



Dr. B. R. Ambedkar National Institute of Technology, Jalandhar

**Fundamentals Of Database Management Project
(Theatre Management System)**

Submitted By :-

Love Sahajbir Singh- 19124035

Nishant Meena- 19124040

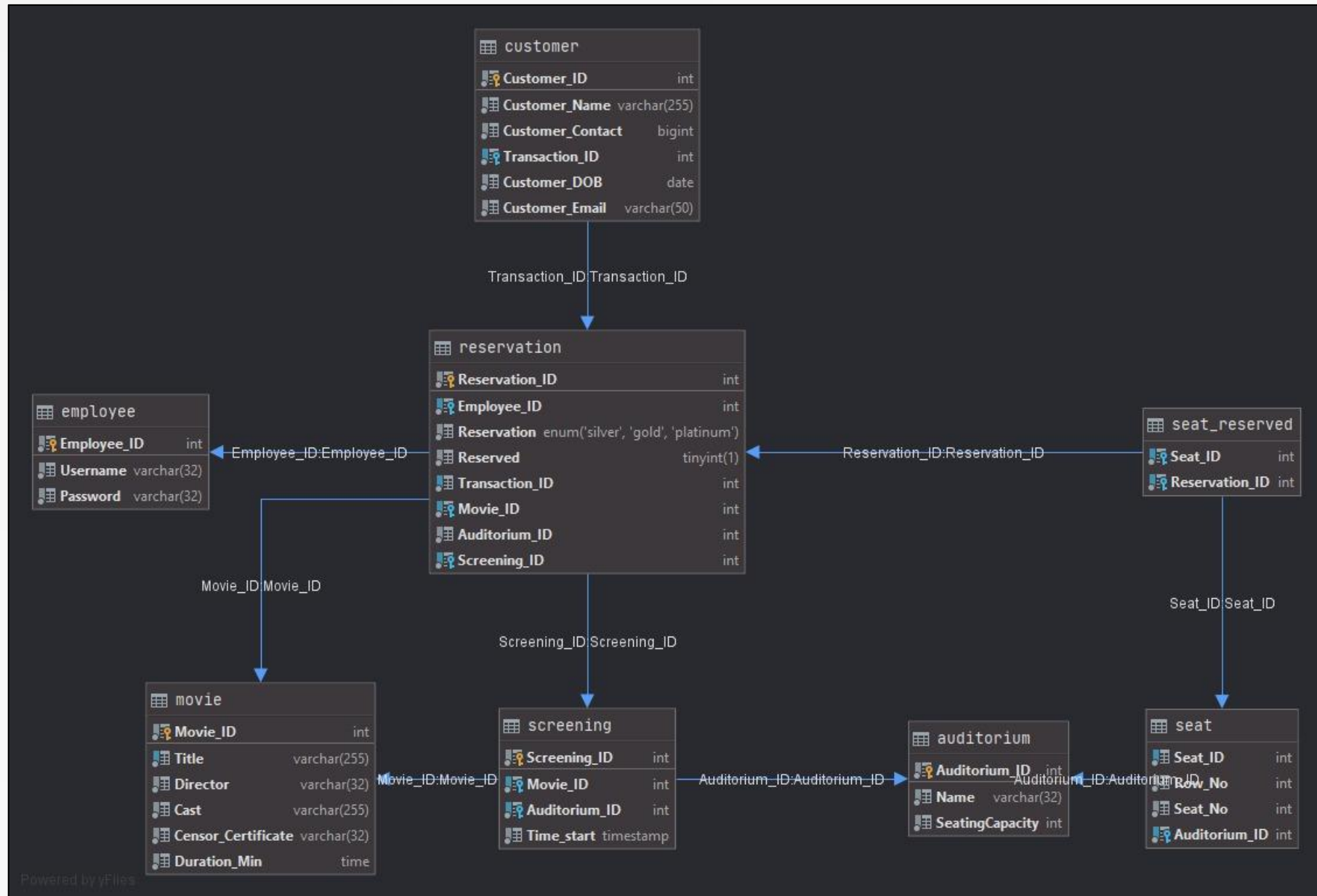
Submitted To :- Dr. Amritpal Singh

Introduction

- In this project , we have build a database for Theatre Management System.
- The main aim of this project is to store the details of Reservation, Customer, Movie,Employee, Auditorium, Screening, Seat, Seat Reserved.
- The main purpose of this type of database is to reduce the manual error involved in storing all these details. Due to automation, many loopholes that exist in the manual maintenance of the records can be removed.



