DSAA 5012
Advanced Data Management for Data Science

LECTURE 7 EXERCISES
STRUCTURED QUERY LANGUAGE (SQL)

EXAMPLE RELATIONAL SCHEMA AND DATABASE

Sailor(<u>sailorId</u>, sName, rating, age)

Boat(<u>boatId</u>, bName, color)

Reserves(<u>*sailorId*, *boatId*, rDate</u>)

Attribute names in italics are foreign key attributes.

Sailor

<u>sailorId</u>	sName	rating	age
22	Dustin	7	45
29	Brutus	1	33
31	Lubber	8	55
32	Andy	8	25
58	Rusty	10	35
64	Horatio	7	35
71	Zorba	10	16
74	Horatio	9	35
85	Art	3	25
95	Bob	3	63
99	Chris	10	30

11 tuples

Reserves

<u>*sailorId*</u>	<u>*boatId*</u>	<u>rDate</u>
22	101	10/10/17
22	102	10/10/17
22	103	08/10/17
22	104	07/10/17
31	102	10/11/17
31	103	06/11/17
31	104	12/11/17
64	101	05/09/17
64	102	08/09/17
74	103	08/09/17
99	104	08/08/17

11 tuples

Boat

<u>boatId</u>	bName	color
101	Interlake	blue
102	Interlake	red
103	Clipper	green
104	Marine	red
105	Serenity	Cyan

5 tuples

EXERCISE 1

Find the boat name and the number of reservations for each red boat.

Sailor

sailorId	sName	rating	age
22	Dustin	7	45
29	Brutus	1	33
31	Lubber	8	55
32	Andy	8	25
58	Rusty	10	35
64	Horatio	7	35
71	Zorba	10	16
74	Horatio	9	35
85	Art	3	25
95	Bob	3	63
99	Chris	10	30

11 tuples

Reserves

sailorId	boatId	rDate
22	101	10/10/17
22	102	10/10/17
22	103	08/10/17
22	104	07/10/17
31	102	10/11/17
31	103	06/11/17
31	104	12/11/17
64	101	05/09/17
64	102	08/09/17
74	103	08/09/17
99	104	08/08/17

11 tuples

Boat

boatId	bName	color
101	Interlake	blue
102	Interlake	red
103	Clipper	green
104	Marine	red
105	Serenity	Cyan

5 tuples

EXERCISE 1

Find the boat name and the number of reservations for each red boat.

☞ **(Interlake, 3), (Marine, 3)**

Is this a correct solution?

```
select bName, count(*) as reservationCount
from Boat natural join Reserves
where color='red'
group by boatId;
```

Illegal!!!
Why?

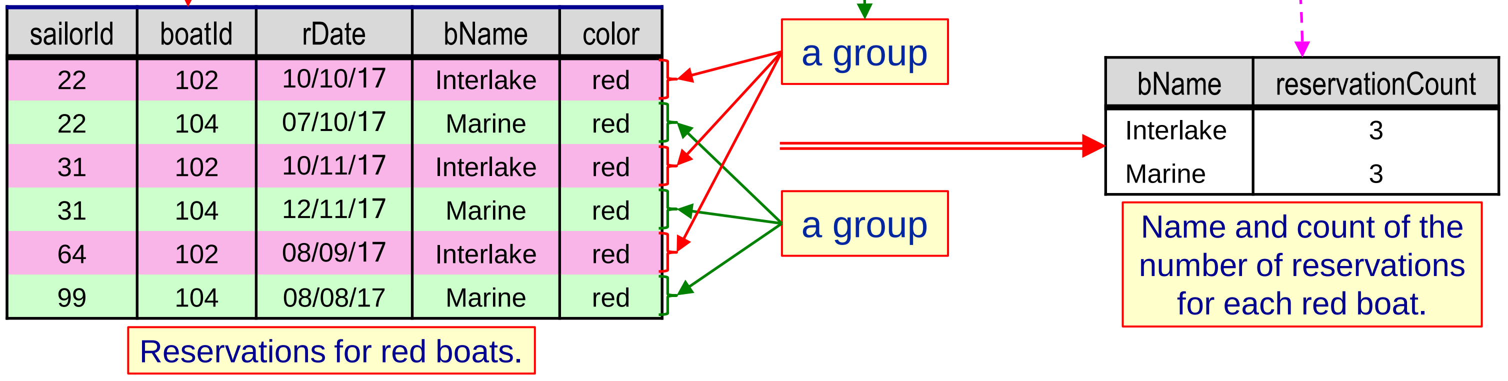
☞ **All non-aggregate attributes in the select clause <u>must</u> appear in the group by clause (i.e., bName must appear in the group by clause).**

EXERCISE 1 (cont'd)

Find the boat name and the number of reservations for each red boat.

