

 CSE304: DATABASE MANAGEMENT SYSTEM

COURSE PROJECT REPORT

*Movie Theatre Database Management*

**Done by:**

Jaswanth. Polarowthu (AP20110010416)

**Under the guidance of: DR. Rajiv Senapati**

**School of Computer Science and Engineering, SRMAP University**

**1**

**Certificate**

Date: 16-Nov-22

This is to certify that the work present in this Project entitled “**Movie Theatre Management System**” has been carried out by **P.Jaswanth** under my/our supervision. The work is genuine, original, and suitable for submission to the SRM University – AP for the award of Bachelor of Technology/Master of Technology in **School of**

**Engineering and Sciences**.

**Supervisor**

(Signature)

Prof. /**Dr.**[Name ]

Designation,

Affiliation.

**Co-supervisor**

(Signature)

Prof. / Dr. [Name]

Designation,

Affiliation.

**2**

**Acknowledgements**

We thank **Dr.Rajiv Senapati sir**, our professor in charge, for the help and direction in

finishing our project on the subject Database Management System. It was fantastic

learning opportunity. This subject has given me an opportunity to explore the field I

have always been curious about. Your insightful counsel and recommendations were

quite beneficial to me as I finished the assignment. I will always be grateful to you

for this.

**3**

# Table of Contents

[Certificate **1**](#_30j0zll)

[Acknowledgements](#_1fob9te) **3**

[Table of Contents](#_gjdgxs) **4**

[Abstract](#_3znysh7) **5**

[Introduction](#_2et92p0) **6**

1. [Methodology 6](#_tyjcwt)
   1. [Identifying the Entities 6](#_3dy6vkm)
      1. Identifying the Attributes and Primary Keys
      2. Identification Of Mapping Cardinalities9
   2. ER Model
   3. Relational Schema [9](#_4d34og8)
   4. [Normalization 10](#_2s8eyo1)
   5. [SQL Code 13](#_17dp8vu)
2. [SQL Queries 25](#_3rdcrjn)

[References 26](#_26in1rg)

**4**

**Abstract:**

The main aim of this project is to build a database for Multiplex. Multiplex is a movie theatre complex with multiple screens within a single complex. They are usually housed in a specially designed building.

5

**INTODUCTION:**

1. Movie Theatre Database System is an application of database management system which provides the user to book movie tickets and manager to keep their records.
2. Movie Theatre Database system revolutionized the trivial system of distribution of Paper tickets and made it easy for users to book tickets from home.

1. This system provides secure, organized, and efficient storage of huge data. One does not have to be physically present at the theatre to get their jobs done.
2. With the advent of various online platform like **BookMyShow and** **Paytm** use type of movie theatre database system, the process became very simple, hustle free and user-friendly.
3. This Project aims to provide an insight of such Movie Theatre Database System.

**Identifying the Entities:**

People, Movie, Booking, Screening, Hall, Seat\_Reserved, Seat

**Identifying the attributes & Primary Keys:**

**People:** It is the set of all the people who book the tickets. Each Person is described by People ID, Name, Age and phone number.

**Attributes:** people\_id, Name, Age, phone\_number.

**Movie:** This entity stores the details of movies to be shown on the screen in the theatre. Each movie is described by its meta-data Movie Id, movie title, movie director, movie length and movie genre.

**Attributes**: movie\_id, movie\_title, movie\_director, movie\_length, movie\_genre.

**Booking:** Booking stores data of booked tickets containing details of price, booking Id.

**Attributes**: booking\_id, price.

6

**Screening:** It is the combined data of the Movie and hall in which it is shown. It describes screening id, screen time, screen data.

**Attributes**: screening\_ID, screen\_time, screen\_date.

**Hall:** This entity stores the data of hall and seats combined described by hall id, number of seats and hall name.

**Attributes**: hall\_ID, no\_of\_seats, hall\_name.

**Seat Reserved:** Based on the booking, this entity stores the data of seat reserve id.

**Attributes**:seat\_Reserv\_ID.

**Seat:** This entity contains the information of seat in a hall and is described by seat id, seat row and seat number.

**Attributes**: seat\_ID, seat\_row, seat\_number.

**7**

**Identification of cardinality between the entities:**

Diagram

Description automatically generated8

**ER Diagram:**

Diagram

Description automatically generated

Relational Model:Diagram

Description automatically generated

9

**Normalization:**

All the functional dependencies present in this database (R is relation) are as follows:

**A picture containing chart

Description automatically generated**

And it can be clearly stated from all the functional dependencies that the closure of G, that is

(G)+ = R.

