Database Management and Query Processing Report

Description :

The Ticket-Booking database keeps track of the different movies showing in a cinema, the employees incharge, and the customers booking the tickets.

The system stores the data of each employee’s name, salary, date of birth, employee-ID, address, and gender. Each employee is associated with exactly one department.

There can be various departments incharge of various activities in the cinema such as cleaning, sound, projection, etc.

The system keeps a track of all the movies that are currently showing and stores their title, rating, duration,start date, end date, trailer link, language, genre, the cast and description of the movie.

Customer or ticket-booker’s details like name, date of birth, age, gender and phone number are stored in the system.

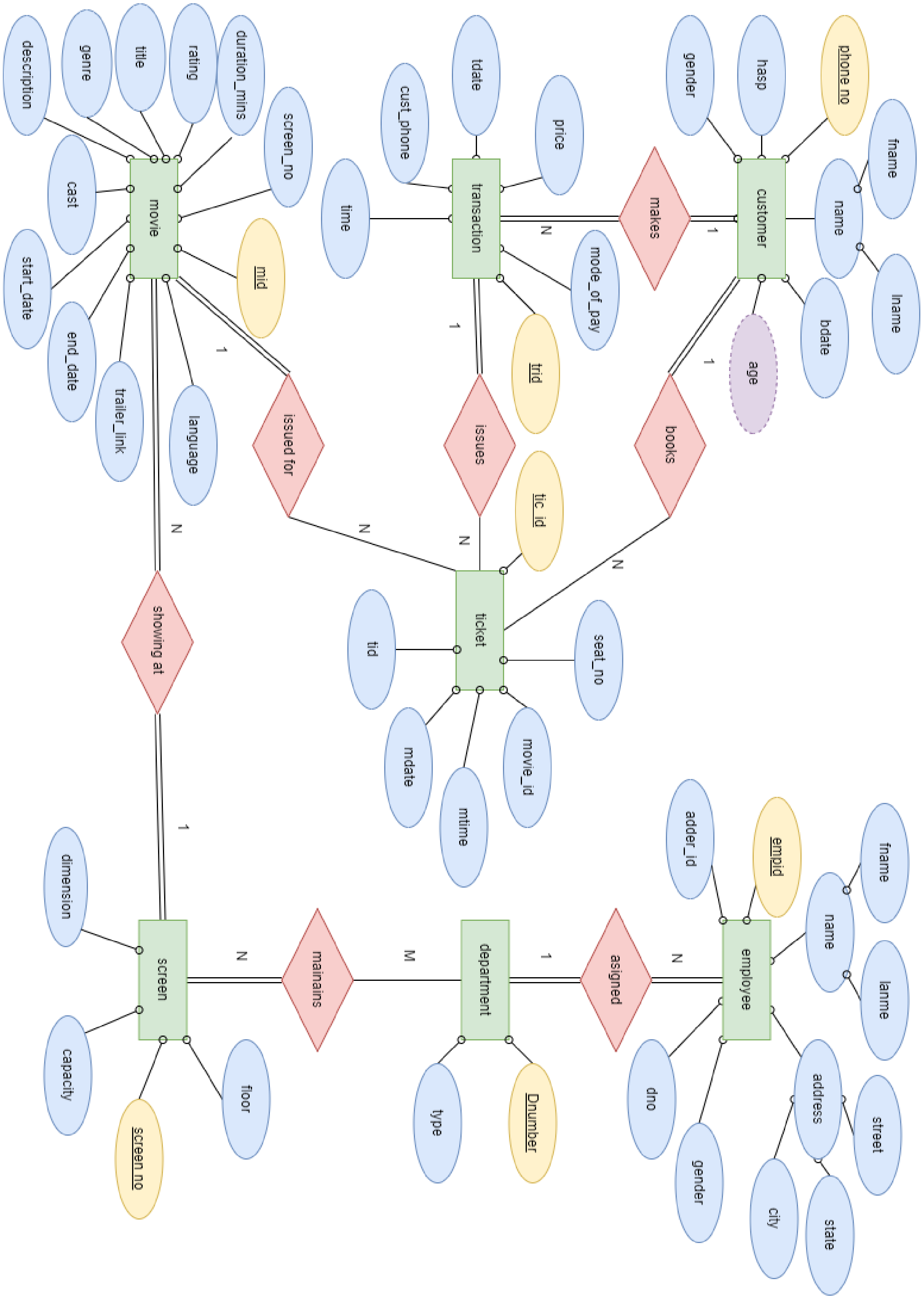
Information about each ticket includes the movie title, show time, date, screen number, seat number and price.

Further with in each department employees are assigned to respective screens.

Each screen has details such as capacity, and screen number.

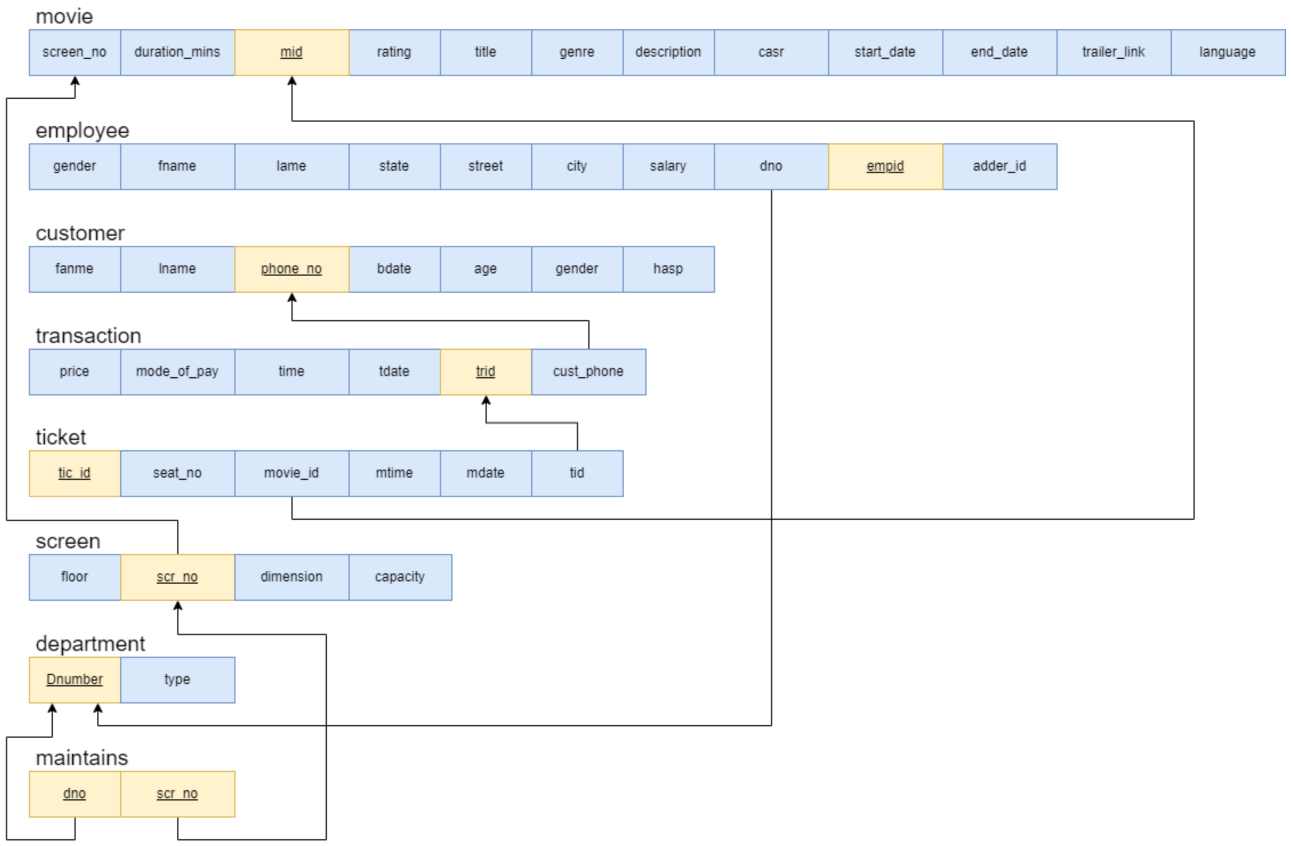
The system has a booking portal that stores customer and ticket information along with transaction details such as transaction ID, date of booking, time of booking, mode of payment and number of tickets.

