Project Title: Movie Theatre Management System

Group Number: 04

Section: D

Supervisor: Kawser Irom Rushee

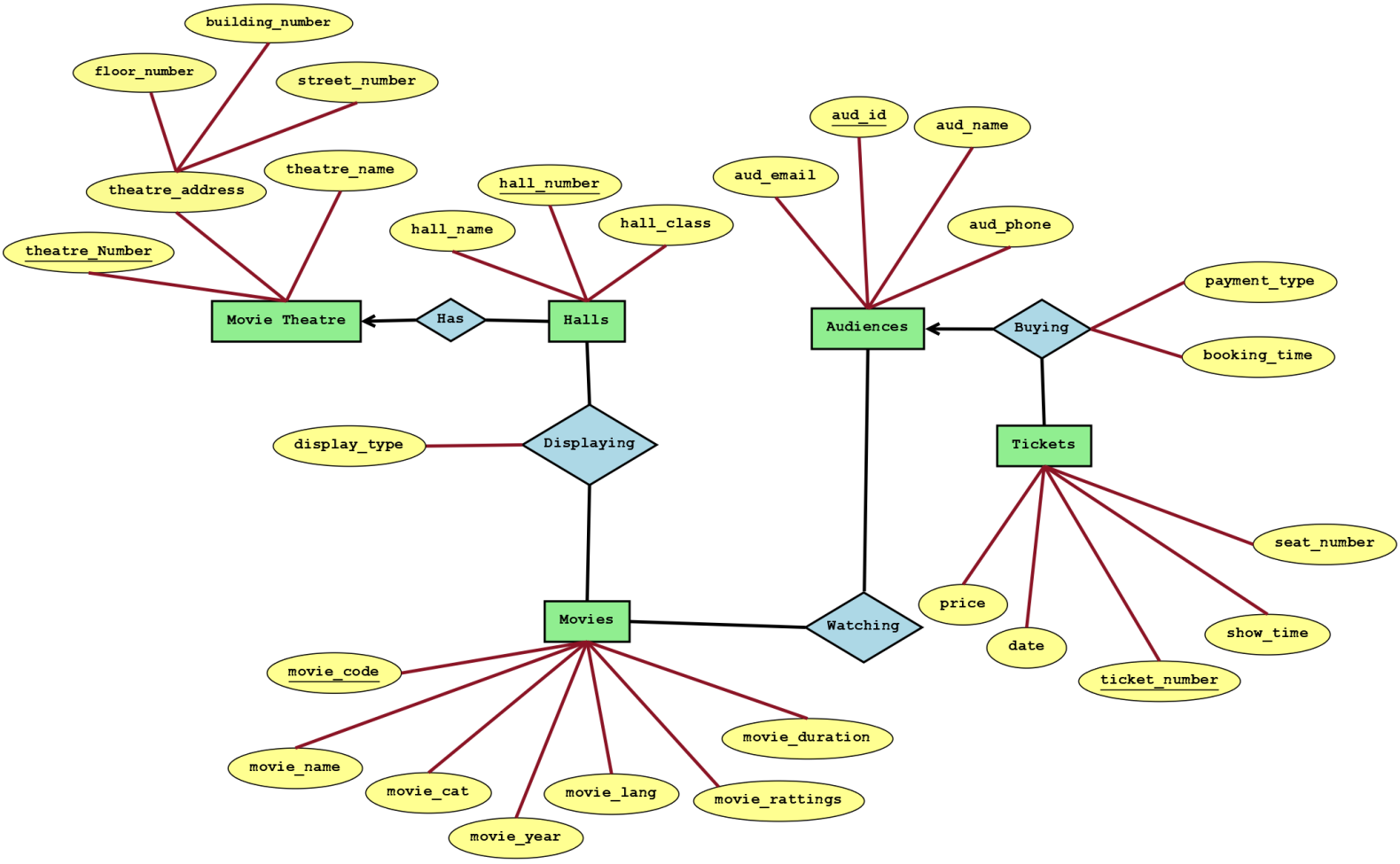
Group Members:

Name	Id
MD. Mahfuzur Rahman	22-47078-1
Nowshin Fariha	22-47074-1
Monisha Sarkar	22-47063-1

Case Study

In a movie theatre there are many halls to display many movies. A movies theatre is identified by theatre address, theatre name and unique theatre number. Theatre address is composed of street number, building number and floor number. Every hall contains a unique hall number, hall name and hall_ class. Every hall display with display type and specific screen time. Movies have attributes named movie name, movie category, movie year, movie language, movie ratting, movie duration and unique movie code. In the movie theatre management system an audience can buy many tickets but one ticket can be bought by exactly one audience. System stores audience name, audience email and audience phone number and unique audience id for audience details. While buying tickets system stores payment type and booking time. A ticket is identified by unique ticket number, date, show time, ticket price and seat number. Audience may watch movies in the specified date and time of the tickets.

