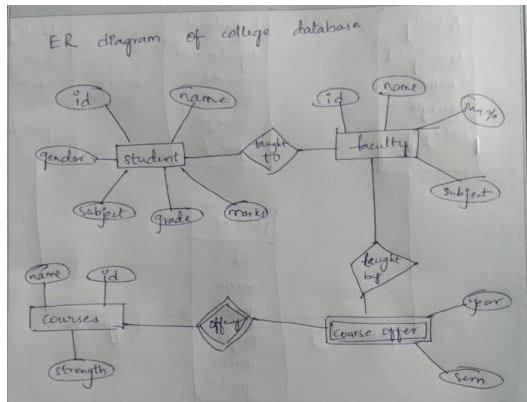


Experiment –1

ER DIAGRAM OF COLLEGE DATABASE

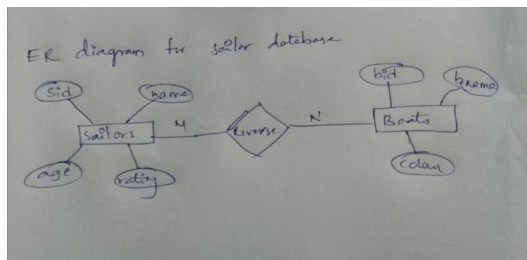
Aim: To draw an ER diagram of college database



Experiment-2

ER DIAGRAM FOR SAILOR BOAT DATABASE

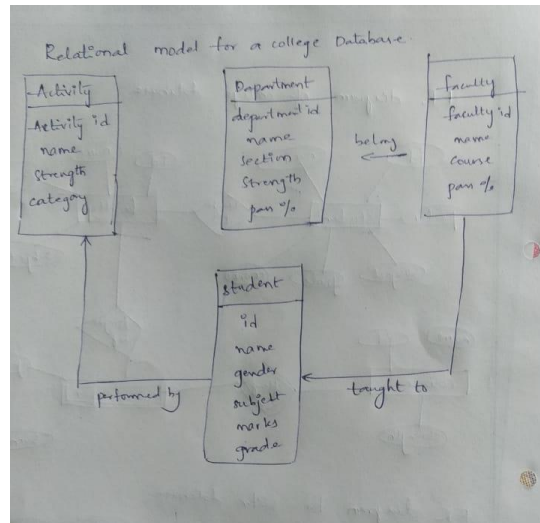
Aim: Draw an ER diagram of sailor boat database



Experiment-3

RELATIONAL MODEL FOR COLLEGE DATABASE

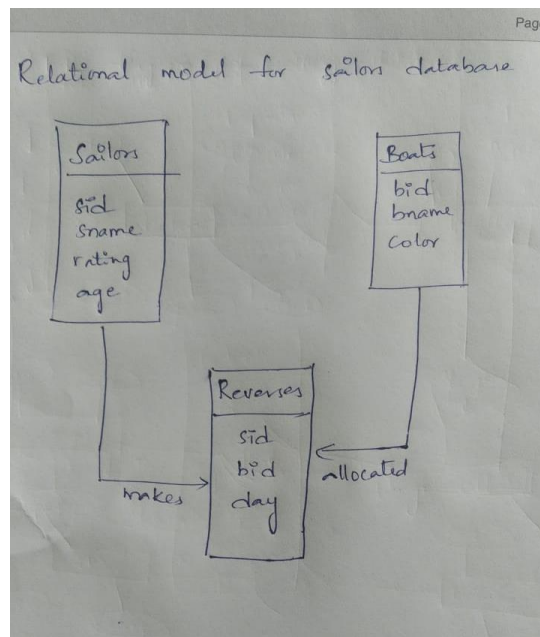
Aim: Draw a relational model for a college database



Experiment-4

RELATIONAL MODEL FOR SAILOR BOAT DATABASE

Aim: draw a relational model for a sailor boat database



Experiment-5

Activity – 1

1.Create schema college

```
2 • CREATE SCHEMA Bvrit_549;
```

2.Create table-student and attributes are studid,studname,gender,subject,marks ,grade

3.Create table-Faculty and attributes are fac_id,fac_name,course,pass percentage

4.Create table-department and attributes are dept-no,dept_name,section,no of students ,pass percentage

