MOVIE TICKET BOOKING

A MINI PROJECT REPORT

Submitted by

PRANEET MISHRA [RA2011051010069]
BHAMIDIPATI KAUSHIK [RA2011051010029]
AYUSH ATHARWA [RA2011051010032]

Under the guidance of

Dr. S. Sindhu

(Assistant professor, Data science and business systems)

In partial satisfaction of the requirements for the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING
With specialisation in <Gaming Technology>

SCHOOL OF COMPUTING

COLLEGE OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR - 603203

APRIL 2023

BONAFIDE CERTIFICATE

Certified that this project report **“ Movie Ticket Booking”** is the bonafide work of **“ PRANEET MISHRA [RA2011051010069], BHAMIDIPATI KAUSHIK [RA2011051010029], AYUSH ATHARWA [RA2011051010032]”** of III Year/VI Sem B.tech(CSE) who carried out the mini project work under my supervision for the course 18CSC303J-Database Management systems in SRM Institute of Science and Technology during the academic year 2022-2023(Even sem).

SIGNATURE

Dr. S. Sindhu
Assistant
Professor
Data Science
And Business
Systems

SIGNATURE

Dr. M. Lakshmi
HOD Department of Data Science
And Business Systems

AIM

The **Movie Ticket Booking System** Database Design is basically aimed to provide complete information of the movie and schedule to the customer, according to which he/she can easily get a ticket instantly and can book a ticket on his/her favorite movies. Admin can use Movie Ticketing System to insert and delete data such as movie descriptions, movie schedules which will update the related webpage and will be accessible by the customers. Admin can update the webpage changing according to the data in the database also admin can check the statistics information from the system. Operate your cinema with better efficiency by automating reservation and ticketing process improve profitability and manage your cinema better by having access to key data in a centralized and systematic view and increase customer satisfaction by giving your customers what they want when it comes to the seat preference.

Based on seat and ticket reservation system allowing booking in a few easy steps. This powerful movie ticketing system can be deployed on any website offering tickets for movies, theater, and other scheduled performances.

PROPOSED WORK DETAILS

The proposed Database consists of 6 tables that are interconnected. The team members work ontables and keep updating them by implementing queries.

DDL Statements:

In the context of SQL, data definition or data description language (DDL) is a syntaxfor creating and modifying database objects such as tables, indices, and users.

- CREATE to create a new table or database.
- ALTER for alteration.
- Truncate to delete data from the table.
- DROP to drop a table
- RENAME to rename a table.

1.Halls Table :-

create table hall(hall_id int,class varchar2(20),no_of_seats int)

Table created.

2.Movies Table:

create table movies(movie_id int,language varchar2(20),movie_name varchar2(20),type varchar2(20))

Table created.

3. Shows Table:-

create table shows(show_id int,movie_id int,day varchar2(20),duration int)

Table created.

4.Prices Table:-

create table prices(price_id int,day varchar2(20),amount int)

Table created.

5. Tickets Table:-

create table prices(price_id int,day varchar2(20),amount int)
Table created.

alter table tickets add movie_name varchar(20)
Table altered.

DML Statements

A data manipulation language (DML) is a computer programming language used foradding (inserting), deleting, and modifying (updating) data in a database.

INSERT – is used to insert data into a table.

UPDATE – is used to update existing data within a table.

DELETE – is used to delete records from a database table.

insert into hall values(1,'gold',50)

1 row(s) inserted

insert into hall values(2,'silver',75)

1 row(s) inserted

insert into hall values(3,'premium',25)

1 row(s) inserted.

insert into hall values(4,'standard',40)

1 row(s) inserted.

select * from hall

```
select * from hall
```

HALL_ID	CLASS	NO_OF_SEATS
1	gold	50
2	silver	75
3	premium	25
4	standard	40

Download CSV

4 rows selected.

insert into movies values(101,'Telugu','RRR','Action')

1 row(s) inserted.

insert into movies values(102,'Hindi','PK','Comedy')

1 row(s) inserted.

insert into movies values(103,'Tamil','Vikram','Action')

1 row(s) inserted.

