

DBMS LAB RECORD (TEST -2)

DIVYASHREE K

1BM19CS054

CSE-4A

Question:-

Program 6:

Consider the following schema for Order Database: SALESMAN (Salesman_id, Name, City, Commission) CUSTOMER (Customer_id, Cust_Name, City, Grade, Salesman_id) ORDERS (Ord_No, Purchase_Amt, Ord_Date, Customer_id, Salesman_id) Write SQL queries to

1. Count the customers with grades above Bangalore's average.
2. Find the name and numbers of all salesmen who had more than one customer.
3. List all salesmen and indicate those who have and don't have customers in their cities (Use UNION operation.)
4. Create a view that finds the salesman who has the customer with the highest order of a day.
5. Demonstrate the DELETE operation by removing

salesman with id 1000. All his orders must also be deleted.

Program 6:

```
create database orderdb1;
```

```
use orderdb1;
```

```
create table salesman(  
    salesman_id varchar(20),  
    salesman_name varchar(20),  
    salesman_city varchar(20),  
    commission varchar(20),  
    primary key(salesman_id)  
);
```

```
create table customer(  
    customer_id varchar(20),  
    customer_name varchar(20),
```

```
customer_city varchar(20),  
grade varchar(20),  
salesman_id varchar(20),  
primary key(customer_id),  
foreign key(salesman_id) references  
salesman(salesman_id) on delete set  
null);
```

```
create table orders(  
ord_no int,  
purchase_amt double,  
ord_date date,  
customer_id varchar(20),  
salesman_id varchar(20),
```

foreign key(salesman_id) references
salesman(salesman_id) on delete
cascade,

foreign key(customer_id) references
customer(customer_id) on delete
cascade

);

insert into salesman
values("1000","JHON","BANGLORE","2
5%"),

("2000","RAVI","BANGLORE","20%"),

("3000","KUMAR","MYSORE","15%"),

("4000","SMITH","DELHI","30%"),

("5000","HARSHA","HYDRABAD","15%
");

select * from salesman;

```
insert into customer
values("10","PREETHI","BANGLORE","1
00","1000"),

("11","VIVEK","MANGLORE","300","10
00"),

("12","BHASKAR","CHENNAI","400","20
00"),

("13","CHETHAN","BANGLORE","200","
2000"),

("14","MAMTHA","BANGLORE","400","
3000");

select * from customer;
```

```
insert into orders
values("50","5000","17-05-
04","10","1000"),

("51","450","17-01-20","10","2000"),

("52","1000","17-02-24","13","2000"),
```

```
("53","3500","17-04-13","14","3000"),
("54","550","17-03-09","12","2000");

select * from orders;

insert into salesman
values("1000","JHON","BANGLORE","2
5%"),

("2000","RAVI","BANGLORE","20%"),
("3000","KUMAR","MYSORE","15%"),
("4000","SMITH","DELHI","30%"),
("5000","HARSHA","HYDRABAD","15%"
);

select * from salesman;

insert into customer
values("10","PREETHI","BANGLORE","1
00","1000"),
```

```
("11","VIVEK","MANGLORE","300","1000"),
```

```
("12","BHASKAR","CHENNAI","400","2000"),
```

```
("13","CHETHAN","BANGLORE","200","2000"),
```

```
("14","MAMTHA","BANGLORE","400","3000");
```

```
select * from customer;
```

```
insert into orders
```

```
values("50","5000","17-05-04","10","1000"),
```

```
("51","450","17-01-20","10","2000"),
```

```
("52","1000","17-02-24","13","2000"),
```

```
("53","3500","17-04-13","14","3000"),
```

```
("54","550","17-03-09","12","2000");
```

select * from orders;

select grade,count(distinct
customer_id) from customer group by
grade having grade > (select

avg(grade) from customer where
customer_city ="BANGLORE");

select salesman_id ,salesman_name
from salesman S where 1 <(select
count(*) from customer

where salesman_id = S.salesman_id);

select salesman.salesman_id
,salesman_name,customer_name,com
mission from

salesman,customer where
salesman_city = customer_city union
select

salesman_id,salesman_name ,'NO
MATCH FOUND',commission from
salesman where not

salesman_city = any(select
customer_city from customer)order by
2 desc;

create view best_salesman as select
b.ord_date
,a.salesman_id,a.salesman_name from

salesman a,orders b where
a.salesman_id=b.salesman_id and
b.purchase_amt=(select

max(purchase_amt) from orders c
where c.ord_date=b.ord_date);

select * from best_salesman;

delete from salesman where
salesman_id = 1000;

Output:

7 | commission varchar(20),

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

	salesman_id	salesman_name	salesman_city	commission
▶	1000	JHON	BANGLORE	25%
	2000	RAVI	BANGLORE	20%
	3000	KUMAR	MYSORE	15%
	4000	SMITH	DELHI	30%
	5000	HARSHA	HYDRABAD	15%
*	NULL	NULL	NULL	NULL

salesman 1 x customer 2 orders 3

Output

7 | commission varchar(20),

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

	customer_id	customer_name	customer_city	grade	salesman_id
▶	10	PREETHI	BANGLORE	100	1000
	11	VIVEK	MANGLORE	300	1000
	12	BHASKAR	CHENNAI	400	2000
	13	CHETHAN	BANGLORE	200	2000
	14	MAMTHA	BANGLORE	400	3000
*	NULL	NULL	NULL	NULL	NULL

salesman 1 customer 2 x orders 3 Apply

Output

7 | commission varchar(20),

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

	ord_no	purchase_amt	ord_date	customer_id	salesman_id
▶	50	5000	2017-05-04	10	1000
	51	450	2017-01-20	10	2000
	52	1000	2017-02-24	13	2000
	53	3500	2017-04-13	14	3000
	54	550	2017-03-09	12	2000

salesman 1 customer 2 orders 3 x Read Only

Output

Question:-

Program 7:

