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)

```
CREATE TABLE ACTOR(
    ACT_ID INT,
    ACT_NAME VARCHAR(10),
    ACT_GENDER VARCHAR(10),
    PRIMARY KEY(ACT_ID)
    );
```

```
INSERT INTO `actor` (`ACT_ID`, `ACT_NAME`, `ACT_GENDER`) VALUES ('1', 'Tracy', 'MA
LE'), ('2', 'Keanu', 'MALE'), ('3', 'Khan', 'MALE'), ('4', 'Brad', 'MALE'), ('5',
'Rai', 'FEMALE')
```

	ACT_ID	ACT_NAME	ACT_GENDER
Edit Copy Delete	1	Tracy	MALE
Edit Copy Delete	2	Keanu	MALE
Edit Copy Delete	3	Khan	MALE
Edit Copy Delete	4	Brad	MALE
Edit Copy Delete	5	Rai	FEMALE

```
CREATE TABLE DIRECTOR(
    DIR_ID INT,
    DIR_NAME VARCHAR(10),
    DIR_PHONE INT,
    PRIMARY KEY(DIR_ID)
    );
```

```
INSERT INTO `director` (`DIR_ID`, `DIR_NAME`, `DIR_PHONE`) VALUES ('10', 'Bont', '
301742'), ('20', 'Steve', '541829'), ('30', 'Barton', '947382'), ('40', 'Downey',
'840656'), ('50', 'Kevin', '729146')
```

	DIR_ID	DIR_NAME	DIR_PHONE
Edit Copy Delete	10	Bont	301742
Edit Copy Delete	20	Steve	541829
Edit Copy Delete	30	Barton	947382
Edit Copy Delete	40	Downey	840656
Edit Copy Delete	50	Kevin	729146

```
CREATE TABLE MOVIES(
    MOV_ID INT,
    MOV_TITLE VARCHAR(10),
    MOV_YEAR VARCHAR(10),
    MOV_LANG VARCHAR(10),
    DIR_ID INT,PRIMARY KEY(MOV_ID),
    FOREIGN KEY(DIR_ID) REFERENCES DIRECTOR(DIR_ID)
```

```
    );
INSERT INTO `movies` (`MOV_ID`, `MOV_TITLE`, `MOV_YEAR`, `MOV_LANG`, `DIR_ID`) VALUES ('11', 'NOTEBOOK', '2018', 'HINDI', '20'), ('22', 'AVENGERS', '2007', 'ENGLISH', '20'), ('33', '1917', '1999', 'ENGLISH', '10'), ('44', 'GUILTY', '2020', 'HINDI', '10'), ('55', 'WAR', '2018', 'HINDI', '50')
```

	MOV_ID	MOV_TITLE	MOV_YEAR	MOV_LANG	DIR_ID
Edit Copy Delete	11	NOTEBOOK	2018	HINDI	20
Edit Copy Delete	22	AVENGERS	2007	ENGLISH	20
Edit Copy Delete	33	1917	1999	ENGLISH	10
Edit Copy Delete	44	GUILTY	2020	HINDI	10
Edit Copy Delete	55	WAR	2018	HINDI	50

```
CREATE TABLE MOVIE_CAST(
    ACT_ID INT,MOV_ID INT,
    ROLE_PLAYED VARCHAR(10),
    FOREIGN KEY(ACT_ID) REFERENCES ACTOR(ACT_ID),
    FOREIGN KEY(MOV_ID) REFERENCES MOVIES(MOV_ID)
    );
```

