SQL: Structured Query Language

Chapter 5

Review

- Relational Algebra (Operational Semantics)
  - Compose “tree” of operators to answer query
  - Used for query plans
- Relational Calculus (Declarative Semantics)
  - Describe what a query’s answer set will include
- Simple and powerful models for query languages

Query Language

- Two sublanguages:
  - **DDL – Data Definition Language**
    - **Define and modify** schema
  - **DML – Data Manipulation Language**
    - Specify queries to **find/retrieve** tuples that satisfy criteria
    - Specify queries to **add/update/delete** tuples

- DBMS is responsible for efficient evaluation.
  - The key: precise semantics for relational queries
  - Optimizer can re-order operations
    - Won't affect query answer.

The SQL Query Language

- The most widely used relational query language.
- Standardized
  - (although most systems add their own “special sauce”)
- We will study basic constructs

Query Optimization

SQL Query

Rel. Algebra Query 1

Rel. Algebra Query 2

.
.
.

Rel. Algebra Query n

QP Engine picks the cheapest one

Example Database

Sailors

sid	sname	rating	age
1	Fred	7	22
2	Jim	2	39
3	Nancy	8	27

Boats

bid	bname	color
101	Nina	red
102	Pinta	blue
103	Santa Maria	red

Reserves

sid	bid	day
1	102	12/9/2015
2	102	13/9/2015

The SQL DDL

```
CREATE TABLE Sailors (
   sid INTEGER,
   sname CHAR(20),
   rating INTEGER,
   age REAL,
   PRIMARY KEY (sid));

CREATE TABLE Boats (
   bid INTEGER,
   bname CHAR (20),
   color CHAR(10),
   PRIMARY KEY (bid));

 CREATE TABLE Reserves (
   sid INTEGER,
   bid INTEGER,
   day DATE,
  PRIMARY KEY (sid, bid, day),
  FOREIGN KEY (sid) REFERENCES Sailors,
  FOREIGN KEY (bid) REFERENCES Boats);
```

sid	sname	rating	age
1	Fred	7	22
2	Jim	2	39
3	Nancy	8	27

bid	bname	color
101	Nina	red
102	Pinta	blue
103	Santa Maria	red

sid	**bid**	**day**
1	102	9/12
2	102	9/13

The SQL DML

- Find all sailors:

Sailors

sid	sname	rating	age
1	Fred	7	22
2	Jim	2	39
3	Nancy	8	27

The SQL DML

Sailors

sid	sname	rating	age
1	Fred	7	22
2	Jim	2	39
3	Nancy	8	27

- Find all sailors:

```
SELECT *
FROM  Sailors S
```

$$\pi_{sid,sname,rating,age}(S)$$

The SQL DML

- To find just `names` and `ratings`:

Sailors

sid	sname	rating	age
1	Fred	7	22
2	Jim	2	39
3	Nancy	8	27
4	Fred	7	45

Small addition to table

The SQL DML

Sailors

sid	sname	rating	age
1	Fred	7	22
2	Jim	2	39
3	Nancy	8	27
4	Fred	7	45

- To find just names and ratings:

```
SELECT S.sname, S.rating
FROM   Sailors S
```

sname	ratig
Fred	7
Jim	2
Nancy	8
Fred	7

MULTI-SET!

The SQL DML

- To find DISTINCT names and ratings (without duplicates):

Sailors

sid	sname	rating	age
1	Fred	7	22
2	Jim	2	39
3	Nancy	8	27
4	Fred	7	45

```
SELECT DISTINCT S.sname, S.rating
FROM   Sailors S
```

sname	rating
Fred	7
Jim	2
Nancy	8

The SQL DML

Sailors

sid	sname	rating	age
1	Fred	7	22
2	Jim	2	39
3	Nancy	8	27

- Find all 27-year-old sailors:

The SQL DML

Sailors

sid	sname	rating	age
1	Fred	7	22
2	Jim	2	39
3	Nancy	8	27

- Find all 27-year-old sailors:

```
SELECT *
FROM   Sailors S
WHERE  S.age=27
```

The SQL DML

Sailors

sid	sname	rating	age
1	Fred	7	22
2	Jim	2	39
3	Nancy	8	27

- Find all 27-year-old sailors:

```
SELECT *
FROM   Sailors S
WHERE  S.age=27
```

sid	sname	rating	age
3	Nancy	8	27

The SQL DML

Sailors

sid	sname	rating	age
1	Fred	7	22
2	Jim	2	39
3	Nancy	8	27

- To find DISTINCT names and ratings, replace 1st line as:

```
SELECT DISTINCT S.sname, S.rating
```

$$\pi_{sname,rating}(\sigma_{age=27}(S))$$

Basic SQL Query

DISTINCT: optional. Answer should not contain duplicates.

