Lab Report

DBMS PROJECT

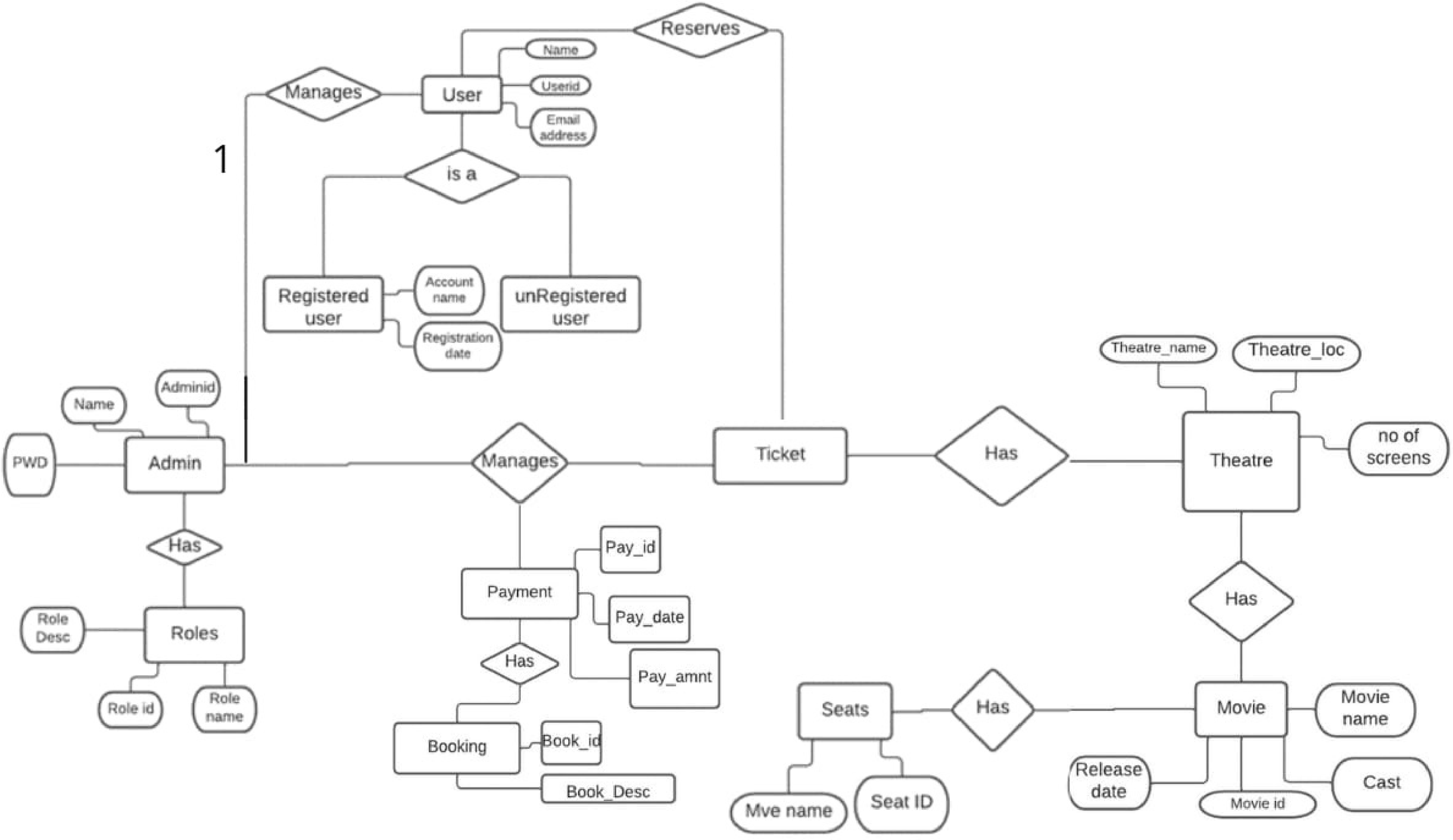
Movie Booking managment System

SELECT * FROM members;

202011037 Vishnu Swaroop
202011046 Nithish Kumar Raju
202011027 Inderjeet Singh
202011062 Kartheesha Ramancha
202011065 Sakshi Singh Dangi