☞ **(Interlake, 3), (Marine, 3)**

```
select bName, count(*) as reservationCount
from Boat natural join Reserves
where color='red'
group by bName, boatId;
```



sailorId	boatId	rDate	bName	color
22	102	10/10/17	Interlake	red
22	104	07/10/17	Marine	red
31	102	10/11/17	Interlake	red
31	104	12/11/17	Marine	red
64	102	08/09/17	Interlake	red
99	104	08/08/17	Marine	red

Reservations for red boats.

bName	reservationCount
Interlake	3
Marine	3

Name and count of the number of reservations for each red boat.

EXERCISE 1 (cont'd)

Find the boat name and the number of reservations for each red boat.

☞ **(Interlake, 3), (Marine, 3)**

Do you see any problems with this solution?

```
select bName, count(*) as reservationCount
from Boat natural join Reserves
where color='red'
group by bName
```

sailorId	boatId	rDate	bName	color
22	102	10/10/17	Interlake	red
22	104	07/10/17	Marine	red
31	102	10/11/17	Interlake	red
31	104	12/11/17	Marine	red
64	102	08/09/17	Interlake	red
99	104	08/08/17	Marine	red

Reservations for red boats.

a group

a group

bName	reservationCount
Interlake	3
Marine	3

Name and count of the number of reservations for each red boat.

EXERCISE 1 (cont'd)

Suppose we change the query to this.

Find the boat name and the number of reservations for each boat.

What is the result?

```
select bName, count(*) as reservationCount
from Boat natural join Reserves
group by bName
```

sailorId	boatId	rDate	bName	color
22	101	10/10/17	Interlake	blue
64	101	05/09/17	Interlake	blue
22	102	10/10/17	Interlake	red
31	102	10/11/17	Interlake	red
64	102	08/09/17	Interlake	red
22	103	08/10/17	Clipper	green
31	103	06/11/17	Clipper	green
74	103	08/09/17	Clipper	green
22	104	07/10/17	Marine	red
31	104	12/11/17	Marine	red
99	104	08/08/17	Marine	red

a group

a group

a group

bName	reservationCount
Interlake	5
Clipper	3
Marine	3

Since bName is not unique, grouping on it can get an incorrect result!

EXERCISE 1 (cont'd)

Find the boat name and the number of reservations for each boat.

Correct solution.

```
select bName, count(*) as reservationCount
from Boat natural join Reserves
group by bName, boatId;
```

Recall: attributes in the **group by** clause do not have to appear in the select clause.

sailorId	boatId	rDate	bName	color
22	101	10/10/17	Interlake	blue
64	101	05/09/17	Interlake	blue
22	102	10/10/17	Interlake	red
31	102	10/11/17	Interlake	red
64	102	08/09/17	Interlake	red
22	103	08/10/17	Clipper	green
31	103	06/11/17	Clipper	green
74	103	08/09/17	Clipper	green
22	104	07/10/17	Marine	red
31	104	12/11/17	Marine	red
99	104	08/08/17	Marine	red

a group

a group

a group

a group

bName	reservationCount
Interlake	2
Interlake	3
Clipper	3
Marine	3

EXERCISE 2

Find the sailor id and number of reservations made for each sailor.

Sailor

sailorId	sName	rating	age
22	Dustin	7	45
29	Brutus	1	33
31	Lubber	8	55
32	Andy	8	25
58	Rusty	10	35
64	Horatio	7	35
71	Zorba	10	16
74	Horatio	9	35
85	Art	3	25
95	Bob	3	63
99	Chris	10	30

11 tuples

Reserves

sailorId	boatId	rDate
22	101	10/10/17
22	102	10/10/17
22	103	08/10/17
22	104	07/10/17
31	102	10/11/17
31	103	06/11/17
31	104	12/11/17
64	101	05/09/17
64	102	08/09/17
74	103	08/09/17
99	104	08/08/17

11 tuples

Boat

boatId	bName	color
101	Interlake	blue
102	Interlake	red
103	Clipper	green
104	Marine	red
105	Serenity	Cyan

5 tuples

EXERCISE 2

Find the sailor id and number of reservations made for each sailor.

