Name- Sanket Anand

DATABASE MANAGEMENT SYSTEM PROJECT

MOVIE TICKET BOOKING SYSTEM

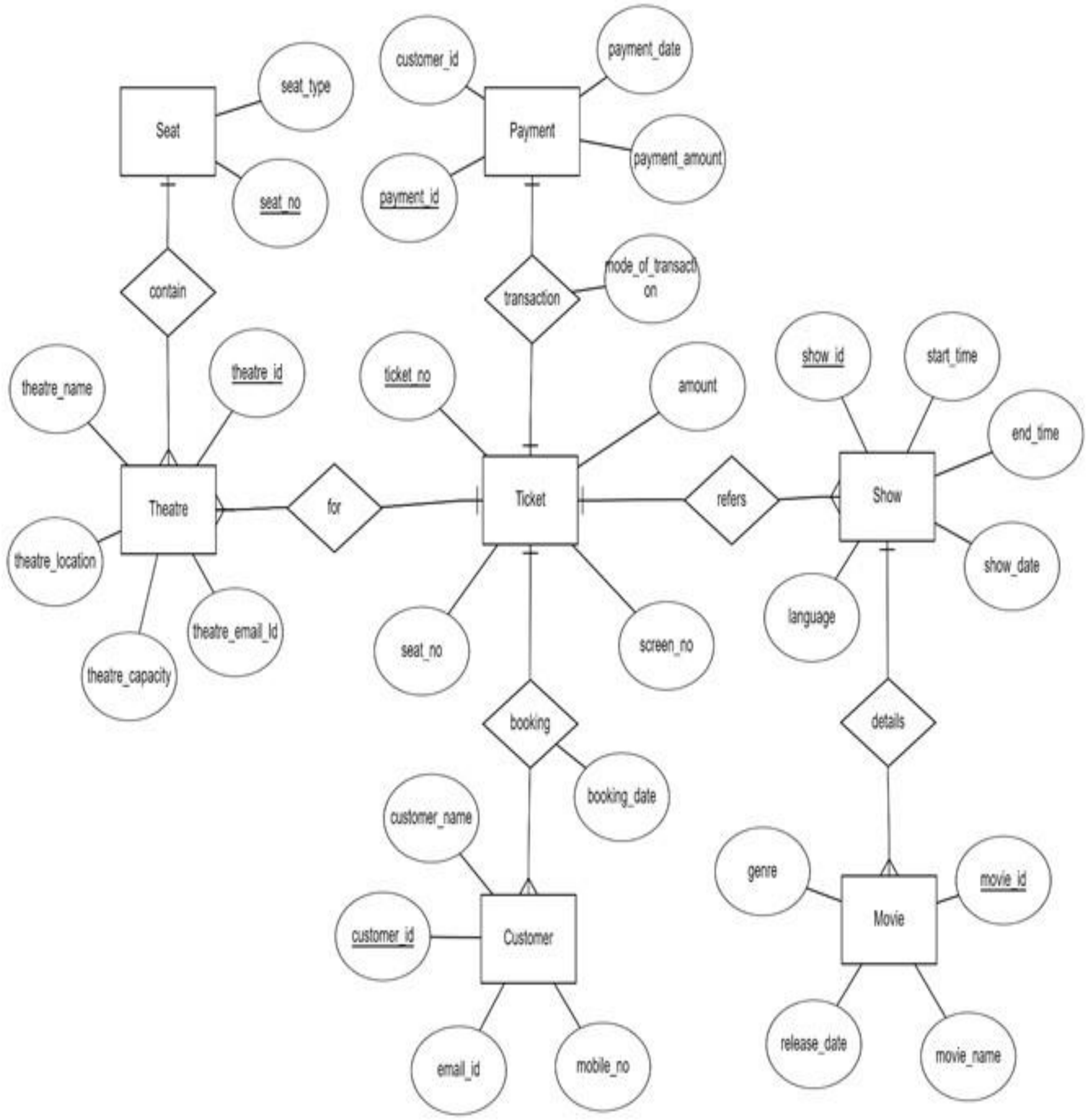
PROBLEM STATEMENT: -

In this project, we have designed a database to store information about the Movie-Ticket booking. The database will contain information about the customers and will be accessible to only the database administrator.

This database will contain the details of the customers, movies, price of tickets, reserved seats , payment options available , capacity of theatre and type of seats available in a theatre etc.

Contents

ER DIAGRAM: -


Seat
seat_type
seat_no
contain
Payment
customer_id
payment_date
payment_amount
payment_id
transaction
mode_of_transacti
on
Theatre
theatre_name
theatre_id
theatre_location
theatre_capacity
theatre_email_Id
for
Ticket
ticket_no
amount
seat_no
screen_no
refers
Show
show_id
start_time
end_time
show_date
language
booking
booking_date
details
Customer
customer_name
customer_id
email_id
mobile_no
Movie
genre
movie_id
release_date
movie_name

Assumptions

1. Ticket

This entity holds complete information about tickets such as amount, seat number, customer id, screen number , etc. Its primary key is Ticket_no and it contains Show_id, Customer_id, Theatre_id, Seat_no as foreign keys from Customer, Show, Seat, Theatre entities.

2. Customer

Customer entity holds information about the customer who bought the ticket such as customer id, Customer name, email id and mobile number. Its primary key is Customer id.

3. Show

This entity holds information about the show of the movie whose ticket has been purchased by the customer. Information like show id, start & end time of movie, language of the movie and movie id(which is foreign key taking reference from Movie entity) is stored in this entity. Its primary key is Show_id.

4. Movie

It holds information about the movie which will be watched by the customer. Details such as Movie id, name of the movie, its genre, etc are stored in this entity. Primary key of this entity is Movie_id.

5. Payment

This entity holds info about the payment done by the customer to buy the movie ticket. It stores Payment id, payment amount, mode of transaction (cash, card, upi, etc.), ticket number (foreign key from Ticket entity), customer id (foreign key from Customer entity). Primary key of this entity is Payment_id.

6. Theatre

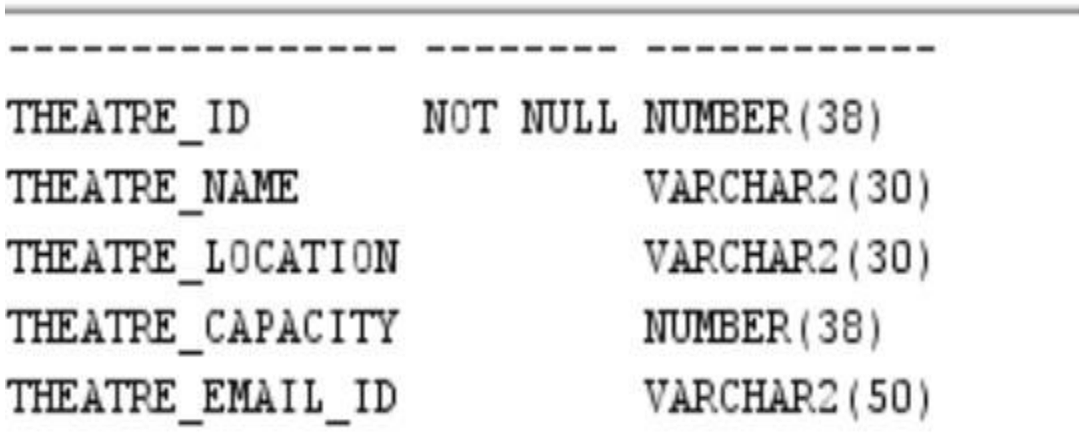
This entity stores information about the theatre details where the movie is being screened. Theatre name, theatre id, theatre location, its capacity and email id is stored in this entity. Primary key is Theatre_id.