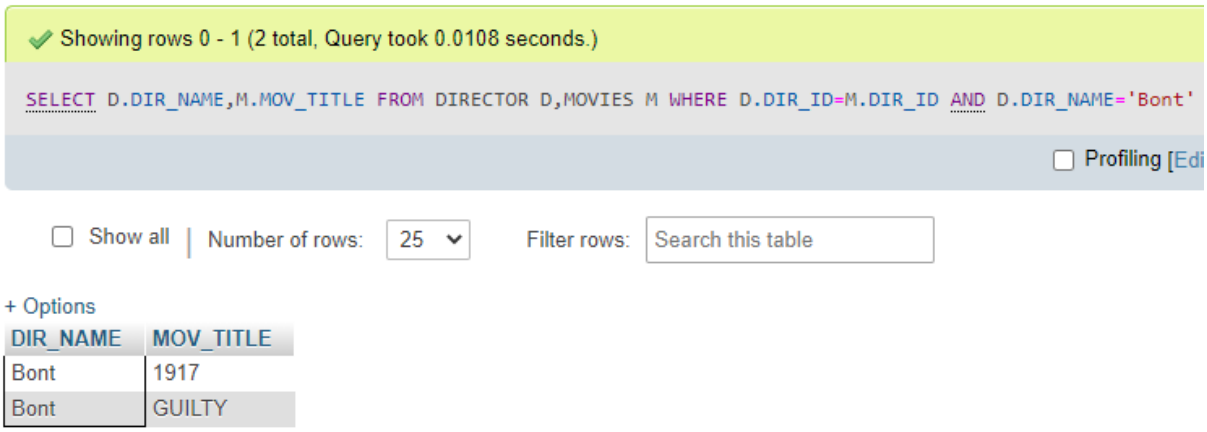
ACT_ID	MOV_ID	ROLE_PLAYED
5	55	VILLAIN
1	55	MAIN LEAD
5	22	SISTER
4	22	MAIN LEAD
4	55	BROTHER
2	44	VILLAIN

```
CREATE TABLE RATING(
    MOV_ID INT,REV_STARS FLOAT,
    FOREIGN KEY(MOV_ID) REFERENCES MOVIES(MOV_ID)
    );
```

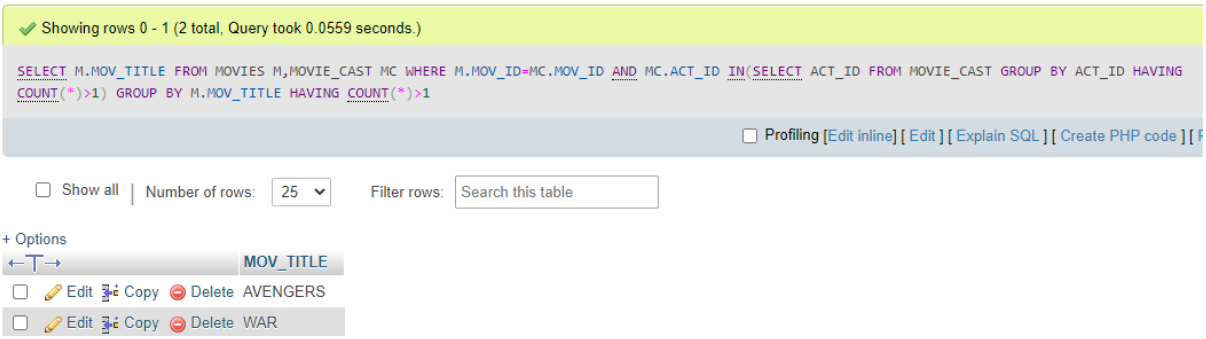
MOV_ID	REV_STARS
11	4
55	3.5
22	2.5
11	3.5
11	4

Write SQL queries to

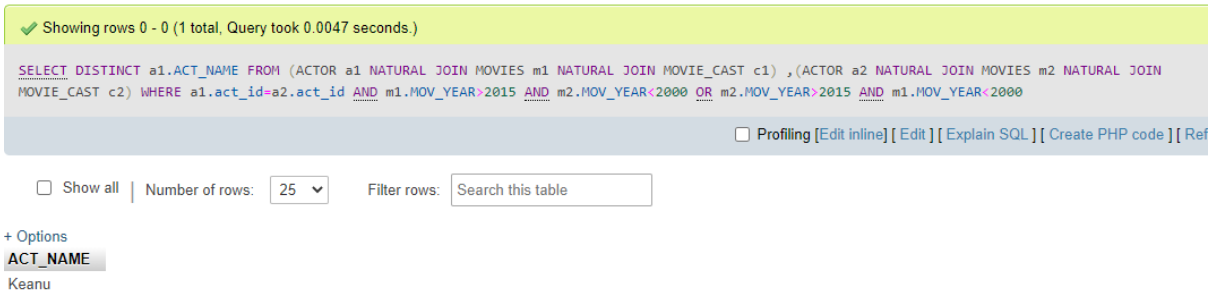
i. List the titles of all movies directed by a specific director.