As there are no multivalued attribute in our database hence the given relation is in 1NF Form.

Now we can clearly see that there are no partial dependencies in our Relation R as partial dependencies occur when a subset of candidate key is capable to derive other non-prime attributes in the relation. But as in our Relation, there is only one attribute

10

hence there cannot be any proper subset of the candidate key capable to derive other attribute.

Hence this proves that our table is in 2NF Form.

Now, we can clearly notice from the dependencies that there are transitive dependencies present (as E, H and N being non-prime derives another non-prime attributes), hence, the table is not in 3NF Form.

After breaking the functional dependencies such that there are no non-prime to non-prime dependencies (transitive dependencies) table would come in 3NF Form and the functional dependencies now looks like:

Table

Description automatically generated with low confidence

Now, in the above dependencies we can clearly notice that there are no dependencies that corresponds to a prime attribute (prime attribute is a proper subset of candidate key, in our case

11

only G is a prime attribute), hence it is safe to mention that our Relation is now in BCNF Form.

Here, the functional dependencies after doing normalization till BCNF Form corresponding to the table present in our database are as followed:

Functional Dependency (i) corresponds to table Seat Reserved

Functional Dependency (ii) corresponds to table Booking

Functional Dependency (iii) corresponds to table People

Functional Dependency (iv) corresponds to table Screening

Functional Dependency (v) corresponds to table Hall

Functional Dependency (vi) corresponds to table Movie

Functional Dependency (vii) corresponds to table Seat.

**12**

**SQL Code:**

>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>#CREATING DATABASE<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

CREATE DATABASE Theatre;

USE Theatre;

>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>#CREATING TABLES<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

**#1. MOVIE TABLE:**

CREATE TABLE Movie(

movie\_ID VARCHAR(6) NOT NULL,

movie\_title VARCHAR(20) NOT NULL,

movie\_director VARCHAR(20) NOT NULL,

movie\_length INT NOT NULL,

movie\_genre VARCHAR(7) NOT NULL,

PRIMARY KEY(movie\_ID)

);

Graphical user interface, text

Description automatically generated with medium confidence

#INSERTING VALUES INTO THE TABLE:

INSERT INTO Movie VALUES('M1','RRR','SS RAJAMOULI','189','DRAMA');

INSERT INTO Movie VALUES('M2','BHEEMLA NAYAK','SAGAR K CHANDRA','178','ACTION');

INSERT INTO Movie VALUES('M3','MIRCHI','KORATALA SIVA','180','FAMILY');

INSERT INTO Movie VALUES('M4','DARLING','KARUNAKARAN','175','LOVE');

INSERT INTO Movie VALUES('M5','ROBO','SHANKAR','173','SCIFI');

13

A picture containing table

Description automatically generated

Table

Description automatically generated

DESC Movie;

Table

Description automatically generated

**14**

**#2.HALL TABLE:**

CREATE TABLE Hall(

hall\_ID VARCHAR(6) NOT NULL,

no\_of\_seats INT NOT NULL,

hall\_name VARCHAR(10) NOT NULL,

PRIMARY KEY(hall\_ID)

);

Graphical user interface, text

Description automatically generated

#INSERTING VALUES INTO THE TABLE:

INSERT INTO Hall VALUES('H1','5','Platinum');

INSERT INTO Hall VALUES('H2','6','Gold');

INSERT INTO Hall VALUES('H3','7','Silver');

Text

Description automatically generated

15

Graphical user interface, application, table

Description automatically generated

DESC Hall;

Graphical user interface, text, application

Description automatically generated

**#3.PEOPLE TABLE:**

CREATE TABLE People(

people\_ID VARCHAR(6) NOT NULL,

NAME VARCHAR(20) NOT NULL,

Age INT NOT NULL,

phone\_number CHAR(10) NOT NULL,

PRIMARY KEY(people\_ID)

);

Graphical user interface, text, application

Description automatically generated

#INSERTING VALUES INTO THE TABLE:

INSERT INTO People VALUES('P1','Dhruva','20','7995556100');

INSERT INTO People VALUES('P2','Tanuja','16','7995129576');