BOOK (Book_id, Title, Publisher_Name,
Pub_Year) BOOK_AUTHORS (Book_id,
Author_Name) PUBLISHER (Name,
Address, Phone) BOOK_COPIES (Book_id,
Branch_id, No-of_Copies) BOOK_LENDING
(Book_id, Branch_id, Card_No, Date_Out,
Due_Date) LIBRARY_BRANCH (Branch_id,
Branch_Name, Address)

Program 7:

```
create database bookdb1;
```

```
use bookdb1;
```

```
CREATE TABLE PUBLISHER
```

```
(NAME VARCHAR(20) PRIMARY KEY,
```

```
PHONE VARCHAR(10),
```

```
ADDRESS VARCHAR(20));
```

```
SELECT * FROM PUBLISHER;

CREATE TABLE BOOK(
BOOK_ID INTEGER PRIMARY KEY,
TITLE VARCHAR(20),
PUB_YEAR VARCHAR(20),
PUBLISHER_NAME VARCHAR(20),
FOREIGN KEY (PUBLISHER_NAME)
REFERENCES PUBLISHER (NAME) ON
DELETE
CASCADE);

SELECT * FROM BOOK;

CREATE TABLE BOOK_AUTHORS(
AUTHOR_NAME VARCHAR(20),
BOOK_ID INTEGER,
```

PRIMARY KEY (BOOK_ID,
AUTHOR_NAME),

FOREIGN KEY (BOOK_ID) REFERENCES
BOOK (BOOK_ID) ON DELETE CASCADE
);

SELECT * FROM BOOK_AUTHORS;

CREATE TABLE LIBRARY_BRANCH (
BRANCH_ID INTEGER PRIMARY KEY,
BRANCH_NAME VARCHAR(20),
ADDRESS VARCHAR(20));

SELECT * FROM LIBRARY_BRANCH;

CREATE TABLE BOOK_COPIES

(

NO_OF_COPIES INTEGER,

BOOK_ID INTEGER,

```
BRANCH_ID INTEGER ,  
PRIMARY KEY (BOOK_ID, BRANCH_ID),  
FOREIGN KEY (BOOK_ID) REFERENCES  
BOOK (BOOK_ID) ON DELETE CASCADE,  
FOREIGN KEY (BRANCH_ID)  
REFERENCES LIBRARY_BRANCH  
(BRANCH_ID) ON  
DELETE CASCADE);  
  
SELECT * FROM BOOK_COPIES;  
  
CREATE TABLE CARD  
(CARD_NO INTEGER PRIMARY KEY);  
  
SELECT * FROM CARD;  
  
CREATE TABLE BOOK_LENDING(  
DATE_OUT DATE,  
DUE_DATE DATE,
```

```
BOOK_ID INTEGER,  
BRANCH_ID INTEGER ,  
CARD_NO INTEGER,  
PRIMARY KEY (BOOK_ID, BRANCH_ID,  
CARD_NO),  
FOREIGN KEY (BOOK_ID) REFERENCES  
BOOK (BOOK_ID) ON DELETE CASCADE,  
FOREIGN KEY (CARD_NO) REFERENCES  
CARD (CARD_NO) ON DELETE  
CASCADE,  
FOREIGN KEY (BRANCH_ID)  
REFERENCES LIBRARY_BRANCH  
(BRANCH_ID) ON  
DELETE CASCADE);  
SELECT * FROM BOOK_LENDING;
```

```
INSERT INTO PUBLISHER VALUES
("MCGRAW-HILL", "9989076587",
"BANGALORE"),

("PEARSON", "9889076565",
"NEWDELHI"),

("RANDOM HOUSE", "7455679345",
"HYDRABAD"),

("HACHETTE LIVRE", "8970862340",
"CHENNAI"),

("GRUPO
PLANETA","7756120238","BANGALORE
");
```

```
INSERT INTO BOOK VALUES
("1","DBMS","JAN-2017", "MCGRAW-
HILL"),

("2","ADBMS","JUN-2016", "MCGRAW-
HILL"),
```


("3","CN","SEP-2016", "PEARSON"),

("4","CG","SEP-2015","GRUPO
PLANETA"),

("5","OS","MAY-2016","PEARSON");

INSERT INTO BOOK_AUTHORS VALUES
("NAVATHE","1"),

("NAVATHE","2"),

("TANENBAUM","3"),

("EDWARD ANGEL","4"),

("GALVIN","5");

