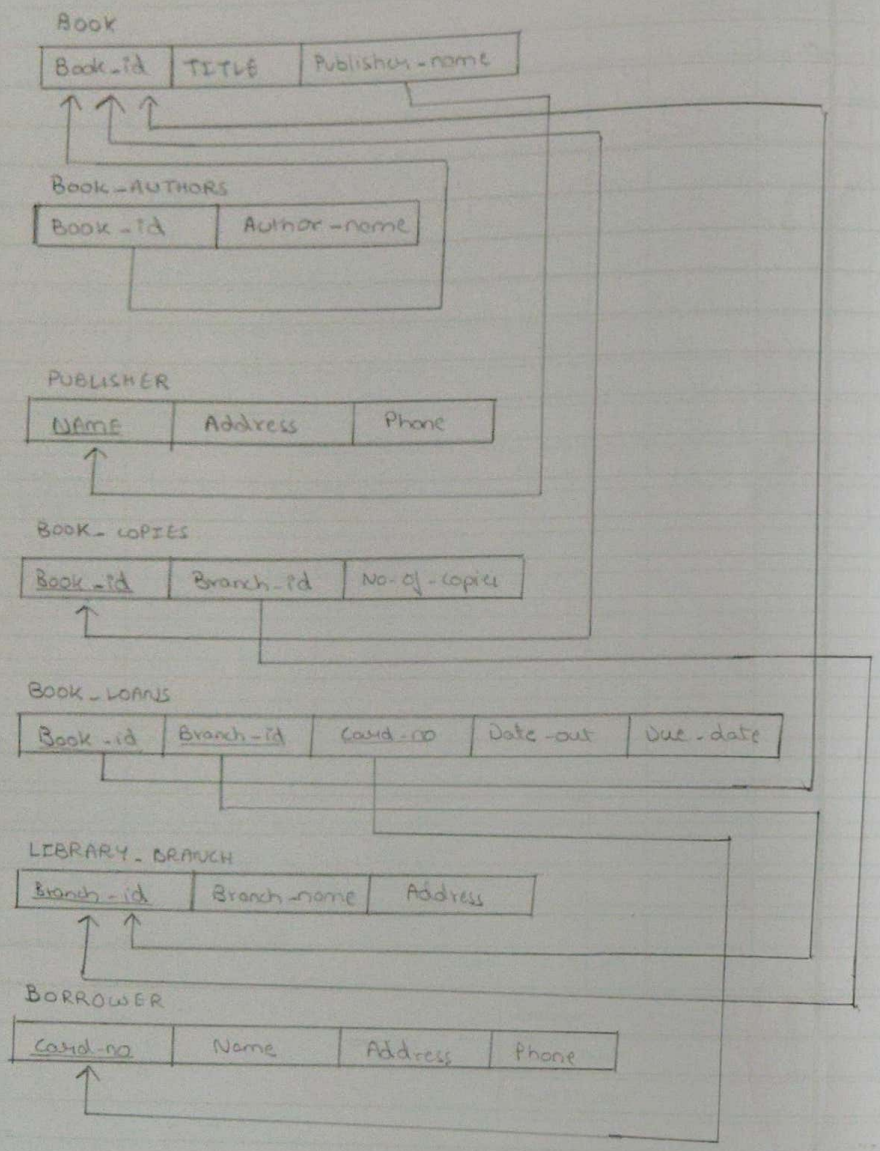


Date :

Exp. No. 01

Exp. Title

Page No.

Consider the following schema for a Library Database:
BOOK (Book_id, Title, Name, Pub_Years)
BOOK_AUTHORS (Book_id, Authod_Name)
PUBLISHER (Name, Address, Phone)
BOOK_COPIES (Book_Id, Branch_id, No_of_Copies)
BOOK_LENDING (Book_id, Branch_id, Card No, Date_out, Due_date)
LIBRARY_BRANCH (Branch_id, Branch_Name, Address)

Create Table:

Create table publisher (name varchar(18) primary key, address varchar(10), phone varchar(10));