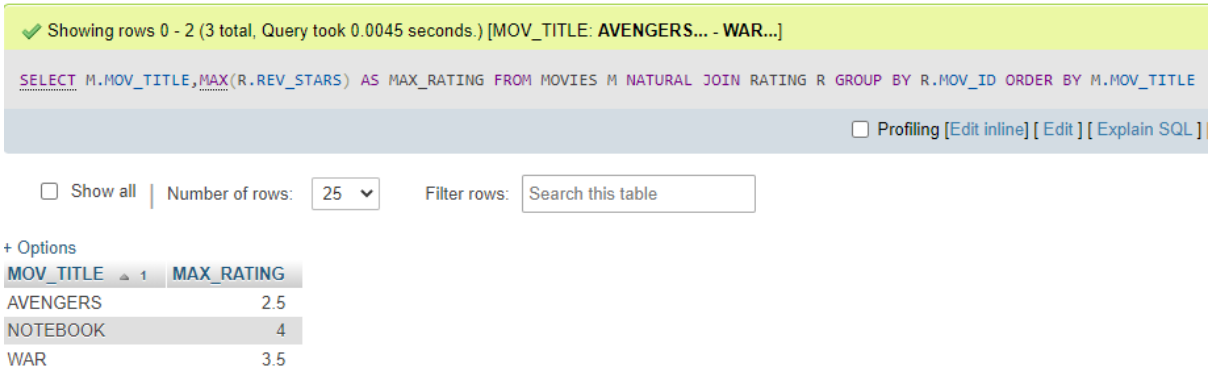
ii. Find the movie names where one or more actors acted in two or more movies.



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 ‘Steve’ to 5.

```
UPDATE RATING SET REV_STARS=5 WHERE MOV_ID IN (SELECT M.MOV_ID FROM MOVIES M NATURAL JOIN DIRECTOR D WHERE D.DIR_NAME='Steve')
```

[Edit inline] [E

⚠ Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

✔ Showing rows 0 - 4 (5 total, Query took 0.0008 seconds.)

```
SELECT * FROM RATING
```