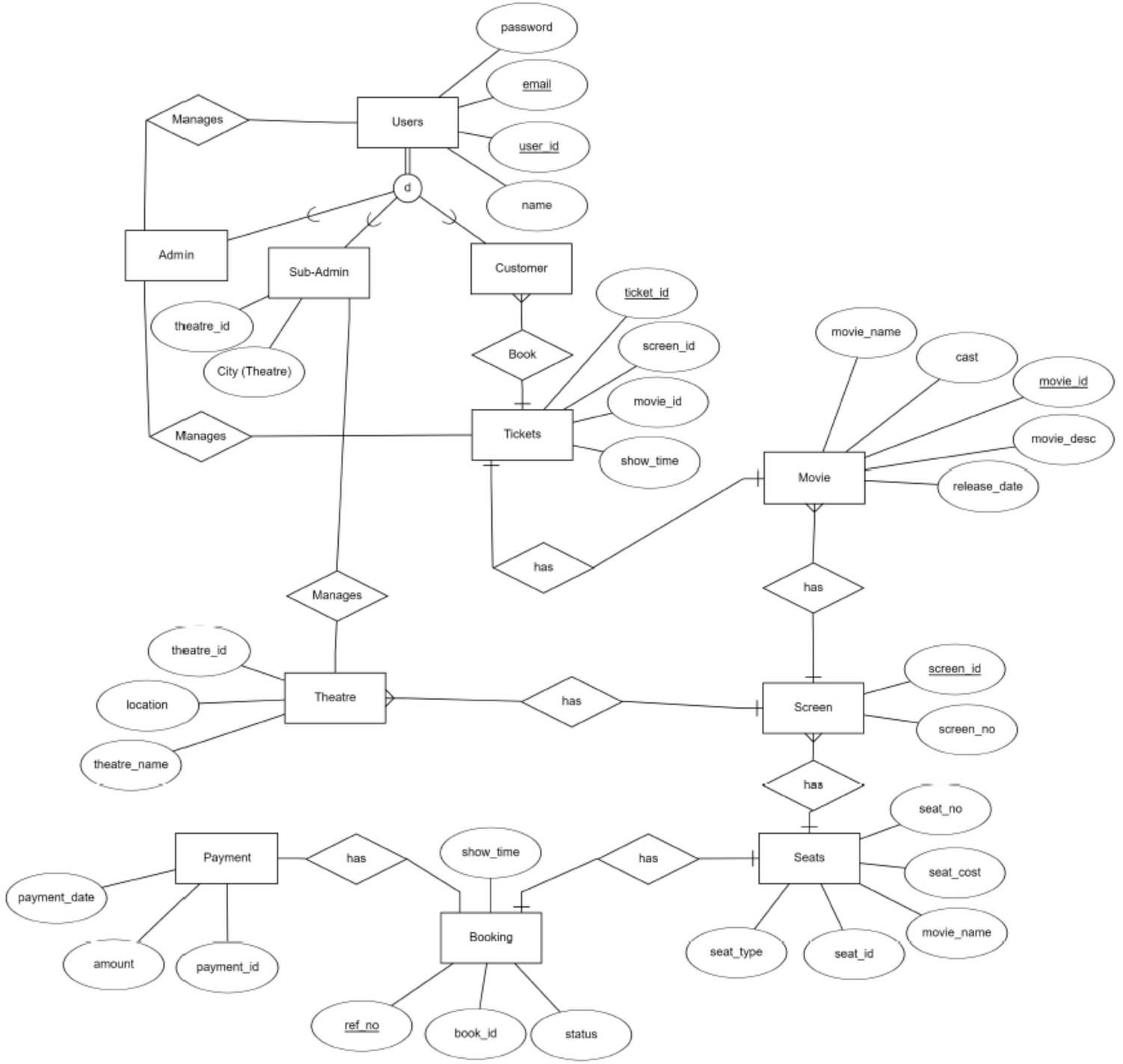
ER Model

Entity Relationship Model defines
the conceptual view of the database.


Reserves
Name
Userid
Email address
Manages
User
1
is a
Registered user
Account name
Registration date
unRegistered user
Theatre_name
Theatre_loc
Adminid
Name
PWD
Admin
Manages
Ticket
Has
Theatre
no of screens
Has
Pay_id
Payment
Pay_date
Pay_amnt
Has
Role Desc
Roles
Role id
Role name
Booking
Book_id
Book_Desc
Seats
Has
Movie
Movie name
Release date
Movie id
Cast
Mve name
Seat ID

EER Model

Extended Entity-Relationship Model adds more features to the ER Model. Basically, a high-level design for the database.

