VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

DATA BASE MANAGEMENT SYSTEM
LAB REPORT

(19CS4PCDBM)

Submitted by

RAVI SAJJANAR (1BM19CS127)

Under the Guidance of
Prof. Sheetal V A
Assistant Professor, BMSCE

in partial fulfillment for the award of the degree of
BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING

B.M.S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)
BENGALURU-560019

Mar-2021 to Jun-2022

B. M. S. College of Engineering,

Bull Temple Road, Bangalore 560019

(Affiliated To Visvesvaraya Technological University, Belgaum)

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the Lab Assignment work entitled "**Database Management System** " carried out by **RAVI SAJJANAR (1BM19CS127)** who is the bonafide students of **B. M. S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visveswaraiah Technological University, Belgaum during the year 2021-2022. The Lab report has been approved as it satisfies the academic requirements in respect of **Database Management System (19CS4PCDBM) LAB** work prescribed for the said degree.

Signature of the Guide
Prof. Prof. Sheetal VA
Assistant Professor
BMSCE, Bengaluru

Signature of the HOD
Dr. Umadevi V
Associate Prof.& Head, Dept. of CSE
BMSCE, Bengaluru

External Viva

Name of the Examiner | Signature with date

1._____ _____

2. _____ _____

INDEX

S.NO	PROGRAM	PAGE
01	Insurance Database	4
02	Banking Enterprise Database	15
03	Supplier Database	23
04	Student Faculty Database	32
05	Airline Flight Database	42
06	Order Processing Database	56
07	Book dealer Database	67
08	Student Enrolment Database	75
09	Movie Database	84
10	College Database	95

PROGRAM 1: INSURANCE DATABASE

Consider the Insurance database given below. The data types are specified.

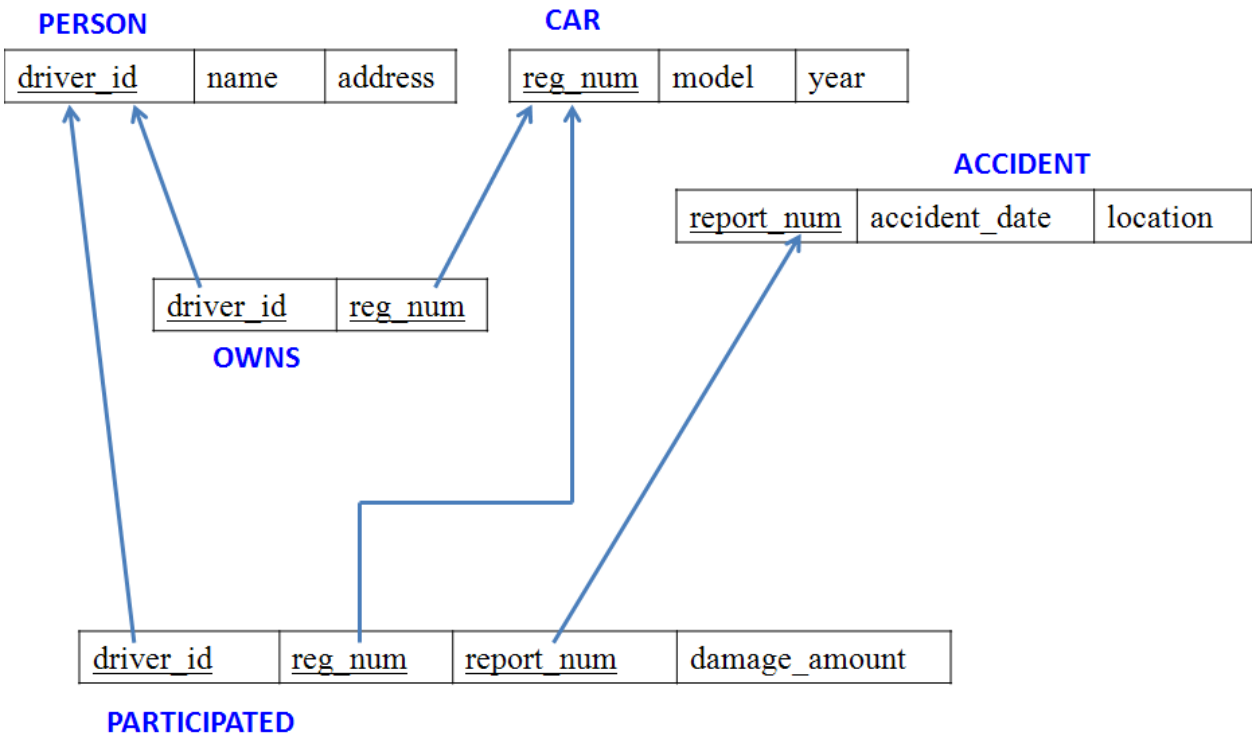
PERSON (driver_id: String, name: String, address: String)
CAR (reg_num: String, model: String, year: int)
ACCIDENT (report_num: int, accident_date: date, location: String)
OWNS (driver_id: String, reg_num: String)
PARTICIPATED (driver_id: String, reg_num: String, report_num: int, damage_amount: int)

i) Create the above tables by properly specifying the primary keys and the foreign keys.
ii) Enter at least five tuples for each relation.
iii) Demonstrate how you

a. Update the damage amount to 25000 for the car with a specific reg-num(example 'K A053408') for which the accident report number was 12.
b. Add a new accident to the database.

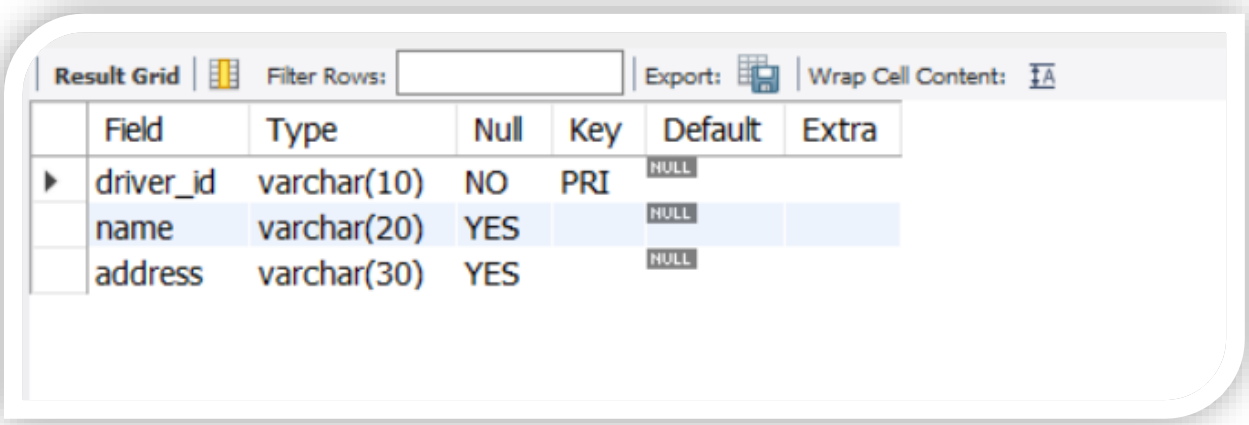
iv)Find the total number of people who owned cars that involved in accidents in 2008.
v)Find the number of accidents in which cars belonging to a specific model (example)were involved.

Schema Diagram:



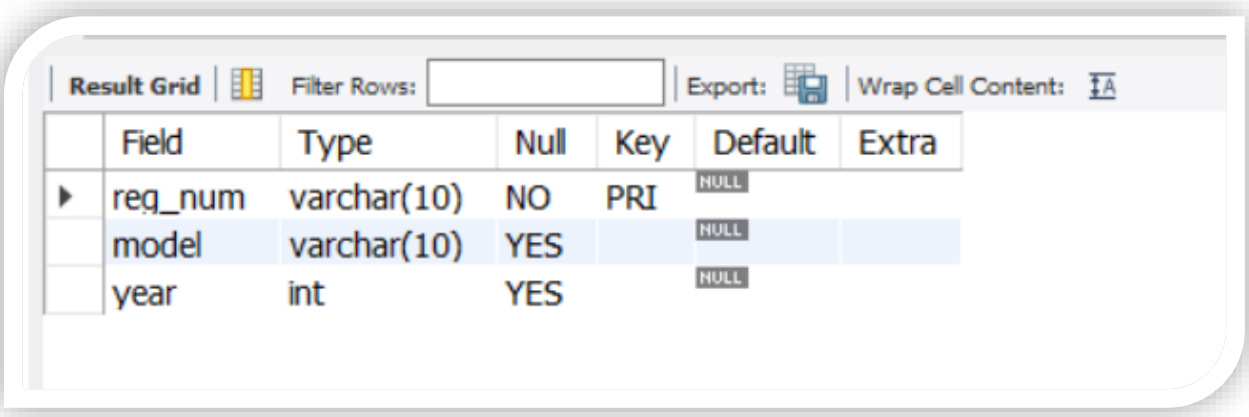
-- Creating the database and the tables

```
create database insurance;
use insurance;

create table person(
        driver_id varchar(10),
        name varchar(20),
        address varchar(30),
        primary key(driver_id)
);

desc person;
```



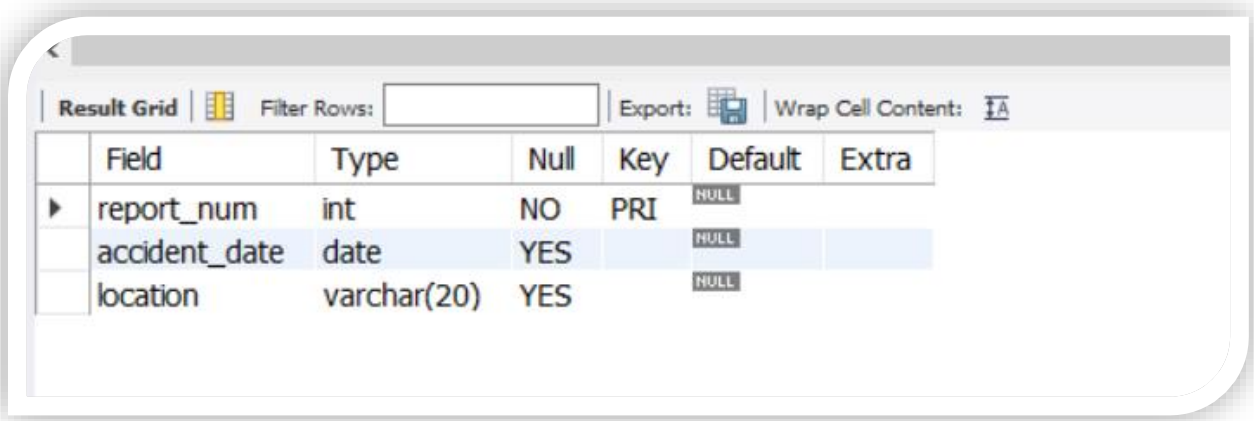
Field	Type	Null	Key	Default	Extra
driver_id	varchar(10)	NO	PRI	NULL	
name	varchar(20)	YES		NULL	
address	varchar(30)	YES		NULL	

```
create table car(
        reg_num varchar(10),
        model varchar(10),
        year int,
        primary key(reg_num)
);

desc car;
```



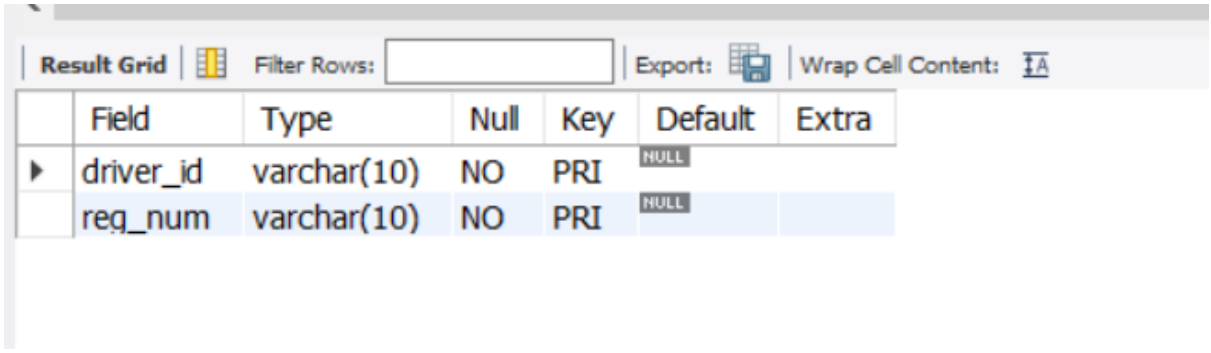
Field	Type	Null	Key	Default	Extra
reg_num	varchar(10)	NO	PRI	NULL	
model	varchar(10)	YES		NULL	
year	int	YES		NULL	

```
create table accident(
        report_num int,
        accident_date date,
        location varchar(20),
        primary key(report_num)
);
```



Field	Type	Null	Key	Default	Extra
report_num	int	NO	PRI	NULL	
accident_date	date	YES		NULL	
location	varchar(20)	YES		NULL	

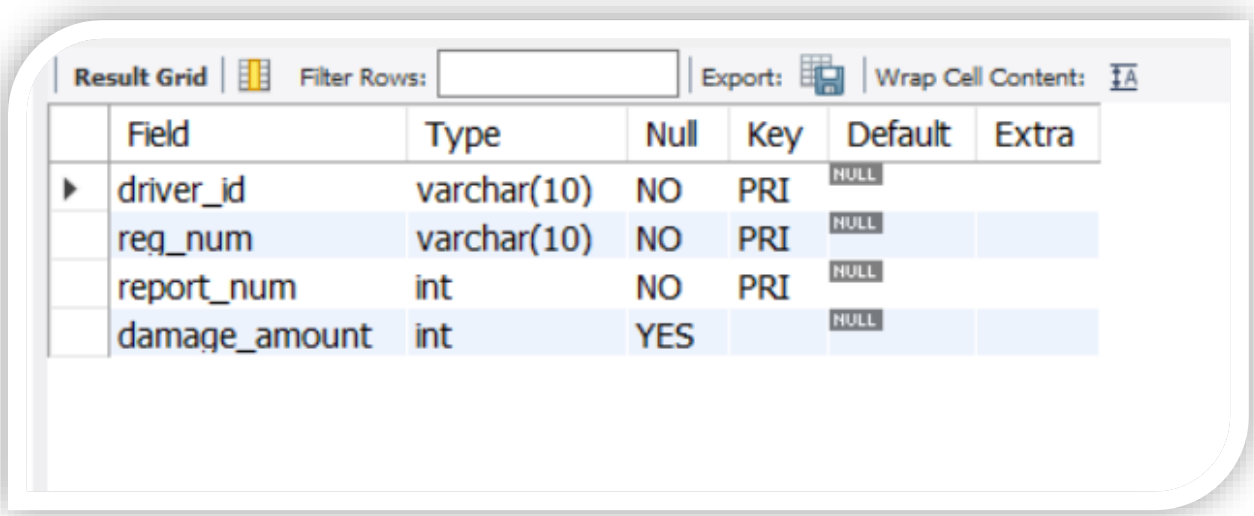
```
create table owns(
        driver_id varchar(10),
        reg_num varchar(10),
        primary key(driver_id,reg_num),
        foreign key(driver_id) references person(driver_id),
        foreign key(reg_num) references car(reg_num)
);

desc owns;
```



Field	Type	Null	Key	Default	Extra
driver_id	varchar(10)	NO	PRI	NULL	
reg_num	varchar(10)	NO	PRI	NULL	

```
create table participated(
        driver_id varchar(10),
        reg_num varchar(10),
        report_num int,
        damage_amount int,
```

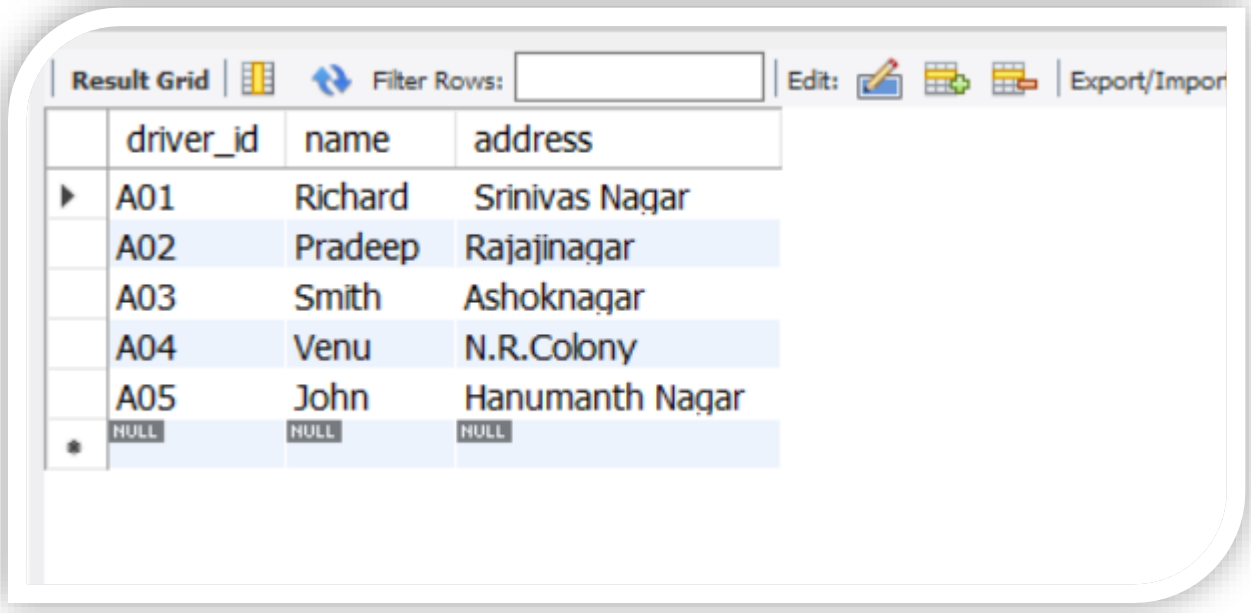
```
        primary key(driver_id,reg_num,report_num),
        foreign key(driver_id) references person(driver_id),
        foreign key(reg_num) references car(reg_num),
        foreign key(report_num) references accident(report_num)
);

desc participated;
```



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

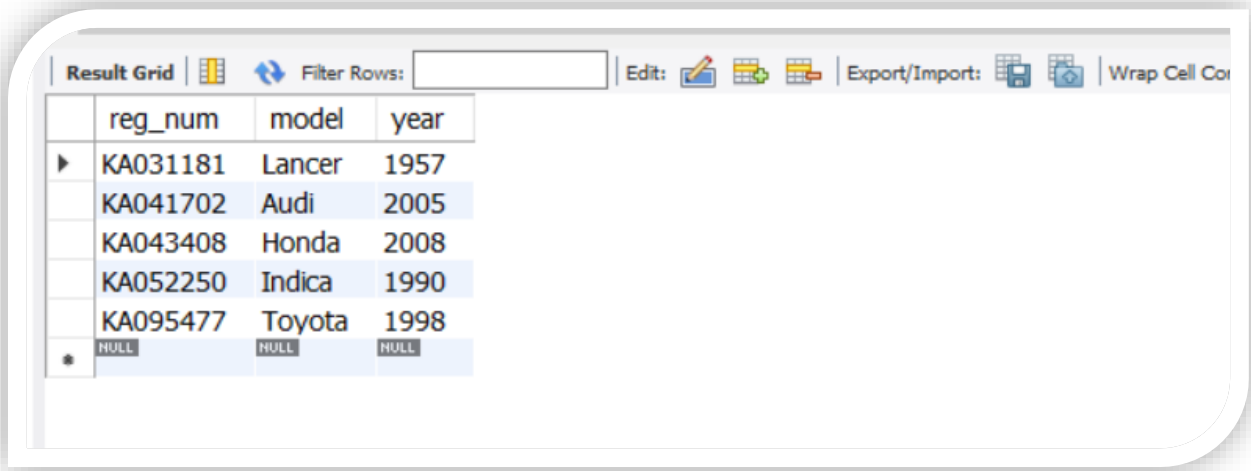
Field	Type	Null	Key	Default	Extra
driver_id	varchar(10)	NO	PRI	NULL	
reg_num	varchar(10)	NO	PRI	NULL	
report_num	int	NO	PRI	NULL	
damage_amount	int	YES		NULL	

```
insert into person values('A01','Richard',' Srinivas Nagar');
insert into person values('A02','Pradeep','Rajajinagar');
insert into person values('A03','Smith','Ashoknagar');
insert into person values('A04','Venu','N.R.Colony');
insert into person values('A05','John','Hanumanth Nagar');

commit;

select * from person;
```



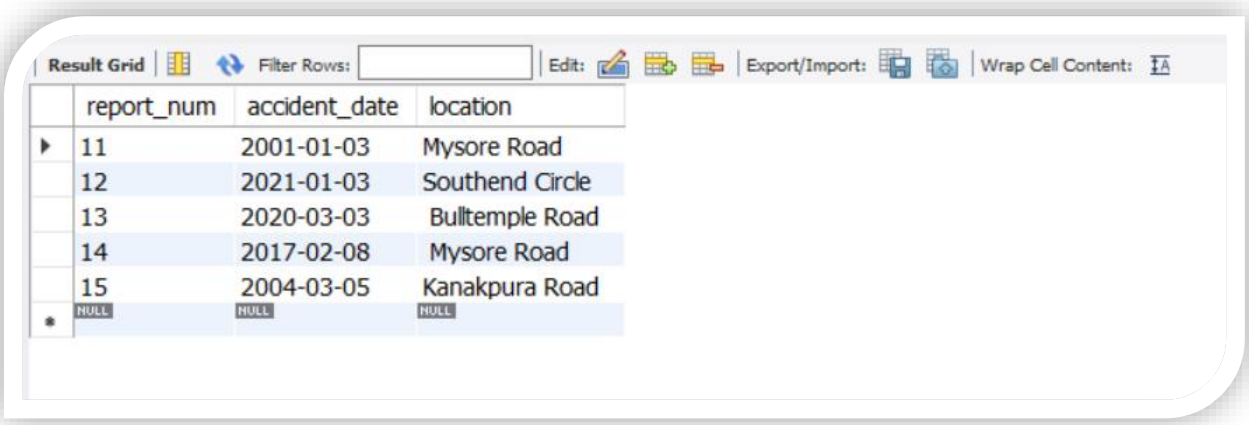
driver_id	name	address
A01	Richard	Srinivas Nagar
A02	Pradeep	Rajajinagar
A03	Smith	Ashoknagar
A04	Venu	N.R.Colony
A05	John	Hanumanth Nagar
NULL	NULL	NULL

```
insert into car values('KA031181','Lancer',1957);
insert into car values('KA041702','Audi',2005);
insert into car values('KA043408','Honda',2008);
insert into car values('KA052250','Indica',1990);
insert into car values('KA095477','Toyota',1998);

commit;
select * from car;
```



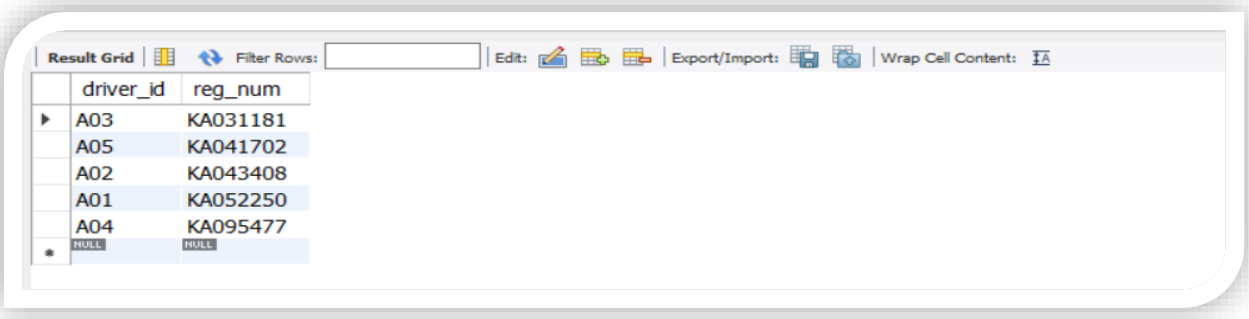
reg_num	model	year
KA031181	Lancer	1957
KA041702	Audi	2005
KA043408	Honda	2008
KA052250	Indica	1990
KA095477	Toyota	1998
NULL	NULL	NULL

```
insert into accident values(11,'2001-01-03','Mysore Road');
insert into accident values(12,'2021-01-03','Southend Circle');
insert into accident values(13,'2020-03-03',' Bulltemple Road');
insert into accident values(14,' 2017-02-08',' Mysore Road');
insert into accident values(15,'2004-03-05','Kanakpura Road');
commit;

select * from accident;
```



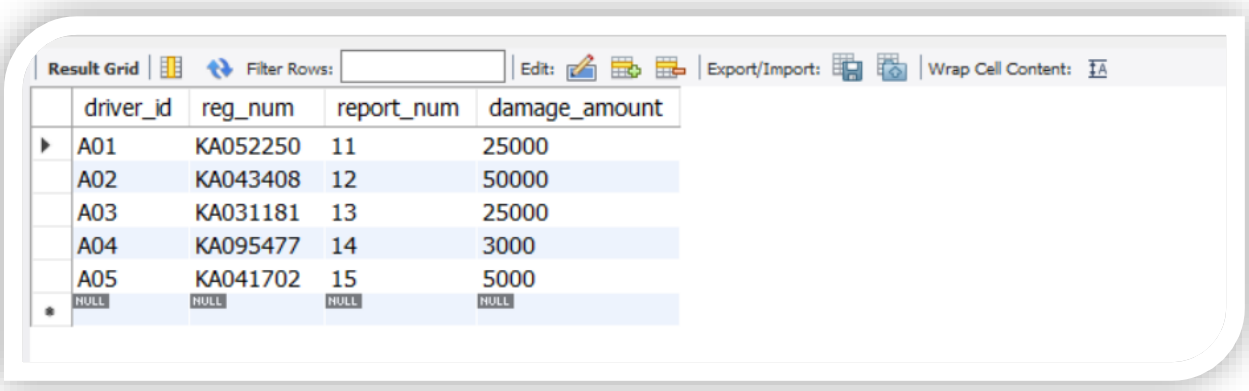
report_num	accident_date	location
11	2001-01-03	Mysore Road
12	2021-01-03	Southend Circle
13	2020-03-03	Bulltemple Road
14	2017-02-08	Mysore Road
15	2004-03-05	Kanakpura Road

```
insert into owns values ('A01','KA052250');
insert into owns values ('A02','KA043408');
insert into owns values ('A03','KA031181');
insert into owns values ('A04','KA095477');
insert into owns values ('A05','KA041702');
commit;
select * from owns;
```



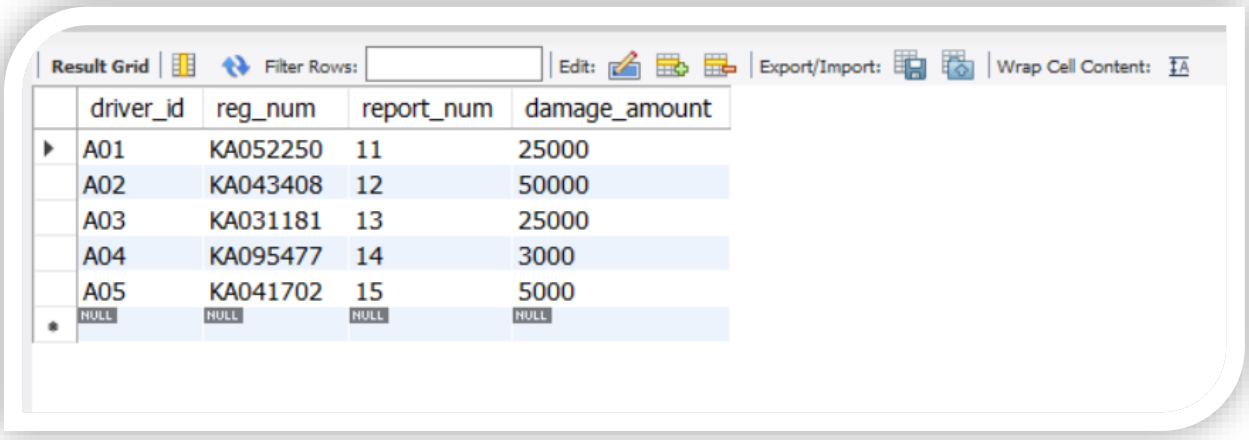
driver_id	reg_num
A03	KA031181
A05	KA041702
A02	KA043408
A01	KA052250
A04	KA095477

```
insert into participated values ('A01','KA052250',11, 25000);
insert into participated values ('A02','KA043408',12, 50000);
insert into participated values ('A03','KA031181',13, 25000);
insert into participated values ('A04','KA095477',14, 3000);
insert into participated values ('A05','KA041702',15, 5000);
commit;

select * from participated;
```



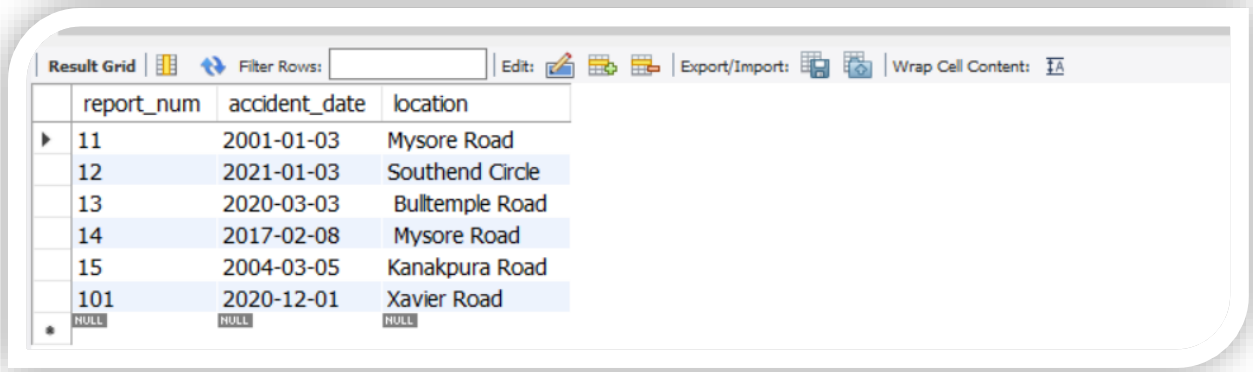
driver_id	reg_num	report_num	damage_amount
A01	KA052250	11	25000
A02	KA043408	12	50000
A03	KA031181	13	25000
A04	KA095477	14	3000
A05	KA041702	15	5000
NULL	NULL	NULL	NULL

```
update participated
set damage_amount = 2500
where reg_num='KA031111';

select * from participated;
```



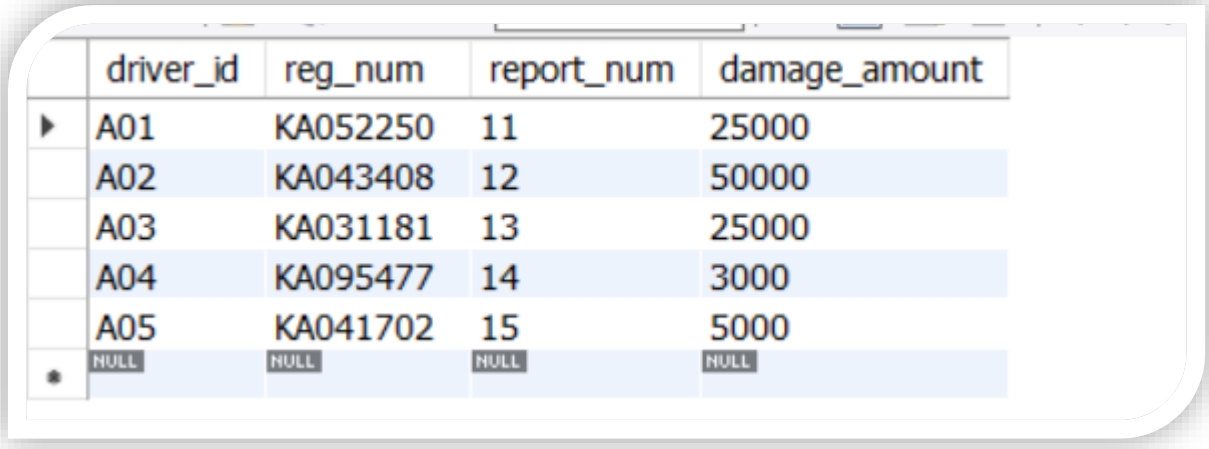
driver_id	reg_num	report_num	damage_amount
A01	KA052250	11	25000
A02	KA043408	12	50000
A03	KA031181	13	25000
A04	KA095477	14	3000
A05	KA041702	15	5000
NULL	NULL	NULL	NULL

insert into accident values(101,'2020-12-01','Xavier Road');
insert into participated values('A01','KA031111',101, 1001);
commit;
select * from accident;



report_num	accident_date	location
11	2001-01-03	Mysore Road
12	2021-01-03	Southend Circle
13	2020-03-03	Bulltemple Road
14	2017-02-08	Mysore Road
15	2004-03-05	Kanakpura Road
101	2020-12-01	Xavier Road
NULL	NULL	NULL

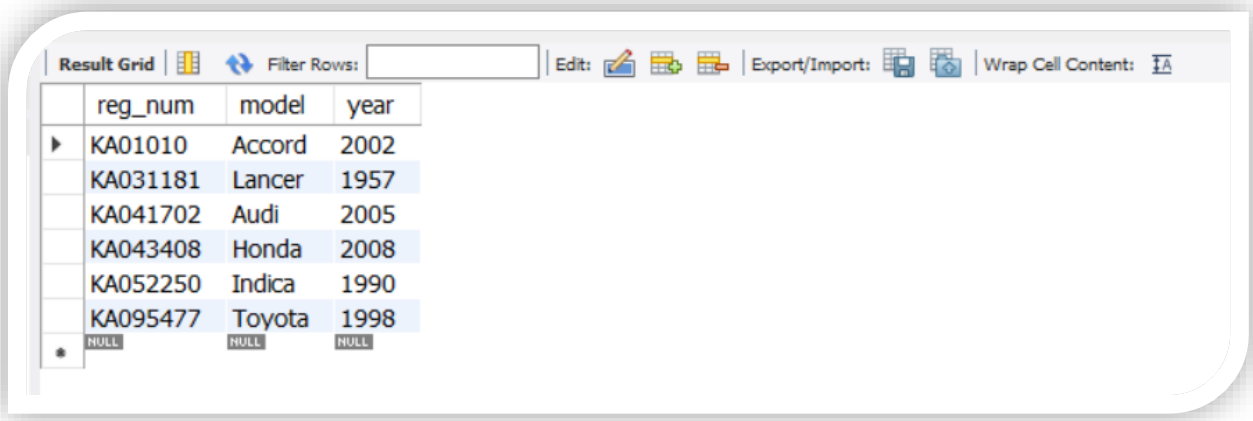
select * from participated;



driver_id	reg_num	report_num	damage_amount
A01	KA052250	11	25000
A02	KA043408	12	50000
A03	KA031181	13	25000
A04	KA095477	14	3000
A05	KA041702	15	5000
NULL	NULL	NULL	NULL

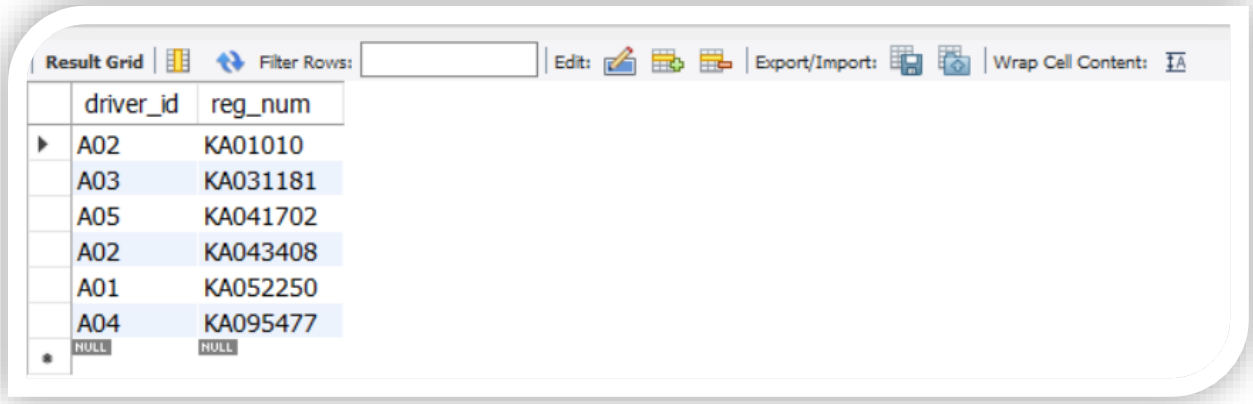
insert into car values('KA01010', 'Accord', 2002);
insert into owns values('A02', 'KA01010');
insert into accident values(200, '2008-12-01', 'Pinto Road');
insert into participated values('A02', 'KA01010', 200, 500);
commit;

select * from car;



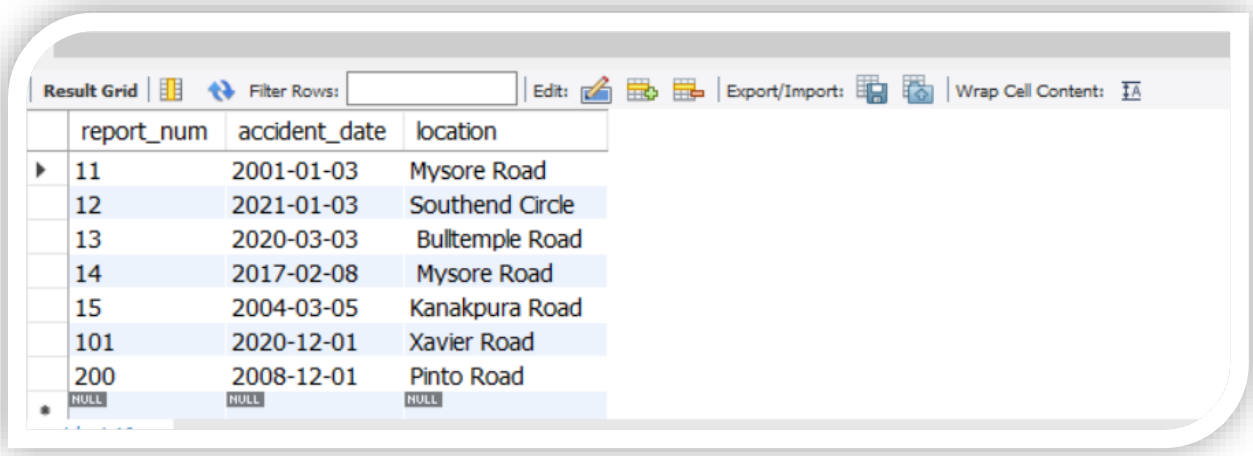
reg_num	model	year
KA01010	Accord	2002
KA031181	Lancer	1957
KA041702	Audi	2005
KA043408	Honda	2008
KA052250	Indica	1990
KA095477	Toyota	1998
NULL	NULL	NULL

select * from owns;