ER Model :



movie
description
genre
title
rating
duration_mins
screen_no
cast
start_date
end_date
trailer_link
language
mid
transaction
cust_phone
tdate
price
time
mode_of_pay
trid
makes
customer
gender
hasp
phone no
name
fname
lname
bdate
age
issued for
issues
books
ticket
tic_id
tid
seat_no
mdate
mtime
movie_id
showing at
screen
dimension
capacity
screen no
floor
mainains
department
type
Dnumber
asigned
employee
adder_id
empid
name
fname
lanme
dno
gender
address
street
city
state
1
N
1
1
1
1
N
N
N
N
1
N
M
1
N

Schema :



movie

screen_no	duration_mins	mid	rating	title	genre	description	casr	start_date	end_date	trailer_link	language

employee

gender	fname	lame	state	street	city	salary	dno	empid	adder_id

customer

fanme	lname	phone_no	bdate	age	gender	hasp

transaction

price	mode_of_pay	time	tdate	trid	cust_phone

ticket

tic_id	seat_no	movie_id	mtime	mdate	tid

screen

floor	scr_no	dimension	capacity

department

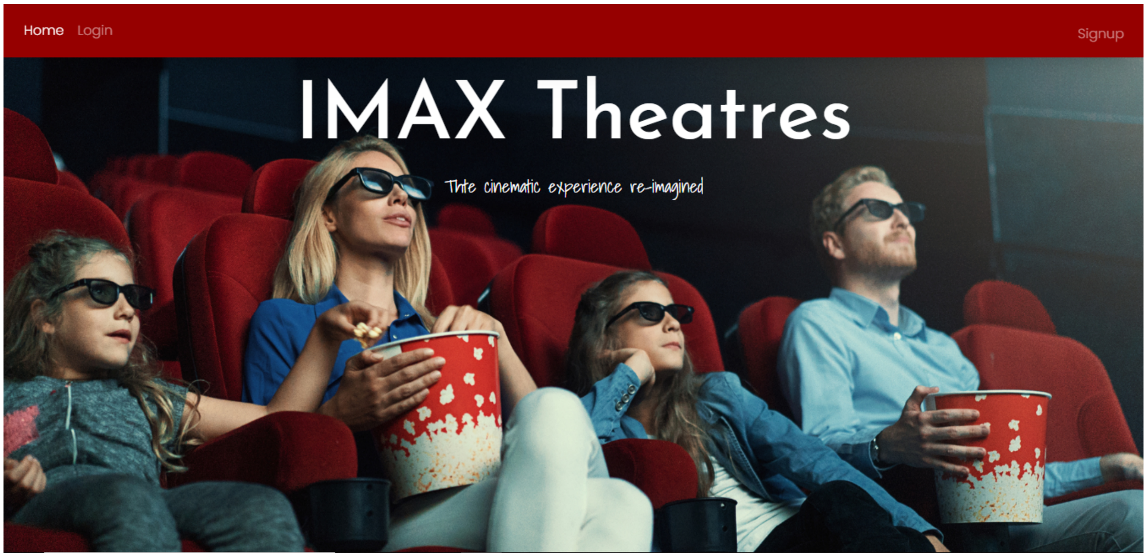
Dnumber	type

maintains

dno	scr_no

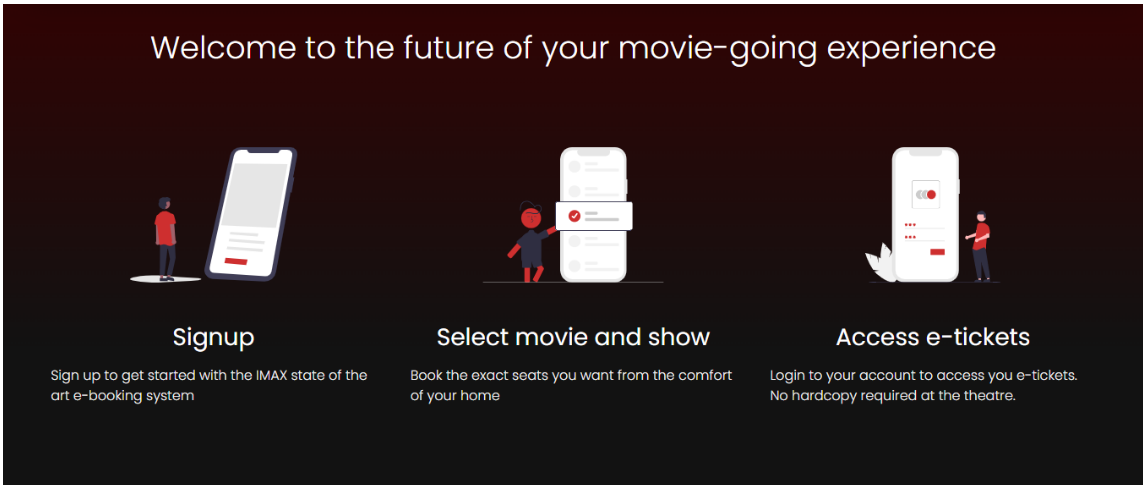
Normalised upto 3rd Normal Form.