INSERT INTO LIBRARY_BRANCH VALUES
("10","RR NAGAR","BANGALORE"),

("11","RNSIT","BANGALORE"),

("12","RAJAJI NAGAR", "BANGALORE"),

("13","NITTE","MANGALORE"),

```
("14","MANIPAL","UDUPI");
```

```
INSERT INTO BOOK_COPIES VALUES
```

```
("10", "1",
```

```
"10"),("5","1","11"),("2","2","12"),("5",  
"2","13"),("7","3","14"),("1","5","10"),(  
"3","4","11");
```

```
INSERT INTO CARD VALUES
```

```
("100"),("101"),("102"),("103"),("104");
```

```
INSERT INTO BOOK_LENDING VALUES
```

```
("17-01-01","17-06-  
01","1","10","101"),
```

```
("17-01-17","17-03-17","3", "14",  
"101"),
```

```
("17-02-21","17-04-21", 2, 13, 101),
```

```
("17-03-15","17-07-15", "4", "11",  
"101"),
```

```
("17-04-12","17-05-12", "1", "11",  
"104");
```

```
SELECT B.BOOK_ID, B.TITLE,  
B.PUBLISHER_NAME,
```

```
A.AUTHOR_NAME,C.NO_OF_COPIES,L.  
BRANCH_ID
```

```
FROM BOOK B, BOOK_AUTHORS A,  
BOOK_COPIES C, LIBRARY_BRANCH L
```

```
WHERE B.BOOK_ID=A.BOOK_ID AND  
B.BOOK_ID=C.BOOK_ID AND
```

```
L.BRANCH_ID=C.BRANCH_ID;
```

```
SELECT CARD_NO FROM  
BOOK_LENDING
```

```
WHERE DATE_OUT BETWEEN "2017-  
01-01" AND "2017-07-01"
```

```
GROUP BY CARD_NO
```

HAVING COUNT(*)>3;

DELETE FROM BOOK

WHERE BOOK_ID=3;

CREATE VIEW YEAR_OF_PUBLICATION
AS SELECT PUB_YEAR

FROM BOOK;

SELECT * FROM
YEAR_OF_PUBLICATION;

CREATE VIEW
BOOKS_AVAILABLE_IN_LIBRARY AS

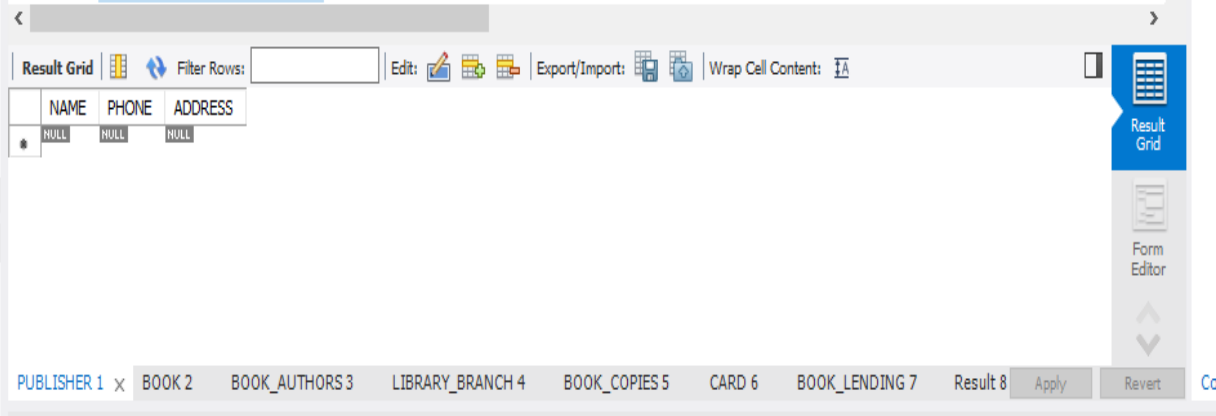
SELECT B.BOOK_ID, B.TITLE,
C.NO_OF_COPIES

FROM BOOK B, BOOK_COPIES C,
LIBRARY_BRANCH L

WHERE B.BOOK_ID=C.BOOK_ID

AND C.BRANCH_ID=L.BRANCH_ID;

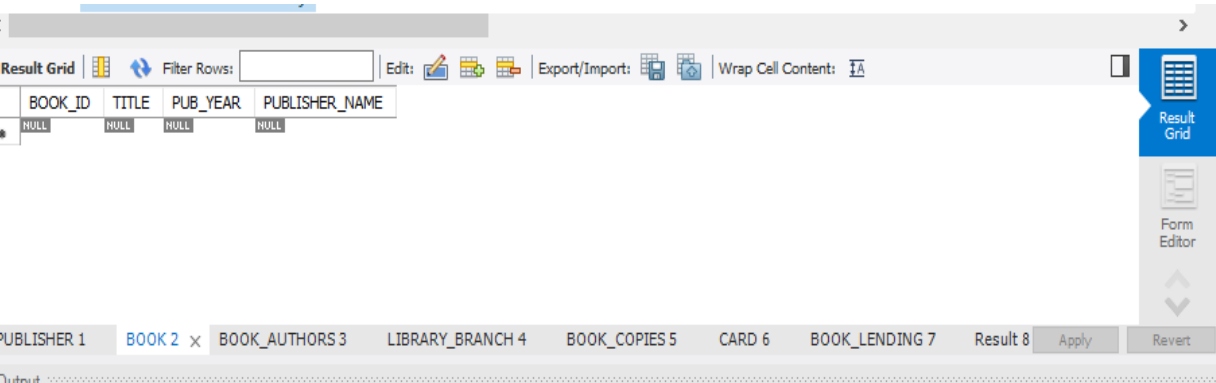
SELECT * FROM
BOOKS_AVAILABLE_IN_LIBRARY;
Output:



