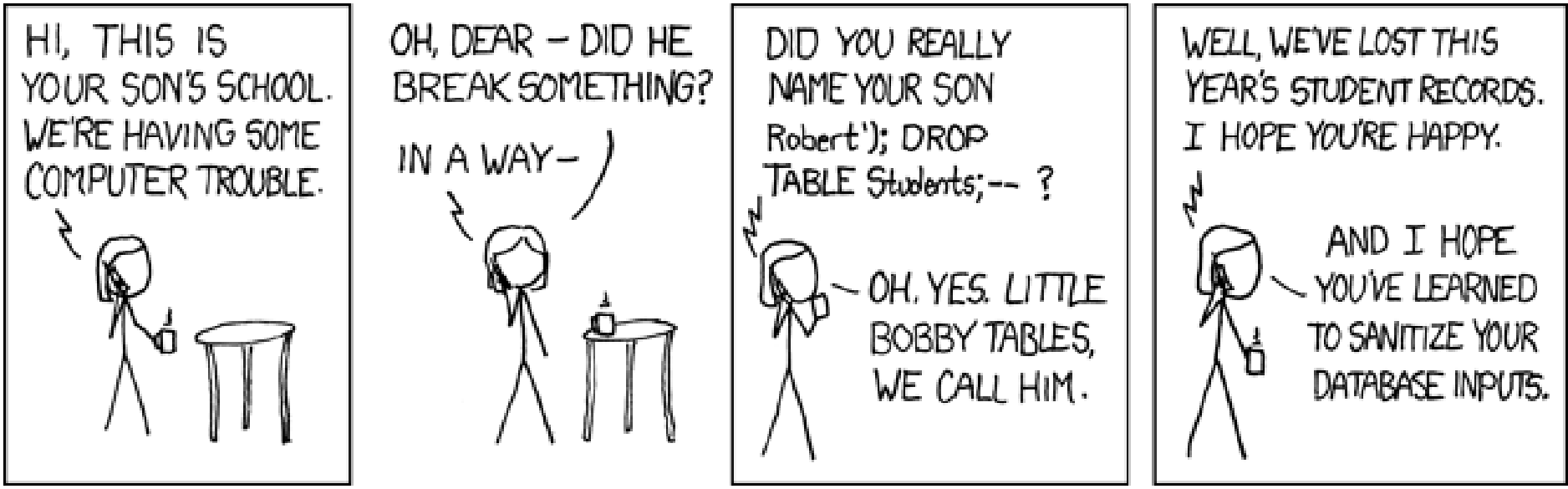
SQL: The Query Language
Part II

R&G - Chapter 5



Announcement

- **Midterm exam (Oct 26 & 28 tentative)**
- **Assignment 2 is available in Moodle**
  - Relational Algebra and SQL
  - Due time: Monday Oct 31 (11:59pm this time!)

SQL

- **The form:**

  SELECT A_1, A_2, ..., A_n
  FROM r_1, r_2, ..., r_m
  WHERE P

  - A_i represents an attribute
  - r_i represents a relation
  - P is a predicate

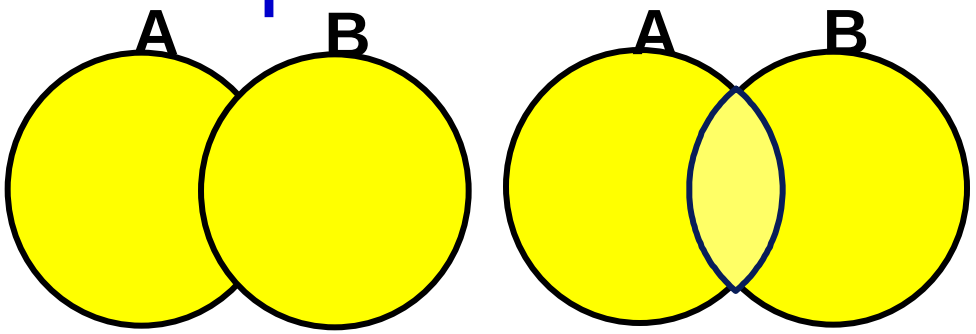
- **This query is equivalent to the relational algebra expression:**

$$\prod_{A_1, A_2, \ldots, A_n} (\sigma_P (r_1 \times r_2 \times \ldots \times r_m))$$