Database Design



Creating Tables

**Create table Reservation (Reservation_ID
int auto_increment primary key,
Employee_ID int not null,
Reservation enum ('Silver', 'Gold',
'Platinum') default 'Silver' not null,
Reserved tinyint(1) default 0 not null,
Transaction_ID int not null,
Movie_ID int not null,
Auditorium_ID int not null,
Screening_ID int not null,
constraint
reservation_employee_Transaction_index
unique (Transaction_ID));**

```
mysql> Select* from Reservation;
```

Reservation_ID	Employee_ID	Reservation	Reserved	Transaction_ID	Movie_ID	Auditorium_ID	Screening_ID
101	2510	Silver	1	54678	1	1	139
102	2510	Silver	1	76543	1	1	139
103	2510	Silver	0	87765	1	1	139
104	2510	Gold	1	49869	1	1	139
105	2510	Gold	1	45698	1	1	139
106	2510	Gold	0	39849	1	1	139
107	2510	Platinum	1	45670	1	1	139
108	2510	Platinum	1	12987	1	1	139
109	2510	Platinum	0	72839	1	1	139
201	1205	Silver	1	83259	5	2	239
202	1205	Silver	1	76899	5	2	239
203	1205	Silver	0	56328	5	2	239
204	1205	Gold	1	50982	5	2	239
205	1205	Gold	1	57458	5	2	239
206	1205	Gold	0	52938	5	2	239
207	1205	Platinum	1	24378	5	2	239
208	1205	Platinum	1	63478	5	2	239
209	1205	Platinum	0	56788	5	2	239
301	2901	Silver	1	34780	8	3	339
302	2901	Silver	1	12378	8	3	339
303	2901	Silver	0	20394	8	3	339
304	2901	Gold	1	12347	8	3	339
305	2901	Gold	1	34798	8	3	339
306	2901	Gold	0	23489	8	3	339
307	2901	Platinum	1	39847	8	3	339
308	2901	Platinum	1	23748	8	3	339
309	2901	Platinum	0	39800	8	3	339

Create table customer
(Customer_ID int not null
primary key,
Customer_Name
varchar(255) not null,
Customer_Contact bigint not
null,
Transaction_ID int not null,
Customer_DOB date not null,
Customer_Email
varchar(50) not null,
constraint
Customer_Transaction_ID_u
index unique
(Transaction_ID));

```
mysql> Select* from Customer;
```

Customer_ID	Customer_Name	Customer_Contact	Transaction_ID	Customer_DOB	Customer_Email
1	Jiya	9559834463	54678	2003-12-02	jiya0212@gmail.com
2	Sahaj	6519160263	76543	2000-02-12	lovesahaj1225@protonmail.com
3	Anusha	8413049646	87765	2000-01-23	anusha2301@gmail.com
4	Bhavleen	7358958044	49869	1988-10-20	bhavleen2010@gmail.com
5	Naveen	8506972994	45698	1965-03-12	naveen1203@gmail.com
6	Aryan	9922528204	39849	1976-11-09	aryan0911@gmail.com
7	Surbhi	8131189118	45670	2012-08-30	surbhi3008@gmai.com
8	Manya	8844549128	12987	1991-01-18	manya1809@gmail.com
9	Ram	6286916457	72839	1960-02-28	ram2801@gmail.com
10	Shaurya	6031079825	83259	2009-09-05	shaurya342@gmail.com
11	Jahnavi	6719576060	76899	1990-12-05	jahnavi0512@gmail.com
12	Nishant	8869945831	56328	1985-10-25	nishant2510@gmail.com
13	Aanchal	7526550218	50982	1979-01-23	aanchal12301@gmail.com
14	Richa	6560319723	57458	1974-09-03	richa03019@gmail.com
15	Jatin	6719576060	52938	1982-06-24	jatin2401@gmail.com
16	Vivek	8979693405	24378	1970-09-21	vivek2120@gmail.com
17	Divya	6191852274	63478	1989-10-05	divya0401@gmail.com
18	Sonal	8837264694	56788	1976-07-17	sonal19198@gmail.com
19	Rajiv	7973786242	34780	1983-05-15	rajiv1234@gmail.com
20	Sumit	7088602806	12378	1988-11-25	sumit2551@gmail.com
21	Geeta	9571395450	20394	2000-07-24	geeta2113@gmail.com
22	Shivani	7772786691	12347	1997-09-25	shiv345@gmail.com
23	Sameer	8606019115	34798	1996-07-01	sam24@gmail.com
24	Reena	9089630929	23489	1990-05-04	reenam33@gmail.com
25	Bijoy	6235425961	39847	1997-04-21	bijoyy55@gmail.com
26	Ryan	7811169969	23748	1990-09-25	ryangold@gmail.com
27	Dev	6211269113	39800	1999-06-17	dev345@gmail.com