Result Grid

	NAME	PHONE	ADDRESS
*	NULL	NULL	NULL

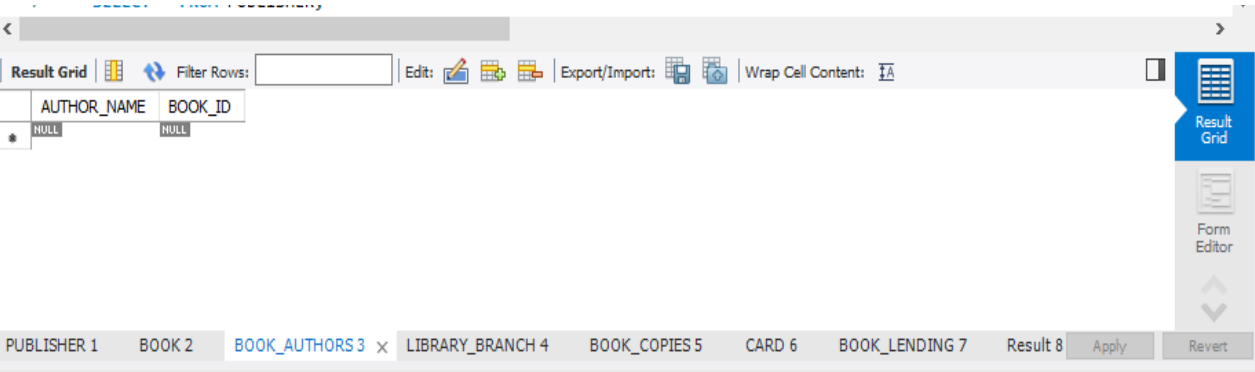
PUBLISHER 1 x BOOK 2 BOOK_AUTHORS 3 LIBRARY_BRANCH 4 BOOK_COPIES 5 CARD 6 BOOK_LENDING 7 Result 8 Apply Revert



Result Grid

	BOOK_ID	TITLE	PUB_YEAR	PUBLISHER_NAME
*	NULL	NULL	NULL	NULL

PUBLISHER 1 BOOK 2 x BOOK_AUTHORS 3 LIBRARY_BRANCH 4 BOOK_COPIES 5 CARD 6 BOOK_LENDING 7 Result 8 Apply Revert



Result Grid

	AUTHOR_NAME	BOOK_ID
*	NULL	NULL

PUBLISHER 1 BOOK 2 BOOK_AUTHORS 3 x LIBRARY_BRANCH 4 BOOK_COPIES 5 CARD 6 BOOK_LENDING 7 Result 8 Apply Revert

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

	BRANCH_ID	BRANCH_NAME	ADDRESS
*	NULL	NULL	NULL

Result Grid

Form Editor

PUBLISHER 1

BOOK 2

BOOK_AUTHORS 3

LIBRARY_BRANCH 4

BOOK_COPIES 5

CARD 6

BOOK_LENDING 7

Result 8

Apply

Revert

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

	NO_OF_COPIES	BOOK_ID	BRANCH_ID
*	NULL	NULL	NULL

Result Grid

Form Editor

PUBLISHER 1

BOOK 2

BOOK_AUTHORS 3

LIBRARY_BRANCH 4

BOOK_COPIES 5

CARD 6

BOOK_LENDING 7

Result 8

Apply

Revert

Output

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

	CARD_NO
*	NULL

Result Grid

Form Editor

PUBLISHER 1

BOOK 2

BOOK_AUTHORS 3

LIBRARY_BRANCH 4

BOOK_COPIES 5

CARD 6

BOOK_LENDING 7

Result 8

Apply

Revert

Output

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

	DATE_OUT	DUE_DATE	BOOK_ID	BRANCH_ID	CARD_NO
*	NULL	NULL	NULL	NULL	NULL

Result Grid

Form Editor

PUBLISHER 1

BOOK 2

BOOK_AUTHORS 3

LIBRARY_BRANCH 4

BOOK_COPIES 5

CARD 6

BOOK_LENDING 7 x

Result 8

Apply

Revert

Output

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	BOOK_ID	TITLE	PUBLISHER_NAME	AUTHOR_NAME	NO_OF_COPIES	BRANCH_ID
▶	1	DBMS	MCGRAW-HILL	NAVATHE	10	10
	1	DBMS	MCGRAW-HILL	NAVATHE	5	11
	2	ADBMS	MCGRAW-HILL	NAVATHE	2	12
	2	ADBMS	MCGRAW-HILL	NAVATHE	5	13
	3	CN	PEARSON	TANENBAUM	7	14
	4	CG	GRUPO PLANETA	EDWARD ANGEL	3	11
	5	OS	PEARSON	GALVIN	1	10

Result Grid

Form Editor

Field Types

BOOK_COPIES 5

CARD 6

BOOK_LENDING 7

Result 8 x

BOOK_LENDING 9

YEAR_OF_PUBLICATION 10

BOOKS_AVAILABLE_IN_LIBRAR. ⓘ

Read Only

Output

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	CARD_NO
▶	101

Result Grid

Form Editor

BOOK_COPIES 5

CARD 6

BOOK_LENDING 7

Result 8

BOOK_LENDING 9 x

YEAR_OF_PUBLICATION 10

BOOKS_AVAILABLE_IN_LIBRAR. ⓘ

Read Only

Output

7 • SELECT * FROM DUPLICATED.