7. Seat

This entity holds information about the seat of the theatre which has been allocated to the customer to watch the movie. It stores info such as Seat number , Seat type(recliner, deluxe, etc.) and theatre id (foreign key from Theatre entity). Primary key of this entity is Seat_no.

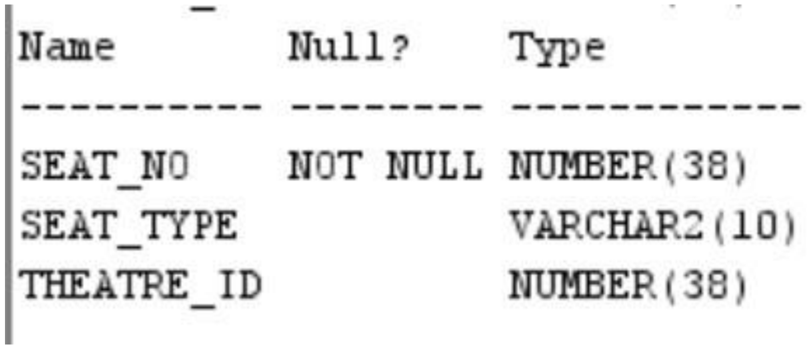
Database schema

1. Theatre



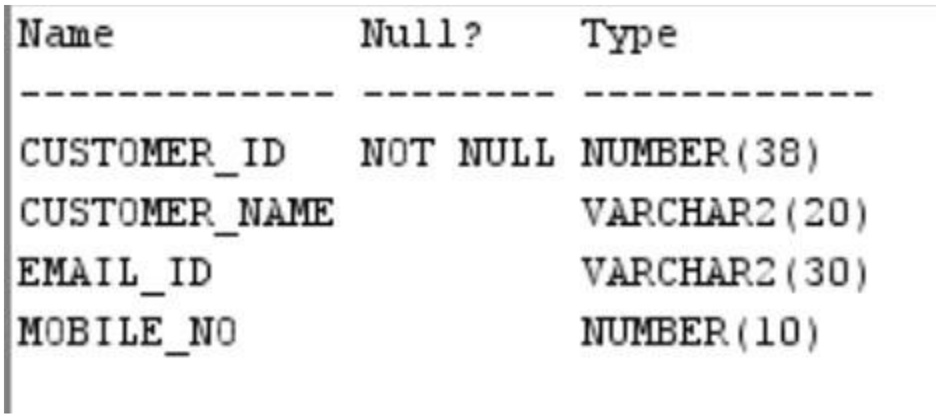
```
--------------- -------- -----------
THEATRE_ID        NOT NULL NUMBER(38)
THEATRE_NAME               VARCHAR2(30)
THEATRE_LOCATION           VARCHAR2(30)
THEATRE_CAPACITY           NUMBER(38)
THEATRE_EMAIL_ID           VARCHAR2(50)
```

2. Seat



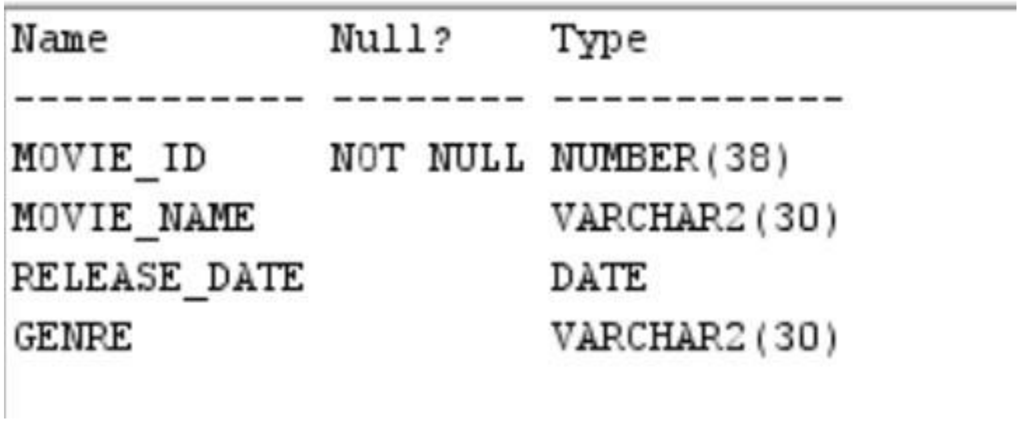
```
Name       Null?    Type
---------- -------- -----------
SEAT_NO    NOT NULL NUMBER(38)
SEAT_TYPE           VARCHAR2(10)
THEATRE_ID          NUMBER(38)
```

3. Customer



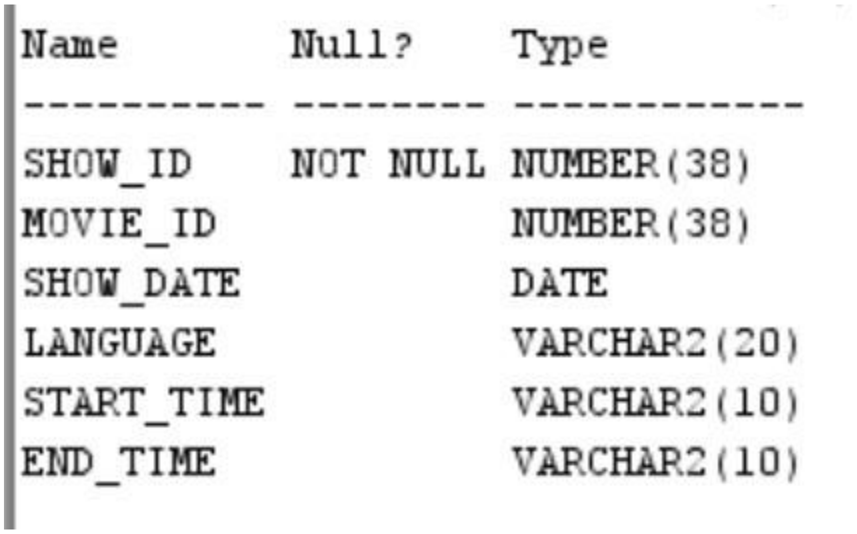
```
Name          Null?    Type
------------- -------- ------------
CUSTOMER_ID   NOT NULL NUMBER(38)
CUSTOMER_NAME          VARCHAR2(20)
EMAIL_ID               VARCHAR2(30)
MOBILE_NO              NUMBER(10)
```

