Al Imam Mohammad Ibn Saud Islamic University
College of Computer and Information Sciences

Information Systems Department

Course Project

Course Title:	Introduction to Database
Course Code:	IS 320
Semester/Year:	Spring 2020
Course Instructor:	T. Sara Aljomman

Project Title:	Cinema Booking System

Agreement

1- This project was me and my team members NOT external party who performed this project.
2- I participated with the group members to accomplish this project effectively and almost equally.
3- This project is totally free from copy and any type of cheating from other students' works and projects.
4- This project is free from illegal copy from any resources and intellectual property breaches.
Based on above I sign below, and I accept any corrective action taken in case I breach or don't fulfil the above commitments.

Student Name	Student ID	Section No.	Signature	Mark (20)
Lateefah Alhaidari	**439020524**	**372**		
Nora Alsuliman	**439019013**	**372**		
Bayan Alhumaidan	**434026414**	**372**		
Bayan Alamir	**435031007**	**372**		

Table of Contents

1.conceptual part

1.1 Introduction

This database is developed for VOX Cinema, to keep track of the bookings by customers either online or at the location. And what movie they saw and the theatre hall it was shown in. It also keeps track of employees who are making the booking at the location. Movies at our cinemas are only shown once.

1.2 Cinema booking system scenario

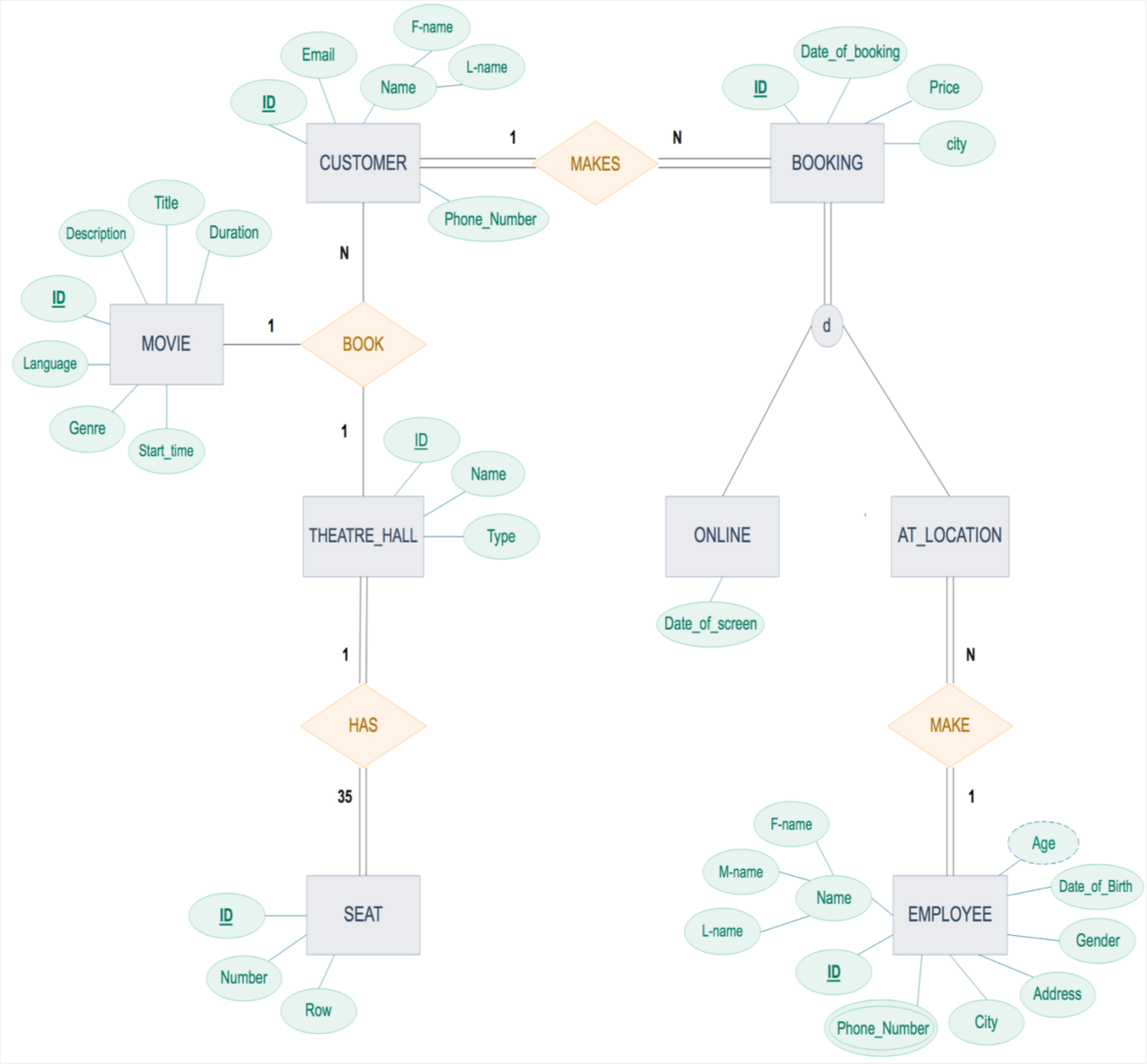
Bookings have a unique ID, price, city, and date of booking. There are only two ways of booking, either **online** or **at the location**. We need the customer's information when they make a booking, customers can make many bookings but at least one, each booking must be done by one customer, each **customer** has a unique ID, name (first and last), email, and phone number. Many customers can book 1 movie in 1 theatre hall.Each **theatre hall** has a unique ID, name, and type, **Movies** have a unique ID, title, language of the movie,the duration of the movie, the genre, movie description, and the start time of the movie. A theatre hall must have 35 seats but a seat can only be in one theatre hall, each **seat** has a unique ID,and the seats row and number.

For bookings made at the location we need the information of the employee making the booking there. Each **employee** has a unique ID, name (first, middle, and last), gender, DOB, phone number (employee may have more than one phone number), address, and city they are working at. We have at least one employee making bookings at the location,Employee can make several number of bookings. For the online bookings we need to know the date the booking is made for.