Create table Movie (Movie_ID int not null,
Title varchar(32) not null,
Director varchar(32) not null,
Cast varchar(255) not null,
Censor_Certificate varchar(32) not null,
Duration_Min time not null,
constraint PK_Movie primary key (Movie_ID));

Create table Employee (Employee_ID int not null,
Username varchar(32) not null,
Password varchar(32) not null,
constraint PK_Employee primary key
(Employee_ID));

Employee_ID	Username	Password
1205	Rajeev_Sharma	Rajeev@1205
2510	Suresh_Gupta	Suresh@2510
2901	Anil_Roy	Anil@2901

mysql> Select* from Movie;

Movie_ID	Title	Director	Cast	Censor_Certificate	Duration_Min
1	Jurassic Park	Steven Spielberg	Jeff Goldblum, Sam Neill, Laura Dern, Samuel L. Jackson, Richard Attenborough	PG-13	02:08:00
2	The Shining	Stanley Kubric	Jack Nicholson, Shelly Duvall, Danny Lloyd, Scatman Crothers, Joe Trukel	R	02:26:00
3	Good Will Hunting	Gus Van Sant	Matt Damon, Ben Affleck, Robin Williams, Casey Affleck, Minnie Driver	R	02:08:00
4	Knives Out	Rian Johnson	Ana de Armas, Chris Evans, Daniel Craig, Joseph Gordan-Levitt, Katherine Langford	PG-13	02:10:00
5	Dead Poet Society	Peter Weir	Robin Williams, Ethan Hawke, Robert Sean Leonard, Josh Charles, Gale Hansen	PG	02:08:00
6	The Godfather	Francis Ford Coppola	Al Pacino, Marlon Brando, James Caan, Diane Keaton, Talia Shire, John Cazale	R	02:55:00
7	The Lord of Rings	Peter Jackson	Elijah Wood, Iam McKellen, Orlando Bloom, Viggo Mortensen, Sean Astin, Sean Bean	PG-13	02:58:00
8	Star Wars - A New Hope	George Lucas	Carrie Fisher, James Earl Jones, Alec Guinness, Harrison Ford, Mark Hamil	PG	02:01:00
9	500 Days of Summer	Marc Webb	Joseph Gordan-Leviit, Zooey Deschanel, Matthew Fray Gubler, Chole Grace Moretz, Clark Gregg	PG-13	01:45:00
10	The Dark Knight	Christopher Nolan	Christian Bale, Gary Oldman, Heath Ledger, Micheal Cane, Morgan Freeman, Cillian Murphy	PG-13	02:32:00

Create table Auditorium
(Auditorium_ID int not null,
Name varchar(32) not null,
SeatingCapacity int not null,
constraint PK_Auditorium primary key
(Auditorium_ID));

Auditorium_ID	Name	SeatingCapacity
1	Auditorium_First	9
2	Auditorium_Second	9
3	Auditorium_Third	9

Create table Screening
(Screening_ID int not null,
Movie_ID int not null,
Auditorium_ID int not null,
Screening_Start timestamp,
constraint PK_Screening primary key
(Screening_ID));

Screening_ID	Movie_ID	Auditorium_ID	Screening_Start
139	1	1	2020-11-26 13:30:00
239	5	2	2020-11-26 14:00:00
339	8	3	2020-11-26 12:30:00

Create table Seat

(Seat_ID int not null,

Row_No int not null,

Seat_No int not null,

Auditorium_ID int not null,

constraint PK_Seat unique (Seat_ID));

```
mysql> Select* from Seat_Reserved;
```

Seat_ID	Reservation_ID
111	101
112	102
121	104
122	105
131	107
132	108
211	201
212	202
221	204
222	205
231	207
232	208
311	301
312	302
321	304
322	305
331	307
332	308

```
mysql> Select* from Seat;
```