5.Create table-activity and attributes are act_id,act_name,no of students opted,category of activity

```
3 • CREATE TABLE student(studid integer,studname varchar(50),gender varchar(20),subject varchar(20),marks integer,grade varchar(10));
4 • desc student;
5 • CREATE TABLE faculty(fac_id integer,fac_name varchar(20),course varchar(10),passpercentage integer);
6 • desc faculty;
7 • CREATE TABLE department(dept_no integer,dept_name varchar(20),section varchar(10),noofstudents integer, passpercentage integer);
8 • desc department;
9 • CREATE TABLE activity(act_id integer,act_name varchar(20),noofstudentsopted integer,rateofactivity varchar(20));
```

19 • desc faculty;

Field	Type	Null	Key	Default	Extra
fac_id	int	NO	PRI	NULL	
fac_name	varchar(20)	YES		NULL	
course	varchar(10)	YES		NULL	
passpercentage	int	YES		NULL	
total_mentor_details	varchar(100)	YES		NULL	

Student table:

Field	Type	Null	Key	Default	Extra
studid	int	NO	PRI	NULL	
studname	varchar(50)	YES		NULL	
gender	varchar(10)	YES		NULL	
subject	varchar(20)	YES		NULL	
marks	int	YES		NULL	
grade	varchar(10)	YES		NULL	
address	varchar(50)	YES		NULL	
course	varchar(20)	YES		NULL	

Activity – 2

1.Add address in student table ,change the datatype size for student name and make studid as primary key

```
3 • ALTER TABLE student ADD(address varchar(50));
4 • ALTER TABLE student MODIFY studname varchar(50);
5 • Alter table student add(primary key(studid));
```

2.Add faculty total mentor details ,make4 fac_id as primary key

```
17 • ALTER TABLE faculty ADD(total_mentor_details varchar(100));
18 • Alter table faculty add(primary key(fac_id));
```

3.add no of students in wise and make dept no as primary key

```
20 • ALTER TABLE department ADD(no_of_students_in_wise integer);
21 • Alter table department add(primary key(dept no));
```

4.Add faculty name and change the size of act_name and act_id as primary key

```
22 • ALTER TABLE activity ADD(faculty_name varchar(30));
24 • ALTER TABLE activity MODIFY act_name varchar(30);
25 • Alter table activity add(primary key(act_id));
```

Experiment-6

Activity – 3

1.Insert 5 instances in each table and display the result

```
28 • Insert into student values(1,'arun','M','DBMS',80,'A','x');
29 • Insert into student values(2,'haril','M','DBMS',80,'A','y');
30 • Insert into student values(3,'poojish','M','DBMS',85,'A','z');
31 • Insert into student values(4,'srest','M','DBMS',50,'B','a');
32 • Insert into student values(5,'vikash','M','DBMS',75,'B','b');
33 • Select * from student;
34 • Insert into faculty values(1,'a','a','x','80','abc');
35 • Insert into faculty values(2,'b','b','y','80','bby');
36 • Insert into faculty values(3,'c','c','z','80','ccz');
37 • Insert into faculty values(4,'d','d','u','95','dki');
38 • Insert into faculty values(5,'e','e','v','80','eev');
39 • Select * from faculty;
40 • Insert into department values(1,'CS',1,'A',80,80);
41 • Insert into department values(2,'CS',2,'B',80,80,50);
```

Result Grid

studid	studname	gender	subject	marks	grade	address	course
1	arun	M	DBMS	80	A	x	DBMS
2	haril	M	DBMS	80	A	y	DBMS
3	poojish	M	DBMS	85	A	z	DBMS
4	srest	M	DBMS	50	B	a	DBMS
5	vikash	M	DBMS	75	B	b	DBMS

```
34 • Insert into faculty values(1,'a','a','x','80','abc');
35 • Insert into faculty values(2,'b','b','y','80','bby');
36 • Insert into faculty values(3,'c','c','z','80','ccz');
37 • Insert into faculty values(4,'d','d','u','95','dki');
38 • Insert into faculty values(5,'e','e','v','80','eev');
39 • Select * from faculty;
40 • Insert into department values(1,'CS',1,'A',80,80);
41 • Insert into department values(2,'CS',2,'B',80,80,50);
```

