SCC201: Databases

Week5-Continue: SQL and Advanced SQL Scripts

By
Uraz C Turker

Aggregate Operators

Lancaster University

```
COUNT (*)
COUNT ( [DISTINCT] A)
SUM ( [DISTINCT] A)
AVG ( [DISTINCT] A)
MAX (A)
MIN (A)
```

- Significant extension of relational algebra.
- They are used to write statistical queries
- Mainly used for reporting, such as
  - the total sales in 2004,
  - average, max, min income of employees
  - Total number of employees hired/fired in 2004

Aggregate Operators

Lancaster University

COUNT (*)
COUNT ([DISTINCT] A)
SUM ([DISTINCT] A)
AVG ([DISTINCT] A)
MAX (A)
MIN (A)

The total number of sailors in the club?

```
SELECT  COUNT (*)
FROM  Sailors S;
```

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

Figure 4.15 An Instance *S*3 of Sailors

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

Figure 4.16 An Instance *R*2 of Reserves

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

Figure 4.17 An Instance *B*1 of Boats

Aggregate Operators

Lancaster University

Average age of sailors in the club
Whose rating is 10?

```
SELECT  AVG (S.age)
FROM  Sailors S
WHERE  S.rating=10;
```

COUNT (*)
COUNT ([DISTINCT] A)
SUM ([DISTINCT] A)
AVG ([DISTINCT] A)
MAX (A)
MIN (A)

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

Figure 4.15 An Instance *S*3 of Sailors

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

Figure 4.16 An Instance *R*2 of Reserves

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

Figure 4.17 An Instance *B*1 of Boats

Aggregate Operators

COUNT (*)
COUNT ([DISTINCT] A)
SUM ([DISTINCT] A)
AVG ([DISTINCT] A)
MAX (A)
MIN (A)

Average distinct ages of sailors
Whose rating is 10?

```
SELECT  AVG ( DISTINCT S.age)
FROM  Sailors S
WHERE  S.rating=10;
```

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

Figure 4.15 An Instance *S3* of Sailors

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

Figure 4.16 An Instance *R2* of Reserves

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

Figure 4.17 An Instance *B1* of Boats

Aggregate Operators

COUNT (*)
COUNT ([DISTINCT] A)
SUM ([DISTINCT] A)
AVG ([DISTINCT] A)
MAX (A)
MIN (A)

Names of sailors whose rating is equal to the maximum rating in the club.

```
SELECT  S.sname
FROM  Sailors S
WHERE  S.rating= (SELECT  MAX(S2.rating)
                  FROM  Sailors S2);
```

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

Figure 4.15 An Instance *S*3 of Sailors

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

Figure 4.16 An Instance *R*2 of Reserves

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

Figure 4.17 An Instance *B*1 of Boats

Aggregate Operators

COUNT (*)
COUNT ([DISTINCT] A)
SUM ([DISTINCT] A)
AVG ([DISTINCT] A)
MAX (A)
MIN (A)

Number of sailors whose rating is equal to the maximum rating in the club.

```
SELECT  COUNT(S.sid)
FROM  Sailors S
WHERE  S.rating= (SELECT  MAX(S2.rating)
                  FROM  Sailors S2)
```

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

Figure 4.15 An Instance $S3$ of Sailors

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

Figure 4.16 An Instance $R2$ of Reserves

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

Figure 4.17 An Instance $B1$ of Boats

Aggregate Operators

COUNT (*)
COUNT ([DISTINCT] A)
SUM ([DISTINCT] A)
AVG ([DISTINCT] A)
MAX (A)
MIN (A)

Lancaster University

How many different ratings are there in the club?

SELECT COUNT (S.rating)
AS Res FROM Sailors S;

Above query is not correct. Think why!

SELECT COUNT (DISTINCT S.rating)
AS Res FROM Sailors S;

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

Figure 4.15 An Instance *S*3 of Sailors

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

Figure 4.16 An Instance *R*2 of Reserves

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

Figure 4.17 An Instance *B*1 of Boats

Find name and age of the oldest sailor(s)

- This query is correct and it is allowed in the SQL/92 standard, but is not supported in some systems.

```
SELECT  S.sname, S.age
FROM  Sailors S
WHERE  S.age =
        (SELECT  MAX (S2.age)
         FROM  Sailors S2);
```

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

Figure 4.15 An Instance $S3$ of Sailors

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

Figure 4.16 An Instance $R2$ of Reserves

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

Figure 4.17 An Instance $B1$ of Boats

Find name and age of the oldest sailor(s)

- This query is valid for all systems .

```
SELECT  S.sname, S.age
FROM  Sailors S
WHERE  (SELECT  MAX (S2.age)
        FROM  Sailors S2)
        = S.age;
```

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

Figure 4.15 An Instance *S*3 of Sailors

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

Figure 4.16 An Instance *R*2 of Reserves

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

Figure 4.17 An Instance *B*1 of Boats

GROUP BY and HAVING

- So far, we've applied aggregate operators to all (qualifying) tuples. Sometimes, we want to apply them to each of several *groups* of tuples.