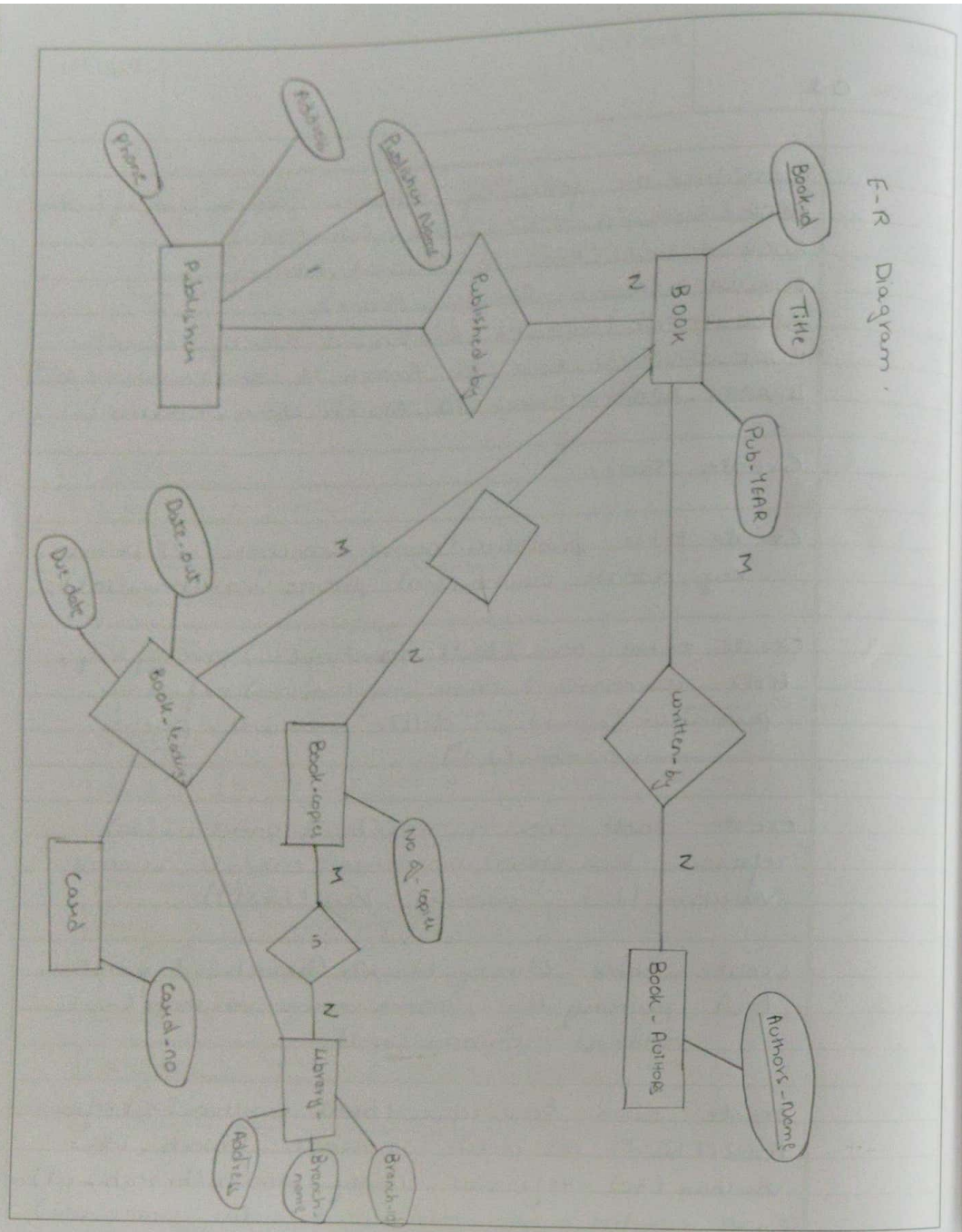
Create table book (bkid varchar(30) primary key, title varchar(30), name varchar(20) references publisher (name) on delete cascade, pyear number(4));

create table book_authors (bkid varchar(20) references book (bkid) on delete cascade, aname varchar(20), primary key (bkid));

create table library_branch (branch_id varchar(30) primary key, branch_name varchar(30), address varchar(30));

create table book_copies (bkid varchar(30) references book1 (bkid) on delete cascade, branch_id varchar(30) references library_branch1 (branch_id) on cascade, cardno number, date_out date, due_date date);

E-R Diagram:



Date :	Exp. Title	Page No.
Exp. No.		