ER Diagram


building_number
floor_number
street_number
theatre_address
theatre_name
theatre_Number
Movie Theatre
Has
Halls
hall_number
hall_name
hall_class
aud_id
aud_name
aud_email
aud_phone
Audiences
Buying
payment_type
booking_time
Tickets
seat_number
price
date
show_time
ticket_number
Watching
Movies
Displaying
display_type
movie_code
movie_name
movie_cat
movie_year
movie_lang
movie_rattings
movie_duration

NORMALIZATION

Has:

UNF:1st: theatre_number, floor_number, building _number, street_number, theatre_name, hall_name, hall_number, hall_class.

1NF:1st: theatre_number, hall_number, floor_number, building_number, street_number, theatre_name, hall_name, hall_class.

2NF: 1st: hall_number, hall_name, hall_class, theatre_number.

2nd: theatre_number, floor_number, building_number, street_number, theatre_name.

3NF: 1st: hall_number, hall_name, theatre_number.

2nd: hall_name, hall_class.

3rd: theatre_number, theatre_name , floor_number, building_number, street_number.

BUYING :

UNF: 1st: aud_id, aud_name, aud_phone, aud_email, ticket_number, price, date, show_time, seat_number, payment_type, booking_time.

1NF: 1st: aud_id, aud_name, aud_phone, aud_email, ticket number, price, show_time, date, seat_number, payment_type, booking_time.

2NF: 1st: aud_id, aud_name, aud_phone, aud_email.

2nd: ticket_number, price, date, show time, seat number.

3rd: serial_no,aud_id, ticket_number, payment type, booking time.

3NF: 1st: aud_id, aud_name, aud_phone, aud_email.

2nd: ticket_number, price, show_time, date, seat_number.

3rd: serial_no, aud_id, ticket_number, payment_type, booking_time.

WATCHING

UNF: 1st: aud_id, aud_name, aud_phone,aud_email, movie_code, movie_name, movie_cat, movie_year, movie _lang, movie_rattings, movie_duration.

1NF: 1st: aud_id, aud_name, aud_phone,aud_email, movie_code, movie_name, movie_cat, movie_year, movie_lang, movie_rattings, movie_duration.

2NF: 1st: aud_id ,aud_name, aud_phone,aud_email

2nd: movie code, movie_name, movie_cat, movie_year, movie_lang, movie _rattings, movie_duration.

3rd:serial_no,aud_id, movie_code.

3NF: 1st: aud_Id, aud_name, aud_phone,aud_email

2nd: movie_code, movie_name, movie_category, movie_year, movie_lang, movie_rattings, movie_duration.

3rd: Serial_no, audience_id, movie_code.

DISPLAYING

UNF: 1st: hall number, hall name, hall_class, movie code, movie_name, movie_cat, movie_year, movie_lang, movie_ratings, movie_duration, display_type.

1NF: 1st: hall number, hall name, hall_class, movie code, movie_name, movie_cat, movie_year, movie_lang, movie_ratings, movie_duration, display_type.

2NF: 1st: hall_number, hall name, hall class.

2nd: movie code, movie_name, movie_cat, movie_year, movie_lang, movie_ratings, movie_duration.

3rd: serial_no,hall_number, movie_code, display_type.

3NF: 1st: hall number, hall_name .

2nd: hall name, hall_class.

3rd: movie code, movie_name, movi_ cat, movie_year, movie_lang, movie_ratings, movie_duration.

4th: Serial no, hall_number, movie_code, display_type.

FINAL TABLE:

1. Hall_info: hall number , hall_name, theater_number.
2. Hall_class: hall name, hall_class.
3. Theatre : theatre_number, theatre_name, floor_number, building_number, street_number.
4. Audience: aud_id , aud_name, aud_phone, aud_email.
5. Booking_details: serial no , Aud_id , ticket_number, payment_type, booking_time.
6. Ticket: ticket number , price, show_time, seat_number.
7. Movie: movie code , movie_name, movie_cat, movie year, movie_lang, movie_rattings, movie_duration.
8. Movie_for_aud : serial no ,Audience_id , movie_code.
9. Display: serial no ,Hall_number, movie_code, display_type.