Result Grid

fac_id	fac_name	course	percentage	total_mentor_details
1	aa	x	80	abc
2	bb	y	80	bby
3	cc	z	80	ccz
4	dd	u	95	dki
5	ee	v	80	eev

```

40 * Insert into department values(1,'CSE','A',47,85,60);
41 * Insert into department values(2,'CSE','B',49,82,55);
42 * Insert into department values(3,'CSE','C',52,83,50);
43 * Insert into department values(4,'IT','A',46,80,54);
44 * Insert into department values(5,'IT','B',42,82,57);
45 * Select * from department;
46 * Insert into activity values(101,'act1','09','c1','fac1');

```

dept_no	dept_name	section	no_of_students	percentage	no_of_students_p_name
1	CSE	A	47	85	60
2	CSE	A	49	82	55
4	IT	A	46	80	54
5	IT	B	42	82	57

```

46 * Insert into activity values(101,'act1','09','c1','fac1');
47 * Insert into activity values(102,'act2','10','c2','fac2');
48 * Insert into activity values(103,'act3','10','c3','fac3');
49 * Insert into activity values(104,'act4','10','c4','fac4');
50 * Insert into activity values(105,'act5','17','c5','fac5');
51 * Select * from activity;
52 * Select studid,marks from student;

```

act_id	act_name	no_of_students	category	faculty_name
101	act1	60	c1	fac1
102	act2	55	c2	fac2
103	act3	50	c3	fac3
104	act4	54	c4	fac4
105	act5	57	c5	fac5

2.Display student no,marks from student table

```

52 * Select studid,marks from student;
53 * Select fac_id,fac_name from faculty;
54 #activity4
55 * Select * from student;
56 * Select * from student where grade='A';
57 * Select * from student where marks<50;
58 * Select studid,studname from student;

```

studid	marks
1	90
2	80
3	85
4	90
5	75

3.Display faculty no,name from faculty table

```

69 * Select s.studid,f.fac_id from student s,faculty f where s.course=f.course;
70 #activity 5
71 * Update student set marks=50 where studid=1;

```

studid	fac_id
2	2
3	3
4	4
5	5

Experiment-7

Activity – 4

1.Display 1 to 5 students details

studid	studname	gender	subject	marks	grade
1	ashu	F	DBMS	10	A
2	bindu	F	DBMS	80	A
3	shivani	F	DBMS	85	A
4	sai	M	DBMS	60	B
5	vardhan	M	DBMS	75	B+

2.Display who got grade A

```

56 * Select * from student where grade='A';
57 * Select * from student where marks<50;
58 * Select studid,studname from student;
59 * Select studid,studname from student where marks BETWEEN 50 AND 60 and gender='M';
60 * Select * from student where marks<70;

```

studid	studname	gender	subject	marks	grade	address	course
1	aran	M	DBMS	90	A	x	DBMS
2	hari	M	DBMS	80	A	y	y
3	poojith	M	DBMS	85	A	z	z

3. Display whose marks is less than 50

```

57 * Select * from student where marks<50;
58 * Select studid,studname from student;
59 * Select studid,studname from student where marks BETWEEN 50 AND 60 and gender='M';
60 * Select * from student where marks<70;

```

studid	studname	gender	subject	marks	grade	address	course
--------	----------	--------	---------	-------	-------	---------	--------

4.Display student id and name

```

61 * Select studid,studname from student;

```

studid	studname
--------	----------

5.Display the student id and name whose mark is 50 to 60and female

```

61 * Select studid,studname from student;

```

studid	studname
--------	----------

6.Display the list of students who gets greater than 70

```

60 * Select * from student where marks>70;
61 * Select * from student;
62 * Select a.act_id,a.act_name from activity a;
63 * Alter table student add(course varchar(20));
64 * Update student set course='x' where studid=1;

```

studid	studname	gender	subject	marks	grade	address	course
1	aran	M	DBMS	90	A	x	DBMS
2	hari	M	DBMS	80	A	y	y
3	poojith	M	DBMS	85	A	z	z
4	prath	M	DBMS	50	B	a	u
5	ahil	M	DBMS	75	B+	b	v

7.Delete the failure students

Display complete table

```

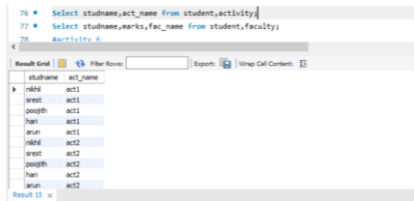
61 * Select * from student;
62 * Select a.act_id,a.act_name from activity a;
63 * Alter table student add(course varchar(20));
64 * Update student set course='x' where studid=1;
65 * Update student set course='y' where studid=2;
66 * Update student set course='z' where studid=3;