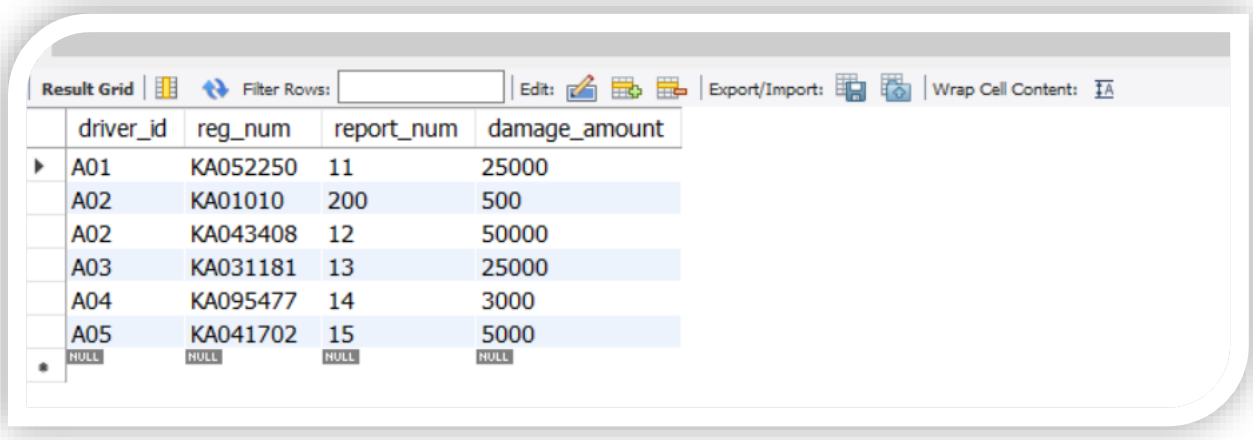
driver_id	reg_num
A02	KA01010
A03	KA031181
A05	KA041702
A02	KA043408
A01	KA052250
A04	KA095477
NULL	NULL

select * from accident;



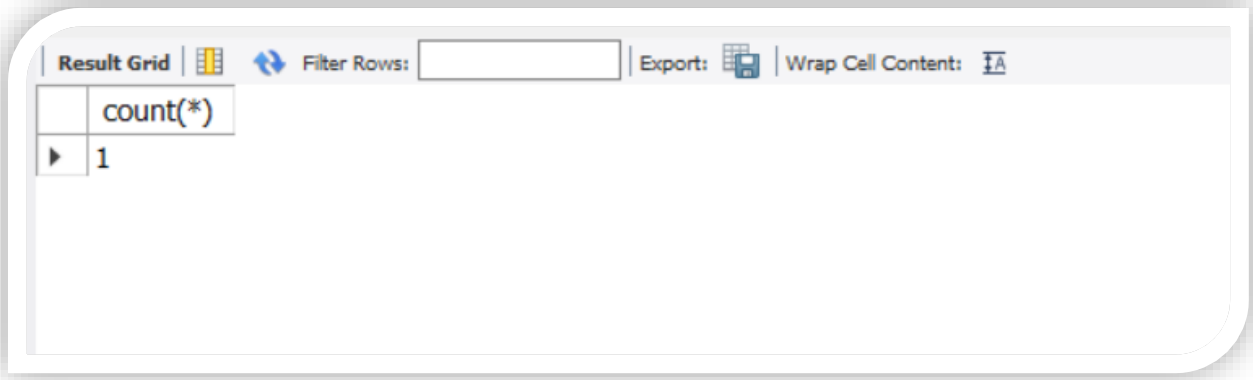
report_num	accident_date	location
11	2001-01-03	Mysore Road
12	2021-01-03	Southend Circle
13	2020-03-03	Bulltemple Road
14	2017-02-08	Mysore Road
15	2004-03-05	Kanakpura Road
101	2020-12-01	Xavier Road
200	2008-12-01	Pinto Road
NULL	NULL	NULL

select * from participated;



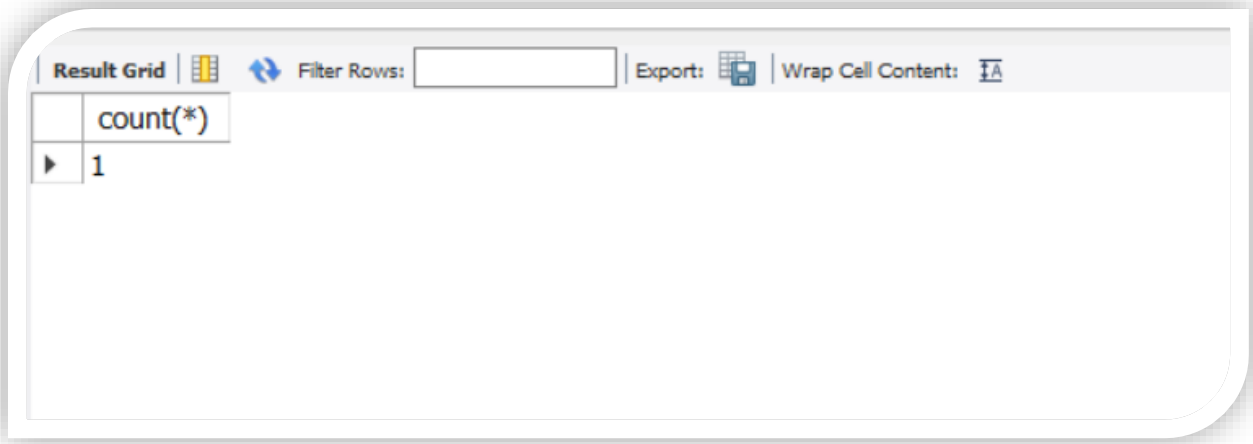
driver_id	reg_num	report_num	damage_amount
A01	KA052250	11	25000
A02	KA01010	200	500
A02	KA043408	12	50000
A03	KA031181	13	25000
A04	KA095477	14	3000
A05	KA041702	15	5000
NULL	NULL	NULL	NULL

select count(*) from accident where year(accident_date)=2008;



count(*)
1

select count(*) from participated where reg_num in (select reg_num from car where model="Accord");



count(*)
1

PROGRAM 2: **BANKING ENTERPRISE DATABASE**

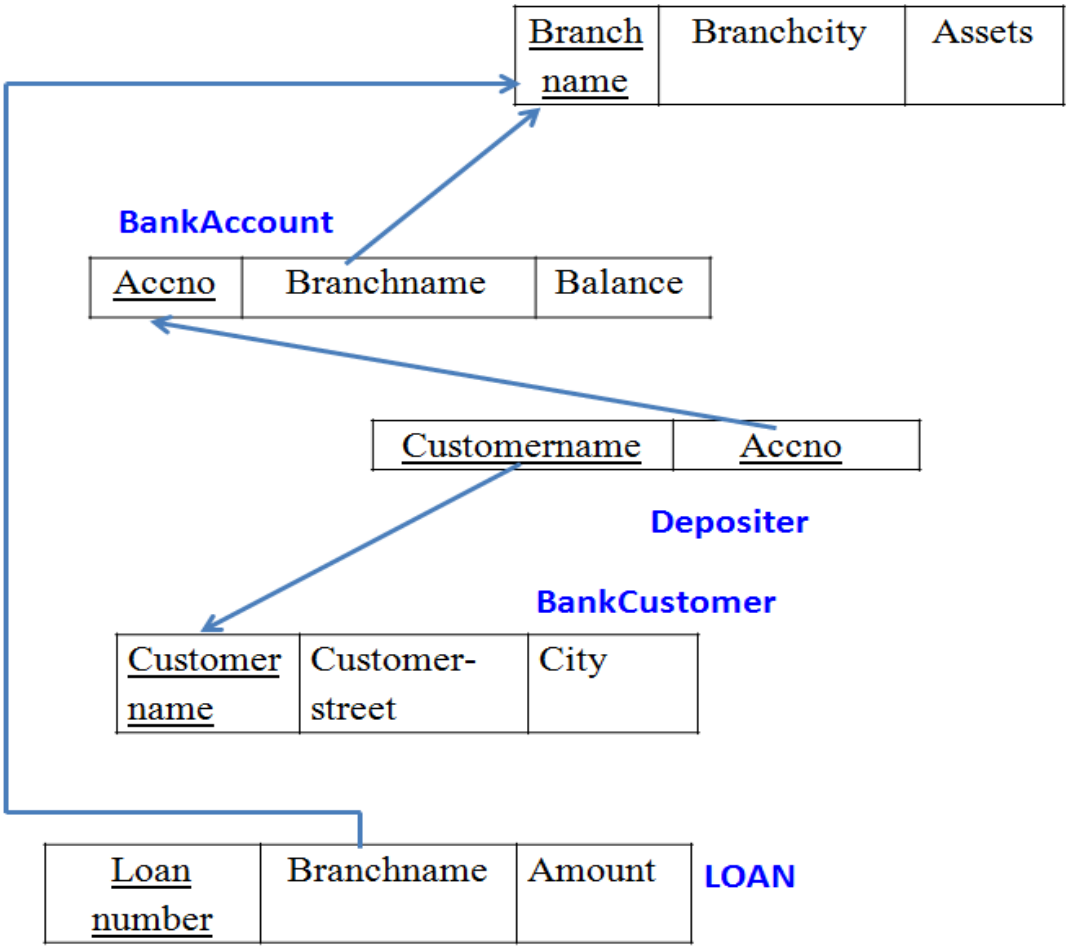
Consider the following database for a banking enterprise.

Branch (branch-name: String, branch-city: String, assets: real)
BankAccount(accno: int, branch-name: String, balance: real)
BankCustomer (customer-name: String, customer-street: String, customer-city: String)
Depositer(customer-name: String, accno: int)
Loan (loan-number: int, branch-name: String, amount: real)

i. Create the above tables by properly specifying the primary keys and the foreign keys.
ii. Enter at least five tuples for each relation.
iii. Find all the customers who have at least two accounts at the *Main* branch (ex. SBI_ResidencyRoad).
iv. Find all the customers who have an account at *all* the branches located in a specific city (Ex. Delhi).
v. Demonstrate how you delete all account tuples at every branch located in a specific city (Ex. Bombay).

INTRODUCTION: This database is developed for supporting banking facilities. Details of the branch along with the accounts and loans handled by them are recorded. Also details of the depositors of the corresponding branches are maintained.

Schema Diagram



-- Creating the Database and the tables

```
create database bank;
use bank;

create table branch (
        branch_name varchar(25),
    branch_city varchar(15),
    assets int,
    primary key (branch_name)
);
```

```
create table bank_account (
        accno int,
    branch_name varchar(25),
    balance int,
    primary key (accno),
    foreign key (branch_name) references branch(branch_name)
);

create table bank_customer (
        customer_name varchar(10),
    customer_street varchar(25),
    customer_city varchar(15),
    primary key (customer_name)
);

create table depositer (
        customer_name varchar(10),
        accno int,
    primary key(customer_name, accno),
    foreign key (customer_name) references bank_customer(customer_name),
    foreign key (accno) references bank_account(accno)
);

create table loan (
        loan_number int,
    branch_name varchar(25),
    amount int,
    primary key (loan_number),
    foreign key (branch_name) references branch(branch_name)
);

insert into branch values('SBI_Chamrajpet', 'Bangalore', 50000);
insert into branch values('SBI_ResidencyRoad', 'Bangalore', 10000);
insert into branch values('SBI_ShivajiRoad', 'Bombay', 20000);
insert into branch values('SBI_ParliamentRoad', 'Delhi', 10000);
insert into branch values('SBI_Jantarmantar', 'Delhi', 20000);
```

```
commit;

insert into bank_account values(1, 'SBI_Chamrajpet', 2000);
insert into bank_account values(2, 'SBI_ResidencyRoad', 5000);
insert into bank_account values(3, 'SBI_ShivajiRoad', 6000);
insert into bank_account values(4, 'SBI_ParliamentRoad', 9000);
insert into bank_account values(5, 'SBI_Jantarmantar', 8000);
insert into bank_account values(6, 'SBI_ShivajiRoad', 4000);
insert into bank_account values(8, 'SBI_ResidencyRoad', 4000);
insert into bank_account values(9, 'SBI_ParliamentRoad', 3000);
insert into bank_account values(10, 'SBI_ResidencyRoad', 5000);
insert into bank_account values(11, 'SBI_Jantarmantar', 2000);
commit;

insert into bank_customer values ('Avinash', 'Bull_Temple_Road',
'Bangalore');
insert into bank_customer values ('Dinesh', 'Bannergatta_Road', 'Bangalore');
insert into bank_customer values ('Mohan', 'National_College_Road',
'Bangalore');
insert into bank_customer values ('Nikhil', 'Akbar_Road', 'Delhi');
insert into bank_customer values ('Ravi', 'Prithviraj_Road', 'Delhi');
commit;

insert into depositer values('Avinash', 1);
insert into depositer values('Dinesh', 2);
insert into depositer values('Nikhil', 4);
insert into depositer values('Ravi', 5);
insert into depositer values('Avinash', 8);
insert into depositer values('Nikhil', 9);
insert into depositer values('Dinesh', 10);
insert into depositer values('Nikhil', 11);
commit;

insert into loan values(1, 'SBI_Chamrajpet', 1000);
insert into loan values(2, 'SBI_ResidencyRoad', 2000);
insert into loan values(3, 'SBI_ShivajiRoad', 3000);
```

```
insert into loan values(4, 'SBI_ParliamentRoad', 4000);
insert into loan values(5, 'SBI_Jantarmantar', 5000);
commit;
```

```
select * from branch;
```

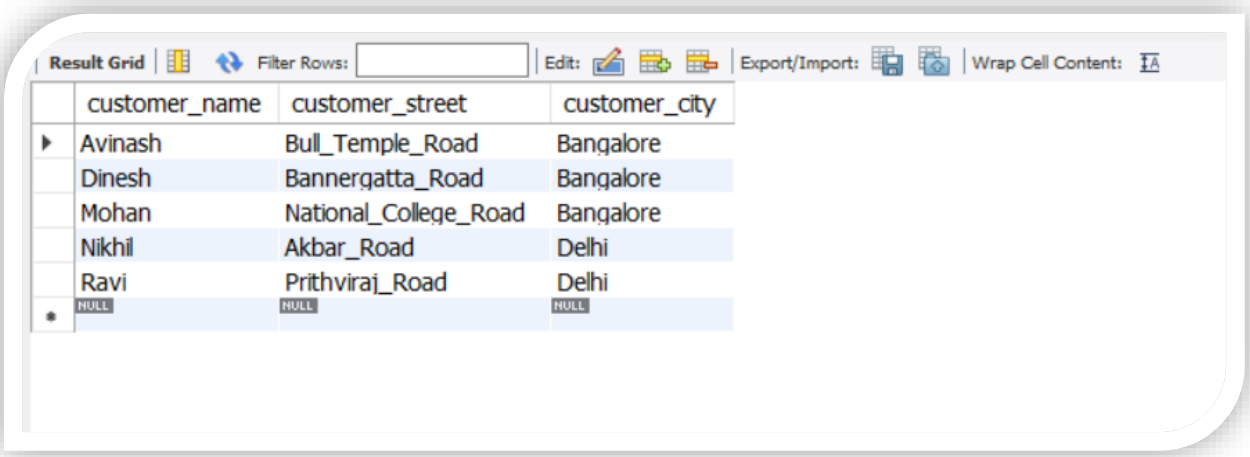
branch_name	branch_city	assets
SBI_Chamrajpet	Bangalore	50000
SBI_Jantarmantar	Delhi	20000
SBI_ParliamentRoad	Delhi	10000
SBI_ResidencyRoad	Bangalore	10000
SBI_ShivajiRoad	Bombay	20000
NULL	NULL	NULL

```
select * from bank_account;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

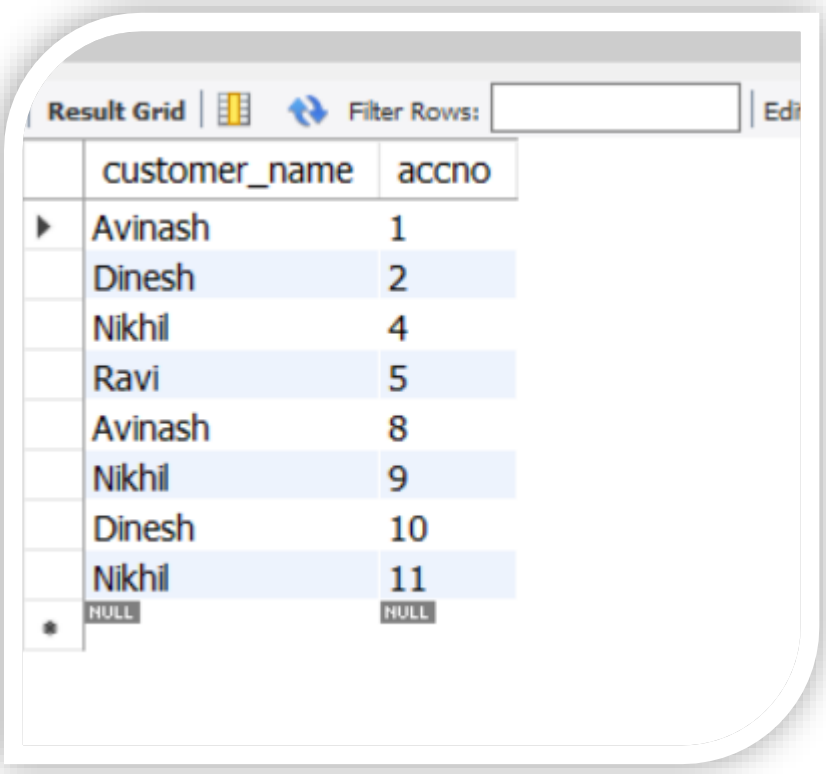
accno	branch_name	balance
1	SBI_Chamrajpet	2000
2	SBI_ResidencyRoad	5000
3	SBI_ShivajiRoad	6000
4	SBI_ParliamentRoad	9000
5	SBI_Jantarmantar	8000
6	SBI_ShivajiRoad	4000
8	SBI_ResidencyRoad	4000
9	SBI_ParliamentRoad	3000
10	SBI_ResidencyRoad	5000
11	SBI_Jantarmantar	2000
NULL	NULL	NULL

```
select * from bank_customer;
```



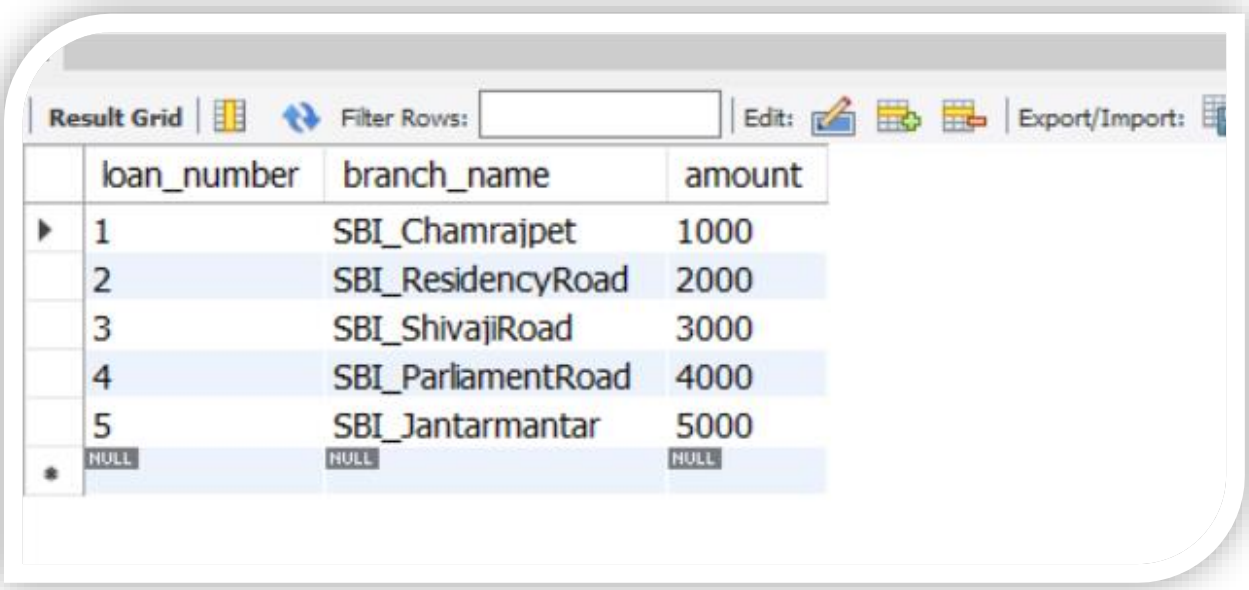
customer_name	customer_street	customer_city
Avinash	Bull_Temple_Road	Bangalore
Dinesh	Bannergatta_Road	Bangalore
Mohan	National_College_Road	Bangalore
Nikhil	Akbar_Road	Delhi
Ravi	Prithviraj_Road	Delhi
NULL	NULL	NULL

select * from depositer;



customer_name	accno
Avinash	1
Dinesh	2
Nikhil	4
Ravi	5
Avinash	8
Nikhil	9
Dinesh	10
Nikhil	11
NULL	NULL

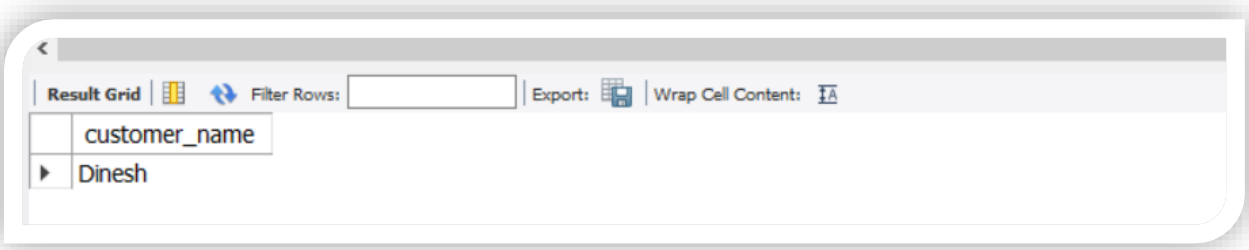
select * from loan;



loan_number	branch_name	amount
1	SBI_Chamrajpet	1000
2	SBI_ResidencyRoad	2000
3	SBI_ShivajiRoad	3000
4	SBI_ParliamentRoad	4000
5	SBI_Jantarmantar	5000
NULL	NULL	NULL

-- Query 3

```
select distinct c.customer_name from bank_customer c,bank_account b where
exists(select d.customer_name,count(d.customer_name) from depositer
d,bank_account ba where ba.accno = d.accno and
c.customer_name = d.customer_name and ba.branch_name =
'SBI_ResidencyRoad' group by d.customer_name having
count(d.customer_name)>=2);
```

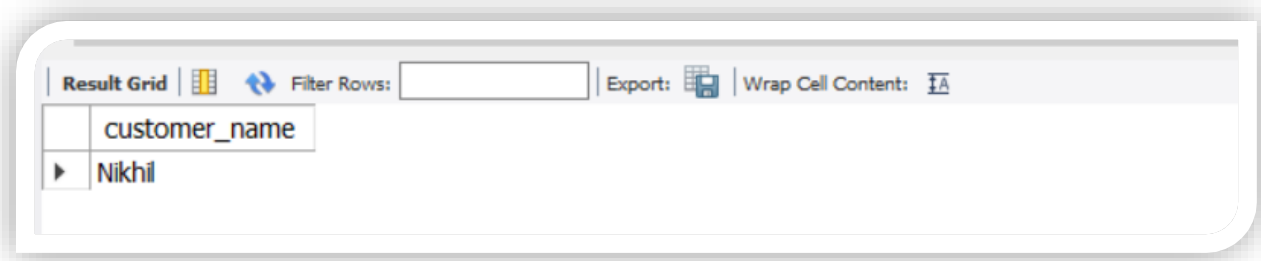


customer_name
Dinesh

-- Query 4

```
select d.customer_name from depositer d,branch b,bank_account a
where b.branch_name=a.branch_name
AND a.accno=d.accno
and branch_city='Delhi'
group by d.customer_name
```

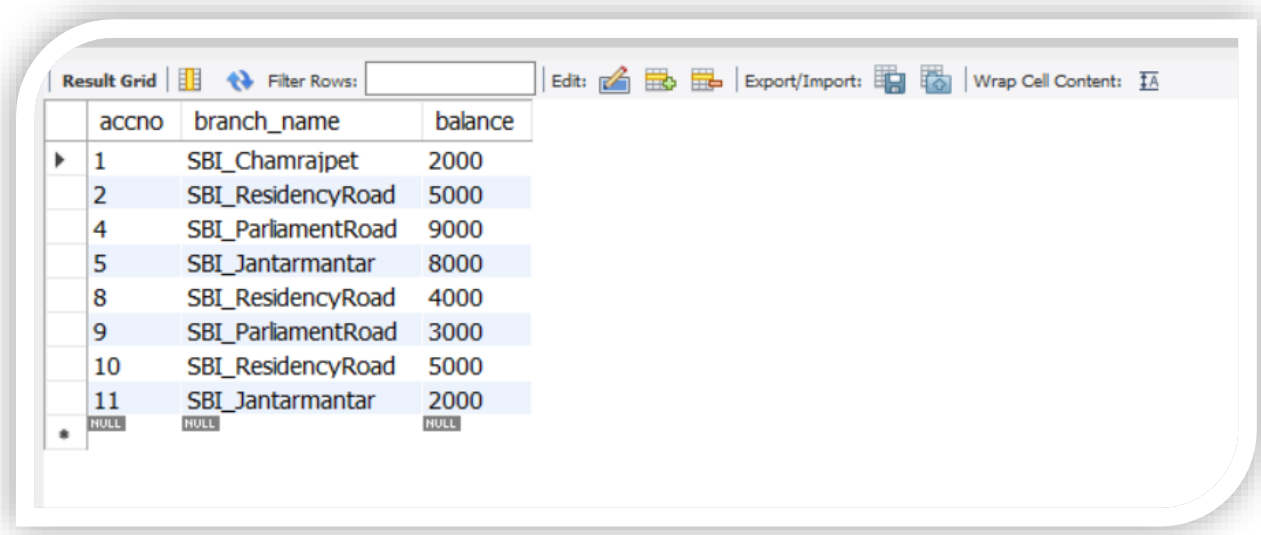
```
HAVING COUNT(distinct b.branch_name)=(
        SELECT COUNT(branch_name)
        FROM branch
        WHERE branch_city='Delhi');
```



customer_name
Nikhil

-- Query 5

```
delete from bank_account where branch_name in (select branch_name from
branch where branch_city = 'Bombay');
select * from bank_account;
```



accno	branch_name	balance
1	SBI_Chamrajpet	2000
2	SBI_ResidencyRoad	5000
4	SBI_ParliamentRoad	9000
5	SBI_Jantarmantar	8000
8	SBI_ResidencyRoad	4000
9	SBI_ParliamentRoad	3000
10	SBI_ResidencyRoad	5000
11	SBI_Jantarmantar	2000
NULL	NULL	NULL

PROGRAM 3: SUPPLIER DATABASE

Consider the following schema:

SUPPLIERS(sid: integer, sname: string, address: string)

PARTS(pid: integer, pname: string, color: string)

CATALOG(sid: integer, pid: integer, cost: real)

The Catalog relation lists the prices charged for parts by Suppliers.

Write the following queries in SQL:

i) Find the pnames of parts for which there is some supplier.

ii) Find the snames of suppliers who supply every part.

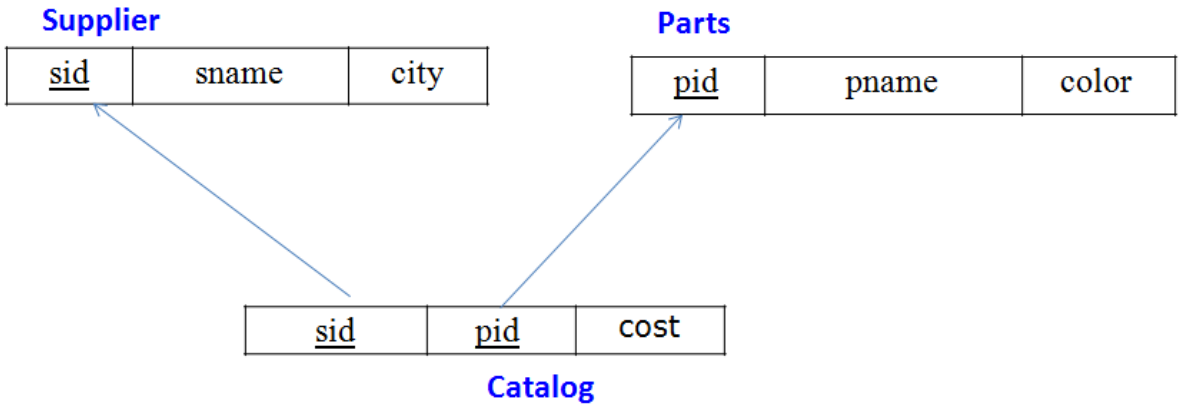
iii) Find the snames of suppliers who supply every red part.

iv) Find the pnames of parts supplied by Acme Widget Suppliers and by no one else.

v) Find the sids of suppliers who charge more for some part than the average cost of that part (averaged over all the suppliers who supply that part).

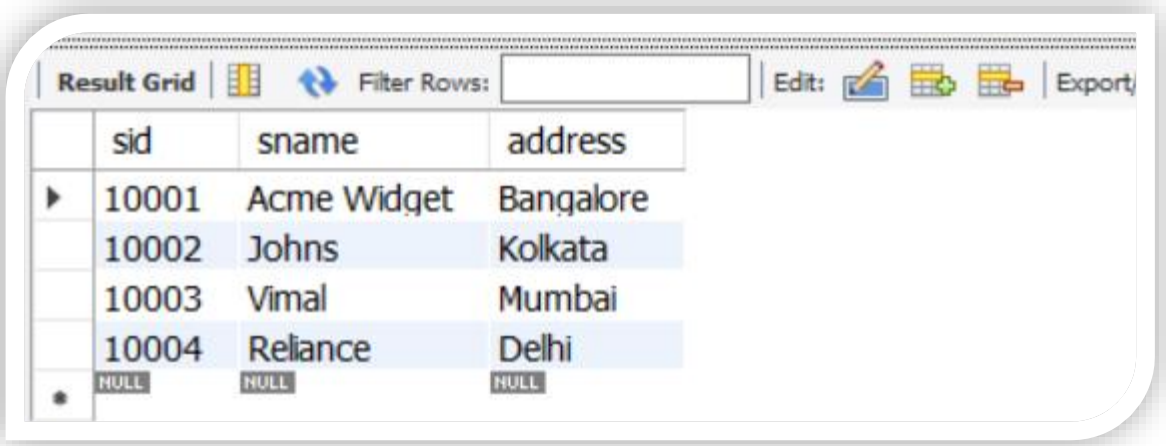
vi) For each part, find the sname of the supplier who charges the most for that part.

Schema Diagram



-- Creating databases and tables

```
Create  database supplier;
use supplier;
create table SUPPLIERS(sid integer,sname varchar(20),address varchar(40),primary key(sid));
INSERT INTO `supplier`.`suppliers` (`sid`, `sname`, `address`) VALUES ('10001', 'Acme Widget', 'Bangalore');
INSERT INTO `supplier`.`suppliers` (`sid`, `sname`, `address`) VALUES ('10002', 'Johns', 'Kolkata');
INSERT INTO `supplier`.`suppliers` (`sid`, `sname`, `address`) VALUES ('10003', 'Vimal', 'Mumbai');
INSERT INTO `supplier`.`suppliers` (`sid`, `sname`, `address`) VALUES ('10004', 'Reliance', 'Delhi');
commit;
select* from SUPPLIERS;
```



create table PARTS(pid integer,pname varchar(20),color varchar(30),primary key(pid));

INSERT INTO `supplier`.`parts` (`pid`, `pname`, `color`) VALUES ('20001', 'Book', 'Red');

INSERT INTO `supplier`.`parts` (`pid`, `pname`, `color`) VALUES ('20002', 'Pen', 'Red');

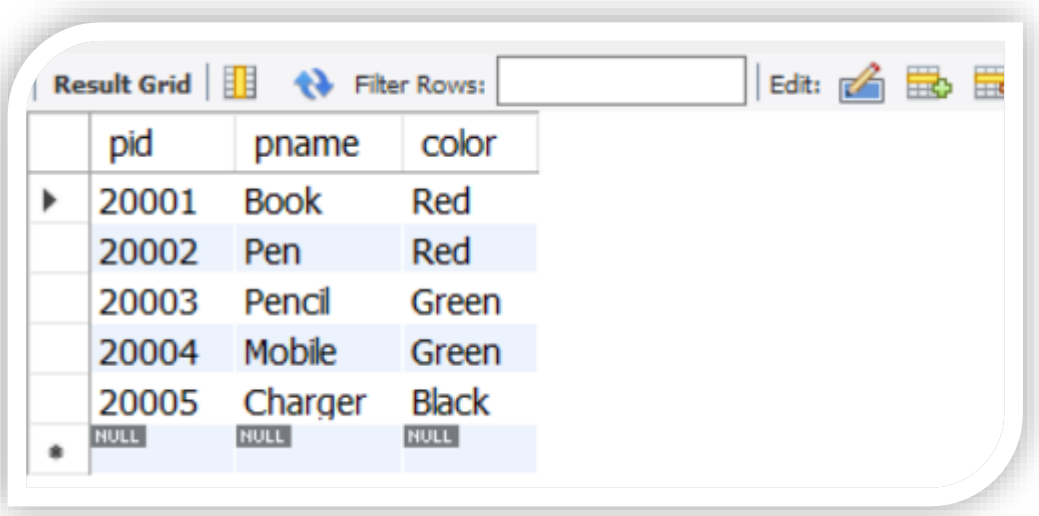
INSERT INTO `supplier`.`parts` (`pid`, `pname`, `color`) VALUES ('20003', 'Pencil', 'Green');

INSERT INTO `supplier`.`parts` (`pid`, `pname`, `color`) VALUES ('20004', 'Mobile', 'Green');

INSERT INTO `supplier`.`parts` (`pid`, `pname`, `color`) VALUES ('20005', 'Charger', 'Black');

commit;

select* from PART;



create table CATALOG(sid integer,pid integer,foreign key(sid) references SUPPLIERS(sid),foreign key(pid) references PARTS(pid),

cost integer,primary key(sid,pid));

INSERT INTO `supplier`.`catalog` (`sid`, `pid`, `cost`) VALUES ('10001', '20001', '10');

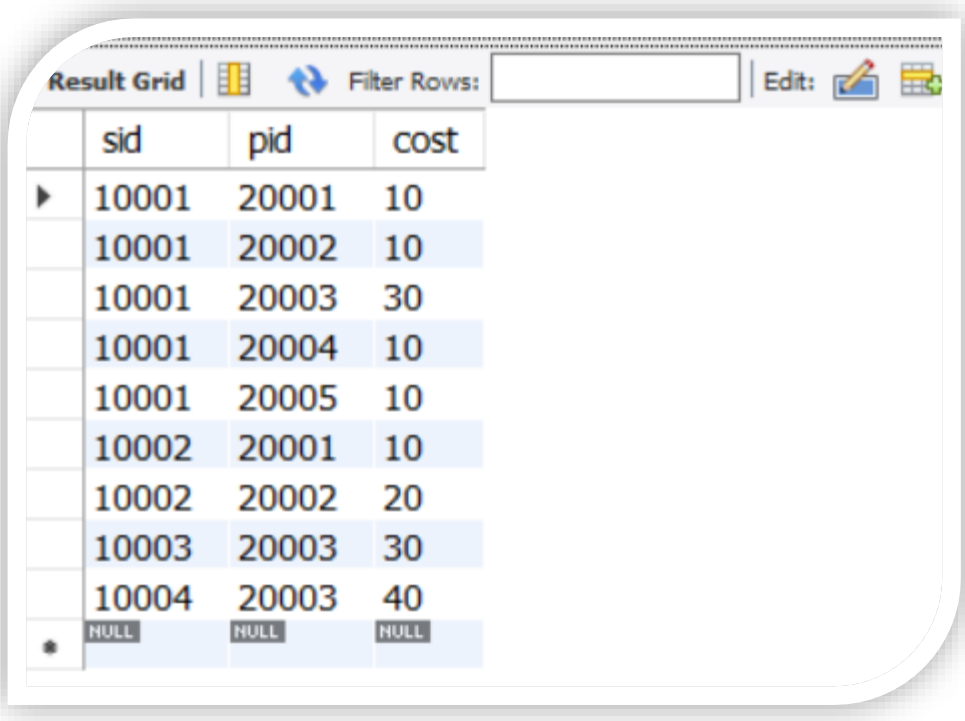
INSERT INTO `supplier`.`catalog` (`sid`, `pid`, `cost`) VALUES ('10001', '20002', '10');

INSERT INTO `supplier`.`catalog` (`sid`, `pid`, `cost`) VALUES ('10001', '20003', '30');

INSERT INTO `supplier`.`catalog` (`sid`, `pid`, `cost`) VALUES ('10001', '20004', '10');

INSERT INTO `supplier`.`catalog` (`sid`, `pid`, `cost`) VALUES ('10001', '20005', '10');

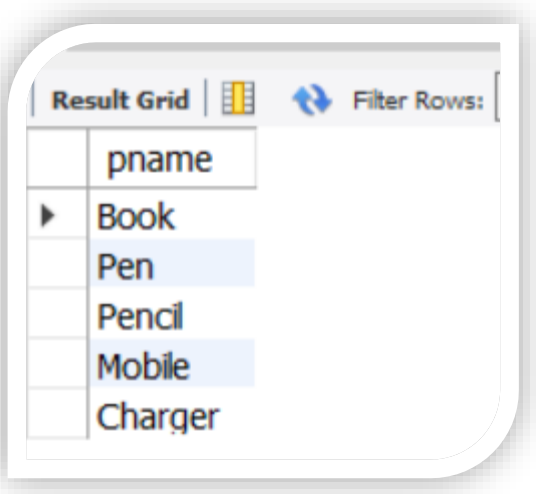
```
INSERT INTO `supplier`.`catalog` (`sid`, `pid`, `cost`) VALUES ('10002', '20001', '10');
INSERT INTO `supplier`.`catalog` (`sid`, `pid`, `cost`) VALUES ('10002', '20002', '20');
INSERT INTO `supplier`.`catalog` (`sid`, `pid`, `cost`) VALUES ('10003', '20003', '30');
INSERT INTO `supplier`.`catalog` (`sid`, `pid`, `cost`) VALUES ('10004', '20003', '40');

commit;
select* from CATALOG;
```



sid	pid	cost
10001	20001	10
10001	20002	10
10001	20003	30
10001	20004	10
10001	20005	10
10002	20001	10
10002	20002	20
10003	20003	30
10004	20003	40
NULL	NULL	NULL

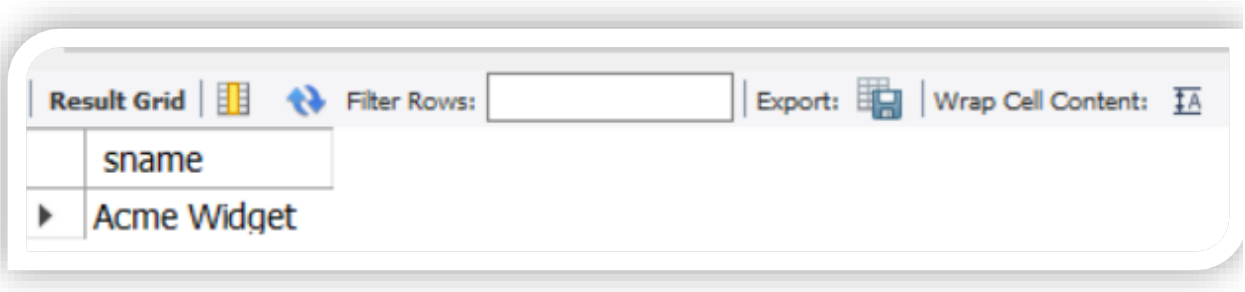
-- Query 1