Consider: *Find the age of the youngest sailor for each rating level.*

In general, we don't know how many rating levels exist and what the rating values for these levels are!

- Suppose we know that rating values go from 1 to 10; we can write 10 queries that look like this (!):

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

Figure 4.15 An Instance *S*3 of Sailors

For i = 1, 2, … , 10:

```
SELECT  MIN (S.age)
FROM  Sailors S
WHERE  S.rating = i
```

Queries With GROUP BY and HAVING

```
SELECT      [DISTINCT] target-list
FROM        relation-list
WHERE       qualification
GROUP BY    grouping-list
HAVING      group-qualification
```

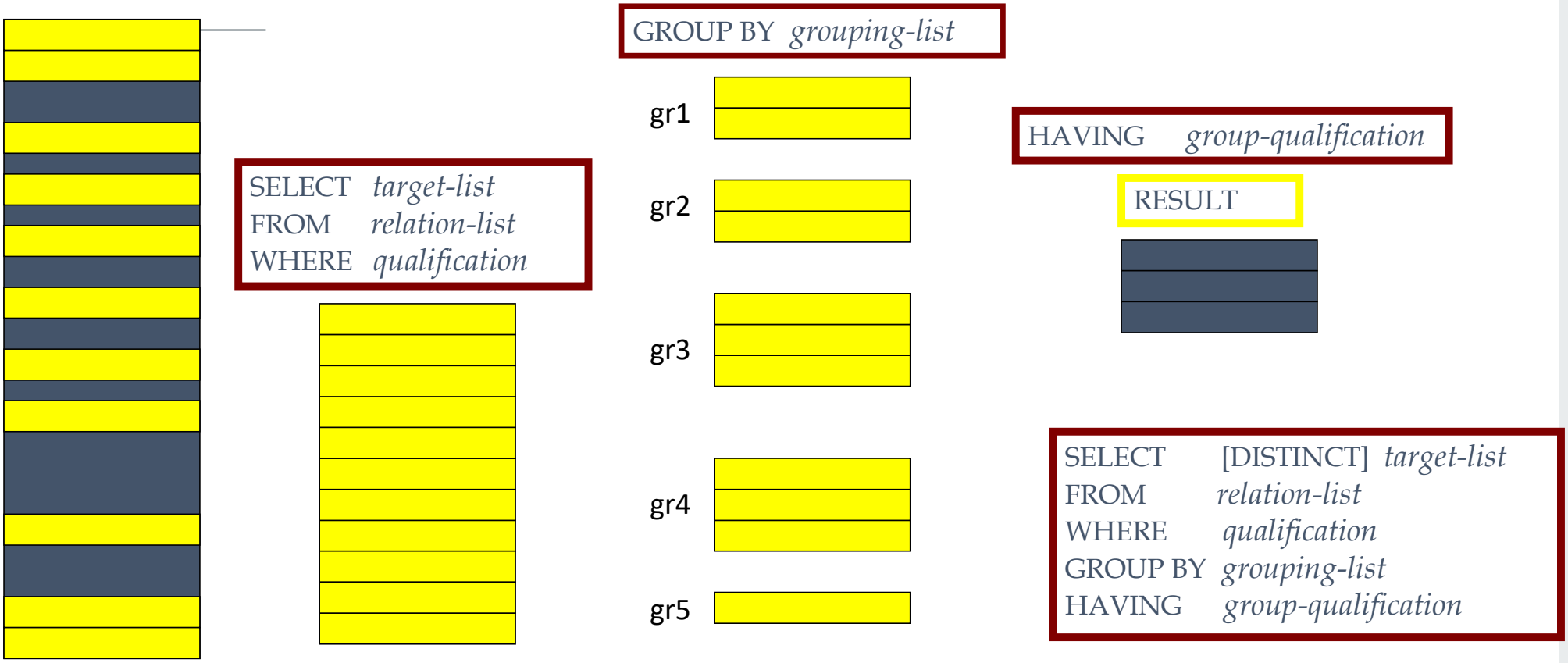
- The *target-list* contains (i) attribute list(**ii) terms with aggregate operations (e.g., MIN (*S.age*)).**