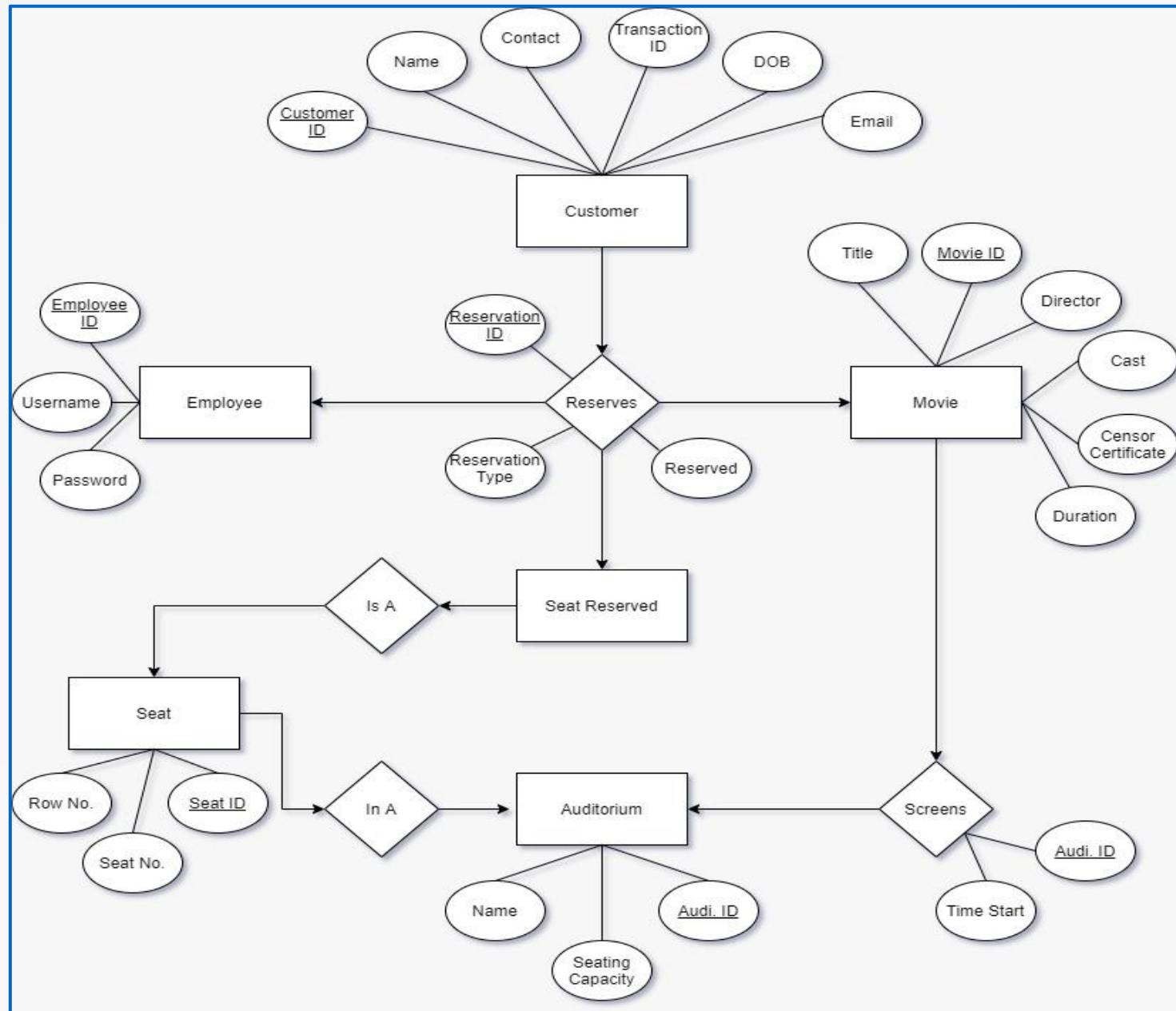
Seat_ID	Row_No	Seat_No	Auditorium_ID
111	1	1	1
112	1	2	1
113	1	3	1
121	2	1	1
122	2	2	1
123	2	3	1
131	3	1	1
132	3	2	1
133	3	3	1
211	1	1	2
212	1	2	2
213	1	3	2
221	2	1	2
222	2	2	2
223	2	3	2
231	3	1	2
232	3	2	2
233	3	3	2
311	1	1	2
312	1	2	2
313	1	3	2
321	2	1	2
322	2	2	2
323	2	3	2
331	3	1	2
332	3	2	2
333	3	3	2