☞ **(22, 4), (29, 0), (31, 3), (32, 0), (58, 0), (64, 2), (71, 0), (74, 1), (85, 0), (95, 0), (99, 1)**

```
select sailorId, count(sailorId) as reservationCount
from Reserves
group by sailorId;
```

sailorId	reservationCount
22	4
31	3
64	2
74	1
99	1

How to include all sailors?

How about joining Sailor and Reserves?

```
select sailorId, count(sailorId) as reservationCount
from Sailor natural join Reserves
group by sailorId;
```

sailorId	reservationCount
22	4
31	3
64	2
74	1
99	1

What's the problem?

EXERCISE 2 (cont'd)

Find the sailor id and number of reservations made for each sailor.

☞ **(22, 4), (29, 0), (31, 3), (32, 0), (58, 0), (64, 2), (71, 0), (74, 1), (85, 0), (95, 0), (99, 1)**

sailorId	sName	rating	age	boatId	rDate
22	Dustin	7	45	101	10/10/17
22	Dustin	7	45	102	10/10/17
22	Dustin	7	45	103	08/10/17
22	Dustin	7	45	104	07/10/17
31	Lubber	8	55	102	10/11/17
31	Lubber	8	55	103	06/11/17
31	Lubber	8	55	104	12/11/17
64	Horatio	7	35	101	05/09/17
64	Horatio	7	35	102	08/09/17
74	Horatio	9	35	103	08/09/17
99	Chris	10	30	104	08/08/17
29	Brutus	1	33	-	-
32	Andy	8	25	-	-
58	Rusty	10	35	-	-
71	Zorba	10	16	-	-
85	Art	3	25	-	-
95	Bob	3	63	-	-

```
select sailorId, count(sailorId) as reservationCount
from Sailor natural join Reserves
group by sailorId;
```

$\Downarrow$

sailorId	reservationCount
22	4
31	3
64	2
74	1
99	1

☞ Some Sailor tuples have no match in the Reserves relation.

How to deal with this problem?

EXERCISE 2 (cont'd)

Find the sailor id and number of reservations made for each sailor.

☞ **(22, 4), (29, 0), (31, 3), (32, 0), (58, 0), (64, 2), (71, 0), (74, 1), (85, 0), (95, 0), (99, 1)**

```
select sailorId, count(boatId) as reservationCount
from Sailor natural left outer join Reserves
group by sailorId;
```

Recall: **left outer join** keeps all copies of the common attributes;
natural left outer join keeps only one copy of the common attributes.

Is this a correct solution?

No! Why?

```
select sailorId, count(sailorId) as reservationCount
from Sailor natural left outer join Reserves
group by sailorId;
```

Counting is done on the sailor ids and all of them appear at least once in the result.

EXERCISE 3

Find the records (tuples) of the sailors with the highest rating.

Sailor

sailorId	sName	rating	age
22	Dustin	7	45
29	Brutus	1	33
31	Lubber	8	55
32	Andy	8	25
58	Rusty	10	35
64	Horatio	7	35
71	Zorba	10	16
74	Horatio	9	35
85	Art	3	25
95	Bob	3	63
99	Chris	10	30

11 tuples

Reserves

sailorId	boatId	rDate
22	101	10/10/17
22	102	10/10/17
22	103	08/10/17
22	104	07/10/17
31	102	10/11/17
31	103	06/11/17
31	104	12/11/17
64	101	05/09/17
64	102	08/09/17
74	103	08/09/17
99	104	08/08/17

11 tuples

Boat

boatId	bName	color
101	Interlake	blue
102	Interlake	red
103	Clipper	green
104	Marine	red
105	Serenity	Cyan

5 tuples

EXERCISE 3

Find the records (tuples) of the sailors with the highest rating.

☞ **(58, Rusty, 10, 35), (71, Zorba, 10, 16), (99, Chris, 10, 30)**

Is this a correct solution?

No! Why?

```
select *
from Sailor
where rating=max(rating);
```

There is no max(rating) value to compare in the where clause.

☞ The max rating value must be obtained by a select statement!

Is this a correct solution?

No! Why?

```
select *, max(rating)
from Sailor;
```

A query that returns multiple tuples cannot contain an aggregate function.

☞ There are multiple tuples in the result, but only one max value!