```

studid	studname	gender	subject	marks	grade	address	course
1	aran	M	DBMS	90	A	x	DBMS
2	hari	M	DBMS	80	A	y	y
3	poojith	M	DBMS	85	A	z	z
4	prath	M	DBMS	50	B	a	u
5	ahil	M	DBMS	75	B+	b	v

8.Display activity id,name using object

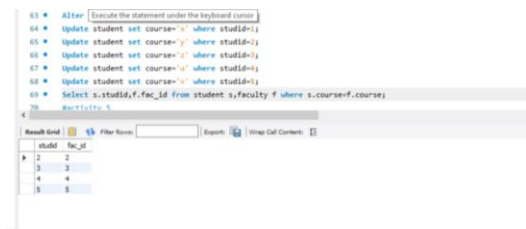
```
76 * select studname,act_name from student,activity;
77 * select studname,marks,fac_name from student,faculty;
78 * Activity 6
```



studname	act_name
rohit	act1
arav	act1
poorvi	act1
hari	act1
arun	act1
rohit	act2
arav	act2
poorvi	act2
hari	act2
arun	act2

9.Add course to student table then insert values .Display student id,faculty id using course name condition with object

```
63 * Alter [Execute the statement under the keyboard cursor]
64 * Update student set course='i' where studid=1;
65 * Update student set course='j' where studid=1;
66 * Update student set course='i' where studid=1;
67 * Update student set course='j' where studid=1;
68 * Update student set course='i' where studid=1;
69 * Select s.studid,f.fac_id from student s,faculty f where s.course=f.course;
70 * Activity 5
```



studid	fac_id
2	2
3	3
4	4
5	5

Activity – 5

1.change mark to 50 whose id is 4

```
71 * Update student set marks=50 where studid=4;
```

2.change name whose id is 3

```
72 * Update student set studname='poorvi' where studid=3;
```

3. change activity name whose id between 5 to 7

4.change department section to A whose id is less than 5

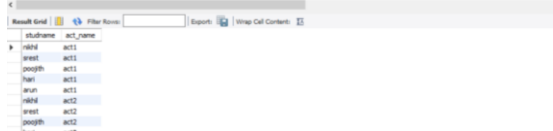
```
73 * Update activity set act_name='abc' where act_id BETWEEN 5 AND 7;
74 * Update department set section='A' where dept_no=3;
```

5.delete row who has id = 3

```
75 * Delete from department where dept_no=3;
```

6.select student name and activity name from student and activity table


```
76 * Select studname,act_name from student,activity;
77 * Select studname,marks,fac_name from student,faculty;
78 *activity 6
```



studname	act_name
nhdh	act1
wrest	act1
jooqph	act1
han	act1
arun	act1
nhdh	act2
wrest	act2
jooqph	act2

7.select student name and mark from student and faculty name from faculty

```
-- main
77 * Select studname,marks,fac_name from student,faculty;
78 *activity 6
79 * Select dept_name,fac_name from department,faculty where dept_name LIKE 'ck' and fac_name LIKE 'M';
80 * Select act_name from activity where act_name LIKE 'act5';
81 * Select studname from student ORDER BY studid DESC;
82 * Select fac_name from faculty ORDER BY fac_name ASC;
83 * Select studname from student ORDER BY grade;
```




studname	marks	fac_name
nhdh	75	aa
wrest	50	aa
jooqph	85	aa
han	80	aa
arun	90	aa
nhdh	75	bb
wrest	50	bb
jooqph	85	bb
han	80	bb
arun	90	bb

Activity – 6

1.select department starts from 'c' and faculty name ends with "


```
79 * Select dept_name,fac_name from department,faculty where dept_name LIKE 'ck' and fac_name LIKE 'M';
80 * Select act_name from activity where act_name LIKE 'act5';
81 * Select studname from student ORDER BY studid DESC;
82 * Select fac_name from faculty ORDER BY fac_name ASC;
83 * Select studname from student ORDER BY grade;
```



dept_name	fac_name
-----------	----------

2.select activity having characters between 'ck'