```
SELECT DISTINCT P.pname
FROM Parts P, Catalog C
WHERE P.pid = C.pid;
```



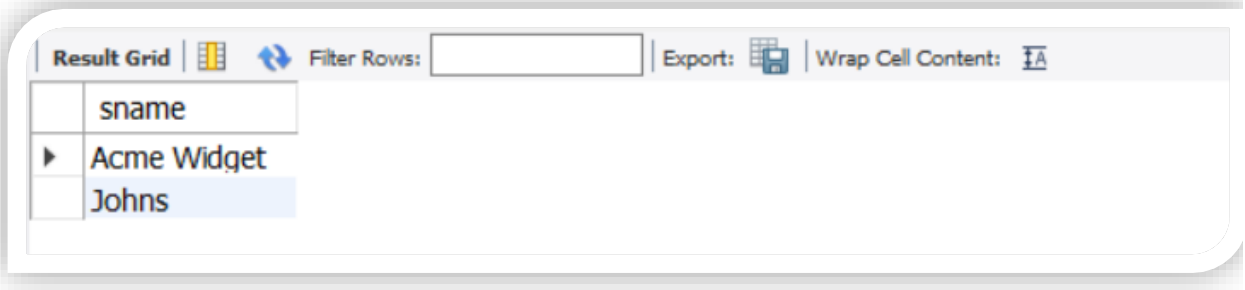
-- Query 2

```
select S.sname from SUPPLIERS S where not exists
(select P.pid from PARTS P where not exists
 (select C.sid from CATALOG C where C.sid = S.sid and C.pid =
P.pid));
```



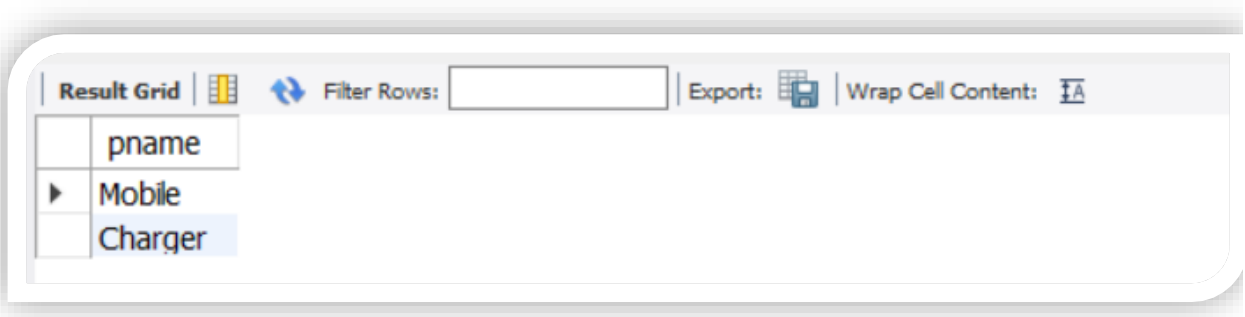
-- Query 3

```
select S.sname from SUPPLIERS S where not exists
 (select P.pid from PARTS P where P.color = 'Red' and
 (not exists (select C.sid from CATALOG C where C.sid = S.sid and
C.pid = P.pid)));
```



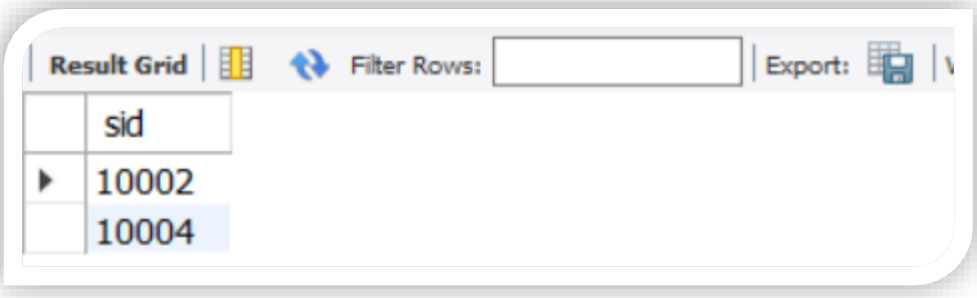
-- Query 4

```
 select P.pname from PARTS P, CATALOG C, SUPPLIERS S
 where P.pid = C.pid and C.sid = S.sid and S.sname = 'Acme Widget'
 and not exists (select * from CATALOG C1, SUPPLIERS S1
 where P.pid = C1.pid and C1.sid = S1.sid and S1.sname <> 'Acme
Widget');
```



-- Query 5

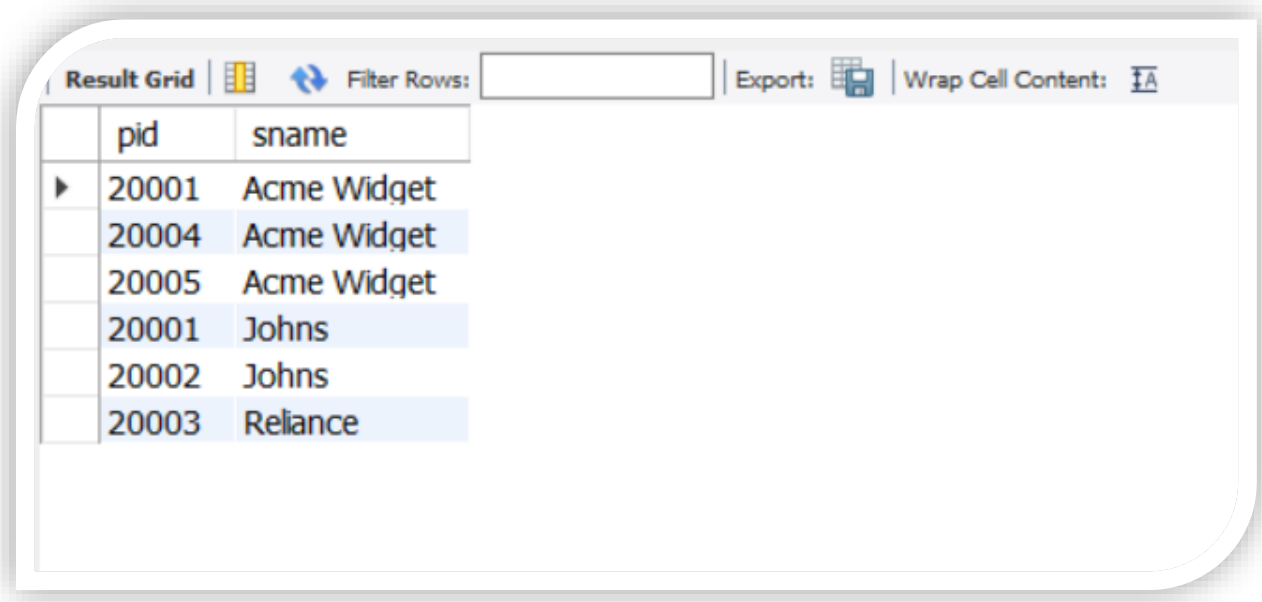
```
SELECT DISTINCT C.sid FROM Catalog C
WHERE C.cost > ( SELECT AVG (C1.cost)
FROM Catalog C1
WHERE C1.pid = C.pid );
```



-- Query 6

```
SELECT P.pid, S.sname
FROM Parts P, Suppliers S, Catalog C
WHERE C.pid = P.pid
```

```
AND C.sid = S.sid
AND C.cost = (SELECT MAX(C1.cost)
FROM Catalog C1
WHERE C1.pid = P.pid);
```



pid	sname
20001	Acme Widget
20004	Acme Widget
20005	Acme Widget
20001	Johns
20002	Johns
20003	Reliance

PROGRAM 4: STUDENT FACULTY DATABASE

Consider the following database for student enrollment for course :

STUDENT(snum: integer, sname: string, major: string, lvl: string, age: integer)

CLASS(cname: string, meets at: time, room: string, fid: integer)

ENROLLED(snum: integer, cname: string)

FACULTY(fid: integer, fname: string, deptid: integer)

The meaning of these relations is straightforward; for example, Enrolled has one record per student-class pair such that the student is enrolled in the class. Level(lvl) is a two character code with 4 different values (example: Junior: JR etc)

Write the following queries in SQL. No duplicates should be printed in any of the answers.

i. Find the names of all Juniors (level = JR) who are enrolled in a class taught by
ii. Find the names of all classes that either meet in room R128 or have five or more Students enrolled.
iii. Find the names of all students who are enrolled in two classes that meet at the same time.
iv. Find the names of faculty members who teach in every room in which some class is taught.
v. Find the names of faculty members for whom the combined enrollment of the courses that they teach is less than five.
vi. Find the names of students who are not enrolled in any class.
vii. For each age value that appears in Students, find the level value that appears most often. For example, if there are more FR level students aged 18 than SR, JR, or SO students aged 18, you should print the pair (18, FR).

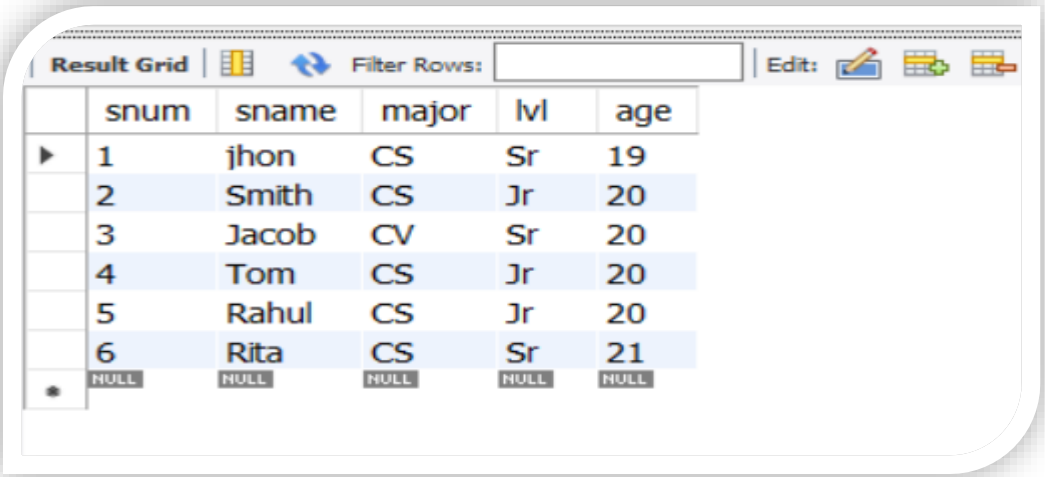
-- Creating database and tables

```
CREATE DATABASE student_faculty;
USE student_faculty;
CREATE TABLE student(
        snum INT,
        sname VARCHAR(10),
        major VARCHAR(2),
        lvl VARCHAR(2),
        age INT, primary key(snum));

CREATE TABLE faculty(
        fid INT,fname VARCHAR(20),
        deptid INT,
    PRIMARY KEY(fid));

CREATE TABLE class(
        cname VARCHAR(20),
        metts_at TIMESTAMP,
        room VARCHAR(10),
    fid INT,
        PRIMARY KEY(cname),
        FOREIGN KEY(fid) REFERENCES faculty(fid));
```

```
CREATE TABLE enrolled(
        snum INT,
        cname VARCHAR(20),
        PRIMARY KEY(snum,cname),
        FOREIGN KEY(snum) REFERENCES student(snum),
        FOREIGN KEY(cname) REFERENCES class(cname));

INSERT INTO STUDENT VALUES(1, 'jhon', 'CS', 'Sr', 19);
INSERT INTO STUDENT VALUES(2, 'Smith', 'CS', 'Jr', 20);
INSERT INTO STUDENT VALUES(3 , 'Jacob', 'CV', 'Sr', 20);
INSERT INTO STUDENT VALUES(4, 'Tom ', 'CS', 'Jr', 20);
INSERT INTO STUDENT VALUES(5, 'Rahul', 'CS', 'Jr', 20);
INSERT INTO STUDENT VALUES(6, 'Rita', 'CS', 'Sr', 21);

INSERT INTO FACULTY VALUES(11, 'Harish', 1000);
INSERT INTO FACULTY VALUES(12, 'MV', 1000);
INSERT INTO FACULTY VALUES(13 , 'Mira', 1001);
INSERT INTO FACULTY VALUES(14, 'Shiva', 1002);
INSERT INTO FACULTY VALUES(15, 'Nupur', 1000);

insert into class values('class1', '12/11/15 10:15:16', 'R1', 14);
insert into class values('class10', '12/11/15 10:15:16', 'R128', 14);
insert into class values('class2', '12/11/15 10:15:20', 'R2', 12);
insert into class values('class3', '12/11/15 10:15:25', 'R3', 11);
```

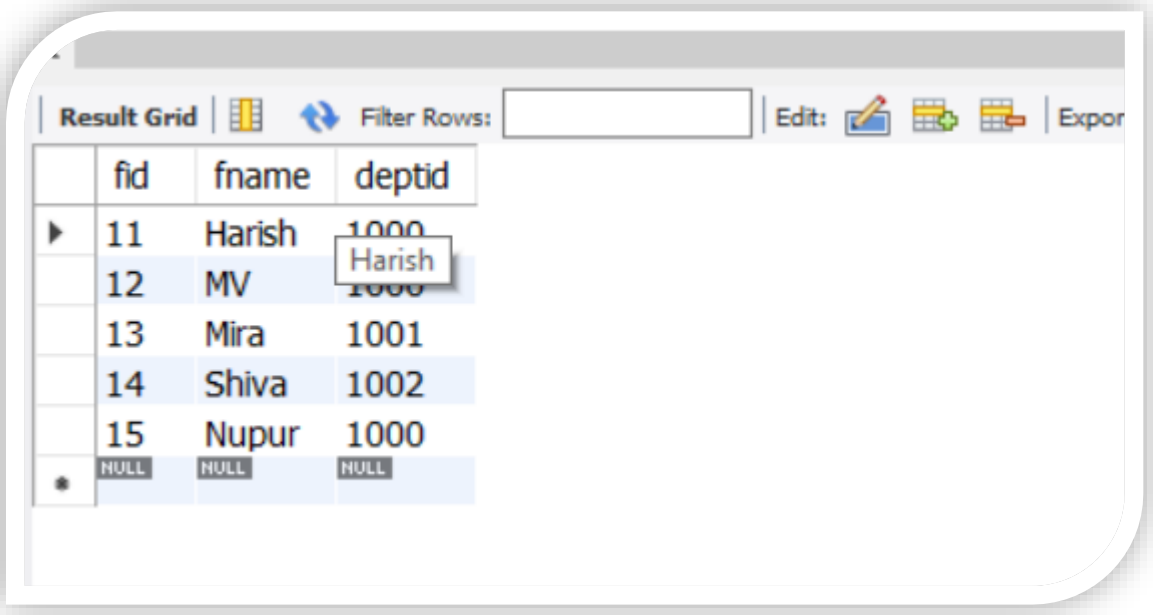
```
insert into class values('class4', '12/11/15 20:15:20', 'R4', 14);
insert into class values('class5', '12/11/15 20:15:20', 'R3', 15);
insert into class values('class6', '12/11/15 13:20:20', 'R2', 14);
insert into class values('class7', '12/11/15 10:10:10', 'R3', 14);

insert into enrolled values(1, 'class1');
insert into enrolled values(2, 'class1');
insert into enrolled values(3, 'class3');
insert into enrolled values(4, 'class3');
insert into enrolled values(5, 'class4');
insert into enrolled values(1, 'class5');
insert into enrolled values(2, 'class5');
insert into enrolled values(3, 'class5');
insert into enrolled values(4, 'class5');
insert into enrolled values(5, 'class5');
```

Student table

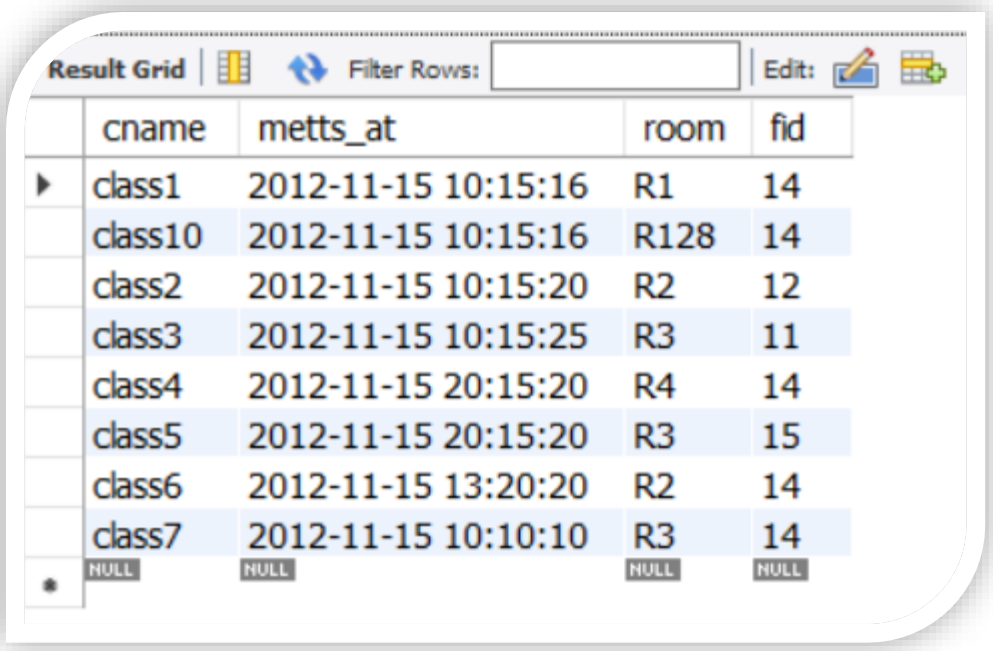


snum	sname	major	lvl	age
1	jhon	CS	Sr	19
2	Smith	CS	Jr	20
3	Jacob	CV	Sr	20
4	Tom	CS	Jr	20
5	Rahul	CS	Jr	20
6	Rita	CS	Sr	21
NULL	NULL	NULL	NULL	NULL

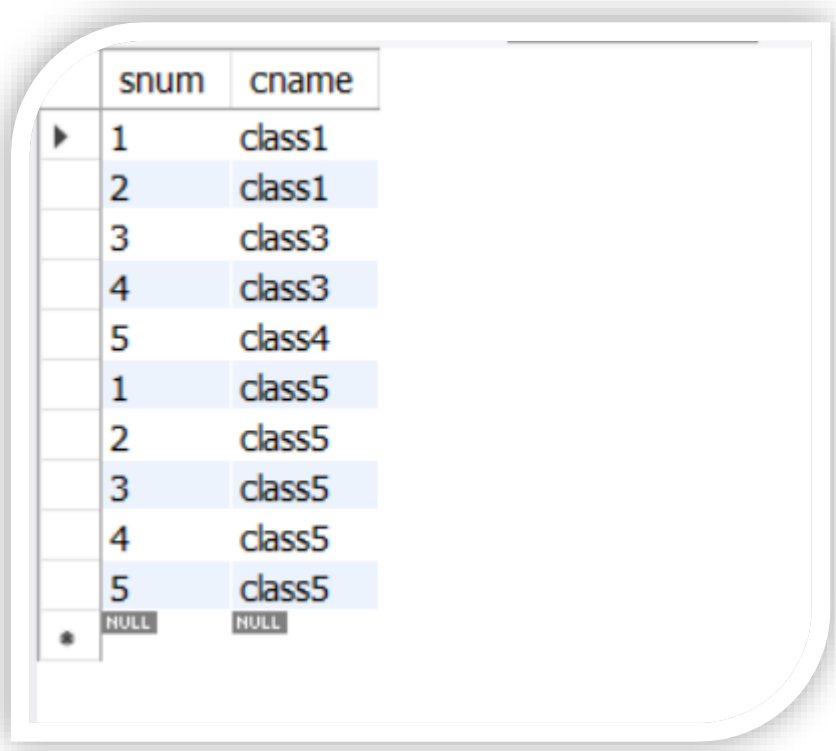
Faculty table



Class table



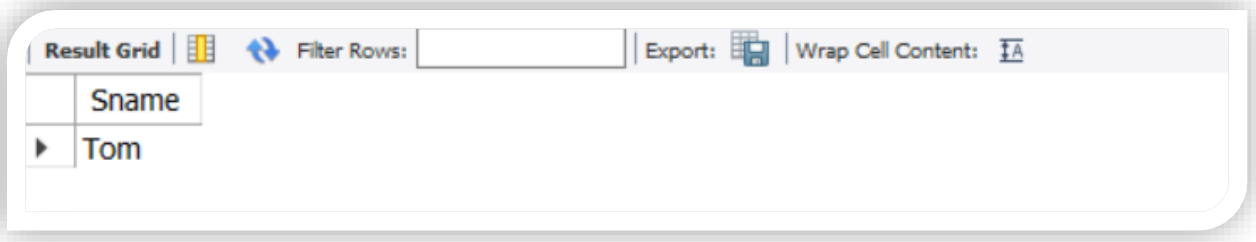
Enrolled table



snum	cname
1	class1
2	class1
3	class3
4	class3
5	class4
1	class5
2	class5
3	class5
4	class5
5	class5
NULL	NULL

-- Query 1

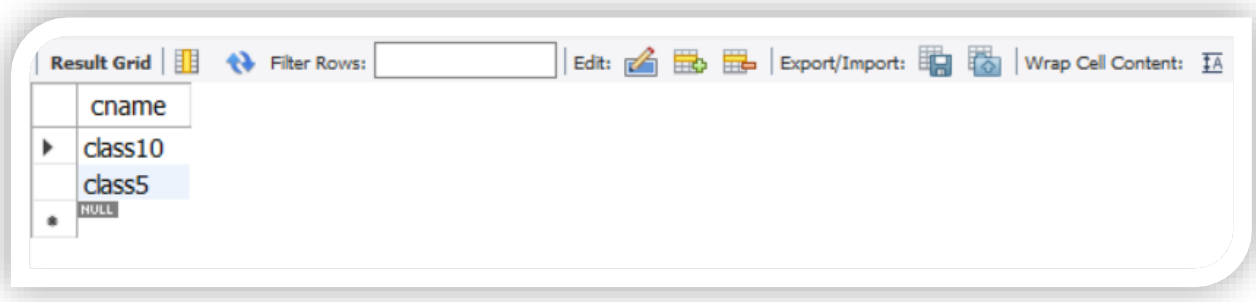
```
SELECT DISTINCT S.Sname
FROM Student S, Class C, Enrolled E, Faculty F
WHERE S.snum = E.snum AND E.cname = C.cname AND C.fid = F.fid
AND
F.fname = 'Harish' AND S.lvl = 'Jr';
```



Sname
Tom

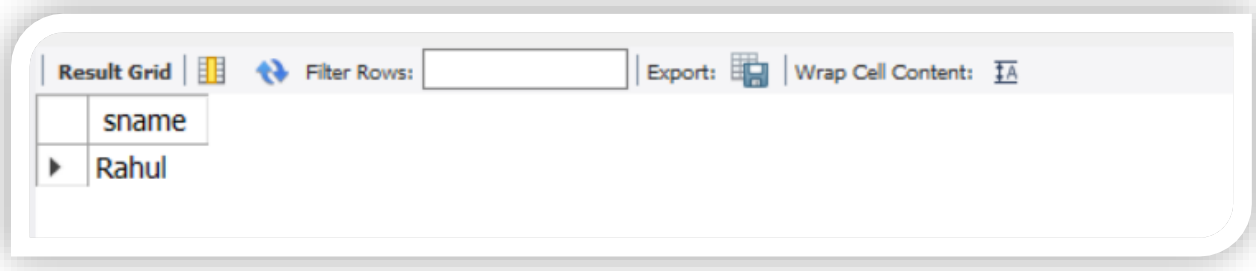
-- Query 2

```
SELECT DISTINCT cname
        FROM class
        WHERE room='R128'
        OR
        cname IN (SELECT e.cname FROM enrolled e GROUP BY e.cname
HAVING COUNT(*)>=5);
```



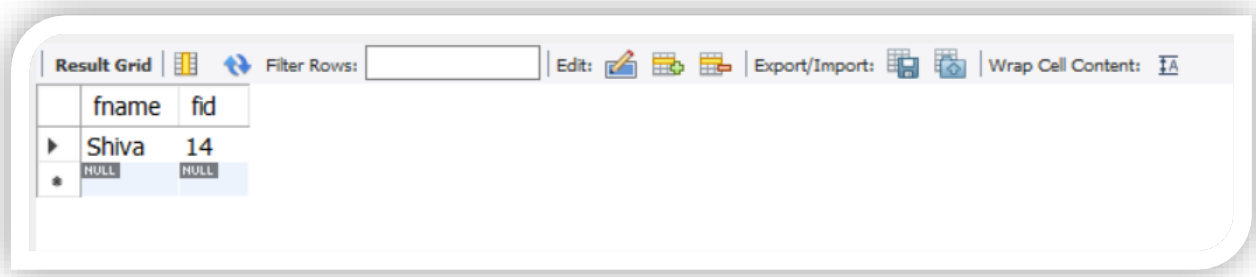
-- Query 3

```
SELECT DISTINCT S.sname
FROM Student S
WHERE S.snum IN (SELECT E1.snum
                FROM Enrolled E1, Enrolled E2, Class C1, Class C2
                WHERE E1.snum = E2.snum AND E1.cname <> E2.cname
                AND E1.cname = C1.cname
                AND E2.cname = C2.cname AND C1.metts_at =
C2.metts_at);
```



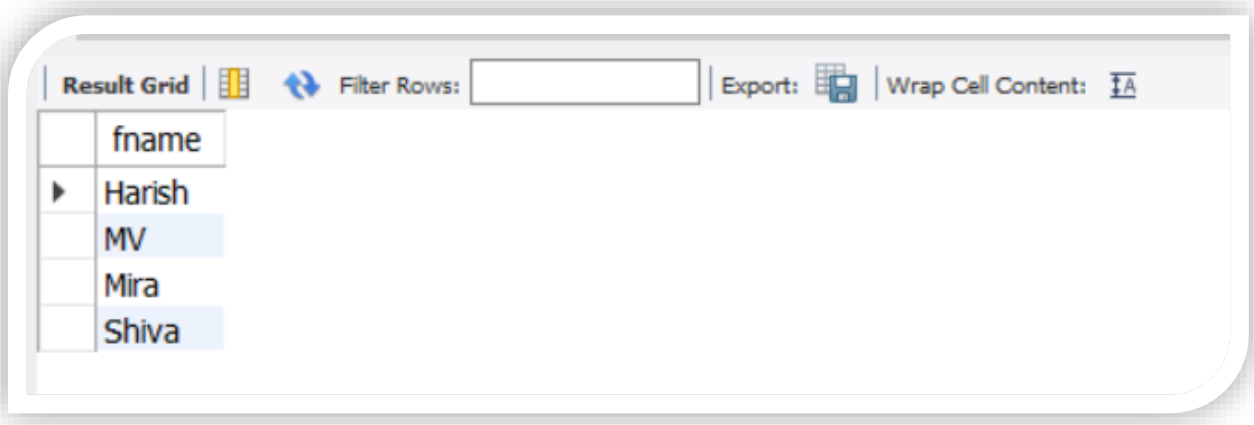
-- Query 4

```
SELECT f.fname,f.fid
                FROM faculty f
          WHERE f.fid in ( SELECT fid FROM class
                GROUP BY fid HAVING COUNT(*)=(SELECT
COUNT(DISTINCT room) FROM class) );
```



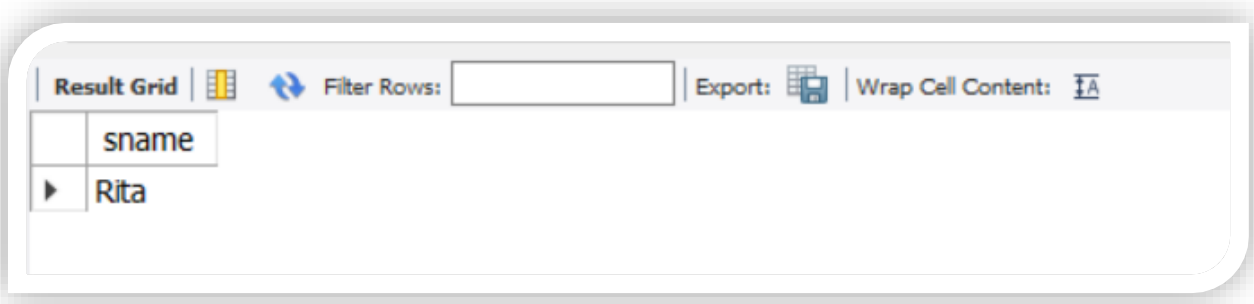
-- Query 5

```
SELECT DISTINCT F.fname
FROM Faculty F
WHERE 5 > (SELECT COUNT(E.snum)
FROM Class C, Enrolled E
WHERE C.cname = E.cname
AND C.fid = F.fid);
```



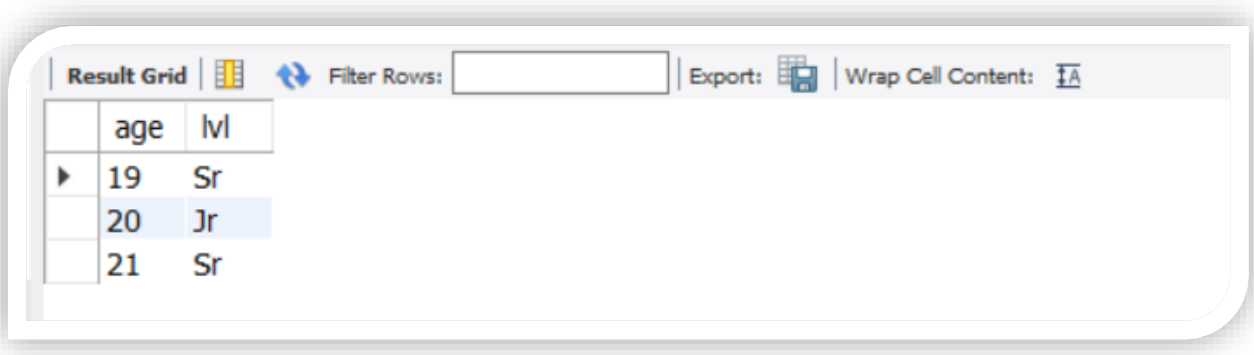
-- Query 6

```
SELECT DISTINCT S.sname
FROM Student S
WHERE S.snum NOT IN (SELECT E.snum
FROM Enrolled E );
```



-- Query 7

```
SELECT S.age, S.lvl
FROM STUDENT S
GROUP BY S.age, S.lvl
HAVING S.lvl IN(SELECT S1.lvl
        FROM STUDENT S1
        WHERE S1.age=S.age
        GROUP BY S1.age, S1.lvl
        HAVING COUNT(*)  >= ALL (SELECT COUNT(*)
                FROM STUDENT S2
                WHERE S1.age=S2.age
                GROUP BY S2.lvl, S2.age))
ORDER BY S.age;
```



PROGRAM 5: AIRLINE FLIGHT DATABASE

Consider the following database that keeps track of airline flight information:

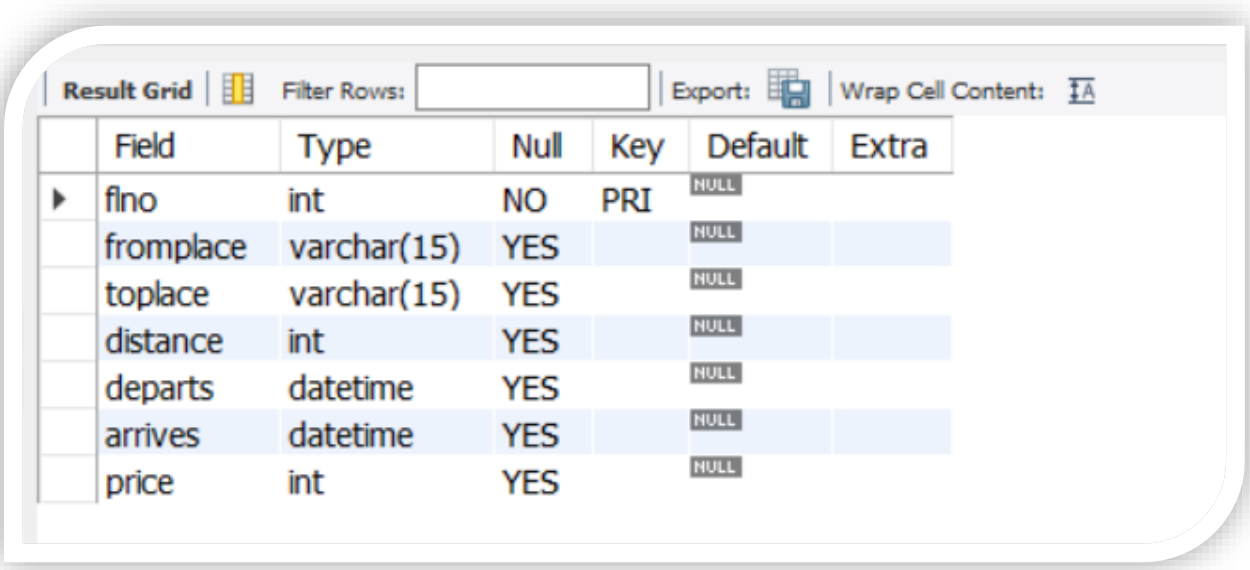
FLIGHTS(flno: integer, from: string, to: string, distance: integer, departs: time, arrives: time, price: integer)
AIRCRAFT(aid: integer, aname: string, cruisingrange: integer)
CERTIFIED(eid: integer, aid: integer)
EMPLOYEES(eid: integer, ename: string, salary: integer)

Note that the Employees relation describes pilots and other kinds of employees as well; Every pilot is certified for some aircraft, and only pilots are certified to fly.

Write each of the following queries in SQL.

i. Find the names of aircraft such that all pilots certified to operate them have salaries more than Rs.80,000.
ii. For each pilot who is certified for more than three aircrafts, find the eid and the maximum cruisingrange of the aircraft for which she or he is certified.
iii. Find the names of pilots whose salary is less than the price of the cheapest route from Bengaluru to Frankfurt.
iv. For all aircraft with cruisingrange over 1000 Kms, find the name of the aircraft and the average salary of all pilots certified for this aircraft.
v. Find the names of pilots certified for some Boeing aircraft.
vi. Find the aids of all aircraft that can be used on routes from Bengaluru to New Delhi.
vii. A customer wants to travel from Madison to New York with no more than two changes of flight. List the choice of departure times from Madison if the customer wants to arrive in New York by 6 p.m.

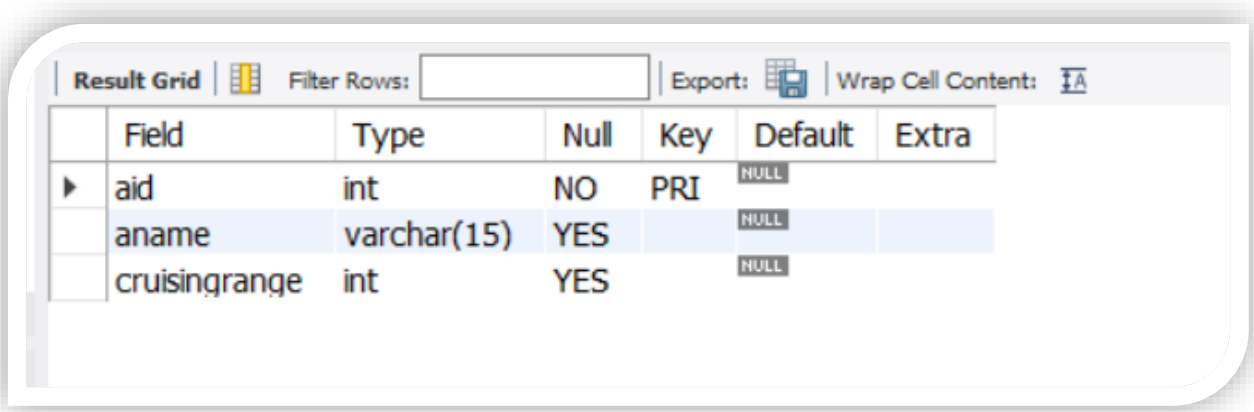
```
--  Creating database and tables
create database flightdb;
use flightdb;

create table flights(
        flno int,
    fromplace varchar(15),
    toplace varchar(15),
    distance int,
    departs datetime,
    arrives datetime,
    price int,
    primary key (flno)
);
desc flights;
```



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

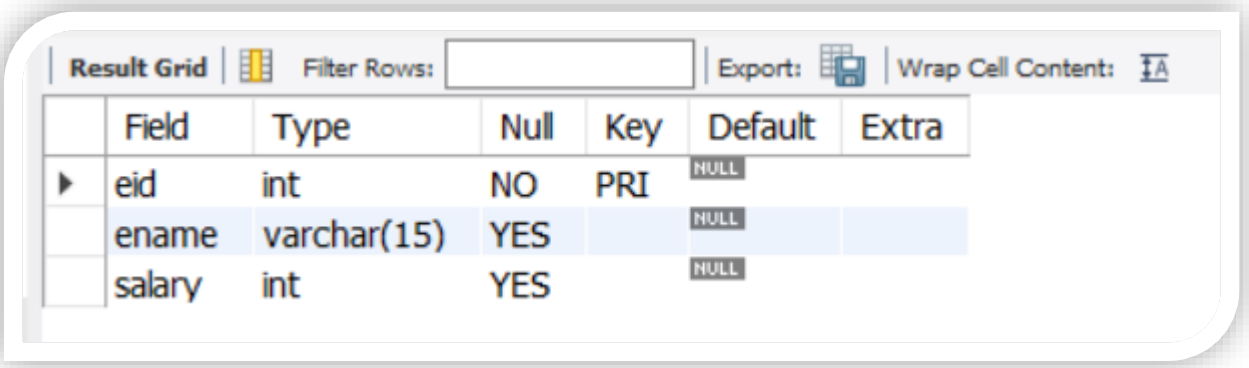
Field	Type	Null	Key	Default	Extra
flno	int	NO	PRI	NULL	
fromplace	varchar(15)	YES		NULL	
toplace	varchar(15)	YES		NULL	
distance	int	YES		NULL	
departs	datetime	YES		NULL	
arrives	datetime	YES		NULL	
price	int	YES		NULL	