Explanation of each form :



IMAX Theatres

Thte cinematic experience re-imagined



Welcome to the future of your movie-going experience

Signup

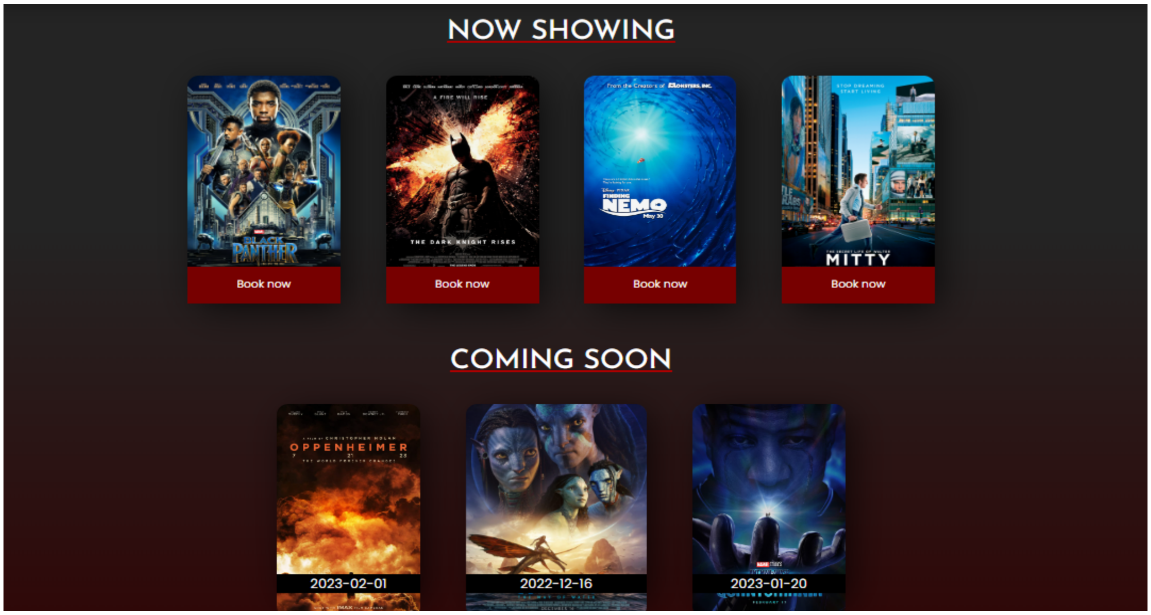
Sign up to get started with the IMAX state of the art e-booking system

Select movie and show

Book the exact seats you want from the comfort of your home

Access e-tickets

Login to your account to access you e-tickets. No hardcopy required at the theatre.



NOW SHOWING

COMING SOON

2023-02-01

2022-12-16

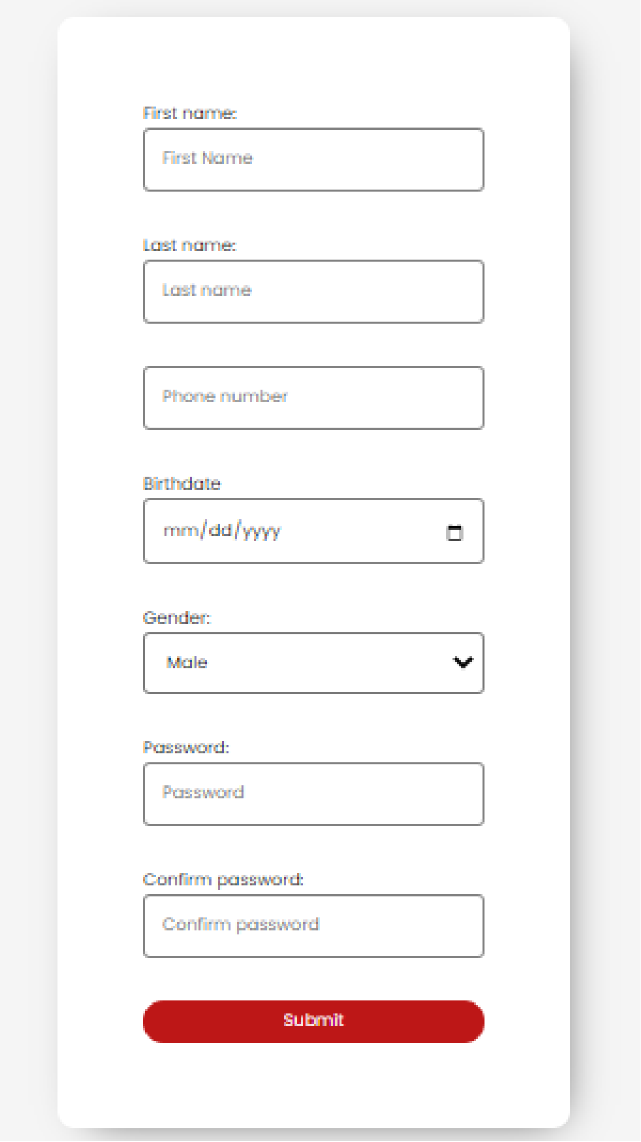
2023-01-20

New user signup

In order to book tickets, the user needs to be signed up to the platofrm. If already signed up then he can simply login. If not then he has to enter all his details in the signup form.

1. **Enter details**

   Enter all the details requested in the signup form.



First name:

First Name

Last name:

Last name

Phone number

Birthdate

mm/dd/yyyy

Gender:

Male

Password:

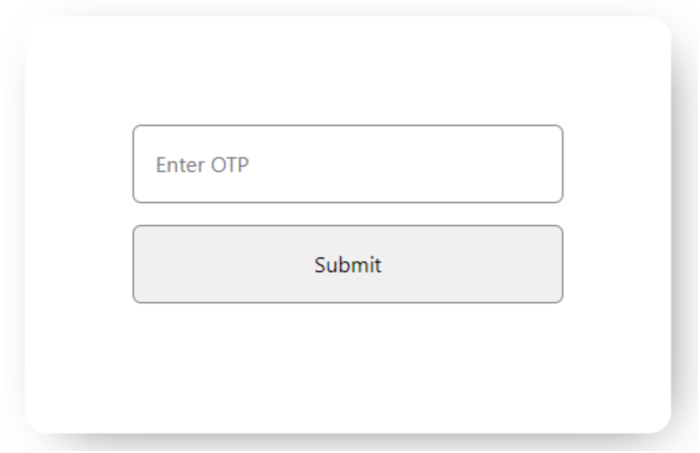
Password

Confirm password:

Confirm password

2. OTP Verification

Once the user, has filled the signup form, he will receive a unique OTP via SMS through the phone number provided in the registration form.