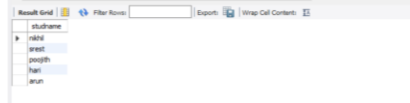
```
80 * Select act_name from activity where act_name LIKE 'act5';
81 * Select studname from student ORDER BY studid DESC;
82 * Select fac_name from faculty ORDER BY fac_name ASC;
83 * Select studname from student ORDER BY grade;
```



act_name
act1
act2
act3
act4
act5

3.display students list descending order of student id

```
81 * Select studname from student ORDER BY studid DESC;
82 * Select fac_name from faculty ORDER BY fac_name ASC;
83 * Select studname from student ORDER BY grade;
```

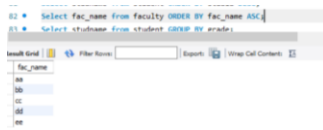


studname
nhdh
wrest
jooqph
han
arun

4.display faculty name ascending order

```
82 • Select fac_name from faculty ORDER BY fac_name ASC;
```

```
83 • Select studname from student GROUP BY grade;
```

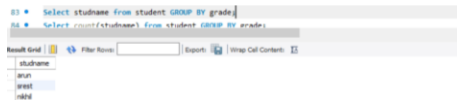


fac_name
aa
bb
cc
dd
ee

5.display students list based on grade

```
83 • Select studname from student GROUP BY grade;
```

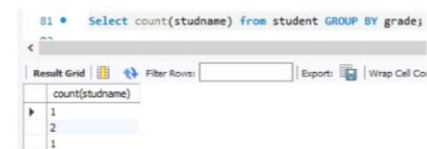
```
84 • Select count(studname) from student GROUP BY grade;
```



studname
anun
asad
asad

6.display students having grade a using group by

```
81 • Select count(studname) from student GROUP BY grade;
```

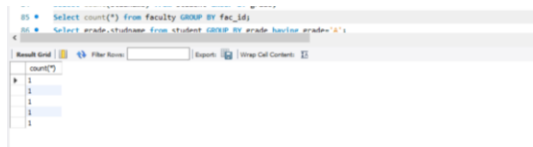


count(studname)
1
2
1

7.group by faculty id and display

```
85 • Select count(*) from faculty GROUP BY fac_id;
```

```
86 • Select grade,studname from student GROUP BY grade having grade='A';
```

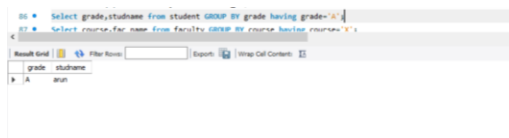


count(*)
1
1
1
1

8.display the students list whose grade is A using having

```
85 • Select grade,studname from student GROUP BY grade having grade='A';
```

```
86 • Select course,fac_name from faculty GROUP BY course having course='X';
```

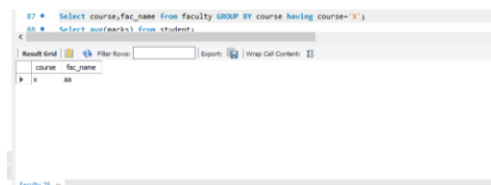


grade	studname
A	anun

9.display the faculty list who are teaching subject pps

```
87 • Select course,fac_name from faculty GROUP BY course having course='X';
```

```
88 • Select avg(marks) from student;
```



course	fac_name
X	aa

faculty 25

10.Apply aggregate functions in students marks- min,max,sum,count,avg

```
00 • Select avg(marks) from student;
```

```
01 • Select min(marks) from student;
```

avg(marks)
76.0000

```
00 • Select max(marks) from student;
```

```
01 • Select min(marks) from student;
```

max(marks)
90

```
01 • Select sum(marks) from student;
```

```
02 • Select count(marks) from student;
```

sum(marks)
380

```
02 • Select count(marks) from student;
```

count(marks)
5

Experiment-8

1.create a tables for sailors,boats and reserves

```
1 • CREATE SCHEMA sailors_boats;
```

```
2 • CREATE TABLE boat(bid integer,bname varchar(30),color varchar(20));
```

```
3 • CREATE TABLE sailor(sid integer,sname varchar(30),rating integer,age real);
```

```
4 • CREATE TABLE reserves(sid integer,bid integer,DAY DATE);
```

```
5 • desc sailor;
```

```
6 • desc boat;
```