4. Movie



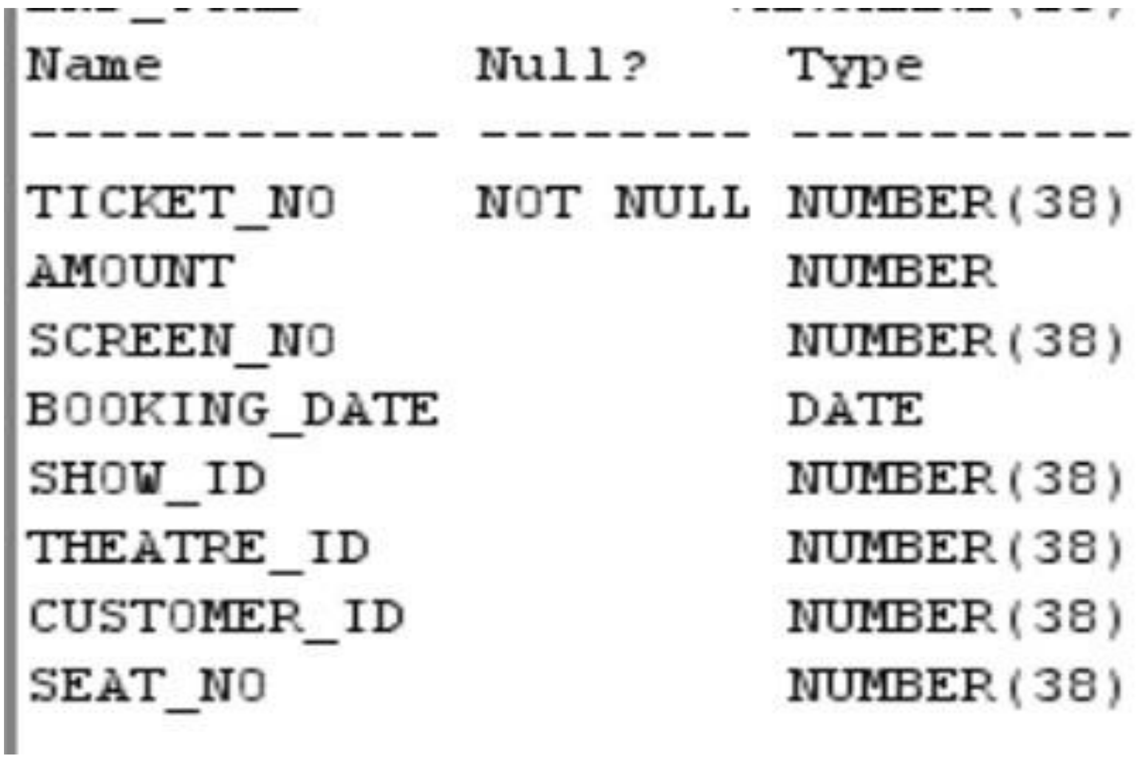
```
Name         Null?    Type
------------ -------- ------------
MOVIE_ID     NOT NULL NUMBER(38)
MOVIE_NAME            VARCHAR2(30)
RELEASE_DATE          DATE
GENRE                 VARCHAR2(30)
```

5. Show



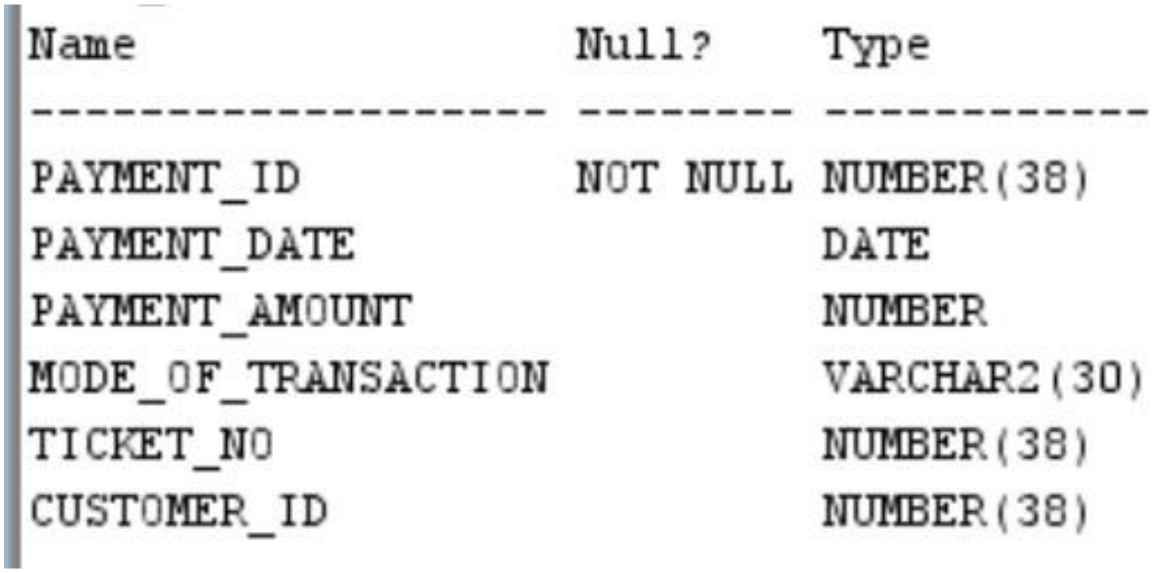
```
Name       Null?    Type
---------- -------- ------------
SHOW_ID    NOT NULL NUMBER(38)
MOVIE_ID            NUMBER(38)
SHOW_DATE           DATE
LANGUAGE            VARCHAR2(20)
START_TIME          VARCHAR2(10)
END_TIME            VARCHAR2(10)
```