3. Successful signup

Once the user successfully enters the correct OTP, he will be registered to the platform and is automatically signed in.

.

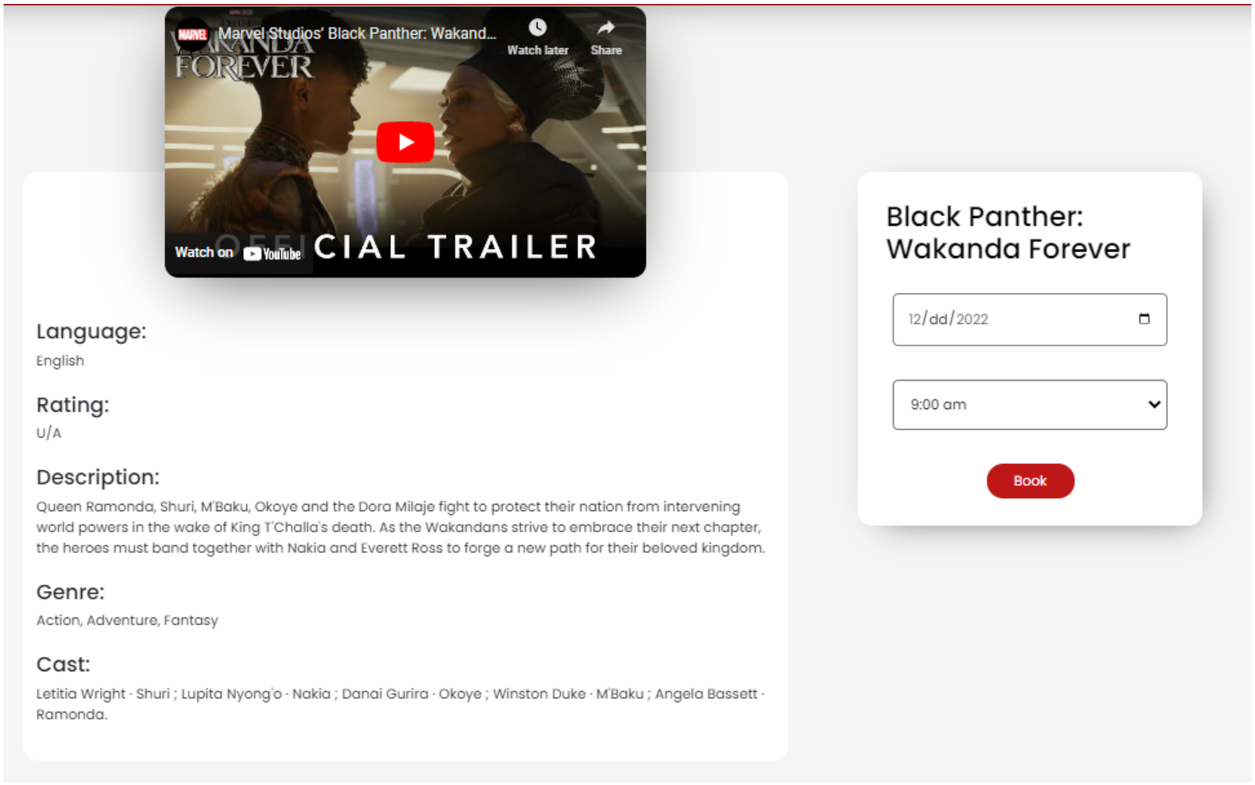
Booking movie tickets

1. Select Movie

Select the move you wish to book by clicking the ‘Book now’ button on the home page.

2. Select show date and time

This will display the details of the movie along with option to select the show date and time.



Language:
English

Rating:
U/A

Description:
Queen Ramonda, Shuri, M'Baku, Okoye and the Dora Milaje fight to protect their nation from intervening world powers in the wake of King T'Challa's death. As the Wakandans strive to embrace their next chapter, the heroes must band together with Nakia and Everett Ross to forge a new path for their beloved kingdom.

Genre:
Action, Adventure, Fantasy

Cast:
Letitia Wright · Shuri ; Lupita Nyong'o · Nakia ; Danai Gurira · Okoye ; Winston Duke · M'Baku ; Angela Bassett · Ramonda.

Black Panther: Wakanda Forever

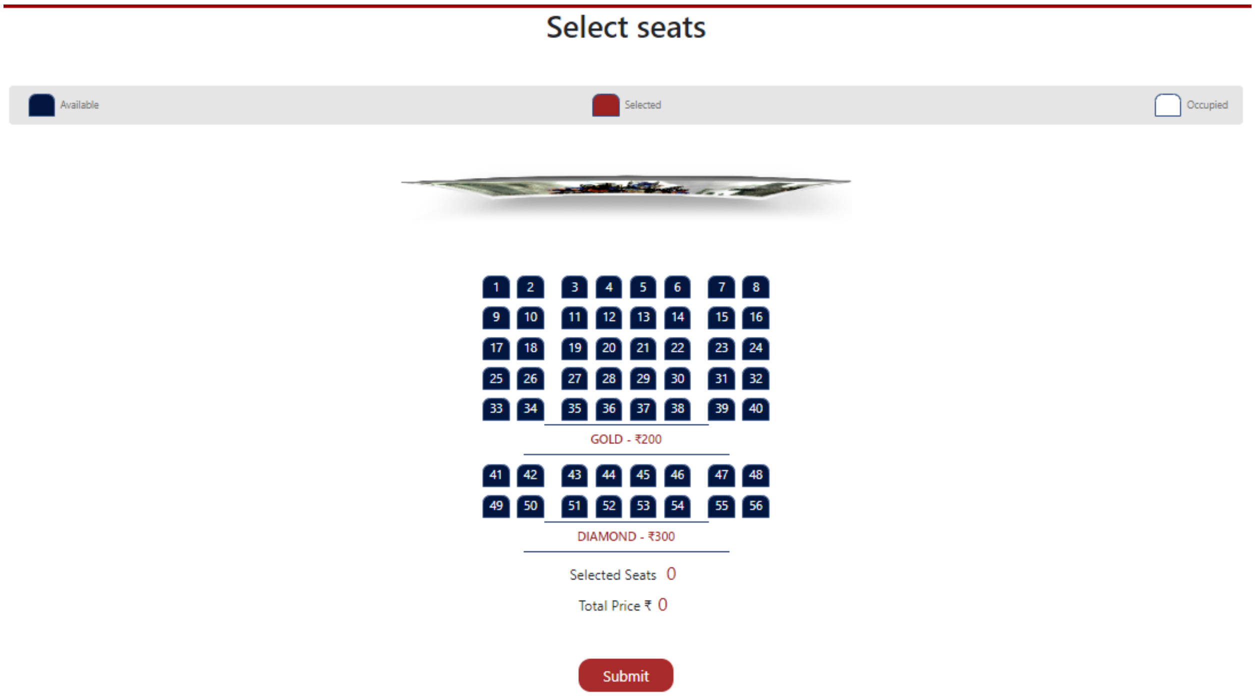
12/dd/2022

9:00 am

3. Select seats

After selecting the time and date, clicking on the ‘Book’ button will allow you to select which seats you want to book

.