Field	Type	Null	Key	Default	Extra
sid	int	YES		NULL	
sname	varchar(30)	YES		NULL	
rating	int	YES		NULL	
age	double	YES		NULL	

Field	Type	Null	Key	Default	Extra
bid	int	YES		NULL	
bname	varchar(30)	YES		NULL	
color	varchar(20)	YES		NULL	

Field	Type	Null	Key	Default	Extra
sid	int	YES		NULL	
bid	int	YES		NULL	
DAY	date	YES		NULL	

2.Insert 5 values in each table

Sailor table:

```
8 • Insert into sailor values(22,'Dustin',7,45.0);
```

```
9 • Insert into sailor values(29,'Brutus',1,33.0);
```

```
10 • Insert into sailor values(31,'Lubber',8,55.5);
```

```
11 • Insert into sailor values(32,'Andy',8,25.5);
```

```
12 • Insert into sailor values(58,'Rusty',10,35.0);
```

Boat table:

```

22 • Insert into boat values(101,'Interlake','blue');
23 • Insert into boat values(102,'Interlake','red');
24 • Insert into boat values(103,'Clipper','green');
25 • Insert into boat values(104,'Marine','red');

```

Reserve table:

```

14 • Insert into reserves values(22,101,'98/10/10');
15 • Insert into reserves values(22,102,'98/10/10');
16 • Insert into reserves values(31,102,'98/10/11');
17 • Insert into reserves values(64,101,'98/5/9');
18 • Insert into reserves values(74,103,'98/8/9');

```

3.Display all records

Select * from sailor;

sid	sname	rating	age
22	Dustin	7	45
29	Brutus	1	33
31	Lubber	8	55.5
32	Andy	8	25.5
58	Rusty	10	35

26 • Select * from boat;

bid	bname	color
101	Interlake	blue
102	Interlake	red
103	Clipper	green
104	Marine	red

16 • Select * from reserves;

sid	bid	DAY
22	101	1998-05-09
22	102	1998-10-11
31	102	1998-10-11
64	101	1998-05-09

4.find the names and ages of the sailors

31 • Select sname,age from sailor;

sname	age
Dustin	45
Brutus	33
Lubber	55.5
Andy	25.5
Rusty	35

5.find all the sailors with a rating above 8

33 • Select sid from sailor where rating>=8;

sid
31
32
58

6.find all sailors name with rating above 7 & age above 25

35 • `Select sname from sailor where rating > 7 union Select sname from sailor where age > 30`

sname
Lubber
Andy
Rusty
Dustin

7.Display all the names and colors of the boat

37 • `Select bname,color from boat;`

bname	color
Interlake	blue
Interlake	red
Copper	green
Marine	red

8.Find all the boats with red color

39 • `Select bname,bid from boat where color = 'red';`

bname	bid
Interlake	102
Marine	104

9.find the names of sailors who have reserved boat number 101

41 • `Select sname from sailor S,reserves R where S.sid=R.sid and bid = 101`

sname
Dustin

10.find the sids of sailor who have reserved pink boat

43 • `Select R.sid from boat B,reserves R where B.bid=R.bid and B.color = 'pink'`

R.sid
101

11.find names of sailors who have reserved red boat

45 • `Select S.sname from sailor S,reserves R,boat B where S.sid = R.sid and R.bid=B.bid and B.color='red'`

S.sname
Dustin

12.find the colors of boat reserved by same name(provide any name in the table)



13.find the names of the sailors who have atleast one boat



14.find the names of the sailor who have reserved two different boats



15.find the names of the sailors who have reserved a red or a green boat



16.find the names of sailors who have reserved both a red and a green boat



17.find the names of sailors who have reserved boat 3

```
59 * select S.sname from sailor S where S.sid in (select R.sid from reserves R where R.bid
60      (select B.bid from boat B where B.color='red'));
```

19.find the names of sailors who have not reserved a red boat(nq)

```
62 * select S.sname from sailor S where S.sid not in (select R.sid from reserves R where R
63      not in (select B.bid from boat B where B.color='red'));
```

20.find the names of sailors who have reserved boat number 103(exists)

```
61 * select S.sname from sailor S where exists (select * from reserves R where R.bid=103 and S.sid=R.s
```