Insert and Display commands.

```
insert into publisher values ('&name', '&address',
                              & phone);
insert into bookl values ('&bkid', '&title',
                          '&name', & pyear);
insert into book_authors values ('&bkid', '&aname');
insert into library_branch values ('&branchid',
    '& branch_name', '& address');
insert into book_copies values ('&bkid', '&branch_id
                                & numcopies);
insert into
book_lending values ('&bkid', '&branchid',
                     &cardno, '&date_out', '&due_date');
```

QUERIES

1. Retrieve details of all the books in the library at (BKID, TITLE, PUBLISHER NAME, AUTHOR, NO OF COPIES)

```
Select b.bkid, b.title, b.name, ba.aname, bc.branch
    id, bc.numcopies
from book b, book_authors ba, book_copies bc
where b.bkid = ba.bkid and
b.bkid = bc.bkid
```

BKID	TITLE	NAME	NAME	BRANCH_ID	Num Copies
CS111	C	PHI	Navathe	KUCSE01	20
CS112	C++	pearson	Scott	KUEEE02	34
CS113	DBMS	gold	Kothari	KUECE01	33

Select * from Publisher;

NAME	ADDRESS	PHONE
PHI	Delhi	9877777777
Pearson	delhi	9812345678
gold	bangalore	9844123345
Star	mysore	9898998666
vikas	Hubli	9889965116

Select * from book;

BKID	TITLE	NAME	PYEAR
CS111	C	PHI	2010
CS112	C++	pearson	2012
CS113	DBMS	gold	2015
CS114	CNS	gold	2020
CS115	Python	vikas	2023

Select * from book-authors:

BKID	ANAME
CS111	Nawathe
CS112	Scott
CS113	Kottur
CS114	Sabesh
CS115	Teerance

Select * from Library-branch

BRANCH ID	BRANCH-NAME	ADDRESS
KUCSE01	CSE	BALARI
KUECE02	EEE	HOSPET
KUEEE03	ECE	MYSORE
KUAIML01	AIML	HOSPETE
KUDATO1	DATA	DELHI

Date :	Exp. Title	Page No.
Exp. No.		

2. Get the particulars of borrowers who have borrowed more than three books but from JAN-2017 to jun-2017

```
select b.bkid, bl.branchid, bl.cardno
from book b, book_lending bl
where b.bkid = bl.bkid and bl.cardno
in
(select cardno
from book_lending bl
where due_date between '1-JAN-2016' and '1-SEP-2022'
and date_out between '1-JAN-2016' and '1-SEP-2022'
group by cardno having count(cardno)>=3);
```

BKID	BRANCHID	CARDNO
CS111	KUAIML01	111
CS112	KUEEE02	111
CS113	KUECE01	111
CS114	KUECE01	111

3. Delete a book in Book table. Update the contents of other tables to reflect this manipulation operation.

```
delete from book
where bkid = '115';
```

1 row deleted

Select * from book-copies;

BKID	BRANCH-ID	NUM COPIES
CS111	KUBCSE01	20
CS112	KUEEE02	34
CS113	KUECE01	33
CS114	KUDATA01	70
CS115	KUAIML01	22

Select * from book-lending;

BKID	BRANCHID	CARDNO	DATE-OUT	DUE-DATE
CS111	KUAIML01	111	17-AUG-16	20-SEP-16
CS112	KUEEE02	111	03-JUL-20	05-SEP-20
CS113	KUECE01	111	03-JUL-16	05-AUG-16
CS114	KUECE01	111	12-AUG-23	20-SEP-23
CS115	KUCSE01	112	13-SEP-22	25-SEP-22
CS116	KUECE02	1234	22-JAN-22	25-JAN-22

Date :
Exp. No.

Exp. Title

Page No.