SQL default: duplicates are *not* eliminated! (Result is a "multiset")

target-list : List of expressions over attributes of tables in *relation-list*

qualification : Comparisons combined using AND, OR and NOT. Comparisons are:

Attr *op* const or Attr1 *op* Attr2, where *op* is one of >,<,=,≥,≤,≠ etc.

relation-list : List of relation names, possibly with a *range-variable* after each name

Query Semantics

SELECT [DISTINCT] *target-list*
FROM *relation-list*
WHERE *qualification*

1. FROM : compute cross product of tables.
2. WHERE : Check conditions, discard tuples that fail.
3. SELECT : Delete unwanted fields.
4. DISTINCT (optional) : eliminate duplicate rows.

- Note: likely a terribly inefficient strategy!
  - Query optimizer will find more efficient plans.

Conceptual SQL Evaluation

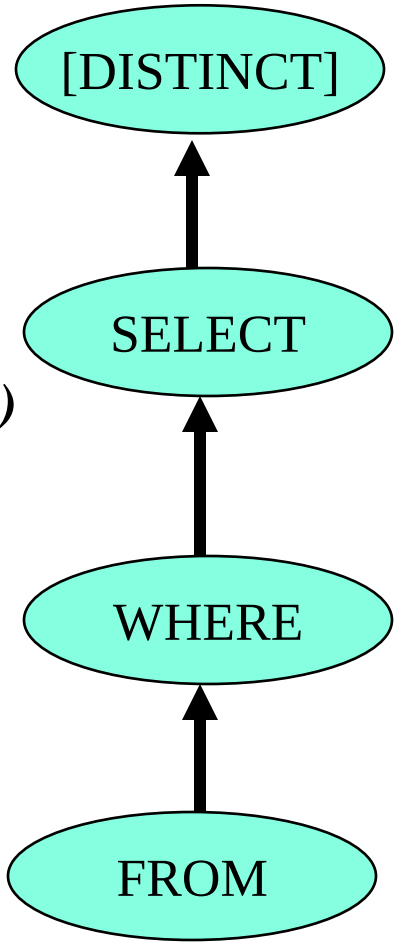
```
SELECT      [DISTINCT]  target-list
FROM        relation-list
WHERE       qualification
```

Eliminate duplicates

Project away columns (just keep those used in SELECT, GBY, HAVING)

Apply selections (eliminate rows)

Relation cross-product



Find sailor names who’ve reserved at least one boat

Find sailor names who've reserved at least one boat

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

Find sailor names who've reserved at least one boat

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

sid	sname	rating	age
1	Fred	7	22
2	Jim	2	39
3	Nancy	8	27

X

sid	bid	day
1	102	9/12
2	102	9/13
2	101	10/20
2	103	11/20

Find sailor names who've reserved at least one boat

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

S.sid	…	R.sid	R.bid	…
1		1	102	
1		2	102	
1		2	101	
1		2	103	
2		1	102	
2		2	102	
2		2	101	
2		2	103	
3		1	102	
3		2	102	
3		2	101	
3		2	103	

Find sailor names who've reserved at least one boat

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

S.sid	…	R.sid	R.bid	…
1		1	102	
1		2	102	
1		2	101	
1		2	103	
2		1	102	
2		2	102	
2		2	101	
2		2	103	
3		1	102	
3		2	102	
3		2	101	
3		2	103	

Find sailor names who've reserved at least one boat

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

S.sid	S.sname	R.sid	R.bid	…
1	Fred	1	102	
2	Jim	2	102	
2	Jim	2	101	
2	Jim	2	103	

Find sailor names who've reserved at least one boat

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

A sane QP engine will never really materialize cross product!

