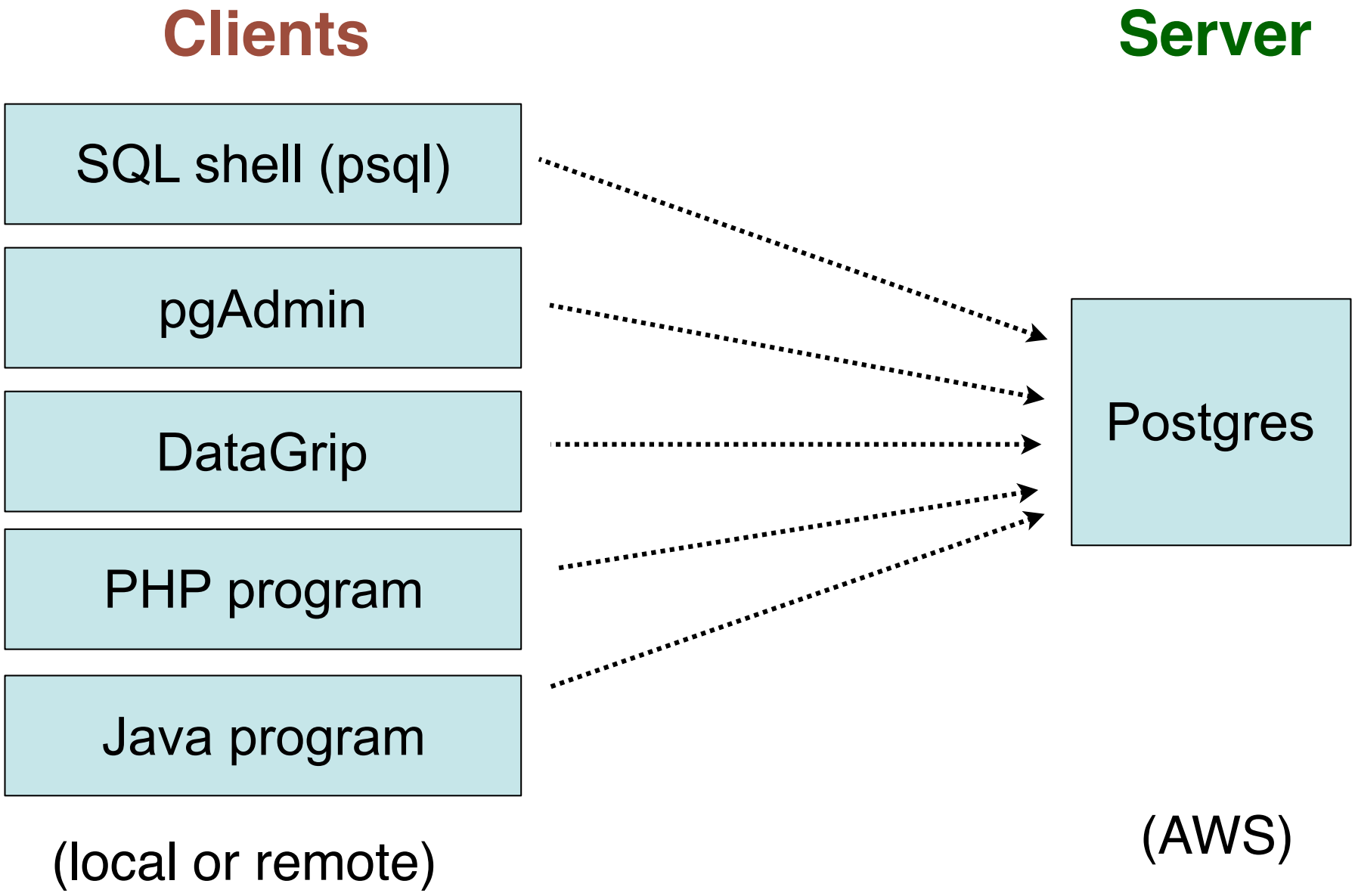
SQL
Review

CMPSCI 445

Spring 2018

Connecting to Postgres

(Needed for HW 2)



Connection parameters

- host:port
- database-name
- username
- password

The SQL Query Language

Structured **Q**uery **L**anguage

- Developed by IBM (system R) in the 1970s
- Need for a standard since it is used by many vendors
- Evolving standard
  - SQL-86
  - SQL-89 (minor revision)
  - SQL-92 (major revision)
  - SQL-99 (major extensions)
  - SQL-2003 (minor revisions)
  - SQL-2006 (XML related revisions)
  - SQL-2008 (minor revisions)
  - SQL-2011 (minor revisions)