insert into movies values(104,'Kannada','Charlie','Drama')

1 row(s) inserted.

select * from movies

```
select * from movies
```

MOVIE_ID	LANGUAGE	MOVIE_NAME	TYPE
101	Telugu	RRR	Action
102	Hindi	PK	Comedy
103	Tamil	Vikram	Action
104	Kannada	Charlie	Drama

Download CSV

4 rows selected.

insert into shows values(501,102,'Thursday',3)

1 row(s) inserted.

insert into shows values(502,101,'Friday',3)

1 row(s) inserted.

insert into shows values(503,103,'Wednesday',3)

1 row(s) inserted.

insert into shows Values(504,104,'Saturday',3)

1 row(s) inserted

select * from shows

```
select * from shows
```

SHOW_ID	MOVIE_ID	DAY	DURATION
501	102	Thursday	3
502	101	Friday	3
503	103	Wednesday	3
504	104	Saturday	3

Download CSV

4 rows selected.

insert into prices values(701,'Thursday',500)

1 row(s) inserted.

insert into prices values(702,'Friday',350)

1 row(s) inserted.

insert into prices values(703,'Wednesday',200)

1 row(s) inserted.

insert into prices values(704,'Sunday',750)

1 row(s) inserted.

insert into prices values(705,'M0nday',250)

1 row(s) inserted.

select * from prices

insert into tickets values(1001,5,45,'RRR')
1 row(s) inserted.
insert into tickets values(1002,7,39,'PK')
1 row(s) inserted.
insert into tickets values(1003,2,28,'Vikram')
1 row(s) inserted.
insert into tickets values(1004,10,55,'Vikram')
1 row(s) inserted.
insert into tickets values(1005,9,99,'RRR')
1 row(s) inserted.

```
select * from tickets
```

TICKET_ID	NOOFSEAT	SEATNO	MOVIE_NAME
1001	5	45	RRR
1002	7	39	PK
1003	2	28	Vikram
1004	10	55	Vikram
1005	9	99	RRR

Download CSV

5 rows selected.

update prices set amount=300 where price_id=703

1 row(s) updated.

delete from prices where amount=250

1 row(s) deleted.

delete from tickets where seatno=99

1 row(s) deleted.

```
select * from prices
```

PRICE_ID	DAY	AMOUNT
701	Thursday	500
702	Friday	350
703	Wednesday	300
704	Sunday	750

Download CSV

4 rows selected.

```
select * from tickets
```

TICKET_ID	NOOFSEAT	SEATNO	MOVIE_NAME
1001	5	45	RRR
1002	7	39	PK
1003	2	28	Vikram
1004	10	55	Vikram

Download CSV

4 rows selected.

commit

Statement processed.

INBUILT FUNCTIONS:

A built-in function is a function that is already available in a programming language, application, or another tool that can be accessed by end users.

```
select sum(amount) from prices
```