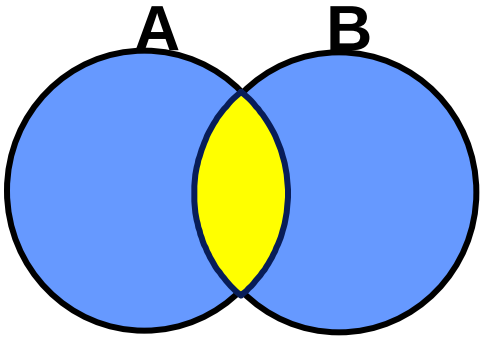
Example Schemas

```
Sailors(sid, sname, rating, age)

Boats(bid, bname, color)

Reserves(sid, bid, day)
   FOREIGN KEY (bid) REFERENCES Boats,
   FOREIGN KEY (sid) REFERENCES sailors
)
```

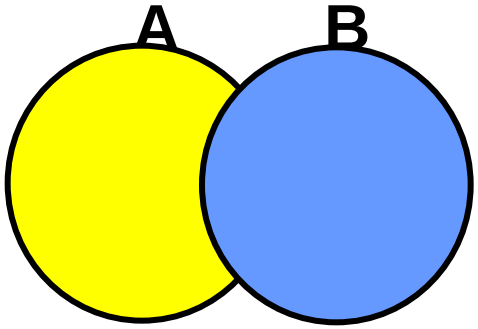
Set Operators



UNION/UNION ALL



INTERSECT



MINUS/EXCEPT

Set Operations

Union: R ∪ S

In SQL:

```
SELECT *  FROM R
UNION
SELECT  * FROM S;
```

Intersection: R ∩ S

In SQL:

```
SELECT *  FROM R
INTERSECT
SELECT * FROM S;
```

Set difference: R - S

In SQL:

```
SELECT *  FROM R
EXCEPT
SELECT  * FROM S;
```

Example of Union

Find sids of sailors who've reserved a red **or** a green boat

Solution 1 (Without using Set Operations)

$$\pi_{sid}(\sigma_{color='red' \vee color='green'} Boats \bowtie \text{Re}serves)$$

```
SELECT R.sid
FROM Boats B,Reserves R
WHERE R.bid=B.bid AND
(B.color='red'OR B.color='green')
```

Example of Union

Find sids of sailors who've reserved a red **or** a green boat

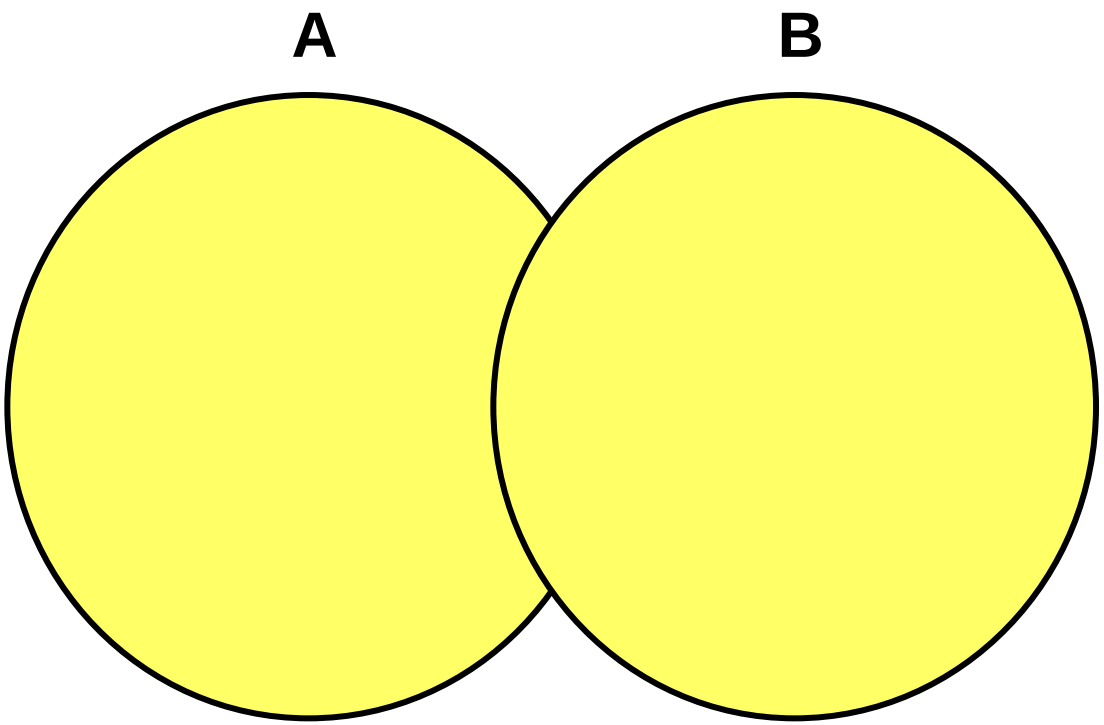
Solution 2 (with Set operations)