Conceptual Evaluation

- The cross-product of *relation-list* is computed, tuples that fail *qualification* are discarded, `*unnecessary'* fields are deleted, and the remaining tuples are partitioned into groups by the value of attributes in *grouping-list*.
- The *group-qualification* is then applied to eliminate some groups. Expressions in *group-qualification* must have a *single value per group*!
- One answer tuple is generated per qualifying group.

```
SELECT      [DISTINCT]  target-list
FROM        relation-list
WHERE       qualification
GROUP BY    grouping-list
HAVING      group-qualification
```

Conceptual Evaluation



Conceptual Evaluation

Find the sailors with age $\geq$18, for each rating with at least 2 such sailors

Lancaster University

Age = 20
Age = 25
Age = 19
Age=70
Age = 60
Age = 40
Age = 33
Age = 32
Age = 18
Age = 22
Age = 39

```
SELECT  S.rating
FROM  Sailors S
WHERE  S.age >= 18
```

Rating = 4
Rating =2
Rating =3
Rating=2
Rating=3
Rating=5
Rating=1
Rating=4
Rating=3
Rating=4
Rating=1

GROUP BY *S.rating*

gr1	Rating=1 Rating=1
gr2	Rating=2 Rating=2
gr3	Rating=3 Rating=3 Rating=3
gr4	Rating=4 Rating=4 Rating=4
gr5	Rating=5

```
HAVING  COUNT (*) > 1
```

RESULT

Rating = 1
Rating = 2
Rating = 3
Rating = 4

```
SELECT  S.rating
FROM  Sailors S
WHERE  S.age >= 18
GROUP BY  S.rating
HAVING  COUNT (*) > 1
```

Find the age of the youngest sailor with age ≥ 18, for each rating with at least 2 such sailors

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
71	zorba	10	16.0
64	horatio	7	35.0
29	brutus	1	33.0
58	rusty	10	35.0

Find the age of the youngest sailor with age $\geq$ 18, for each rating with at least 2 such sailors

```
SELECT  S.rating
FROM  Sailors S
WHERE  S.age >= 18
```

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
71	zorba	10	16.0
64	horatio	7	35.0
29	brutus	1	33.0
58	rusty	10	35.0

Find the age of the youngest sailor with age $\geq$ 18, for each rating with at least 2 such sailors

```
SELECT  S.rating
FROM  Sailors S
WHERE  S.age >= 18
GROUP BY S.rating
```

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
71	zorba	10	16.0
64	horatio	7	35.0
29	brutus	1	33.0
58	rusty	10	35.0

Find the age of the youngest sailor with age $\geq$ 18, for each rating with at least 2 such sailors

```
SELECT  S.rating
FROM  Sailors S
WHERE  S.age >= 18
GROUP BY S.rating
HAVING COUNT(*)>1
```

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
71	zorba	10	16.0
64	horatio	7	35.0
29	brutus	1	33.0
58	rusty	10	35.0

Find the age of the youngest sailor with age $\geq$ 18, for each rating with at least 2 such sailors

```
SELECT  S.rating
FROM  Sailors S
WHERE  S.age >= 18
GROUP BY S.rating
HAVING COUNT(*)>1
```

Minimum AGE ?

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
71	zorba	10	16.0
64	horatio	7	35.0
29	brutus	1	33.0
58	rusty	10	35.0

Find the age of the youngest sailor with age $\geq$ 18, for each rating with at least 2 such sailors

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
71	zorba	10	16.0
64	horatio	7	35.0
29	brutus	1	33.0
58	rusty	10	35.0

```
SELECT  S.rating
FROM  Sailors S
WHERE  S.age >= 18
GROUP BY S.rating
HAVING COUNT(*)>1
```

Minimum AGE ?

So, introduce a new attribute in the solution that selects the youngest age..

Find the age of the youngest sailor with age $\geq$ 18, for each rating with at least 2 such sailors

```
SELECT  S.rating, Min(age)
FROM  Sailors S
WHERE  S.age >= 18
GROUP BY S.rating
HAVING COUNT(*)>1
```

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
71	zorba	10	16.0
64	horatio	7	35.0
29	brutus	1	33.0
58	rusty	10	35.0

Find the age of the youngest sailor with age $\geq$ 18, for each rating with at least 2 such sailors

```
SELECT  S.rating, Min(age)
FROM  Sailors S
WHERE  S.age >= 18
GROUP BY S.rating
HAVING COUNT(*)>1
```

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
71	zorba	10	16.0
64	horatio	7	35.0
29	brutus	1	33.0
58	rusty	10	35.0

rating	age
1	33.0
7	45.0
7	35.0
8	55.5
10	35.0

Find the age of the youngest sailor with age $\geq$ 18, for each rating with at least 2 such sailors

```
SELECT S.rating, MIN (S.age)
FROM  Sailors S
WHERE  S.age >= 18
GROUP BY  S.rating
HAVING  COUNT (*) > 1
```

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
71	zorba	10	16.0
64	horatio	7	35.0
29	brutus	1	33.0
58	rusty	10	35.0

rating	age
1	33.0
7	45.0
7	35.0
8	55.5
10	35.0

rating	
7	35.0

- Only S.rating and S.age are mentioned in the SELECT, GROUP BY or HAVING clauses;
- 2nd column of result is unnamed. (Use AS to name it.)