6. Ticket



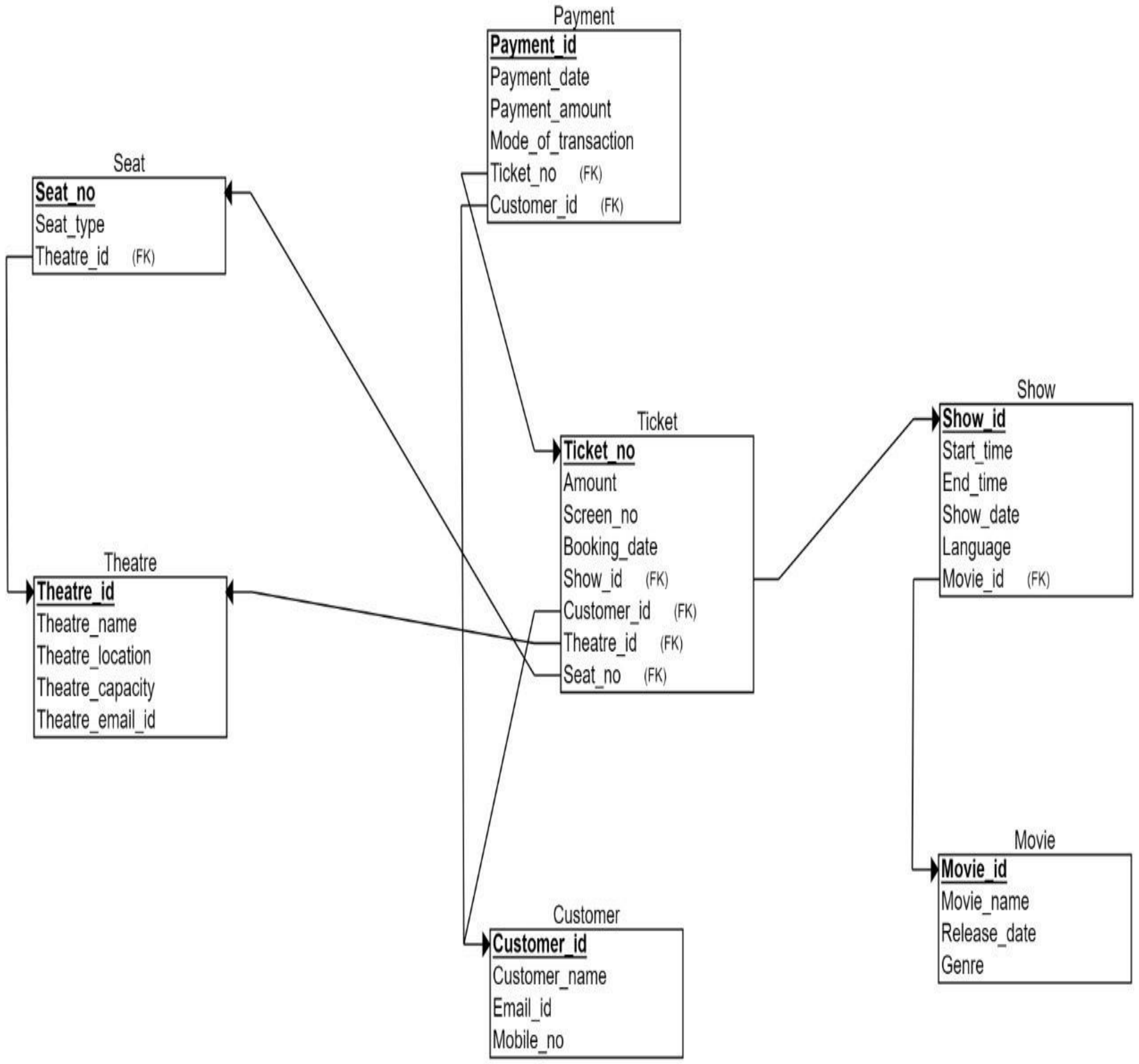
```
Name          Null?     Type
------------- --------- ----------
TICKET_NO     NOT NULL  NUMBER(38)
AMOUNT                  NUMBER
SCREEN_NO               NUMBER(38)
BOOKING_DATE            DATE
SHOW_ID                 NUMBER(38)
THEATRE_ID              NUMBER(38)
CUSTOMER_ID             NUMBER(38)
SEAT_NO                 NUMBER(38)
```

7. Payment



```
Name                 Null?    Type
-------------------- -------- ------------
PAYMENT_ID           NOT NULL NUMBER(38)
PAYMENT_DATE                  DATE
PAYMENT_AMOUNT                NUMBER
MODE_OF_TRANSACTION           VARCHAR2(30)
TICKET_NO                     NUMBER(38)
CUSTOMER_ID                   NUMBER(38)
```

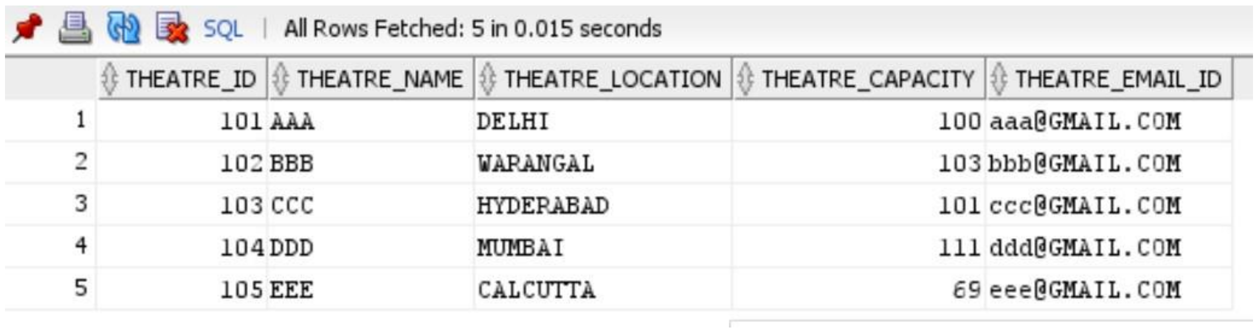
Relational Schema:-



Creation and Insertion of Tables

1.Theatre

```
CREATE TABLE THEATRE(THEATRE_ID INT PRIMARY KEY,
            THEATRE_NAME VARCHAR(30),
            THEATRE_LOCATION VARCHAR(30),
            THEATRE_CAPACITY INT,
            THEATRE_EMAIL_ID VARCHAR(50));

INSERT INTO THEATRE VALUES(101,'AAA','DELHI',100,'aaa@GMAIL.COM');

INSERT INTO THEATRE VALUES(102,'BBB','WARANGAL',103,'bbb@GMAIL.COM');

INSERT INTO THEATRE VALUES(103,'CCC','HYDERABAD',101,'ccc@GMAIL.COM');

INSERT INTO THEATRE VALUES(104,'DDD','MUMBAI',111,'ddd@GMAIL.COM');

INSERT INTO THEATRE VALUES(105,'EEE','CALCUTTA',69,'eee@GMAIL.COM');
```

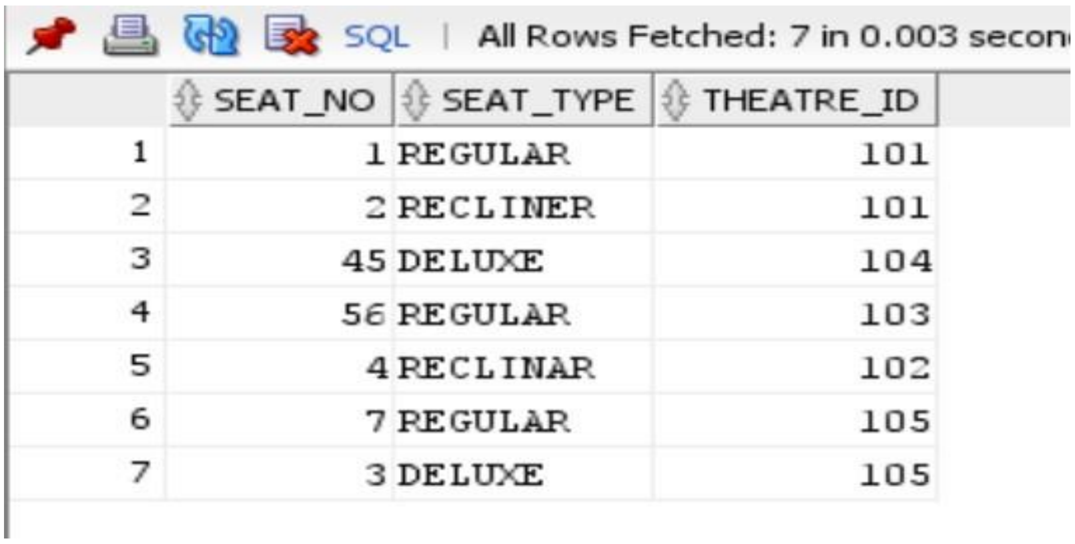


	THEATRE_ID	THEATRE_NAME	THEATRE_LOCATION	THEATRE_CAPACITY	THEATRE_EMAIL_ID
1	101	AAA	DELHI	100	aaa@GMAIL.COM
2	102	BBB	WARANGAL	103	bbb@GMAIL.COM
3	103	CCC	HYDERABAD	101	ccc@GMAIL.COM
4	104	DDD	MUMBAI	111	ddd@GMAIL.COM
5	105	EEE	CALCUTTA	69	eee@GMAIL.COM