Table Creation and Data Insertion

Hall_info:

Table Creation:

```
create table hall_info (hall_number number(4,0) constraint pkh primary key,
                        hall_name varchar2(32),
                        constraint fkkk foreign key(hall_name) references hall_class(hall_name),
                        theatre_number number(4,0),
                        constraint fktn foreign key(theatre_number) references theatre(theatre_number));
describe hall_info
```

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **HALL_INFO**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
HALL_INFO	HALL_NUMBER	Number	-	4	0	1	-	-	-
	HALL_NAME	Varchar2	32	-	-	-	✓	-	-
	THEATRE_NUMBER	Number	-	4	0	-	✓	-	-

1 - 3

Data Insertion:

```
insert into hall_info(hall_number,hall_name,theatre_number)values(111,'MOVIE_LOVERS',405);
insert into hall_info(hall_number,hall_name,theatre_number)values(222,'MOVIE_ZONE',606);
insert into hall_info(hall_number,hall_name,theatre_number)values(333,'CRAZY_CITY',201);
insert into hall_info(hall_number,hall_name,theatre_number)values(444,'GOLDEN_SCREEN',105);
insert into hall_info(hall_number,hall_name,theatre_number)values(555,'THE_BEST_VIEW',306);
select*from hall_info
```

Results Explain Describe Saved SQL History

HALL_NUMBER	HALL_NAME	THEATRE_NUMBER
111	MOVIE_LOVERS	405
222	MOVIE_ZONE	606
333	CRAZY_CITY	201
444	GOLDEN_SCREEN	105
555	THE_BEST_VIEW	306

5 rows returned in 0.00 seconds CSV Export

Hall Class:

Table Creation:

```
create table hall_class(hall_name varchar2(32) constraint pkhn primary key,
                        hall_class varchar2(16) default('REGULAR'));
describe hall_class
```

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **HALL_CLASS**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
HALL_CLASS	HALL_NAME	Varchar2	32	-	-	1	-	-	-
	HALL_CLASS	Varchar2	16	-	-	-	✓	('REGULAR')	-

1 - 2

Data Insertion:

```
insert into hall_class(hall_name,hall_class)values('MOVIE_LOVERS','1st_CLASS');
insert into hall_class(hall_name,hall_class)values('MOVIE_ZONE','2nd_CLASS');
insert into hall_class(hall_name,hall_class)values('CRAZY_CITY','2nd_CLASS');
insert into hall_class(hall_name,hall_class)values('GOLDEN_SCREEN','regu_CLASS');
insert into hall_class(hall_name,hall_class)values('THE_BEST_VIEW','1st_CLASS');
insert into hall_class(hall_name,hall_class)values('ATLANTIS','2nd_CLASS');
insert into hall_class(hall_name,hall_class)values('PRIME_LANDS','1st_CLASS');
insert into hall_class(hall_name,hall_class)values('THE_ARCTIC','regu_CLASS');
insert into hall_class(hall_name,hall_class)values('THE_SKYLINE','1st_CLASS');
select*from hall_class
```

Results Explain Describe Saved SQL History

HALL_NAME	HALL_CLASS
MOVIE_LOVERS	1st_CLASS
MOVIE_ZONE	2nd_CLASS
CRAZY_CITY	2nd_CLASS
GOLDEN_SCREEN	regu_CLASS
THE_BEST_VIEW	1st_CLASS
ATLANTIS	2nd_CLASS
PRIME_LANDS	1st_CLASS
THE_ARCTIC	regu_CLASS
THE_SKYLINE	1st_CLASS

9 rows returned in 0.00 seconds CSV Export

Theatre:

Table Creation:

```
create table theatre(theatre_number number(4,0) constraint pkt primary key,
                     theatre_name varchar2(16) not null,
                       floor_number number(2,0) not null,
                          building_number varchar2(16),
                             street_number varchar2(16));
describe theatre
```

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **THEATRE**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
THEATRE	THEATRE_NUMBER	Number	-	4	0	1	-	-	-
	THEATRE_NAME	Varchar2	16	-	-	-	-	-	-
	FLOOR_NUMBER	Number	-	2	0	-	-	-	-
	BUILDING_NUMBER	Varchar2	16	-	-	-	✓	-	-
	STREET_NUMBER	Varchar2	16	-	-	-	✓	-	-