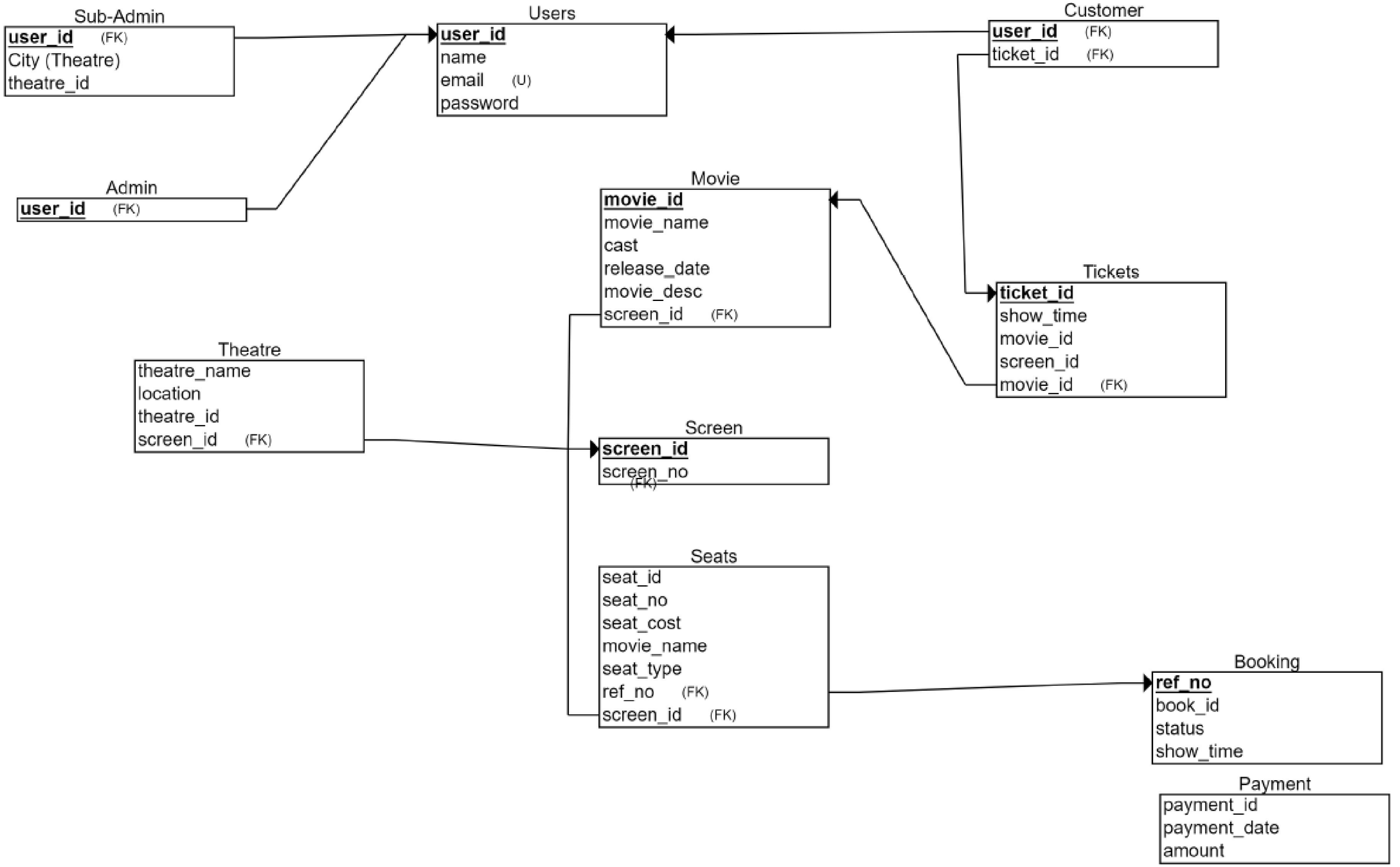

password
email
user_id
name
Users
Manages
d
Admin
Sub-Admin
Customer
theatre_id
City (Theatre)
ticket_id
screen_id
movie_id
show_time
Book
Tickets
Manages
movie_name
cast
movie_id
movie_desc
release_date
Movie
has
has
Manages
theatre_id
location
theatre_name
Theatre
has
screen_id
screen_no
Screen
has
seat_no
seat_cost
movie_name
seat_type
seat_id
Seats
has
show_time
has
Payment
has
payment_date
amount
payment_id
Booking
ref_no
book_id
status

Concepts Used

We Have Used disjoint specializationand Disjoint Concept in the Users Entity Because User can be admin ,sub admin and Customer and also we have used Disjoint because an individual of the User class may be a member of only one specialized subclass

Relational Model

Relational Model represents how data is stored in Relational Databases. A relational database stores data in the form of relations (tables).