1.3 Entities

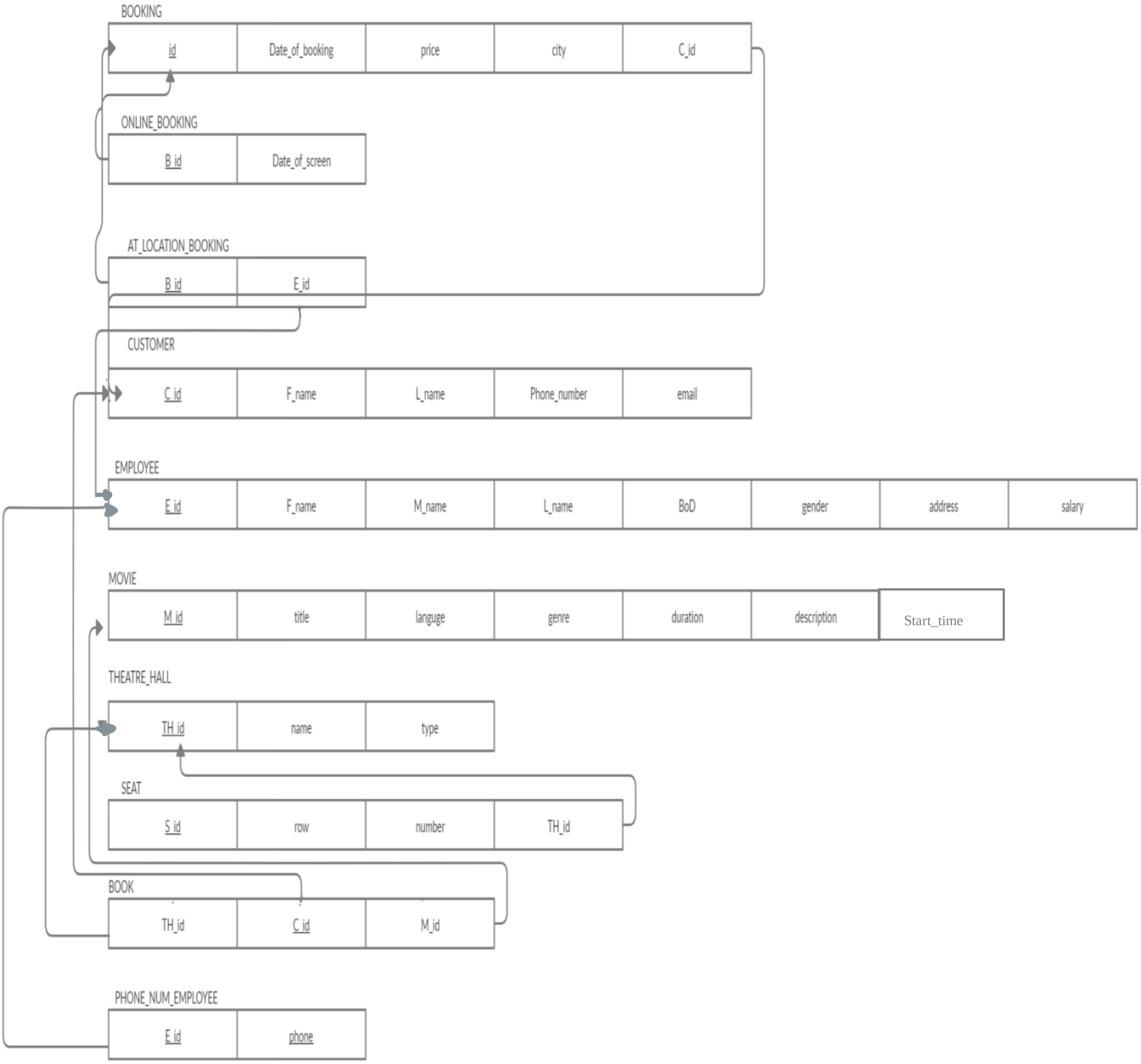
BOOKING
ONILINE_BOOKING
AT_LOCATION_BOOKING
CUSTOMER
THEATRE_HALL
MOVIE
SEATS
EMPLOYEE

1.4 EER for cinema booking system



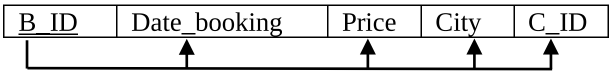
2.Logical part:

2.1 Relational database schema diagram:



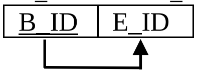
2.2 Normalization

BOOKING



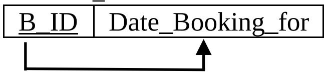
1NF: Yes, because there’s no multivalued attributes or composite attributes or nested relations.
2NF: Yes, because the attributes are fully dependent.
3NF: Yes, because there’s no transitive dependency.

AT_LOCATION_BOOKING



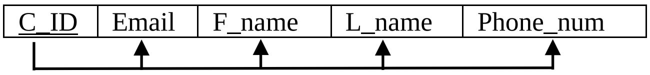
1NF: Yes, because there’s no multivalued attributes or composite attributes or nested relations.
2NF: Yes, because the attributes are fully dependent.
3NF: Yes, because there’s no transitive dependency.

ONLINE_BOOKING



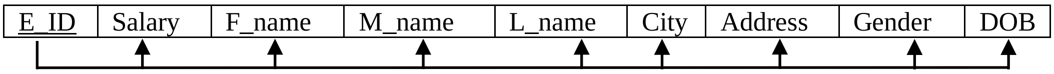
1NF: Yes, because there’s no multivalued attributes or composite attributes or nested relations.
2NF: Yes, because the attributes are fully dependent.
3NF: Yes, because there’s no transitive dependency.

CUSTOMER



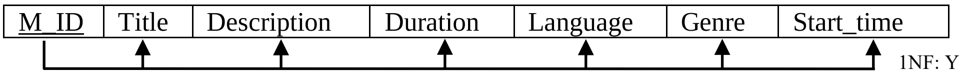
1NF: Yes, because there’s no multivalued attributes or composite attributes or nested relations.
2NF: Yes, because the attributes are fully dependent.
3NF: Yes, because there’s no transitive dependency.

EMPLOYEE



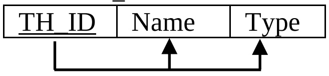
1NF: Yes, because there’s no multivalued attributes or composite attributes or nested relations.
2NF: Yes, because the attributes are fully dependent.
3NF: Yes, because there’s no transitive dependency.

MOVIE



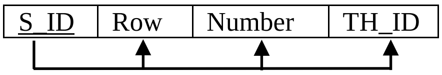
1NF: Yes, because there’s no multivalued attributes or composite attributes or nested relations.
2NF: Yes, because the attributes are fully dependent.
3NF: Yes, because there’s no transitive dependency.

THEATRE_HALL



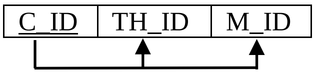
1NF: Yes, because there’s no multivalued attributes or composite attributes or nested relations.
2NF: Yes, because the attributes are fully dependent.
3NF: Yes, because there’s no transitive dependency.