1 - 5

Data Insertion:

```
insert into theatre (theatre_number,theatre_name,floor_number,building_number,street_number)
                     values(405,'ambica_palace',06,'house#2/B','road#12,nikunja');
insert into theatre (theatre_number,theatre_name,floor_number,building_number,street_number)
                     values(606,'deepam_talkies',02,'house#9/F','road#24,pragati');
insert into theatre (theatre_number,theatre_name,floor_number,building_number,street_number)
                     values(201,'sanget_cinema',05,'house#4/c','road#w36,mirpur');
insert into theatre (theatre_number,theatre_name,floor_number,building_number,street_number)
                     values(105,'7D_cinema',07,'house#5/A','road#s72,mawa');
insert into theatre (theatre_number,theatre_name,floor_number,building_number,street_number)
                     values(306,'sanjay_cineplex',06,'house#11/N','road#vw9,tejgaon');
insert into theatre (theatre_number,theatre_name,floor_number,building_number,street_number)
                    values(999,'popcorn_palace',04,'house#77/A','road#78,gulsan');
select*from theatre;
```

Results Explain Describe Saved SQL History

THEATRE_NUMBER	THEATRE_NAME	FLOOR_NUMBER	BUILDING_NUMBER	STREET_NUMBER
405	ambica_palace	6	house#2/B	road#12,nikunja
606	deepam_talkies	2	house#9/F	road#24,pragati
201	sanget_cinema	5	house#4/c	road#w36,mirpur
105	7D_cinema	7	house#5/A	road#s72,mawa
306	sanjay_cineplex	6	house#11/N	road#vw9,tejgaon
999	popcorn_palace	4	house#77/A	road#78,gulsan

6 rows returned in 0.01 seconds CSV Export

Audience:

Table Craetion:

```
create table audience(aud_id number(4,0) constraint pka primary key,
                      aud_name varchar2(32) not null,
                       aud_phone number(11) not null,
                          aud_email varchar2(32));
describe audience
```

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **AUDIENCE**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
AUDIENCE	AUD_ID	Number	-	4	0	1	-	-	-
	AUD_NAME	Varchar2	32	-	-	-	-	-	-
	AUD_PHONE	Number	-	11	0	-	-	-	-
	AUD_EMAIL	Varchar2	32	-	-	-	✓	-	-

1 - 4

Data Insertion:

```
insert into audience(aud_id,aud_name,aud_phone,aud_email)values(1212,'MESSI',54387364656,'MESSI@GMAIL.COM');
insert into audience(aud_id,aud_name,aud_phone,aud_email)values(1449,'RONALDO',351929200,'RONALDO@GMAIL.COM');
insert into audience(aud_id,aud_name,aud_phone,aud_email)values(9999,'NEYMAR',55787364,'NEYMAR@GMAIL.COM');
insert into audience(aud_id,aud_name,aud_phone,aud_email)values(2242,'DYBALA',54787364656,'DYBALA@GMAIL.COM');
insert into audience(aud_id,aud_name,aud_phone,aud_email)values(1234,'BUSQUETS',34787364656,'BUSQUESTS@GMAIL.COM');
select*from audience
```

Results Explain Describe Saved SQL History

AUD_ID	AUD_NAME	AUD_PHONE	AUD_EMAIL
1212	MESSI	54387364656	MESSI@GMAIL.COM
1449	RONALDO	351929200	RONALDO@GMAIL.COM
9999	NEYMAR	55787364	NEYMAR@GMAIL.COM
2242	DYBALA	54787364656	DYBALA@GMAIL.COM
1234	BUSQUETS	34787364656	BUSQUESTS@GMAIL.COM

5 rows returned in 0.00 seconds CSV Export

Booking_details:

Table Creation:

```
create table booking_details(serial_no number(4,0) constraint pkbds primary key,
                             aud_id number(4,0),
                             constraint fkbda foreign key(aud_id) references audience(aud_id),
                                ticket_number number(4,0),
                                constraint fkbdt foreign key(ticket_number) references ticket(ticket_number),
                                   payment_type varchar2(32),
                                     booking_time date not null);
alter table booking_details add constraint ckbpt check(payment_type in('CASH_PAYMENT','BKASH_PAYMENT','NAGAD_PAYMENT'));
describe booking_details
```