Sub-Admin
user_id (FK)
City (Theatre)
theatre_id
Users
user_id
name
email (U)
password
Customer
user_id (FK)
ticket_id (FK)
Admin
user_id (FK)
Movie
movie_id
movie_name
cast
release_date
movie_desc
screen_id (FK)
Tickets
ticket_id
show_time
movie_id
screen_id
movie_id (FK)
Theatre
theatre_name
location
theatre_id
screen_id (FK)
Screen
screen_id
screen_no
(FK)
Seats
seat_id
seat_no
seat_cost
movie_name
seat_type
ref_no (FK)
screen_id (FK)
Booking
ref_no
book_id
status
show_time
Payment
payment_id
payment_date
amount

Relational Algebra Queries

Relational algebra is a procedural query language, which takes instances of relations as input and yields instances of relations as output. It uses operators to perform queries. An operator can be either unary or binary. They accept relations as their input and yield relations as their output

Database

SubAdmin = { user_Id, city, TheatreId
'01', 'Hyderabad', '121'
'02', 'Diu', '122'
'03', 'Sagar', '123'
'04', 'Hyderabad', '124'
}
Users={ userID, Name, Email, Password
'01', 'Tom', '62diu.in', 20
'02', 'Jerry', '27diu.in', 34
'03', 'Putin', '65diu.in', 35
'05', 'Trump', '37diu.in', 20
}
Movie = { MovieName, cast, movie_desc, screen_Id , releaseDate ,movieId , theatre_Id
'Harry Potter', 'Emma watson', 'novels by J. K. Rowling', 'abc' ,'2021-2-23','001', '120'
'Twilight', 'Robert Pattinson', 'vampires', '002' ,'2022-1-2','234', '122'
'Mr.Bean', 'Robert', 'Humor', '003' ,'2026-11-27','456', '121'
'TVD', 'pattinson', 'Thriller', '004' ,'2024-11-27','456','123'
}

Database

Tickets = { Showtime, movieID, screen_id, ticketId
'11:20' , '001' , 'abc', 003
}

Admin = { user_id , City
'21' ,'Guntur'
'22' ,'Hyderabad'
'23' , 'mumbai'
'24' , 'sagar'
}

Theatre ={ theatreName, Location, theatre_id, screen_id
'ABC' , 'Hyderabad' ,'121','abc'
'XYZ' ,'Diu','122','001'
}

Screen = { screenID , Screen_no , Movie_Name , theatre_id, no_of_seats
'abc','2','Bahubali' , '121' , '200'
'001','3','TVD','123', '50'
}

Seats = { seatNo ,seatID, seatCost,Movie_name,seatType,refNo,screen_Id
'21','234','400','Harry Potter','Platinum','345','001'
}

Database

Booking ={ref_No,booking_id,status,show_time
'345','@etet','Available','11:20'
'234','@ryrey','Available','10:00'
}
Customer ={ useRid,ticket_ID
'01',003
'02' ,003
}
Payment = { pay_id , pay_date , pay_amount , booking_ID ,status_pay,user_id

'202' , '2020-01-23' , '1200', '@etet' , 'Completed','01'
'203' , '2020-01-24' , '200', '@ryrey' , 'Completed', '02'
'204' , '2020-01-25' , '2000' , '@ueru' , 'failed', '03'
}

Queries

For Checking the status of Movie

π Movie_name,seatNo,status (Seats ⋈ refNo = ref_No Booking)

"To Check Whether the screen is present in the theatre"

π screenID , Screen_no (Theatre ⋈ screen_id = screenID Screen)

To listout the movies present in the theatre

π MovieName, cast, movie_desc , Location (Theatre ⋈ theatre_id = theatre_Id Movie)

To check which movie is present in which screen

π Screen_no,screenID,MovieName (Movie ⋈ MovieName = Movie_Name Screen)