Profiling [Edit inline] [Edit] [Explain SQL] [Crea

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options

MOV_ID	REV_STARS
11	5
55	3.5
22	5
11	5
11	5

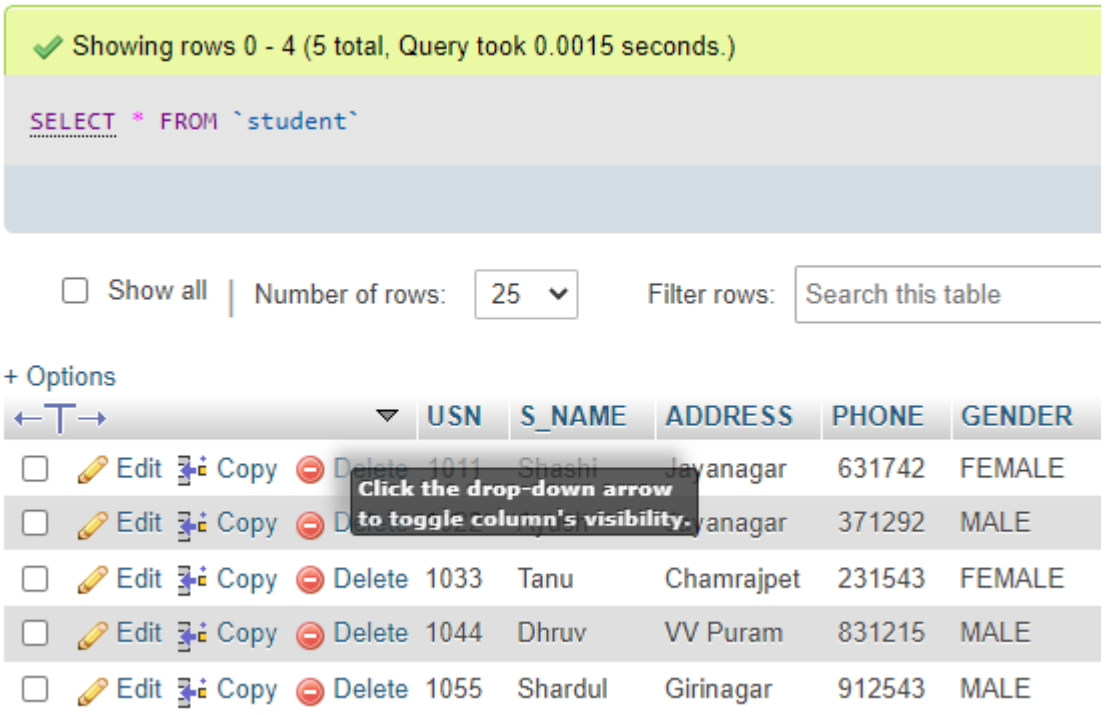
****

********** Program 10 in next page**********

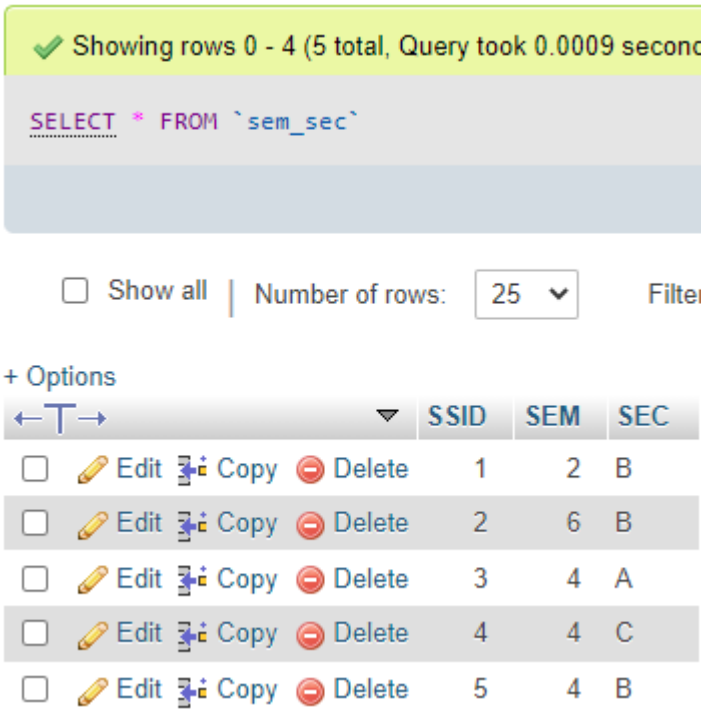
POGRAM 10:COLLEGE DATABASE

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)

```
CREATE TABLE STUDENT(
    USN INT,S_NAME VARCHAR(10),
    ADDRESS VARCHAR(20),
    PHONE INT,
    GENDER VARCHAR(10),
    PRIMARY KEY(USN)
    );
```

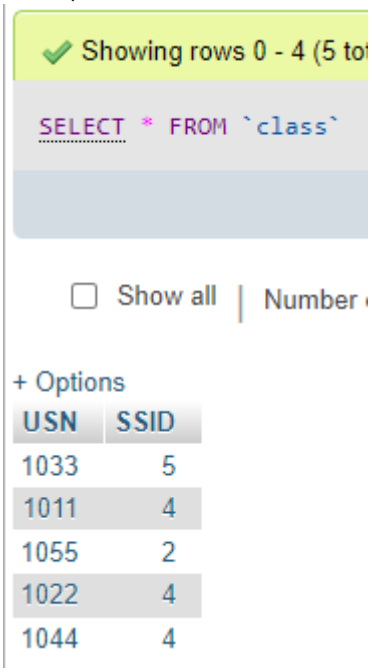



```
CREATE TABLE SEM_SEC(
    SSID INT,
    SEM INT,
    SEC VARCHAR(5),
    PRIMARY KEY(SSID)
    );
```



	SSID	SEM	SEC
Edit Copy Delete	1	2	B
Edit Copy Delete	2	6	B
Edit Copy Delete	3	4	A
Edit Copy Delete	4	4	C
Edit Copy Delete	5	4	B