Create table Seat_Reserved

(Seat_ID int not null,

Reservation_ID int not null);

ER Diagram



Descriptions of Database

Reservation Table

Field	Type	Null	Key	Default	Extra
Reservation_ID	int	NO	PRI	NULL	auto_increment
Employee_ID	int	NO		NULL	
Reservation Reserved	enum('Silver','Gold','Platinum') tinyint(1)	NO		Silver 0	
Transaction_ID	int	NO	UNI	NULL	
Movie_ID	int	NO		NULL	
Auditorium_ID	int	NO		NULL	
Screening_ID	int	NO		NULL	

Customer Table

Field	Type	Null	Key	Default	Extra
Customer_ID	int	NO	PRI	NULL	
Customer_Name	varchar(255)	NO		NULL	
Customer_Contact	bigint	NO		NULL	
Transaction_ID	int	NO	UNI	NULL	
Customer_DOB	date	NO		NULL	
Customer_Email	varchar(50)	NO		NULL	

Employee Table

Field	Type	Null	Key	Default	Extra
Auditorium_ID	int	NO	PRI	NULL	
Name	varchar(32)	NO		NULL	
SeatingCapacity	int	NO		NULL	

Movie Table

Field	Type	Null	Key	Default	Extra
Employee_ID	int	NO	PRI	NULL	
Username	varchar(32)	NO		NULL	
Password	varchar(32)	NO		NULL	

Auditorium Table

Field	Type	Null	Key	Default	Extra
Auditorium_ID	int	NO	PRI	NULL	
Name	varchar(32)	NO		NULL	
SeatingCapacity	int	NO		NULL	

Screening Table

Field	Type	Null	Key	Default	Extra
Screening_ID	int	NO	PRI	NULL	
Movie_ID	int	NO		NULL	
Auditorium_ID	int	NO		NULL	
Screening_Start	timestamp	YES		NULL	

Seat Table

Field	Type	Null	Key	Default	Extra
Seat_ID	int	NO	PRI	NULL	
Row_No	int	NO		NULL	
Seat_No	int	NO		NULL	
Auditorium_ID	int	NO		NULL	

Seat Reserved Table

Field	Type	Null	Key	Default	Extra
Seat_ID	int	NO		NULL	
Reservation_ID	int	NO		NULL	

Assigning Foreign Key

```
mysql> Alter table Reservation
->      add constraint reservation_employee_Employee_ID_fk
->      foreign key (Employee_ID) references employee (Employee_ID);
Query OK, 27 rows affected (0.09 sec)
Records: 27 Duplicates: 0 Warnings: 0

mysql>
mysql> Alter table Reservation
->      add constraint reservation_movie_Movie_ID_fk
->      foreign key (Movie_ID) references movie (Movie_ID);
Query OK, 27 rows affected (0.07 sec)
Records: 27 Duplicates: 0 Warnings: 0

mysql>
mysql> Alter table Reservation
->      add constraint reservation_screening_Screening_ID_fk
->      foreign key (Screening_ID) references screening (Screening_ID);
Query OK, 27 rows affected (0.07 sec)
Records: 27 Duplicates: 0 Warnings: 0

mysql>
mysql> Alter table seat_reserved
-> add constraint seat_reserved_reservation_Reservation_ID_fk
-> foreign key (Reservation_ID) references reservation (Reservation_ID);
Query OK, 18 rows affected (0.06 sec)
Records: 18 Duplicates: 0 Warnings: 0

mysql>
mysql> Alter table seat_reserved
-> add constraint seat_reserved_seat_Seat_ID_fk
-> foreign key (Seat_ID) references seat (Seat_ID);
Query OK, 18 rows affected (0.07 sec)
Records: 18 Duplicates: 0 Warnings: 0

mysql>
mysql> Alter table seat
-> add constraint seat_auditorium_Auditorium_ID_fk
-> foreign key (Auditorium_ID) references auditorium (Auditorium_ID);
Query OK, 27 rows affected (0.06 sec)
Records: 27 Duplicates: 0 Warnings: 0
```

```
mysql> Alter table screening
-> add constraint screening_auditorium_Auditorium_ID_fk
-> foreign key (Auditorium_ID) references auditorium (Auditorium_ID);
Query OK, 3 rows affected (0.07 sec)
Records: 3 Duplicates: 0 Warnings: 0

mysql>
mysql> Alter table screening
-> add constraint screening_movie_Movie_ID_fk
-> foreign key (Movie_ID) references movie (Movie_ID);
Query OK, 3 rows affected (0.07 sec)
Records: 3 Duplicates: 0 Warnings: 0

mysql>
mysql> Alter table Customer
->      add constraint customer_reservation_Transaction_ID_fk
->      foreign key (Transaction_ID) references reservation (Transaction_ID);
Query OK, 27 rows affected (0.06 sec)
Records: 27 Duplicates: 0 Warnings: 0
```