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **BOOKING_DETAILS**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
BOOKING_DETAILS	SERIAL_NO	Number	-	4	0	1	-	-	-
	AUD_ID	Number	-	4	0	-	✓	-	-
	TICKET_NUMBER	Number	-	4	0	-	✓	-	-
	PAYMENT_TYPE	Varchar2	32	-	-	-	✓	-	-
	BOOKING_TIME	Date	7	-	-	-	-	-	-

1 - 5

Data Insertion:

```
insert into booking_details(serial_no,aud_id,ticket_number,payment_type,booking_time)
                            values(1,1212,1212,'CASH_PAYMENT',to_date('01/12/2022/16:00:00','dd/mm/yyyy/hh24:mi:ss'));
insert into booking_details(serial_no,aud_id,ticket_number,booking_time)
                            values(2,1449,2323,to_date('05/12/2022/17:34:40','dd/mm/yyyy/hh24:mi:ss'));
insert into booking_details(serial_no,aud_id,ticket_number,booking_time)
                            values(3,9999,3434,to_date('09/12/2022/13:23:02','dd/mm/yyyy/hh24:mi:ss'));
insert into booking_details(serial_no,aud_id,ticket_number,payment_type,booking_time)
                            values(4,2242,4545,'BKASH_PAYMENT',to_date('10/12/2022/09:47:23','dd/mm/yyyy/hh24:mi:ss'));
insert into booking_details(serial_no,aud_id,ticket_number,payment_type,booking_time)
                            values(5,1234,5656,'NAGAD_PAYMENT',to_date('11/12/2022/14:31:40','dd/mm/yyyy/hh24:mi:ss'));
select*from booking_details
```

Results Explain Describe Saved SQL History

SERIAL_NO	AUD_ID	TICKET_NUMBER	PAYMENT_TYPE	BOOKING_TIME
1	1212	1212	CASH_PAYMENT	01-DEC-22
2	1449	2323	CASH_PAYMENT	05-DEC-22
3	9999	3434	CASH_PAYMENT	09-DEC-22
4	2242	4545	BKASH_PAYMENT	10-DEC-22
5	1234	5656	NAGAD_PAYMENT	11-DEC-22

5 rows returned in 0.00 seconds CSV Export

Ticket:

Table Craetion:

```
create table ticket(ticket_number number(4,0) constraint pktn primary key,
                    seat_number number(3,0) not null,
                      show_time date not null,
                       price number(4,0) not null);
describe ticket
```

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **TICKET**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
TICKET	TICKET_NUMBER	Number	-	4	0	1	-	-	-
	SEAT_NUMBER	Number	-	3	0	-	-	-	-
	SHOW_TIME	Date	7	-	-	-	-	-	-
	PRICE	Number	-	4	0	-	-	-	-

1 - 4

Data Insertion:

```
insert into ticket values(1212,12,to_date('10/12/2022/10:00:00','dd/mm/yyyy/hh:mi:ss'),1200);
insert into ticket values(2323,24,to_date('20/12/2022/16:00:00','dd/mm/yyyy/hh24:mi:ss'),1500);
insert into ticket values(3434,36,to_date('25/12/2022/19:30:00','dd/mm/yyyy/hh24:mi:ss'),1000);
insert into ticket values(4545,60,to_date('05/01/2023/11:30:00','dd/mm/yyyy/hh24:mi:ss'),1200);
insert into ticket values(5656,48,to_date('01/01/2023/20:00:00','dd/mm/yyyy/hh24:mi:ss'),1400);
select* from ticket
```

Results Explain Describe Saved SQL History

TICKET_NUMBER	SEAT_NUMBER	SHOW_TIME	PRICE
1212	12	10-DEC-22	1200
2323	24	20-DEC-22	1500
3434	36	25-DEC-22	1000
4545	60	05-JAN-23	1200
5656	48	01-JAN-23	1400

5 rows returned in 0.00 seconds CSV Export

Movie:

Table Craetion :

```
create table movie(movie_code number(4,0) constraint pkm primary key,
                   movie_name varchar2(16) not null,
                      movie_cat varchar2(16),
                        movie_year number(4,0) constraint chky check(movie_year between 2000 and 2023) not null,
                           movie_lang varchar2(16) default('ENGLISH'),
                             movie_duration varchar2(16),
                             movie_rattings float);
describe movie
```

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **MOVIE**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
MOVIE	MOVIE_CODE	Number	-	4	0	1	-	-	-
	MOVIE_NAME	Varchar2	16	-	-	-	-	-	-
	MOVIE_CAT	Varchar2	16	-	-	-	✓	-	-
	MOVIE_YEAR	Number	-	4	0	-	-	-	-
	MOVIE_LANG	Varchar2	16	-	-	-	✓	('ENGLISH')	-
	MOVIE_DURATION	Varchar2	16	-	-	-	✓	-	-
	MOVIE_RATTINGS	Float	22	126	-	-	✓	-	-

