More SQL: Nested Queries

Today's Database

Sailors

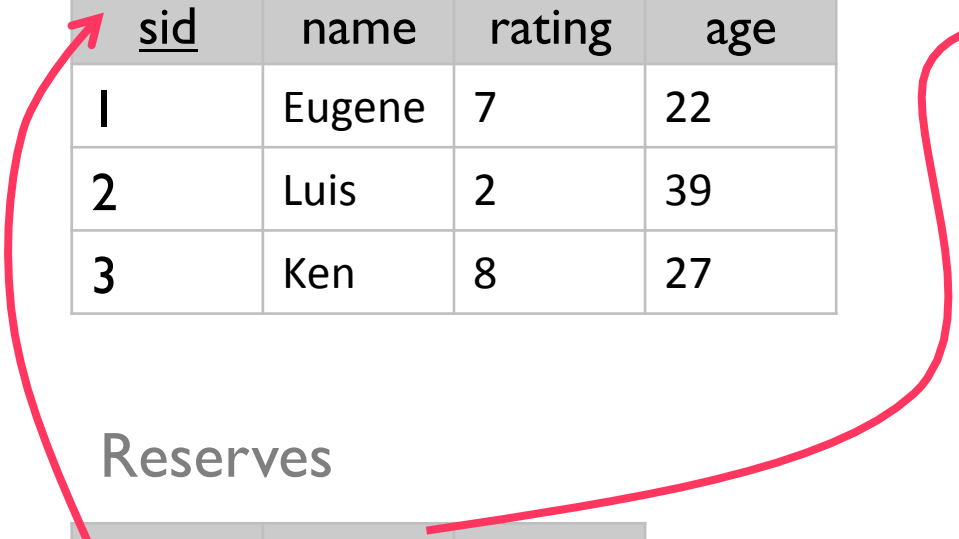
sid	name	rating	age
1	Eugene	7	22
2	Luis	2	39
3	Ken	8	27

Boats

bid	name	color
101	Legacy	red
102	Melon	blue
103	Mars	red

Reserves

sid	bid	day
1	102	9/12
2	102	9/13
2	103	9/14
2	103	9/15



Is Reserves table correct?
Day should be part of key

Today's Database

Reserves

sid	bid	day
1	102	9/12
2	102	9/13
2	103	9/14
2	103	9/15

PRIMARY KEY (sid, bid)

Sailor can only reserve a boat (e.g. 102) **once**

PRIMARY KEY (sid, bid, day)

Boat (e.g. 102) reserved by 2 sailors on same day

Today's Database

PRIMARY KEY (sid, bid, day)
  Boat (e.g. 102) reserved by 2 sailors on same day

+ UNIQUE (bid, day)? Works!

PRIMARY KEY (sid, bid, day) + UNIQUE (bid, day) =
    PRIMARY KEY(bid, day) + sid NOT NULL

HW1 bugs

Missing CHECK constraints

```
Prof(
  type text,
  check(text = ‘junior’ or text = ‘senior’),
)
```

CHECK constraints

Useful for single table constraints

Uncommon in reality, but not unheard of

Remember that they exist for tests

Bank: CHECK(balance > 0)

School: CHECK(

    isInstructor OR degree IS NOT NULL)

UNION, INTERSECT, EXCEPT

Algebra: ∪, ∩, -

Combine results from two queries:

SELECT [query1] UNION SELECT [query2]

By default: *distinct results*! (set semantics)
(operator) ALL: Keep duplicates: multi-set

sid of Sailors that reserved red or blue boat

```
SELECT   DISTINCT R.sid
FROM     Boats B, Reserves R
WHERE    B.bid = R.bid AND
         (B.color = ‘red’ OR B.color = ‘blue’)
```

OR

```
SELECT   R.sid
FROM     Boats B, Reserves R
WHERE    B.bid = R.bid AND B.color = ‘red’
UNION
SELECT   R.sid
FROM     Boats B, Reserves R
WHERE    B.bid = R.bid AND B.color = ‘blue’
```

sid of Sailors that reserved red or blue boat

```
SELECT   R.sid
FROM     Boats B, Reserves R
WHERE    B.bid = R.bid AND
         (B.color = ‘red’ OR B.color = ‘blue’)
```