Queries

Display all details from Reservation Table with Platinum Reservation Type.

```
mysql> Select * from Reservation where Reservation='Platinum';
```

Reservation_ID	Employee_ID	Reservation	Reserved	Transaction_ID	Movie_ID	Auditorium_ID	Screening_ID
107	2510	Platinum	1	45670	1	1	139
108	2510	Platinum	1	12987	1	1	139
109	2510	Platinum	0	72839	1	1	139
207	1205	Platinum	1	24378	5	2	239
208	1205	Platinum	1	63478	5	2	239
209	1205	Platinum	0	56788	5	2	239
307	2901	Platinum	1	39847	8	3	339
308	2901	Platinum	1	23748	8	3	339
309	2901	Platinum	0	39800	8	3	339

Display Transaction_ID with False Reservation Status.

```
mysql> Select Transaction_ID from Reservation where Reserved=False;
```

Transaction_ID
87765
39849
72839
56328
52938
56788
20394
23489
39800

Change Employee_ID size to 8 and change column Duration_Min column to Duration_In_Hours.

```
mysql> Alter table Reservation modify Employee_ID int(8) not null;
Query OK, 0 rows affected, 1 warning (0.06 sec)
Records: 0 Duplicates: 0 Warnings: 1

mysql>
mysql> Alter table Movie rename column Duration_Min to Duration_In_Hours ;
Query OK, 0 rows affected (0.06 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

Display Transaction_ID which contains 67 in it.

```
mysql> Select Transaction_ID from Reservation where Transaction_ID like '%67%';
+-----+
| Transaction_ID |
+-----+
|          45670 |
|          54678 |
|          56788 |
+-----+
```

Display Reservation Status with starting Transaction_ID as 87.

```
mysql> Select Reserved from Reservation where Transaction_ID like '87__';
```

Reserved
0

Display distinct Employee_ID.

```
mysql> Select distinct Employee_Id from Reservation;
```

Employee_Id
2510
1205
2901

Display Reservation_ID, Employee_ID and Transaction_ID with True Reservation Status and Transaction_ID greater than 80000.

```
mysql> Select Reservation_ID, Employee_ID ,Transaction_ID from Reservation where Reserved=True And Transaction_ID>=80000;
+-----+-----+-----+
| Reservation_ID | Employee_ID | Transaction_ID |
+-----+-----+-----+
|          201  |         1205 |          83259 |
+-----+-----+-----+
```

Display Movie Titles with PG-13 Censor_Certificate or Duration less than 2 hours.

```
mysql> Select Title from Movie Where Censor_Certificate='PG-13' or Duration_In_Hours<'02:00:00';
+-----+
| Title |
+-----+
| Jurassic Park |
| Knives Out |
| The Lord of Rings |
| 500 Days of Summer |
| The Dark Knight |
+-----+
5 rows in set (0.00 sec)

mysql> _
```

Display Movie Titles Alphabetically.

```
mysql> Select* from Movie order by Title;
```

Movie_ID	Title	Director	Cast	Censor_Certificate	Duration_In_Hours
9	500 Days of Summer	Marc Webb	Joseph Gordan-Leviit, Zooey Deschanel, Matthew Fray Gubler, Chole Grace Moretz, Clark Gregg	PG-13	01:45:00
5	Dead Poet Society	Peter Weir	Robin Williams, Ethan Hawke, Robert Sean Leonard, Josh Charles, Gale Hansen	PG	02:08:00
3	Good Will Hunting	Gus Van Sant	Matt Damon, Ben Affleck, Robin Williams, Casey Affleck, Minnie Driver	R	02:08:00
1	Jurassic Park	Steven Spielberg	Jeff Goldblum, Sam Neill, Laura Dern, Samuel L. Jackson, Richard Attenborough	PG-13	02:08:00
4	Knives Out	Rian Johnson	Ana de Armas, Chris Evans, Daniel Craig, Joseph Gordan-Levitt, Katherine Langford	PG-13	02:10:00
8	Star Wars - A New Hope	George Lucas	Carrie Fisher, James Earl Jones, Alec Guinness, Harrison Ford, Mark Hamil	PG	02:01:00
10	The Dark Knight	Christopher Nolan	Christian Bale, Gary Oldman, Heath Ledger, Micheal Cane, Morgan Freeman, Cillian Murphy	PG-13	02:32:00
6	The Godfather	Francis Ford Coppola	Al Pacino, Marlon Brando, James Caan, Diane Keaton, Talia Shire, John Cazale	R	02:55:00
7	The Lord of Rings	Peter Jackson	Elijah Wood, Iam McKellen, Orlando Bloom, Viggo Mortensen, Sean Astin, Sean Bean	PG-13	02:58:00
2	The Shining	Stanley Kubric	Jack Nicholson, Shelly Duvall, Danny Lloyd, Scatman Crothers, Joe Trukel	R	02:26:00

Display Censor_Certificate of Movie with Movie_ID 1,3,5.