1 - 7

Data insertion:

```
insert into movie(movie_code,movie_name,movie_cat,movie_year,movie_lang,movie_duration,movie_rattings)
                  values(2212,'DEAR_ZINDAGI','FANTASY',2019,'HINDI','3h:46min',3.5);
insert into movie(movie_code,movie_name,movie_cat,movie_year,movie_lang,movie_duration,movie_rattings)
                  values(4545,'BAR BAR DEKHO','ROMANCE',2017,'HINDI','2h:46min',4.1);
insert into movie(movie_code,movie_name,movie_cat,movie_year,movie_lang,movie_duration,movie_rattings)
                  values(6626,'THE DARK KNIGHT','ACTION',2011,'ENGLISH','2h:36min',4.3);
insert into movie(movie_code,movie_name,movie_cat,movie_year,movie_duration,movie_rattings)
                  values(1126,'AVATAR','FANTASY',2005,'2h:58min',4.7);
insert into movie(movie_code,movie_name,movie_cat,movie_year,movie_lang,movie_duration,movie_rattings)
                  values(1456,'THE LION KING','ADVENTURE',2015,'ENGLISH','2h:26min',4.9);
select*from movie
```

Results Explain Describe Saved SQL History

MOVIE_CODE	MOVIE_NAME	MOVIE_CAT	MOVIE_YEAR	MOVIE_LANG	MOVIE_DURATION	MOVIE_RATTINGS
2212	DEAR_ZINDAGI	FANTASY	2019	HINDI	3h:46min	3.5
4545	BAR BAR DEKHO	ROMANCE	2017	HINDI	2h:46min	4.1
6626	THE DARK KNIGHT	ACTION	2011	ENGLISH	2h:36min	4.3
1126	AVATAR	FANTASY	2005	ENGLISH	2h:58min	4.7
1456	THE LION KING	ADVENTURE	2015	ENGLISH	2h:26min	4.9

5 rows returned in 0.00 seconds CSV Export

Movie_for_aud:

Table_creation:

```
create table movie_for_audience(serial_no number(4,0) constraint pkms primary key,
                                aud_id number(4,0), movie_code number(4,0),
                                   constraint fkmam foreign key(movie_code) references movie(movie_code));
describe movie_for_audience

alter table movie_for_audience add constraint fkmaa foreign key(aud_id) references audience(aud_id);
```

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **MOVIE_FOR_AUDIENCE**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
MOVIE_FOR_AUDIENCE	SERIAL_NO	Number	-	4	0	1	-	-	-
	AUD_ID	Number	-	4	0	-	✓	-	-
	MOVIE_CODE	Number	-	4	0	-	✓	-	-

1 - 3

Data Insertion:

```
insert into movie_for_audience values(1,1212,2212);
insert into movie_for_audience values(2,1449,4545);
insert into movie_for_audience values(3,9999,6626);
insert into movie_for_audience values(4,2242,1126);
insert into movie_for_audience values(5,1234,1456);
select*from movie_for_audience
```

Results Explain Describe Saved SQL History

SERIAL_NO	AUD_ID	MOVIE_CODE
1	1212	2212
2	1449	4545
3	9999	6626
4	2242	1126
5	1234	1456

5 rows returned in 0.00 seconds CSV Export

Display:

Table Creation:

```
create table display(serial_no number(4,0) constraint pkdsn primary key,
                     hall_number number(4,0),
                     constraint fkdhn foreign key(hall_number) references hall_info(hall_number),
                         movie_code number(4,0),
                         constraint fkdmn foreign key(movie_code) references movie(movie_code),
                            display_type varchar2(16) default('2D'));
describe display
```

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **DISPLAY**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
DISPLAY	SERIAL_NO	Number	-	4	0	1	-	-	-
	HALL_NUMBER	Number	-	4	0	-	✓	-	-
	MOVIE_CODE	Number	-	4	0	-	✓	-	-
	DISPLAY_TYPE	Varchar2	16	-	-	-	✓	('2D')	-

