DBMS ASSIGNMENT-3

Sailors(sid: integer, sname: string, rating: integer, age: real)

Boats(bid: integer, bname: string, color: string)

Reserves(sid: integer, bid: integer, day: date)

sid	sname	rating	age
22	Dustin	7	45.0
29	Brutus	1	33.0
31	Lubber	8	55.5
32	Andy	8	25.5
58	Rusty	10	35.0
64	Horatio	7	35.0
71	Zorba	10	16.0
74	Horatio	9	35.0
85	Art	3	25.5
95	Bob	3	63.5

sid	bid	day
22	101	10/10/98
22	102	10/10/98
22	103	10/8/98
22	104	10/7/98
31	102	11/10/98
31	103	11/6/98
31	104	11/12/98
64	101	9/5/98
64	102	9/8/98
74	103	9/8/98

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

EXERCISE-9: (NESTED QUERIES)

1.Increments for the ratings of persons who have sailed two

different boats on the same day.

mysql > select rating+1 as rating from Sailors where sid in(select sid from Reserves where day in(select day from Reserves group by day having count(day)=2) group by sid having count(sid)=2);

+-	-		-			T
 -	r	at	i	ng		
+-	-		-			۱
 -				8	;	
4-	_		_			٠

2.Ages of sailors whose name begins and ends with 'B' and has at least 3 characters

```
mysql > select age from Sailors where sid in(select sid from Sailors where name like 'B %B');
+----+
| age |
+----+
| 63.5 |
```

3.Find the sids of all sailors who have reserved red boats but not green boats.

```
select sid from Reserves where bid in(select bid from boat where
color='red') except select sid from Reserves where bid in(select bid from
boat where color='green');
Dutput
sid
64
```

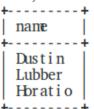
4.All the sailors who have a rating of 10 or have reserved boat 104.

mysql > select name from Sailors where sid in(select sid from Sailors where rating=10) UNION select name from Sailors where sid in(select sid from Reserves where bid=104);

```
| name |
| Rusty |
| Zorba
| Dustin
| Lubber
```

5. write a nested query to Find the names of sailors who have reserved boat 103.

mysql > select name from Sailors where sid in(select sid from Reserves where bid= 103);



6.write a nested query to Find the names of sailors who have reserved a

red boat.

mysql > select name from Sailors where sid in(select sid from Reserves where bid in(select bid from boat where color='red'));

```
| name |
| Dustin |
| Lubber
| Horatio
```

7.write a nested query to Find the names of sailors who have not reserved a red boat.

mysql > select name from Sailors where sid in(select sid from Reserves where sid not in(select distinct sid from Reserves where bid in(select bid from boat where color='red')));

```
| name |
| Horatio |
```

8. Nested query to Find sailors whose rating is better than some sailor called Horatio.

mysql>select name from Sailors where rating>some(select rating from Sailors where name='Horatio');

```
| name |
| Lubber
| Andy
| Rusty
| Zorba
| Hbratio
```

9. Nested query to find the sailors with highest rating.

mysql > select name from Sailors where rating=(select max(rating) from Sailors);

```
+----+
| name |
+----+
| Rusty |
| Zorba |
```

10. Find the names of sailors who have reserved all boats.

mysql > select name from Sailors where sid in(select sid from Reserves group by sid having count(sid) = 4);

```
name |
Dustin |
```

EXERCISE: 10 (Aggregation Functions)

1.Average age of Sailors

2. Average age of sailors with a rating of 10.

mysql > select avg(age) from Sailors where rating=10;

```
avg(age) |
| 25.5 |
```

3.Name and age of the oldest sailor

mysql > select name, age from Sailors where age=(select max(age) from Sailors);

4.Count the number of sailors.

nysql > select count(name)from Sailors;

```
count (nane) |
```

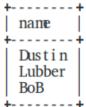
5.number of different sailor names.

+----+

mysql > select count(distinct name) from Sailors; +-----+ | count(distinct name) | +-----+ | 9 |

6.Find the names of sailors who are older than the oldest sailor with a rating of 10.

mysql>select name from Sailors where age≾select age from Sailors where age=(se lect max(age)from Sailors where rating=10) and rating=10);



7. Find the age of the youngest sailor for each rating level.

mysql > select rating, min(age) as age from Sailors group by rating;

-		<u></u>
į	rating	age
j	1 3 7 8 9 10	33 25. 5 35 25. 5 35 16

8.Find the age of the youngest sailor who is eligible to vote (i.e., is at least 18 years old) for each rating level with at least two such sailors.

mysql>select rating, min(age) as age from Sailors where age >= 18 group by rating having count(name) >= 2;

+	+
rating	
3 7 8	25. 5 35 25. 5

9.For each red boat, find the number of reservations for this boat.

mysql > select distinct r.bid, count(day) from Reserves as r inner join boat as b on r.bid=b.bid and b.color='red' group by r.bid;

bi d	count (day)
102	3
104	2

10.Find the average age of sailors for each rating level that has at least two sailors.

mysql > select rating, avg(age) from Sailors group by rating having count(name) >= 2

+	+
rating	avg(age)
3 7	44. 5 40
8	40. 5
10	25. 5
+	

11.Find the average age of sailors who are of voting age (i.e., at least 18 years old) for each rating level that has at least two sailors.

mysql > select rating, avg(age) as avg_age from Sailors as s1 where age >= 18 group by rating having 2 <= (select count(*) From Sailors as s2 where s1. rating=s2. rating);

rating	i
3	44. 5
7	40
8	40. 5
10	35

12.Find the average age of sailors who are of voting age (i.e., at least 18 years old) for each rating level that has at least two such sailors.

nysql>select rating, avg(age) as avg_age from Sailors as s1 where age ≥18 group by rating having 2<=(select count(*) from Sailors as s2 where s1.rating=s2.rating and s2.age ≥18);

+	++
rating	avg_age
3	44.5
8	40 40. 5
4	

13.Find those ratings for which the average age of sailors is the minimum over all ratings.

mysql > select s.rating, avg(s.age) as avg age from Sailors as s group by s.rating having avg age <=ALL(select avg(s2.age) from Sailors as s2 group by s2.rating);

+	- +
rating	avg_age
10	25. 5
+	

[NOTE: SINCE EXCEPT KEYWORD IS NOT WORKING IN TERMINAL ONLINE COMPILER IS USED]