21.find the sailors whose rating is better than some sailor called name

```
63 * select * from sailor S where S.rating>(select s2.rating from sailor s2 where s2.sname='Dusti')
```

22.find sailors whose rating is better than every sailor called name

```
65 * select * from sailor S where S.rating>all(select s2.rating from sailor s2 where s2.sname='Dusti')
```

23.find the sailors with the highest rating.

```
67 * select * from sailor S where S.rating >=all(select s2.rating from sailor s2);
```

24.find the average age of all sailors

```
69 * select avg(S.age) from sailor S;
```

25.find the average age of sailors with a rating of 10.

```
71 • select avg(S.age) from sailor S where S.rating=10;
```

avg(S.age)
35

26.count the number of sailors

```
73 • select count(*) from sailor S;
```

count(*)
5

27.count the number of different sailor ratings

```
73 • select count(*) from sailor S;
```

count(*)
5

28.find the name and age of the oldest sailor

```
77 • select S.sname, S.age from sailor S where (select max(S2.age) from sailor S2)->S
```

sname	age
Lubber	55.5

29.find the name of sailors who are older than the oldest sailor with a rating of 10.

```
79 • select S.sname from sailor S where S.age>(select max(S2.age) from sailor S2 where S2.rating=10);
```

sname
Quinn
Lubber

30.find the age of the youngest sailor for each rating level.

```
81 • select S.rating,min(S.age) from sailor S group by S.rating;
```

rating	min(S.age)
7	45
1	33
8	25.5
10	35

31.find the age of the youngest sailor who is eligible to vote(l.e.,atleast 18 years old)for each rating level with atleast two such sailors.

```
83 • select S.rating,min(S.age) from sailor S where S.age>=18 group by S.rating having count(*) >= 2
```

rating	min(S.age)
8	25.5

32.for each red boat,find the number of reservation for this boat.

```
85 • select R.boat,count(*) as reservationcount from boat R, reserves R1 where R.boat=R.boat and R.colors='red' group by R.boat
```

boat	reservationcount
101	2

33.find all sailors name according to names.

```
87 • select S.sname from sailor S group by S.sname
```

sname
Dustin
Brutus
Lubber
Andy
Rusty

34.find all sailors details according to rating

```
89 • select * from sailor S group by S.rating
```

sid	sname	rating	age
22	Dustin	7	45
29	Brutus	1	33
31	Lubber	8	55.5
58	Rusty	10	35

35.find all sailors details according to rating (highest first),if ratings are same then according to age.

```
91 • select * from sailor S order by rating desc,age asc
```

sid	sname	rating	age
58	Rusty	10	35
32	Andy	8	25.5
31	Lubber	8	55.5
22	Dustin	7	45
29	Brutus	1	33

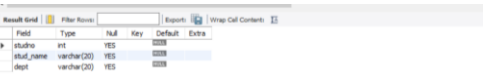
Experiment-9

Views

1.create a table student attributes
stdno,name,dept

2.Display

```
1 • create table students(stdno integer,std_name varchar(20),dept varchar(20));
2 • desc students;
```



Field	Type	Null	Key	Default	Extra
stdno	int	YES			
std_name	varchar(20)	YES			
dept	varchar(20)	YES			

3.create view and select

```
3 • create view stud_view as select stdno,std_name,dept from students;
4 • select * from stud_view;
```



stdno	std_name	dept
-------	----------	------

4.Insert values

```
5 • Insert into stud_view values(549,'poorja','CSE');
6 • Insert into stud_view values(512,'poorja','CSE');
7 • Insert into stud_view values(558,'Akshaya','CSE');
8 • alter view stud_view as select stdno,std_name,dept from students where stdno<512;
```



stdno	std_name	dept
512	Pooja	CSE

5.create view with check option

Insert values and alter tables

```
8 • alter view stud_view as select stdno,std_name,dept from students where stdno<512;
9 • create view student_view as select stdno,std_name,dept from students with check option;
10 • Insert into student_view values(555,'harika','CSE');
11 • alter view student_view as select stdno,std_name,dept from students where stdno<555;
12 • drop view student_view;
13 • delete from stud_view where stdno<555;
```

Experiment-10

Triggers

1.create a table with attribute sname and
another table with newname

```
1 * Create table test1(sname varchar(20));  
2 * Create table test_audit1(newname varchar(20));
```