EXERCISE 3 (cont'd)

Find the records (tuples) of the sailors with the highest rating.

☞ **(58, Rusty, 10, 35), (71, Zorba, 10, 16), (99, Chris, 10, 30)**

```
select *
from Sailor
where rating=(select max(rating)
              from Sailor);
```

sailorId	sName	rating	age
22	Dustin	7	45
29	Brutus	1	33
31	Lubber	8	55
32	Andy	8	25
58	Rusty	10	35
64	Horatio	7	35
71	Zorba	10	16
74	Horatio	9	35
85	Art	3	25
95	Bob	3	63
99	Chris	10	30

All sailors.

sailorId	sName	rating	age
58	Rusty	10	35
71	Zorba	10	16
99	Chris	10	30

Sailors with the maximum rating.

max(rating)
10

the maximum rating.

Sailor(sailorId, sName, rating, age)

EXERCISE 3 (cont'd)

Use set membership

Find the records (tuples) of the sailors with the highest rating.

☞ **(58, Rusty, 10, 35), (71, Zorba, 10, 16), (99, Chris, 10, 30)**

```
select *
from Sailor
where rating >=all (select rating
                    from Sailor);
```

sailorId	sName	rating	age
22	Dustin	7	45
29	Brutus	1	33
31	Lubber	8	55
32	Andy	8	25
58	Rusty	10	35
64	Horatio	7	35
71	Zorba	10	16
74	Horatio	9	35
85	Art	3	25
95	Bob	3	63
99	Chris	10	30

All sailors.

sailorId	sName	rating	age
58	Rusty	10	35
71	Zorba	10	16
99	Chris	10	30

Sailors with the highest rating.

rating
7
1
8
8
10
7
10
9
3
3
10

All ratings.

Sailor(<u>sailorId</u>, sName, rating, age)

EXERCISE 3 (cont'd)

What is the result if we replace “>=all” with “>all”?

☞ Recall “>all” is equivalent to greater than the maximum.

```
select *
from Sailor
where rating>all (select rating
                  from Sailor);
```

sailorId	sName	rating	age
22	Dustin	7	45
29	Brutus	1	33
31	Lubber	8	55
32	Andy	8	25
58	Rusty	10	35
64	Horatio	7	35
71	Zorba	10	16
74	Horatio	9	35
85	Art	3	25
95	Bob	3	63
99	Chris	10	30

All sailors.

No sailors are selected!
Why?

No ratings are > the maximum rating!

rating
7
1
8
8
10
7
10
9
3
3
10

All ratings.

Sailor(sailorId, sName, rating, age)

EXERCISE 3 (cont'd)

What is the result if we replace ">=all" with ">=some"?

☞ Recall ">some" is equivalent to greater than the minimum.

```
select *
from Sailor
where rating>=some (select rating
                     from Sailor);
```

sailorId	sName	rating	age
22	Dustin	7	45
29	Brutus	1	33
31	Lubber	8	55
32	Andy	8	25
58	Rusty	10	35
64	Horatio	7	35
71	Zorba	10	16
74	Horatio	9	35
85	Art	3	25
95	Bob	3	63
99	Chris	10	30

All sailors.

All sailors are selected!
Why?

All ratings are >= the minimum rating!

rating
7
1
8
8
10
7
10
9
3
3
10

All ratings.

Sailor(sailorId, sName, rating, age)

EXERCISE 4

DO NOT use JOIN

Find the names of sailors who have reserved a red boat.

Use *only* set membership

Sailor

sailorId	sName	rating	age
22	Dustin	7	45
29	Brutus	1	33
31	Lubber	8	55
32	Andy	8	25
58	Rusty	10	35
64	Horatio	7	35
71	Zorba	10	16
74	Horatio	9	35
85	Art	3	25
95	Bob	3	63
99	Chris	10	30

11 tuples

Reserves

sailorId	boatId	rDate
22	101	10/10/17
22	102	10/10/17
22	103	08/10/17
22	104	07/10/17
31	102	10/11/17
31	103	06/11/17
31	104	12/11/17
64	101	05/09/17
64	102	08/09/17
74	103	08/09/17
99	104	08/08/17

11 tuples

Boat

boatId	bName	color
101	Interlake	blue
102	Interlake	red
103	Clipper	green
104	Marine	red
105	Serenity	Cyan

5 tuples

EXERCISE 4

DO NOT use JOIN

Find the names of sailors who have reserved a red boat.