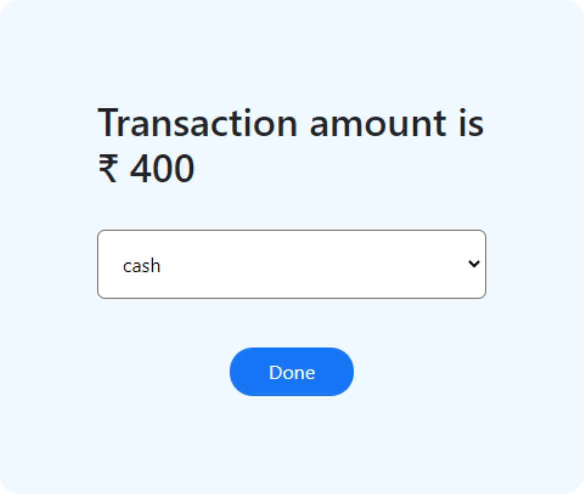
Select seats

Selected Seats 0

Total Price ₹ 0

4. Transaction

After selecting the seats you wish to book, the next step is to make the transaction by selecting the mode of payement.

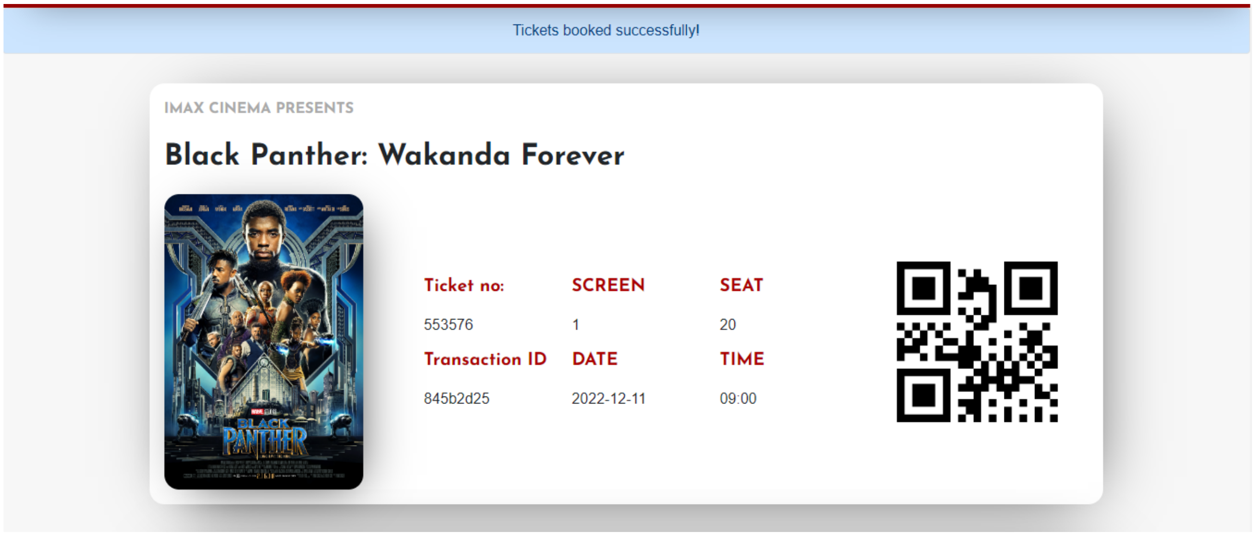


Transaction amount is ₹ 400

cash

5. Tickets booked and generated

You should now be able to see the tickets that you have booked each of which have a unique ticket ID and details about the movie, date and time. Each ticket also has a unique QR code which is used to authenticate it at the ticket counter. The user also has the ability to view their booked tickets by clcicking ‘History’ on the navigation bar.



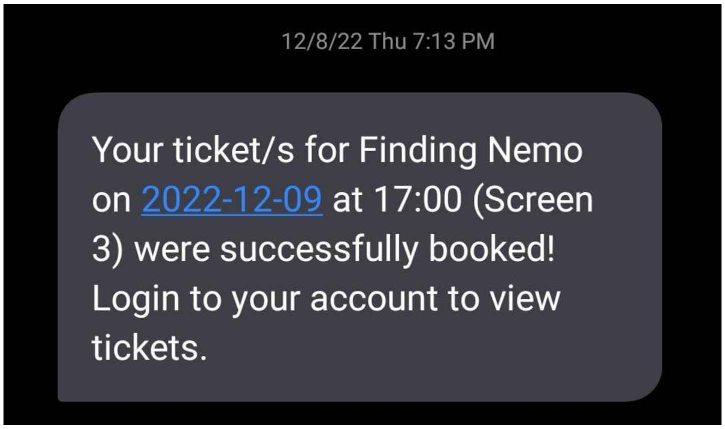
Tickets booked successfully!

IMAX CINEMA PRESENTS

Black Panther: Wakanda Forever

Ticket no:	SCREEN	SEAT
553576	1	20
Transaction ID	DATE	TIME
845b2d25	2022-12-11	09:00