SUM(AMOUNT)
1900

Download CSV

```
select avg(duration) from shows
```

AVG(DURATION)
3

Download CSV

```
select max(noofseat) from tickets
```

MAX(NOOFSEAT)
10

Download CSV

```
select min(no_of_seats) from hall
```

MIN(NO_OF_SEATS)
25

Download CSV

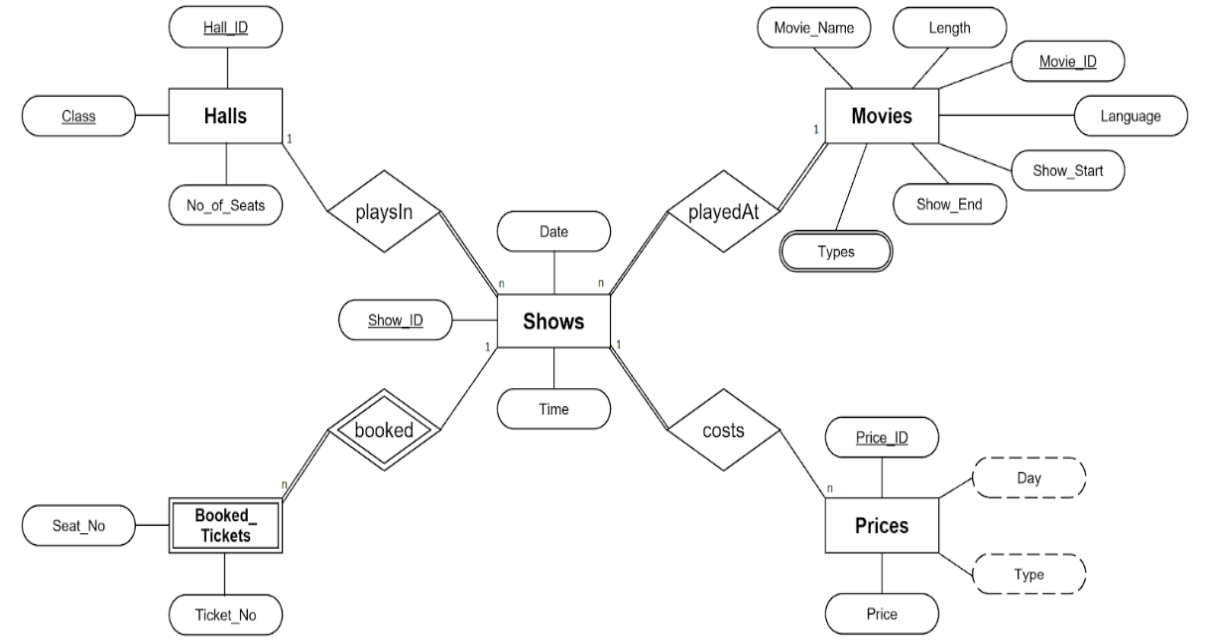
```
select current_timestamp from movies
```

CURRENT_TIMESTAMP
04-JUL-22 10.10.10.395640 AM US/PACIFIC
04-JUL-22 10.10.10.395640 AM US/PACIFIC
04-JUL-22 10.10.10.395640 AM US/PACIFIC
04-JUL-22 10.10.10.395640 AM US/PACIFIC

Download CSV

4 rows selected.

ER-Diagram:-


Hall_ID
Class
Halls
No_of_Seats
playsIn
Movie_Name
Length
Movie_ID
Movies
Language
Show_Start
Show_End
Types
playedAt
Date
Show_ID
Shows
Time
booked
costs
Seat_No
Booked_ Tickets
Ticket_No
Price_ID
Day
Prices
Type
Price

JOINS:-

A joins clause is used to combine rows from two or more tables, based on a related column between them.

Different Types of SQL Joins:

(INNER) JOIN: Returns records that have matching values in both tables

LEFT (OUTER) JOIN: Returns all records from the left table, and the matched records from the right table

RIGHT (OUTER) JOIN: Returns all records from the right table, and the matched records from the left table

FULL (OUTER) JOIN: Returns all records when there is a match in either left or right table.

NATURAL JOIN: Returns all records based on the common columns in the two tables being joined.

SELF JOIN: A table is joined with itself.

```
select movies.movie_name,movies.language,shows.duration from movies left join shows on movies.movie_id=shows.movie_id
```

MOVIE_NAME	LANGUAGE	DURATION
PK	Hindi	3
RRR	Telugu	3
Vikram	Tamil	3
Charlie	Kannada	3

Download CSV

4 rows selected.

```
select shows.day,shows.show_id,movies.type from shows join movies on shows.movie_id=movies.movie_id
```

DAY	SHOW_ID	TYPE
Friday	502	Action
Thursday	501	Comedy
Wednesday	503	Action
Saturday	504	Drama

Download CSV

4 rows selected.

```
select tickets.ticket_id,tickets.seatno,movies.movie_name from tickets cross join movies
```

TICKET_ID	SEATNO	MOVIE_NAME
1001	45	RRR
1002	39	RRR
1003	28	RRR
1004	55	RRR
1001	45	PK
1002	39	PK
1003	28	PK
1004	55	PK
1001	45	Vikram
1002	39	Vikram
1003	28	Vikram
1004	55	Vikram
1001	45	Charlie
1002	39	Charlie
1003	28	Charlie
1004	55	Charlie

Download CSV

16 rows selected.