SEAT



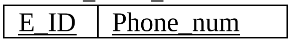
1NF: Yes, because there’s no multivalued attributes or composite attributes or nested relations.
2NF: Yes, because the attributes are fully dependent.
3NF: Yes, because there’s no transitive dependency.

BOOK



1NF: Yes, because there’s no multivalued attributes or composite attributes or nested relations.
2NF: Yes, because the attributes are fully dependent.
3NF: Yes, because there’s no transitive dependency.

PHONE_NUM_EMPLOYEE

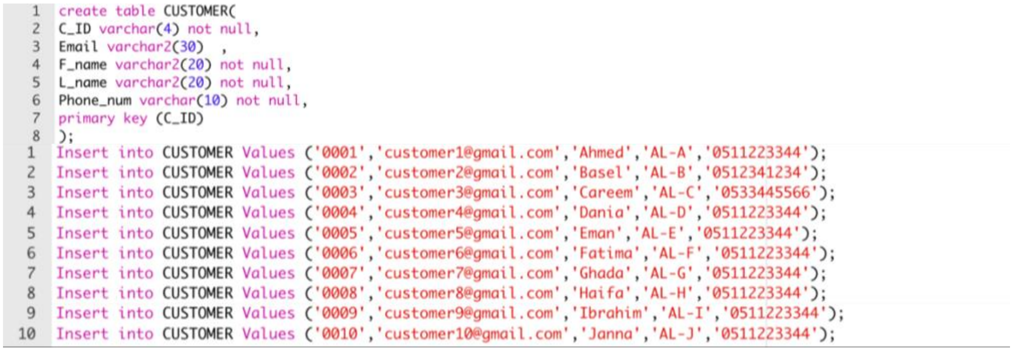


1NF: Yes, because there’s no multivalued attributes or composite attributes or nested relations.
2NF: Yes, because the attributes are fully dependent.
3NF: Yes, because there’s no transitive dependency.

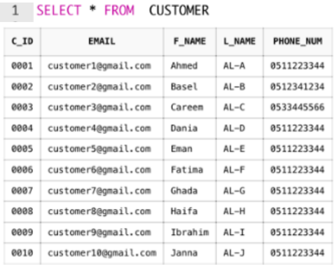
3. Physical part(implementation):

3.1 Schema Implementation:

1. Customer



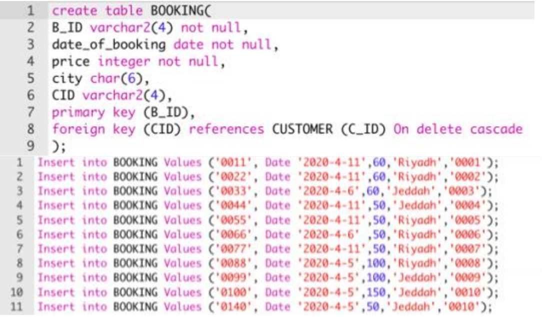
```
create table CUSTOMER(
C_ID varchar(4) not null,
Email varchar2(30)  ,
F_name varchar2(20) not null,
L_name varchar2(20) not null,
Phone_num varchar(10) not null,
primary key (C_ID)
);
Insert into CUSTOMER Values ('0001','customer1@gmail.com','Ahmed','AL-A','0511223344');
Insert into CUSTOMER Values ('0002','customer2@gmail.com','Basel','AL-B','0512341234');
Insert into CUSTOMER Values ('0003','customer3@gmail.com','Careem','AL-C','0533445566');
Insert into CUSTOMER Values ('0004','customer4@gmail.com','Dania','AL-D','0511223344');
Insert into CUSTOMER Values ('0005','customer5@gmail.com','Eman','AL-E','0511223344');
Insert into CUSTOMER Values ('0006','customer6@gmail.com','Fatima','AL-F','0511223344');
Insert into CUSTOMER Values ('0007','customer7@gmail.com','Ghada','AL-G','0511223344');
Insert into CUSTOMER Values ('0008','customer8@gmail.com','Haifa','AL-H','0511223344');
Insert into CUSTOMER Values ('0009','customer9@gmail.com','Ibrahim','AL-I','0511223344');
Insert into CUSTOMER Values ('0010','customer10@gmail.com','Janna','AL-J','0511223344');
```



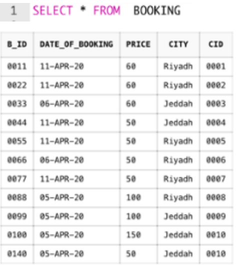
```
SELECT * FROM  CUSTOMER
```

C_ID	EMAIL	F_NAME	L_NAME	PHONE_NUM
0001	customer1@gmail.com	Ahmed	AL-A	0511223344
0002	customer2@gmail.com	Basel	AL-B	0512341234
0003	customer3@gmail.com	Careem	AL-C	0533445566
0004	customer4@gmail.com	Dania	AL-D	0511223344
0005	customer5@gmail.com	Eman	AL-E	0511223344
0006	customer6@gmail.com	Fatima	AL-F	0511223344
0007	customer7@gmail.com	Ghada	AL-G	0511223344
0008	customer8@gmail.com	Haifa	AL-H	0511223344
0009	customer9@gmail.com	Ibrahim	AL-I	0511223344
0010	customer10@gmail.com	Janna	AL-J	0511223344

2. Booking



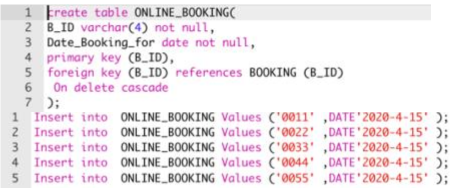
```
create table BOOKING(
B_ID varchar2(4) not null,
date_of_booking date not null,
price integer not null,
city char(6),
CID varchar2(4),
primary key (B_ID),
foreign key (CID) references CUSTOMER (C_ID) On delete cascade
);
Insert into BOOKING Values ('0011', Date '2020-4-11',60,'Riyadh','0001');
Insert into BOOKING Values ('0022', Date '2020-4-11',60,'Riyadh','0002');
Insert into BOOKING Values ('0033', Date '2020-4-6',60,'Jeddah','0003');
Insert into BOOKING Values ('0044', Date '2020-4-11',50,'Jeddah','0004');
Insert into BOOKING Values ('0055', Date '2020-4-11',50,'Riyadh','0005');
Insert into BOOKING Values ('0066', Date '2020-4-6' ,50,'Riyadh','0006');
Insert into BOOKING Values ('0077', Date '2020-4-11',50,'Riyadh','0007');
Insert into BOOKING Values ('0088', Date '2020-4-5',100,'Riyadh','0008');
Insert into BOOKING Values ('0099', Date '2020-4-5',100,'Jeddah','0009');
Insert into BOOKING Values ('0100', Date '2020-4-5',150,'Jeddah','0010');
Insert into BOOKING Values ('0140', Date '2020-4-5',50,'Jeddah','0010');
```