```
create table aircraft(
        aid int,
    aname varchar(15),
    cruisingrange int,
    primary key (aid)
);
desc aircraft;
```



Field	Type	Null	Key	Default	Extra
aid	int	NO	PRI	NULL	
aname	varchar(15)	YES		NULL	
cruisingrange	int	YES		NULL	

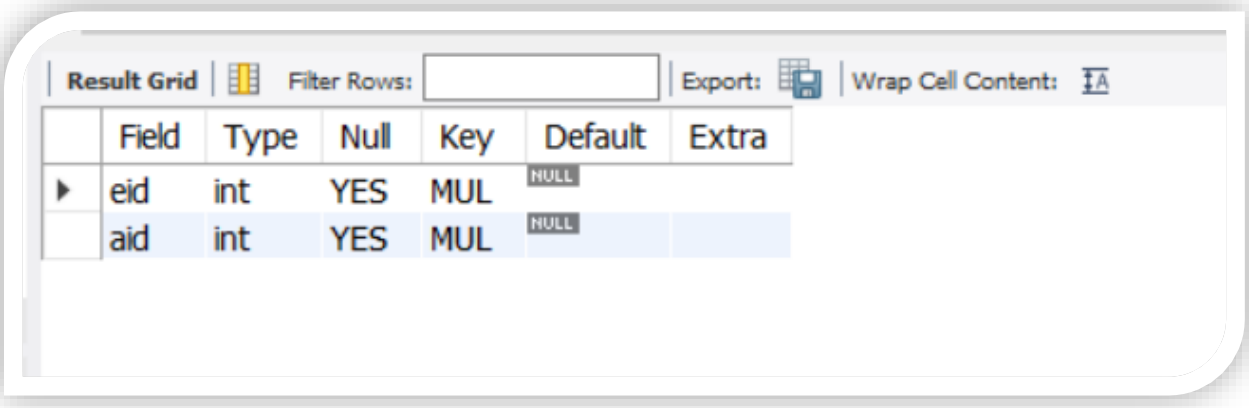
```
create table employees (
        eid int,
    ename varchar(15),
    salary int,
    primary key (eid)
);
desc employees;
```



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
eid	int	NO	PRI	NULL	
ename	varchar(15)	YES		NULL	
salary	int	YES		NULL	

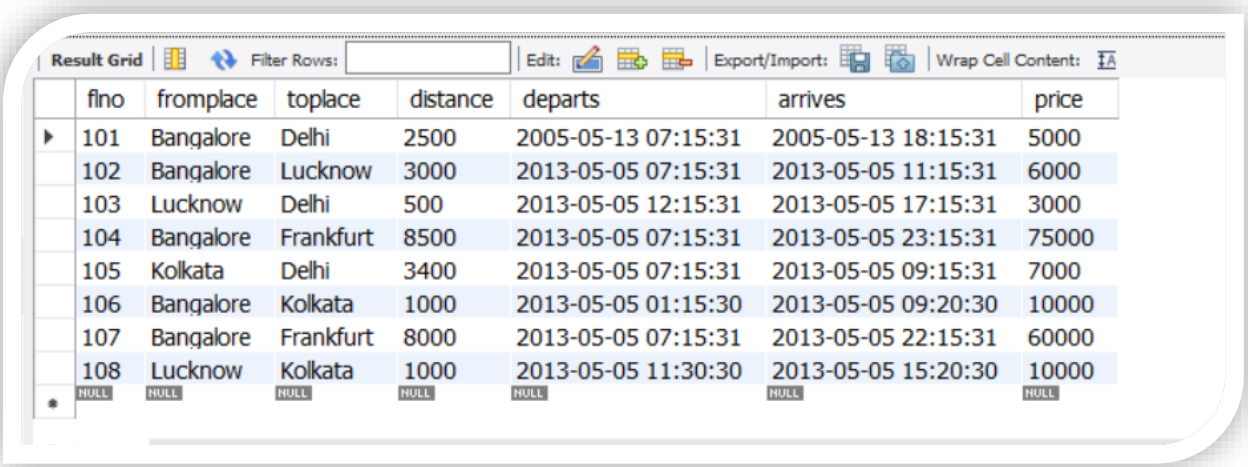
```
create table certified (
        eid int,
    aid int,
    foreign key (eid) references employees(eid),
    foreign key (aid) references aircraft(aid)
);
desc certified;
```



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
eid	int	YES	MUL	NULL	
aid	int	YES	MUL	NULL	

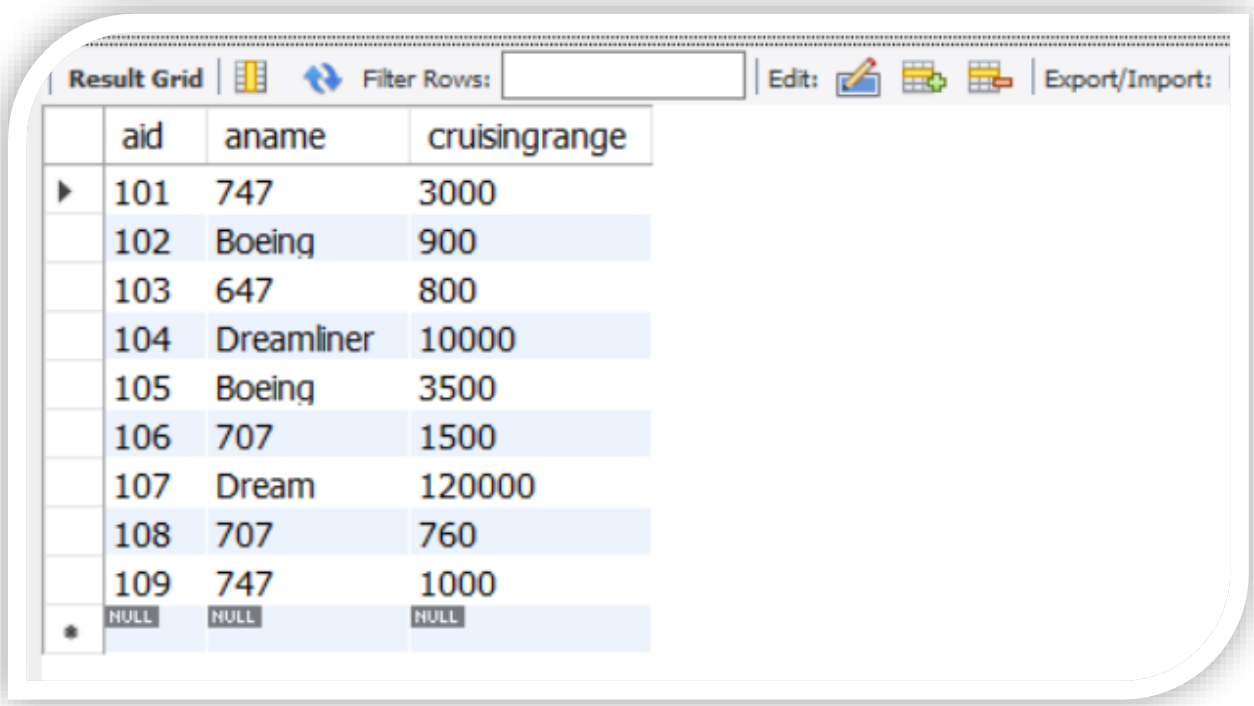
```
insert into flights values(101, 'Bangalore', 'Delhi', 2500, '2005-05-13 07:15:31', '2005-05-13 18:15:31', 5000);
insert into flights values(102, 'Bangalore', 'Lucknow', 3000, '2013-05-05 07:15:31', '2013-05-05 11:15:31', 6000);
insert into flights values(103, 'Lucknow', 'Delhi', 500, '2013-05-05 12:15:31', '2013-05-05 17:15:31', 3000);
insert into flights values(107, 'Bangalore', 'Frankfurt', 8000, '2013-05-05 07:15:31', '2013-05-05 22:15:31', 60000);
insert into flights values(104, 'Bangalore', 'Frankfurt', 8500, '2013-05-05 07:15:31', '2013-05-05 23:15:31', 75000);
insert into flights values(105, 'Kolkata', 'Delhi', 3400, '2013-05-05 07:15:31', '2013-05-05 09:15:31', 7000);
insert into flights values(106, 'Bangalore', 'Kolkata', 1000, '2013-05-05 01:15:30', '2013-05-05 09:20:30', 10000);
insert into flights values(108, 'Lucknow', 'Kolkata', 1000, '2013-05-05 11:30:30', '2013-05-05 15:20:30', 10000);

commit;

select * from flights;
```



Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

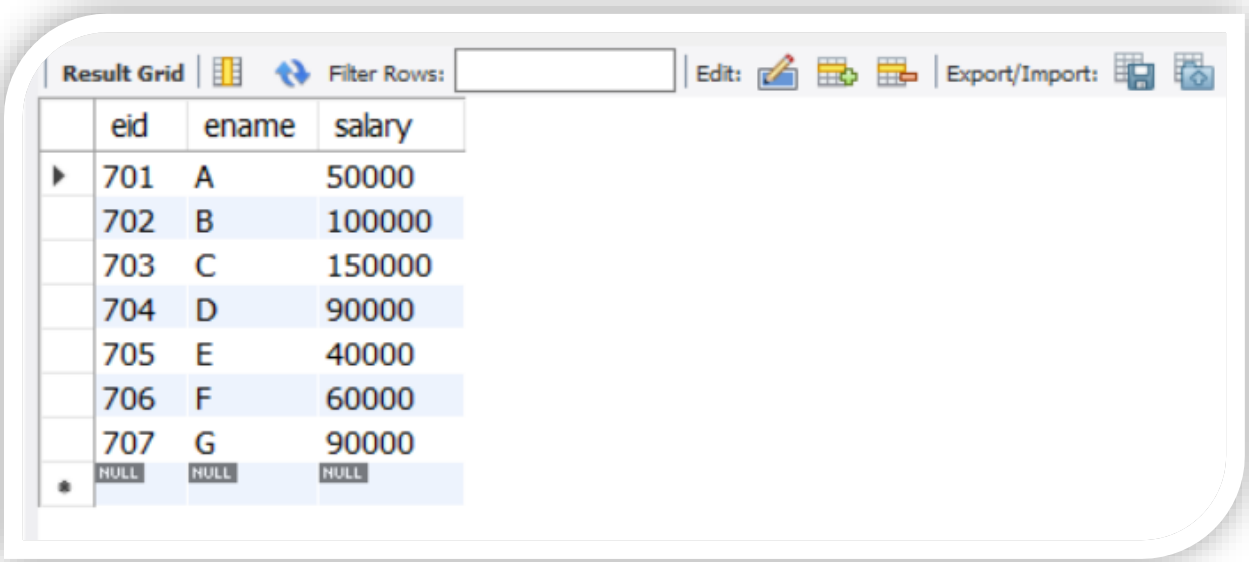
flno	fromplace	toplace	distance	departs	arrives	price
101	Bangalore	Delhi	2500	2005-05-13 07:15:31	2005-05-13 18:15:31	5000
102	Bangalore	Lucknow	3000	2013-05-05 07:15:31	2013-05-05 11:15:31	6000
103	Lucknow	Delhi	500	2013-05-05 12:15:31	2013-05-05 17:15:31	3000
104	Bangalore	Frankfurt	8500	2013-05-05 07:15:31	2013-05-05 23:15:31	75000
105	Kolkata	Delhi	3400	2013-05-05 07:15:31	2013-05-05 09:15:31	7000
106	Bangalore	Kolkata	1000	2013-05-05 01:15:30	2013-05-05 09:20:30	10000
107	Bangalore	Frankfurt	8000	2013-05-05 07:15:31	2013-05-05 22:15:31	60000
108	Lucknow	Kolkata	1000	2013-05-05 11:30:30	2013-05-05 15:20:30	10000
NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
insert into aircraft values(101, '747', 3000);
insert into aircraft values(102, 'Boeing', 900);
insert into aircraft values(103, '647', 800);
insert into aircraft values(104, 'Dreamliner', 10000);
insert into aircraft values(105, 'Boeing', 3500);
insert into aircraft values(106, '707', 1500);
insert into aircraft values(107, 'Dream', 120000);
insert into aircraft values(108, '707', 760);
insert into aircraft values(109, '747', 1000);
commit;

select * from aircraft;
```



aid	aname	cruisingrange
101	747	3000
102	Boeing	900
103	647	800
104	Dreamliner	10000
105	Boeing	3500
106	707	1500
107	Dream	120000
108	707	760
109	747	1000
NULL	NULL	NULL

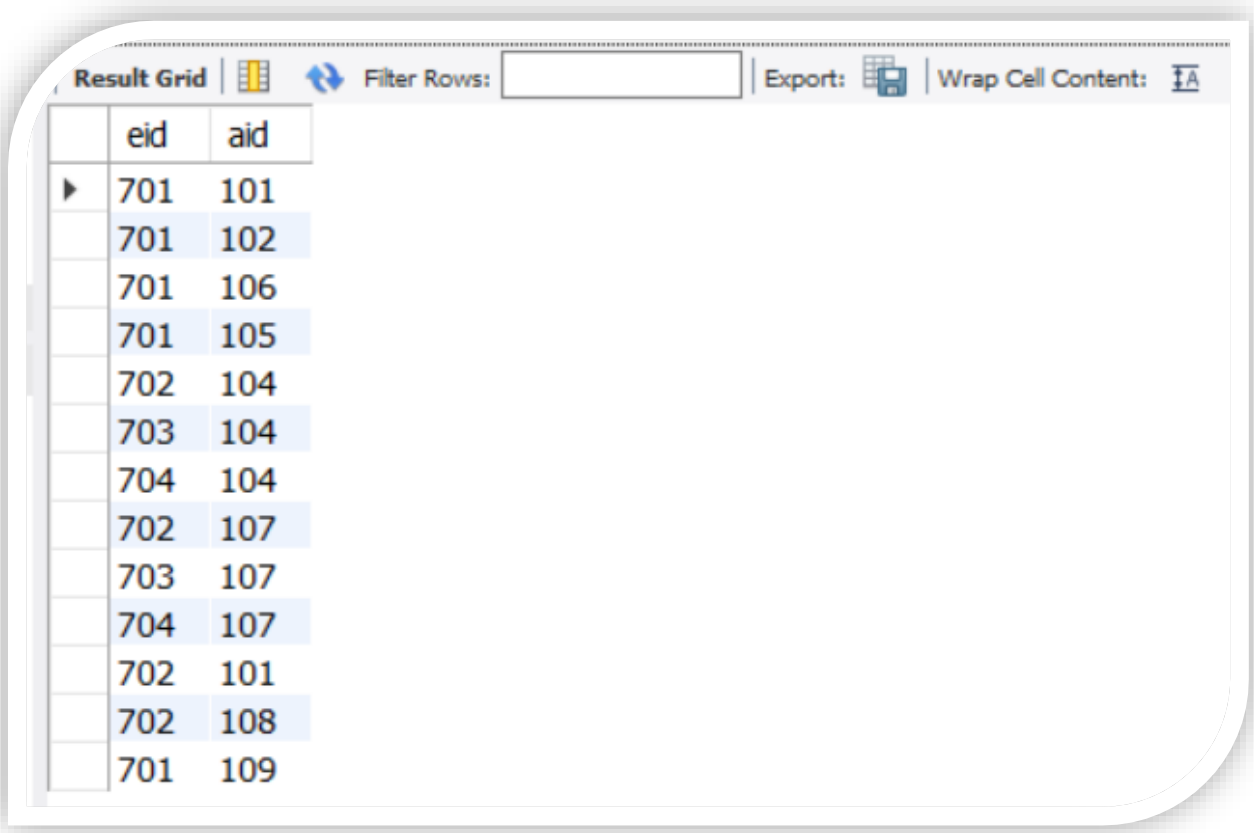
```
insert into employees values(701, 'A', 50000);
insert into employees values(702, 'B', 100000);
insert into employees values(703, 'C', 150000);
insert into employees values(704, 'D', 90000);
insert into employees values(705, 'E', 40000);
insert into employees values(706, 'F', 60000);
insert into employees values(707, 'G', 90000);
commit;

select * from employees;
```



eid	ename	salary
701	A	50000
702	B	100000
703	C	150000
704	D	90000
705	E	40000
706	F	60000
707	G	90000
NULL	NULL	NULL

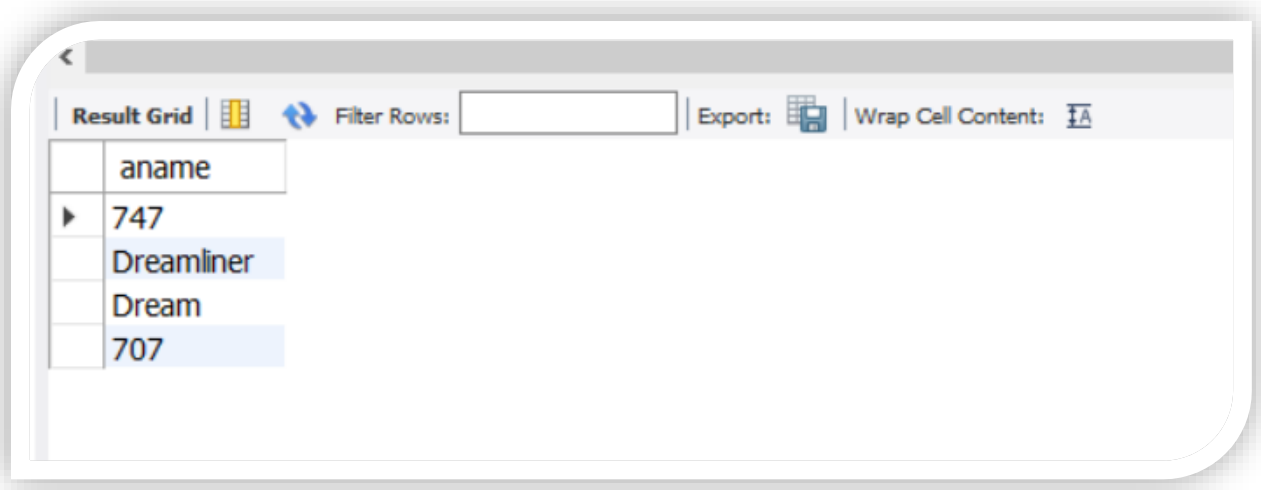
```
insert into certified values(701, 101);
insert into certified values(701, 102);
insert into certified values(701, 106);
insert into certified values(701, 105);

insert into certified values(702, 104);
insert into certified values(703, 104);
insert into certified values(704, 104);

insert into certified values(702, 107);
insert into certified values(703, 107);
insert into certified values(704, 107);

insert into certified values(702, 101);
```

```
insert into certified values(702, 108);
insert into certified values(701, 109);
commit;
select * from certified;
```

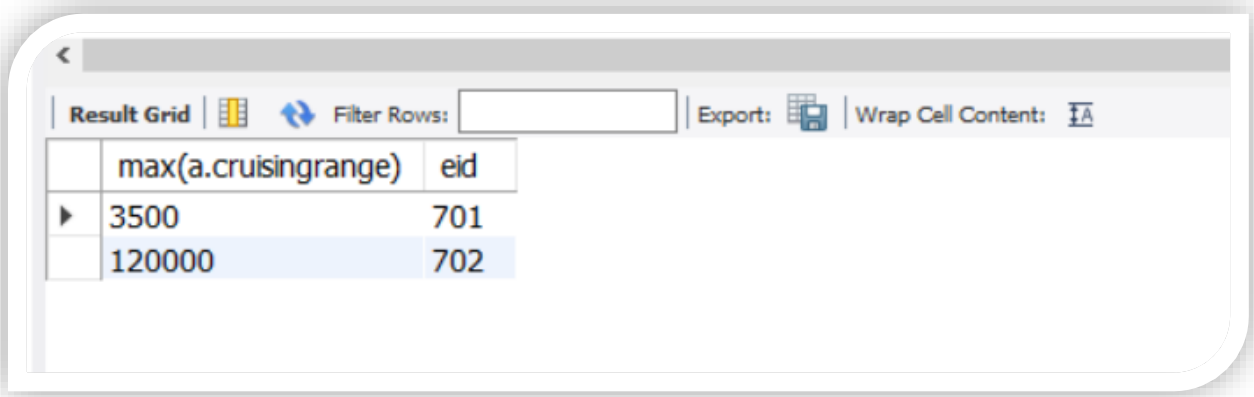


eid	aid
701	101
701	102
701	106
701	105
702	104
703	104
704	104
702	107
703	107
704	107
702	101
702	108
701	109

```
-- Query 1
select distinct a.aname from aircraft a where a.aid in (
        select c.aid from certified c, employees e where
    c.eid = e.eid and not exists(
                select * from employees e1 where e1.eid=e.eid and
e1.salary<80000
    )
);
```



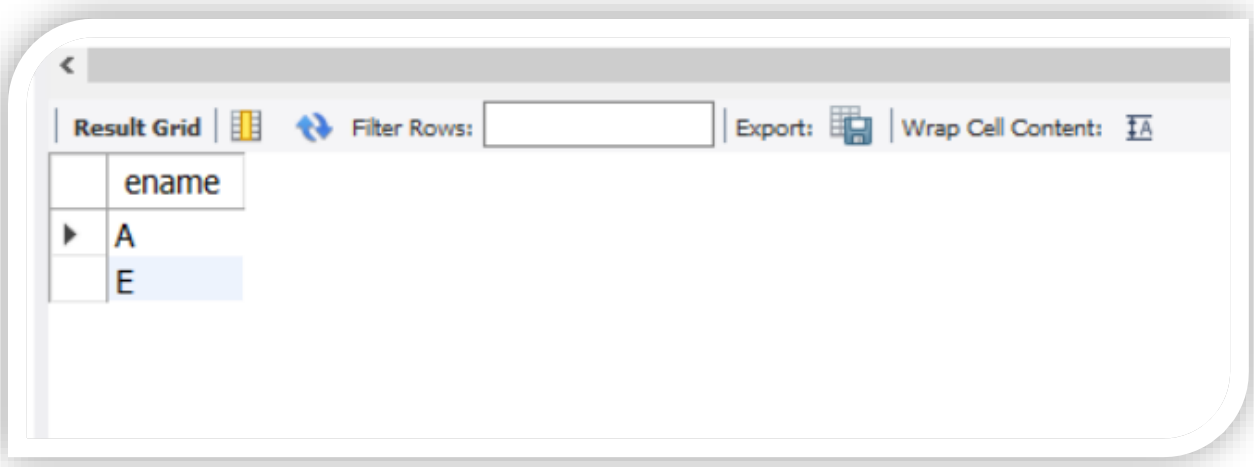

```
-- Query 2
select max(a.cruisingrange), c.eid from certified c, aircraft a where c.aid =
a.aid group by c.eid having count(c.eid)>3;
```



-- Query 3

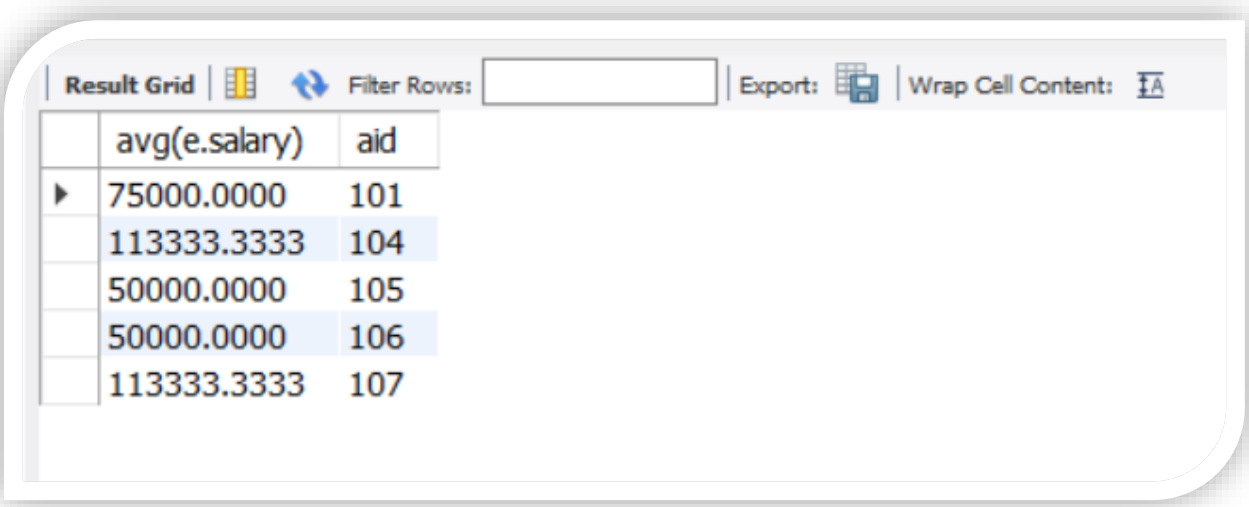
select ename from employees where salary <(

select min(price) from flights where fromplace='Bangalore' and toplace='Frankfurt');



-- Query 4

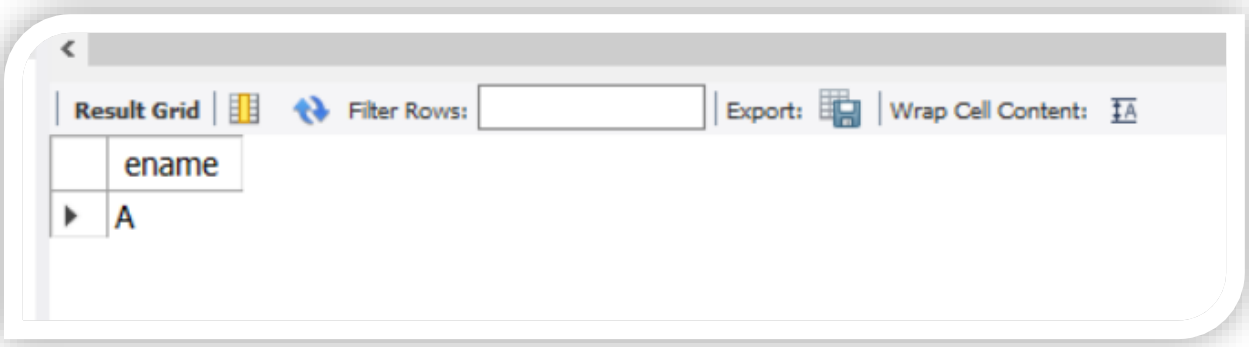
```
select avg(e.salary), c.aid from certified c, employees e where c.aid in(
select aid from aircraft where cruisingrange>1000) and e.eid = c.eid group by
c.aid;
```



avg(e.salary)	aid
75000.0000	101
113333.3333	104
50000.0000	105
50000.0000	106
113333.3333	107

-- Query 5

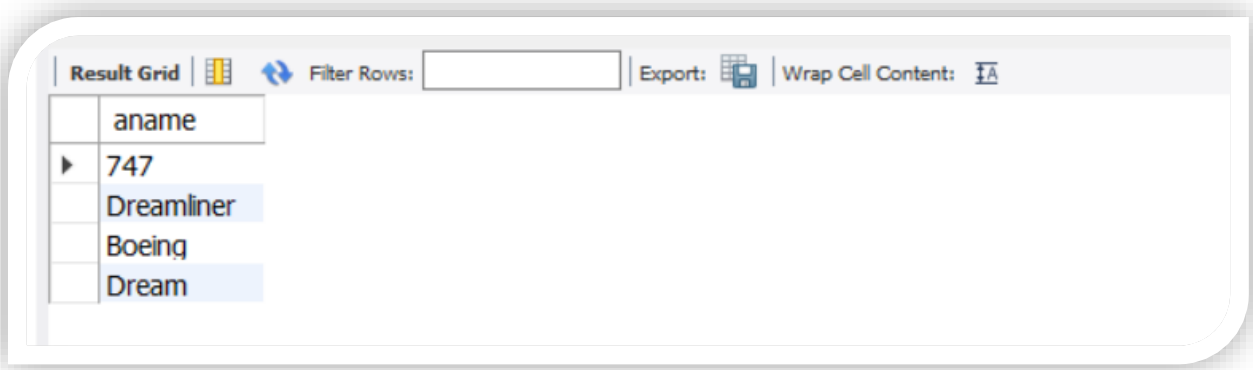
```
select ename from employees where eid in(
select eid from certified where aid in(
select aid from aircraft where aname = 'Boeing'));
```



ename
A

-- Query 6

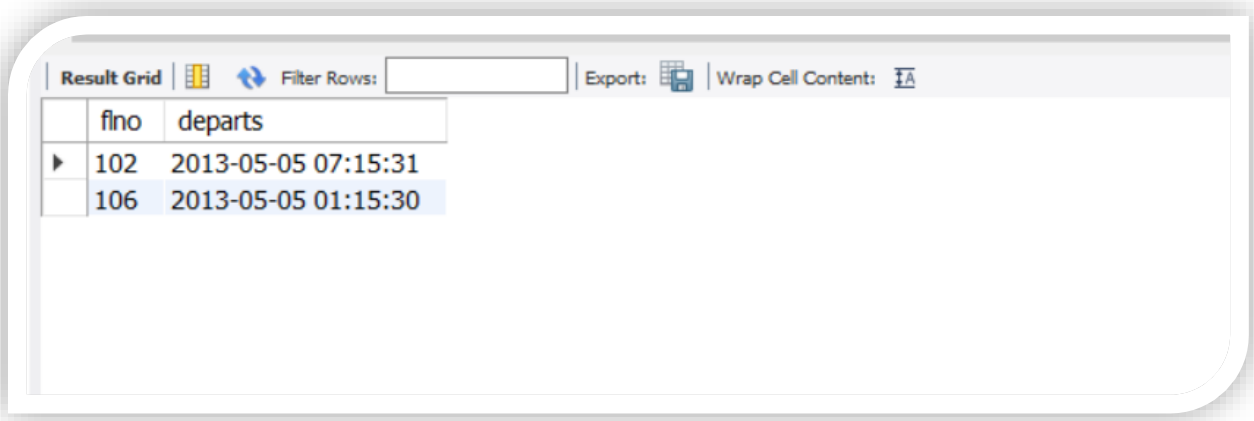
select aname from aircraft where cruisingrange > any (select distance from flights where fromplace='Bangalore' and toplace='Delhi');



-- Query 7

```
SELECT F.flno, F.departs
FROM flights F
WHERE F.flno IN ( ( SELECT F0.flno
 FROM flights F0
 WHERE F0.fromplace = 'Bangalore' AND F0.toplace = 'Kolkata'
 AND extract(hour from F0.arrives) < 18 )
 UNION
( SELECT F0.flno
 FROM flights F0, flights F1
 WHERE F0.fromplace = 'Bangalore' AND F0.toplace <> 'Kolkata'
 AND F0.toplace = F1.fromplace AND F1.toplace = 'Kolkata'
 AND F1.departs > F0.arrives
```

```
 AND extract(hour from F1.arrives) < 18)
 UNION
( SELECT F0.flno
 FROM flights F0, flights F1, flights F2
 WHERE F0.fromplace = 'Bangalore'
 AND F0.toplace = F1.fromplace
 AND F1.toplace = F2.fromplace
 AND F2.toplace = 'Kolkata'
 AND F0.toplace <> 'Kolkata'
 AND F1.toplace <> 'Kolkata'
 AND F1.departs > F0.arrives
 AND F2.departs > F1.arrives
 AND extract(hour from F2.arrives) < 18));
```



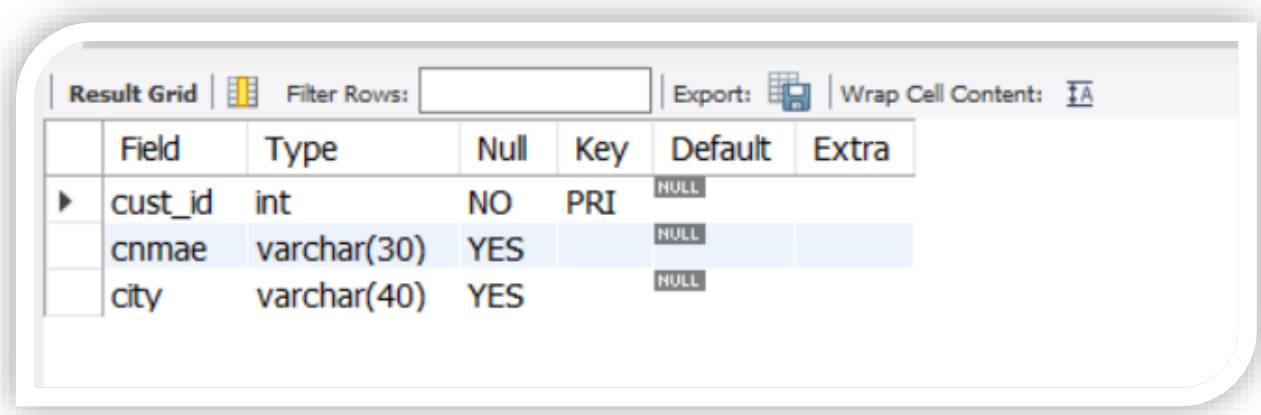
PROGRAM 6: Order Database

Consider the following relations for an Order Processing database application in a company.