2.create a trigger to insert second table before
inserting the first table

```
1 * Create trigger before_test_update  
2 * before insert on test1  
3 * for each row insert into test_audit1  
4 * set newname = new.sname;  
5 * Insert into test1 values ('maria');  
6 * select * from test1;  
7 * select * from test_audit1;
```

Result Grid	Filter Rows	Export	Wrap Cell Content
sname			
maria			

- Create a table for account details with
attributes account number and an amount

```
14 * Create trigger before_insert_  
15 * before insert on acc_details  
16 * for each row  
17 * begin  
18 * if (new.amount < 0) then  
19 * set new.amount = 0;  
20 * elseif (new.amount > 500) then  
21 * set new.amount = 500;  
22 * end if;  
23 * end //  
24 * delimiter ;
```

```
25 * Insert into acc_details values (100, -14.99);  
26 * Select * from acc_details;
```

Result Grid	Filter Rows	Export	Wrap Cell Content
accno amount			
100 0.00			

```
27 * Insert into acc_details values (200, 500.00);  
28 * Select * from acc_details;
```

Result Grid	Filter Rows	Export	Wrap Cell Content
accno amount			
100 0.00			
200 500.00			

```
29 * Insert into acc_details values (300, -14.99);  
30 * Select * from acc_details;
```

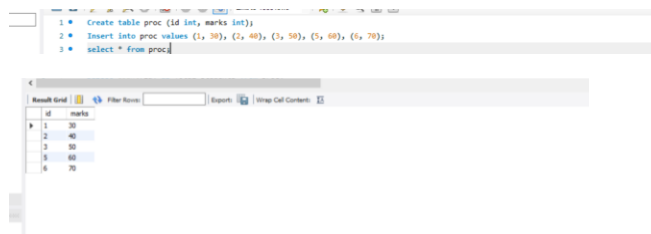
Result Grid	Filter Rows	Export	Wrap Cell Content
accno amount			
100 0.00			
200 500.00			
300 0.00			

Experiment-11

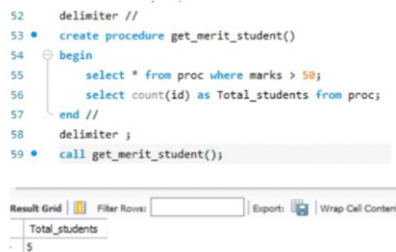
Procedures

1. Create a table with attributes students and marks

Insert values into tables



- Create a procedure to get the merit students(marks>50)
- Create a procedure to get marks of a given id(using in)



The screenshot shows a SQL editor with the following code:

```
61 delimiter //
62 • create procedure get_marks(IN stdid int)
63 begin
64     select marks from proc where (id = stdid);
65 end //
66 delimiter ;
67 • call get_marks(6);
```

Below the code is a 'Result Grid' showing the output of the procedure:

id	marks
5	60
6	70



Result Grid	Filter Rows: <input type="text"/>	Exports:	Wrap Cell Cont
marks			
70			

Experiment-12

Cursors

- Create a table with attributes students and marks
- Insert values into tables

```
CREATE TABLE curs(id int, marks int);
INSERT INTO curs (id,marks)VALUES(1,30),(2,40),(3,50),(5,60),
DELIMITER \\\
```

- Create a procedure and fetch the marks of given id using a cursor

```
71 DELIMITER \\\
72 • CREATE PROCEDURE cur_marks(id1 int)
73 BEGIN
74     DECLARE m1 int;
75     DECLARE cur1 CURSOR FOR SELECT marks FROM curs WHERE id = :
76     OPEN cur1;
77     FETCH cur1 INTO m1;
78     SELECT m1;
79     CLOSE cur1;
80 END \\\
81 DELIMITER ;
82 • CALL cur_marks(2);
```

Result Grid
m1
40

- Create a procedure and fetch the highest marks using a cursor

```
84 DELIMITER \\\
85 • CREATE PROCEDURE cur_mark1()
86 BEGIN
87     DECLARE m1 int;
88     DECLARE cur1 CURSOR FOR SELECT max(marks) FROM curs
89     OPEN cur1;
90     FETCH cur1 INTO m1;
91     SELECT m1;
92     CLOSE cur1;
93 END \\\
94 DELIMITER ;
95 • CALL cur_mark1();
```

Result Grid	
	m1
►	70