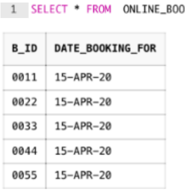
```
SELECT * FROM  BOOKING
```

B_ID	DATE_OF_BOOKING	PRICE	CITY	CID
0011	11-APR-20	60	Riyadh	0001
0022	11-APR-20	60	Riyadh	0002
0033	06-APR-20	60	Jeddah	0003
0044	11-APR-20	50	Jeddah	0004
0055	11-APR-20	50	Riyadh	0005
0066	06-APR-20	50	Riyadh	0006
0077	11-APR-20	50	Riyadh	0007
0088	05-APR-20	100	Riyadh	0008
0099	05-APR-20	100	Jeddah	0009
0100	05-APR-20	150	Jeddah	0010
0140	05-APR-20	50	Jeddah	0010

3. Online Booking



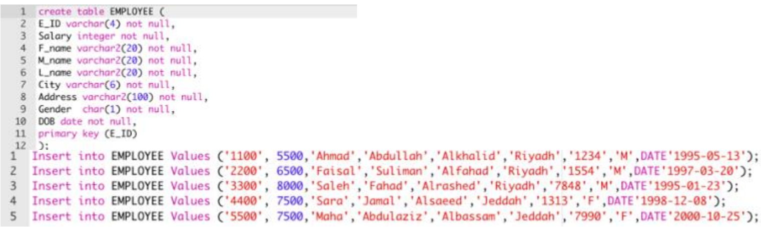
```
create table ONLINE_BOOKING(
B_ID varchar(4) not null,
Date_Booking_for date not null,
primary key (B_ID),
foreign key (B_ID) references BOOKING (B_ID)
 On delete cascade
);
Insert into  ONLINE_BOOKING Values ('0011'  ,DATE'2020-4-15' );
Insert into  ONLINE_BOOKING Values ('0022'  ,DATE'2020-4-15' );
Insert into  ONLINE_BOOKING Values ('0033'  ,DATE'2020-4-15' );
Insert into  ONLINE_BOOKING Values ('0044'  ,DATE'2020-4-15' );
Insert into  ONLINE_BOOKING Values ('0055'  ,DATE'2020-4-15' );
```



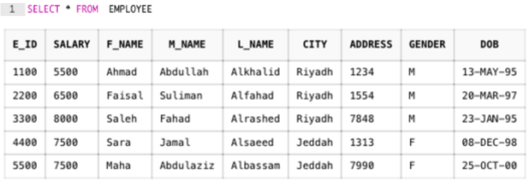
```
SELECT * FROM  ONLINE_BOOKING
```

B_ID	DATE_BOOKING_FOR
0011	15-APR-20
0022	15-APR-20
0033	15-APR-20
0044	15-APR-20
0055	15-APR-20

4. Employee



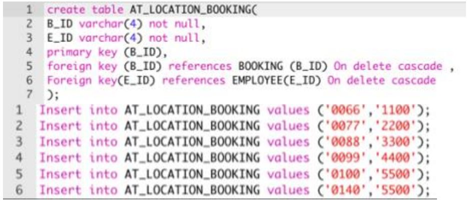
```
create table EMPLOYEE (
E_ID varchar(4) not null,
Salary integer not null,
F_name varchar2(20) not null,
M_name varchar2(20) not null,
L_name varchar2(20) not null,
City varchar(6) not null,
Address varchar2(100) not null,
Gender  char(1) not null,
DOB date not null,
primary key (E_ID)
);
Insert into EMPLOYEE Values ('1100', 5500,'Ahmad','Abdullah','Alkhalid','Riyadh','1234','M',DATE'1995-05-13');
Insert into EMPLOYEE Values ('2200', 6500,'Faisal','Suliman','Alfahad','Riyadh','1554','M',DATE'1997-03-20');
Insert into EMPLOYEE Values ('3300', 8000,'Saleh','Fahad','Alrashed','Riyadh','7848','M',DATE'1995-01-23');
Insert into EMPLOYEE Values ('4400', 7500,'Sara','Jamal','Alsaeed','Jeddah','1313','F',DATE'1998-12-08');
Insert into EMPLOYEE Values ('5500', 7500,'Maha','Abdulaziz','Albassam','Jeddah','7990','F',DATE'2000-10-25');
```



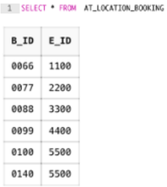
```
SELECT * FROM  EMPLOYEE
```

E_ID	SALARY	F_NAME	M_NAME	L_NAME	CITY	ADDRESS	GENDER	DOB
1100	5500	Ahmad	Abdullah	Alkhalid	Riyadh	1234	M	13-MAY-95
2200	6500	Faisal	Suliman	Alfahad	Riyadh	1554	M	20-MAR-97
3300	8000	Saleh	Fahad	Alrashed	Riyadh	7848	M	23-JAN-95
4400	7500	Sara	Jamal	Alsaeed	Jeddah	1313	F	08-DEC-98
5500	7500	Maha	Abdulaziz	Albassam	Jeddah	7990	F	25-OCT-00

5. At Location Booking



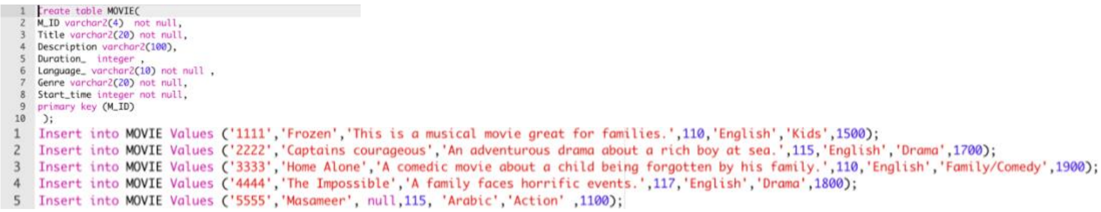
```
create table AT_LOCATION_BOOKING(
B_ID varchar(4) not null,
E_ID varchar(4) not null,
primary key (B_ID),
foreign key (B_ID) references BOOKING (B_ID) On delete cascade ,
Foreign key(E_ID) references EMPLOYEE(E_ID) On delete cascade
);
Insert into AT_LOCATION_BOOKING values ('0066','1100');
Insert into AT_LOCATION_BOOKING values ('0077','2200');
Insert into AT_LOCATION_BOOKING values ('0088','3300');
Insert into AT_LOCATION_BOOKING values ('0099','4400');
Insert into AT_LOCATION_BOOKING values ('0100','5500');
Insert into AT_LOCATION_BOOKING values ('0140','5500');
```