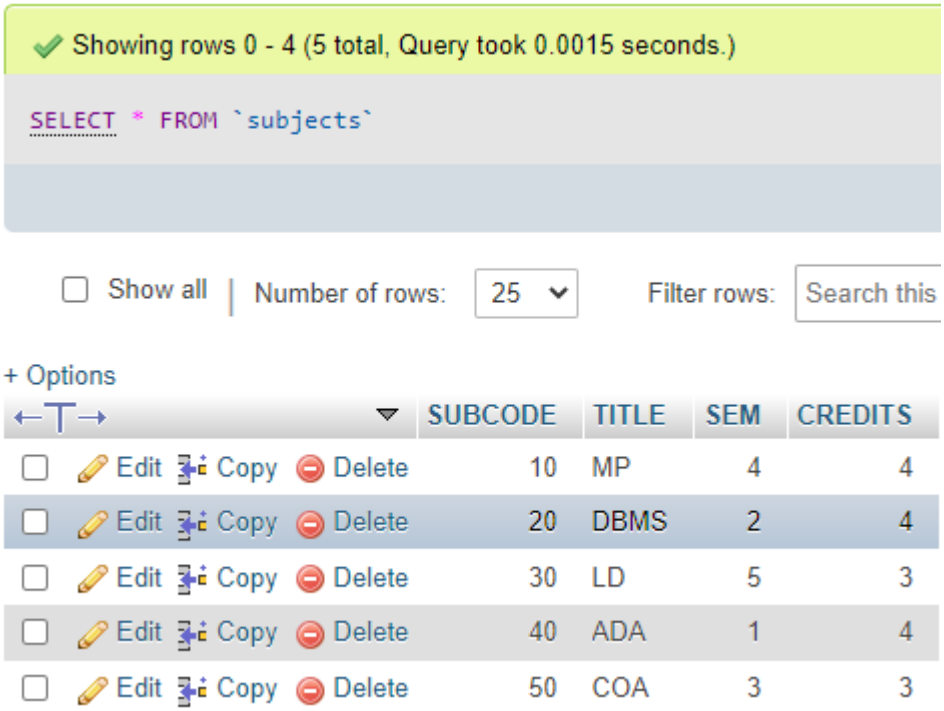
```
CREATE TABLE CLASS(
    USN INT,
    SSID INT,
    FOREIGN KEY(USN) REFERENCES STUDENT(USN),
    FOREIGN KEY(SSID) REFERENCES SEM_SEC(SSID)
    );
```



USN	SSID
1033	5
1011	4
1055	2
1022	4
1044	4

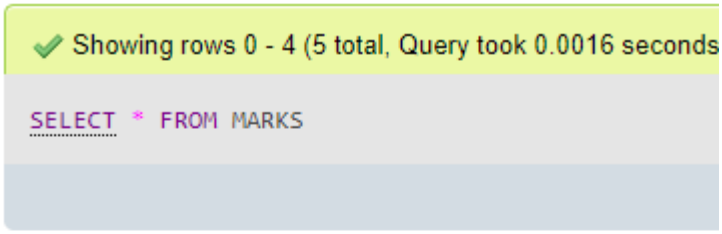
```
CREATE TABLE SUBJECTS(
    SUBCODE INT,
    TITLE VARCHAR(20),
    SEM INT,
```

```
    CREDITS INT,
    PRIMARY KEY(SUBCODE)
    );
```



	SUBCODE	TITLE	SEM	CREDITS
Edit Copy Delete	10	MP	4	4
Edit Copy Delete	20	DBMS	2	4
Edit Copy Delete	30	LD	5	3
Edit Copy Delete	40	ADA	1	4
Edit Copy Delete	50	COA	3	3

```
CREATE TABLE MARKS(
    USN INT,
    SUBCODE INT,
    SSID INT,
    TEST1 INT,
    TEST2 INT,
    TEST3 INT,
    FOREIGN KEY(USN) REFERENCES STUDENT(USN),
    FOREIGN KEY(SSID) REFERENCES SEM_SEC(SSID),
    FOREIGN KEY(SUBCODE) REFERENCES SUBJECTS(SUBCODE)
    );
```