4. Partition the Book table based on year of publication. Demonstrate its working with a simple query.

```
select bkid, title, name, pyear
from book
group by pyear, bkid, title, name;
```

Bkid	TITLE	NAME	PYEAR
CS114	CNS	gdd	2010
CS113	DBMS	gold	2015
CS112	C++	pearson	2012
CS115	python	vikas	2023
CS114	C	PHI	2010

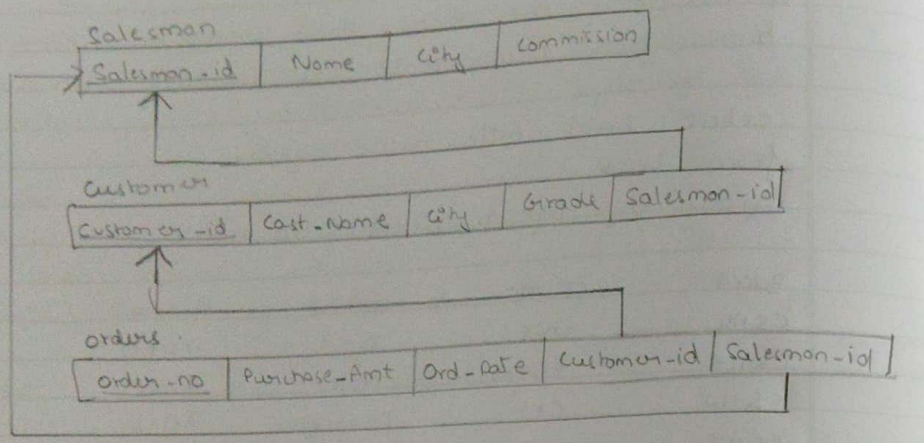
5. Create a view of all books and its no of copies that are currently available in the library.

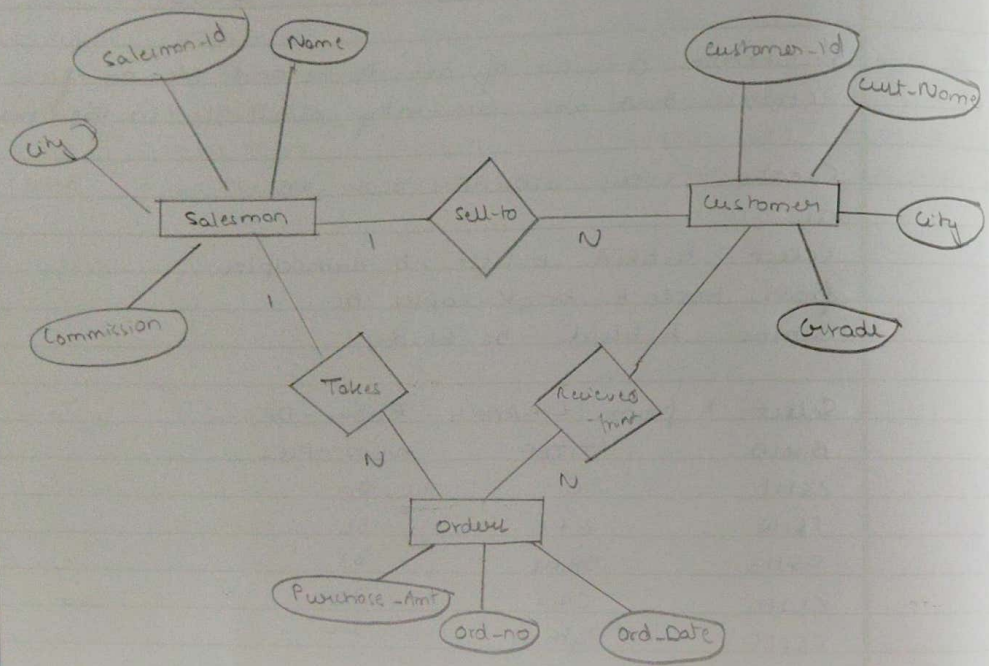
```
Create view LIBRARY-BOOK-DB.
as
select b.bkid, b.title, bc.numcopies
from book b, book-copies bc
where b.bkid = bc.bkid;

Select * from LIBRARY-BOOK-DB;
```

BKID	TITLE	NUMCOPIES
CS111	C	20
CS112	C++	34
CS113	DBMS	33
CS114	CNS	70
CS115	Python	22

SCHEMA DIAGRAM





Date :

Exp. No. 02

Exp. Title

Page No.