2. Seat

```
CREATE TABLE SEAT(SEAT_NO INT PRIMARY KEY,
          SEAT_TYPE VARCHAR(10),
          THEATRE_ID INT,
          FOREIGN KEY (THEATRE_ID) REFERENCES THEATRE(THEATRE_ID));
```

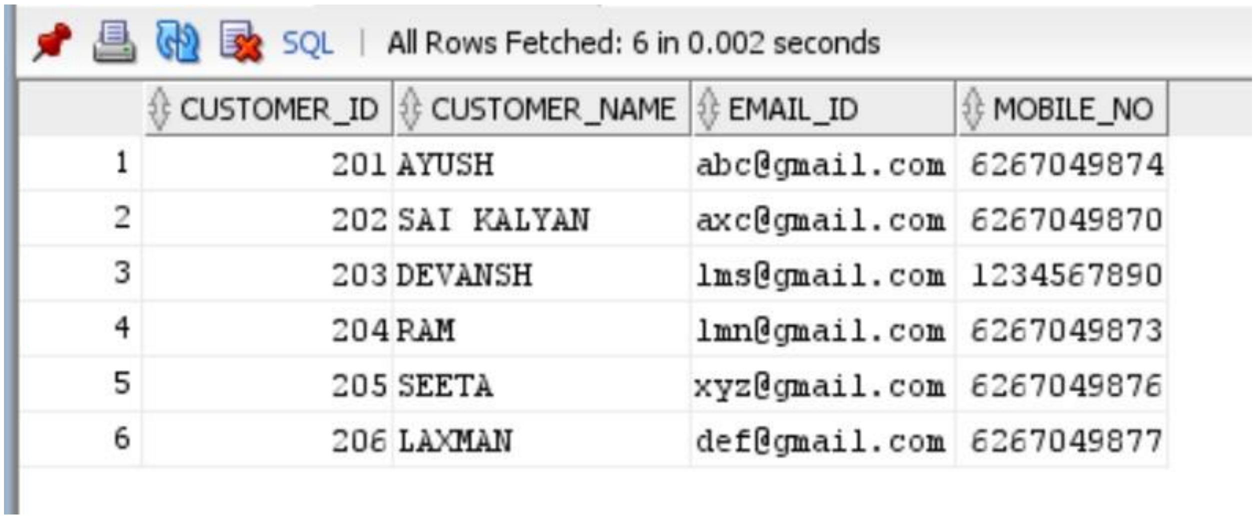
```
INSERT INTO SEAT VALUES(1,'REGULAR',101);

INSERT INTO SEAT VALUES(2,'RECLINER',101);

INSERT INTO SEAT VALUES(45,'DELUXE',104);

INSERT INTO SEAT VALUES(56,'REGULAR',103);

INSERT INTO SEAT VALUES(4,'RECLINAR',102);

INSERT INTO SEAT VALUES(7,'REGULAR',105);

INSERT INTO SEAT VALUES(3,'DELUXE',105);
```



	SEAT_NO	SEAT_TYPE	THEATRE_ID
1	1	REGULAR	101
2	2	RECLINER	101
3	45	DELUXE	104
4	56	REGULAR	103
5	4	RECLINAR	102
6	7	REGULAR	105
7	3	DELUXE	105

3. Customer

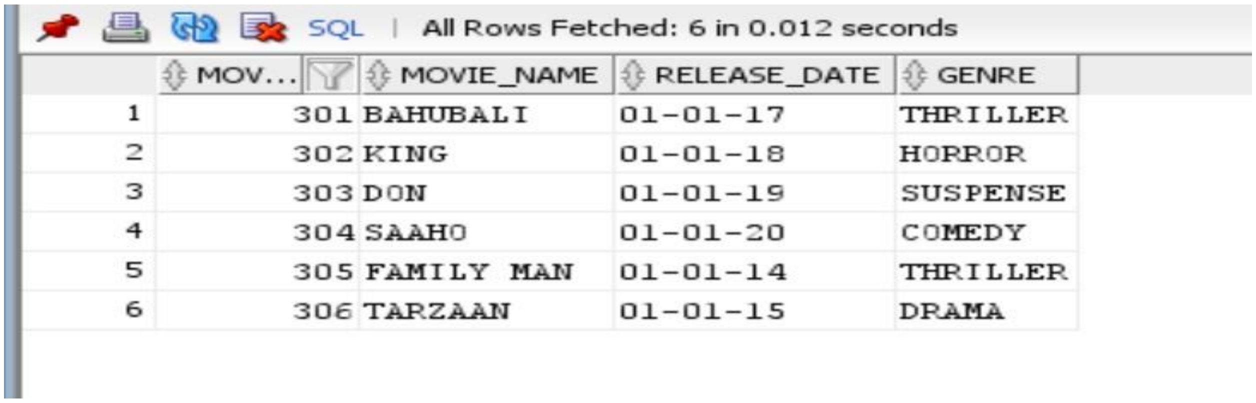
```
CREATE TABLE CUSTOMER(CUSTOMER_ID INT PRIMARY KEY,
            CUSTOMER_NAME VARCHAR(20),
            EMAIL_ID VARCHAR(30),
            MOBILE_NO NUMBER(10));

INSERT INTO CUSTOMER VALUES(201,'AYUSH','abc@gmail.com',6267049874);

INSERT INTO CUSTOMER VALUES(202,'SAI KALYAN','axc@gmail.com',6267049870);

INSERT INTO CUSTOMER VALUES(203,'DEVANSH','lms@gmail.com',1234567890);

INSERT INTO CUSTOMER VALUES(204,'RAM','lmn@gmail.com',6267049873);

INSERT INTO CUSTOMER VALUES(205,'SEETA','xyz@gmail.com',6267049876); INSERT INTO CUSTOMER
VALUES(206,'LAXMAN','def@gmail.com',6267049877);
```


SQL | All Rows Fetched: 6 in 0.002 seconds

	CUSTOMER_ID	CUSTOMER_NAME	EMAIL_ID	MOBILE_NO
1	201	AYUSH	abc@gmail.com	6267049874
2	202	SAI KALYAN	axc@gmail.com	6267049870
3	203	DEVANSH	lms@gmail.com	1234567890
4	204	RAM	lmn@gmail.com	6267049873
5	205	SEETA	xyz@gmail.com	6267049876
6	206	LAXMAN	def@gmail.com	6267049877

4. Movie

```
CREATE TABLE MOVIE(MOVIE_ID INT PRIMARY KEY ,
            MOVIE_NAME VARCHAR(30),
            RELEASE_DATE DATE ,
            GENRE VARCHAR(30));

INSERT INTO MOVIE VALUES(301,'BAHUBALI','01-01-2017','THRILLER');

INSERT INTO MOVIE VALUES(302,'KING','01-01-2018','HORROR');

INSERT INTO MOVIE VALUES(303,'DON','01-01-2019','SUSPENSE');

INSERT INTO MOVIE VALUES(304,'SAAHO','01-01-2020','COMEDY');

INSERT INTO MOVIE VALUES(305,'FAMILY MAN','01-01-2014','THRILLER');

INSERT INTO MOVIE VALUES(306,'TARZAAN','01-01-2015','DRAMA');
```