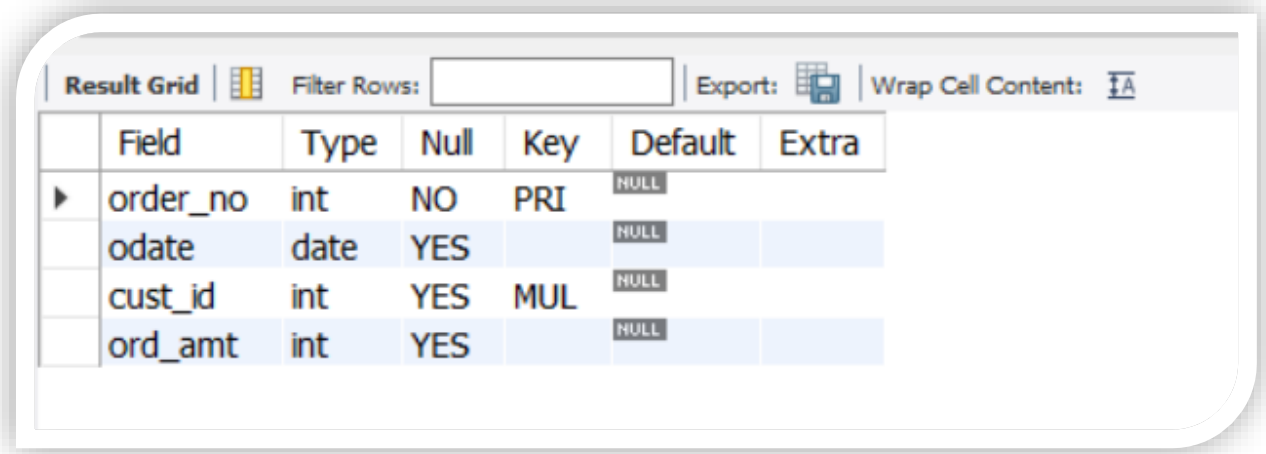
CUSTOMER (CUST #: int, cname: String, city: String)
ORDER (order #: int, odate: date, cust #: int, ord-Amt: int)
ITEM (item #: int, unit-price: int)
ORDER-ITEM (order #: int, item #: int, qty: int)
WAREHOUSE (warehouse #: int, city: String)
SHIPMENT (order #: int, warehouse #: int, ship-date: date)

i) Create the above tables by properly specifying the primary keys and the foreign keys and the foreign keys.
ii) Enter at least five tuples for each relation.
iii) Produce a listing: CUSTNAME, #oforders, AVG_ORDER_AMT, where the middle column is the total
numbers of orders by the customer and the last column is the average order amount for that customer.
iv) List the order# for orders that were shipped from all warehouses that the company has in a specific city.
v) Demonstrate how you delete item# 10 from the ITEM table and make that field null in the
ORDER_ITEM table.

```
create database order_processing;

/* CUSTOMER Table */
create table customer(
    cust_id int,
    cnmae varchar(30),
    city varchar(40),
    primary key(cust_id));

desc customer;
```

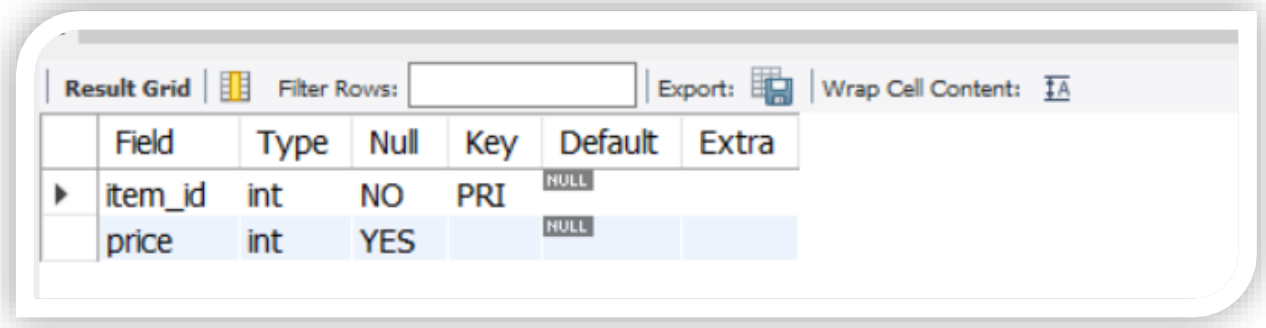



```
/* ORDERS Table */
create table orders(
    order_no int,
    odate date,
    cust_id int,
    ord_amt int,
    primary key(order_no),
    foreign key(cust_id)references customer (cust_id));

desc orders;
```



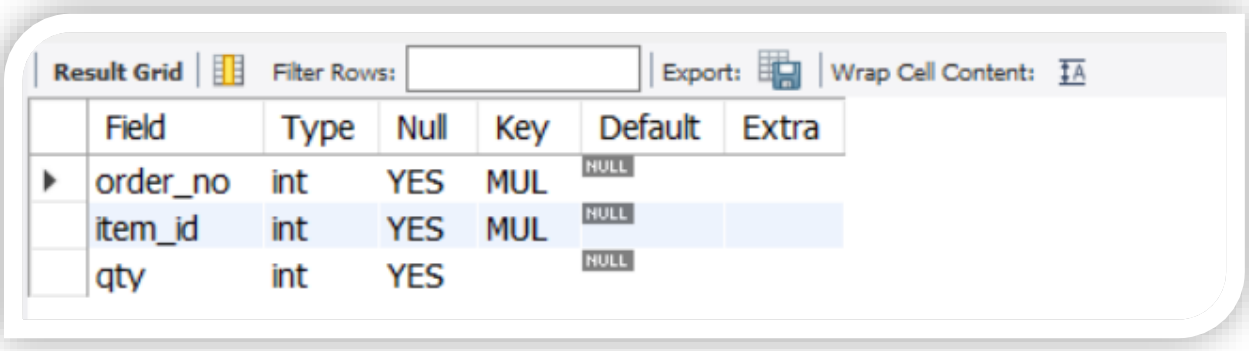
Field	Type	Null	Key	Default	Extra
order_no	int	NO	PRI	NULL	
odate	date	YES		NULL	
cust_id	int	YES	MUL	NULL	
ord_amt	int	YES		NULL	

```
/* ITEM Table */
create table item(
    item_id int,
    price int,
    primary key(item_id));

desc item;
```



Field	Type	Null	Key	Default	Extra
item_id	int	NO	PRI	NULL	
price	int	YES		NULL	

```
/* ORDER_ITEM Table */
create table order_item(
    order_no int,
    item_id int,
    qty int,
    foreign key(order_no)references orders (order_no),
    foreign key(item_id)references item(item_id)on delete set NULL);

desc order_item;
```

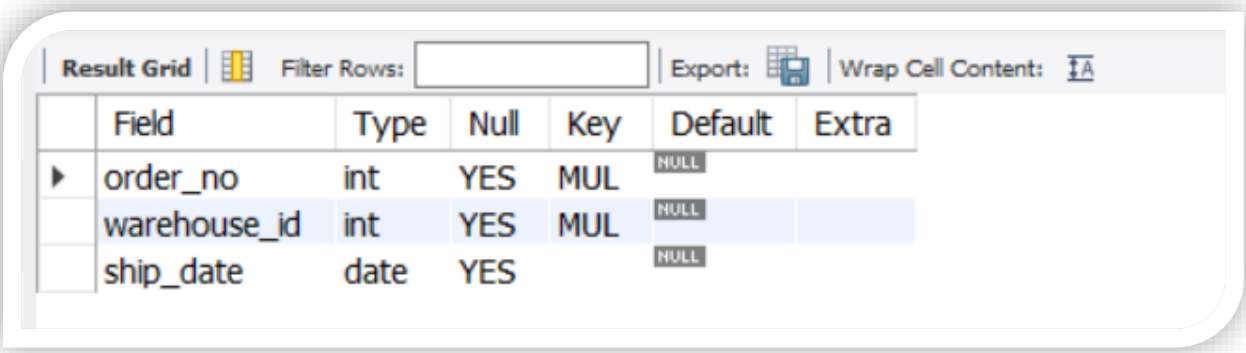


Field	Type	Null	Key	Default	Extra
order_no	int	YES	MUL	NULL	
item_id	int	YES	MUL	NULL	
qty	int	YES		NULL	

```
/* WAREHOUSE Table */
create table warehouse(
    warehouse_id int,
    city varchar(40),
    primary key(warehouse_id));

desc warehouse;
```

Field	Type	Null	Key	Default	Extra
warehouse_id	int	NO	PRI	NULL	
city	varchar(40)	YES		NULL	

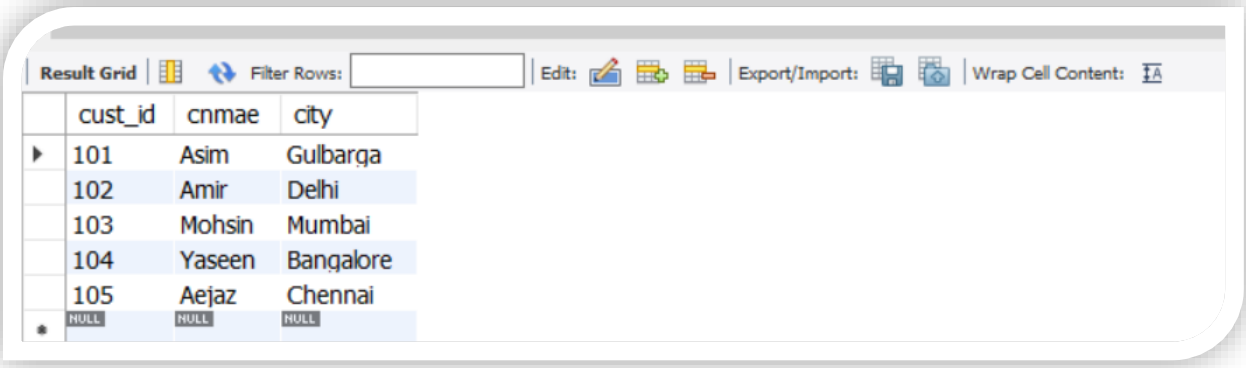
```
/* SHIPMENT Table */
create table shipment(
    order_no int,
    warehouse_id int,
    ship_date date,
    foreign key(order_no)references orders(order_no),
    foreign key(warehouse_id)references warehouse(warehouse_id));

desc shipment;
```



Field	Type	Null	Key	Default	Extra
order_no	int	YES	MUL	NULL	
warehouse_id	int	YES	MUL	NULL	
ship_date	date	YES		NULL	

/* Inserts Values To The CUSTOMER Table */
INSERT INTO customer (cust_id, cnmae, city) VALUES (101, 'Asim', 'Gulbarga');
INSERT INTO customer (cust_id, cnmae, city) VALUES (102, 'Amir', 'Delhi');
INSERT INTO customer (cust_id, cnmae, city) VALUES (103, 'Mohsin', 'Mumbai');
INSERT INTO customer (cust_id, cnmae, city) VALUES (104, 'Yaseen', 'Bangalore');
INSERT INTO customer (cust_id, cnmae, city) VALUES (105, 'Aejaz', 'Chennai');

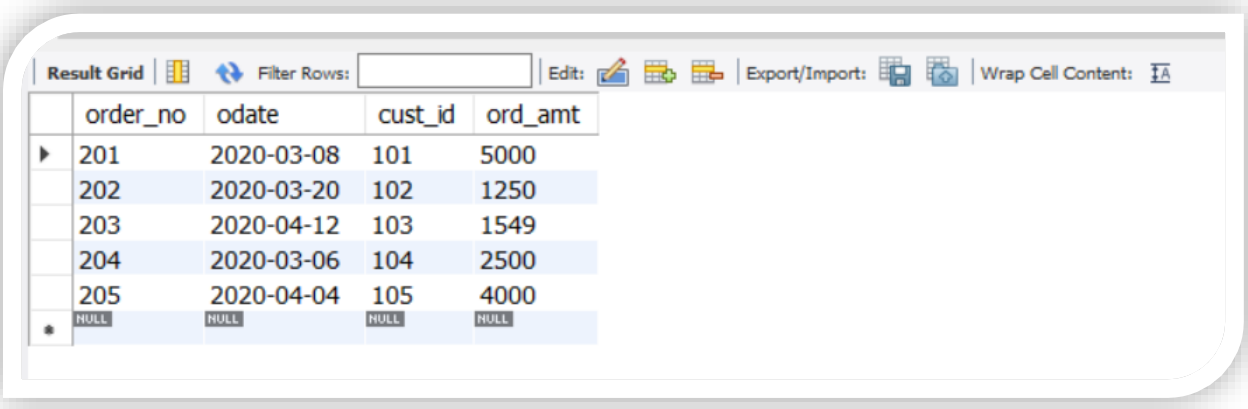
select * from customer;



cust_id	cnmae	city
101	Asim	Gulbarga
102	Amir	Delhi
103	Mohsin	Mumbai
104	Yaseen	Bangalore
105	Aejaz	Chennai
NULL	NULL	NULL

/* Inserts Values To The ORDERS Table */
INSERT INTO orders (order_no, odate, cust_id, ord_amt) VALUES ('201', '2020-03-08', '101', '5000');
INSERT INTO orders (order_no, odate, cust_id, ord_amt) VALUES ('202', '2020-03-20', '102', '1250');
INSERT INTO orders (order_no, odate, cust_id, ord_amt) VALUES ('203', '2020-04-12', '103', '1549');
INSERT INTO orders (order_no, odate, cust_id, ord_amt) VALUES ('204', '2020-03-06', '104', '2500');
INSERT INTO orders (order_no, odate, cust_id, ord_amt) VALUES ('205', '2020-04-04', '105', '4000');

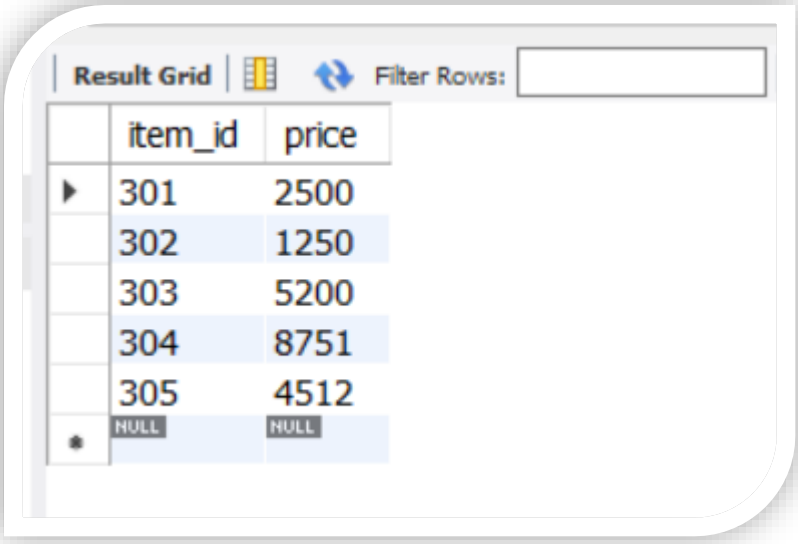
select * from orders;



order_no	odate	cust_id	ord_amt
201	2020-03-08	101	5000
202	2020-03-20	102	1250
203	2020-04-12	103	1549
204	2020-03-06	104	2500
205	2020-04-04	105	4000
NULL	NULL	NULL	NULL

/* Inserts Values To The ITEM Table */
INSERT INTO item (item_id, price) VALUES ('301', '2500');
INSERT INTO item (item_id, price) VALUES ('302', '1250');
INSERT INTO item (item_id, price) VALUES ('303', '5200');
INSERT INTO item (item_id, price) VALUES ('304', '8751');
INSERT INTO item (item_id, price) VALUES ('305', '4512');

select * from item;

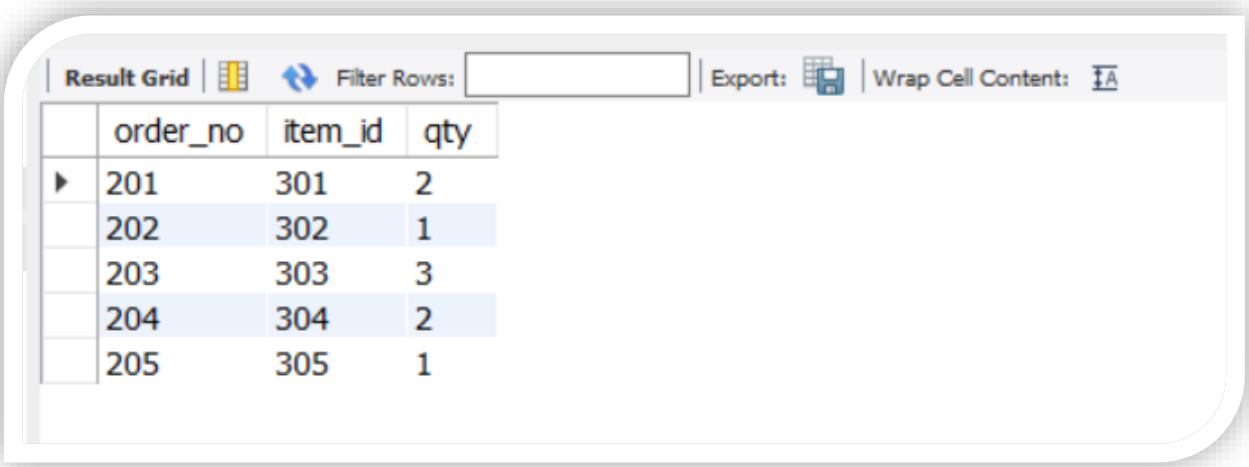


item_id	price
301	2500
302	1250
303	5200
304	8751
305	4512
NULL	NULL

/* Inserts Values To The ORDER_ITEM Table */
INSERT INTO order_item (order_no, item_id, qty) VALUES ('201', '301', '2');
INSERT INTO order_item (order_no, item_id, qty) VALUES ('202', '302', '1');
INSERT INTO order_item (order_no, item_id, qty) VALUES ('203', '303', '3');
INSERT INTO order_item (order_no, item_id, qty) VALUES ('204', '304', '2');

INSERT INTO order_item (order_no, item_id, qty) VALUES ('205', '305', '1');

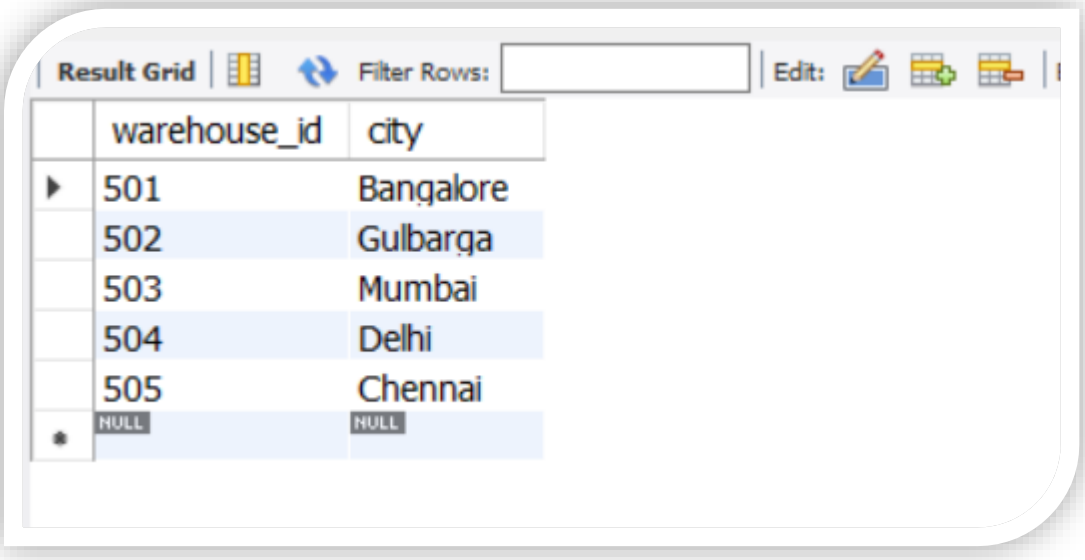
select * from order_item;



order_no	item_id	qty
201	301	2
202	302	1
203	303	3
204	304	2
205	305	1

/* Inserts Values To The WAREHOUSE Table */
INSERT INTO warehouse (warehouse_id, city) VALUES ('501', 'Bangalore');
INSERT INTO warehouse (warehouse_id, city) VALUES ('502', 'Gulbarga');
INSERT INTO warehouse (warehouse_id, city) VALUES ('503', 'Mumbai');
INSERT INTO warehouse (warehouse_id, city) VALUES ('504', 'Delhi');
INSERT INTO warehouse (warehouse_id, city) VALUES ('505', 'Chennai');

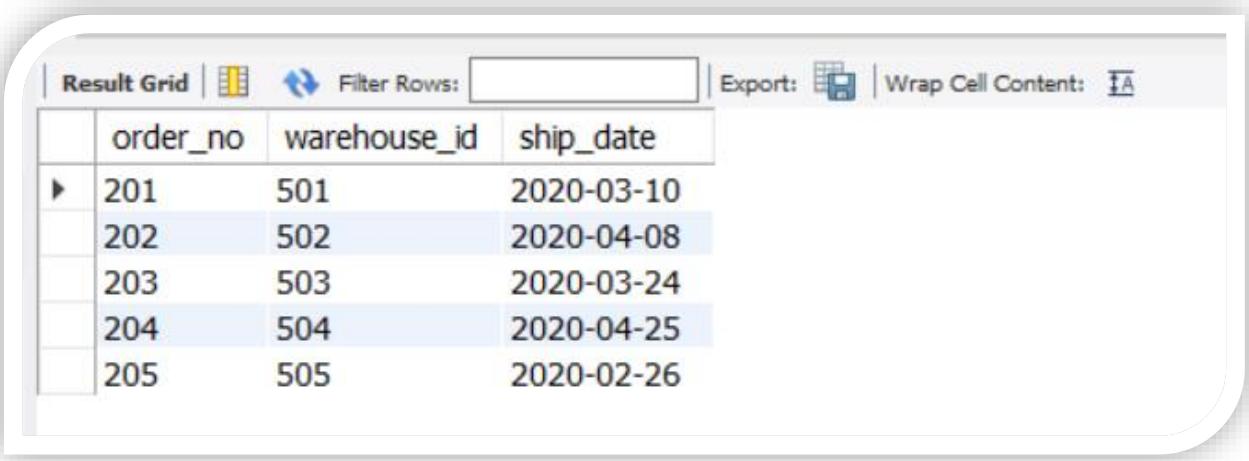
select * from warehouse;



warehouse_id	city
501	Bangalore
502	Gulbarga
503	Mumbai
504	Delhi
505	Chennai
NULL	NULL

/* Inserts Values To The SHIPMENT Table */
INSERT INTO shipment (order_no, warehouse_id, ship_date) VALUES ('201', '501', '2020-03-10');
INSERT INTO shipment (order_no, warehouse_id, ship_date) VALUES ('202', '502', '2020-04-08');
INSERT INTO shipment (order_no, warehouse_id, ship_date) VALUES ('203', '503', '2020-03-24');
INSERT INTO shipment (order_no, warehouse_id, ship_date) VALUES ('204', '504', '2020-04-25');
INSERT INTO shipment (order_no, warehouse_id, ship_date) VALUES ('205', '505', '2020-02-26');

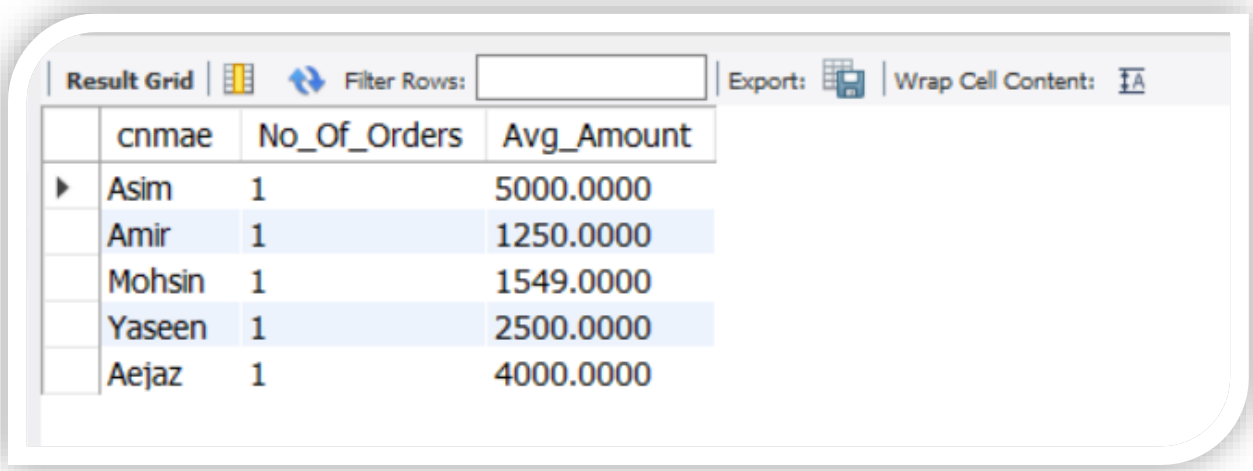
select * from shipment;


Result Grid | Filter Rows: | Export: | Wrap Cell Content:

order_no	warehouse_id	ship_date
201	501	2020-03-10
202	502	2020-04-08
203	503	2020-03-24
204	504	2020-04-25
205	505	2020-02-26

/* 01. Produce a listing: CUSTNAME, #oforders, AVG_ORDER_AMT, where the middle column is the total numbers of orders by the customer and the last column is the average order amount for that customer. */

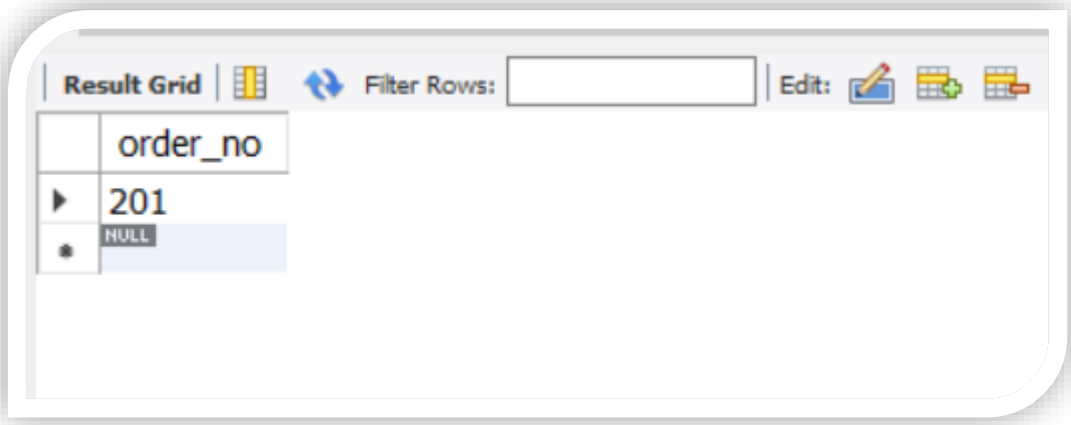
select C.cnmae,count(order_no) as No_Of_Orders,avg(ord_amt) as Avg_Amount
from customer C,orders O
where C.cust_id = O.cust_id
group by C.cnmae;



cnmae	No_Of_Orders	Avg_Amount
Asim	1	5000.0000
Amir	1	1250.0000
Mohsin	1	1549.0000
Yaseen	1	2500.0000
Aejaz	1	4000.0000

/* 02. List the order# for orders that were shipped from all warehouses that the company has in a specific city. */

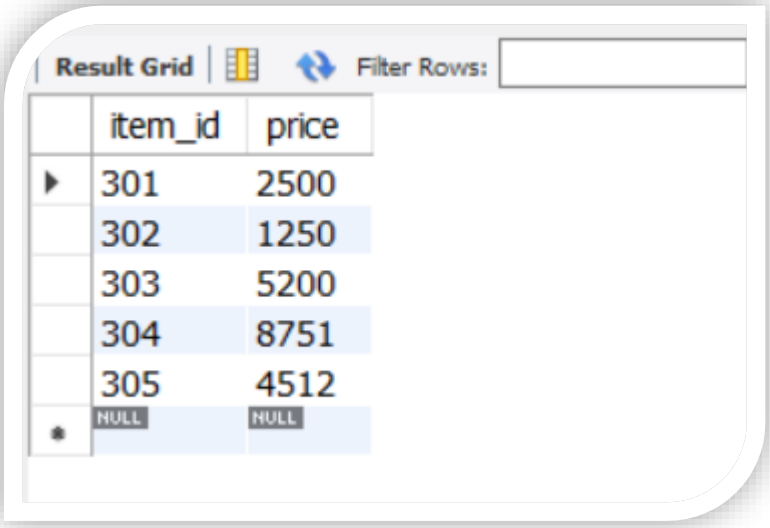
```
select order_no from orders where order_no in
(select order_no from shipment
where warehouse_id in
(select warehouse_id from warehouse
where city = 'Bangalore'));
```



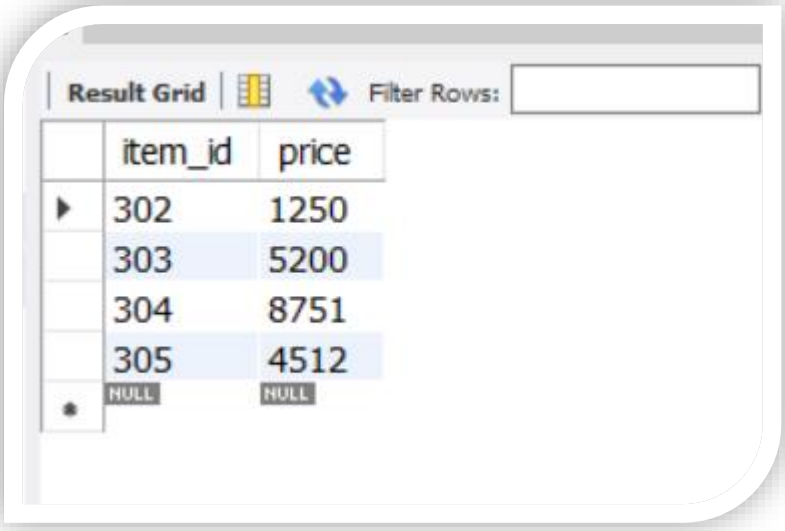
order_no
201
NULL

/* 03. Demonstrate how you delete item# 10 from the ITEM table and make that field null in the ORDER_ITEM table. */

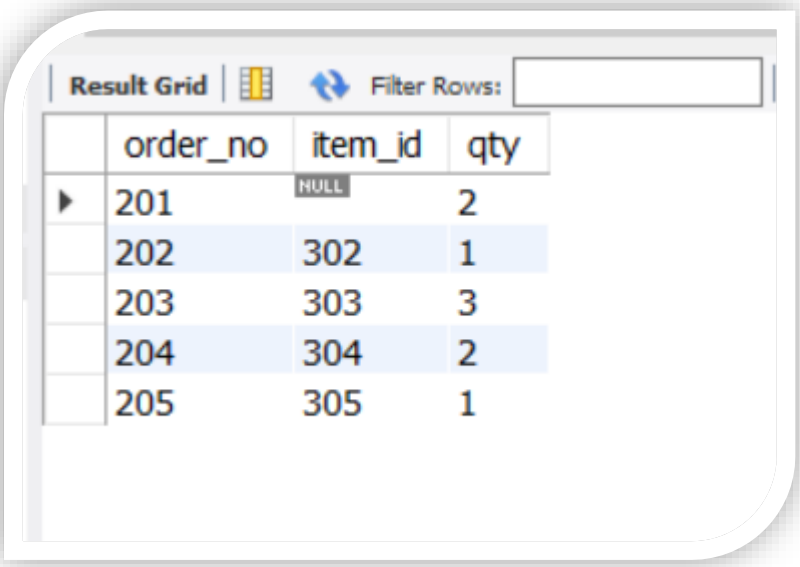
select * from item;



delete from item where item_id = 301;
select * from item;



select * from order_item;



order_no	item_id	qty
201	NULL	2
202	302	1
203	303	3
204	304	2
205	305	1

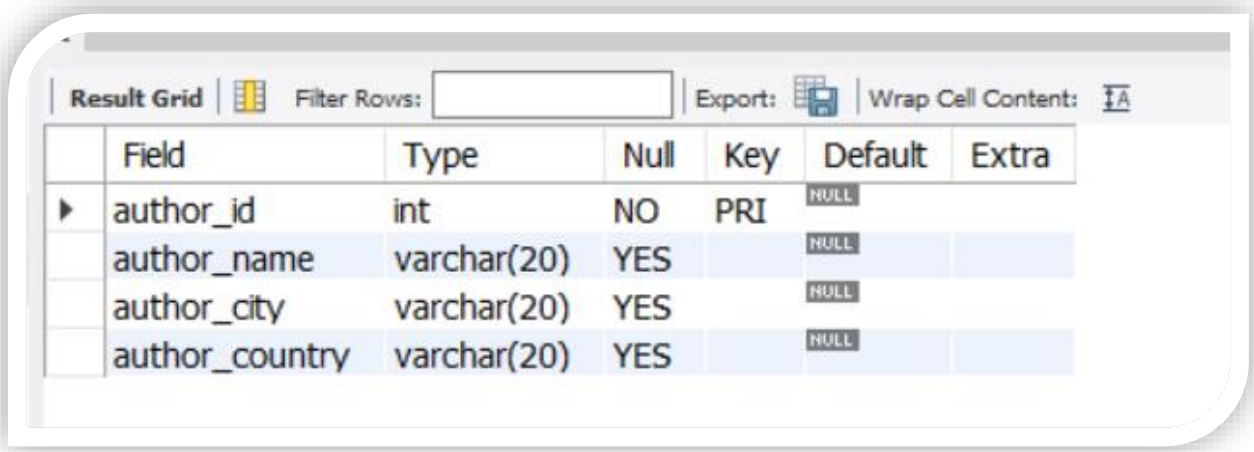
commit;

PROGRAM 7. BOOK DEALER DATABASE

The following tables are maintained by a book dealer:
AUTHOR(author-id: int, name: String, city: String, country: String)
PUBLISHER(publisher-id: int, name: String, city: String, country: String)
CATALOG(book-id: int, title: String, author-id: int, publisher-id: int, category-id: int, year: int, price: int)
CATEGORY(category-id: int, description: String)
ORDER-DETAILS(order-no: int, book-id: int, quantity: int)

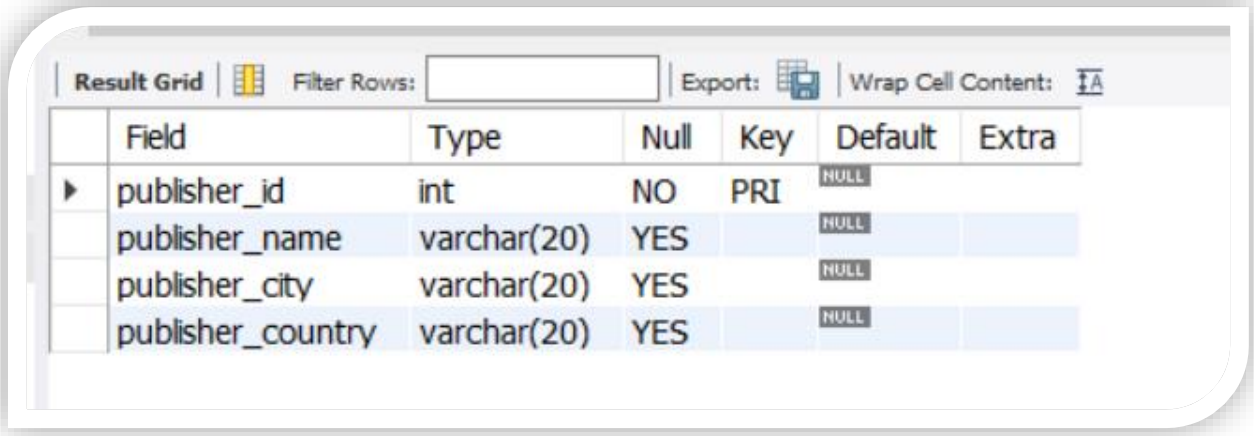
i) Create the above tables by properly specifying the primary keys and the foreign keys.
ii) Enter at least five tuples for each relation.
iii) Give the details of the authors who have 2 or more books in the catalog and the price of the books in the catalog and the year of publication is after 2000.
iv) Find the author of the book which has maximum sales.
v) Demonstrate how you increase the price of books published by a specific publisher by 10%.

```
create database book_dealer;
use book_dealer;

create table author (
 author_id INT,
 author_name VARCHAR(20),
 author_city VARCHAR(20),
 author_country VARCHAR(20),
 PRIMARY KEY(author_id));
desc author;
```



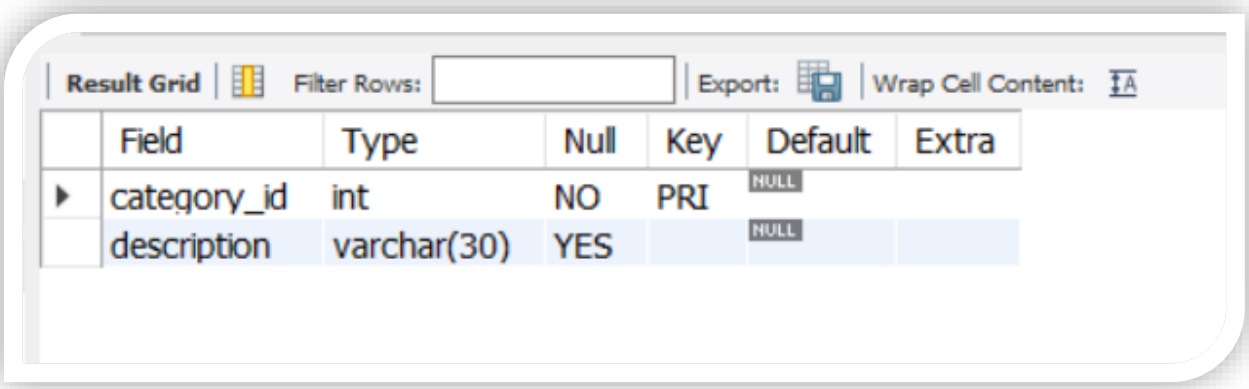
Field	Type	Null	Key	Default	Extra
author_id	int	NO	PRI	NULL	
author_name	varchar(20)	YES		NULL	
author_city	varchar(20)	YES		NULL	
author_country	varchar(20)	YES		NULL	

```
create table publisher (
publisher_id INT,
publisher_name VARCHAR(20),
publisher_city VARCHAR(20),
publisher_country VARCHAR(20),
PRIMARY KEY(publisher_id));
desc publisher;
```



Field	Type	Null	Key	Default	Extra
publisher_id	int	NO	PRI	NULL	
publisher_name	varchar(20)	YES		NULL	
publisher_city	varchar(20)	YES		NULL	
publisher_country	varchar(20)	YES		NULL	

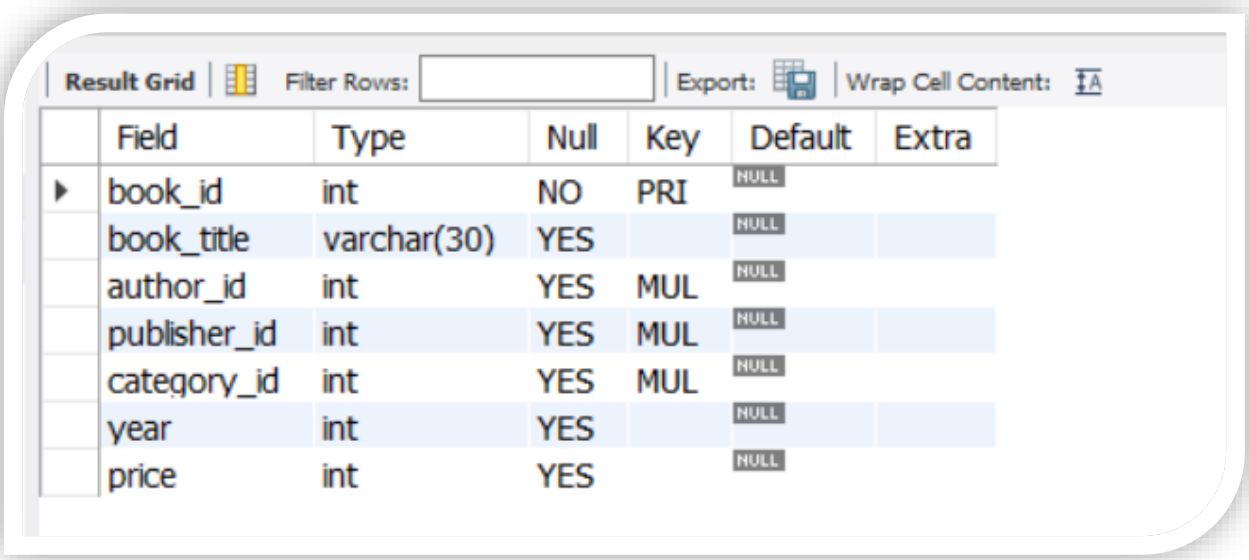
```
create table category (
category_id INT,
description VARCHAR(30),
PRIMARY KEY(category_id));
desc category;
```