Consider the following schema for order database:
SALESMAN (Salesman-id, Name, city, commission)
CUSTOMER (Customer-id, cust-Name, city, Grade, salesman-id)
ORDERS (ord-NO, purchase-Amt, ord-Date, customer-id, salesmen-id).

CREATION OF TABLES:

```
Create table sales (sid number primary key,
  sname varchar(10), scity varchar(10), commission number);

Create table customer (cid number primary key,
cname varchar(10), city varchar(10), grade number(2),
sid number references sales(sid) on delete cascade

Create table orders (ono number(5) primary key,
pamount number, odate date, cid number
references customer(cid) on delete cascade, sid
number reference sales(sid) on delete cascade);
```

Insert:

```
insert into sales values (&sid, '&sname', '&city',
                          '&commission');
insert into customer values (&cid, '&cname',
                          '&city', &grade, &sid);
insert into orders values (&no, &pamount, '&odate',
                          &cid, &sid);
```

Select * from Sale;

SID	SNAME	SCITY	COMMISSION
111	David	Bangalore	10
112	Sam	mysore	20
113	SK	Ballari	20
114	pavan	Mysore	30
115	ram	Hubli	40

Select * from customer;

CID	CNAME	CITY	GRADE	SID
221	priya	Banglore	100	111
222	Suma	mysore	200	111
223	malli	Ballari	111	111
224	Raj	Hospet	100	111
225	Ravi	Raichur	500	112
226	Vijay	Banglore	200	113

Select * from orders;

ONO	PAmount	ODATE	CID	SID
551	6574	02-JAN-17	221	111
552	43251	05-FEB-17	222	113
553	3426	06-MAR-17	224	111
554	3527	16-MAR-17	224	113
556	2300	02-MAR-25	225	111
557	3400	16-JAN-25	221	112
558	2435	20-JAN-17	224	114

Date :
Exp. No.

Exp. Title

Page No.

Queries.

Q1) Count the customers with grades above the bangalore's average.

```
Select grade, count(*) as NO_OF_CUSTOMERS
from customer
group by grade
having grade > (select avg(grade)
from customer
where city = 'Banglore');
```

GRADE	NO_OF_CUSTOMERS
500	1
200	2

Q2) Find the name and no of all salesmen who had more than one customers.

```
Select Sname, sid
from Sale
where sid in
(select sid from orders
 group by sid
 having count(cid)>1);
```

David	111
SK	113

Date :
Exp. No.

Exp. Title

Page No.

Q5) List all the salesman and indicate those who have and dont have in their city (use UNION Operation)

```
(select s.sid, s.sname, s.scity, c.cid, c.city, c.cname
 from salu s, customers c
 where s.sid = c.sid and c.city = s.scity)
 UNION
(select s.sid, c.cname, c.scity, c.cid, c.city,
        'NO CUSTOMERS IN CITY'
 from salu s, customers c
 where c.sid = c.sid and c.city != s.scity);
```

SID	SNAME	SCITY	CID	CITY	CNAME
111	David	Bangalore	221	Banglore	NO CUSTOMER IN CITY
111	David	Bangalore	222	Mysore	NO CUSTOMER IN CITY
111	David	Bangalore	223	Ballari	NO CUSTOMER IN CITY
111	David	Bangalore	224	Hospet	NO CUSTOMER IN CITY
112	Sam	mysore	225	Raichur	NO CUSTOMER IN CITY
113	Sk	Ballari	226	Banglore	NO CUSTOMER IN CITY

Q6) Create a view that finds the salesmen who has the customer with the highest order of a day.

```
SQL > create view HIGHEST_ORDERS
as select s.sid, s.sname, o1.odate, o1.pamount
from salu s, orders o1
where s.sid = o1 and o1.pamount =
(select max(o2.pamount) from orders o2 where
    o1.odate = o2.odate);

view created.
```

Date :

Exp. No.