INSERT INTO People VALUES('P3','Rishi','17','8977703366');

INSERT INTO People VALUES('P4','Tony','19','9390254779');

INSERT INTO People VALUES('P5','Likhith','18','9908087704');

INSERT INTO People VALUES('P6','Bharath','17','6305677735'); 16

Text, table

Description automatically generated with medium confidence

Graphical user interface, table

Description automatically generated

DESC People;

Graphical user interface, text

Description automatically generated

**#4.SCREENING TABLE:**

CREATE TABLE Screening(

screening\_ID VARCHAR(6) NOT NULL,

movie\_ID VARCHAR(6) NOT NULL,

hall\_ID VARCHAR(6) NOT NULL,

screen\_time TIME NOT NULL, 17

screen\_date DATE NOT NULL,

PRIMARY KEY(screening\_ID),

FOREIGN KEY (movie\_ID) REFERENCES Movie(movie\_ID),

FOREIGN KEY (hall\_ID) REFERENCES Hall(hall\_ID)

);

Graphical user interface, text, application

Description automatically generated

#INSERTING VALUES INTO THE TABLE:

INSERT INTO Screening VALUES('SCR1','M1','H2','17:35:00','2019-10-05');

INSERT INTO Screening VALUES('SCR2','M4','H1','18:40:00','2019-10-05');

INSERT INTO Screening VALUES('SCR3','M3','H2','13:20:00','2019-10-06');

INSERT INTO Screening VALUES('SCR4','M2','H3','17:35:00','2019-10-06');

INSERT INTO Screening VALUES('SCR5','M1','H2','11:00:00','2019-10-07');

Text

Description automatically generated

Table

Description automatically generated

18

DESC Screening;

Graphical user interface, application, table

Description automatically generated

**#5.SEAT TABLE:**

CREATE TABLE Seat(

seat\_ID VARCHAR(6) NOT NULL,

seat\_number INT NOT NULL,

seat\_row CHAR(6) NOT NULL,

hall\_ID VARCHAR(6) NOT NULL,

PRIMARY KEY(seat\_ID),

FOREIGN KEY (hall\_ID) REFERENCES Hall(hall\_ID)

);

Graphical user interface, text, application

Description automatically generated

#INSERTING VALUES INTO THE TABLE:

INSERT INTO Seat VALUES('H1S1','2','B','H1');

INSERT INTO Seat VALUES('H1S2','1','A','H1');

INSERT INTO Seat VALUES('H1S3','3','C','H1');

INSERT INTO Seat VALUES('H1S4','2','D','H1');

INSERT INTO Seat VALUES('H1S5','5','A','H1'); 19

INSERT INTO Seat VALUES('H2S1','4','B','H2');

INSERT INTO Seat VALUES('H2S2','7','B','H2');

INSERT INTO Seat VALUES('H2S3','8','C','H2');

INSERT INTO Seat VALUES('H2S4','3','R','H2');

INSERT INTO Seat VALUES('H2S5','5','A','H2');

INSERT INTO Seat VALUES('H2S6','9','C','H2');

INSERT INTO Seat VALUES('H3S1','6','R','H3');

INSERT INTO Seat VALUES('H3S2','7','R','H3');

INSERT INTO Seat VALUES('H3S3','1','A','H3');

INSERT INTO Seat VALUES('H3S4','9','A','H3');

INSERT INTO Seat VALUES('H3S5','6','C','H3');

INSERT INTO Seat VALUES('H3S6','2','B','H3');

INSERT INTO Seat VALUES('H3S7','4','R','H3');

Table, Excel

Description automatically generated

20

Table

Description automatically generated with medium confidence

Table

Description automatically generated with medium confidence

DESC Seat;

A screenshot of a computer

Description automatically generated

**#6.BOOKING**

CREATE TABLE Booking(

booking\_ID VARCHAR(6) NOT NULL,

price INT NOT NULL,

screening\_ID VARCHAR(6) NOT NULL,

people\_ID VARCHAR(6) NOT NULL,

PRIMARY KEY(booking\_ID),

FOREIGN KEY (screening\_ID) REFERENCES Screening(screening\_ID), 21

FOREIGN KEY (people\_ID) REFERENCES People(people\_ID)

);

Graphical user interface, text, application

Description automatically generated

#INSERTING VALUES INTO THE TABLE:

INSERT INTO Booking VALUES('B1','120','SCR1','P5');