Two parts of SQL

- Data Definition Language (DDL)
  - Create/alter/delete tables and their attributes
  - establish and modify schema
- Data Manipulation Language (DML)
  - Query and modify database instance

SQL Overview

- Query capabilities
  - SELECT-FROM-WHERE blocks,
  - Basic features, ordering, duplicates
  - Set operations (union, intersect, except)
  - Aggregation & Grouping
    - Nested queries (correlation)
  - Null values

Example database

Sailors (sid, sname, rating, age)
Boats (bid, bname, color)
Reserves (sid, bid, day)

Sailors

sid	sname	rating	age
29	brutus	1	33
85	art	3	25.5
95	bob	3	63.5
96	frodo	3	25.5
22	dustin	7	45
64	horatio	7	35
31	lubber	8	55.5
32	andy	8	25.5
74	horatio	9	35
58	rusty	10	35
71	zorba	10	16

Reserves

sid	bid	day
22	101	10/10
22	102	10/10
22	103	10/8
22	104	10/7
31	102	11/10
31	103	11/6
31	104	11/12
64	101	9/5
64	102	9/8
74	103	9/8

Boats

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

SQL Query

Basic form: (plus many many extensions)

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

For example:

```
SELECT    sid, sname, rating, age
FROM      Sailors
WHERE     age > 21
```

Basic SQL Query

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

- ***target-list*** A list of attributes of relations in *relation-list*
- ***relation-list*** A list of relation names (possibly with a *range-variable* after each name).
- ***qualification*** Comparisons (Attr *op* const or Attr1 *op* Attr2, where *op* is one of $<, >, =, \leq, \geq, \neq$) combined using AND, OR and NOT.
- DISTINCT is an optional keyword indicating that the answer should not contain duplicates. Default is that duplicates are *not* eliminated!

Note confusing terminology ☹

Relational Algebra v. SQL

```
SELECT    sname, age
FROM      Sailors
WHERE     age > 21
```

Conditions in the WHERE clause are like selection: $\sigma_{age<21}$

Conditions in the SELECT clause are like projection: $\Pi_{sname,age}$

Eliminating Duplicates

```
SELECT  DISTINCT sname
FROM    Sailors
```

Compare to:

```
SELECT  sname
FROM    Sailors
```

sname
brutus
art
bob
frodo
dustin
horatio
lubber
andy

sname
brutus
art
bob
frodo
dustin
horatio
lubber
andy
horatio

Default behavior does **not** eliminate duplicates.

Ordering the Results

```
SELECT   sname, rating, age
FROM     Sailors
WHERE    age > 18
ORDER BY  rating, sname
```

Ordering is ascending, unless you specify the DESC keyword.

Ties are broken by the second attribute on the ORDER BY list, etc.

i-clicker #1

- What does the following SQL query compute on the Sailors instance below?

```
SELECT   sid, sname, rating
FROM     Sailors
WHERE    age < 50
ORDER BY sname, rating
```

Sailors

sid	sname	rating	age
29	brutus	1	33
85	art	3	25.5
95	bob	3	63.5
96	frodo	3	25.5

i-clicker #1

Select the correct output relation:

A

sid	sname	rating
29	brutus	1
85	art	3
96	frodo	3

B

sid	sname	rating
85	art	3
29	brutus	1
96	frodo	3

C

sid	sname	rating	age
85	art	3	25.5
29	brutus	1	33
96	frodo	3	25.5

D

sname	rating
brutus	1
art	3
frodo	3

Answer on next slide

i-clicker #1

Select the correct output relation:

A

sid	sname	rating
29	brutus	1
85	art	3
96	frodo	3

B

sid	sname	rating
85	art	3
29	brutus	1
96	frodo	3

C

sid	sname	rating	age
85	art	3	25.5
29	brutus	1	33
96	frodo	3	25.5

D

sname	rating
brutus	1
art	3
frodo	3

Conceptual Evaluation Strategy

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

- Semantics of an SQL query defined in terms of a conceptual evaluation strategy:
  - Compute the cross-product of ***relation-list***.
  - Discard resulting tuples if they fail ***qualifications***.
  - Delete attributes that are not in ***target-list***.
  - If **DISTINCT** is specified, eliminate duplicate rows.
- Probably the least efficient way to compute a query -- optimizer will find more efficient plan.