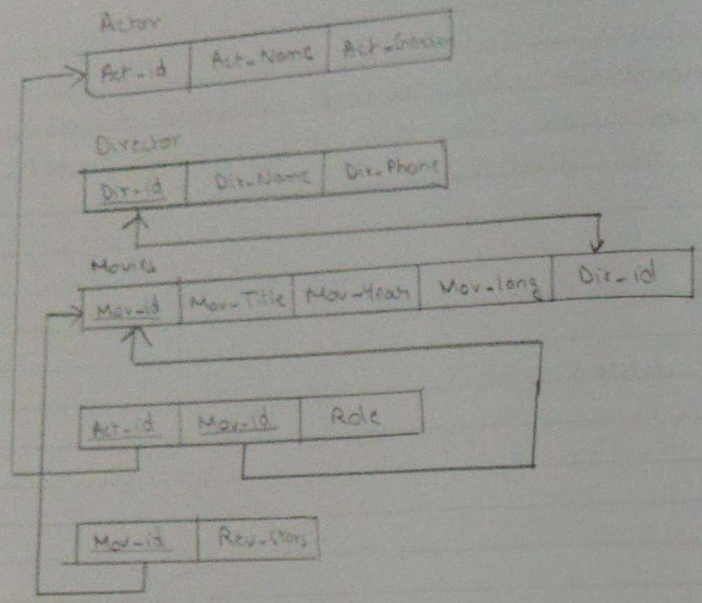
Exp. Title

Page No.

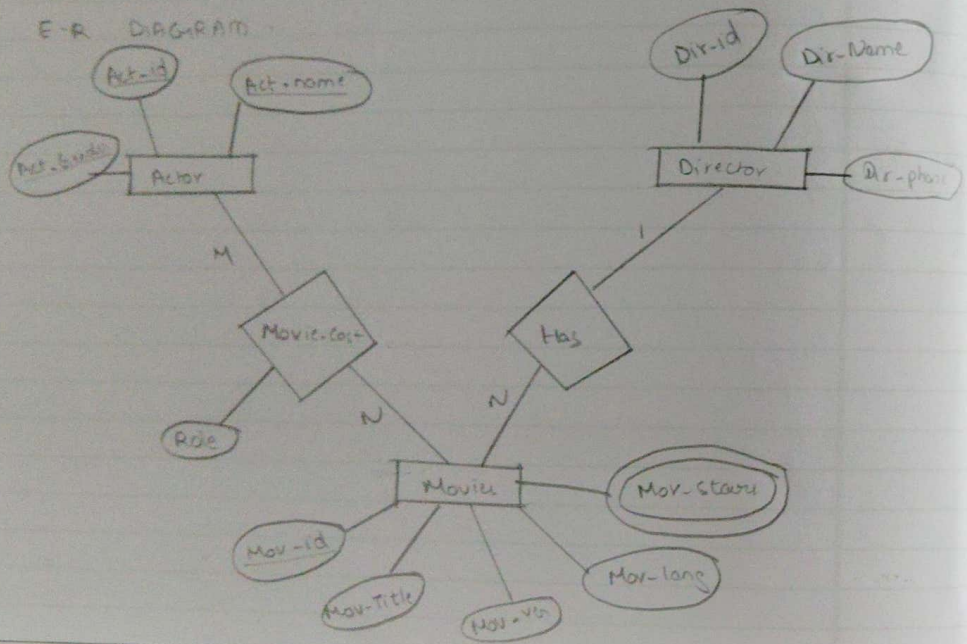
05) Demonstrate the DELETE operation by removing salesman with ID 1000 all his orders must also be deleted.

```
delete from sales
where sid = 118;

1 row deleted.
```



E-R DIAGRAM



Date:	Exp. Title MOVIES DATABASE	Page No.
Exp. No. 03		

Consider the schema for movie Database:
ACTOR (Act-id, Act-Name, Act-Gender)
DIRECTOR (Dir-id, Dir-Name, Dis-phone)
MOVIES (mov-id, mov-Title, mov-year, mov-lang, Dir-id)
MOVIES-CAST (Act-id, mov-id, Role)
RATING (mov-id, Rev-Stars)

Write SQL queries to

1. List the titles of all movies directed by 'Hitchcock'
2. Find the movie names where one or more actors acted in two or more movies.
3. List all actors who acted in a movie before 2000 and also in a movie after 2015 (use Joins operation)
4. Find the title of movies and number of stars for each movie that has at least one rating and finding the highest no. of stars that movie received sort the result by movie title.
5. Update rating of all movies directed by 'Steven Spielberg' to 5

CREATON OF TABLES.

```
Create table actor (actid number primary key,
    aname varchar (10), gender varchar (6));

Create table director (did number primary key,
    dname varchar (30), phone number);
```

Date :
Exp. No.