Use *only* set membership

☞ Dustin, Lubber, Horatio, Chris

```
select sName
from Sailor
where sailorId in (select distinct sailorId
                   from Reserves
                   where boatId in (select boatId
                                    from Boat
                                    where color='red'));
```

sName
Dustin
Lubber
Horatio
Chris

Names of sailors who have reserved a red boat.

sailorId
22
31
64
99

Unique ids of sailors who have reserved a red boat.

boatId
102
104

Ids of red boats.

Sailor(sailorId, sName, rating, age)

Reserves(*sailorId, boatId, rDate*)

Boat(boatId, bName, color)

EXERCISE 4 (cont'd)

What if we replace the first **in** with **not in**?

Stated in words, what does this result represent?

```
select sName
from Sailor
where sailorId not in (select distinct sailorId
                       from Reserves
                       where boatId in (select boatId
                                        from Boat
                                        where color='red'));
```

sName
Brutus
Andy
Rusty
Zorba
Horatio
Art
Bob

Names of sailors who have not reserved a red boat (including reserved no boat).

sailorId
22
31
64
99

Unique ids of sailors who have reserved a red boat.

boatId
102
104

Ids of red boats.

Sailor(sailorId, sName, rating, age)

Reserves(*sailorId, boatId, rDate*)

Boat(boatId, bName, color)

EXERCISE 4 (cont'd)

What if we replace the second **in** with **not in**?

```
select sName
from Sailor
where sailorId in (select distinct sailorId
                   from Reserves
                   where boatId not in (select boatId
                                        from Boat
                                        where color='red'));
```

sName
Dustin
Lubber
Horatio
Horatio

Names of sailors who have reserved a boat other than a red boat (excludes sailors who have not reserved any boat).

sailorId
22
31
64
74

Ids of sailors who have reserved a boat other than a red boat.

Stated in words, what does this result represent?

boatId
102
104

Ids of red boats.

Sailor(sailorId, sName, rating, age)

Reserves(*sailorId, boatId, rDate*)

Boat(boatId, bName, color)

EXERCISE 4 (cont'd)

What if we replace both **in**'s with **not in**?

Stated in words, what does this result represent?

```
select sName
from Sailor
where sailorId not in (select distinct sailorId
                       from Reserves
                       where boatId not in (select boatId
                                            from Boat
                                            where color='red'));
```

sName
Brutus
Andy
Rusty
Zorba
Art
Bob
Chris

Names of sailors who have reserved only a red boat (i.e., Chris) or have reserved no boat.

sailorId
22
31
64
74

Ids of sailors who have reserved a boat other than a red boat.

boatId
102
104

Ids of red boats.

Sailor(sailorId, sName, rating, age)

Reserves(*sailorId, boatId, rDate*)

Boat(boatId, bName, color)

STRUCTURED QUERY LANGUAGE (SQL)

EXERCISE 4

to be continued …

EXERCISE 4

Find the names of sailors who have reserved a red boat. **Use exists**

☞ **Dustin, Lubber, Horatio, Chris**

```
select sName
from Sailor S
where exists (select *
              from Reserves natural join Boat
              where Reserves.sailorId=S.sailorId
                    and color='red');
```

sailorId	sName
22	Dustin
29	Brutus
31	Lubber
32	Andy
58	Rusty
64	Horatio
71	Zorba
74	Horatio
85	Art
95	Bob
99	Chris

Reserves natural join Boat where color='red'

boatId	sailorId	rDate	bName	color
102	22	10/10/17	Interlake	red
102	64	08/09/17	Interlake	red
102	31	10/11/17	Interlake	red
104	22	07/10/17	Marine	red
104	99	08/08/17	Marine	red
104	31	12/11/17	Marine	red

Sailor(sailorId, sName, rating, age)

Reserves(*sailorId, boatId, rDate*)

Boat(boatId, bName, color)

EXERCISE 4

Find the names of sailors who have reserved a red boat.