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
category_id	int	NO	PRI	NULL	
description	varchar(30)	YES		NULL	

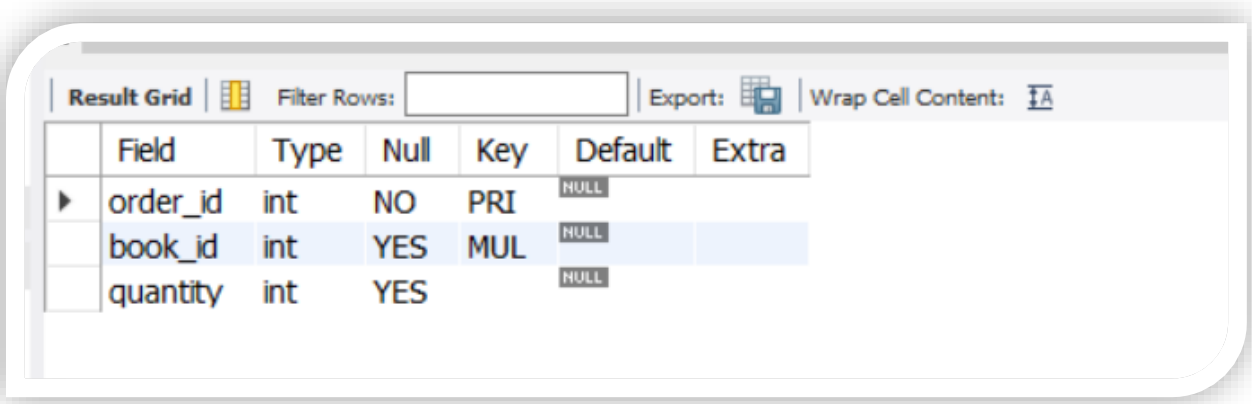
```
CREATE TABLE catalog(
 book_id INT,
 book_title VARCHAR(30),
 author_id INT,
 publisher_id INT,
 category_id INT,
 year INT,
 price INT,
 PRIMARY KEY(book_id),
 FOREIGN KEY(author_id) REFERENCES author(author_id),
 FOREIGN KEY(publisher_id) REFERENCES publisher(publisher_id),
 FOREIGN KEY(category_id) REFERENCES category(category_id) );
desc catalog;
```



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
book_id	int	NO	PRI	NULL	
book_title	varchar(30)	YES		NULL	
author_id	int	YES	MUL	NULL	
publisher_id	int	YES	MUL	NULL	
category_id	int	YES	MUL	NULL	
year	int	YES		NULL	
price	int	YES		NULL	

```
CREATE TABLE orderdetails(
order_id INT,
book_id INT,
quantity INT,
PRIMARY KEY(order_id),
FOREIGN KEY(book_id) REFERENCES catalog(book_id));
desc orderdetails;
```

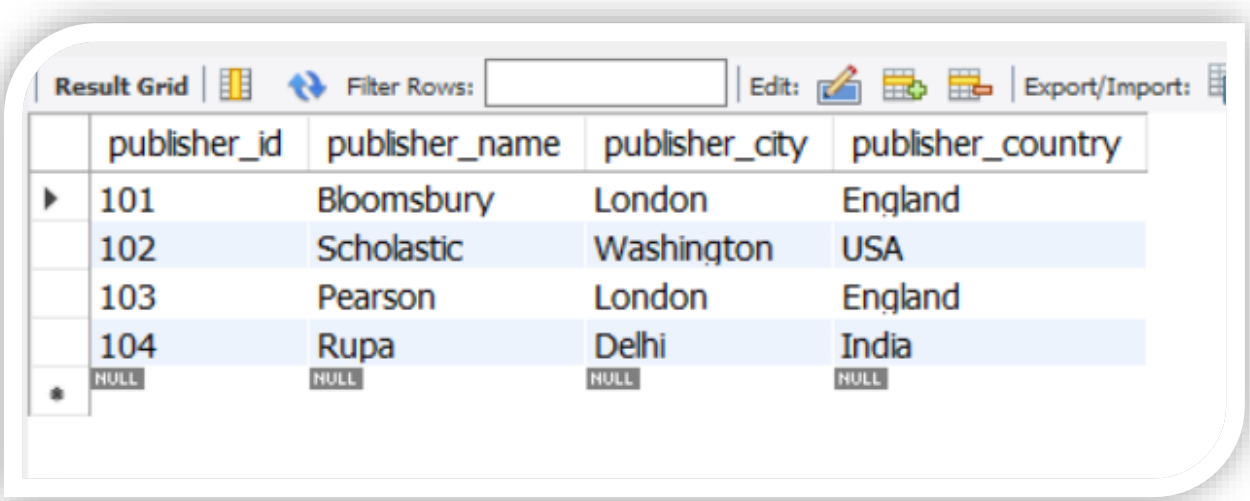


Field	Type	Null	Key	Default	Extra
order_id	int	NO	PRI	NULL	
book_id	int	YES	MUL	NULL	
quantity	int	YES		NULL	

```
INSERT INTO author (author_id,author_name,author_city,author_country) VALUES
(1001,'JK Rowling','London','England'),
(1002,'Chetan Bhagat','Mumbai','India'),
(1003,'John McCarthy','Chicago','USA'),
(1004,'Dan Brown','California','USA') ;
select * from author;
```

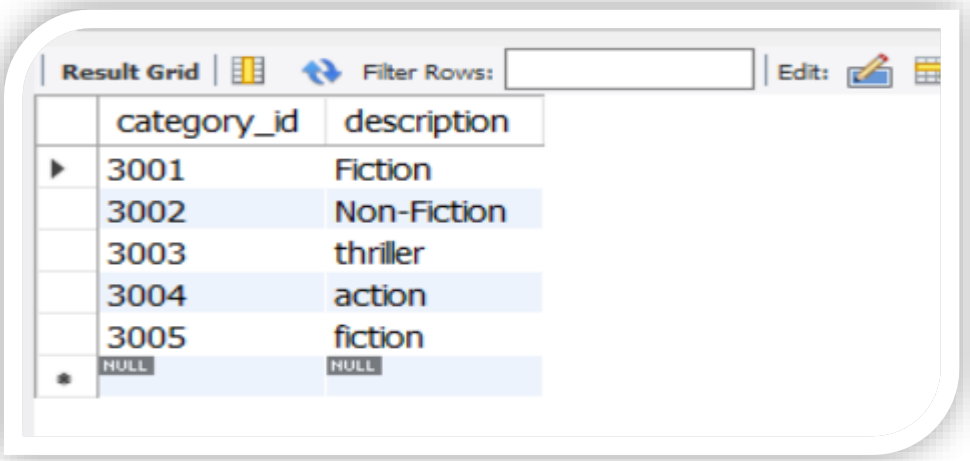
author_id	author_name	author_city	author_country
1001	JK Rowling	London	England
1002	Chetan Bhagat	Mumbai	India
1003	John McCarthy	Chicago	USA
1004	Dan Brown	California	USA
NULL	NULL	NULL	NULL

```
INSERT INTO publisher
(publisher_id,publisher_name,publisher_city,publisher_country) VALUES
 (101,'Bloomsbury','London','England'),
 (102,'Scholastic','Washington','USA'),
 (103,'Pearson','London','England'),
 (104,'Rupa','Delhi','India');
 select * from publisher;
```



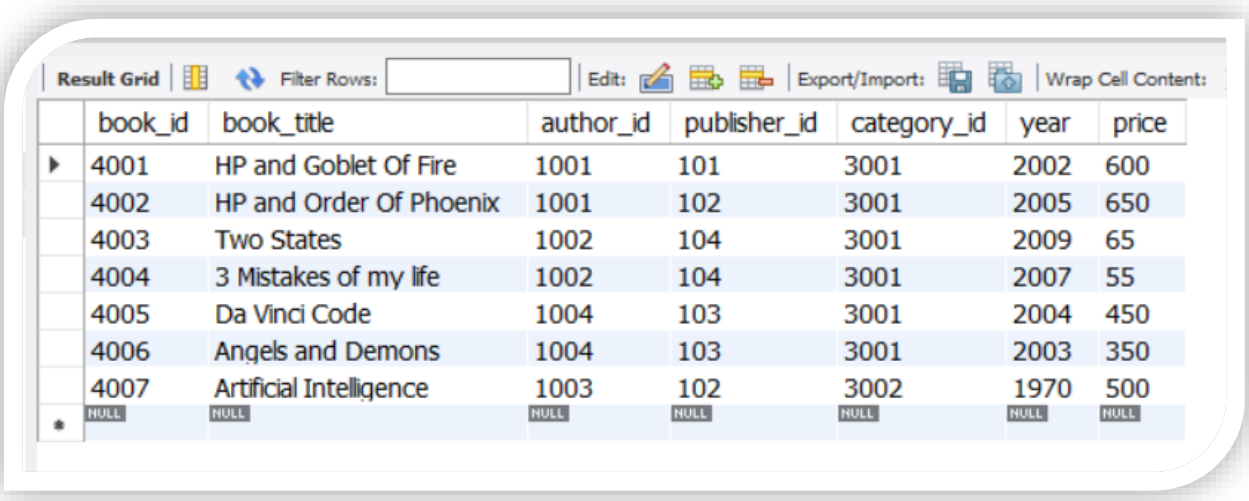
publisher_id	publisher_name	publisher_city	publisher_country
101	Bloomsbury	London	England
102	Scholastic	Washington	USA
103	Pearson	London	England
104	Rupa	Delhi	India
NULL	NULL	NULL	NULL

```
INSERT INTO category (category_id,description) VALUES
 (3001,'Fiction'),
 (3002,'Non-Fiction'),
 (3003,'thriller'),
 (3004,'action'),
 (3005,'fiction') ;
select * from category;
```



category_id	description
3001	Fiction
3002	Non-Fiction
3003	thriller
3004	action
3005	fiction
NULL	NULL

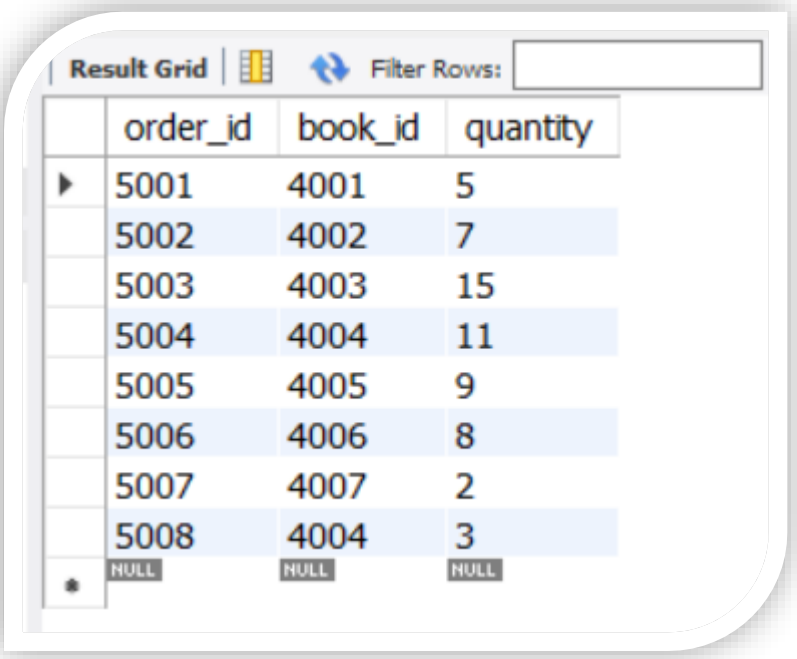
```
INSERT INTO catalog
(book_id,book_title,author_id,publisher_id,category_id,year,price) VALUES
 (4001,'HP and Goblet Of Fire',1001,101,3001,2002,600),
 (4002,'HP and Order Of Phoenix',1001,102,3001,2005,650),
 (4003,'Two States',1002,104,3001,2009,65),
 (4004,'3 Mistakes of my life',1002,104,3001,2007,55),
 (4005,'Da Vinci Code',1004,103,3001,2004,450),
 (4006,'Angels and Demons',1004,103,3001,2003,350),
 (4007,'Artificial Intelligence',1003,102,3002,1970,500);
 select * from catalog;
```



Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

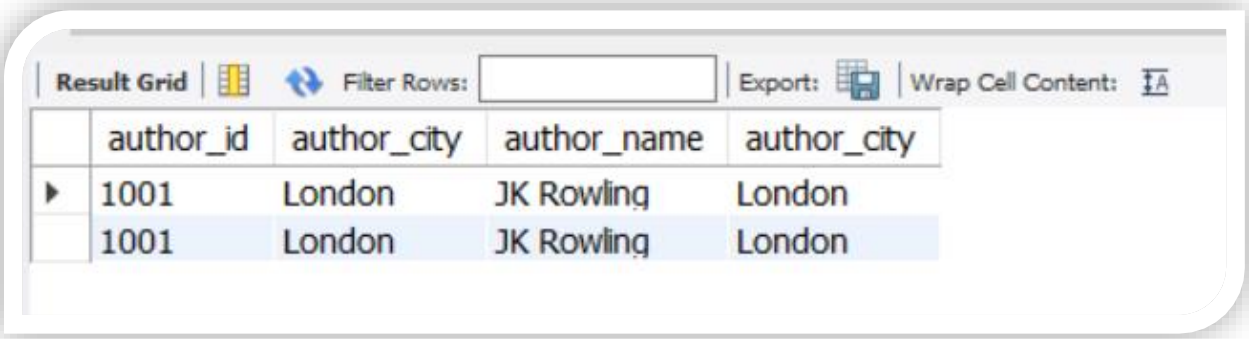
book_id	book_title	author_id	publisher_id	category_id	year	price
4001	HP and Goblet Of Fire	1001	101	3001	2002	600
4002	HP and Order Of Phoenix	1001	102	3001	2005	650
4003	Two States	1002	104	3001	2009	65
4004	3 Mistakes of my life	1002	104	3001	2007	55
4005	Da Vinci Code	1004	103	3001	2004	450
4006	Angels and Demons	1004	103	3001	2003	350
4007	Artificial Intelligence	1003	102	3002	1970	500
NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
INSERT INTO orderdetails (order_id,book_id,quantity) VALUES
 (5001,4001,5),
 (5002,4002,7),
 (5003,4003,15),
 (5004,4004,11),
 (5005,4005,9),
 (5006,4006,8),
 (5007,4007,2),
 (5008,4004,3) ;
 select * from orderdetails;
```



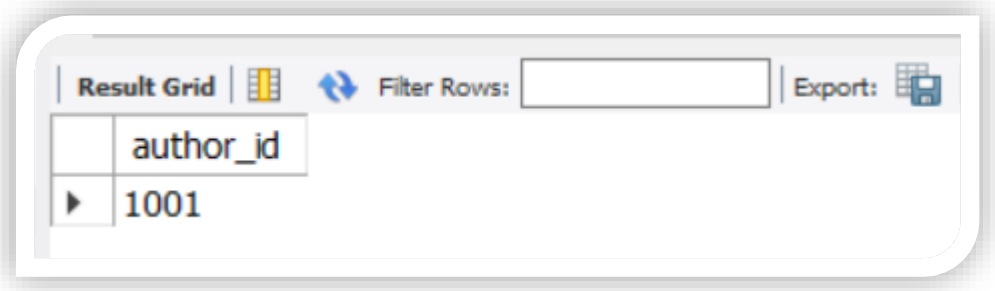
order_id	book_id	quantity
5001	4001	5
5002	4002	7
5003	4003	15
5004	4004	11
5005	4005	9
5006	4006	8
5007	4007	2
5008	4004	3
NULL	NULL	NULL

```
select a.author_id,a.author_city,a.author_name,a.author_city
from author a join catalog c on a.author_id=c.author_id where
year>2000 and c.author_id in(select c.author_id from catalog having
count(c.author_id)>2 and price>500 );
```

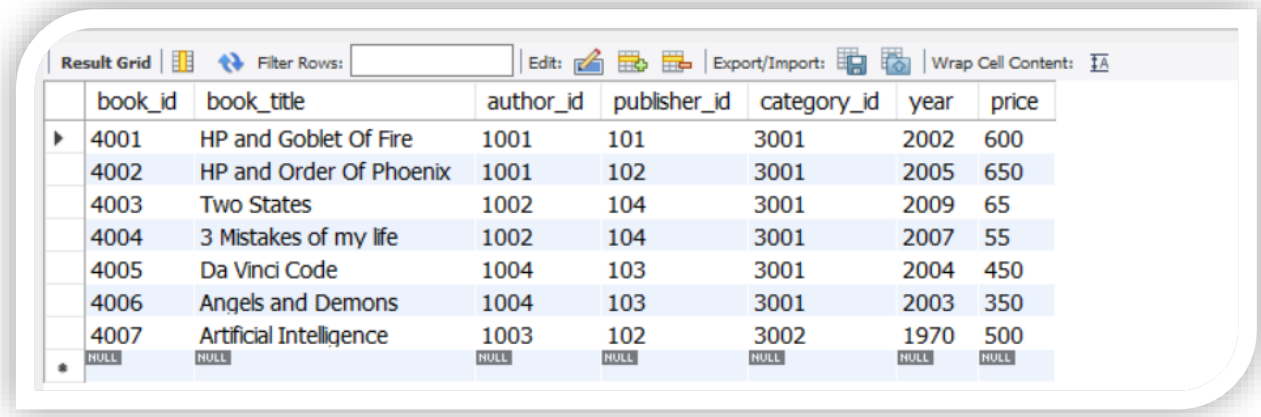


author_id	author_city	author_name	author_city
1001	London	JK Rowling	London
1001	London	JK Rowling	London

select a.author_id from author a join catalog c on
a.author_id=c.author_id join orderdetails o on c.book_id=o.book_id
having max(o.quantity);



update catalog set price=(0.1*price)+price where publisher_id=2001;
select * from catalog;
commit;



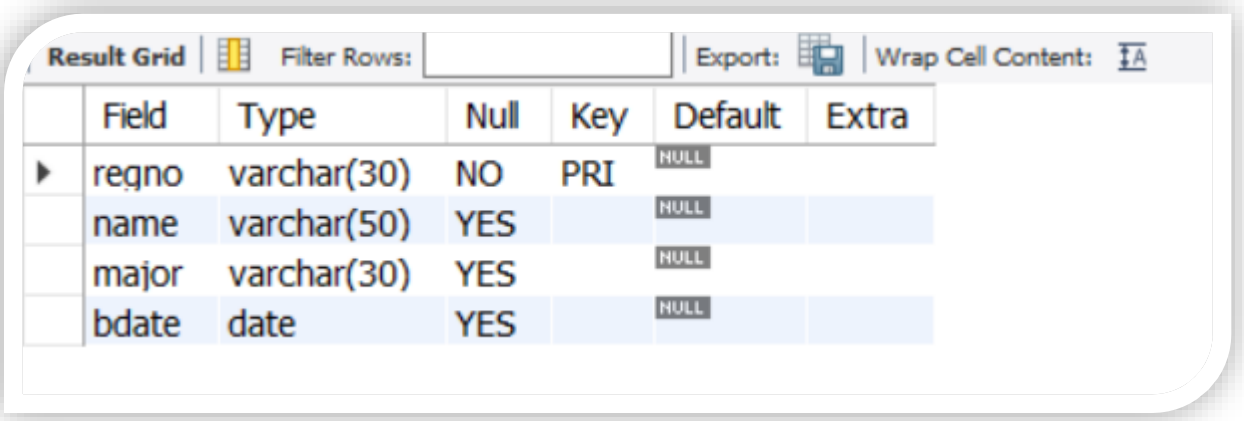
book_id	book_title	author_id	publisher_id	category_id	year	price
4001	HP and Goblet Of Fire	1001	101	3001	2002	600
4002	HP and Order Of Phoenix	1001	102	3001	2005	650
4003	Two States	1002	104	3001	2009	65
4004	3 Mistakes of my life	1002	104	3001	2007	55
4005	Da Vinci Code	1004	103	3001	2004	450
4006	Angels and Demons	1004	103	3001	2003	350
4007	Artificial Intelligence	1003	102	3002	1970	500
NULL	NULL	NULL	NULL	NULL	NULL	NULL

PROGRAM 8: STUDENT ENROLLMENT DATABASE

Consider the following database of student enrollment in courses and books adopted for each course.
STUDENT (regno: String, name: String, major: String, bdate: date)
COURSE (course #: int, cname: String, dept: String)
ENROLL (regno: String, cname: String, sem: int, marks: int)
BOOK_ADOPTION (course #: int, sem: int, book-ISBN: int)
TEXT(book-ISBN:int, book-title:String, publisher:String, author:String)

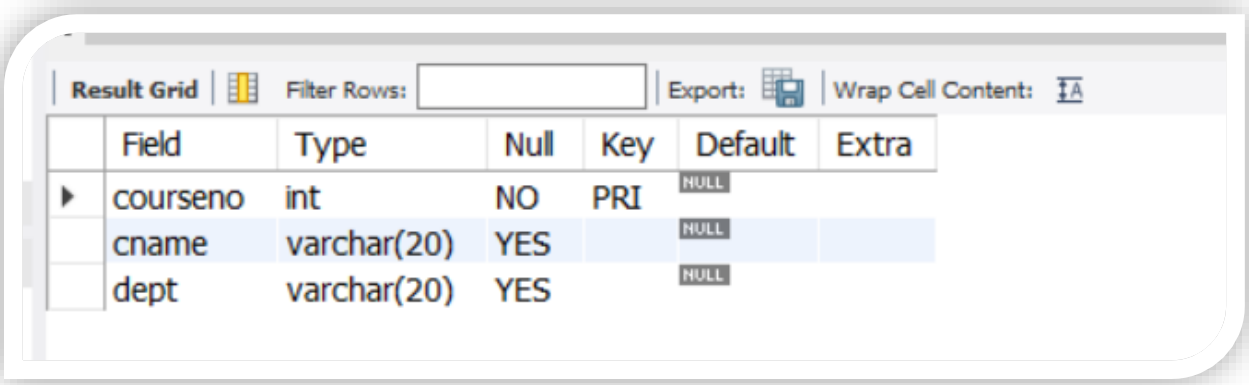
i) Create the above tables by properly specifying the primary keys and the foreign keys.
ii) Enter at least five tuples for each relation.
iii) Demonstrate how you add a new text book to the database and make this book be adopted by some
department.
iv) Produce a list of text books (include Course #, Book-ISBN, Book-title) in the alphabetical order for
courses offered by the 'CS' department that use more than two books.
v) List any department that has all its adopted books published by a specific publisher.

```
create database Stud_Enrollment;
use Stud_Enrollment;

CREATE TABLE student(
 regno VARCHAR(30),
 name VARCHAR(50),
 major VARCHAR(30),
 bdate DATE,
 PRIMARY KEY (regno));
desc student;
```



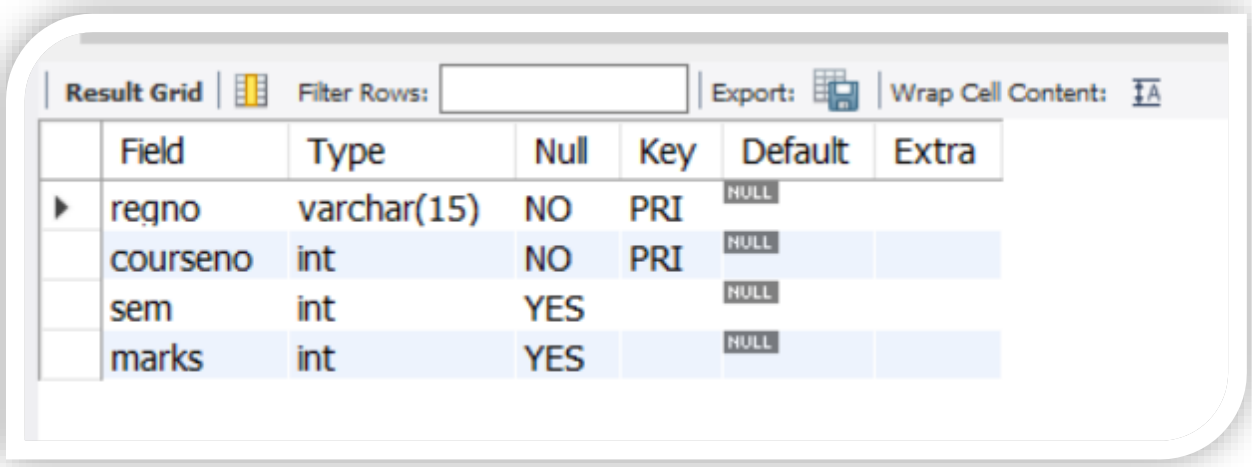
Field	Type	Null	Key	Default	Extra
regno	varchar(30)	NO	PRI	NULL	
name	varchar(50)	YES		NULL	
major	varchar(30)	YES		NULL	
bdate	date	YES		NULL	

```
CREATE TABLE course(
 courseno INT,
 cname VARCHAR(20),
 dept VARCHAR(20),
 PRIMARY KEY (courseno));
desc course;
```



Field	Type	Null	Key	Default	Extra
courseno	int	NO	PRI	NULL	
cname	varchar(20)	YES		NULL	
dept	varchar(20)	YES		NULL	

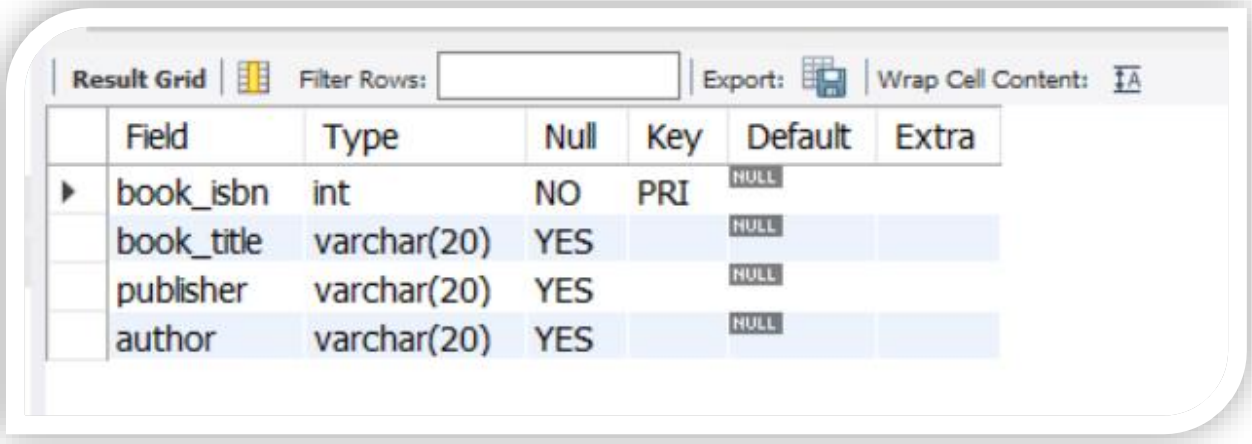
```
CREATE TABLE enroll(
 regno VARCHAR(15),
 courseno INT,
 sem INT(3),
 marks INT(4),
 PRIMARY KEY (regno,courseno),
 FOREIGN KEY (regno) REFERENCES student (regno),
 FOREIGN KEY (courseno) REFERENCES course (courseno));
desc enroll;
```



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
regno	varchar(15)	NO	PRI	NULL	
courseno	int	NO	PRI	NULL	
sem	int	YES		NULL	
marks	int	YES		NULL	

```
CREATE TABLE text(
 book_isbn INT(5),
 book_title VARCHAR(20),
 publisher VARCHAR(20),
 author VARCHAR(20),
 PRIMARY KEY (book_isbn));
desc text;
```

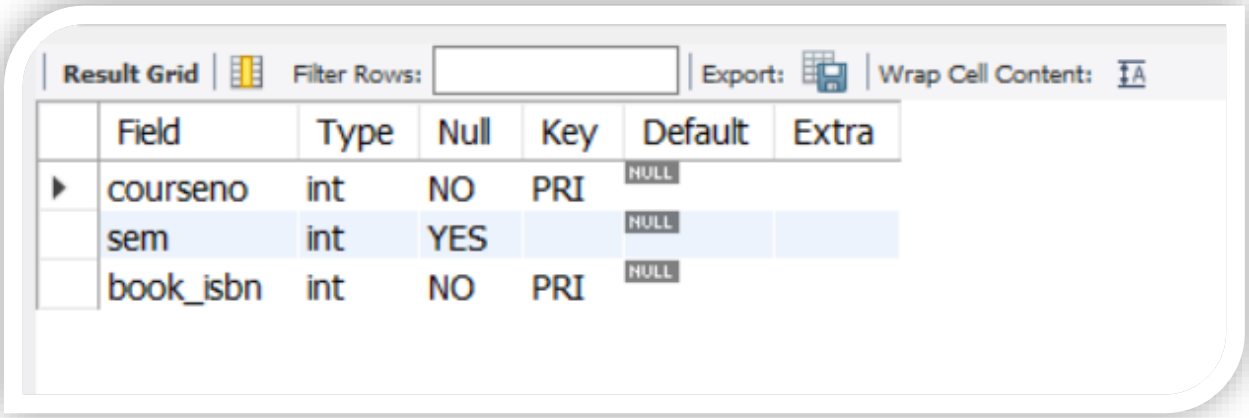


Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
book_isbn	int	NO	PRI	NULL	
book_title	varchar(20)	YES		NULL	
publisher	varchar(20)	YES		NULL	
author	varchar(20)	YES		NULL	

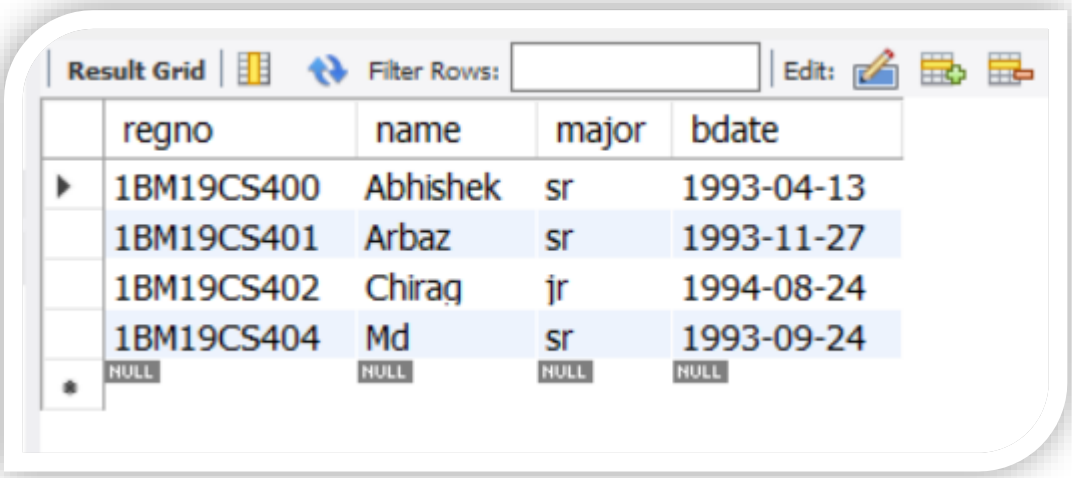
```
CREATE TABLE book_adoption(
 courseno INT,
 sem INT(3),
 book_isbn INT(5),
 PRIMARY KEY (courseno,book_isbn),
 FOREIGN KEY (courseno) REFERENCES course (courseno) on update cascade,
 FOREIGN KEY (book_isbn) REFERENCES text(book_isbn)on update cascade
);
```

desc book_adoption;



Field	Type	Null	Key	Default	Extra
courseno	int	NO	PRI	NULL	
sem	int	YES		NULL	
book_isbn	int	NO	PRI	NULL	

INSERT INTO student (regno,name,major,bdate) VALUES
('1BM19CS404','Md','sr','19930924'),
('1BM19CS401','Arbaz','sr','19931127'),
('1BM19CS400','Abhishek','sr','19930413'),
('1BM19CS402','Chirag','jr','19940824');
select * from student;



regno	name	major	bdate
1BM19CS400	Abhishek	sr	1993-04-13
1BM19CS401	Arbaz	sr	1993-11-27
1BM19CS402	Chirag	jr	1994-08-24
1BM19CS404	Md	sr	1993-09-24
NULL	NULL	NULL	NULL

INSERT INTO course VALUES (111,'OS','CSE'),
(112,'TFCS','CSE'),
(113,'ADA','ISE'),
(114,'DBMS','CSE'),
(115,'C','ECE');
select * from course;

Result Grid | Filter Rows: | Edit:

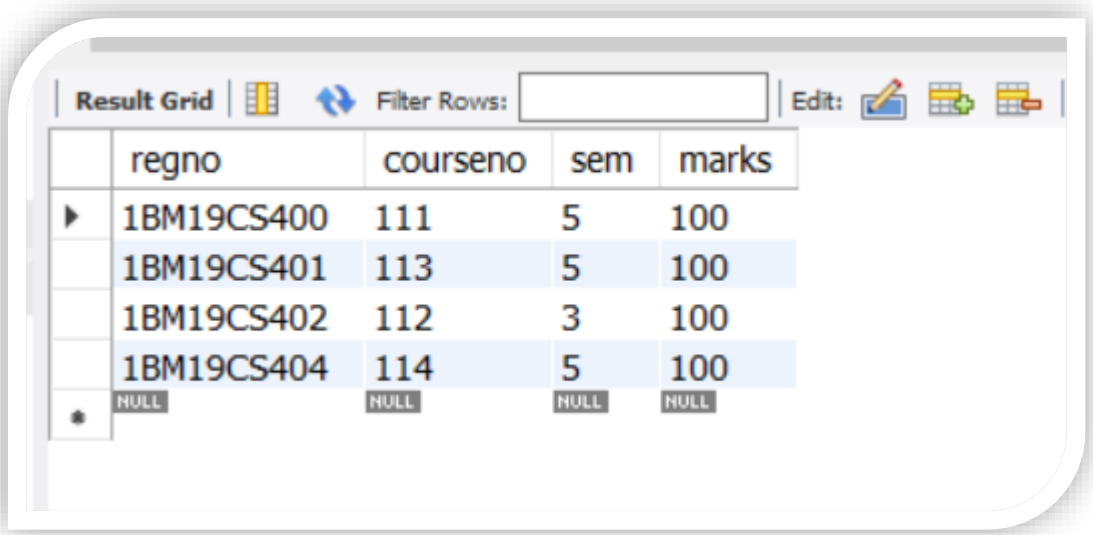
courseno	cname	dept
111	OS	CSE
112	TFCS	CSE
113	ADA	ISE
114	DBMS	CSE
115	C	ECE
NULL	NULL	NULL

```
INSERT INTO text (book_isbn,book_title,publisher,author)VALUES
 (10,'DATABASE SYSTEMS','PEARSON','SCHIELD'),
 (900,'OPERATING SYS','PEARSON','LELAND'),
 (901,'CIRCUITS','HALL INDIA','BOB'),
 (902,'SYSTEM SOFTWARE','PETERSON','JACOB'),
 (903,'SCHEDULING','PEARSON','PATIL'),
 (904,'DATABASE SYSTEMS','PEARSON','JACOB'),
 (905,'DATABASE MANAGER','PEARSON','BOB'),
 (906,'SIGNALS','HALL INDIA','SUMIT');
select * from text;
```

Result Grid | Filter Rows: | Edit: | Export/Import:

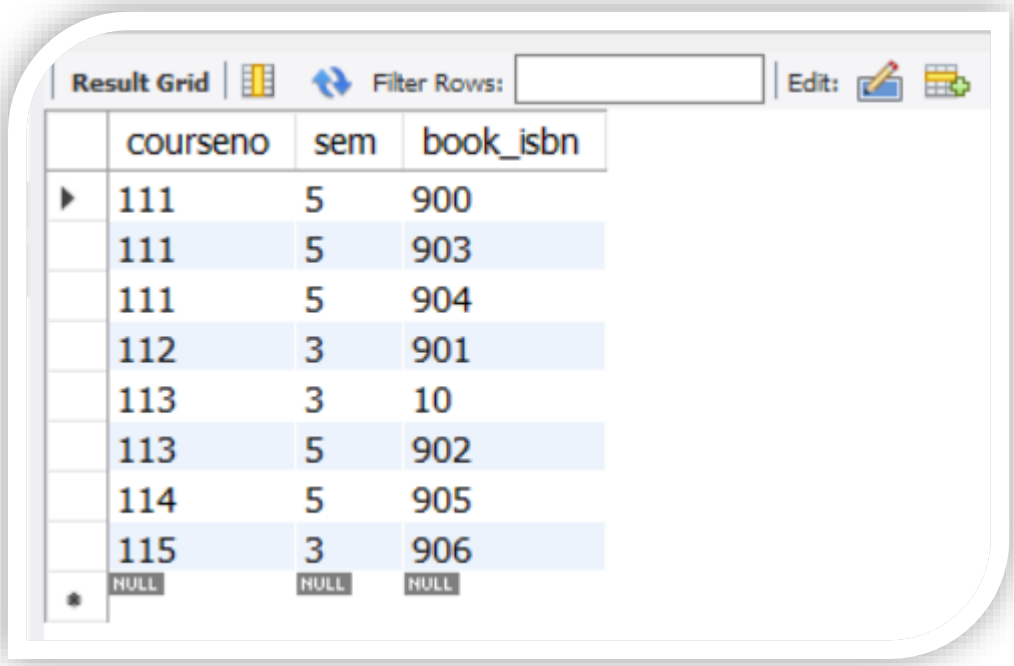
book_isbn	book_title	publisher	author
10	DATABASE SYSTEMS	PEARSON	SCHIELD
900	OPERATING SYS	PEARSON	LELAND
901	CIRCUITS	HALL INDIA	BOB
902	SYSTEM SOFTWARE	PETERSON	JACOB
903	SCHEDULING	PEARSON	PATIL
904	DATABASE SYSTEMS	PEARSON	JACOB
905	DATABASE MANAGER	PEARSON	BOB
906	SIGNALS	HALL INDIA	SUMIT
NULL	NULL	NULL	NULL