- Find sailors who've reserved red boats *Tempred*;
- Find sailors who've reserved green boats *Tempgreen*;
- find the Union of *Tempred* and *Tempgreen*

```
SELECT  R.sid
FROM  Boats B, Reserves R
WHERE R.bid=B.bid AND B.color='red'
UNION
SELECT R.sid
FROM  Boats B, Reserves R
WHERE R.bid=B.bid AND B.color='green'
```

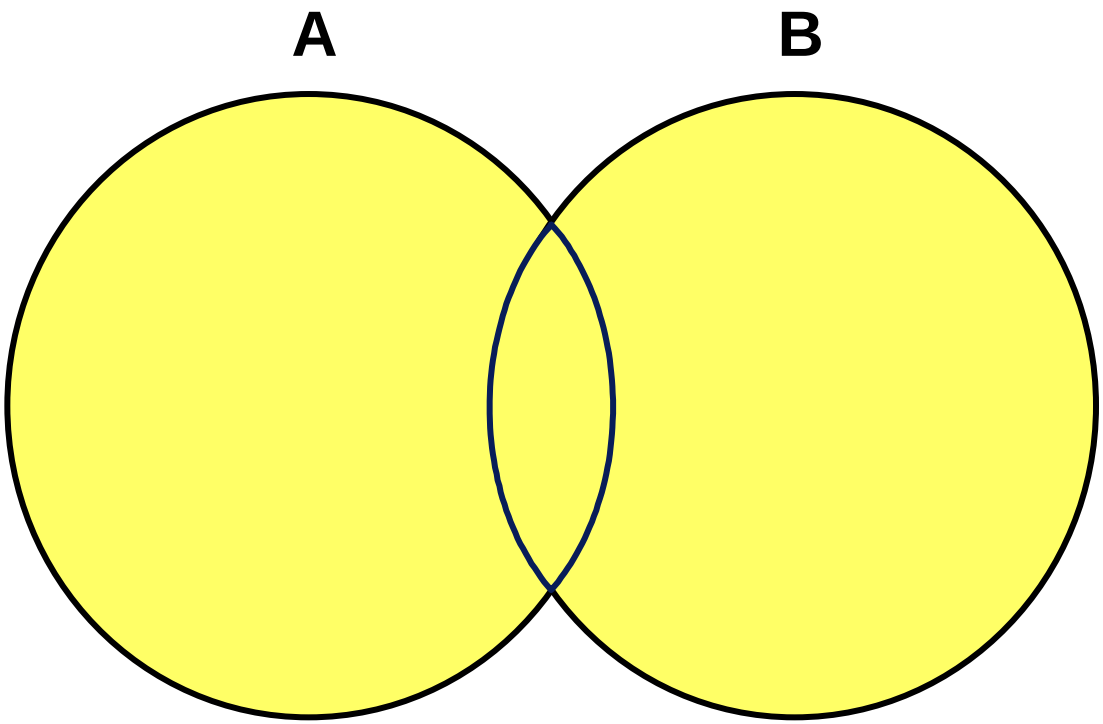
Question: how to revise the SQL statement if the query is changed to be "find names of sailors who've reserved a red or a green boat

UNION Operator



The `UNION` operator returns results from both queries after eliminating duplications.

UNION ALL Operator



The `UNION ALL` operator returns results from both queries, including all duplications.