S.sid	S.sname	R.sid	R.bid	…
1	Fred	1	102	
2	Jim	2	102	
2	Jim	2	101	
2	Jim	2	103	

Find sailor names who've reserved at least one boat

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

- Would adding DISTINCT to this query make a difference?

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

?

Find names of sailors who've reserved boat 102

Sailors

sid	sname	rating	age
1	Fred	7	22
2	Jim	2	39
3	Nancy	8	27

Reserves

sid	bid	day
1	102	9/12
2	102	9/13
2	101	10/20
3	104	11/20

Find names of sailors who've reserved boat 102

```
SELECT S.sname
FROM   Sailors S, Reserves R
WHERE  S.sid=R.sid AND R.bid=102
```

Sailors

sid	sname	rating	age
1	Fred	7	22
2	Jim	2	39
3	Nancy	8	27

Reserves

sid	bid	day
1	102	9/12
2	102	9/13
2	101	10/20
3	104	11/20

Find names of sailors who've reserved boat 102

```
SELECT S.sname
FROM   Sailors S, Reserves R
WHERE  S.sid=R.sid AND R.bid=102
```

Sailors

sid	sname	rating	age
1	Fred	7	22
2	Jim	2	39
3	Nancy	8	27

Reserves

sid	bid	day
1	102	9/12
2	102	9/13
2	101	10/20
3	104	11/20

Important !!!

What if you forget to write the join condition !

```
SELECT S.sname
FROM   Sailors S, Reserves R
WHERE  R.bid=102
```

A Note on Range Variables

- Really needed only if ambiguity could arise.
- The previous query can also be written as:

```
SELECT sname
FROM   Sailors, Reserves
WHERE  Sailors.sid=Reserves.sid
       AND bid=102
```

It is good style,however, to use range variables always!

About Range Variables

List pairs of sailors where the first sailor is older than the second.

– same table used multiple times in FROM (**self-join**)

```
SELECT  X.sname, X.age, Y.sname, Y.age
FROM    Sailors X, Sailors Y
WHERE   X.age > Y.age
```

About Range Variables

```
SELECT  X.sname, X.age, Y.sname, Y.age
FROM    Sailors X, Sailors Y
WHERE   X.age > Y.age
```

Sailors

sid	sname	rating	age
1	Fred	7	22
2	Jim	2	39
3	Nancy	8	27

X

sid	sname	rating	age
1	Fred	7	22
2	Jim	2	39
3	Nancy	8	27

Y

sid	sname	rating	age
1	Fred	7	22
2	Jim	2	39
3	Nancy	8	27

About Range Variables

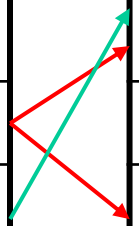
```
SELECT  X.sname, X.age, Y.sname, Y.age
FROM    Sailors X, Sailors Y
WHERE   X.age > Y.age
```

X

sid	sname	rating	age
1	Fred	7	22
2	Jim	2	39
3	Nancy	8	27

Y

sid	sname	rating	age
1	Fred	7	22
2	Jim	2	39
3	Nancy	8	27



X.sname	X.age	Y.sname	Y.age
Jim	39	Fred	22
Jim	39	Nancy	27
Nancy	27	Fred	22

Arithmetic Expressions

```
SELECT S.age, S.age-5 AS age1, 2*S.age AS age2
FROM   Sailors S
WHERE  S.sname = 'dustin'
```

```
SELECT S1.sname AS name1, S2.sname AS name2
FROM   Sailors S1, Sailors S2
WHERE  2*S1.rating = S2.rating - 1
```

String Comparisons

```
SELECT S.sname
FROM   Sailors S
WHERE  S.sname LIKE ‘A_%A'
```

ANA
ANNA
AYLA
ALINA
…

‘_’ stands for any one character and ‘%’ stands for 0 or more arbitrary characters.