```
INSERT INTO enroll (regno,courseno,sem,marks) VALUES
 ('1BM19CS404',114,5,100),
 ('1BM19CS401',113,5,100),
 ('1BM19CS400',111,5,100),
 ('1BM19CS402',112,3,100);
select * from enroll;
```



regno	courseno	sem	marks
1BM19CS400	111	5	100
1BM19CS401	113	5	100
1BM19CS402	112	3	100
1BM19CS404	114	5	100
NULL	NULL	NULL	NULL

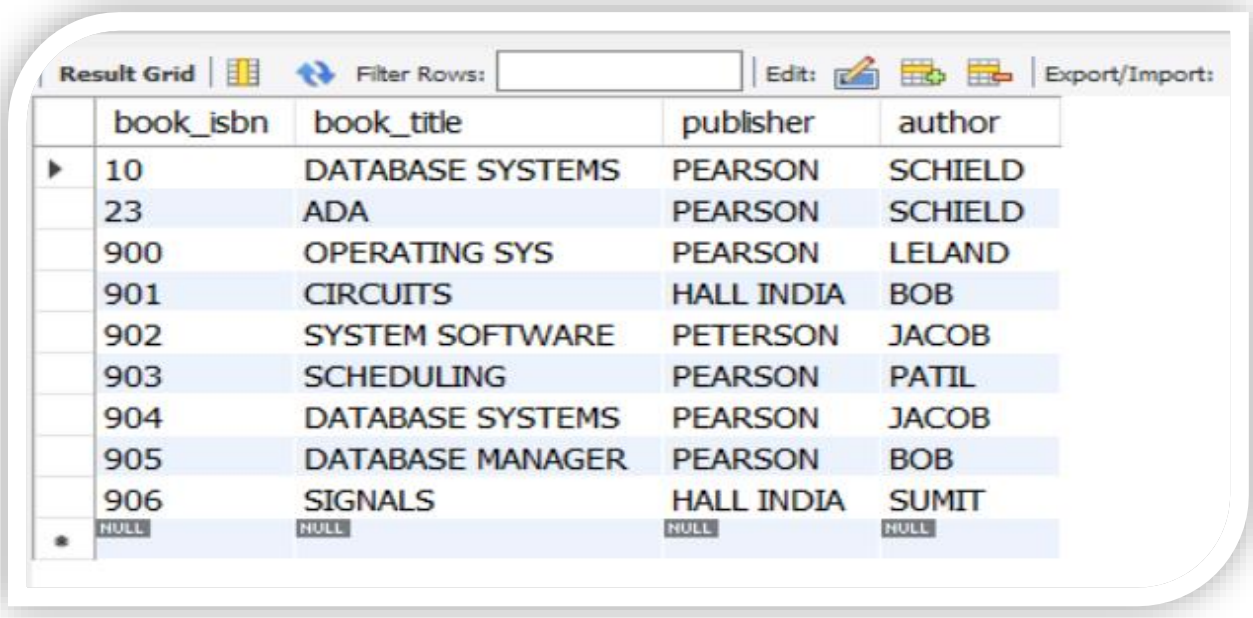
```
INSERT INTO book_adoption (courseno,sem,book_isbn) VALUES
(111,5,900),
(111,5,903),
(111,5,904),
(112,3,901),
(113,3,10),
(114,5,905),
(113,5,902),
(115,3,906);
select * from book_adoption;
commit;
```



courseno	sem	book_isbn
111	5	900
111	5	903
111	5	904
112	3	901
113	3	10
113	5	902
114	5	905
115	3	906
NULL	NULL	NULL

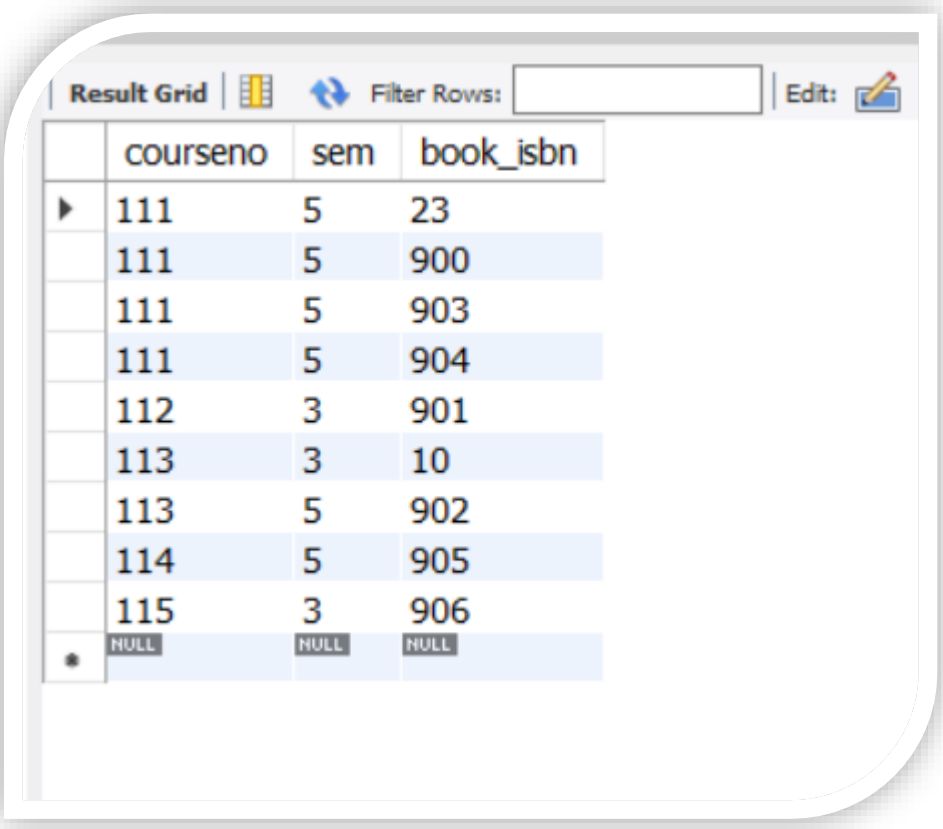
/*QUERIES:
01. Demonstrate how you add a new text book to the database
and make this book be adopted by some department */

```
INSERT INTO text (book_isbn, book_title, publisher,author) VALUES
(23,'ADA','PEARSON','SCHIELD');
SELECT * FROM TEXT;
```



book_isbn	book_title	publisher	author
10	DATABASE SYSTEMS	PEARSON	SCHIELD
23	ADA	PEARSON	SCHIELD
900	OPERATING SYS	PEARSON	LELAND
901	CIRCUITS	HALL INDIA	BOB
902	SYSTEM SOFTWARE	PETERSON	JACOB
903	SCHEDULING	PEARSON	PATIL
904	DATABASE SYSTEMS	PEARSON	JACOB
905	DATABASE MANAGER	PEARSON	BOB
906	SIGNALS	HALL INDIA	SUMIT
NULL	NULL	NULL	NULL

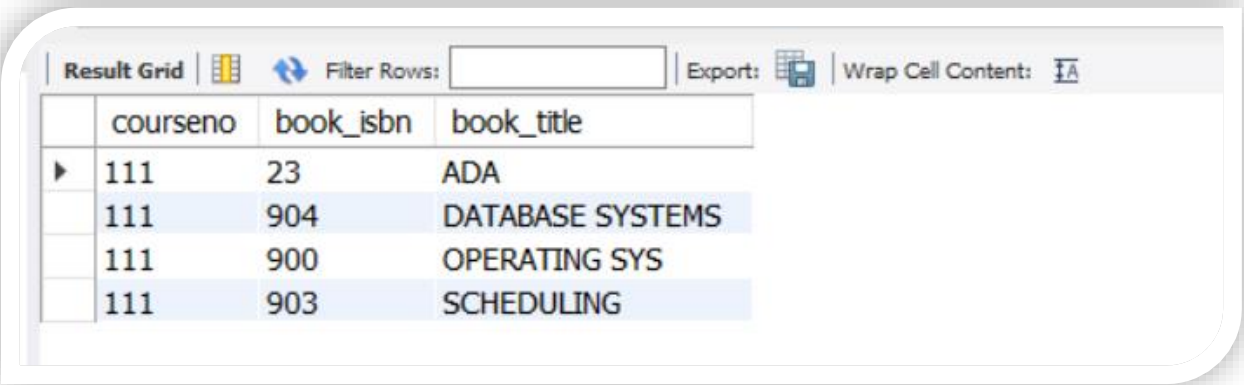
INSERT INTO book_adoption (courseno,sem,book_isbn) VALUES (111,5,23);
SELECT * FROM BOOK_ADOPTION;


Result Grid | Filter Rows: | Edit:

courseno	sem	book_isbn
111	5	23
111	5	900
111	5	903
111	5	904
112	3	901
113	3	10
113	5	902
114	5	905
115	3	906
NULL	NULL	NULL

/*02. Produce a list of text books [include course #, Book ISBN, Book title] in the alphabetical order for courses offered by the "CS " department that use more than two books. STUDENT ENROLLMENT DATABASE*/

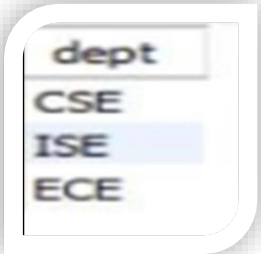
```
SELECT c.courseno,t.book_isbn,t.book_title
FROM course c,book_adoption ba,text t
WHERE c.courseno=ba.courseno
AND ba.book_isbn=t.book_isbn
AND c.dept='CSE'
AND 2<( SELECT COUNT(book_isbn)
        FROM book_adoption b
        WHERE c.courseno=b.courseno)
ORDER BY t.book_title;
```



courseno	book_isbn	book_title
111	23	ADA
111	904	DATABASE SYSTEMS
111	900	OPERATING SYS
111	903	SCHEDULING

/*03. List any department that has all its adopted books published by a specific publisher.
*/

```
SELECT DISTINCT c.dept FROM course c
WHERE c.dept IN ( SELECT c.dept FROM course
c,book_adoption b,text t WHERE c.courseno=b.courseno
AND t.book_isbn=b.book_isbn
AND t.publisher='PEARSON') AND c.dept
NOT IN (SELECT c.dept FROM course c,book_adoption b,text t
        WHERE c.courseno=b.courseno
    AND t.book_isbn=b.book_isbn AND t.publisher ='PEARSON');
```



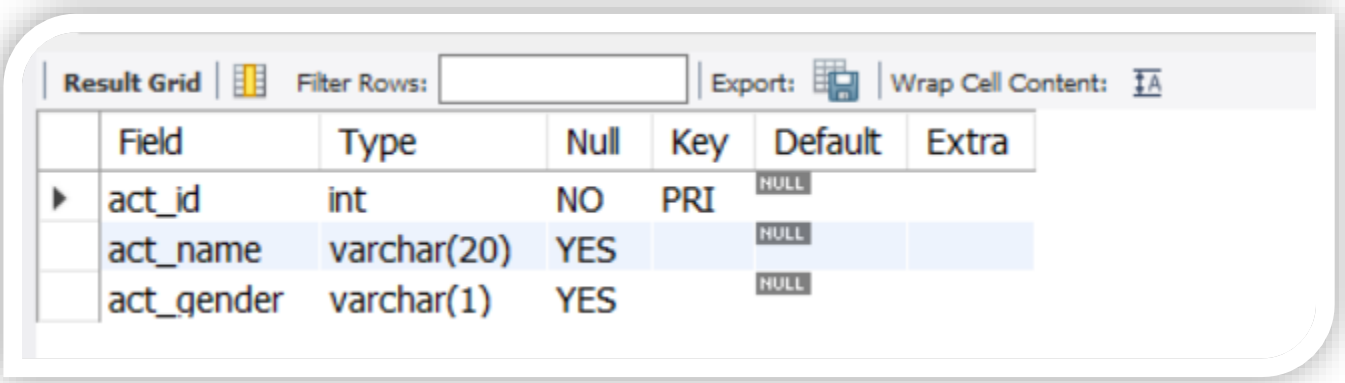
dept
CSE
ISE
ECE

PROGRAM 9: MOVIE DATABASE

Consider the schema for Movie Database:
ACTOR(Act_id, Act_Name, Act_Gender)
DIRECTOR(Dir_id, Dir_Name, Dir_Phone)
MOVIES(Mov_id, Mov_Title, Mov_Year, Mov_Lang, Dir_id)
MOVIE_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 JOIN operation).
4. Find the title of movies and number of stars for each movie that has at least one rating and find the highest
number of stars that movie received. Sort the result by movie title.
5. Update rating of all movies directed by 'Steven Spielberg' to 5.

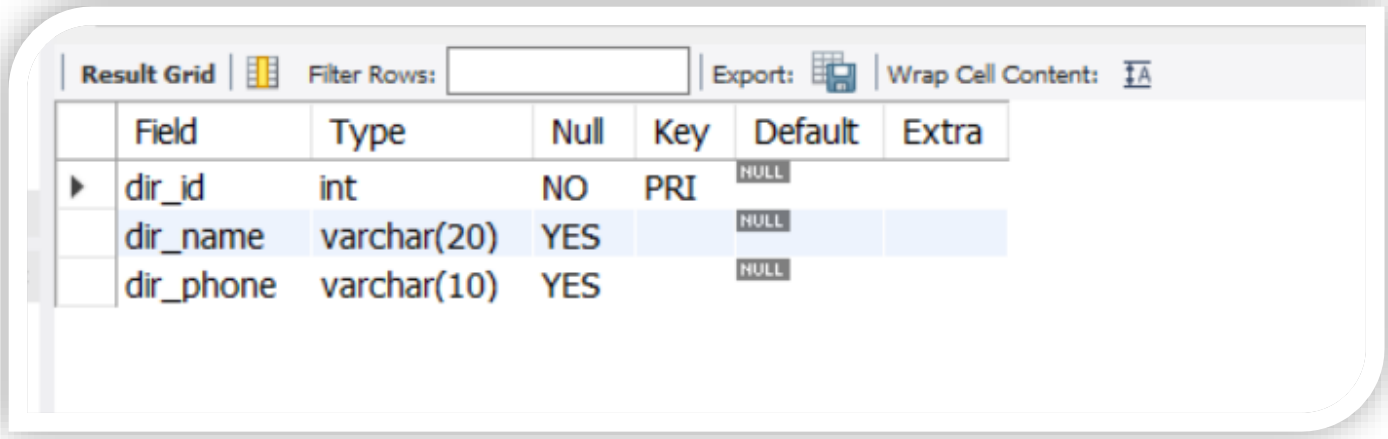
```
create database m_movie;
use m_movie;

create table actor(act_id int,
act_name varchar(20),
act_gender varchar(1),
primary key(act_id));
 desc actor;
```



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
act_id	int	NO	PRI	NULL	
act_name	varchar(20)	YES		NULL	
act_gender	varchar(1)	YES		NULL	

```
create table director(
dir_id int,
dir_name varchar(20),
dir_phone varchar(10),
primary key(dir_id));
desc director;
```

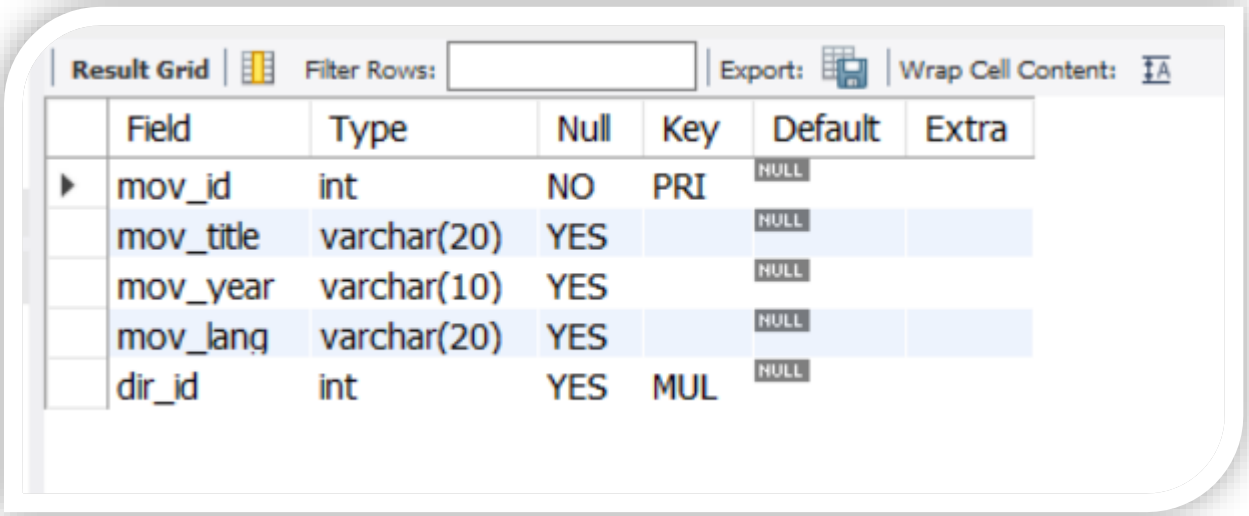


Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
dir_id	int	NO	PRI	NULL	
dir_name	varchar(20)	YES		NULL	
dir_phone	varchar(10)	YES		NULL	

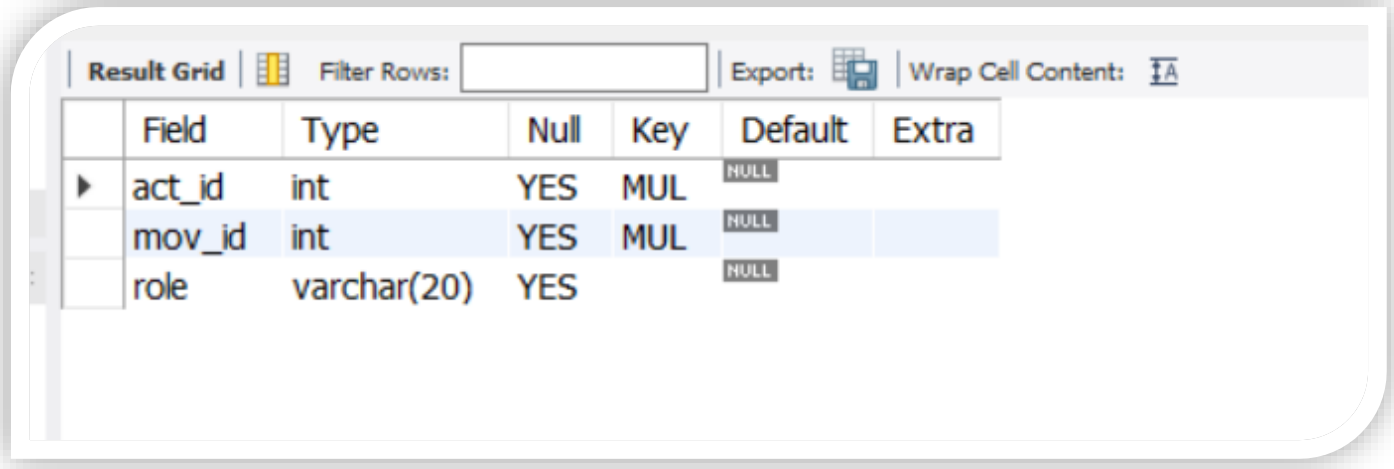
```
create table movies(
mov_id int,
mov_title varchar(20),
mov_year varchar(10),
mov_lang varchar(20),
dir_id int,
primary key(mov_id),
foreign key(dir_id) references director(dir_id));
```

desc movies;



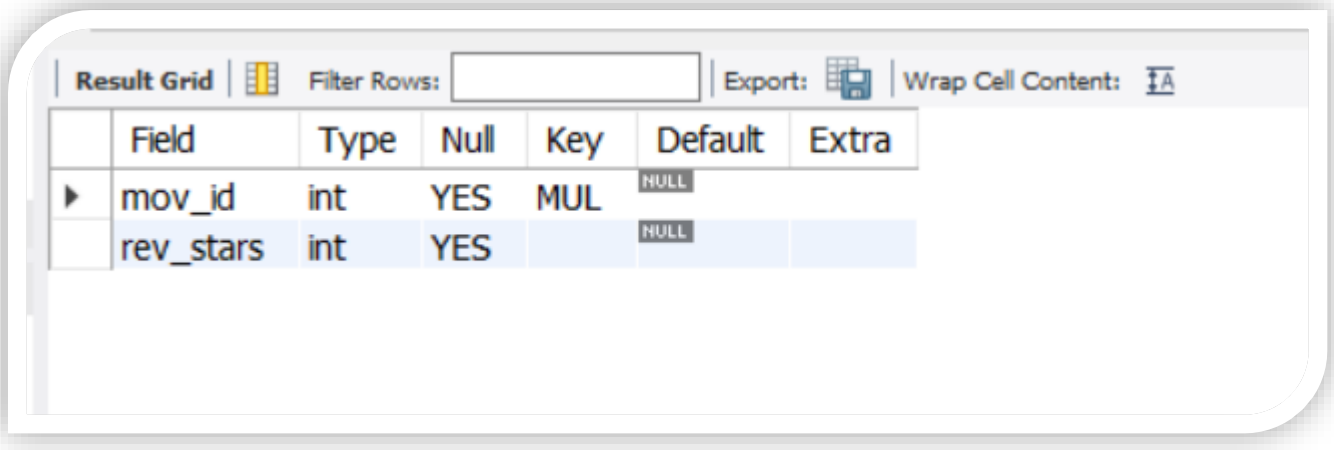
Field	Type	Null	Key	Default	Extra
mov_id	int	NO	PRI	NULL	
mov_title	varchar(20)	YES		NULL	
mov_year	varchar(10)	YES		NULL	
mov_lang	varchar(20)	YES		NULL	
dir_id	int	YES	MUL	NULL	

create table movie_cast(
act_id int,
mov_id int,
role varchar(20),
foreign key(act_id) references actor(act_id),
foreign key(mov_id) references movies(mov_id));
desc movie_cast;



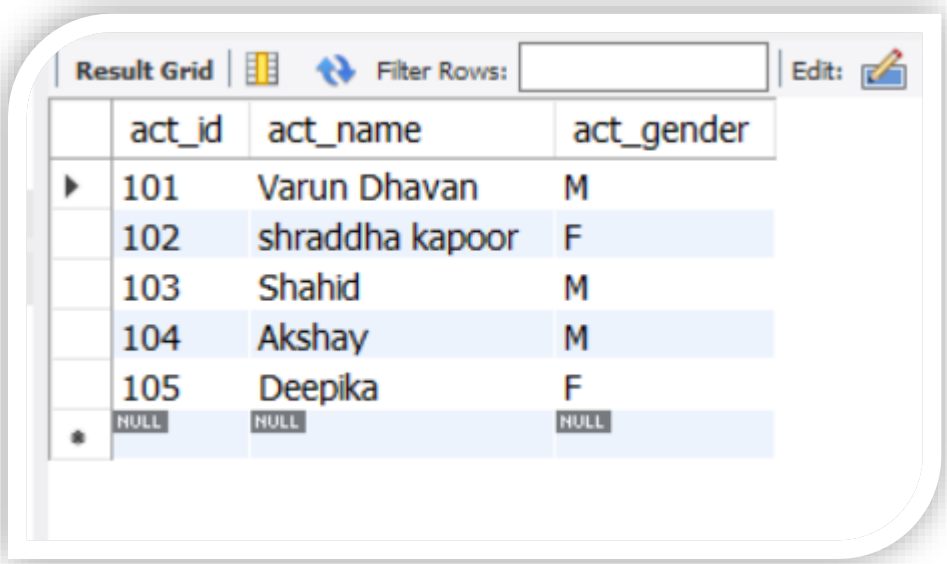
Field	Type	Null	Key	Default	Extra
act_id	int	YES	MUL	NULL	
mov_id	int	YES	MUL	NULL	
role	varchar(20)	YES		NULL	

```
create table rating(
mov_id int,
rev_stars int,
foreign key(mov_id) references movies(mov_id));
desc rating;
```



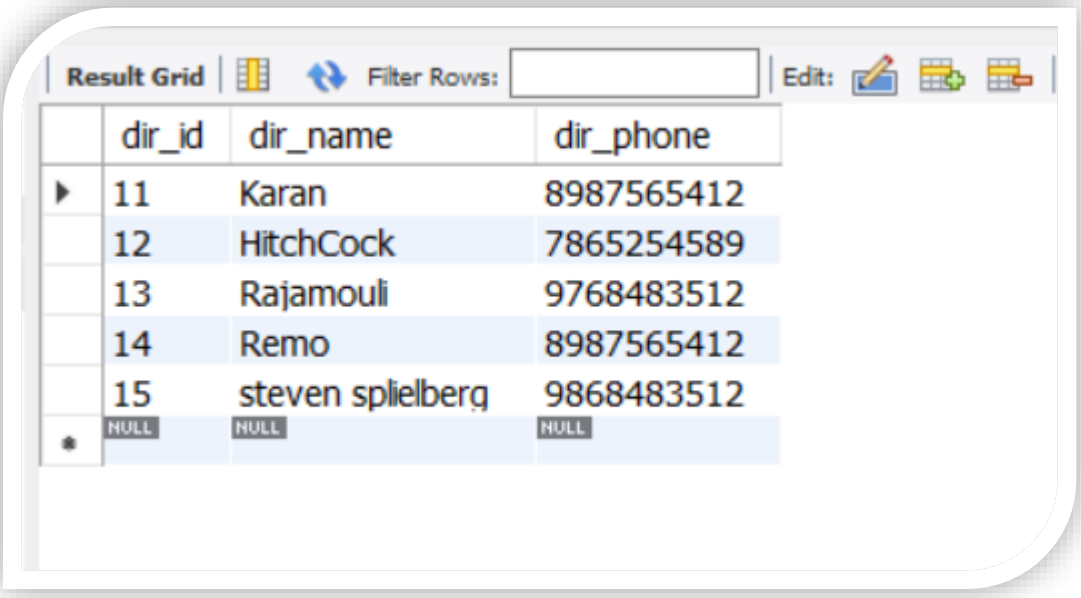
Field	Type	Null	Key	Default	Extra
mov_id	int	YES	MUL	NULL	
rev_stars	int	YES		NULL	

```
INSERT INTO actor (act_id, act_name, act_gender) VALUES ('101', 'Varun
Dhavan', 'M');
INSERT INTO actor (act_id, act_name, act_gender) VALUES ('102',
'shraddha kapoor', 'F');
INSERT INTO actor (act_id, act_name, act_gender) VALUES ('103',
'Shahid', 'M');
INSERT INTO actor (act_id, act_name, act_gender) VALUES ('104',
'Akshay', 'M');
INSERT INTO actor (act_id, act_name, act_gender) VALUES ('105',
'Deepika', 'F');
select * from actor;
```



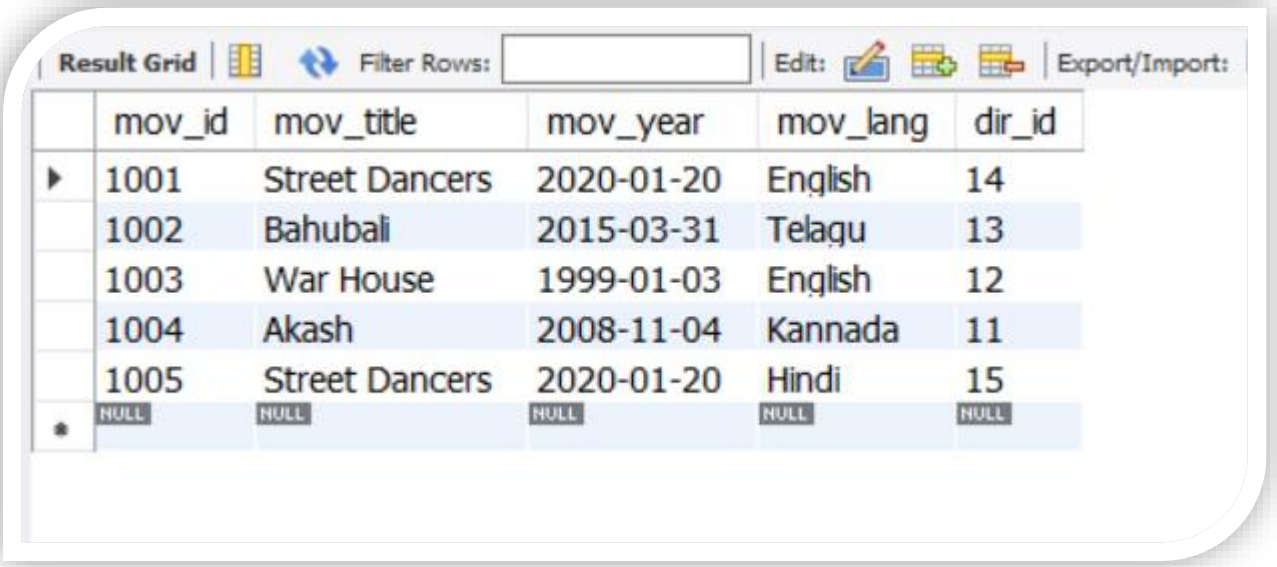
act_id	act_name	act_gender
101	Varun Dhavan	M
102	shraddha kapoor	F
103	Shahid	M
104	Akshay	M
105	Deepika	F
NULL	NULL	NULL

```
INSERT INTO director (dir_id, dir_name, dir_phone) VALUES ('11', 'Karan
', '8987565412');
INSERT INTO director (dir_id, dir_name, dir_phone) VALUES ('12',
'HitchCock', '7865254589');
INSERT INTO director (dir_id, dir_name, dir_phone) VALUES ('13',
'Rajamouli ', '9768483512');
INSERT INTO director (dir_id, dir_name, dir_phone) VALUES ('14', 'Remo
', '8987565412');
INSERT INTO director (dir_id, dir_name, dir_phone) VALUES ('15', 'steven
splielberg ', '9868483512');
select * from director;
```



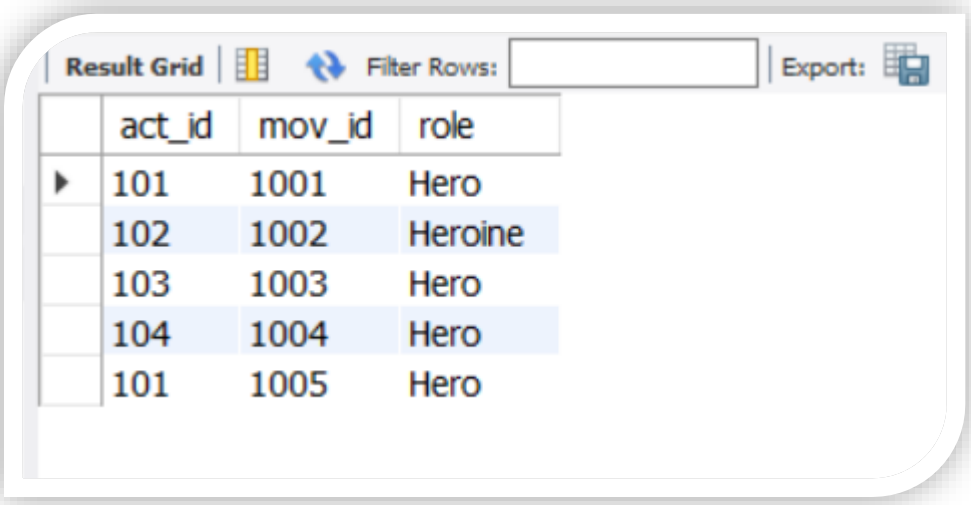
dir_id	dir_name	dir_phone
11	Karan	8987565412
12	HitchCock	7865254589
13	Rajamouli	9768483512
14	Remo	8987565412
15	steven splielberg	9868483512
NULL	NULL	NULL

```
INSERT INTO movies (mov_id, mov_title, mov_year, mov_lang, dir_id)
VALUES ('1001', 'Street Dancers', '2020-01-20', 'English', '14');
INSERT INTO movies (mov_id, mov_title, mov_year, mov_lang, dir_id)
VALUES ('1002', 'Bahubali', '2015-03-31', 'Telagu', '13');
INSERT INTO movies (mov_id, mov_title, mov_year, mov_lang, dir_id)
VALUES ('1003', 'War House', '1999-01-03', 'English', '12');
INSERT INTO movies (mov_id, mov_title, mov_year, mov_lang, dir_id)
VALUES ('1004', 'Akash', '2008-11-04', 'Kannada', '11');
INSERT INTO movies (mov_id, mov_title, mov_year, mov_lang, dir_id)
VALUES ('1005', 'Street Dancers', '2020-01-20', 'Hindi', '15');
select * from movies;
```



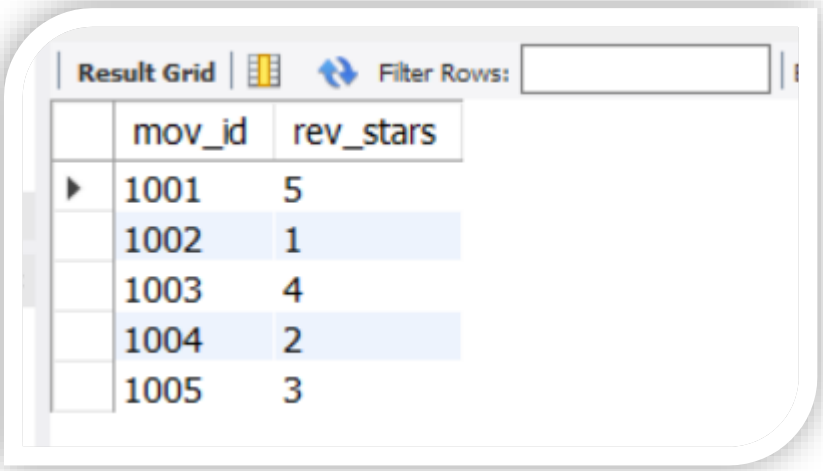
mov_id	mov_title	mov_year	mov_lang	dir_id
1001	Street Dancers	2020-01-20	English	14
1002	Bahubali	2015-03-31	Telagu	13
1003	War House	1999-01-03	English	12
1004	Akash	2008-11-04	Kannada	11
1005	Street Dancers	2020-01-20	Hindi	15
NULL	NULL	NULL	NULL	NULL

```
INSERT INTO movie_cast (act_id, mov_id, role) VALUES ('101', '1001', 'Hero');
INSERT INTO movie_cast (act_id, mov_id, role) VALUES ('102', '1002', 'Heroine');
INSERT INTO movie_cast (act_id, mov_id, role) VALUES ('103', '1003', 'Hero');
INSERT INTO movie_cast (act_id, mov_id, role) VALUES ('104', '1004', 'Hero');
INSERT INTO movie_cast (act_id, mov_id, role) VALUES ('101', '1005', 'Hero');
select * from movie_cast;
```



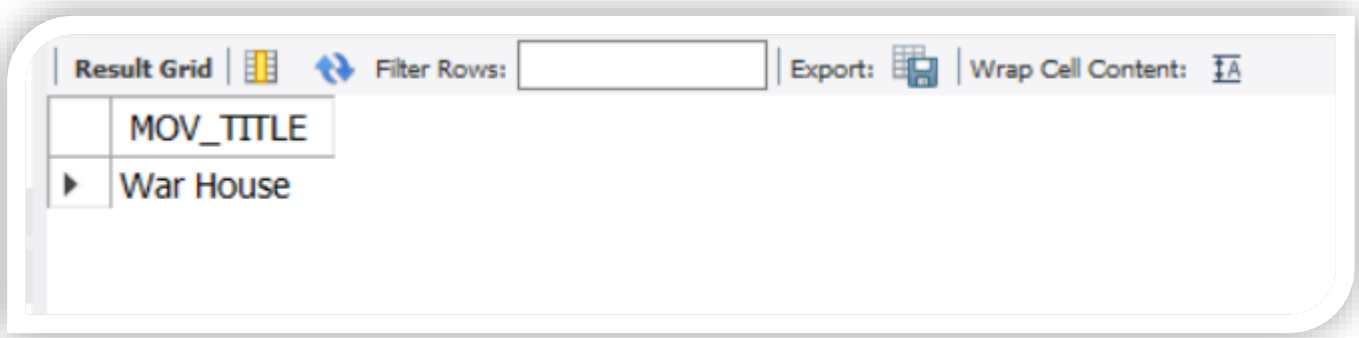
act_id	mov_id	role
101	1001	Hero
102	1002	Heroine
103	1003	Hero
104	1004	Hero
101	1005	Hero

```
INSERT INTO rating (mov_id, rev_stars) VALUES ('1001', '5');
INSERT INTO rating (mov_id, rev_stars) VALUES ('1002', '1');
INSERT INTO rating (mov_id, rev_stars) VALUES ('1003', '4');
INSERT INTO rating (mov_id, rev_stars) VALUES ('1004', '2');
INSERT INTO rating (mov_id, rev_stars) VALUES ('1005', '3');
select * from rating;
```



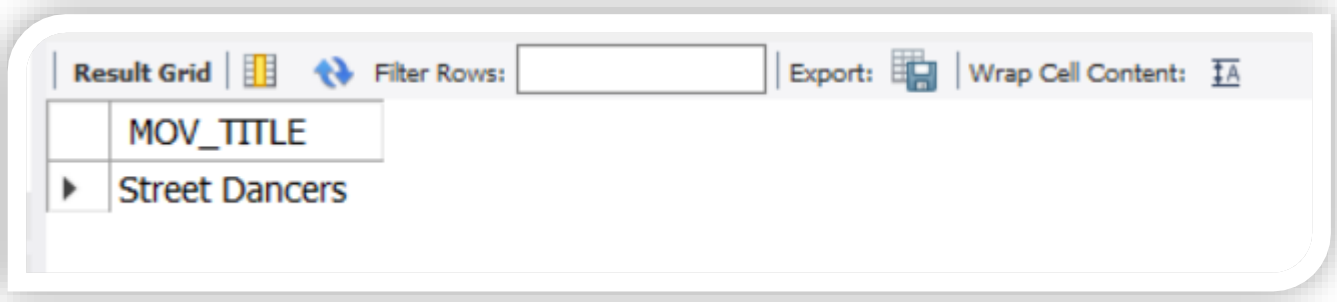
mov_id	rev_stars
1001	5
1002	1
1003	4
1004	2
1005	3