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

PUB_YEAR
JAN-2017
JUN-2016
SEP-2015
MAY-2016

BOOK_COPIES 5 CARD 6 BOOK_LENDING 7 Result 8 BOOK_LENDING 9 YEAR_OF_PUBLICATION 10 x BOOKS_AVAILABLE_IN_LIBRAR. Read Only

Output :

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

BOOK_ID	TITLE	NO_OF_COPIES
1	DBMS	10
1	DBMS	5
2	ADBMS	2
2	ADBMS	5
4	CG	3
5	OS	1

BOOK_COPIES 5 CARD 6 BOOK_LENDING 7 Result 8 BOOK_LENDING 9 YEAR_OF_PUBLICATION 10 BOOKS_AVAILABLE_IN_LIBRAR. Read Only

Output :

Question:-

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.

Program 8:

```
CREATE DATABASE COLLEGE;
```

```
USE COLLEGE;
```

```
CREATE TABLE student(  
    regno VARCHAR(15),  
    sname VARCHAR(20),  
    major VARCHAR(20),  
    bdate DATE,  
    PRIMARY KEY (regno) );
```

```
CREATE TABLE course(  
    courseno INT,
```

```
cname VARCHAR(20),  
dept VARCHAR(20),  
PRIMARY KEY (courseno) );  
  
select * from course;  
  
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) );
```

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

```
CREATE TABLE book_adoption(  
    courseno INT,  
    sem INT(3),  
    book_isbn INT(5),  
    PRIMARY KEY (courseno,book_isbn),  
    FOREIGN KEY (courseno)  
REFERENCES course (courseno),
```

```
FOREIGN KEY (book_isbn)
REFERENCES text(book_isbn) );
```

```
INSERT INTO student
(regno,sname,major,bdate) VALUES
('1pe11cs002','b','sr','19930924'),
('1pe11cs003','c','sr','19931127'),
('1pe11cs004','d','sr','19930413'),
('1pe11cs005','e','jr','19940824');
```

```
INSERT INTO student
(regno,sname,major,bdate) VALUES
('1pe11cs001','a','jr','19930922');
```

```
INSERT INTO course VALUES  
(111,'OS','CSE'),  
    (112,'EC','CSE'),  
    (113,'SS','ISE'),  
    (114,'DBMS','CSE'),  
    (115,'SIGNALS','ECE');
```

```
INSERT INTO text 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');
```

```
INSERT INTO enroll  
(regno,courseno,sem,marks) VALUES  
('1pe11cs001',115,3,100),  
    ('1pe11cs002',114,5,100),  
    ('1pe11cs003',113,5,100),  
    ('1pe11cs004',111,5,100),
```

```
('1pe11cs005',112,3,100);
```

```
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 student;
```

```
select * from course;
```



```
select * from enroll;
```

```
select * from book_adoption;
```

```
select * from text;
```

```
/*4. 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.*/
```

```
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;
```

/*5. 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='HALL INDIA')
    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 != 'HALL INDIA');
Output:
```

120

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

	courseno	cname	dept
*	NULL	NULL	NULL

course 1 x student 2 course 3 enroll 4 book_adoption 5 text 6 Result 7 course 8

Apply Revert

Output

120

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

	regno	sname	major	bdate
▶	1pe11cs001	a	jr	1993-09-22
	1pe11cs002	b	sr	1993-09-24
	1pe11cs003	c	sr	1993-11-27
	1pe11cs004	d	sr	1993-04-13
	1pe11cs005	e	jr	1994-08-24
*	NULL	NULL	NULL	NULL

course 1 student 2 x course 3 enroll 4 book_adoption 5 text 6 Result 7 course 8

Apply Revert

Output

120

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

	courseno	cname	dept
▶	111	OS	CSE
	112	EC	CSE
	113	SS	ISE
	114	DBMS	CSE
	115	SIGNALS	ECE
*	NULL	NULL	NULL

course 1 student 2 course 3 x enroll 4 book_adoption 5 text 6 Result 7 course 8

Apply Revert

Output

120

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

	regno	courseno	sem	marks
▶	1pe11cs001	115	3	100
	1pe11cs002	114	5	100
	1pe11cs003	113	5	100
	1pe11cs004	111	5	100
	1pe11cs005	112	3	100
*	NULL	NULL	NULL	NULL

course 1 student 2 course 3 enroll 4 x book_adoption 5 text 6 Result 7 course 8

Apply Revert

Output

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

	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

course 1

student 2

course 3

enroll 4

book_adoption 5

text 6

Result 7

course 8

Apply

Revert

Result Grid

Form Editor

119

119

<

>

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

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

Result Grid

Form Editor

course 1

student 2

course 3

enroll 4

book_adoption 5

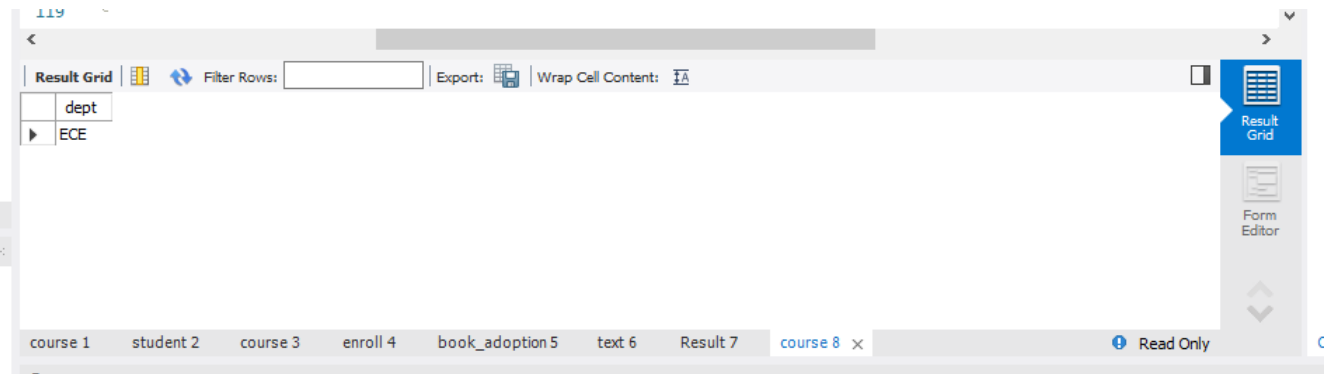
text 6

Result 7 x

course 8

Read Only

Output



Question:-

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

- i. List the titles of all movies directed by 'Hitchcock'.
- ii. Find the movie names where one or more actors acted in two or more movies.
- iii. List all actors who acted in a movie before 2000 and also in a movie after 2015 (use JOIN operation).
- iv. 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.
- v. Update rating of all movies directed by 'Steven Spielberg' to 5.

Program 9:

CREATE DATABASE MOVIE;

USE MOVIE;

CREATE TABLE ACTOR (