SUBQUERIES:-

A query within a query is known as subquery.

```
select ticket_id,movie_name,seatno from tickets where noofseat>=(select noofseat from tickets where ticket_id=1004)
```

TICKET_ID	MOVIE_NAME	SEATNO
1004	Vikram	55

Download CSV

```
select movie_name,language from movies where movie_id in(select movie_id from movies where type='Action')
```

MOVIE_NAME	LANGUAGE
RRR	Telugu
Vikram	Tamil

Download CSV

2 rows selected.

```
select price_id,day from prices where amount>(select avg(amount) from prices)
```

PRICE_ID	DAY
701	Thursday
704	Sunday

Download CSV

2 rows selected.

PL/SQL PROGRAMS:

PROCEDURES-

```
CREATE OR REPLACE PROCEDURE today_is AS
BEGIN
DBMS_OUTPUT.PUT_LINE( 'Today is ' || TO_CHAR(SYSDATE, 'DL') );
END today_is;
```

```
Procedure created.
```

```
BEGIN
  today_is();
END;
```

```
Statement processed.
Today is Monday, July 04, 2022
```

FUNCTIONS:

```
create or replace function totalseats
return integer
as
total integer:=0;
begin
select sum(no_of_seats) into total from hall;
return total;
end totalseats;
```

```
Function created.
```

```
declare
answer integer;


begin
answer:=totalseats();
   dbms_output.put_line('Total seats in hall is ' || answer);
end;
```

```
Statement processed.
Total seats in hall is 190
```

TRIGGERS:

```
CREATE OR REPLACE TRIGGER display_price_changes
BEFORE DELETE OR INSERT OR UPDATE ON prices
FOR EACH ROW
WHEN (NEW.price_id > 0)
DECLARE
   price_diff number;
BEGIN
   price_diff := :NEW.amount  - :OLD.amount;
   dbms_output.put_line('Old price: ' || :OLD.amount);
   dbms_output.put_line('New price: ' || :NEW.amount);
   dbms_output.put_line('Price difference: ' || price_diff);
END;
```

```
Trigger created.
```

CURSORS:-

```
DECLARE
   total_rows number(2);
BEGIN
   UPDATE  prices
   SET amount = amount + 100;
   IF sql%notfound THEN
      dbms_output.put_line('no prices updated');
   ELSIF sql%found THEN
      total_rows := sql%rowcount;
      dbms_output.put_line( total_rows || ' prices updated ');
   END IF;
END;
```

```
Statement processed.
Old price: 500
New price: 600
Price difference: 100
Old price: 350
New price: 450
Price difference: 100
Old price: 300
New price: 400
Price difference: 100
Old price: 750
New price: 850
Price difference: 100
4 prices updated
```

SET OPERATORS:-

```
select movie_id from shows union select movie_id from movies
```

MOVIE_ID
101
102
103
104

Download CSV

4 rows selected.

```
select movie_id from shows union all select movie_id from movies
```

MOVIE_ID
102
101
103
104
101
102
103
104

Download CSV

8 rows selected

```
select movie_name from movies intersect select movie_name from tickets
```

MOVIE_NAME
PK
RRR
Vikram

Download CSV

3 rows selected.

```
select movie_name from movies minus select movie_name from tickets
```

MOVIE_NAME
Charlie

Download CSV

VIEWS:-

```
create view movieview as select movie_id,movie_name,language from movies
```

View created.

```
select * from movieview
```

MOVIE_ID	MOVIE_NAME	LANGUAGE
101	RRR	Telugu
102	PK	Hindi
103	Vikram	Tamil
104	Charlie	Kannada

Download CSV

4 rows selected.

```
delete from movieview where movie_id=103
```

1 row(s) deleted.

```
select * from movieview
```

MOVIE_ID	MOVIE_NAME	LANGUAGE
101	RRR	Telugu
102	PK	Hindi
104	Charlie	Kannada

Download CSV

3 rows selected.

```
drop view movieview
```

View dropped.

RESULTS:-

Successfully implemented Movie Ticket Booking system Database with all the necessary SQL Queries.