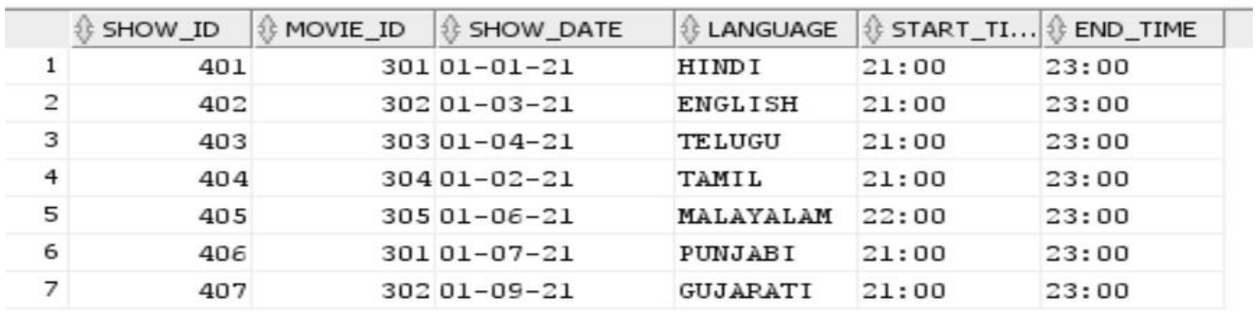


SQL | All Rows Fetched: 6 in 0.012 seconds

	MOV...	MOVIE_NAME	RELEASE_DATE	GENRE
1	301	BAHUBALI	01-01-17	THRILLER
2	302	KING	01-01-18	HORROR
3	303	DON	01-01-19	SUSPENSE
4	304	SAAHO	01-01-20	COMEDY
5	305	FAMILY MAN	01-01-14	THRILLER
6	306	TARZAAN	01-01-15	DRAMA

5. Show

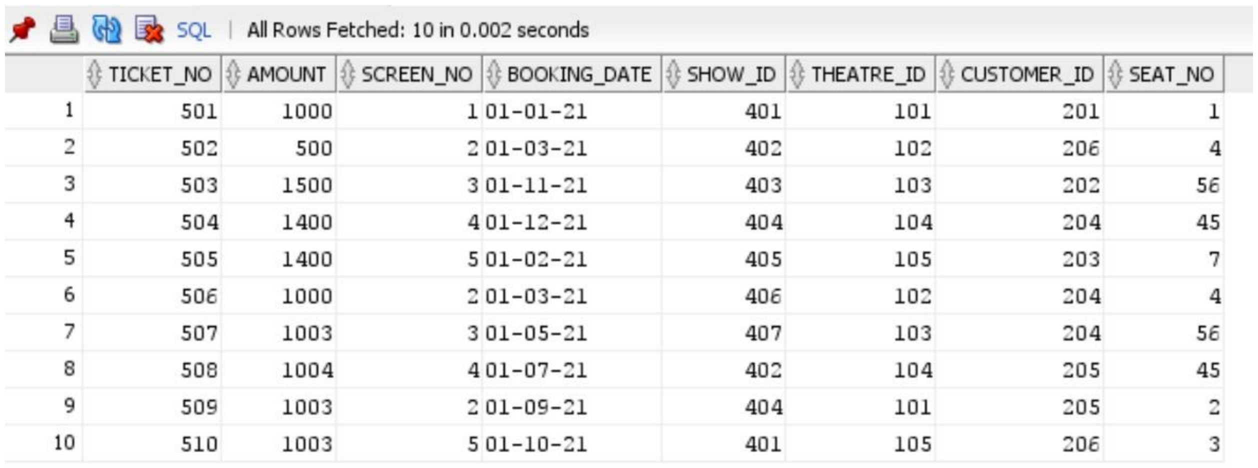
```
CREATE TABLE SHOW(SHOW_ID INT PRIMARY KEY ,
            MOVIE_ID INT,
            START_TIME TIMESTAMP,
            END_TIME TIMESTAMP,
            SHOW_DATE DATE ,
            LANGUAGE VARCHAR(20),
            FOREIGN KEY (MOVIE_ID) REFERENCES MOVIE(MOVIE_ID));

INSERT INTO SHOW VALUES(401,301,'01-01-2021','HINDI','21:00','23:00');

INSERT INTO SHOW VALUES(402,302,'01-03-2021','ENGLISH','21:00','23:00');

INSERT INTO SHOW VALUES(403,303,'01-04-2021','TELUGU','21:00','23:00');

INSERT INTO SHOW VALUES(404,304,'01-02-2021','TAMIL','21:00','23:00');

INSERT INTO SHOW VALUES(405,305,'01-06-2021','MALAYALAM','22:00','23:00');

INSERT INTO SHOW VALUES(406,301,'01-07-2021','PUNJABI','21:00','23:00');

INSERT INTO SHOW VALUES(407,302,'01-09-2021','GUJARATI','21:00','23:00');
```



	SHOW_ID	MOVIE_ID	SHOW_DATE	LANGUAGE	START_TI...	END_TIME
1	401	301	01-01-21	HINDI	21:00	23:00
2	402	302	01-03-21	ENGLISH	21:00	23:00
3	403	303	01-04-21	TELUGU	21:00	23:00
4	404	304	01-02-21	TAMIL	21:00	23:00
5	405	305	01-06-21	MALAYALAM	22:00	23:00
6	406	301	01-07-21	PUNJABI	21:00	23:00
7	407	302	01-09-21	GUJARATI	21:00	23:00

