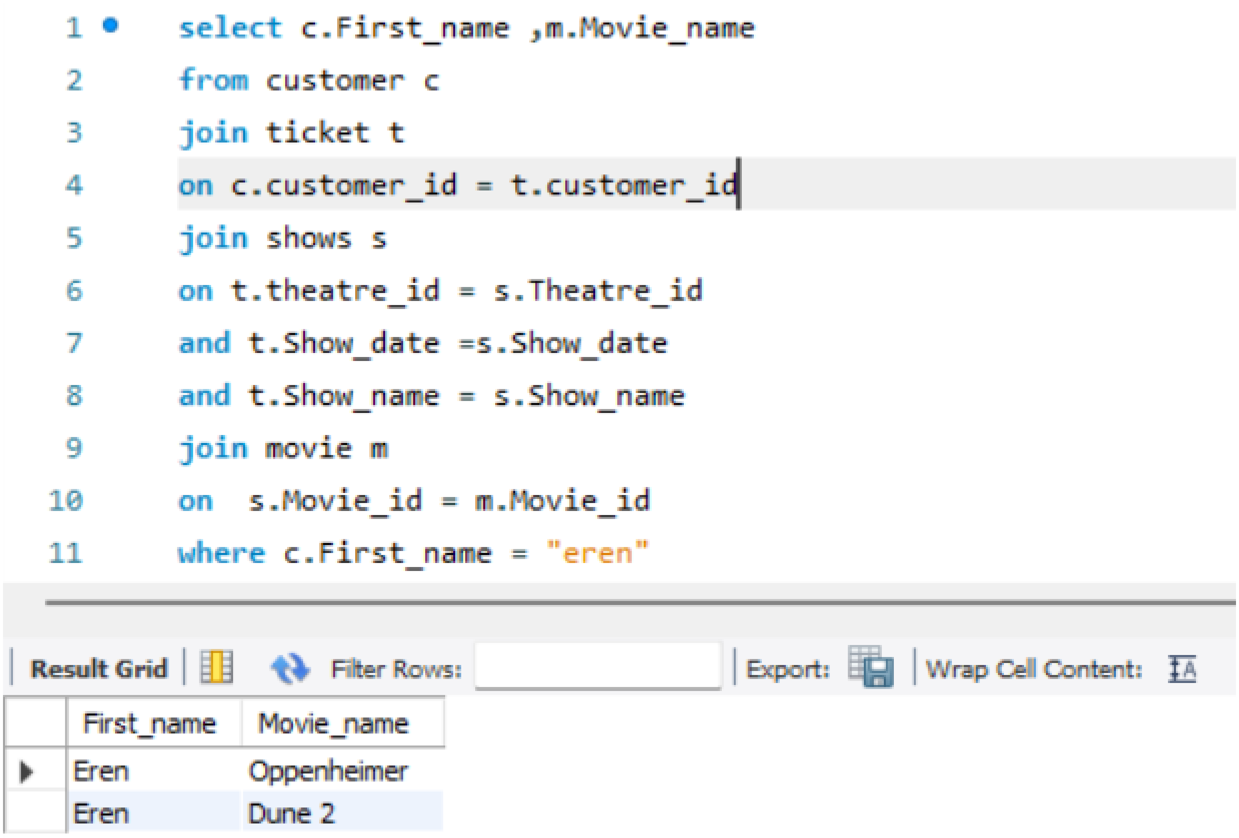
seat_type_name	theatre_name	seat_type_capacity
Sofa	AMB Cinemas	50
Sofa	INOX GVK One	40
Sofa	PVR Panjagutta	35

4. Name all the movies watched by customer with first name Eren?

Booking another ticket for Eren for better understanding



```sql
-- Another payment for Eren
INSERT INTO Payment VALUES (308, 1300, 'Card', '2025-07-12', 1);

-- Another ticket for a different movie
INSERT INTO Ticket VALUES (408, 1, 203, 308, 'EveningShow', '2025-07-11');

-- Seat for second ticket
INSERT INTO Seat VALUES (408, 'C5', 'Recliner', 203);
```



```sql
select c.First_name ,m.Movie_name
from customer c
join ticket t
on c.customer_id = t.customer_id
join shows s
on t.theatre_id = s.Theatre_id
and t.Show_date =s.Show_date
and t.Show_name = s.Show_name
join movie m
on  s.Movie_id = m.Movie_id
where c.First_name = "eren"
```

First_name	Movie_name
Eren	Oppenheimer
Eren	Dune 2