```
ACT_ID INT,  
ACT_NAME VARCHAR (20),  
ACT_GENDER CHAR (1),  
PRIMARY KEY (ACT_ID));
```

```
CREATE TABLE DIRECTOR (  
DIR_ID INT,  
DIR_NAME VARCHAR (20),  
DIR_PHONE LONG,  
PRIMARY KEY (DIR_ID));
```

```
CREATE TABLE MOVIES (  
MOV_ID INT,  
MOV_TITLE VARCHAR (25),
```



```
MOV_YEAR INT,  
MOV_LANG VARCHAR (12),  
DIR_ID INT,  
PRIMARY KEY (MOV_ID),  
FOREIGN KEY (DIR_ID) REFERENCES  
DIRECTOR (DIR_ID));
```

```
CREATE TABLE MOVIE_CAST (  
ACT_ID INT,  
MOV_ID INT,  
AROLE VARCHAR(10),  
PRIMARY KEY (ACT_ID, MOV_ID),  
FOREIGN KEY(ACT_ID) REFERENCES  
ACTOR(ACT_ID) ON DELETE CASCADE,
```

```
FOREIGN KEY(MOV_ID) REFERENCES  
MOVIES(MOV_ID) ON DELETE  
CASCADE);
```

```
CREATE TABLE RATING (  
MOV_ID INT,  
REV_STARS VARCHAR (25),  
PRIMARY KEY (MOV_ID),  
FOREIGN KEY (MOV_ID) REFERENCES  
MOVIES (MOV_ID));
```

```
INSERT INTO ACTOR VALUES  
(301,'ANUSHKA','F');
```

```
INSERT INTO ACTOR VALUES  
(302,'PRABHAS','M');
```

```
INSERT INTO ACTOR VALUES  
(303,'PUNITH','M');
```

```
INSERT INTO ACTOR VALUES  
(304,'JERMY','M');
```

```
INSERT INTO DIRECTOR VALUES  
(60,'RAJAMOULI', 8751611001);
```

```
INSERT INTO DIRECTOR VALUES  
(61,'HITCHCOCK', 7766138911);
```

```
INSERT INTO DIRECTOR VALUES  
(62,'FARAN', 9986776531);
```

```
INSERT INTO DIRECTOR VALUES  
(63,'STEVEN SPIELBERG', 8989776530);
```

```
INSERT INTO MOVIES VALUES  
(1001,'BAHUBALI-2', 2017,'TELAGU',  
60);
```

```
INSERT INTO MOVIES VALUES  
(1002,'BAHUBALI-1', 2015, 'TELAGU',  
60);
```

```
INSERT INTO MOVIES VALUES  
(1003,'AKASH', 2008, 'KANNADA', 61);
```

```
INSERT INTO MOVIES VALUES  
(1004,'WAR HORSE', 2011, 'ENGLISH',  
63);
```

```
INSERT INTO MOVIE_CAST VALUES  
(301, 1002, 'HEROINE');
```

```
INSERT INTO MOVIE_CAST VALUES  
(301, 1001, 'HEROINE');
```

```
INSERT INTO MOVIE_CAST VALUES  
(303, 1003, 'HERO');
```

```
INSERT INTO MOVIE_CAST VALUES  
(303, 1002, 'GUEST');
```

```
INSERT INTO MOVIE_CAST VALUES  
(304, 1004, 'HERO');
```

```
INSERT INTO RATING VALUES (1001, 4);
```

```
INSERT INTO RATING VALUES (1002, 2);
```

```
INSERT INTO RATING VALUES (1003, 5);
```

```
INSERT INTO RATING VALUES (1004, 4);
```

```
select * from rating;
```

```
select * from actor;
```

```
select * from director;
```

```
select * from movies;
```

```
select * from movie_cast;
```

/*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(*)>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;
```