OR

```
SELECT   R.sid
FROM     Boats B, Reserves R
WHERE    B.bid = R.bid AND B.color = ‘red’
UNION ALL
SELECT   R.sid
FROM     Boats B, Reserves R
WHERE    B.bid = R.bid AND B.color = ‘blue’
```

sid of Sailors that reserved red and blue boat

```
SELECT   R.sid
FROM     Boats B, Reserves R
WHERE    B.bid = R.bid AND
         (B.color = ‘red’ AND B.color = ‘blue’)
```

```
SELECT   R.sid
FROM     Boats B, Reserves R
WHERE    B.bid = R.bid AND B.color = ‘red’
INTERSECT
SELECT   R.sid
FROM     Boats B, Reserves R
WHERE    B.bid = R.bid AND B.color = ‘blue’
```

sid of Sailors that reserved red and blue boat

Can use self-join instead

```
SELECT   DISTINCT R1.sid
FROM     Boats B1, Reserves R1
WHERE
         B1.bid = R1.bid AND

         B1.color = ‘red’
```

sid of Sailors that reserved red and blue boat

Can use self-join instead

```
SELECT   DISTINCT R1.sid
FROM     Boats B1, Reserves R1, Boats B2,Reserves R2
WHERE
         B1.bid = R1.bid AND

         B1.color = ‘red’
```

sid of Sailors that reserved red and blue boat

Can use self-join instead

```
SELECT   DISTINCT R1.sid
FROM     Boats B1, Reserves R1, Boats B2,Reserves R2
WHERE
         B1.bid = R1.bid AND
         B2.bid = R2.bid AND
         B1.color = ‘red’ AND B2.color = ‘blue’
```

sid of Sailors that reserved red and blue boat

Can use self-join instead

```
SELECT   DISTINCT R1.sid
FROM     Boats B1, Reserves R1, Boats B2,Reserves R2
WHERE    R1.sid = R2.sid AND
         B1.bid = R1.bid AND
         B2.bid = R2.bid AND
         B1.color = ‘red’ AND B2.color = ‘blue’
```

sids of sailors that haven't reserved a boat

```
SELECT   S.sid
FROM     Sailors S

EXCEPT

SELECT   S.sid
FROM     Sailors S, Reserves R
WHERE    S.sid = R.sid
```

Nested Queries

```
SELECT  S.sid
FROM    Sailors S
WHERE   S.sid IN (SELECT   R.sid
                  FROM     Reserves R
                  WHERE    R.bid = 101)
```

Many clauses can contain SQL queries

WHERE, FROM, HAVING, SELECT

Conceptual model:

```
for each Sailors tuple
   run the subquery and evaluate qualification
```

Nested Query vs Join

```
SELECT  S.sid
FROM    Sailors S
WHERE   S.sid IN (SELECT   R.sid
                  FROM     Reserves R
                  WHERE    R.bid = 101)
```

```
SELECT  S.sid
FROM    Sailors S, Reserves R
WHERE   S.sid = R.sid AND R.bid = 101
```

What if a student reserved a boat more than once?

Nested: No duplicates **Join: Duplicates**

SET Comparison Operators

x IN *r*: True if value *x* appears in *r*

EXISTS *r*: True if relation *r* is not empty (NOT EXISTS)

x (operator) ANY r: True if x (operator) is true for any row in r

E.g. x IN r is equivalent to x = ANY r

x (operator) ALL r: True if x (operator) is true for all rows in r

E.g. x NOT IN r is equivalent to x <> ALL r

Reference outer table in nested query

```
SELECT   S.sid
FROM     Sailors S
WHERE    EXISTS (SELECT   *
                 FROM     Reserves R
                 WHERE    R.bid = 101 AND
                          S.sid = R.sid)
```

Outer table referenced in nested query

Conceptual model:

```
for each Sailors tuple
   run the subquery and evaluate qualification
```