Find the age of the youngest sailor with age $\geq$ 18, for each rating with at least 2 such sailors

```
SELECT S.rating, ag=MIN (S.age)
FROM  Sailors S
WHERE  S.age >= 18
GROUP BY  S.rating
HAVING  COUNT (*) > 1;
```

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
71	zorba	10	16.0
64	horatio	7	35.0
29	brutus	1	33.0
58	rusty	10	35.0

rating	age
1	33.0
7	45.0
7	35.0
8	55.5
10	35.0

- Only S.rating and S.age are mentioned in the SELECT, GROUP BY or HAVING clauses;
- 2nd column of result is unnamed. (Use AS to name it.)

rating	ag
7	35.0

Answer relation

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

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

Figure 4.15 An Instance $S3$ of Sailors

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

Figure 4.16 An Instance $R2$ of Reserves

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

Figure 4.17 An Instance $B1$ of Boats

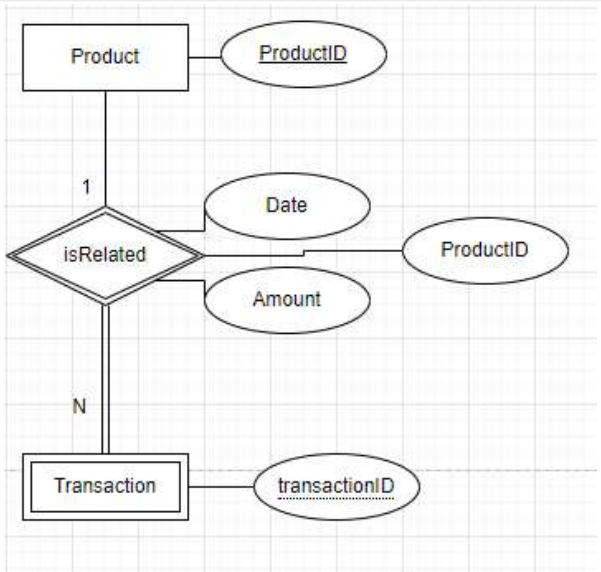
For each red boat, find the number of reservations for this boat

```
SELECT  B.bid,  COUNT (*) AS scount
FROM  Boats B, Reserves R
WHERE  R.bid=B.bid AND B.color='red'
GROUP BY  B.bid
```

Exemple

- *"We are running an instrument store in Uxbridge, London. We want to create a database that keeps transaction information with transaction date, ProductID (customer purchased), and amount_purchased, transactionID. Moreover we have products with unique ProductID, and name, price, product picture, and amount_in_Store.*
- *We can issue one transaction per product, and we will not keep records of a transaction if the purchased product is deleted (removed from the store)."*

- Draw ERD, Reveal Relational Schema and Integrity Constraints and write SQL code to create the database.



Product(ProductID INTEGER, name TEXT, price REAL, picture BLOB, amount_in_Store INTEGER);
IC: Primary Key ProductID

TransactionIsRelated(ProductID INTEGER, Amount REAL, Date TEXT, P_ProductID INTEGER, TransactionID INTEGER)

IC: Primary Key (P_PruductID,transactionID)
IC: Foreign Key (P_ProductID) References Product ON DELETE CASCADE

CREATE TABLE Product(ProductID INTEGER,… PRIMARY KEY(ProductId));

CREATE TABLE TransactionIsRelated(ProductID INTEGER, Amount REAL, Date TEXT, P_ProductID INTEGER, TransactionID INTEGER,Primary Key (ProductID,TransactionID), Foreign Key (P_ProductID) References Product(ProductID) ON DELETE CASCADE
);

Summary

- SQL is a very powerful declarative scripting language.
- Unfortunately, there are many ways of writing an SQL script for a given query.
  - Bad ones will consume CPU time and memory.
  - Good ones are hardware and time friendly.
    - Relational algebra is the key to deriving optimum SQL queries.

Summary

- SQL is a very powerful declarative scripting language.
- Unfortunately, there are many ways of writing an SQL script for a given query.
  - Bad ones will consume CPU time and memory.
  - Good ones are hardware and time friendly.
    - Relational algebra is the key to deriving optimum SQL queries.