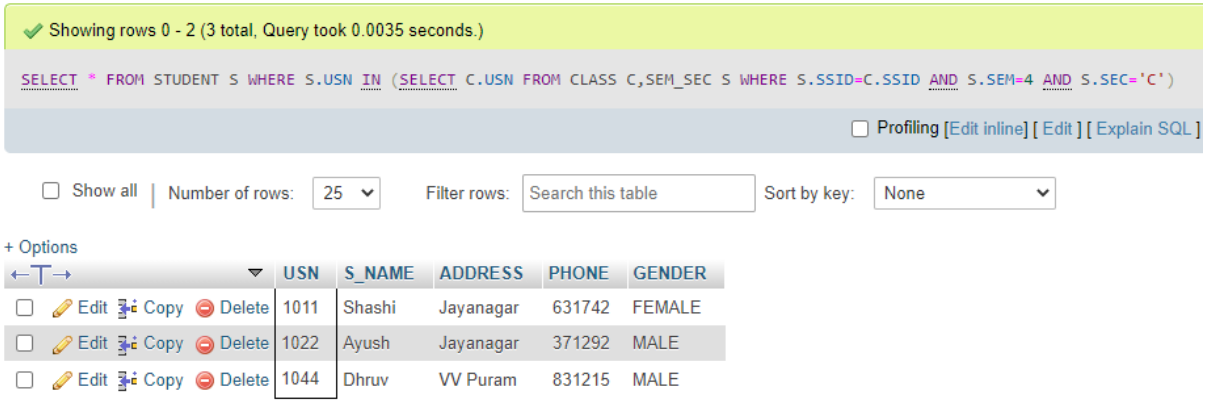


+ Options

USN	SUBCODE	SSID	TEST1	TEST2	TEST3
1033	10	5	19	19	20
1055	30	2	19	19	19
1022	40	4	12	18	16
1044	10	4	10	12	11
1011	20	4	15	14	13

Write SQL queries to

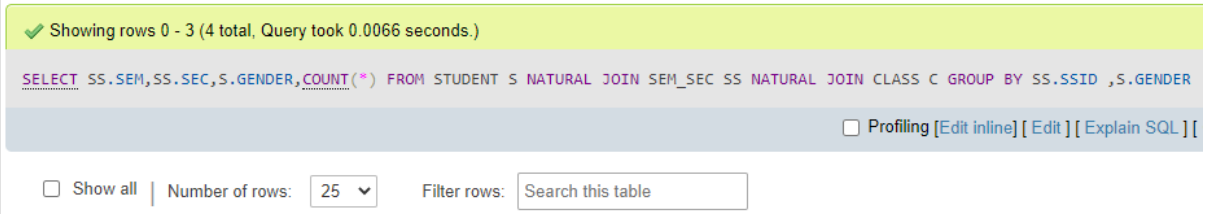
i. List all the student details studying in fourth semester 'C' section



+ Options

	USN	S_NAME	ADDRESS	PHONE	GENDER
Edit Copy Delete	1011	Shashi	Jayanagar	631742	FEMALE
Edit Copy Delete	1022	Ayush	Jayanagar	371292	MALE
Edit Copy Delete	1044	Dhruv	VV Puram	831215	MALE

ii. Compute the total number of male and female students in each semester and in each section.



+ Options

SEM	SEC	GENDER	COUNT(*)
6	B	MALE	1
4	C	FEMALE	1
4	C	MALE	2
4	B	FEMALE	1

iii. Create a view of Test1 marks of student USN '1022' in all subjects.

```
CREATE VIEW USN_22(USN,SUB,MARKS) AS SELECT M.USN,S.TITLE,M.TEST1 FROM
MARKS M,SUBJECTS S WHERE M.SUBCODE=S.SUBCODE AND M.USN=1022;
```

```
SELECT * FROM USN_22
```

+ Options

	USN	SUB	MARKS
Edit Copy Delete	1022	ADA	12

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

```
ALTER TABLE MARKS ADD COLUMN FINAL_ALL FLOAT;
UPDATE MARKS SET FINAL_ALL=((TEST1+TEST2+TEST3)-
LEAST(TEST1,TEST2,TEST3))/2;
SELECT * FROM MARKS;
```

USN	SUBCODE	SSID	TEST1	TEST2	TEST3	FINAL_ALL
1033	10	5	19	19	20	19.5
1055	30	2	19	19	19	19
1022	40	4	12	18	16	17
1044	10	4	10	12	11	11.5
1011	20	4	15	14	13	14.5

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.

```
ALTER TABLE MARKS ADD COLUMN CATEGORY VARCHAR(20);
UPDATE MARKS SET CATEGORY=
                        CASE
           WHEN FINAL_ALL>=17 AND FINAL_ALL<=20 THEN 'OUTSTANDING'
                    WHEN FINAL_ALL>=12 AND FINAL_ALL<17 THEN 'AVERAGE'
                    WHEN FINAL_ALL<12 THEN 'WEAK'
                        END;
SELECT * FROM MARKS;
```

+ Options

USN	SUBCODE	SSID	TEST1	TEST2	TEST3	FINAL_ALL	CATEGORY
1033	10	5	19	19	20	19.5	OUTSTANDING
1055	30	2	19	19	19	19	OUTSTANDING
1022	40	4	12	18	16	17	OUTSTANDING
1044	10	4	10	12	11	11.5	WEAK
1011	20	4	15	14	13	14.5	AVERAGE

****