Use with clause

☞ **Dustin, Lubber, Horatio, Chris**

```
with RedBoatReservations (sailorId) as
(select sailorId
 from Reserves natural join Boat
 where color='red')
select distinct sName
from Sailor natural join RedBoatReservations;
```

sailorId	sName
22	Dustin
29	Brutus
31	Lubber
32	Andy
58	Rusty
64	Horatio
71	Zorba
74	Horatio
85	Art
95	Bob
99	Chris

RedBoatReservations
sailorId
22
64
31
22
99
31

Sailor(sailorId, sName, rating, age)

Reserves(*sailorId, boatId, rDate*)

Boat(boatId, bName, color)

EXERCISES 5, 6, 7

Sailor(sailorId, sName, rating, age)

Boat(boatId, bName, color)

Reserves(*sailorId, boatId,* rDate)

Exercise 5: Find the ratings and the average age of the ratings where a rating's average age is equal to the minimum average age of all ratings.

Exercise 6: Find the boat name and number of reservations made for each boat. Do not use any subqueries. Do not create any derived tables.

Exercise 7: Find the age of the youngest adult sailor (i.e., age≥18) for each rating for which there are at least 2 adult sailors (i.e., 2 sailors whose age is ≥ 18) with the same rating. Do not create any derived tables.

EXERCISE 5

Find the ratings and the average age of the ratings where a rating's average age is equal to the minimum average age of all ratings.

☞ **(10, 27)**

Is this a correct solution?

No! Why?

```
select rating
from Sailor
where avg(age)=min(select avg(age)
                   from Sailor
                   group by rating);
```

Cannot use "**where avg**(age)=" since **avg**(age) is not an attribute of Sailor!

Cannot use "**min**(…". Illegal SQL!

Is this a correct solution?

No! Why?

```
select rating
from Sailor
group by rating
having age=(select avg(age)
            from Sailor
            group by rating);
```

avgAge
33
44
40
40
35
27

Cannot use "**having** age=" since age is not in the **select** or **group by** clauses. Illegal SQL!

Subquery returns multiple values.

EXERCISE 5 (cont'd)

Find the ratings and the average age of the ratings where a rating's average age is equal to the minimum average age of all ratings.

☞ **(10, 27)**

```
select rating, avg(age)
from Sailor
group by rating
having avg(age) = (select min(avg(age))
                   from Sailor
                   group by rating);
```

rating	avgAge
1	33
3	44
7	40
8	40
9	35
10	27

The average age for each rating.

rating	avgAge
10	27

The ratings where the average age is equal to the minimum average age of all ratings.

avgAge
27

The minimum average age of all ratings.

EXERCISE 5 (cont'd)

Find the ratings and the average age of the ratings where a rating's average age is equal to the minimum average age of all ratings.

☞ **(10, 27)**

```
select rating, avg(age)
from Sailor
group by rating
having avg(age) <=all (select avg(age)
                       from Sailor
                       group by rating);
```

Can we replace <=all with >=some?

No! Why?

Will include all ratings.

rating	avgAge
1	33
3	44
7	40
8	40
9	35
10	27

The average age for each rating.

rating	avgAge
10	27

The ratings where the average age is equal to the minimum average age of all ratings.

avgAge
27

The minimum average age of all ratings.

EXERCISE 5 (cont'd)

Find the ratings and the average age of the ratings where a rating's average age is equal to the minimum average age of all ratings.

☞ **(10, 27)**

```
select rating, avgAge
from (select rating, avg(age) as avgAge
         from Sailor
         group by rating) temp
where avgAge=(select min(avgAge)
                        from temp);
```

- This query is correct SQL but will not execute in Oracle.
  - Returns the error "`table or view does not exist`".

☞ **Oracle restricts the scope of the alias temp to the outer select.**

EXERCISE 5 (cont'd)

Find the ratings and the average age of the ratings where a rating's average age is equal to the minimum average age of all ratings.

☞ **(10, 27)**

```
with temp as
     (select rating, avg(age) as avgAge
      from Sailor
      group by rating)
select rating, avgAge
from temp
where avgAge=(select min(avgAge)
              from temp);
```

temp

rating	avgAge
1	33
3	44
7	40
8	40
9	35
10	27

The average age for each rating.

rating	avgAge
10	27

The ratings where the average age is equal to the minimum average age over all ratings.

avgAge
27

The minimum average age in the **temp** relation.

Sailor(sailorId, sName, rating, age)

Do not use any subqueries.

EXERCISE 6

Do not create any derived tables.

Find the boat name and number of reservations for each boat.

☞ **(Clipper, 3), (Interlake, 2), (Interlake, 3), (Marine, 3), (Serenity, 0)**

```
select bName, count(bName) as reservationCount
from Boat natural left outer join Reserves
group by bName;
```

bName	reservation Count
Clipper	3
Interlake	5
Marine	3
Serenity	1

The count for Serenity is incorrect; should be 0.
Interlake should have two separate counts.