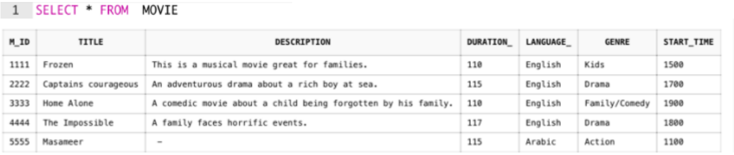
```
SELECT * FROM  AT_LOCATION_BOOKING
```

B_ID	E_ID
0066	1100
0077	2200
0088	3300
0099	4400
0100	5500
0140	5500

6. Movie



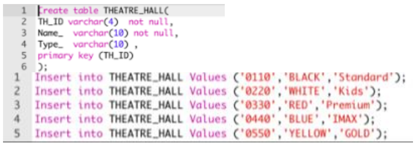
```
Create table MOVIE(
M_ID varchar2(4)  not null,
Title varchar2(20) not null,
Description varchar2(100),
Duration_  integer ,
Language_ varchar2(10) not null ,
Genre varchar2(20) not null,
Start_time integer not null,
primary key (M_ID)
 );
Insert into MOVIE Values ('1111','Frozen','This is a musical movie great for families.',110,'English','Kids',1500);
Insert into MOVIE Values ('2222','Captains courageous','An adventurous drama about a rich boy at sea.',115,'English','Drama',1700);
Insert into MOVIE Values ('3333','Home Alone','A comedic movie about a child being forgotten by his family.',110,'English','Family/Comedy',1900);
Insert into MOVIE Values ('4444','The Impossible','A family faces horrific events.',117,'English','Drama',1800);
Insert into MOVIE Values ('5555','Masameer', null,115, 'Arabic','Action' ,1100);
```



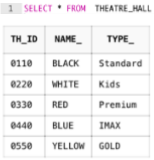
```
SELECT * FROM  MOVIE
```

M_ID	TITLE	DESCRIPTION	DURATION_	LANGUAGE_	GENRE	START_TIME
1111	Frozen	This is a musical movie great for families.	110	English	Kids	1500
2222	Captains courageous	An adventurous drama about a rich boy at sea.	115	English	Drama	1700
3333	Home Alone	A comedic movie about a child being forgotten by his family.	110	English	Family/Comedy	1900
4444	The Impossible	A family faces horrific events.	117	English	Drama	1800
5555	Masameer	-	115	Arabic	Action	1100

7. Theatre Hall



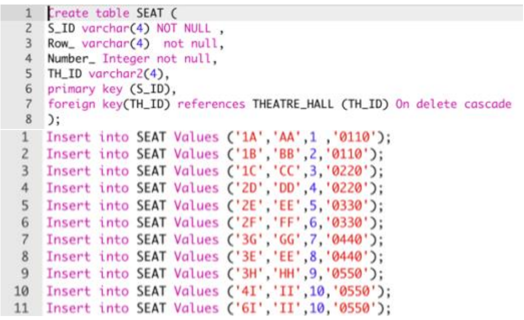
```
Create table THEATRE_HALL(
TH_ID varchar(4)  not null,
Name_  varchar(10) not null,
Type_  varchar(10) ,
primary key (TH_ID)
);
Insert into THEATRE_HALL Values ('0110','BLACK','Standard');
Insert into THEATRE_HALL Values ('0220','WHITE','Kids');
Insert into THEATRE_HALL Values ('0330','RED','Premium');
Insert into THEATRE_HALL Values ('0440','BLUE','IMAX');
Insert into THEATRE_HALL Values ('0550','YELLOW','GOLD');
```



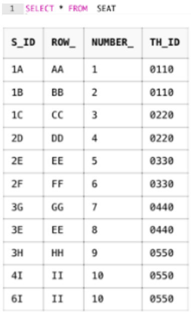
```
SELECT * FROM  THEATRE_HALL
```

TH_ID	NAME_	TYPE_
0110	BLACK	Standard
0220	WHITE	Kids
0330	RED	Premium
0440	BLUE	IMAX
0550	YELLOW	GOLD

8. Seat



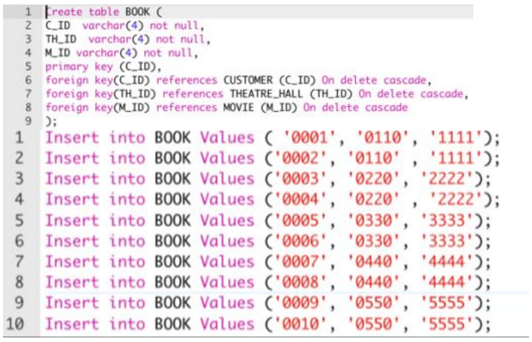
```
Create table SEAT (
S_ID varchar(4) NOT NULL ,
Row_ varchar(4)  not null,
Number_ Integer not null,
TH_ID varchar2(4),
primary key (S_ID),
foreign key(TH_ID) references THEATRE_HALL (TH_ID) On delete cascade
);
Insert into SEAT Values ('1A','AA',1 ,'0110');
Insert into SEAT Values ('1B','BB',2,'0110');
Insert into SEAT Values ('1C','CC',3,'0220');
Insert into SEAT Values ('2D','DD',4,'0220');
Insert into SEAT Values ('2E','EE',5,'0330');
Insert into SEAT Values ('2F','FF',6,'0330');
Insert into SEAT Values ('3G','GG',7,'0440');
Insert into SEAT Values ('3E','EE',8,'0440');
Insert into SEAT Values ('3H','HH',9,'0550');
Insert into SEAT Values ('4I','II',10,'0550');
Insert into SEAT Values ('6I','II',10,'0550');
```



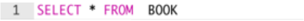
```
SELECT * FROM  SEAT
```

S_ID	ROW_	NUMBER_	TH_ID
1A	AA	1	0110
1B	BB	2	0110
1C	CC	3	0220
2D	DD	4	0220
2E	EE	5	0330
2F	FF	6	0330
3G	GG	7	0440
3E	EE	8	0440
3H	HH	9	0550
4I	II	10	0550
6I	II	10	0550