/*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 MOV_TITLE, MAX(REV_STARS)
FROM MOVIES
INNER JOIN RATING USING (MOV_ID)
GROUP BY MOV_TITLE
```


HAVING MAX(REV_STARS)>0

ORDER BY MOV_TITLE;

/*5. Update rating of all movies
directed by 'Steven Spielberg' to 5 KL
*/

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'));

select * from rating;

Output:

MOV_ID	REV_STARS
1001	4
1002	2
1003	5
1004	4
NULL	NULL

ACT_ID	ACT_NAME	ACT_GENDER
301	ANUSHKA	F
302	PRABHAS	M
303	PUNITH	M
304	JERMY	M
NULL	NULL	NULL

DIR_ID	DIR_NAME	DIR_PHONE
60	RAJAMOULI	8751611001
61	HITCHCOCK	7766138911
62	FARAN	9986776531
63	STEVEN SPIELBERG	8989776530
NULL	NULL	NULL

MOV_ID	MOV_TITLE	MOV_YEAR	MOV_LANG	DIR_ID
1001	BAHUBALI-2	2017	TELAGU	60
1002	BAHUBALI-1	2015	TELAGU	60
1003	AKASH	2008	KANNADA	61
1004	WAR HORSE	2011	ENGLISH	63
NULL	NULL	NULL	NULL	NULL

rating 1 actor 2 director 3 movies 4 x movie_cast 5 MOVIES 6 Result 7 Result 8 Result 9

Output

ACT_ID	MOV_ID	AROLE
301	1001	HEROINE
301	1002	HEROINE
303	1002	GUEST
303	1003	HERO
304	1004	HERO
NULL	NULL	NULL

rating 1 actor 2 director 3 movies 4 movie_cast 5 x MOVIES 6 Result 7 Result 8 Result 9

Output

MOV_TITLE
AKASH

rating 1 actor 2 director 3 movies 4 movie_cast 5 MOVIES 6 x Result 7 Result 8 Result 9

Output

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
MOV_TITLE			
BAHUBALI-1			

rating 1 actor 2 director 3 movies 4 movie_cast 5 MOVIES 6 Result 7 x Result 8 Result 9 Read Only

Output

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
ACT_NAME	MOV_TITLE	MOV_YEAR	
ANUSHKA	BAHUBALI-2	2017	

rating 1 actor 2 director 3 movies 4 movie_cast 5 MOVIES 6 Result 7 Result 8 x Result 9 Read Only

Output

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
MOV_TITLE	MAX(REV_STARS)		
AKASH	5		
BAHUBALI-1	2		
BAHUBALI-2	4		
WAR HORSE	4		

rating 1 actor 2 director 3 movies 4 movie_cast 5 MOVIES 6 Result 7 Result 8 Result 9 x Read Only

Output

Question:-

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

- i. List all the student details studying in fourth semester 'C' section.
- ii. Compute the total number of male and female students in each semester and in each section.
- iii. Create a view of Test1 marks of student USN '1BI15CS101' in all subjects.

iv. Calculate the FinalIA (average of best two test marks) and update the corresponding table for all students.

v. 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.

Program 10:

```
CREATE DATABASE COLLEGEDB;
```

```
USE COLLEGEDB;
```

```
CREATE TABLE STUDENT (
```

```
USN VARCHAR (10),  
SNAME VARCHAR (25),  
ADDRESS VARCHAR (25),  
PHONE LONG,  
GENDER CHAR (1),  
PRIMARY KEY (USN));
```

```
select * from student;
```

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

```
select * from semsec;
```

```
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));
```

```
select * from class;
```

```
CREATE TABLE SUBJECT (  
    SUBCODE VARCHAR (8),  
    TITLE VARCHAR (20),
```



```
SEM INT,  
CREDITS INT,  
PRIMARY KEY (SUBCODE));  
select * from subject;
```

```
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));

select * from iamarks;

INSERT INTO STUDENT VALUES
('1RN13CS020','AKSHAY','BELAGAVI',
8877881122,'M');

INSERT INTO STUDENT VALUES
('1RN13CS062','SANDHYA','BENGALUR
U', 7722829912,'F');

INSERT INTO STUDENT VALUES
('1RN13CS091','TEESHA','BENGALURU',
7712312312,'F');

```
INSERT INTO STUDENT VALUES  
('1RN13CS066','SUPRIYA','MANGALUR  
U', 8877881122,'F');
```

```
INSERT INTO STUDENT VALUES  
('1RN14CS010','ABHAY','BENGALURU',  
9900211201,'M');
```

```
INSERT INTO STUDENT VALUES  
('1RN14CS032','BHASKAR','BENGALURU  
, 9923211099,'M');
```

```
INSERT INTO STUDENT VALUES  
('1RN14CS025','ASMI','BENGALURU',  
7894737377,'F');
```

```
INSERT INTO STUDENT VALUES  
('1RN15CS011','AJAY','TUMKUR',  
9845091341,'M');
```

```
INSERT INTO STUDENT VALUES  
('1RN15CS029','CHITRA','DAVANGERE',  
7696772121,'F');
```

```
INSERT INTO STUDENT VALUES  
('1RN15CS045','JEEVA','BELLARY',  
9944850121,'M');
```

```
INSERT INTO STUDENT VALUES  
('1RN15CS091','SANTOSH','MANGALUR  
U', 8812332201,'M');
```

```
INSERT INTO STUDENT VALUES  
('1RN16CS045','ISMAIL','KALBURGI',  
9900232201,'M');
```

```
INSERT INTO STUDENT VALUES  
('1RN16CS088','SAMEERA','SHIMOGA',  
9905542212,'F');
```

```
INSERT INTO STUDENT VALUES  
('1RN16CS122','VINAYAKA','CHIKAMAG  
ALUR', 8800880011,'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 CLASS VALUES
('1RN15CS045','CSE4B');

INSERT INTO CLASS VALUES
('1RN15CS091','CSE4C');

INSERT INTO CLASS VALUES
('1RN16CS045','CSE3A');

INSERT INTO CLASS VALUES
('1RN16CS088','CSE3B');

INSERT INTO CLASS VALUES
('1RN16CS122','CSE3C');

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 SUBJECT VALUES  
( '10CS75', 'JAVA', 7, 4);
```

```
INSERT INTO SUBJECT VALUES  
( '10CS76', 'SAN', 7, 4);
```

```
INSERT INTO SUBJECT VALUES  
( '15CS51', 'ME', 5, 4);
```

```
INSERT INTO SUBJECT VALUES  
( '15CS52', 'CN', 5, 4);
```

```
INSERT INTO SUBJECT VALUES  
( '15CS53', 'DBMS', 5, 4);
```

```
INSERT INTO SUBJECT VALUES  
( '15CS54', 'ATC', 5, 4);
```

```
INSERT INTO SUBJECT VALUES  
( '15CS55', 'JAVA', 5, 3);
```