6. Ticket

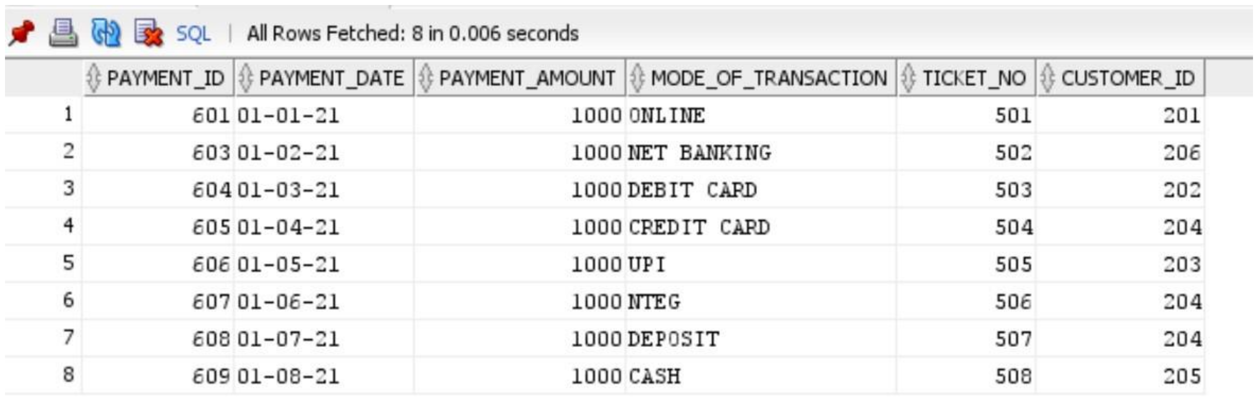
```
CREATE TABLE TICKET(TICKET_NO INT PRIMARY KEY,
                AMOUNT NUMBER,
                SCREEN_NO INT,
                BOOKING_DATE DATE,
                SHOW_ID INT,
                THEATRE_ID INT,
                CUSTOMER_ID INT,
                SEAT_NO INT,
                FOREIGN KEY (SHOW_ID) REFERENCES SHOW(SHOW_ID),
                FOREIGN KEY (THEATRE_ID) REFERENCES THEATRE(THEATRE_ID),
                FOREIGN KEY (CUSTOMER_ID) REFERENCES CUSTOMER(CUSTOMER_ID),
                FOREIGN KEY (SEAT_NO) REFERENCES SEAT(SEAT_NO));

INSERT INTO TICKET VALUES(501,1000,1,'01-01-2021',401,101,201,1);

INSERT INTO TICKET VALUES(502,500,2,'01-03-2021',402,102,206,4);

INSERT INTO TICKET VALUES(503,1500,3,'01-11-2021',403,103,202,56);

INSERT INTO TICKET VALUES(504,1400,4,'01-12-2021',404,104,204,45);

INSERT INTO TICKET VALUES(505,1400,5,'01-02-2021',405,105,203,7);

INSERT INTO TICKET VALUES(506,1000,2,'01-03-2021',406,102,204,4);

INSERT INTO TICKET VALUES(507,1003,3,'01-05-2021',407,103,204,56);

INSERT INTO TICKET VALUES(508,1004,4,'01-07-2021',402,104,205,45);

INSERT INTO TICKET VALUES(509,1003,2,'01-09-2021',404,101,205,2);

INSERT INTO TICKET VALUES(510,1003,5,'01-10-2021',401,105,206,3);
```


All Rows Fetched: 10 in 0.002 seconds

	TICKET_NO	AMOUNT	SCREEN_NO	BOOKING_DATE	SHOW_ID	THEATRE_ID	CUSTOMER_ID	SEAT_NO
1	501	1000	1	01-01-21	401	101	201	1
2	502	500	2	01-03-21	402	102	206	4
3	503	1500	3	01-11-21	403	103	202	56
4	504	1400	4	01-12-21	404	104	204	45
5	505	1400	5	01-02-21	405	105	203	7
6	506	1000	2	01-03-21	406	102	204	4
7	507	1003	3	01-05-21	407	103	204	56
8	508	1004	4	01-07-21	402	104	205	45
9	509	1003	2	01-09-21	404	101	205	2
10	510	1003	5	01-10-21	401	105	206	3

7. Payment

```
CREATE TABLE PAYMENT( PAYMENT_ID INT PRIMARY KEY,
                PAYMENT_DATE DATE,
                PAYMENT_AMOUNT NUMBER,
                MODE_OF_TRANSACTION VARCHAR(30),
                TICKET_NO INT,
                CUSTOMER_ID INT,
                FOREIGN KEY (TICKET_NO) REFERENCES TICKET(TICKET_NO),
                FOREIGN KEY (CUSTOMER_ID) REFERENCES CUSTOMER(CUSTOMER_ID));

INSERT INTO PAYMENT VALUES(601,'01-01-2021',1000,'ONLINE',501,201);

INSERT INTO PAYMENT VALUES(603,'01-02-2021',1000,'NET BANKING',502,206);

INSERT INTO PAYMENT VALUES(604,'01-03-2021',1000,'DEBIT CARD',503,202);

INSERT INTO PAYMENT VALUES(605,'01-04-2021',1000,'CREDIT CARD',504,204);

INSERT INTO PAYMENT VALUES(606,'01-05-2021',1000,'UPI',505,203);

INSERT INTO PAYMENT VALUES(607,'01-06-2021',1000,'NTEG',506,204);

INSERT INTO PAYMENT VALUES(608,'01-07-2021',1000,'DEPOSIT',507,204);

INSERT INTO PAYMENT VALUES(609,'01-08-2021',1000,'CASH',508,205);
```


SQL | All Rows Fetched: 8 in 0.006 seconds

	PAYMENT_ID	PAYMENT_DATE	PAYMENT_AMOUNT	MODE_OF_TRANSACTION	TICKET_NO	CUSTOMER_ID
1	601	01-01-21	1000	ONLINE	501	201
2	603	01-02-21	1000	NET BANKING	502	206
3	604	01-03-21	1000	DEBIT CARD	503	202
4	605	01-04-21	1000	CREDIT CARD	504	204
5	606	01-05-21	1000	UPI	505	203
6	607	01-06-21	1000	NTEG	506	204
7	608	01-07-21	1000	DEPOSIT	507	204
8	609	01-08-21	1000	CASH	508	205

Functional Dependencies and Normalization

1. Ticket

Ticket_no -> {Amount, Screen_no, Booking_date, Show_id, Customer_id, T heatre_id, Seat_no}

Since all the fields depend on Ticket_no, (Ticket_no)$^{+}$-> R.

Hence, Ticket_no is Primary Key.

Since all the attributes depend on the primary key and have no transitive dependency, the table is in 3NF.

2. Customer

Customer_id -> {Customer_name, Email_id, Mobile_no} Since all the fields depend on Customer_id, (Customer_id)$^{+}$ -> R.

Hence, Customer_id is Primary Key.

Since all the attributes are fully functional dependent on the primary key, hence the table is in BCNF.

3. Payment

Payment_id -> {Payment_date, Payment_amount, Ticket_no, Customer_id, Mode_of_transaction}

Since all the fields depend on Payment_id, (Payment_id)$^{+}$ -> R.

Hence, Payment_id is Primary Key.

Since all the attributes depend on the primary key and have no transitive dependency, the table is in 3NF.

4. Show

Show_id -> {Start_time, End_time, Show_date, Language, Movie_id} Since all the fields depend on Show_id, $(\text{Show\_id})^+ \to R$.
Hence, Show_id is Primary Key.
Since all the attributes depend on the primary key and have no transitive dependency, the table is in 3NF.

5. Movie

Movie_id -> {Movie_name, Release_date, Genre} Since all the fields depend on Movie_id, $(\text{Movie\_id})^+ \to R$.
Hence, Movie_id is Primary Key.
Since all the attributes are fully functional dependent on the primary key, hence the table is in BCNF.

6. Seat

Seat_no -> {Seat_type, Theatre_id}
Since all the fields depend on Seat_no, $(\text{Seat\_no})^+ \to R$.
Hence, Seat_no is Primary Key.
Since all the attributes depend on the primary key and have no transitive dependency, the table is in 3NF.