RA equiv:

$\times$

σ

Π

Example of Conceptual Evaluation

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0

sid	bid	day
22	101	10/10/96
58	103	11/12/96

```
SELECT  S.sname
FROM    Sailors S, Reserves R
WHERE   S.sid=R.sid AND R.bid=103
```

(sid)	sname	rating	age	(sid)	bid	day
22	dustin	7	45.0	22	101	10/10/96
22	dustin	7	45.0	58	103	11/12/96
31	lubber	8	55.5	22	101	10/10/96
31	lubber	8	55.5	58	103	11/12/96
58	rusty	10	35.0	22	101	10/10/96
58	rusty	10	35.0	58	103	11/12/96

Example

```
SELECT sname
FROM Sailors S, Reserves R, Boats B
WHERE S.sid = R.sid AND R.bid = B.bid
      AND B.color = ‘red’
```

What does this query compute?

Find the names of sailors who have reserved a red boat

i-clicker #2

- Which SQL query computes the following:

Find the colors of boats reserved by 'Lubber'

A

```
SELECT B.color
FROM Sailors S, Reserves R, Boats B
WHERE S.sid = R.sid AND R.bid = B.bid
```

B

```
SELECT B.color
FROM Sailors S, Boats B
WHERE S.sid = B.bid AND S.sname = 'Lubber'
```

C

```
SELECT B.color
FROM Sailors S, Reserves R, Boats B
WHERE S.sid = R.sid AND R.bid = B.bid AND S.sname = 'Lubber'
```

Answer on next slide

i-clicker #2

- Which SQL query computes the following:

Find the colors of boats reserved by 'Lubber'

A

```
SELECT B.color
FROM Sailors S, Reserves R, Boats B
WHERE S.sid = R.sid AND R.bid = B.bid
```

B

```
SELECT B.color
FROM Sailors S, Boats B
WHERE S.sid = B.bid AND S.sname = 'Lubber'
```

C

```
SELECT B.color
FROM Sailors S, Reserves R, Boats B
WHERE S.sid = R.sid AND R.bid = B.bid AND S.sname = 'Lubber'
```

Range Variables in SQL

Purchase (buyer, seller, store, product)

Find all stores that sold at least one product that was sold at 'BestBuy':

```
SELECT DISTINCT  x.store
FROM     Purchase AS x, Purchase AS y
WHERE   x.product = y.product AND y.store = 'BestBuy'
```

Please write in SQL

Self-join on Flights:
The departure and arrival cities of trips consisting of two direct flights.

```
SELECT   F1.depart, F2.arrive
FROM     Flights as F1, Flights as F2
WHERE    F1.arrive = F2.depart
```

FLIGHTS

depart	arrive
NYC	Reno
NYC	Oakland
Boston	Tampa
Oakland	Boston
Tampa	NYC

SQL Overview

- Query capabilities
  - SELECT-FROM-WHERE blocks,
  - Basic features, ordering, duplicates
  - Set operations (union, intersect, except)
  - Aggregation & Grouping
    - Nested queries (correlation)
  - Null values

Set operations

- UNION
- INTERSECTION
- EXCEPT (sometimes called MINUS)

- Recall: schemas must match for these operations.

UNION example

Find the names of sailors who have reserved a red or a green boat.

```
SELECT sname
FROM Sailors S, Reserves R, Boats B
WHERE S.sid = R.sid AND R.bid = B.bid AND B.color = ‘red’

UNION

SELECT sname
FROM Sailors S, Reserves R, Boats B
WHERE S.sid = R.sid AND R.bid = B.bid AND B.color = ‘green’
```

UNION

- Recall: duplicates **ARE NOT** eliminated by default in basic SELECT-FROM-WHERE queries.
- Duplicates **ARE** eliminated by default for UNION queries.
- To preserve duplicates in UNION, you must use UNION ALL

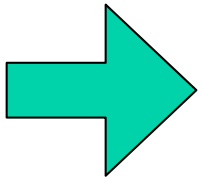
UNION example, alternative:

Find the names of sailors who have reserved a red or a green boat.

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

A small change in this query...

*Find the names of sailors who have reserved a red **or** a green boat.*



*Find the names of sailors who have reserved a red **and** a green boat.*

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

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

This doesn't work! What does this query return?