1 - 4

Data Insertion:

```
insert into display(serial_no,hall_number,movie_code,display_type) values(1,111,2212,'3D');
insert into display(serial_no,hall_number,movie_code) values(2,222,4545);
insert into display(serial_no,hall_number,movie_code) values(3,333,6626);
insert into display(serial_no,hall_number,movie_code,display_type) values(4,444,1126,'3D');
insert into display(serial_no,hall_number,movie_code,display_type) values(5,555,6767,'7D');
select*from display
```

Results Explain Describe Saved SQL History

SERIAL_NO	HALL_NUMBER	MOVIE_CODE	DISPLAY_TYPE
1	111	2212	3D
2	222	4545	2D
3	333	6626	2D
4	444	1126	3D
5	555	6767	7D

5 rows returned in 0.00 seconds CSV Export

Joining

Equijoin

1.Display all the audience who booked tickets on cash.

```
select a.*
from audience a,booking_details b
where a.aud_id=b.aud_id and payment_type='CASH_PAYMENT';
```

Results Explain Describe Saved SQL History

AUD_ID	AUD_NAME	AUD_PHONE	AUD_EMAIL
1212	MESSI	54387364656	MESSI@GMAIL.COM
1449	RONALDO	351929200	RONALDO@GMAIL.COM
9999	NEYMAR	55787364	NEYMAR@GMAIL.COM

3 rows returned in 0.02 seconds CSV Export

2.Display all the movie details and hall information which are shown on 2D display or shown at hall name “MOVIE_ LOVER”.

```
select m.*,h.*
from movie m,display d,hall_info h
where m.movie_code=d.movie_code and h.hall_number=d.hall_number
      and (d.display_type='2D' or h.hall_name='MOVIE_LOVERS') ;
```

Results Explain Describe Saved SQL History

MOVIE_CODE	MOVIE_NAME	MOVIE_CAT	MOVIE_YEAR	MOVIE_LANG	MOVIE_DURATION	MOVIE_RATTINGS	HALL_NUMBER	HALL_NAME	THEATRE_NUMBER
2212	DEAR_ZINDAGI	FANTASY	2019	HINDI	3h:46min	3.5	111	MOVIE_LOVERS	405
4545	BAR BAR DEKHO	ROMANCE	2017	HINDI	2h:46min	4.1	222	MOVIE_ZONE	606
6626	THE DARK KNIGHT	ACTION	2011	ENGLISH	2h:36min	4.3	333	CRAZY_CITY	201

3 rows returned in 0.00 seconds CSV Export

Outer join

1.Display all the hall names along with the theatre details if any theatre doesn't have any hall mark the theatre as 'Under construction'.

```
select t.*,NVL(h.hall_name,'Under construction') hall_name
from theatre t,hall_info h
where t.theatre_number=h.theatre_number(+)
```

Results Explain Describe Saved SQL History

THEATRE_NUMBER	THEATRE_NAME	FLOOR_NUMBER	BUILDING_NUMBER	STREET_NUMBER	HALL_NAME
105	7D_cinema	7	house#5/A	road#s72,mawa	GOLDEN_SCREEN
201	sanget_cinema	5	house#4/c	road#w36,mirpur	CRAZY_CITY
306	sanjay_cineplex	6	house#11/N	road#vw9,tejgaon	THE_BEST_VIEW
405	ambica_palace	6	house#2/B	road#12,nikunja	MOVIE_LOVERS
606	deepam_talkies	2	house#9/F	road#24,pragati	MOVIE_ZONE
999	popcorn_palace	4	house#77/A	road#78,gulsan	Under construction

6 rows returned in 0.02 seconds CSV Export

2.Display all the hall names and theatre information along with movie names if any movie does not shown yet mark it on the hall name as 'Not Shown'.

```
select m.movie_name,NVL(h.hall_name,'Not shown') hall_name,t.*
from movie m,display d,theatre t,hall_info h
where m.movie_code=d.movie_code(+)
      and h.hall_number(+)=d.hall_number
      and h.theatre_number=t.theatre_number(+);
```

Results Explain Describe Saved SQL History

MOVIE_NAME	HALL_NAME	THEATRE_NUMBER	THEATRE_NAME	FLOOR_NUMBER	BUILDING_NUMBER	STREET_NUMBER
DEAR_ZINDAGI	MOVIE_LOVERS	405	ambica_palace	6	house#2/B	road#12,nikunja
BAR BAR DEKHO	MOVIE_ZONE	606	deepam_talkies	2	house#9/F	road#24,pragati
THE DARK KNIGHT	CRAZY_CITY	201	sanget_cinema	5	house#4/c	road#w36,mirpur
AVATAR	GOLDEN_SCREEN	105	7D_cinema	7	house#5/A	road#s72,mawa
3EDIOTS	THE_BEST_VIEW	306	sanjay_cineplex	6	house#11/N	road#vw9,tejgaon
THE LION KING	Not shown	-	-	-	-	-