7. Theatre

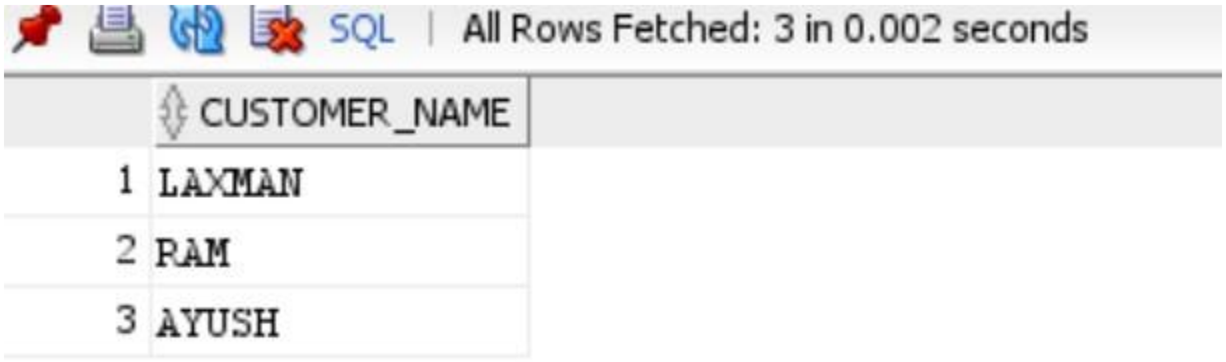
Theatre_id -> {Theatre_name, Theatre_location, Theatre_capacity, Theatre_email_id}
Since all the fields depend on Theatre_id, $(\text{Theatre\_id})^+ \to R$.
Hence, Theatre_id is Primary Key.
Since all the attributes are fully functional dependent on the primary key, hence the table is in BCNF.

Queries:

1.Find the customer name who watched the movie Bahubali.

```
SELECT CUSTOMER_NAME FROM CUSTOMER
WHERE CUSTOMER_ID IN (
```

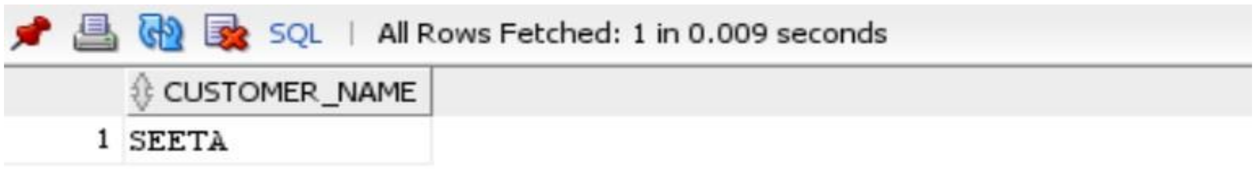
SELECT CUSTOMER_ID
FROM TICKET
WHERE SHOW_ID IN (
SELECT SHOW_ID
FROM SHOW
WHERE MOVIE_ID =(
SELECT MOVIE_ID
FROM MOVIE
WHERE MOVIE_NAME='BAHUBALI')));


SQL | All Rows Fetched: 3 in 0.002 seconds

	CUSTOMER_NAME
1	LAXMAN
2	RAM
3	AYUSH

2. Find out the customer's name who paid in CASH.

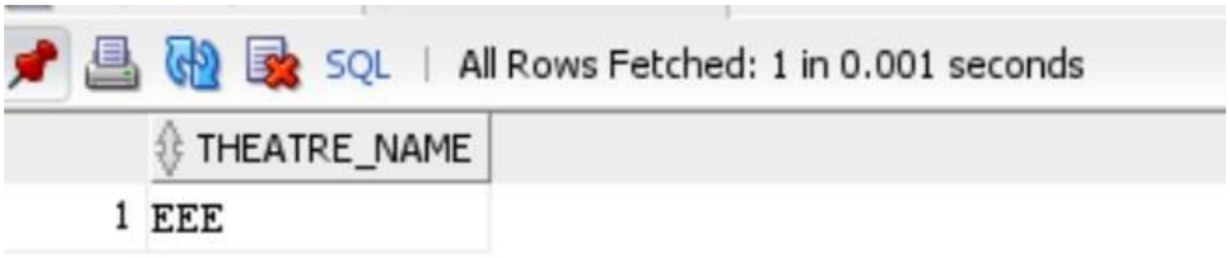
SELECT CUSTOMER_NAME
FROM CUSTOMER
WHERE CUSTOMER_ID=(SELECT CUSTOMER_ID FROM PAYMENT WHERE MODE_OF_TRANSACTION='CASH');


SQL | All Rows Fetched: 1 in 0.009 seconds

	CUSTOMER_NAME
1	SEETA

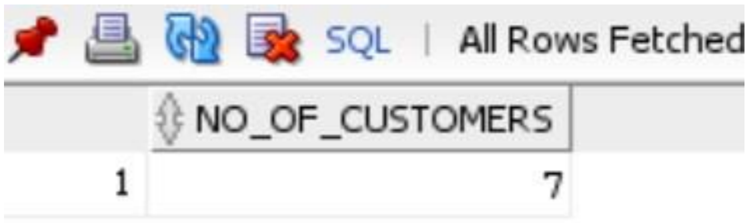
3. List the theatre whose capacity is less than 100.

SELECT THEATRE_NAME
FROM THEATRE
WHERE THEATRE_CAPACITY<100;


```
SQL | All Rows Fetched: 1 in 0.001 seconds
    THEATRE_NAME
  1 EEE
```

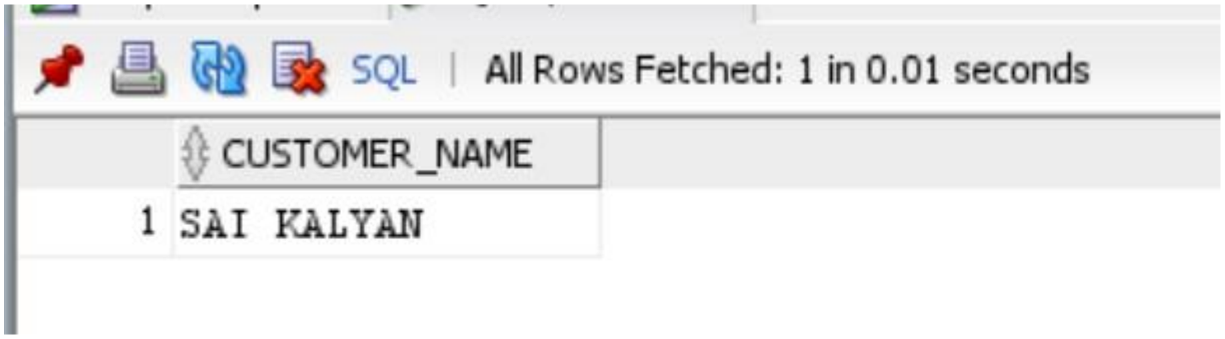
4. Find the number of customers who bought tickets whose price is greater than 1000?

SELECT COUNT(*) AS NO_OF_CUSTOMERS
FROM TICKET
WHERE AMOUNT>1000;


```
SQL | All Rows Fetched
    NO_OF_CUSTOMERS
  1               7
```

5. Find all customers who watched at least 1 telugu movie.

SELECT DISTINCT CUSTOMER_NAME
FROM CUSTOMER
WHERE CUSTOMER_ID IN (
SELECT CUSTOMER_ID
FROM TICKET
WHERE SHOW_ID IN(
SELECT SHOW_ID
FROM SHOW
WHERE LANGUAGE ='TELUGU'));


```
SQL | All Rows Fetched: 1 in 0.01 seconds
    CUSTOMER_NAME
  1 SAI KALYAN
```