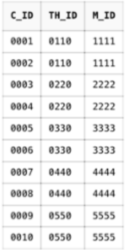
9. Book



```
Create table BOOK (
C_ID  varchar(4) not null,
TH_ID  varchar(4) not null,
M_ID varchar(4) not null,
primary key (C_ID),
foreign key(C_ID) references CUSTOMER (C_ID) On delete cascade,
foreign key(TH_ID) references THEATRE_HALL (TH_ID) On delete cascade,
foreign key(M_ID) references MOVIE (M_ID) On delete cascade
);
Insert into BOOK Values ( '0001', '0110', '1111');
Insert into BOOK Values ('0002', '0110' , '1111');
Insert into BOOK Values ('0003', '0220', '2222');
Insert into BOOK Values ('0004', '0220' , '2222');
Insert into BOOK Values ('0005', '0330', '3333');
Insert into BOOK Values ('0006', '0330', '3333');
Insert into BOOK Values ('0007', '0440', '4444');
Insert into BOOK Values ('0008', '0440', '4444');
Insert into BOOK Values ('0009', '0550', '5555');
Insert into BOOK Values ('0010', '0550', '5555');
```

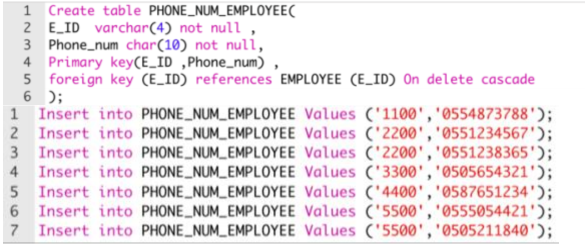


```
SELECT * FROM  BOOK
```

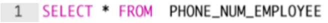


C_ID	TH_ID	M_ID
0001	0110	1111
0002	0110	1111
0003	0220	2222
0004	0220	2222
0005	0330	3333
0006	0330	3333
0007	0440	4444
0008	0440	4444
0009	0550	5555
0010	0550	5555

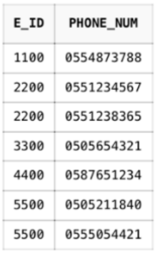
10. Phone Number Employee



```
Create table PHONE_NUM_EMPLOYEE(
E_ID  varchar(4) not null ,
Phone_num char(10) not null,
Primary key(E_ID ,Phone_num) ,
foreign key (E_ID) references EMPLOYEE (E_ID) On delete cascade
);
Insert into PHONE_NUM_EMPLOYEE Values ('1100','0554873788');
Insert into PHONE_NUM_EMPLOYEE Values ('2200','0551234567');
Insert into PHONE_NUM_EMPLOYEE Values ('2200','0551238365');
Insert into PHONE_NUM_EMPLOYEE Values ('3300','0505654321');
Insert into PHONE_NUM_EMPLOYEE Values ('4400','0587651234');
Insert into PHONE_NUM_EMPLOYEE Values ('5500','0555054421');
Insert into PHONE_NUM_EMPLOYEE Values ('5500','0505211840');
```



```
SELECT * FROM  PHONE_NUM_EMPLOYEE
```



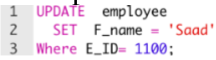
E_ID	PHONE_NUM
1100	0554873788
2200	0551234567
2200	0551238365
3300	0505654321
4400	0587651234
5500	0505211840
5500	0555054421

3.2 Query implementations:

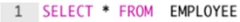
3.2.1 Data update/deletion:

1. List 2 different update queries related to your tables.

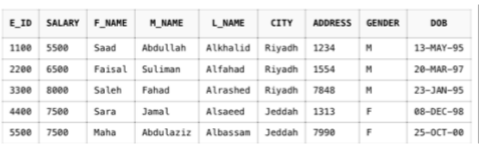
1.1 Update the first name of employee who’s E_ID=’1100’ to ‘Saad’.



```
UPDATE  employee
  SET  F_name = 'Saad'
Where E_ID= 1100;
```

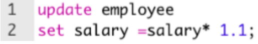


```
SELECT * FROM  EMPLOYEE
```

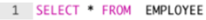


E_ID	SALARY	F_NAME	M_NAME	L_NAME	CITY	ADDRESS	GENDER	DOB
1100	5500	Saad	Abdullah	Alkhalid	Riyadh	1234	M	13-MAY-95
2200	6500	Faisal	Suliman	Alfahad	Riyadh	1554	M	20-MAR-97
3300	8000	Saleh	Fahad	Alrashed	Riyadh	7848	M	23-JAN-95
4400	7500	Sara	Jamal	Alsaeed	Jeddah	1313	F	08-DEC-98
5500	7500	Maha	Abdulaziz	Albassam	Jeddah	7990	F	25-OCT-00

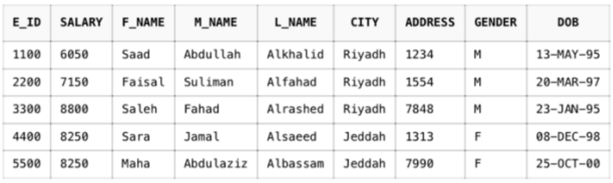
1.2 Update the salary of all employees with a 10% raise.



```
update employee
set salary =salary* 1.1;
```



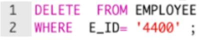
```
SELECT * FROM  EMPLOYEE
```



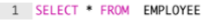
E_ID	SALARY	F_NAME	M_NAME	L_NAME	CITY	ADDRESS	GENDER	DOB
1100	6050	Saad	Abdullah	Alkhalid	Riyadh	1234	M	13-MAY-95
2200	7150	Faisal	Suliman	Alfahad	Riyadh	1554	M	20-MAR-97
3300	8800	Saleh	Fahad	Alrashed	Riyadh	7848	M	23-JAN-95
4400	8250	Sara	Jamal	Alsaeed	Jeddah	1313	F	08-DEC-98
5500	8250	Maha	Abdulaziz	Albassam	Jeddah	7990	F	25-OCT-00

2. List 2 different delete queries related to your tables.

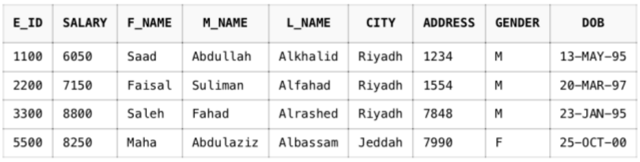
2.1 Delete employee with ID=’4400’.



```
DELETE  FROM EMPLOYEE
WHERE  E_ID= '4400' ;
```