Queries

To check payment details

π pay_amount,Name,userID,status_pay (Users ⋈ userID = user_id Payment)

to print the admins present in hyderabad

π user_id (σ City = 'Hyderabad' σ city = City (SubAdmin × Admin))

to print the moviename,tickets available

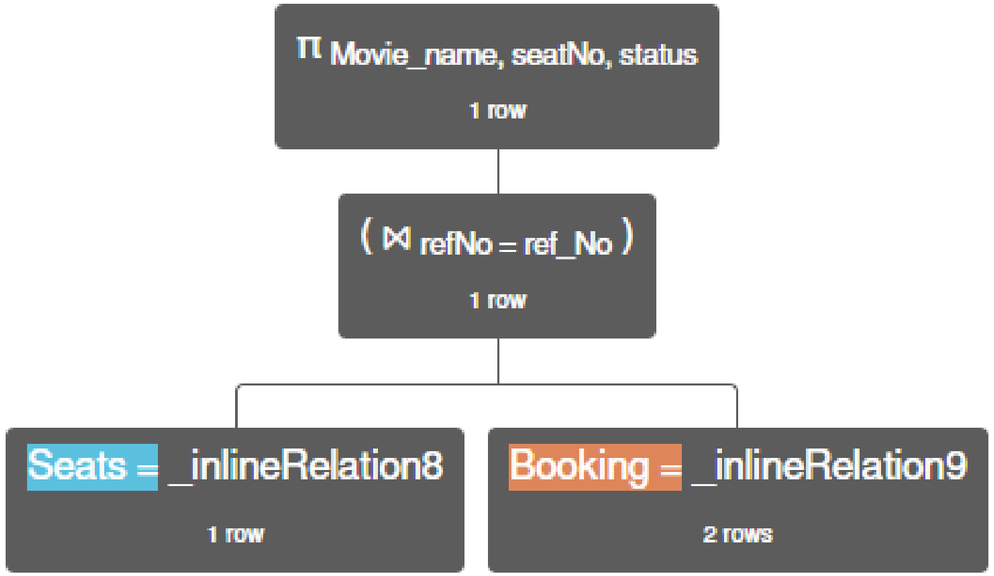
π Movie_Name ,no_of_seats (Screen)

Running A Query of Relational Algebra in RelaX

```
56
57 π Movie_name,seatNo,status ( Seats ⋈  refNo = ref_No Booking   )
```

execute query

download history



π Movie_name, seatNo, status (_inlineRelation8 ⋈ refNo = ref_No _inlineRelation9)

Movie_name	seatNo	status
'Harry Potter'	'21'	'Available'

SQL

SQL is Structured Query Language, which is a computer language for storing, manipulating and retrieving data stored in a relational database. SQL is the standard language for Relational Database System

Database

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| movie_booking      |
| mysql              |
| performance_schema |
| phpmyadmin         |
| test               |
+--------------------+
6 rows in set (0.00 sec)
```

Database

```
mysql> use movie_booking;
Database changed
mysql> show tables;
+--------------------------+
| Tables_in_movie_booking  |
+--------------------------+
| admin                    |
| booking                  |
| customer                 |
| movie                    |
| payment                  |
| screen                   |
| seats                    |
| subadmin                 |
| theatre                  |
| tickets                  |
| users                    |
+--------------------------+
11 rows in set (0.00 sec)
```

Database

```
mysql> desc admin;
+--------+--------------+------+-----+---------+--------+
| Field  | Type         | Null | Key | Default | Extra  |
+--------+--------------+------+-----+---------+--------+
| userId | int(2)       | NO   | PRI | NULL    |        |
| city   | varchar(100) | YES  |     | NULL    |        |
+--------+--------------+------+-----+---------+--------+
2 rows in set (0.02 sec)
```

Database

```
mysql> desc customer;
+------------+----------+------+-----+----------+--------+
| Field      | Type     | Null | Key | Default | Extra |
+------------+----------+------+-----+----------+--------+
| userId     | int(11) | YES  | MUL | NULL     |        |
| ticket_id | int(11) | YES  | MUL | NULL     |        |
+------------+----------+------+-----+----------+--------+
2 rows in set (0.02 sec)
```

Database

```
mysql> desc booking;
+-------------+--------------+------+-----+----------+--------+
| Field       | Type         | Null | Key | Default | Extra |
+-------------+--------------+------+-----+----------+--------+
| ref_number | int(11)      | YES  | MUL | NULL     |        |
| booking_id | varchar(20) | NO   | PRI | NULL     |        |
| status      | tinyint(1)   | YES  |     | NULL     |        |
| showtime    | time         | YES  |     | NULL     |        |
+-------------+--------------+------+-----+----------+--------+
4 rows in set (0.02 sec)
```

Database