```
-- 1. List the titles of all movies directed by ‘Hitchcock’.
   SELECT MOV_TITLE FROM MOVIES WHERE DIR_ID IN (SELECT
DIR_ID FROM DIRECTOR WHERE DIR_NAME = 'HITCHCOCK');
```



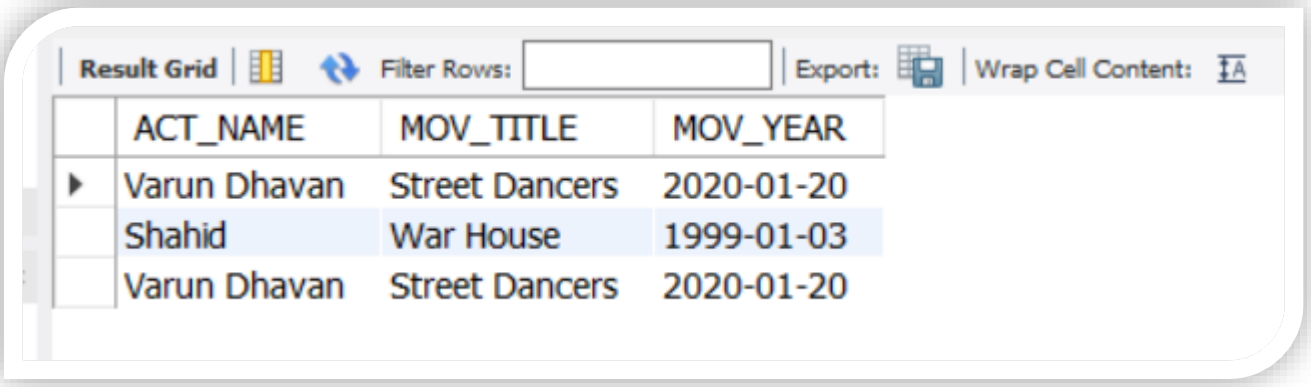
-- 2. Find the movie names where one or more actors acted in two or more movies.

SELECT MOV_TITLE FROM MOVIES M, MOVIE_CAST MV WHERE M.MOV_ID=MV.MOV_ID AND ACT_ID IN (SELECT ACT_ID FROM MOVIE_CAST GROUP BY ACT_ID HAVING count(ACT_ID)>=1) GROUP BY MOV_TITLE HAVING count(mov_title)>1;



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

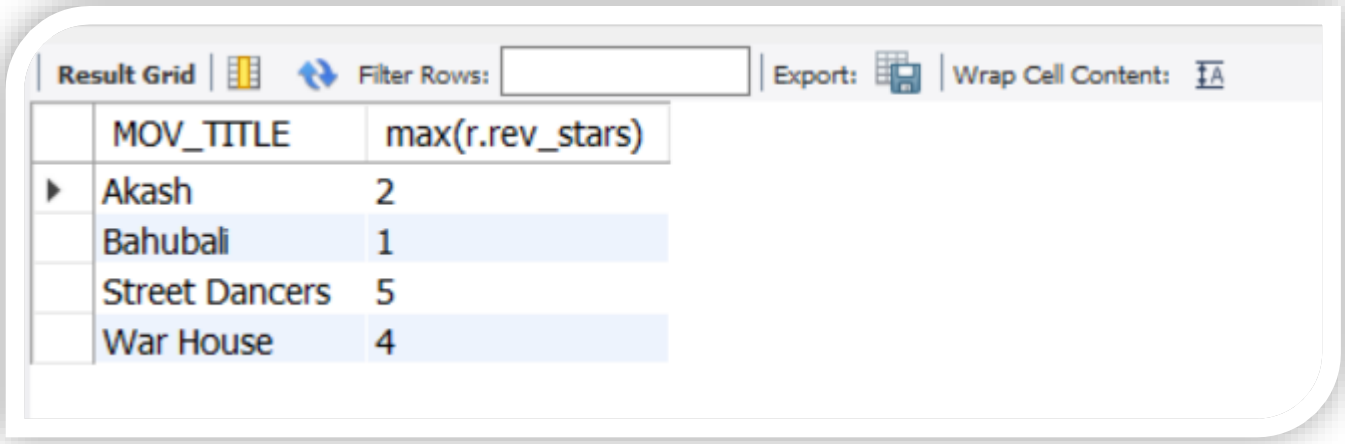
SELECT ACT_NAME, MOV_TITLE, MOV_YEAR FROM ACTOR A JOIN MOVIE_CAST C ON A.ACT_ID=C.ACT_ID JOIN MOVIES M ON C.MOV_ID=M.MOV_ID WHERE M.MOV_YEAR NOT BETWEEN 2000 AND 2015;



ACT_NAME	MOV_TITLE	MOV_YEAR
Varun Dhavan	Street Dancers	2020-01-20
Shahid	War House	1999-01-03
Varun Dhavan	Street Dancers	2020-01-20

-- 4. Find the title of movies and number of stars for each movie that has at least one rating and find the highest number of stars that movie received. Sort the result by movie title.

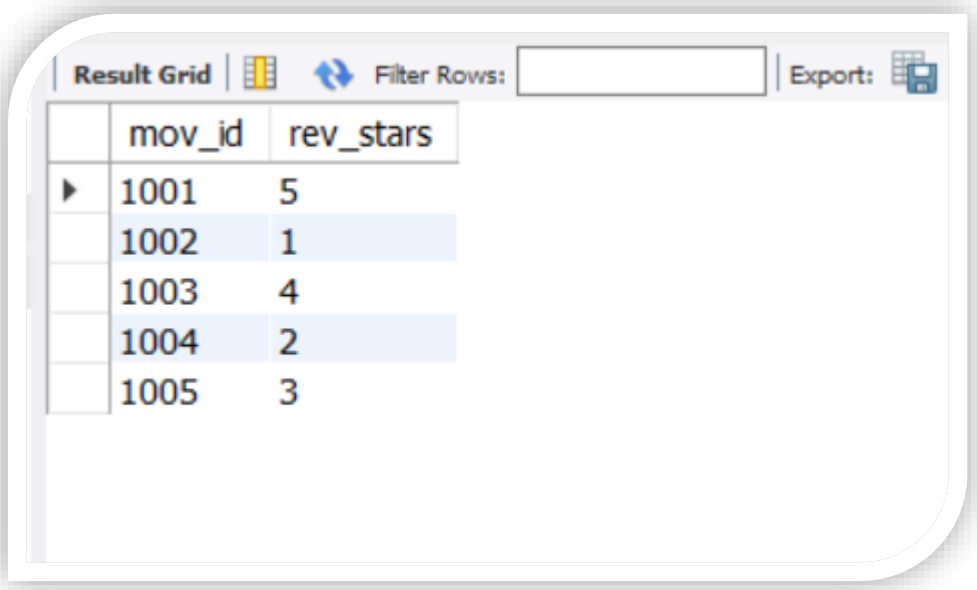
SELECT m.MOV_TITLE, max(r.rev_stars) FROM MOVIES m natural join RATING r where r.rev_stars>=1 GROUP BY m.MOV_TITLE ORDER BY m.MOV_TITLE;



MOV_TITLE	max(r.rev_stars)
Akash	2
Bahubali	1
Street Dancers	5
War House	4

-- 5. Update rating of all movies directed by ‘Steven Spielberg’ to 5

UPDATE RATING SET REV_STARS=5 WHERE MOV_ID IN (SELECT MOV_ID FROM MOVIES WHERE DIR_ID IN (SELECT DIR_ID FROM DIRECTOR WHERE DIR_NAME = 'STEVEN SPIELBERG'));



mov_id	rev_stars
1001	5
1002	1
1003	4
1004	2
1005	3

commit;

PROGRAM 10:COLLEGE DATABASE

Consider the schema for College Database:
STUDENT(USN, SName, Address, Phone, Gender)
SEMSEC(SSID, Sem, Sec)
CLASS(USN, SSID)
SUBJECT(Subcode, Title, Sem, Credits)
IAMARKS(USN, Subcode, SSID, Test1, Test2, Test3, FinalIA)
Write SQL queries to

1. List all the student details studying in fourth semester 'C'section.
2. Compute the total number of male and female students in each semester and in each section.
3. Create a view of Test1 marks of student USN '1BI15CS101' in all subjects.
4. Calculate the FinalIA (average of best two test marks) and update the corresponding table for all students.
5. Categorize students based on the following criterion:
If FinalIA = 17 to 20 then CAT = 'Outstanding'
If FinalIA = 12 to 16 then CAT = 'Average'
If FinalIA< 12 then CAT = 'Weak'
Give these details only for 8th semester A, B, and C section students.

```
drop database college;
create database college;
use college;
CREATE TABLE STUDENT (
USN VARCHAR (10) PRIMARY KEY,
SNAME VARCHAR (25),
ADDRESS VARCHAR (25),
PHONE INT ,
GENDER CHAR (1));
```

```
CREATE TABLE SEMSEC (
SSID VARCHAR (5) PRIMARY KEY,
SEM INT ,
SEC CHAR (1));

CREATE TABLE CLASS (
USN VARCHAR (10),
SSID VARCHAR (5),
PRIMARY KEY (USN, SSID),
FOREIGN KEY (USN) REFERENCES STUDENT (USN),
FOREIGN KEY (SSID) REFERENCES SEMSEC (SSID));

CREATE TABLE SUBJECT (
SUBCODE VARCHAR (8),
TITLE VARCHAR (20),
SEM INT,
CREDITS INT,
PRIMARY KEY (SUBCODE));

CREATE TABLE IAMARKS (
USN VARCHAR (10),
SUBCODE VARCHAR (8),
SSID VARCHAR (5),
TEST1 INT,
TEST2 INT,
TEST3 INT,
FINALIA INT,
PRIMARY KEY (USN, SUBCODE, SSID),
FOREIGN KEY (USN) REFERENCES STUDENT (USN),
FOREIGN KEY (SUBCODE) REFERENCES SUBJECT (SUBCODE),
FOREIGN KEY (SSID) REFERENCES SEMSEC (SSID));


INSERT INTO STUDENT VALUES
('1RN13CS020','AKSHAY','BELAGAVI', 1232654578,'M');
INSERT INTO STUDENT VALUES
```

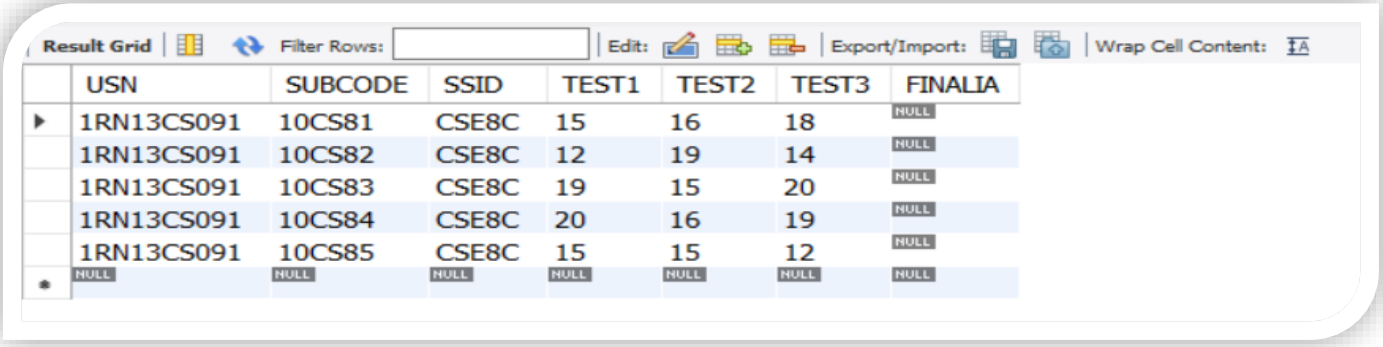
('1RN13CS062','SANDHYA','BENGALURU', 1232654578,'F');
INSERT INTO STUDENT VALUES
('1RN13CS091','TEESHA','BENGALURU', 1232654578,'F');
INSERT INTO STUDENT VALUES
('1RN13CS066','SUPRIYA','MANGALURU', 1232654578,'F');
INSERT INTO STUDENT VALUES
('1RN14CS010','ABHAY','BENGALURU', 1232654578,'M');
INSERT INTO STUDENT VALUES
('1RN14CS032','BHASKAR','BENGALURU', 1232654578,'M');
INSERT INTO STUDENT VALUES
('1RN14CS025','ASMI','BENGALURU', 1232654578,'F');
INSERT INTO STUDENT VALUES ('1RN15CS011','AJAY','TUMKUR',
1232654578,'M');
INSERT INTO STUDENT VALUES
('1RN15CS029','CHITRA','DAVANGERE', 1232654578,'F');
INSERT INTO STUDENT VALUES ('1RN15CS045','JEEVA','BELLARY',
1232654578,'M');
INSERT INTO STUDENT VALUES
('1RN15CS091','SANTOSH','MANGALURU', 1232654578,'M');
INSERT INTO STUDENT VALUES
('1RN16CS045','ISMAIL','KALBURGI', 1232654578,'M');
INSERT INTO STUDENT VALUES
('1RN16CS088','SAMEERA','SHIMOGA', 1232654578,'F');
INSERT INTO STUDENT VALUES
('1RN16CS122','VINAYAKA','CHIKAMAGALUR', 1232654578,'M');

INSERT INTO SEMSEC VALUES ('CSE8A', 8,'A');
INSERT INTO SEMSEC VALUES ('CSE8B', 8,'B');
INSERT INTO SEMSEC VALUES ('CSE8C', 8,'C');
INSERT INTO SEMSEC VALUES ('CSE7A', 7,'A');
INSERT INTO SEMSEC VALUES ('CSE7B', 7,'B');
INSERT INTO SEMSEC VALUES ('CSE7C', 7,'C');
INSERT INTO SEMSEC VALUES ('CSE6A', 6,'A');
INSERT INTO SEMSEC VALUES ('CSE6B', 6,'B');
INSERT INTO SEMSEC VALUES ('CSE6C', 6,'C');

```
INSERT INTO SEMSEC VALUES ('CSE5A', 5,'A');
INSERT INTO SEMSEC VALUES ('CSE5B', 5,'B');
INSERT INTO SEMSEC VALUES ('CSE5C', 5,'C');
INSERT INTO SEMSEC VALUES ('CSE4A', 4,'A');
INSERT INTO SEMSEC VALUES ('CSE4B', 4,'B');
INSERT INTO SEMSEC VALUES ('CSE4C', 4,'C');
INSERT INTO SEMSEC VALUES ('CSE3A', 3,'A');
INSERT INTO SEMSEC VALUES ('CSE3B', 3,'B');
INSERT INTO SEMSEC VALUES ('CSE3C', 3,'C');
INSERT INTO SEMSEC VALUES ('CSE2A', 2,'A');
INSERT INTO SEMSEC VALUES ('CSE2B', 2,'B');
INSERT INTO SEMSEC VALUES ('CSE2C', 2,'C');
INSERT INTO SEMSEC VALUES ('CSE1A', 1,'A');
INSERT INTO SEMSEC VALUES ('CSE1B', 1,'B');
INSERT INTO SEMSEC VALUES ('CSE1C', 1,'C');

INSERT INTO CLASS VALUES ('1RN13CS020','CSE8A');
INSERT INTO CLASS VALUES ('1RN13CS062','CSE8A');
INSERT INTO CLASS VALUES ('1RN13CS066','CSE8B');
INSERT INTO CLASS VALUES ('1RN13CS091','CSE8C');
INSERT INTO CLASS VALUES ('1RN14CS010','CSE7A');
INSERT INTO CLASS VALUES ('1RN14CS025','CSE7A');
INSERT INTO CLASS VALUES ('1RN14CS032','CSE7A');
INSERT INTO CLASS VALUES ('1RN15CS011','CSE4A');
INSERT INTO CLASS VALUES ('1RN15CS029','CSE4A');


INSERT INTO SUBJECT VALUES ('10CS81','ACA', 8, 4);
INSERT INTO SUBJECT VALUES ('10CS82','SSM', 8, 4);
INSERT INTO SUBJECT VALUES ('10CS83','NM', 8, 4);
INSERT INTO SUBJECT VALUES ('10CS84','CC', 8, 4);
INSERT INTO SUBJECT VALUES ('10CS85','PW', 8, 4);
INSERT INTO SUBJECT VALUES ('10CS71','OOAD', 7, 4);
INSERT INTO SUBJECT VALUES ('10CS72','ECS', 7, 4);
INSERT INTO SUBJECT VALUES ('10CS73','PTW', 7, 4);
INSERT INTO SUBJECT VALUES ('10CS74','DWDM', 7, 4);
```

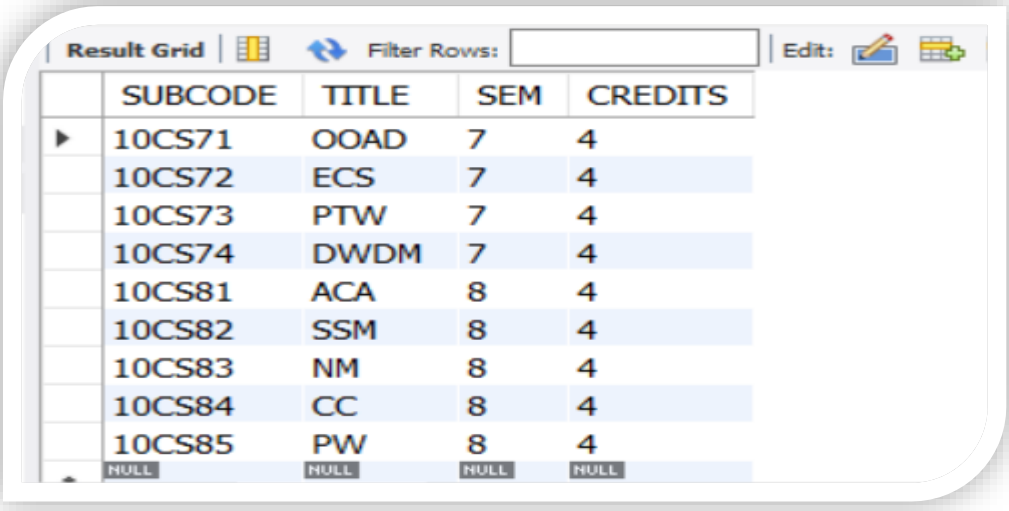
INSERT INTO IAMARKS (USN, SUBCODE, SSID, TEST1, TEST2, TEST3) VALUES ('1RN13CS091','10CS81','CSE8C', 15, 16, 18);
INSERT INTO IAMARKS (USN, SUBCODE, SSID, TEST1, TEST2, TEST3) VALUES ('1RN13CS091','10CS82','CSE8C', 12, 19, 14);
INSERT INTO IAMARKS (USN, SUBCODE, SSID, TEST1, TEST2, TEST3) VALUES ('1RN13CS091','10CS83','CSE8C', 19, 15, 20);
INSERT INTO IAMARKS (USN, SUBCODE, SSID, TEST1, TEST2, TEST3) VALUES ('1RN13CS091','10CS84','CSE8C', 20, 16, 19);
INSERT INTO IAMARKS (USN, SUBCODE, SSID, TEST1, TEST2, TEST3) VALUES ('1RN13CS091','10CS85','CSE8C', 15, 15, 12);

select * from iamarks;


Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

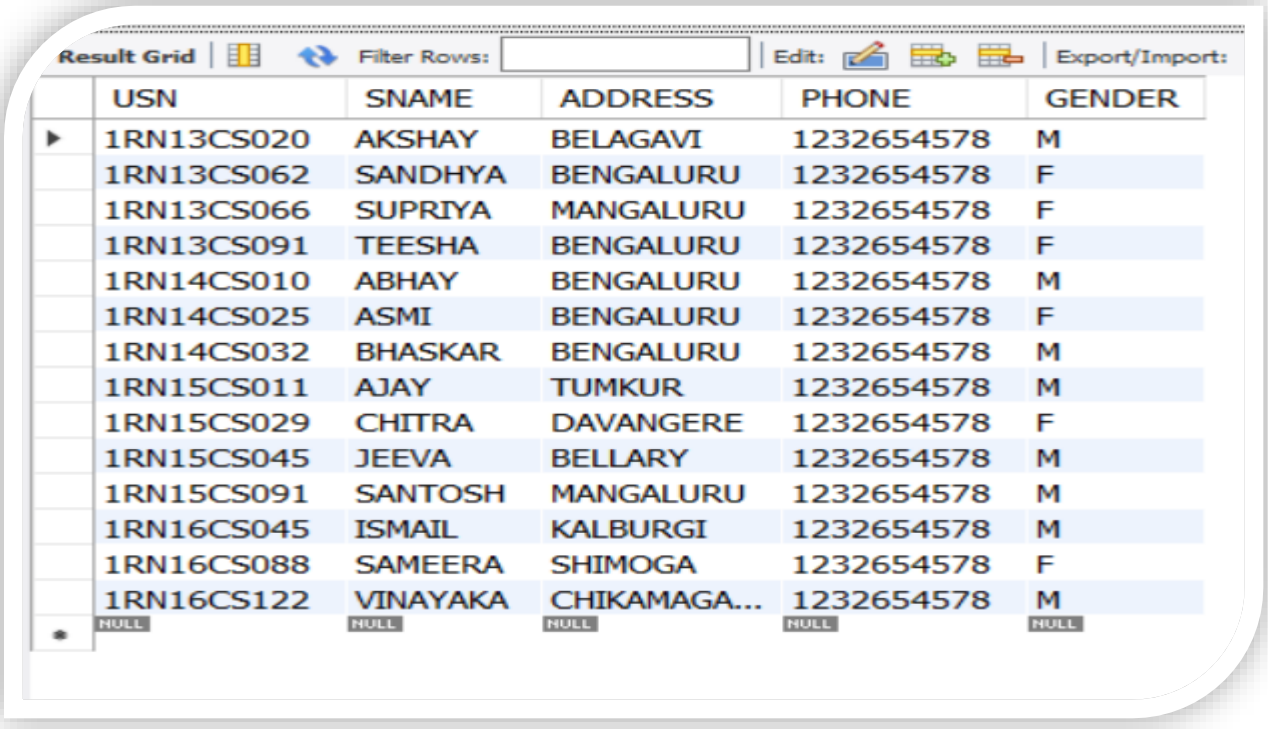
USN	SUBCODE	SSID	TEST1	TEST2	TEST3	FINALIA
1RN13CS091	10CS81	CSE8C	15	16	18	NULL
1RN13CS091	10CS82	CSE8C	12	19	14	NULL
1RN13CS091	10CS83	CSE8C	19	15	20	NULL
1RN13CS091	10CS84	CSE8C	20	16	19	NULL
1RN13CS091	10CS85	CSE8C	15	15	12	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL

select * from subject;


Result Grid | Filter Rows: | Edit:

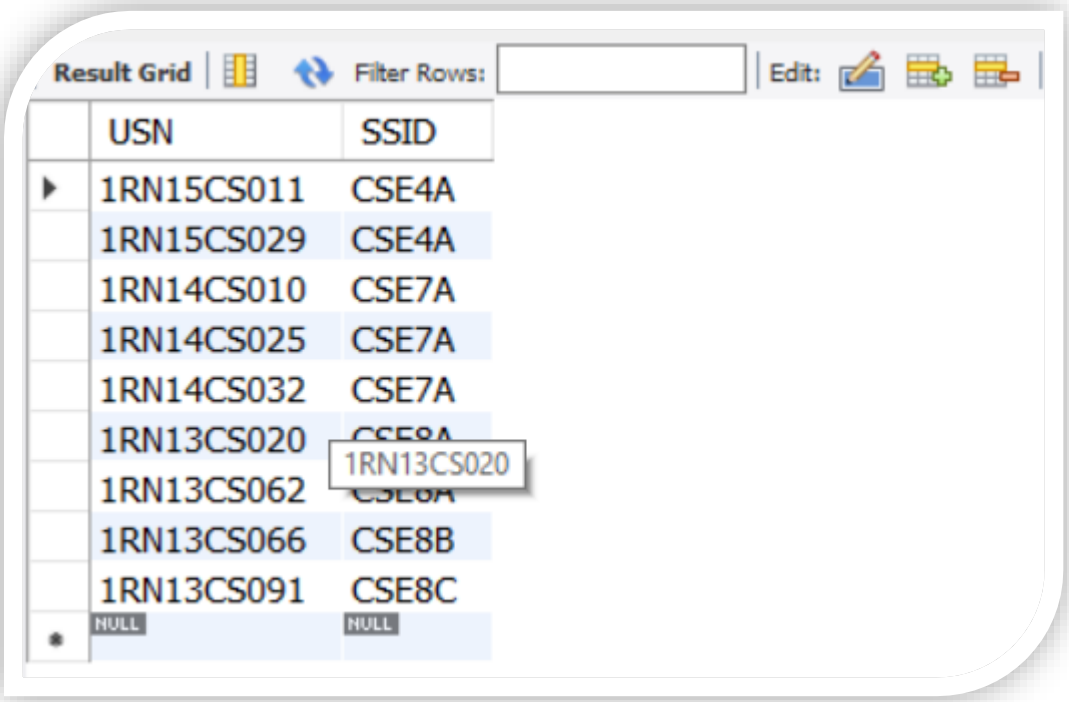
SUBCODE	TITLE	SEM	CREDITS
10CS71	OOAD	7	4
10CS72	ECS	7	4
10CS73	PTW	7	4
10CS74	DWDM	7	4
10CS81	ACA	8	4
10CS82	SSM	8	4
10CS83	NM	8	4
10CS84	CC	8	4
10CS85	PW	8	4
NULL	NULL	NULL	NULL

select * from student;



USN	SNAME	ADDRESS	PHONE	GENDER
1RN13CS020	AKSHAY	BELAGAVI	1232654578	M
1RN13CS062	SANDHYA	BENGALURU	1232654578	F
1RN13CS066	SUPRIYA	MANGALURU	1232654578	F
1RN13CS091	TEESHA	BENGALURU	1232654578	F
1RN14CS010	ABHAY	BENGALURU	1232654578	M
1RN14CS025	ASMI	BENGALURU	1232654578	F
1RN14CS032	BHASKAR	BENGALURU	1232654578	M
1RN15CS011	AJAY	TUMKUR	1232654578	M
1RN15CS029	CHITRA	DAVANGERE	1232654578	F
1RN15CS045	JEEVA	BELLARY	1232654578	M
1RN15CS091	SANTOSH	MANGALURU	1232654578	M
1RN16CS045	ISMAIL	KALBURGI	1232654578	M
1RN16CS088	SAMEERA	SHIMOGA	1232654578	F
1RN16CS122	VINAYAKA	CHIKAMAGA...	1232654578	M
NULL	NULL	NULL	NULL	NULL

select * from class;



USN	SSID
1RN15CS011	CSE4A
1RN15CS029	CSE4A
1RN14CS010	CSE7A
1RN14CS025	CSE7A
1RN14CS032	CSE7A
1RN13CS020	CSE8A
1RN13CS062	CSE8A
1RN13CS066	CSE8B
1RN13CS091	CSE8C
NULL	NULL

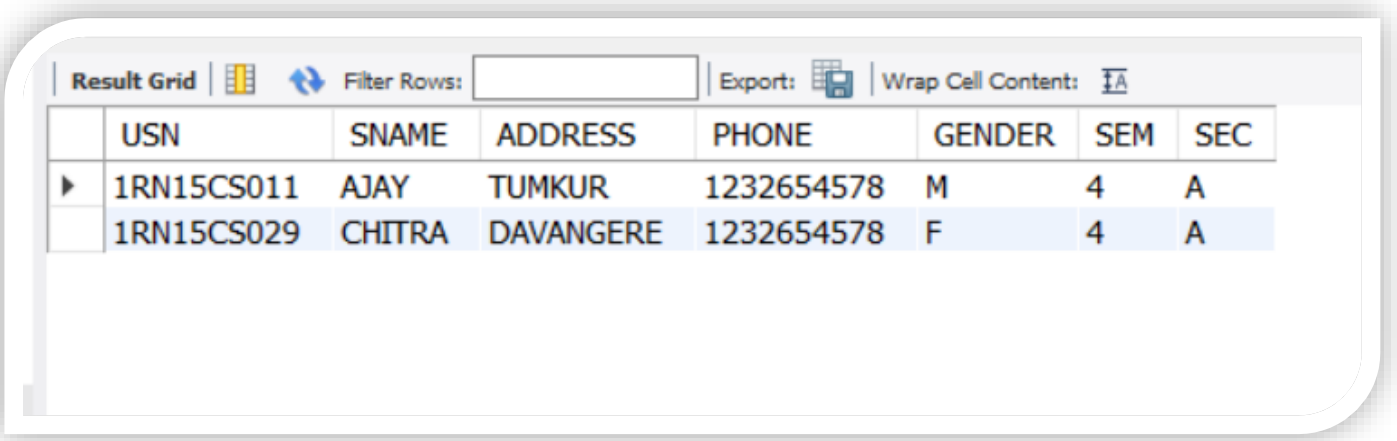
select * from semsec;

Result Grid | Filter Rows: | Edit:

SSID	SEM	SEC
CSE1A	1	A
CSE1B	1	B
CSE1C	1	C
CSE2A	2	A
CSE2B	2	B
CSE2C	2	C
CSE3A	3	A
CSE3B	3	B
CSE3C	3	C
CSE4A	4	A
CSE4B	4	B
CSE4C	4	C
CSE5A	5	A
CSE5B	5	B
CSE5C	5	C
CSE6A	6	A
CSE6B	6	B
CSE6C	6	C
CSE7A	7	A
CSE7B	7	B
CSE7C	7	C
CSE8A	8	A
CSE8B	8	B
CSE8C	8	C
NULL	NULL	NULL

/* List all the student details studying in fourth semester ‘A’ section */

```
SELECT S.*, SS.SEM, SS.SEC
FROM STUDENT S, SEMSEC SS, CLASS C
WHERE S.USN = C.USN AND SS.SSID = C.SSID AND SS.SEM = 4 AND
SS.SEC='A';
```

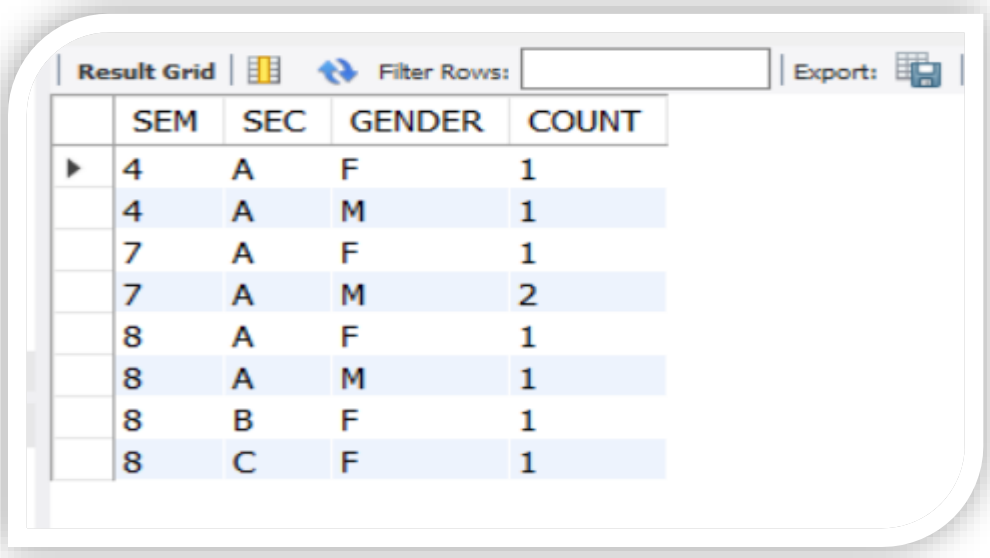


Result Grid | Filter Rows: | Export: | Wrap Cell Content:

USN	SNAME	ADDRESS	PHONE	GENDER	SEM	SEC
1RN15CS011	AJAY	TUMKUR	1232654578	M	4	A
1RN15CS029	CHITRA	DAVANGERE	1232654578	F	4	A

/* Compute the total number of male and female students in each semester and in each section */

```
SELECT SS.SEM, SS.SEC, S.GENDER, COUNT(S.GENDER) AS COUNT
FROM STUDENT S, SEMSEC SS, CLASS C
WHERE S.USN = C.USN AND
SS.SSID = C.SSID
GROUP BY SS.SEM, SS.SEC, S.GENDER
ORDER BY SEM;
```

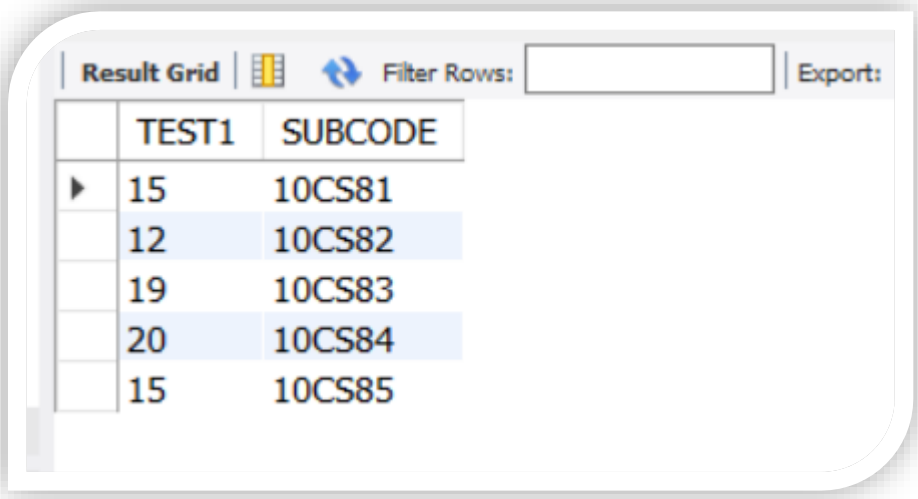


Result Grid | Filter Rows: | Export:

SEM	SEC	GENDER	COUNT
4	A	F	1
4	A	M	1
7	A	F	1
7	A	M	2
8	A	F	1
8	A	M	1
8	B	F	1
8	C	F	1

/* Create a view of Test1 marks of student USN ‘1BI15CS101’ in all subjects */

```
CREATE VIEW STU_TEST1_MARKS_VIEW
AS
SELECT TEST1, SUBCODE
FROM IAMARKS
WHERE USN = '1RN13CS091';
select * from STU_TEST1_MARKS_VIEW;
```

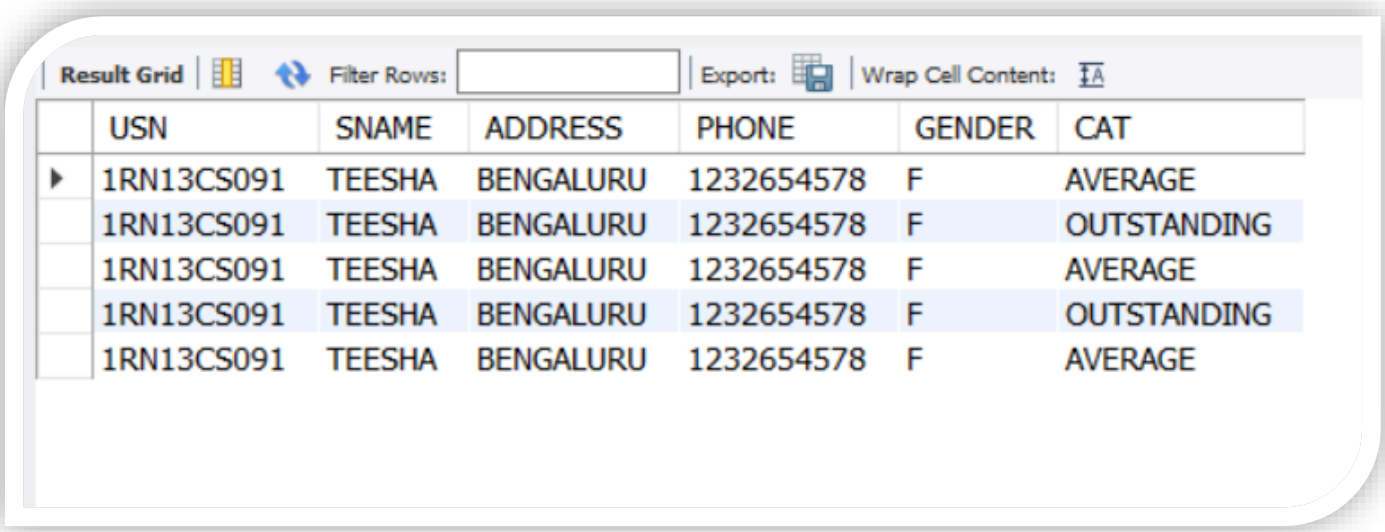


Result Grid | Filter Rows: | Export:

TEST1	SUBCODE
15	10CS81
12	10CS82
19	10CS83
20	10CS84
15	10CS85

/* Categorize students based on the following criterion:
If FinalIA = 17 to 20 then CAT = ‘Outstanding’
If FinalIA = 12 to 16 then CAT = ‘Average’
If FinalIA< 12 then CAT = ‘Weak’
Give these details only for 8th semester A, B, and C section students */

```
update iamarks set finalia=(test1+test2+test2)/3;

SELECT S.USN,S.SNAME,S.ADDRESS,S.PHONE,S.GENDER,
(CASE
WHEN IA.FINALIA BETWEEN 17 AND 20 THEN 'OUTSTANDING'
WHEN IA.FINALIA BETWEEN 12 AND 16 THEN 'AVERAGE'
ELSE 'WEAK'
END) AS CAT
FROM STUDENT S, SEMSEC SS, IAMARKS IA, SUBJECT SUB
WHERE S.USN = IA.USN AND
SS.SSID = IA.SSID AND
SUB.SUBCODE = IA.SUBCODE AND
SUB.SEM = 8;
```


Result Grid | Filter Rows: | Export: | Wrap Cell Content:

USN	SNAME	ADDRESS	PHONE	GENDER	CAT
1RN13CS091	TEESHA	BENGALURU	1232654578	F	AVERAGE
1RN13CS091	TEESHA	BENGALURU	1232654578	F	OUTSTANDING
1RN13CS091	TEESHA	BENGALURU	1232654578	F	AVERAGE
1RN13CS091	TEESHA	BENGALURU	1232654578	F	OUTSTANDING
1RN13CS091	TEESHA	BENGALURU	1232654578	F	AVERAGE

Commit