```
mysql> Select Censor_Certificate from Movie where Movie_ID In(1,3,5);
```

Censor_Certificate
PG-13
R
PG

Display Row_No and Seat_No with Seat_ID between 213 and 231.

```
mysql> Select Row_No,Seat_No from Seat where Seat_ID between 213 And 231;
```

Row_No	Seat_No
1	3
2	1
2	2
2	3
3	1

Display total reserved seats.

```
mysql> Select Sum(Reserved) as Total_Reserved_Seats from Reservation;
```

Total_Reserved_Seats
18

Count how many times each Employee has done reservations.

```
mysql> Select Employee_ID, Count(Employee_ID) from Reservation Group By Employee_ID;
```

Employee_ID	Count(Employee_ID)
1205	9
2510	9
2901	9

Display greatest and smallest Screening_ID.

```
mysql> Select Max(Screening_ID) as Smallest_Screening_ID, Min(Screening_ID) as Greatest_Screening_ID from Screening;
```

Smallest_Screening_ID	Greatest_Screening_ID
339	139

Display Employee's Usernames in capital letters.

```
mysql> Select upper Username from Employee;
```

```
+-----+
| upper(Username) |
+-----+
| RAJEEV_SHARMA   |
| SURESH_GUPTA    |
| ANIL_ROY        |
+-----+
```

Display Movie Titles with 'A' in it.

```
mysql> Select instr(Title,'A') from Movie;
```

```

+-----+
| instr(Title, 'A') |
+-----+
| 4 |
| 0 |
| 0 |
| 0 |
| 3 |
| 9 |
| 0 |
| 3 |
| 6 |
| 6 |
+-----+

```

Display Transaction_Id and a failure message when reservation is failed.

```
mysql> Select Transaction_ID , rpad(Reserved ,20,' Reservation Failed') as Reservation_Status from Reservation where Reserved=False;
```

Transaction_ID	Reservation_Status
87765	0 Reservation Failed
39849	0 Reservation Failed
72839	0 Reservation Failed
56328	0 Reservation Failed
52938	0 Reservation Failed
56788	0 Reservation Failed
20394	0 Reservation Failed
23489	0 Reservation Failed
39800	0 Reservation Failed

Display login details of Employees.

```
mysql> Select concat(Username,Password) as Login_Deatils from Employee;
```

Login_Deatils
Rajeev_SharmaRajeev@1205
Suresh_GuptaSuresh@2510
Anil_RoyAnil@2901

Display Movie Title which is being presented in Auditorium_Second.

```
mysql> Select Title from Movie where Movie_ID in (Select Movie_ID from Reservation where Auditorium_ID=2);
```

Title
Dead Poet Society

Display Username of Employee who is looking after Reservation of Jurassic Park.

```
mysql> Select Username from Employee where Employee_ID in (Select Employee_ID from Reservation where Movie_ID=(Select Movie_ID from Movie where Title='Jurassic Park'));
```

Username
Suresh_Gupta

Display Name of the Auditorium where Screening of Movie starts on 2020-11-26 at 14:00:00.

```
mysql> Select Name from Auditorium where Auditorium_ID in (Select Auditorium_ID from Screening where Screening_Start='2020-11-26,14:00:00');
+-----+
| Name |
+-----+
| Auditorium_Second |
+-----+
```

Display Customer_Name who has booked for Silver and Gold Reservation.