INSERT INTO SUBJECT VALUES
('15CS56','AI', 5, 3);

INSERT INTO SUBJECT VALUES
('15CS41','M4', 4, 4);

INSERT INTO SUBJECT VALUES
('15CS42','SE', 4, 4);

INSERT INTO SUBJECT VALUES
('15CS43','DAA', 4, 4);

INSERT INTO SUBJECT VALUES
('15CS44','MPMC', 4, 4);

INSERT INTO SUBJECT VALUES
('15CS45','OOC', 4, 3);

INSERT INTO SUBJECT VALUES
('15CS46','DC', 4, 3);

INSERT INTO SUBJECT VALUES
('15CS31','M3', 3, 4);

```
INSERT INTO SUBJECT VALUES  
( '15CS32','ADE', 3, 4);
```

```
INSERT INTO SUBJECT VALUES  
( '15CS33','DSA', 3, 4);
```

```
INSERT INTO SUBJECT VALUES  
( '15CS34','CO', 3, 4);
```

```
INSERT INTO SUBJECT VALUES  
( '15CS35','USP', 3, 3);
```

```
INSERT INTO SUBJECT VALUES  
( '15CS36','DMS', 3, 3);
```

```
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);
```

```
/*1. List all the student details studying  
in fourth semester 'C' 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='C';
```

/*2. 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;

/*3. 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;
```

/*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.

*/

```
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;

Output:

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

	USN	SNAME	ADDRESS	PHONE	GENDER
*	NULL	NULL	NULL	NULL	NULL

Result Grid

Form Editor

student 1

semsec 2

class 3

subject 4

iamarks 5

Result 6

Result 7

STU_TEST1_MARKS_VIEW 8

Result 9

Apply

Revert

Output

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

	SSID	SEM	SEC
*	NULL	NULL	NULL

Result Grid

Form Editor

student 1

semsec 2

class 3

subject 4

iamarks 5

Result 6

Result 7

STU_TEST1_MARKS_VIEW 8

Result 9

Apply

Revert

Output

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

	USN	SSID
*	NULL	NULL

Result Grid

Form Editor

student 1

semsec 2

class 3

subject 4

iamarks 5

Result 6

Result 7

STU_TEST1_MARKS_VIEW 8

Result 9

Apply

Revert

Output

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

	SUBCODE	TITLE	SEM	CREDITS
*	NULL	NULL	NULL	NULL

Result Grid

Form Editor

student 1

semsec 2

class 3

subject 4 x

iamarks 5

Result 6

Result 7

STU_TEST1_MARKS_VIEW 8

Result 9

Apply

Revert

Output

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

	USN	SUBCODE	SSID	TEST1	TEST2	TEST3	FINALIA
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Result Grid

Form Editor

student 1

semsec 2

class 3

subject 4

iamarks 5 x

Result 6

Result 7

STU_TEST1_MARKS_VIEW 8

Result 9

Apply

Revert

Output

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	USN	SNAME	ADDRESS	PHONE	GENDER	SEM	SEC
▶	1RN15CS091	SANTOSH	MANGALURU	8812332201	M	4	C

Result Grid

Form Editor

student 1

semsec 2

class 3

subject 4

iamarks 5

Result 6 x

Result 7

STU_TEST1_MARKS_VIEW 8

Result 9

Read Only

Output

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	USN	SNAME	ADDRESS	PHONE	GENDER	CAT
▶	1RN13CS091	TEESHA	BENGALURU	7712312312	F	WEAK
	1RN13CS091	TEESHA	BENGALURU	7712312312	F	WEAK
	1RN13CS091	TEESHA	BENGALURU	7712312312	F	WEAK
	1RN13CS091	TEESHA	BENGALURU	7712312312	F	WEAK
	1RN13CS091	TEESHA	BENGALURU	7712312312	F	WEAK

Result Grid

Form Editor

Read Only

student 1 semsec 2 class 3 subject 4 iamarks 5 Result 6 Result 7 STU_TEST1_MARKS_VIEW 8 Result 9 ×

Output