INSERT INTO Booking VALUES('B2','150','SCR4','P1');

INSERT INTO Booking VALUES('B3','145','SCR3','P4');

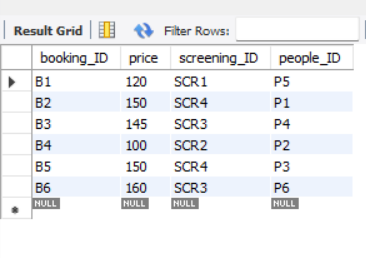
INSERT INTO Booking VALUES('B4','100','SCR2','P2');

INSERT INTO Booking VALUES('B5','150','SCR4','P3');

INSERT INTO Booking VALUES('B6','160','SCR3','P6');

Text

Description automatically generated



DESC Booking;

22

Graphical user interface

Description automatically generated

**#7.SEAT RESERVED**

CREATE TABLE Seat\_Reserverd(

seat\_Reserv\_ID VARCHAR(6) NOT NULL,

screening\_ID VARCHAR(6) NOT NULL,

seat\_ID VARCHAR(6) NOT NULL,

booking\_ID VARCHAR(6) NOT NULL,

PRIMARY KEY(seat\_Reserv\_ID),

FOREIGN KEY (screening\_ID) REFERENCES Screening(screening\_ID),

FOREIGN KEY (seat\_ID) REFERENCES Seat(seat\_ID),

FOREIGN KEY (booking\_ID) REFERENCES Booking(booking\_ID)

);

Graphical user interface, text, application

Description automatically generated

#INSERTING VALUES INTO THE TABLE:

INSERT INTO Seat\_Reserverd VALUES('SR1','SCR1','H3S3','B1');

INSERT INTO Seat\_Reserverd VALUES('SR2','SCR4','H1S2','B2');

INSERT INTO Seat\_Reserverd VALUES('SR3','SCR3','H2S1','B3');

INSERT INTO Seat\_Reserverd VALUES('SR4','SCR2','H3S3','B4');

INSERT INTO Seat\_Reserverd VALUES('SR5','SCR4','H1S1','B5');

INSERT INTO Seat\_Reserverd VALUES('SR6','SCR3','H2S2','B6'); 23

****

**Table

Description automatically generated**

**DESC Seat\_Reserved;**

**Graphical user interface, table

Description automatically generated**

24

>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>#QUERIES<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

1. Which movies are screened on 2019-10-06?

SELECT movie\_title, movie\_director, movie\_length, movie\_genre, hall\_ID

FROM Movie

INNER JOIN Screening

ON Screening.movie\_ID = Movie.movie\_ID

WHERE screen\_date = "2019-10-06";

Graphical user interface, application

Description automatically generated

2.**Show seat map of hall Royal?**

SELECT seat\_ID, seat\_row, seat\_number

FROM Seat

INNER JOIN Hall

ON Seat.hall\_ID = Hall.hall\_ID

WHERE hall\_name = "Silver";

Graphical user interface, table

Description automatically generated

**25**

3. Show the date, time and hall when the film “RRR” will be screened?

SELECT movie\_title, hall\_ID, screen\_date, screen\_time

FROM Screening

INNER JOIN Movie

ON Screening.movie\_ID = Movie.movie\_ID WHERE movie\_title = "RRR";

Graphical user interface, application, table

Description automatically generated

4. Show all the details of people who booked for the film “BHEEMLA NAYAK”?

SELECT Name, Age, phone\_number FROM People

INNER JOIN Booking

ON People.people\_ID = Booking.people\_ID

INNER JOIN Screening

ON Screening.screening\_ID = Booking.screening\_ID INNER JOIN Movie

ON Screening.movie\_ID = Movie.movie\_ID

WHERE movie\_title = "BHEEMLA NAYAK";

Graphical user interface, application

Description automatically generated

25

# References

<https://www.freeprojectz.com/entity-relationship/movie-ticket-booking-system-er-diagram>

<https://www.gdc-tech.com/cinema-solutions/cinema-enterprise-software/theatre-management-system-tms-2000/>

<https://pro.sony/en_IN/products/digital-cinema-software/stm-100>

<https://uniquex.com/rrosettabridge-tms-the-cinemas-operational-hub/>

<https://www.touchdynamic.com/why-a-movie-theatre-management-system-is-crucial-for-revenue-success/>

26