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

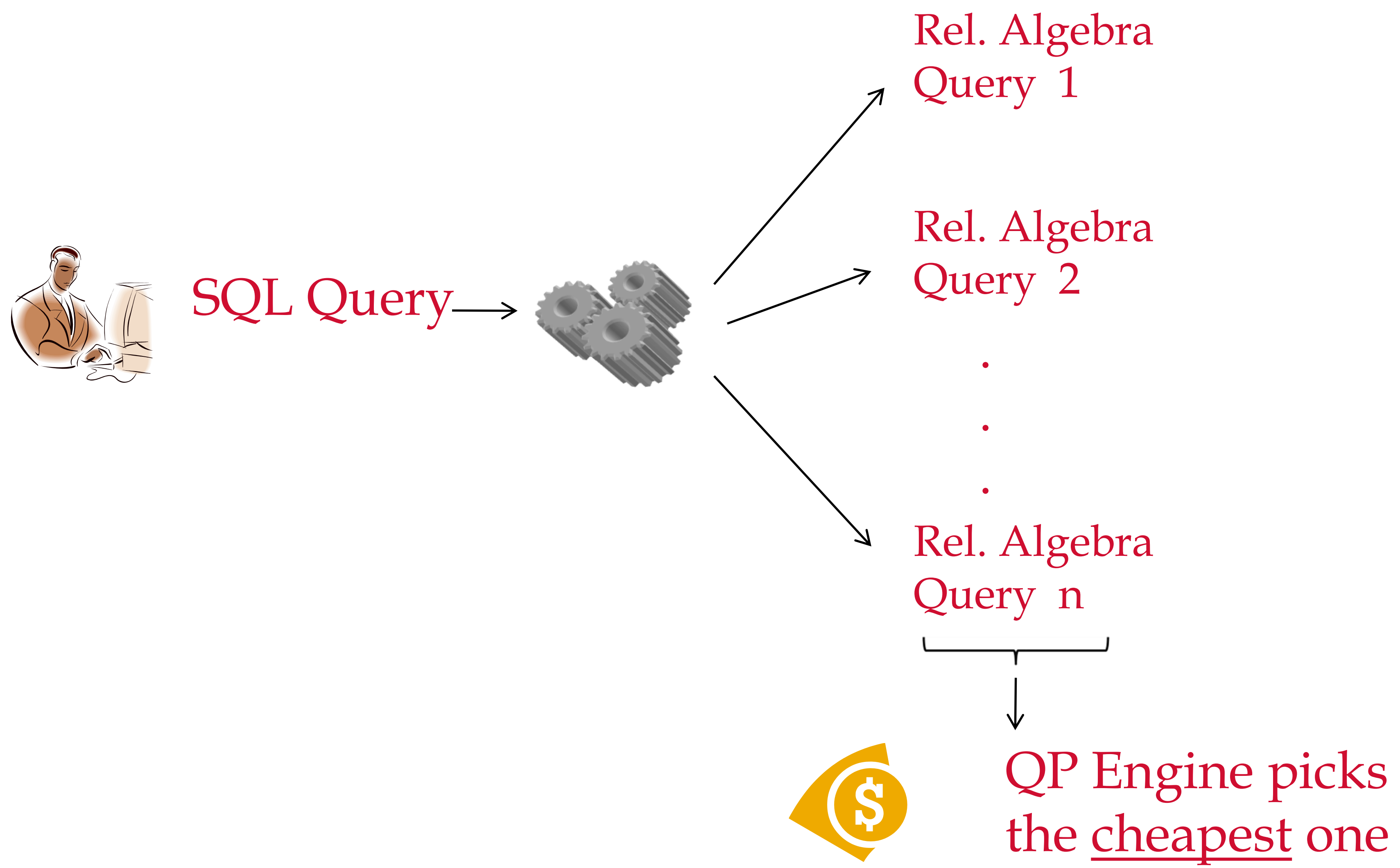
- Two sublanguages:
  - **DDL – Data Definition Language**
    - **Define and modify** schema
  - **DML – Data Manipulation Language**
    - Specify queries to **find/retrieve** tuples that satisfy criteria
    - Specify queries to **add/update/delete** tuples

- 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



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);
```

<u>sid</u>	sname	rating	age
1	Fred	7	22
2	Jim	2	39
3	Nancy	8	27

<u>bid</u>	bname	color
101	Nina	red
102	Pinta	blue
103	Santa Maria	red

<u>sid</u>	**<u>bid</u>**	**<u>day</u>**
1	102	9/12
2	102	9/13

The SQL DML

- Find all sailors:

```
SELECT *
FROM   Sailors S
```

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

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
```

MULTI-SET!
(except…)

sname	rating
Fred	7
Jim	2
Nancy	8
Fred	7

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

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

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
```

- 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*

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

qualification : Comparisons combined using AND, OR and NOT. Comparisons are:
Attr *op* const or Attr1 *op* Attr2,
where *op* is one of $>, <, =, \geq, \leq, \neq$ etc.

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

Query Semantics

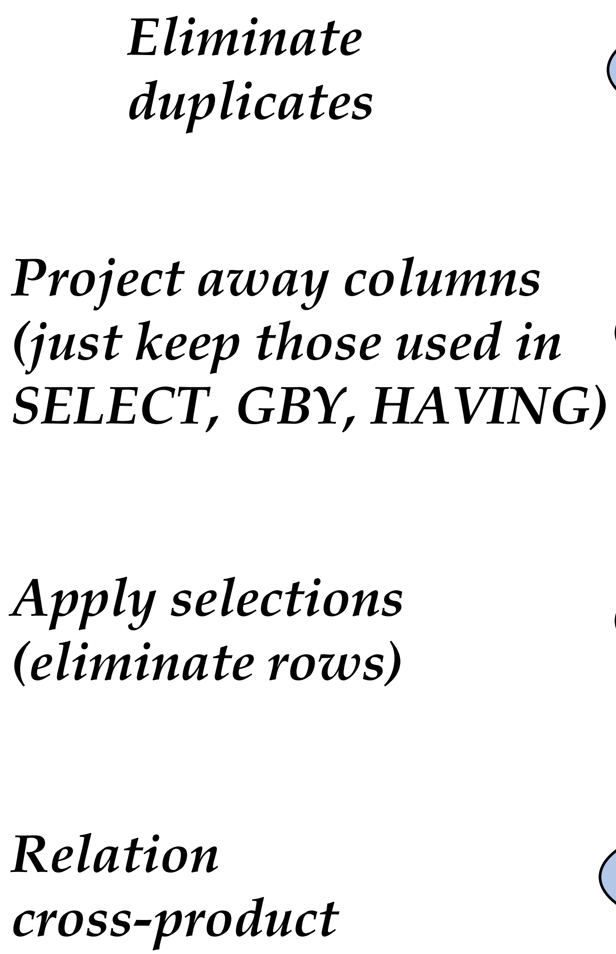
```
SELECT  [DISTINCT]  target-list
FROM                relation-list
WHERE               qualification
```

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
[DISTINCT]
Project away columns (just keep those used in SELECT, GBY, HAVING)
SELECT
Apply selections (eliminate rows)
WHERE
Relation cross-product
FROM

Find sailor names who've reserved at least one boat

A sane QP engine will never really materialize cross product!

```
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

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~~	

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

- Would adding DISTINCT to this query make a difference?

You may lose points in the exam!

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

If you want to **join tables**:
Don’t forget the condition!