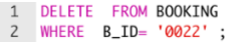


```
SELECT * FROM  EMPLOYEE
```

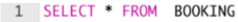


E_ID	SALARY	F_NAME	M_NAME	L_NAME	CITY	ADDRESS	GENDER	DOB
1100	6050	Saad	Abdullah	Alkhalid	Riyadh	1234	M	13-MAY-95
2200	7150	Faisal	Suliman	Alfahad	Riyadh	1554	M	20-MAR-97
3300	8800	Saleh	Fahad	Alrashed	Riyadh	7848	M	23-JAN-95
5500	8250	Maha	Abdulaziz	Albassam	Jeddah	7990	F	25-OCT-00

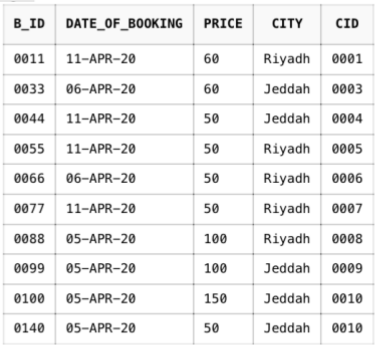
2.2 Delete the booking with Booking ID=’0022’.



```
DELETE  FROM BOOKING
WHERE  B_ID= '0022' ;
```



```
SELECT * FROM  BOOKING
```

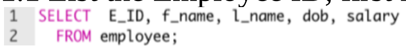


B_ID	DATE_OF_BOOKING	PRICE	CITY	CID
0011	11-APR-20	60	Riyadh	0001
0033	06-APR-20	60	Jeddah	0003
0044	11-APR-20	50	Jeddah	0004
0055	11-APR-20	50	Riyadh	0005
0066	06-APR-20	50	Riyadh	0006
0077	11-APR-20	50	Riyadh	0007
0088	05-APR-20	100	Riyadh	0008
0099	05-APR-20	100	Jeddah	0009
0100	05-APR-20	150	Jeddah	0010
0140	05-APR-20	50	Jeddah	0010

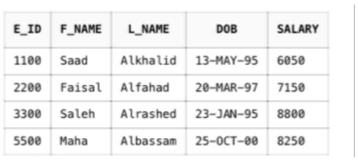
3.2.2 Data Retrieval (Select) Queries:

1. List 2 simple select queries related to your tables.

1.1 List the Employee ID, first name, last name, date of birth and salary of employees.

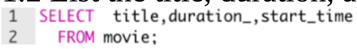


```
SELECT  E_ID, f_name, l_name, dob, salary
  FROM employee;
```

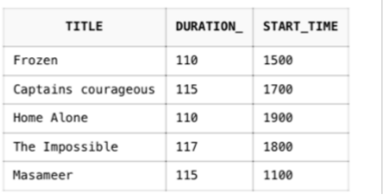


E_ID	F_NAME	L_NAME	DOB	SALARY
1100	Saad	Alkhalid	13-MAY-95	6050
2200	Faisal	Alfahad	20-MAR-97	7150
3300	Saleh	Alrashed	23-JAN-95	8800
5500	Maha	Albassam	25-OCT-00	8250

1.2 List the title, duration, and start time of every movie.



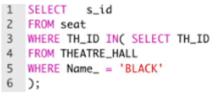
```
SELECT  title,duration_,start_time
  FROM movie;
```



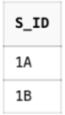
TITLE	DURATION_	START_TIME
Frozen	110	1500
Captains courageous	115	1700
Home Alone	110	1900
The Impossible	117	1800
Masameer	115	1100

2. List 2 nested queries related to your tables.

2.1 List the Seat ID that are in the Theatre halls that are named ‘Black’.

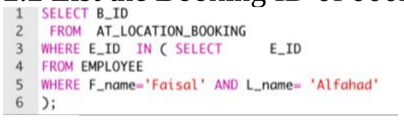


```
SELECT   s_id
FROM seat
WHERE TH_ID IN( SELECT TH_ID
FROM THEATRE_HALL
WHERE Name_ = 'BLACK'
);
```

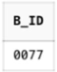


S_ID
1A
1B

2.2 List the Booking ID of bookings made by employee named ‘Faisal Alfahad’.



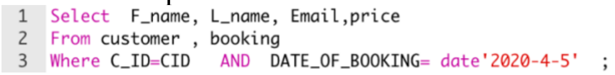
```
SELECT B_ID
 FROM  AT_LOCATION_BOOKING
WHERE E_ID  IN ( SELECT      E_ID
FROM EMPLOYEE
WHERE F_name='Faisal' AND L_name= 'Alfahad'
);
```



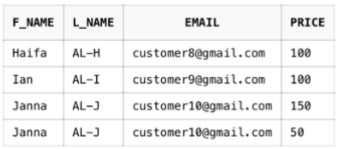
B_ID
0077

3. List 2 simple join queries related to your tables.

3.1 Retrieve the first name, last name, email, price of customers that made a booking on 5th April 2020.

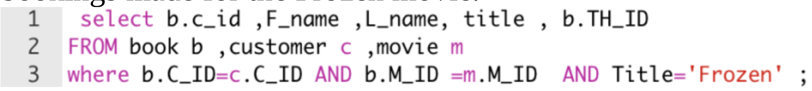


```
Select  F_name, L_name, Email,price
From customer , booking
Where C_ID=CID   AND  DATE_OF_BOOKING= date'2020-4-5'  ;
```

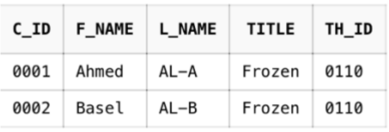


F_NAME	L_NAME	EMAIL	PRICE
Haifa	AL-H	customer8@gmail.com	100
Ian	AL-I	customer9@gmail.com	100
Janna	AL-J	customer10@gmail.com	150
Janna	AL-J	customer10@gmail.com	50

3.2 Retrieve customer ID, first name, last name, title of movie and theatre hall ID for bookings made for the Frozen movie.