Find the names of sailors who have reserved a red ***and*** *a green boat.*

```
SELECT sid, sname
FROM Sailors S, Reserves R, Boats B
WHERE S.sid = R.sid AND R.bid = B.bid AND B.color = ‘red’

INTERSECT

SELECT sid, sname
FROM Sailors S, Reserves R, Boats B
WHERE S.sid = R.sid AND R.bid = B.bid AND B.color = ‘green’
```

SQL Overview

- Query capabilities
  - SELECT-FROM-WHERE blocks,
  - Basic features, ordering, duplicates
  - Set ops (union, intersect, except)
  - Aggregation & Grouping
    - Nested queries (correlation)
  - Null values

Aggregation

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

SQL supports several aggregation operations:

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

Aggregation: Count

```
SELECT  Count(*)
FROM    Sailors
WHERE   rating > 5
```

Except for COUNT, all aggregations apply to a single attribute

Aggregation: Count

COUNT applies to duplicates, unless otherwise stated:

```
SELECT  Count(category)
FROM    Product
WHERE   year > 1995
```

Better:

```
SELECT  Count(DISTINCT category)
FROM    Product
WHERE   year > 1995
```

Aggregation examples

```
CREATE TABLE temp (
  type character(1),
  x integer,
  y double precision
)
```

type	x	y
A	1	10.1
A	2	5.1
A	5	20.2
B	3	0
B	2	0.1
B	1	5.1
B	3	20.2

Query	Result
select count(*)	**7**
select count(x)	**7**
select count(distinct x)	**4**
select count(x,y)	**error**
select count(distinct x), count(distinct y)	**(4,5)**
select min(x), max(y)	**(1,20.2)**
select min(x), max(x)	**(1,5)**
select min(x) ...where type = 'A'	**1**
select min(x) ...where type = 'B'	**1**

Simple Aggregation

Purchase(product, date, price, quantity)

Example 1: find total sales for the entire database

```
SELECT  Sum(price * quantity)
FROM    Purchase
```

Example 1': find total sales of bagels

```
SELECT  Sum(price * quantity)
FROM    Purchase
WHERE   product = 'bagel'
```

GROUP BY and HAVING clauses

- We often want to apply aggregates to each of a number of groups of rows in a relation.

Find the age of the youngest sailor for each rating level.

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

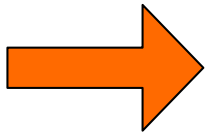
For i = 1, 2, ... 10

Grouping

SAILORS

sid	sname	rating	age
29	brutus	1	33
85	art	3	25.5
95	bob	3	63.5
96	frodo	3	25.5
22	dustin	7	45
64	horatio	7	35
31	lubber	8	55.5
32	andy	8	25.5
74	horatio	9	35
58	rusty	10	35
71	zorba	10	16

```
SELECT S.rating, MIN(S.age)
FROM  Sailors S
GROUP BY S.rating
```



New Table

rating	min
1	33
3	25.5
7	35
8	25.5
9	35
10	16

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 names** (ii) **terms with aggregate operations** (e.g., MIN (*S.age*)).
  - The attribute list (i) must be a subset of *grouping-list*. Intuitively, each answer tuple corresponds to a *group,* and these attributes must have a single value per group.

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.

i-clicker

What is the result of the following query?

R

dept	rate
admin	10
sales	5
admin	25
sales	11
sales	15
hr	15
sales	21
hr	13
admin	9

```
SELECT R.dept, MAX(R.rate)
FROM  R
GROUP BY R.dept
```

i-clicker

Answers:

A

dept	rate
admin	10
hr	13
sales	21

B

dept	rate
admin	25
hr	15
sales	21

C

dept	rate
admin	9
hr	13
sales	5

Answer on next slide

i-clicker

Answers:

A

dept	rate
admin	10
hr	13
sales	21

B

dept	rate
admin	25
hr	15
sales	21

C

dept	rate
admin	9
hr	13
sales	5

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

sid	sname	rating	age
29	brutus	1	33
85	art	3	25.5
95	bob	3	63.5
96	frodo	3	25.5
22	dustin	7	45
64	horatio	7	35
31	lubber	8	55.5
32	andy	8	25.5
74	horatio	9	35
58	rusty	10	35
71	zorba	10	16

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

Answer relation:

rating	minage
3	25.5
7	35
8	25.5