```
mysql>
mysql> Select Customer_Name from Customer where Transaction_ID in (Select Transaction_ID from Reservation where Reservation in ('Silver','Gold'));
+-----+
| Customer_Name |
+-----+
| Jiya |
| Sahaj |
| Anusha |
| Bhavleen |
| Naveen |
| Aryan |
| Shaurya |
| Jahnavi |
| Nishant |
| Aanchal |
| Richa |
| Jatin |
| Rajiv |
| Sumit |
| Geeta |
| Shivani |
| Sameer |
| Reena |
+-----+
```

Display name and contact no. of people in auditorium first.

```
mysql> Select c.Customer_Name, c.Customer_Contact from reservation r join auditorium a on r.Auditorium_ID = a.Auditorium_ID  
-> join customer c on r.Transaction_ID = c.Transaction_ID where a.Name = 'Auditorium_First';
```

Customer_Name	Customer_Contact
Jiya	9559834463
Sahaj	6519160263
Anusha	8413049646
Bhavleen	7358958044
Naveen	8506972994
Aryan	9922528204
Surbhi	8131189118
Manya	8844549128
Ram	6286916457

Display movie title which stars Robin Williams and start after 4 pm.

```
mysql> Select distinct m.Title from movie m join screening s on m.Movie_ID = s.Movie_ID  
-> where m.Cast like '%Robin Williams%' and s.Screening_Start > '16:00:00';
```

Title
Dead Poet Society

Display Employee's Username who booked for Star Wars - A New Hope

```
mysql> Select distinct e.Username from employee e join reservation r on e.Employee_ID = r.Employee_ID  
-> join movie m on r.Movie_ID = m.Movie_ID where m.Title = 'Star Wars - A New Hope';
```

```
+-----+  
| Username |  
+-----+  
| Anil_Roy |  
+-----+
```

Display Seat_ID in Auditorium_Second.

```
mysql> Select distinct s.Seat_ID from seat s join auditorium a on s.Auditorium_ID = a.Auditorium_ID  
-> where a.Name = 'Auditorium_Second';
```

```
+-----+  
| Seat_ID |  
+-----+  
| 211 |  
| 212 |  
| 213 |  
| 221 |  
| 222 |  
| 223 |  
| 231 |  
| 232 |  
| 233 |  
| 311 |  
| 312 |  
| 313 |  
| 321 |  
| 322 |  
| 323 |  
| 331 |  
| 332 |  
| 333 |  
+-----+
```

Create reservation and seat reserved view.

```
mysql> Create view reservation_seat as select distinct reservation.*, seat_reserved.Seat_ID  
-> from reservation join seat_reserved on reservation.Reservation_ID = seat_reserved.Reservation_ID;  
Query OK, 0 rows affected (0.06 sec)
```

Display view Reservation_Seat.

```
mysql> Select* from Reservation_Seat;
```

Reservation_ID	Employee_ID	Reservation	Reserved	Transaction_ID	Movie_ID	Auditorium_ID	Screening_ID	Seat_ID
101	2510	Silver	1	54678	1	1	139	111
102	2510	Silver	1	76543	1	1	139	112
104	2510	Gold	1	49869	1	1	139	121
105	2510	Gold	1	45698	1	1	139	122
107	2510	Platinum	1	45670	1	1	139	131
108	2510	Platinum	1	12987	1	1	139	132
201	1205	Silver	1	83259	5	2	239	211
202	1205	Silver	1	76899	5	2	239	212
204	1205	Gold	1	50982	5	2	239	221
205	1205	Gold	1	57458	5	2	239	222
207	1205	Platinum	1	24378	5	2	239	231
208	1205	Platinum	1	63478	5	2	239	232
301	2901	Silver	1	34780	8	3	339	311
302	2901	Silver	1	12378	8	3	339	312
304	2901	Gold	1	12347	8	3	339	321
305	2901	Gold	1	34798	8	3	339	322
307	2901	Platinum	1	39847	8	3	339	331
308	2901	Platinum	1	23748	8	3	339	332

Create index movie_title_index on movie(Title).

```
mysql> Create index movie_title_index on movie(Title);
Query OK, 0 rows affected (0.07 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

Display Index

```
mysql> Show indexes from Movie;
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
movie	0	PRIMARY	1	Movie_ID	A	10	NULL	NULL		BTREE			YES	NULL
movie	1	movie_title_index	1	Title	A	10	NULL	NULL		BTREE			YES	NULL



Thank You!!!