6 rows returned in 0.01 seconds CSV Export

Sub-Query

1. Display all the movie details whose movie years are grater than movie 3EDIOTS and ratting grater than movie 'DEAR_ ZINDAGI'.

```
select*
from movie
where movie_year>(select movie_year
                  from movie
                   where movie_name='3EDIOTS')
    and movie_rattings>(select movie_rattings
                  from movie
                    where movie_name='DEAR_ZINDAGI')
```

Results Explain Describe Saved SQL History

MOVIE_CODE	MOVIE_NAME	MOVIE_CAT	MOVIE_YEAR	MOVIE_LANG	MOVIE_DURATION	MOVIE_RATTINGS
4545	BAR BAR DEKHO	ROMANCE	2017	HINDI	2h:46min	4.1
6626	THE DARK KNIGHT	ACTION	2011	ENGLISH	2h:36min	4.3
1456	THE LION KING	ADVENTURE	2015	ENGLISH	2h:26min	4.9

3 rows returned in 0.00 seconds CSV Export

2. Display hall numbers and hall names whose theatre number is lower than theatre ‘deepam_ talkies’ or located in the lower floor than theatre '7D_ cinema'.

```
select hall_number,hall_name
from hall_info
where theatre_number<(select theatre_number
                      from theatre
                       where theatre_name='deepam_talkies')
    or theatre_number in(select theatre_number from theatre
                         where floor_number<(select floor_number from theatre
                                             where theatre_name='7D_cinema'))
```

Results Explain Describe Saved SQL History

HALL_NUMBER	HALL_NAME
111	MOVIE_LOVERS
222	MOVIE_ZONE
333	CRAZY_CITY
444	GOLDEN_SCREEN
555	THE_BEST_VIEW

5 rows returned in 0.00 seconds CSV Export

3. Display ticket number along with the price where the tickets cost more than the average price.

```
select ticket_number,price
   from ticket
    where price>(select avg(price)
                    from ticket);
```

Results Explain Describe Saved SQL History

TICKET_NUMBER	PRICE
2323	1500
5656	1400

2 rows returned in 0.05 seconds CSV Export

4. Display all the audience details who watched the movie 'AVATAR' or 'BAR BAR DEKHO'.

```
select*
from audience
where aud_id in(select aud_id
                from movie_for_audience
                    where movie_code in(select movie_code
                                        from movie where movie_name='AVATAR' or movie_name='BAR BAR DEKHO'))
```

Results Explain Describe Saved SQL History

AUD_ID	AUD_NAME	AUD_PHONE	AUD_EMAIL
1449	RONALDO	351929200	RONALDO@GMAIL.COM
2242	DYBALA	54787364656	DYBALA@GMAIL.COM

2 rows returned in 0.00 seconds CSV Export

View

1. **Simple View:** Create a view named movie_details from table movie containing movie names, movie year, and movie category where movie code is grater than 1456.

```
create view movie_details as
  select movie_name,movie_year,movie_cat
    from movie
      where movie_code>1456;
select* from movie_details
```

Results | Explain | Describe | Saved SQL | History

MOVIE_NAME	MOVIE_YEAR	MOVIE_CAT
DEAR_ZINDAGI	2019	FANTASY
BAR BAR DEKHO	2017	ROMANCE
THE DARK KNIGHT	2011	ACTION
3EDIOTS	2009	COMEDY

4 rows returned in 0.02 seconds CSV Export

2. **Complex View:** Create a view named v_theatre that contains the hall numbers, hall names, and theatre numbers for all halls where theatre number is grater than 201. Do not allow any hall to be changed to the another theatre though the view.

```
create view v_theatre as
   select hall_number,hall_name,theatre_number
     from hall_info
       where theatre_number>201
         with check option constraint v_theatre;
select* from v_theatre
```

Results | Explain | Describe | Saved SQL | History

HALL_NUMBER	HALL_NAME	THEATRE_NUMBER
111	MOVIE_LOVERS	405
222	MOVIE_ZONE	606
555	THE_BEST_VIEW	306

3 rows returned in 0.00 seconds CSV Export

Adding constraint:

Adding check constraint on table theatre :