```
mysql> desc movie;
+-------------+---------------+------+-----+---------+-------+
| Field       | Type          | Null | Key | Default | Extra |
+-------------+---------------+------+-----+---------+-------+
| moviename   | varchar(100)  | YES  |     | NULL    |       |
| cast        | varchar(100)  | YES  |     | NULL    |       |
| moviedesc   | varchar(1000) | YES  |     | NULL    |       |
| screenid    | int(3)        | YES  | MUL | NULL    |       |
| releasedate | varchar(100)  | YES  |     | NULL    |       |
| movieid     | int(3)        | NO   | PRI | NULL    |       |
| theatreid   | varchar(4)    | YES  |     | NULL    |       |
+-------------+---------------+------+-----+---------+-------+
7 rows in set (0.02 sec)
```

Database

```
mysql> desc payment;
+----------------+--------------+------+-----+---------+-------+
| Field          | Type         | Null | Key | Default | Extra |
+----------------+--------------+------+-----+---------+-------+
| payment_id     | int(3)       | NO   | PRI | NULL    |       |
| pay_date       | date         | YES  |     | NULL    |       |
| amount         | int(11)      | YES  |     | NULL    |       |
| booking_id     | varchar(20)  | YES  | MUL | NULL    |       |
| payment_status | tinyint(1)   | YES  |     | NULL    |       |
| userId         | int(11)      | YES  | MUL | NULL    |       |
+----------------+--------------+------+-----+---------+-------+
6 rows in set (0.01 sec)
```

Database

```
mysql> desc seats;
+-------------+--------------+------+-----+---------+-------+
| Field       | Type         | Null | Key | Default | Extra |
+-------------+--------------+------+-----+---------+-------+
| seat_id     | int(4)       | NO   | PRI | NULL    |       |
| seat_cost   | int(4)       | YES  |     | NULL    |       |
| movie_name  | varchar(100) | YES  |     | NULL    |       |
| seat_type   | varchar(100) | YES  |     | NULL    |       |
| ref_num     | int(3)       | YES  | MUL | NULL    |       |
| screen_id   | int(3)       | YES  | MUL | NULL    |       |
+-------------+--------------+------+-----+---------+-------+
6 rows in set (0.01 sec)
```

Database

```
mysql> select * from seats;
+---------+-----------+--------------+-----------+---------+------------+
| seat_id | seat_cost | movie_name   | seat_type | ref_num | screen_id |
+---------+-----------+--------------+-----------+---------+------------+
|     234 |       400 | Harry Potter | Platinum  |     202 |          1 |
+---------+-----------+--------------+-----------+---------+------------+
1 row in set (0.00 sec)

mysql> select * from users;
+--------+------------+-------------------------------+-------------+
| userId | name       | email                         | password    |
+--------+------------+-------------------------------+-------------+
|      1 | vishnu     | vishnu@apnatheatre.com        | 12345678    |
|      2 | sakshi     | sakshi@apnatheatre.com        | hellohello  |
|      3 | nithish    | nithish@apnatheatre.com       | helloworld  |
|      4 | inder      | inder@apnatheatre.com         | ummhmm      |
|      5 | kartheesha | kartheesha@apnatheatre.com    | achaaaa     |
+--------+------------+-------------------------------+-------------+
5 rows in set (0.01 sec)
```

Database

```
mysql> select * from payment;
+------------+------------+--------+-------------+----------------+---------+
| payment_id | pay_date   | amount | booking_id  | payment_status | userId |
+------------+------------+--------+-------------+----------------+---------+
|        202 | 2020-01-23 |   1200 | @encodedguy |              1 |       4 |
|        203 | 2020-01-23 |    600 | @vishnu     |              0 |       1 |
+------------+------------+--------+-------------+----------------+---------+
2 rows in set (0.00 sec)

mysql> select * from customer;
+--------+-----------+
| userId | ticket_id |
+--------+-----------+
|      1 |         2 |
|      3 |         2 |
+--------+-----------+
2 rows in set (0.01 sec)
```

Queries