Sailors whose rating is greater than any sailor named “Bobby”

```
SELECT S1.name
FROM   Sailors S1
WHERE  S1.rating > ANY (SELECT   S2.rating
                        FROM     Sailors S2
                        WHERE    S2.name = ‘Bobby’)
```

How are these different?

```
SELECT S1.name
FROM   Sailors S1
WHERE  S1.rating > ANY (SELECT   S2.rating
                        FROM     Sailors S2
                        WHERE    S2.name = ‘Bobby’)
```

```
SELECT S1.name
FROM   Sailors S1
WHERE  S1.rating > ALL (SELECT   S2.rating
                        FROM     Sailors S2
                        WHERE    S2.name = ‘Bobby’)
```

Rewrite INTERSECT using IN

```
SELECT S.sid
FROM   Sailors S
WHERE  S.rating > 2

INTERSECT

SELECT R.sid
FROM   Reserves R
```

```
SELECT S.sid
FROM   Sailors S
WHERE  S.rating > 2 AND
       S.sid IN (
    SELECT R.sid
    FROM   Reserves R
)
```

Similar trick for EXCEPT → NOT IN

What if want *names* instead of sids?

Names are not unique!

Name of sailors that reserved all boats

Hint: All is hard: have "EXISTS" not "FOR ALL"
What about double negation?

reserved all boats == no boat w/out reservation

Can we find boats not reserved by sailor x?

Use that to find sailors who do not have any unreserved boats!

Q1: boats not reserved by Sailor 1

Sailors

sid	name	rating	age
1	Eugene	7	22
2	Luis	2	39
3	Ken	8	27

Boats

bid	name	color
101	Legacy	red
102	Melon	blue
103	Mars	red

Reserves

sid	bid	day
1	102	9/12
2	102	9/13
2	103	9/14
2	101	9/15

Hint: boats reserved by Sailor 1?

Hint: Use a nested query

Want: sailors who reserved all boats

Boats reserved by Sailor 1

```
SELECT    DISTINCT r.bid
FROM      Reserves r
WHERE     r.sid = 1;
```

Boats not reserved by Sailor 1

```
SELECT    b.bid
FROM      Boats b
WHERE     b.bid NOT IN (

    SELECT    r.bid
    FROM      Reserves r
    WHERE     r.sid = 1
);
```

Q2:
All sailors with unreserved boats

Want: sailors who reserved all boats

All sailors with unreserved boats

```
SELECT    s.sid, s.name
FROM      Sailors s
WHERE     EXISTS (

    SELECT    b.bid
    FROM      Boats b
    WHERE     b.bid NOT IN (

        SELECT    r.bid
        FROM      Reserves r
        WHERE     r.sid = s.sid
    )
);
```

Q:
sailors who have reserved all boats

All sailors who reserved all boats

```
SELECT    s.sid, s.name
FROM      Sailors s
WHERE     NOT EXISTS (

    SELECT    b.bid
    FROM      Boats b
    WHERE     b.bid NOT IN (

        SELECT    r.bid
        FROM      Reserves r
        WHERE     r.sid = s.sid
    )
);
```

Sailors that reserved all boats (Division)

Hint: double negation

reserved all boats == ∄ boat ∄ reservation

```
SELECT S.name
FROM   Sailors S
WHERE  NOT EXISTS (
```

Sailors S where there is not

Sailors that reserved all boats (Division)

Hint: double negation

reserved all boats == ∄ boat ∄ reservation

```
SELECT S.name
FROM   Sailors S
WHERE  NOT EXISTS (SELECTB.bid
                   FROM  Boats B
                   WHERE NOT EXISTS (
```

Sailors S where there is not

Any boat where there is not

Sailors that reserved all boats (Division)

Hint: double negation

reserved all boats == ∄ boat ∄ reservation

```
SELECT S.name
FROM   Sailors S
WHERE  NOT EXISTS (SELECT B.bid
                   FROM   Boats B
                   WHERE NOT EXISTS (SELECT R.bid
                                     FROM Reserves R
                                     WHERE  R.sid = S.sid
                                        AND R.bid = B.bid ))
```

Sailors S where there is not

Any boat where there is not

A reservation by S