Examples of UNION and UNION ALL

Sid	Sname	Rating	Age
22	Dustin	7	45.0
31	Lubber	8	55.5
58	Rusty	10	35.5

S1

Sid	Sname	Rating	Age
28	Yuppy	9	35.0
31	Lubber	8	55.5
44	Guppy	5	35.0
58	Rusty	10	35.5

S2

```
SELECT sname
FROM S1
WHERE Age >40
UNION
SELECT sname
FROM S2
WHERE Age >40
```

Sname
Dustin
Lubber

Result

Examples of UNION and UNION ALL

Sid	Sname	Rating	Age
22	Dustin	7	45.0
31	Lubber	8	55.5
58	Rusty	10	35.5

S1

Sid	Sname	Rating	Age
28	Yuppy	9	35.0
31	Lubber	8	55.5
44	Guppy	5	35.0
58	Rusty	10	35.5

S2

```
SELECT sname
FROM S1
WHERE Age >40
UNION ALL
SELECT sname
FROM S2
WHERE Age >40
```

Sname
Dustin
Lubber
Lubber

Result

Intersection

- **Find sids of sailors who've reserved a red <u>and</u> a green boat**
  - Identify sailors who've reserved red boats (in *Tempred*),
  - Identify sailors who've reserved green boats (in *Tempgreen*),
  - find the intersection of *Tempred* and *Tempgreen*

$$\rho(Tempred, \pi_{sid}((\sigma_{color='red'} Boats) \bowtie Reserves))$$

$$\rho\ (Tempgreen, \pi_{sid}((\sigma_{color='green'} Boats) \bowtie Reserves))$$

$$\pi_{sname}((Tempred \cap Tempgreen) \bowtie Sailors)$$

Intersection (Cont.)

- **INTERSECT: Can be used to compute the intersection of any two *union-compatible* sets of tuples.**

```
SELECT sid
FROM Boats B,  Reserves R
WHERE R.bid=B.bid AND B.color='red'
INTERSECT
SELECT sid
FROM Boats B,  Reserves R
WHERE R.bid=B.bid AND B.color='green';
```

Find sid's of sailors who've reserved a red **and** a green boat

- Can we write equivalent SQL statement without using set operations?

```
SELECT R.sid
FROM Boats B,Reserves R
WHERE R.bid=B.bid AND
(B.color='red' AND B.color='green')
```

Find sid's of sailors who've reserved a red **and** a green boat

- Can we write equivalent SQL statement without using set operations?
  - Hint: join with Boats table TWICE (one for red boats, one for greet boats)
  - Solution:

```
SELECT R1.sid
FROM Boats B1, Reserves R1,
     Boats B2, Reserves R2
WHERE R1.sid=R2.sid
      AND R1.bid=B1.bid
      AND R2.bid=B2.bid
      AND B1.color='red'
      AND B2.color='green';
```

Find sid's of sailors who've reserved a red **and** a green boat

- Can we write equivalent SQL statement without using set operations?
  - Hint: join with Boats table TWICE (one for red boat, one for greet boat)
  - Question: can we use one single Reserves table in the two joins (as shown below)?

```
SELECT R1.sid
FROM Boats B1, Boats B2, Reserves R
WHERE R.bid=B1.bid
      AND R.bid=B2.bid
      AND B1.color='red'
      AND B2.color='green';
```

Find sid's of sailors who've reserved a red **and** a green boat

- Can we write equivalent SQL statement without using set operations?
  - Hint: join with Boats table TWICE (one for red boat, one for greet boat)
  - Question: can we use one single Reserves table in the two joins (as shown below)?

```
SELECT R1.sid
FROM Boats B1, Boats B2, Reserves R
WHERE R.bid=B1.bid
      AND R.bid=B2.bid
      AND B1.color='red'
      AND B2.color='green';
```

Find sid's of sailors who've reserved a red **but did not reserve** a green boat

EXCEPT (sometimes called MINUS)

Included in the SQL/92 standard, but many systems don't support them.

```
SELECT sid
FROM Boats B,  Reserves R
WHERE R.bid=B.bid AND B.color='red'
EXCEPT
SELECT sid
FROM Boats B,  Reserves R
WHERE R.bid=B.bid AND B.color='green'
```