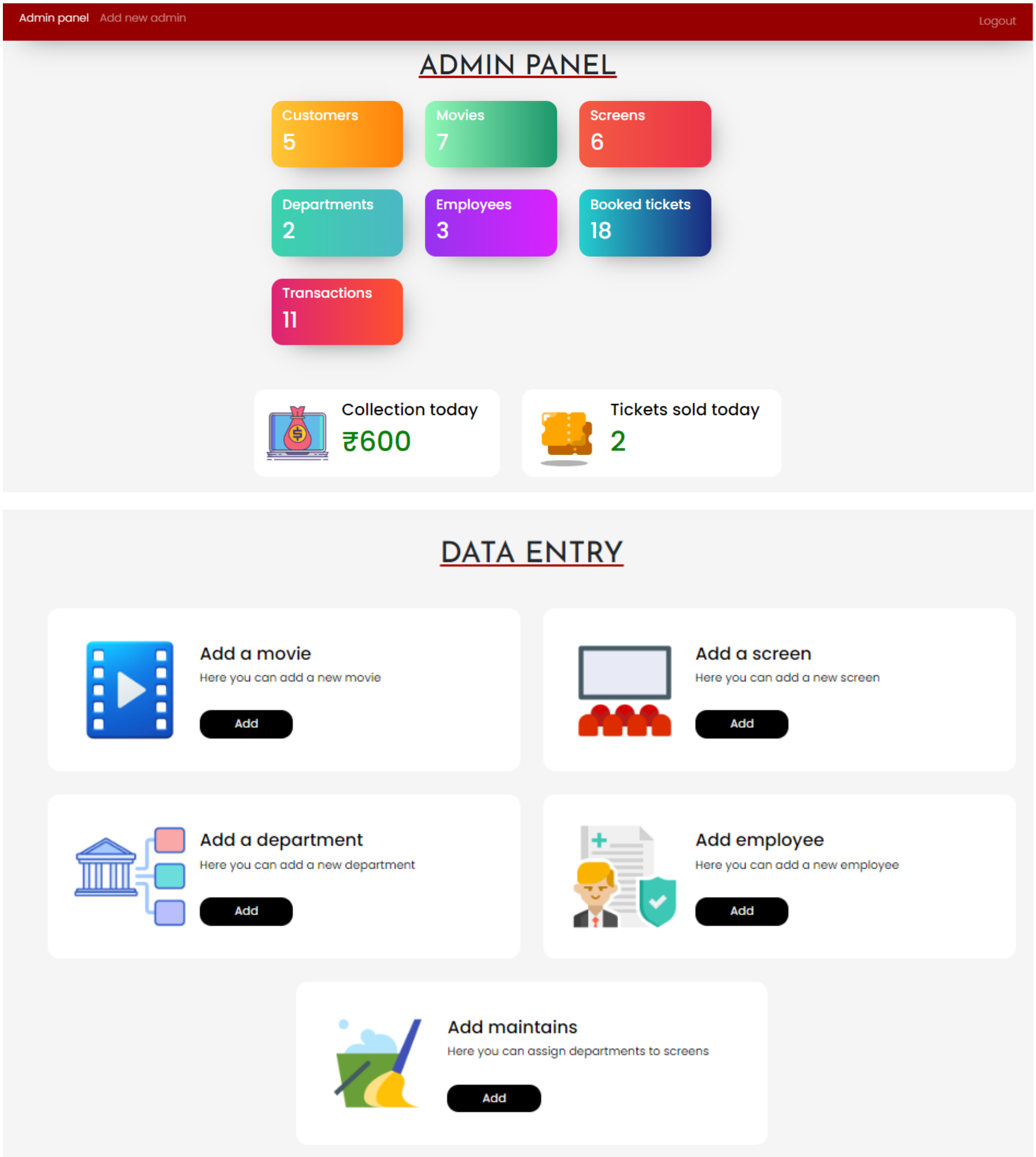
You will also receive an SMS on the registered phone number confirming the booking.



12/8/22 Thu 7:13 PM

Your ticket/s for Finding Nemo on 2022-12-09 at 17:00 (Screen 3) were successfully booked! Login to your account to view tickets.

Admin Panel



ADMIN PANEL

Customers
5

Movies
7

Screens
6

Departments
2

Employees
3

Booked tickets
18

Transactions
11

Collection today
₹600

Tickets sold today
2

DATA ENTRY

Add a movie
Here you can add a new movie

Add a screen
Here you can add a new screen

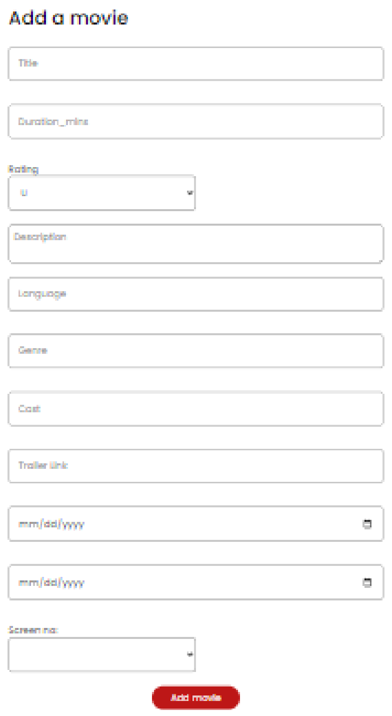
Add a department
Here you can add a new department

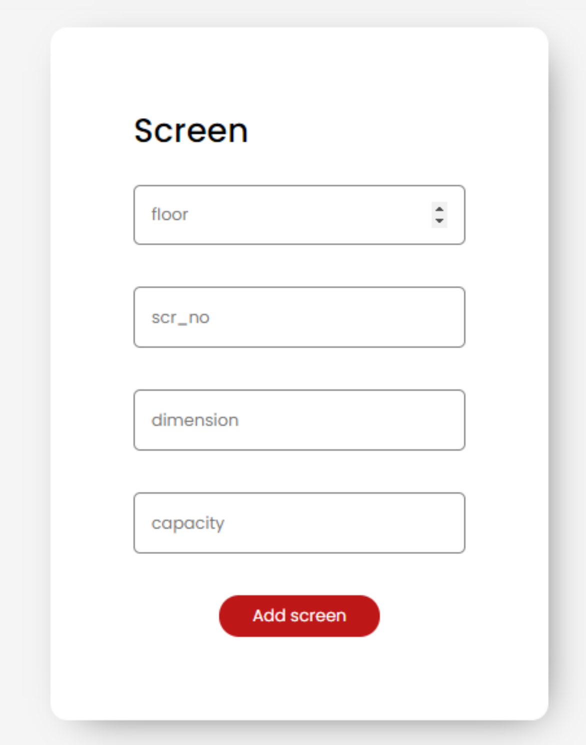
Add employee
Here you can add a new employee

Add maintains
Here you can assign departments to screens

The admin panel is password protected page that allows the administrator of the system to view and search current data , delete data and add new data such as:

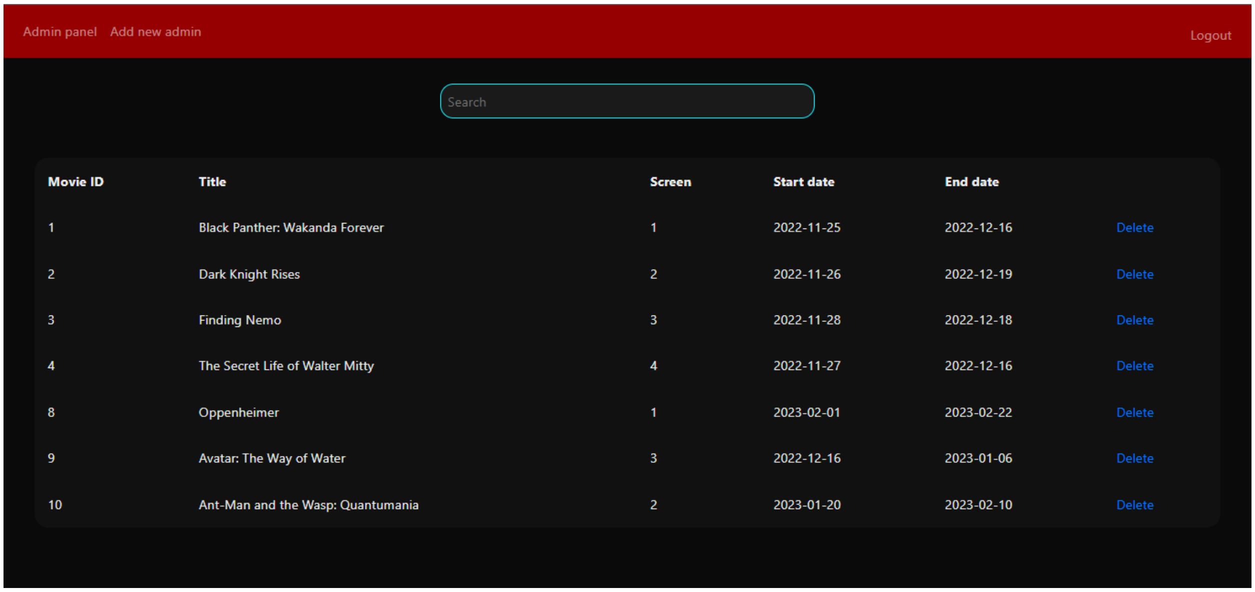
- Movies
- Screens
- Booked tickets
- Transactions made
- Employees
- Departments
- Users/ customers





It also shows then number of tickets booked on the current day and the collection for the day.

The admin panel also shows you at a glance the number of entries to the various entities in the database. Clicking on one of the entities will allow you to search, view or delete the current data for that entity. For example, clicking on ‘Movies’ will show the following page.



Movie ID	Title	Screen	Start date	End date	
1	Black Panther: Wakanda Forever	1	2022-11-25	2022-12-16	Delete
2	Dark Knight Rises	2	2022-11-26	2022-12-19	Delete
3	Finding Nemo	3	2022-11-28	2022-12-18	Delete
4	The Secret Life of Walter Mitty	4	2022-11-27	2022-12-16	Delete
8	Oppenheimer	1	2023-02-01	2023-02-22	Delete
9	Avatar: The Way of Water	3	2022-12-16	2023-01-06	Delete
10	Ant-Man and the Wasp: Quantumania	2	2023-01-20	2023-02-10	Delete

DDL statements :

```
CREATE database booking;

CREATE TABLE customer(
    phone bigint PRIMARY KEY,
    gender ENUM('m', 'f', 'o') default 'm',
    fname varchar(20) NOT null,
    lname varchar(20) NOT null,
    bdate date NOT null,
    age int ,
    hashp varchar(100)
)ENGINE=INNODB;

CREATE TABLE employee(
    empid int AUTO_INCREMENT PRIMARY KEY,
    gender varchar(1) NOT null,
    fname varchar(20) NOT null,
    lname varchar(20) NOT null,
    street varchar(20) NOT null,
    state varchar(20) NOT null,
    city varchar(20) NOT null,
    salary bigint NOT null,
    dno int ,
    adder_id int(4) not null
)ENGINE=INNODB;

CREATE TABLE transaction(
    trid varchar(11) PRIMARY KEY,
    mode_of_pay varchar(20) NOT null,
    price int NOT null,
    tdate date NOT null,
    cust_phone bigint ,
   time time
)ENGINE=INNODB;

CREATE TABLE ticket(
    tic_id varchar(11) PRIMARY KEY,
    seat_no int NOT null,
    movie_id int NOT null,
    mtime varchar(10) NOT null,
    mdate date NOT null,
    tid  varchar(11) NOT NULL
)ENGINE=INNODB;
```

```
CREATE TABLE movie(
    mid int AUTO_INCREMENT PRIMARY KEY,
    screen_no int NOT null,
    duration_mins int NOT null,
    rating varchar(10),
    title longtext not null,
    genre varchar(50),
    description longtext,
    cast varchar(250),
    start_date date not null,
    end_date date not null,
    trailer_link varchar(250) not null,
    language varchar(10) NOT null
)ENGINE=INNODB;

CREATE TABLE screen(
    floor int not null,
    scr_no int AUTO_INCREMENT PRIMARY KEY,
    dimension varchar(10) NOT null,
    capacity int NOT null
)ENGINE=INNODB;

CREATE TABLE department(
    Dnumber int AUTO_INCREMENT PRIMARY KEY,
    type varchar(20) NOT null
)ENGINE=INNODB;

CREATE TABLE maintains(
    Dno int  ,
   scr_no int,
CONSTRAINT pk_maintains PRIMARY KEY (Dno, scr_no)
)ENGINE=INNODB;

CREATE TABLE `admins` (
  `id` int(4) auto_increment PRIMARY KEY,
  `username` varchar(12) NOT NULL,
  `hash` varchar(100) NOT NULL
) ENGINE= INNODB;



create table customer_audit(
    id BIGINT PRIMARY KEY,
    fname varchar(20) NOT null,
    lname varchar(20) NOT null,
    registered date NOT null,
    reg_time varchar(20)
)ENGINE=INNODB;


create table movie_history(
    id int PRIMARY KEY,
    title varchar(50) not null,
    lang varchar(10) not null,
    genre varchar(50),
    deleted date NOT null,
    del_time varchar(10),
    action varcahr(10) NOT NULL DEFAULT 'deleted'
)ENGINE=INNODB;






ALTER TABLE transaction
   ADD CONSTRAINT fk_cust_ph FOREIGN KEY (cust_phone)
            REFERENCES customer (phone)
            ON DELETE CASCADE
            ON UPDATE CASCADE;


ALTER TABLE ticket
   ADD CONSTRAINT fk_trid FOREIGN KEY (tid)
            REFERENCES transaction (trid)
            ON DELETE CASCADE
            ON UPDATE CASCADE;

ALTER TABLE ticket
   ADD CONSTRAINT fk_movie_id FOREIGN KEY (movie_id)
            REFERENCES movie (mid)
            ON DELETE CASCADE
            ON UPDATE CASCADE;


ALTER TABLE movie
   ADD CONSTRAINT fk_screen_no FOREIGN KEY (screen_no)
            REFERENCES screen (scr_no)
```

```
        ON DELETE CASCADE
        ON UPDATE CASCADE;

ALTER TABLE maintains
  ADD CONSTRAINT fk_scr_no FOREIGN KEY (scr_no)
    REFERENCES screen (scr_no)
        ON DELETE CASCADE
        ON UPDATE CASCADE;

ALTER TABLE maintains
  ADD CONSTRAINT fk_dno FOREIGN KEY (Dno)
        REFERENCES department (Dnumber)
        ON DELETE CASCADE
        ON UPDATE CASCADE;


ALTER TABLE employee
  ADD CONSTRAINT fk_dept_no FOREIGN KEY (Dno)
        REFERENCES department (Dnumber)
        ON DELETE set null
        ON UPDATE CASCADE;

ALTER TABLE employee
  ADD CONSTRAINT fk_admin_no FOREIGN KEY (adder_id)
        REFERENCES admins (id)
        ON DELETE set null
        ON UPDATE CASCADE;
```