Most DBMSs now support standard regex as well

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

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

sid
3
3
2

Reserves

sid	bid	day
1	103	12/9/2015
2	102	13/9/2015
2	103	1/1/2020
3	101	1/1/2020
3	102	5/1/2020

Boats

bid	bname	color
101	Nina	red
102	Pinta	green
103	Santa Maria	blue

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

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

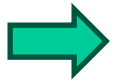
Reserves

sid	bid	day
1	103	12/9/2015
2	102	13/9/2015
2	103	1/1/2020
3	101	1/1/2020
3	102	5/1/2020

Boats

bid	bname	color
101	Nina	red
102	Pinta	green
103	Santa Maria	blue

R.sid	R.bid	R.day	B.bid	B.bname	B.color
1	103	12/9/2015	103	Santa Maria	blue
2	102	13/9/2015	102	Pinta	green
2	103	1/1/2020	103	Santa Maria	blue
3	101	1/1/2020	101	Nina	red
3	102	5/1/2020	102	Pinta	green



R.sid
3
3
2

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

... or:

```
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'
```

Reserves

sid	bid	day
1	103	12/9/2015
2	102	13/9/2015
2	103	1/1/2020
3	101	1/1/2020
3	102	5/1/2020

Boats

bid	bname	color
101	Nina	red
102	Pinta	green
103	Santa Maria	blue

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

... or:

```
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'
```

sid
3

UNION

sid
3
2

sid
3
2

Reserves

sid	bid	day
1	103	12/9/2015
2	102	13/9/2015
2	103	1/1/2020
3	101	1/1/2020
3	102	5/1/2020

Boats

bid	bname	color
101	Nina	red
102	Pinta	green
103	Santa Maria	blue

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

... or:

```
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'
```

sid
3

UNION

sid
3
2

sid
3
2

UNION, EXCEPT, INTERSECT eliminate DUPLICATES!

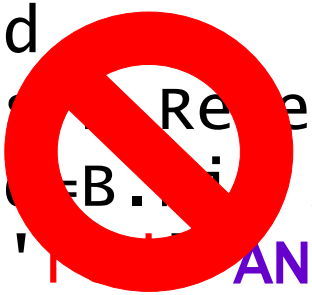
UNION ALL, EXCEPT ALL, INTERSECT ALL keep DUPLICATES!

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

```
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

```
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

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

sid
3

INTERSECT

sid
3
2

sid
3

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

→ Solution with **self join**

Reserves → R1 R2
Boats → B1 B2

Reserves

sid	bid
3	101
3	102
2	102

Boats

bid	color
101	red
102	green

R1.sid	R1.bid	B1.color
3	101	red
3	102	green
2	102	green

R2.sid	R2.bid	B2.color
3	101	red
3	102	green
2	102	green

Should be same

Red boat

Green boat

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

→ Self join based solution

Another way to consider the joins

Reserves

sid	bid
3	101
3	102
2	102

Boats

bid	color
101	red
102	green

Reserves → R1 R2
Boats → B1 B2

R1.sid	R1.bid
3	101
3	101
3	102
3	102
2	102

R2.sid	R2.bid
3	101
3	102
3	101
3	102
2	102

B1.bid	B1.color
101	red
101	red
102	green
102	green
102	green

B2.bid	B2.color
101	red
102	green
101	red
102	green
102	green

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

→ Solution with **self join**
Another way to consider the joins

Reserves

sid	bid
3	101
3	102
2	102

Boats

bid	color
101	red
102	green

Reserves → R1 R2
Boats → B1 B2

R1.sid	R1.bid
3	101
3	101
3	102
3	102
2	102

R2.sid	R2.bid
3	101
3	102
3	101
3	102
2	102

B1.bid	B1.color
101	red
101	red
~~102~~	~~green~~
~~102~~	~~green~~
~~102~~	~~green~~

B2.bid	B2.color
~~101~~	~~red~~
102	green
~~101~~	~~red~~
102	green
102	green

Red boat

Green boat

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

→ Solution with **self join**
Another way to consider the joins

Reserves

sid	bid
3	101
3	102
2	102

Boats

bid	color
101	red
102	green

Reserves → R1 R2
Boats → B1 B2

R1.sid	R1.bid	R2.sid	R2.bid	B1.bid	B1.color	B2.bid	B2.color
3	101	3	101	101	red	~~101~~	~~red~~
3	101	3	102	101	red	102	green
3	102	3	101	~~102~~	~~green~~	~~101~~	~~red~~
3	102	3	102	102	green	102	green
2	102	2	102	~~102~~	~~green~~	102	green

Red boat

Green boat

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

- With self-join:

```
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'
```

DISTINCT would it help here?