Exp. Title

Page No.

```
create table movies (mid number primary key,
mtitle varchar(10), myear number, mlang
varchar(10), did number(5) references
director(did) on delete cascade);

create table movie-cast (actid number
references actor(actid) on delete cascade,
mid number references movies(mid) on delete
cascade, role varchar(10));

create table rating (mid number references movies(mid)
on delete cascade, revstars numbers);

Insert:

insert into actor values (&aid, &aname, &gender);

insert into director values (&did, '&dname', &phone);

insert into movies values (&mid, '&mtitle', '&mlang',
                           &myear, &did);

insert into movie-cast values (&aid, &mid, '&role');

insert into rating values (&mid, &revstars);

Queries:
Q1> List the titles of all the movies directed
    by "hitchcock".
```

Select * from actor;

ACTID	ANAME	GENDER
111	Sam	male
112	bob	male
113	Ariyana	female
114	David	male
115	jim	male
116	kim	female
117	puneeth	male.

7 rows selected.

Select * from director;

DID	DNAME	PHONE
2111	hitchcock	5647382
2112	Steven	657484
2113	john watts	56767
2114	Santosh	56764
2115	John	12345

Select * from movies;

MID	MTITLE	MYEAR	MLANG	DID
500	spark	2013	English	2112
501	rwindow	1954	English	2111
503	Spiderman	2017	English	2113
504	Rajkumar	2020	English	2114
505	kuri	2000	English	2115

Date: | Exp. Title | Page No.

Exp. No.

Select m.mid, m.mtitle
from movies m, director d
where m.did = d.did and d.dname = 'hitchcock';

MID MTITLE
501 rwindow.

Q2) Find the movie names where one or more actors acted in two or more movies.

Select m.mtitle
from movies m, movie-cast mc
where m.mid = mc.mid and mc.actid in
(Select actid from movies-cast
group by actid
having count (actid) >= 2)
group by m.mtitle
having count (*) >2;

MTITLE
spiderman.

Q3) List all actors who acted in a movie before 2000 and also in a movie after 2015 (use JOIN, operation).

Select a.aname, m1.mtitle, m1.myear, m2.year
from actor a, movies m1, movies m2, movie-cast c1, movie-cast c2

Select * from actor movie - cast ;

ACT ID	MID	Role
111	501	main lead
112	503	Supporting
113	501	hero in
114	503	main lead
115	503	negative
116	503	heroin
111	504	hero
112	503	main lead
114	501	Supporting.
115	500	negative

Date :
Exp. No.
Exp. Title
Page No.

where m1.myear < 2000 and m2.myear > 2015 and a.actid = c1.actid .
and c1.mid = m1.mid and a.actid = c2.actid
and c2.mid = m2.mid ;

ANAME	MTITLE	MYEAR	MYEAR
Bob	window	1954	2017
Sam	window	1954	2017

Q4) Find the title of movies and numbers of stars for each movie that has at least one rating, find the highest number of stars that movie received. Return the movie title and number of stars. sort by movie title.

Select m.mtitle, max (r.revstars)
from rating r, movies m
where m.mid = r.mid
group by m.mtitle
order by m.mtitle.

MTITLE	MAX (R.REVSTARS)
Raj kumar	4
jpark	5
window	3
Spidermen	3

Q5) Update rating of all movies directed by 'steven spielberg' to 5.
Update rating

Select * from movie-cast rating;

MID	REVSTARS
500	5
501	3
503	3
504	4
500	3
501	2
504	4
503	3
504	4
500	2
500	3

9 rows created.

Date :
Exp. No.

Exp. Title

Page No.

```
Set revstars = '5'
where mid in
(select m.mid
 from movies m, director d
 where m.did = d.did and dname = 'Steven');
2 rows updated
```