TRIGGERS

```
DELIMITER |
CREATE TRIGGER bdate_insert BEFORE INSERT ON customer
  FOR EACH ROW BEGIN
    SET new.age=DATE_FORMAT(FROM_DAYS(DATEDIFF(NOW(), new.bdate)), '%Y') + 0;
  END;
|
CREATE TRIGGER bdate_update BEFORE UPDATE ON customer
  FOR EACH ROW BEGIN
    SET new.age=DATE_FORMAT(FROM_DAYS(DATEDIFF(NOW(), new.bdate)), '%Y') + 0;
  END;
|
DELIMITER ;
```

```
CREATE TRIGGER before_customer_insert
                  AFTER INSERT
ON customer
   FOR EACH ROW
 INSERT INTO customer_audit
 SET id=new.phone,
   fname=new.fname,
   lname=new.lname,
   `registered`=CURRENT_DATE(),
   `reg_time`=CURTIME();

CREATE TRIGGER before_movie_delete
                  BEFORE DELETE
ON movie
   FOR EACH ROW
 INSERT INTO movie_history
 SET id=old.mid,
         title=old.title,
   `language`=old.`language`,
   `genre`=old.`genre`,
   `delete`=CURRENT_DATE(),
   `del_time`=CURTIME();
```

Views

```
CREATE VIEW ticketsPerMovie AS
SELECT `mid`,COUNT(seat_no) 'no of tickets'
FROM `movie`
inner join ticket
on `mid`=movie_id
GROUP BY mid;

CREATE VIEW deptEmployee AS
SELECT `Dnumber`,`type`,`empid`,`fname`,`lname`
FROM `department`
inner join employee
on dnumber=dno;

CREATE VIEW CollectionPerDay AS
SELECT `mdate` AS date,sum(price) AS 'income per day'
FROM `ticket`
join transaction
on tid = trid
GROUP BY `mdate`;
```

```sql
CREATE VIEW CollectionPerMovie AS
SELECT `movie_id`,title , sum(price) AS 'income per movie'
FROM `ticket`
inner join movie
on movie_id=mid
inner join transaction
on tid=trid
GROUP BY `movie_id`;

CREATE VIEW top3customer AS
SELECT `fname`,`lname`,SUM(price) AS total
FROM `customer`
inner JOIN transaction
ON phone=cust_phone
group by phone
order by total DESC, fname ASC
limit 0,3;

create view timedate AS
select `floor`,`scr_no`,`mdate`,`mtime`,count(mdate)as
occupied,`capacity`,(count(mdate)/capacity)*100 as percent_occupied
from `screen`
join `movie`
on`scr_no` = `screen_no`
join `ticket`
on`mid` =`movie_id`
group by mdate, mtime
order by mdate

CREATE VIEW MovieModerateDuration AS
select `mid`,`screen_no`,`duration_mins`,`title`
from movie
where duration_mins between 120 and 165;

CREATE VIEW empDept AS
SELECT `empid`,`fname`,`lname`,type as departmentType
FROM employee
left join department
on dno=dnumber
where dno=(select dnumber
             from department
          where type = 'sound');
```

```
CREATE VIEW longestMovie AS
SELECT `mid`,`title`,max(`duration_mins`)
from movie;

CREATE VIEW aboveAVGseats AS
SELECT `floor`,`scr_no`,`capacity`
from screen
where capacity>(select avg(`capacity`)
                from screen);

CREATE VIEW minsal AS
SELECT `empid`,`fname`,`lname`,`salary`
from employee
where salary=(select min(`salary`)
                from employee);
CREATE VIEW allNamer AS
SELECT `fname`,`lname`
FROM `employee`
UNION
SELECT `fname`,`lname`
FROM `customer`
ORDER BY `fname` ASC,`lname` ASC;

CREATE VIEW custNemp AS
SELECT `fname`,`lname`
FROM `employee`
INTERSECT
SELECT `fname`,`lname`
FROM `customer`
ORDER BY `fname` ASC,`lname` ASC;

CREATE VIEW empNeverCustomer AS
SELECT `fname`,`lname`
FROM `employee`
where `fname`
NOT IN  (SELECT `fname`
                      FROM `customer`)
ORDER BY `fname` ASC;

CREATE VIEW movieshowing AS
select `movie_id`,concat_ws(' ',`mdate`,`mtime`) AS `datetime`,`title`,
`duration_mins`,`screen_no` from `ticket`
 join`movie`
on`movie_id` = `mid`
```

```
group by concat_ws(' ',`mdate`,`mtime`);

CREATE VIEW nooftickets AS
select `trid`,count(`tid`) AS `no_of_tickets `mode_of_pay`,`booking`.`transaction`.`price` AS
`price`, `tdate`, `cust_phone`, `time`
from `transaction`
join ` ticket`
 on `trid` = ` tid`
group by ` tid`

create view tickertspercustomer as
SELECT `fname`,`lname`,`title`,mdate,mtime,count(mdate) as 'NO. of tickets'
FROM `customer`
inner join transaction
on `phone`=cust_phone
inner join ticket
on tid = trid
inner join movie
on mid = movie_id
group by mdate, mtime;

create view collectionPerMovieNDay AS
SELECT  `mdate`,title,sum(price) AS 'income'
FROM `ticket`
inner join movie
on movie_id=mid
inner join transaction
on tid=trid
GROUP BY `mdate`,`movie_id`;

create view curPlayingMovie AS
SELECT `mdate`,`mtime`,title
FROM `ticket`
join movie
on `movie_id`=mid
WHERE `mdate`=CURDATE() AND (CURRENT_TIME() between  `mtime` and
ADDTIME(`mtime`, "2:00:00"))
order by `mtime`;
```