What's the problem?

How about group on boatId, bName; count boatId?

```
select bName, count(boatId) as reservationCount
from Boat natural left outer join Reserves
group by boatId, bName;
```

bName	reservation Count
Clipper	3
Interlake	2
Interlake	3
Marine	3
Serenity	1

The count for Serenity is still incorrect!

What's the problem?

Do not use any subqueries.

EXERCISE 6 (cont'd)

Do not create any derived tables.

Find the boat name and number of reservations for each boat.

☞ (Clipper, 3), (Interlake, 2), (Interlake, 3), (Marine, 3), (Serenity, 0)

```
select bName, count(boatId) as reservationCount
from Boat natural left outer join Reserves
group by boatId, bName;
```

Boat natural left outer join Reserves

boatId	bName	color	sailorId	rDate
101	Interlake	blue	64	05/09/17
101	Interlake	blue	22	10/10/17
102	Interlake	red	22	10/10/17
102	Interlake	red	64	08/09/17
102	Interlake	red	31	10/11/17
103	Clipper	green	22	08/10/17
103	Clipper	green	31	06/11/17
103	Clipper	green	74	08/09/17
104	Marine	red	22	07/10/17
104	Marine	red	99	08/08/17
104	Marine	red	31	12/11/17
105	Serenity	cyan	(null)	(null)

☞ **We need to count sailorId or rDate!**

Do not use any subqueries.

EXERCISE 6 (cont'd)

Do not create any derived tables.

Find the boat name and number of reservations for each boat.

☞ **(Clipper, 3), (Interlake, 2), (Interlake, 3), (Marine, 3), (Serenity, 0)**

```
select bName, count(sailorId) as reservationCount
from Boat natural left outer join Reserves
group by boatId, bName;
```

bName	reservation Count
Clipper	3
Interlake	2
Interlake	3
Marine	3
Serenity	0

EXERCISE 7

Do not create any derived tables.

Find the age of the youngest adult sailor (i.e., age≥18) for each rating for which there are at least 2 adult sailors with the same rating.

☞ **(3, 25), (7, 35), (8, 25), (10, 30)**

Is this a correct solution?

No! Why?

```
select rating, min(age)
from Sailor S
group by rating
having 1<(select count(*)
          from Sailor
          where S.rating=rating
          and age>=18);
```

sailorId	sName	rating	age	
29	Brutus	1	33	
85	Art	3	25	✓
95	Bob	3	63	
22	Dustin	7	45	✓
64	Horatio	7	35	
31	Lubber	8	55	✓
32	Andy	8	25	
74	Horatio	9	35	
58	Rusty	10	35	
71	Zorba	10	16	✓
99	Chris	10	30	

Group by rating.

Select those groups having both more than one sailor and a sailor whose age is greater than or equal to 18 .

rating	min(age)
3	25
7	35
8	25
10	16

X

Select the rating and minimum age for each selected group.

Sailor(sailorId, sName, rating, age)

EXERCISE 7 (cont'd)

Do not create any derived tables.

Find the age of the youngest adult sailor (i.e., age≥18) for each rating for which there are at least 2 adult sailors with the same rating.

☞ **(3, 25), (7, 35), (8, 25), (10, 30)**

```
select rating, min(age)
from Sailor
where age>=18
group by rating
having count(*)>=2;
```

Sailors whose age is greater than or equal to 18.

sailorId	sName	rating	age
22	Dustin	7	45
29	Brutus	1	33
31	Lubber	8	55
32	Andy	8	25
58	Rusty	10	35
64	Horatio	7	35
74	Horatio	9	35
85	Art	3	25
95	Bob	3	63
99	Chris	10	20

Group the result by rating.

sailorId	sName	rating	age	
29	Brutus	1	33	
85	Art	3	25	✓
95	Bob	3	63	
22	Dustin	7	45	✓
64	Horatio	7	35	
31	Lubber	8	55	✓
32	Andy	8	25	
74	Horatio	9	35	
58	Rusty	10	35	✓
99	Chris	10	30	

Select those groups having at least 2 sailors.

Select the rating and minimum age for each selected group.

rating	min(age)
3	25
7	35
8	25
10	30

Sailor(sailorId, sName, rating, age)