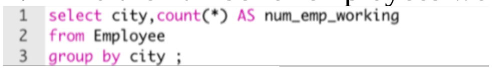
```
 select b.c_id ,F_name ,L_name, title , b.TH_ID
FROM book b ,customer c ,movie m
where b.C_ID=c.C_ID AND b.M_ID =m.M_ID  AND Title='Frozen' ;
```



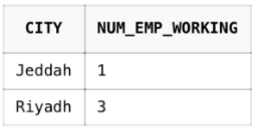
C_ID	F_NAME	L_NAME	TITLE	TH_ID
0001	Ahmed	AL-A	Frozen	0110
0002	Basel	AL-B	Frozen	0110

4. List 2 simple retrieval queries using group by, having clause, and aggregation functions related to your tables.

4.1 Find the number of employees working for each city.

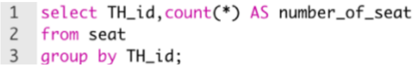


```
select city,count(*) AS num_emp_working
from Employee
group by city ;
```

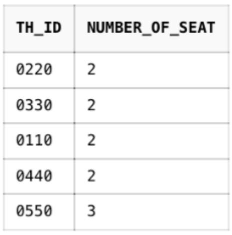


CITY	NUM_EMP_WORKING
Jeddah	1
Riyadh	3

4.2 Find the number of seats in each theatre hall.

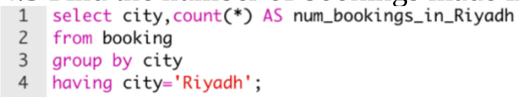


```
select TH_id,count(*) AS number_of_seat
from seat
group by TH_id;
```

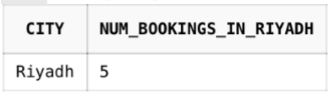


TH_ID	NUMBER_OF_SEAT
0220	2
0330	2
0110	2
0440	2
0550	3

4.3 Find the number of bookings made in Riyadh.

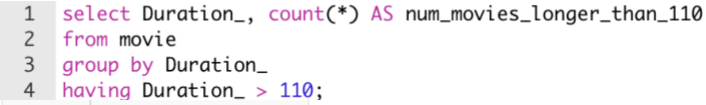


```
select city,count(*) AS num_bookings_in_Riyadh
from booking
group by city
having city='Riyadh';
```

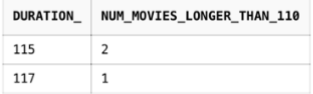


CITY	NUM_BOOKINGS_IN_RIYADH
Riyadh	5

4.4 Find the number of movies that have a duration more than 110 minutes.

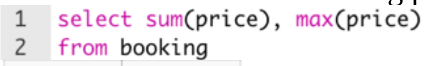


```
select Duration_, count(*) AS num_movies_longer_than_110
from movie
group by Duration_
having Duration_ > 110;
```

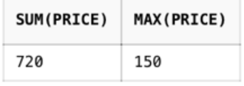


DURATION_	NUM_MOVIES_LONGER_THAN_110
115	2
117	1

4.5 Find the sum of all booking prices and the maximum price.

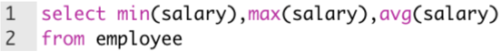


```
select sum(price), max(price)
from booking
```

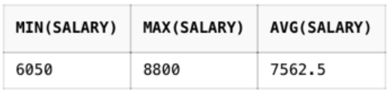


SUM(PRICE)	MAX(PRICE)
720	150

4.6 Find the minimum, maximum, and average salary of employees.



```
select min(salary),max(salary),avg(salary)
from employee
```

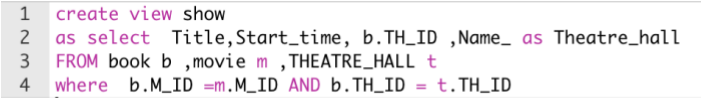


MIN(SALARY)	MAX(SALARY)	AVG(SALARY)
6050	8800	7562.5

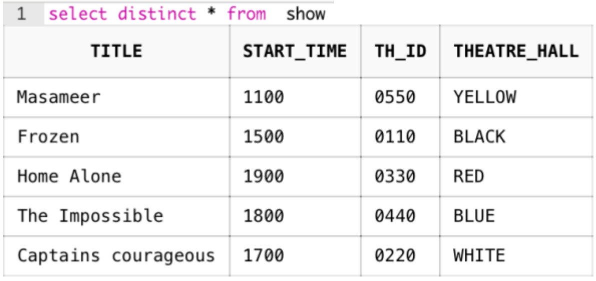
3.2.3 Views:

1. List 2 different views give SQL translations of them and indicate their implementation and solutions.

1.1 Create a view that returns movie title, start time of the movie, which theatre hall it's shown in and the name of that theatre hall.



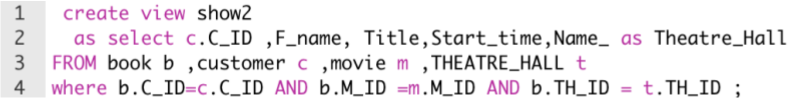
```
create view show
as select  Title,Start_time, b.TH_ID ,Name_ as Theatre_hall
FROM book b ,movie m ,THEATRE_HALL t
where  b.M_ID =m.M_ID AND b.TH_ID = t.TH_ID
```



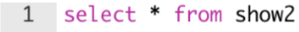
```
select distinct * from  show
```

TITLE	START_TIME	TH_ID	THEATRE_HALL
Masameer	1100	0550	YELLOW
Frozen	1500	0110	BLACK
Home Alone	1900	0330	RED
The Impossible	1800	0440	BLUE
Captains courageous	1700	0220	WHITE

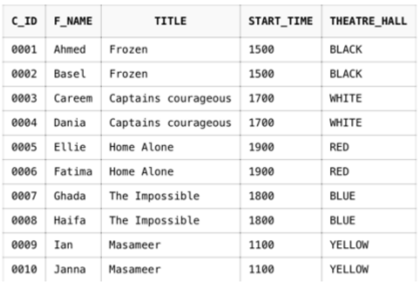
1.2 Create a view that returns customer ID, customer first name, movie title, start time of the movie and in which theatre hall it's shown in.



```
 create view show2
  as select c.C_ID ,F_name, Title,Start_time,Name_ as Theatre_Hall
FROM book b ,customer c ,movie m ,THEATRE_HALL t
where b.C_ID=c.C_ID AND b.M_ID =m.M_ID AND b.TH_ID = t.TH_ID ;
```



```
select * from show2
```



C_ID	F_NAME	TITLE	START_TIME	THEATRE_HALL
0001	Ahmed	Frozen	1500	BLACK
0002	Basel	Frozen	1500	BLACK
0003	Careem	Captains courageous	1700	WHITE
0004	Dania	Captains courageous	1700	WHITE
0005	Ellie	Home Alone	1900	RED
0006	Fatima	Home Alone	1900	RED
0007	Ghada	The Impossible	1800	BLUE
0008	Haifa	The Impossible	1800	BLUE
0009	Ian	Masameer	1100	YELLOW
0010	Janna	Masameer	1100	YELLOW