Find sid and names of sailors who have not reserved boat#102

sid	sname	rating	age
1	Fred	7	22
2	Jim	2	39
3	Nancy	8	27

sid	bid	day
1	102	9/12
2	102	9/13
2	101	10/20
3	104	11/20

Find sid and names of sailors who have not reserved boat#102

(sid and names of all sailors)

EXCEPT

(sid and names of sailors who reserved boat#102)

sid	sname	rating	age
1	Fred	7	22
2	Jim	2	39
3	Nancy	8	27

sid	bid	day
1	102	9/12
2	102	9/13
2	101	10/20
3	104	11/20

Find sid and names of sailors who have not reserved boat#102

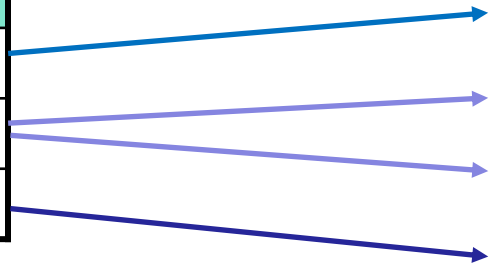
```
SELECT S.sid, S.sname
FROM   Sailors S

EXCEPT

SELECT S.sid, S.sname
FROM   Sailors S, Reserves R
WHERE  S.sid=R.sid AND
       R.bid=102
```

sid	sname
1	Fred
2	Jim
3	Nancy

sid	sname
3	Nancy

sid	sname
1	Fred
2	Jim

sid	sname	rating	age
1	Fred	7	22
2	Jim	2	39
3	Nancy	8	27



sid	bid	day
1	102	9/12
2	102	9/13
2	101	10/20
3	104	11/20

Find sid and names of sailors who have not reserved boat#102

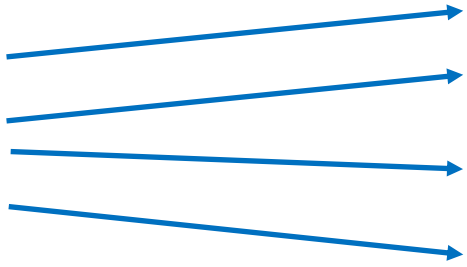
```
SELECT S.sid, S.sname
FROM   Sailors S

EXCEPT

SELECT S.sid, S.sname
FROM   Sailors S, Reserves R
WHERE  S.sid=R.sid AND
       R.bid=102
```

sid	sname
1	Fred
2	Jim
3	Nancy

sid	sname
3	Nancy

sid	sname
1	Fred
2	Jim

sid	sname	rating	age
1	Fred	7	22
2	Jim	2	39
3	Nancy	8	27



sid	bid	day
1	102	9/12
2	102	9/13
2	101	10/20
3	104	11/20

Find sid ~~and names~~ of sailors who have not reserved boat#102

```
SELECT S.sid, S.sname
FROM   Sailors S

EXCEPT

SELECT S.sid, S.sname
FROM   Sailors S, Reserves R
WHERE  S.sid=R.sid AND
       R.bid=102
```

sid	sname	rating	age
1	Fred	7	22
2	Jim	2	39
3	Nancy	8	27
4	Mary	1	17

sid	bid	day
1	102	9/12
2	102	9/13
2	101	10/20
3	104	11/20

Find sid ~~and names~~ of sailors who have not reserved boat#102

```
SELECT S.sid, S.sname
FROM   Sailors S

EXCEPT

SELECT S.sid, S.sname
FROM   Sailors S, Reserves R
WHERE  S.sid=R.sid AND
       R.bid=102
```

sid	sname	rating	age
1	Fred	7	22
2	Jim	2	39
3	Nancy	8	27
4	Mary	1	17

sid	bid	day
1	102	9/12
2	102	9/13
2	101	10/20
3	104	11/20

Find sid ~~and names~~ of sailors who have not reserved boat#102

```
SELECT S.sid, S.sname
FROM   Sailors S

EXCEPT

SELECT S.sid, S.sname
FROM   Sailors S, Reserves R
WHERE  S.sid=R.sid AND
       R.bid=102
```

Can we use Reserves instead?

When is it ok?

sid	sname	rating	age
1	Fred	7	22
2	Jim	2	39
3	Nancy	8	27
4	Mary	1	17

sid	bid	day
1	102	9/12
2	102	9/13
2	101	10/20
3	104	11/20