```
mysql> select * from movie, screen where movie.screenid = screen.screen_id;
+--------------+------------------+-----------------------+----------+-------------+---------+-----------+-----------+------------+------------+------------+-------------+
| moviename    | cast             | moviedesc             | screenid | releasedate | movieid | theatreid | screen_id | screen_num | movie_name | theatre_id | no_of_seats |
+--------------+------------------+-----------------------+----------+-------------+---------+-----------+-----------+------------+------------+------------+-------------+
| Harry Potter | Emma Watson      | Novels by J.K Rowling |        1 | 2021-2-23   |       1 | 120       |         1 |          2 | Bahubali   |        121 |         200 |
| Twilight     | Robert Pattinson | Vampires              |        2 | 2022-1-2    |     234 | 122       |         2 |          3 | TVD        |        122 |         400 |
| TVD          | Pattinson        | Thriller              |        2 | 2024-11-27  |     348 | 123       |         2 |          3 | TVD        |        122 |         400 |
| Mr. Bean     | Robert           | Humor                 |        2 | 2026-11-27  |     456 | 121       |         2 |          3 | TVD        |        122 |         400 |
+--------------+------------------+-----------------------+----------+-------------+---------+-----------+-----------+------------+------------+------------+-------------+
4 rows in set (0.01 sec)
```

Queries

```
mysql> select * from users, customer where users.userId = customer.userId;
+--------+----------+-------------------------+------------+--------+-----------+
| userId | name     | email                   | password   | userId | ticket_id |
+--------+----------+-------------------------+------------+--------+-----------+
|      1 | vishnu   | vishnu@apnatheatre.com  | 12345678   |      1 |         2 |
|      3 | nithish  | nithish@apnatheatre.com | helloworld |      3 |         2 |
+--------+----------+-------------------------+------------+--------+-----------+
2 rows in set (0.00 sec)
```

Queries

```
mysql> select * from booking, payment where booking.booking_id = payment.booking_id;
+------------+-------------+--------+----------+------------+------------+--------+-------------+----------------+--------+
| ref_number | booking_id  | status | showtime | payment_id | pay_date   | amount | booking_id  | payment_status | userId |
+------------+-------------+--------+----------+------------+------------+--------+-------------+----------------+--------+
|        202 | @encodedguy |      1 | 11:20:00 |        202 | 2020-01-23 |   1200 | @encodedguy |              1 |      4 |
|        203 | @vishnu     |      1 | 10:00:00 |        203 | 2020-01-23 |    600 | @vishnu     |              0 |      1 |
+------------+-------------+--------+----------+------------+------------+--------+-------------+----------------+--------+
2 rows in set (0.01 sec)
```

Queries

```
mysql> select * from theatre, subadmin where subadmin.theatreId = theatre.theatre_id;
+--------------+-----------+------------+-----------+--------+-----------+------------+
| theatre_name | location  | theatre_id | screen_id | userId | city      | theatreId |
+--------------+-----------+------------+-----------+--------+-----------+------------+
| ABC          | Hyderabad |        121 |         2 |      1 | hyderabad |        121 |
| XYZ          | Diu       |        122 |         1 |      2 | sagar     |        122 |
| XYZ          | Diu       |        122 |         1 |      3 | diu       |        122 |
| ABC          | Hyderabad |        121 |         2 |      4 | ghaziabad |        121 |
| ABC          | Hyderabad |        121 |         2 |      5 | telangana |        121 |
+--------------+-----------+------------+-----------+--------+-----------+------------+
5 rows in set (0.00 sec)
```

Queries

```
mysql> select * from users where userId > 3;
+--------+-----------+-----------------------------+----------+
| userId | name      | email                       | password |
+--------+-----------+-----------------------------+----------+
|      4 | inder     | inder@apnatheatre.com       | ummhmm   |
|      5 | kartheesha | kartheesha@apnatheatre.com | achaaaa  |
+--------+-----------+-----------------------------+----------+
2 rows in set (0.00 sec)

mysql> select * from users where userId > 3 OR 1=1;
+--------+-----------+-----------------------------+------------+
| userId | name      | email                       | password   |
+--------+-----------+-----------------------------+------------+
|      1 | vishnu    | vishnu@apnatheatre.com      | 12345678   |
|      2 | sakshi    | sakshi@apnatheatre.com      | hellohello |
|      3 | nithish   | nithish@apnatheatre.com     | helloworld |
|      4 | inder     | inder@apnatheatre.com       | ummhmm     |
|      5 | kartheesha | kartheesha@apnatheatre.com | achaaaa    |
+--------+-----------+-----------------------------+------------+
5 rows in set (0.00 sec)
```

Thank You!