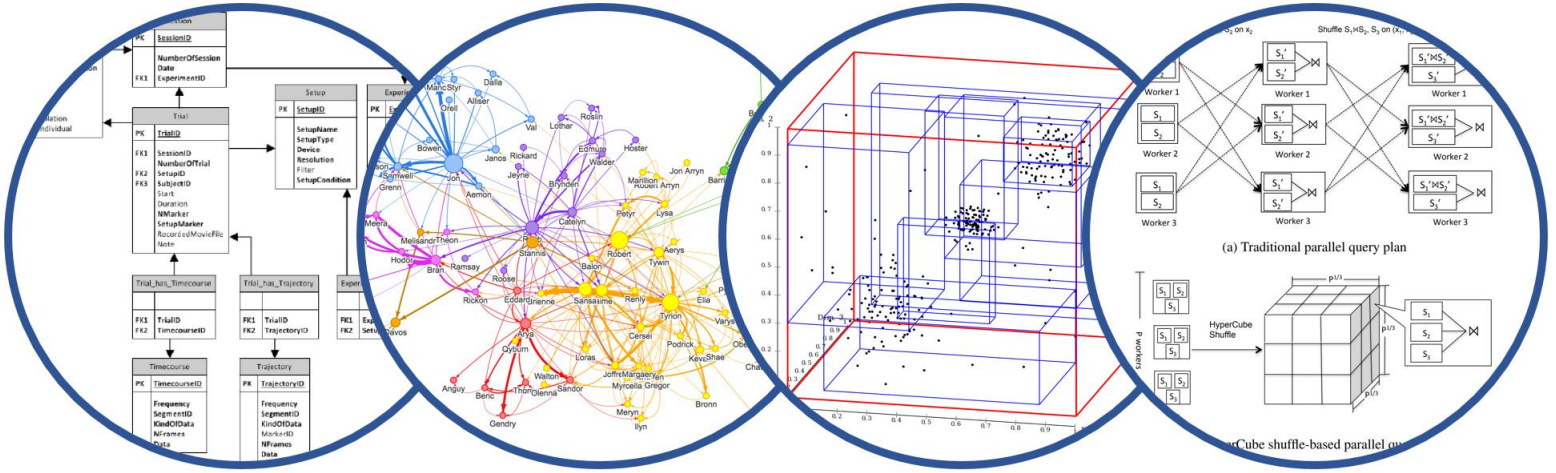


CSE 344: Intro to Data Management

Nulls and Aggregates

Paul G. Allen School of Computer Science and Engineering
University of Washington, Seattle

Announcements

Homework 1 due tonight!

Homework 2:

- Posted
- Due next Wednesday
- Sqlite

Recap: Inner Join

For each employee, find the cars that they drive

```
SELECT P.Name, R.Car
FROM Payroll AS P
     JOIN Regist AS R
     ON P.UserID = R.UserID;
```

Name	Car
Jack	Charger
Magda	Civic
Magda	Pinto

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

Regist

UserID	Car
123	Charger
567	Civic
567	Pinto

Recap: Inner Join

For each employee, find the cars that they drive

```
SELECT P.Name, R.Car
FROM Payroll AS P
     JOIN Regist AS R
     ON P.UserID = R.UserID;
```

Name	Car
Jack	Charger
Magda	Civic
Magda	Pinto

Allison, Dan are missing

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

Regist

UserID	Car
123	Charger
567	Civic
567	Pinto

Recap: Outer Join

For each employee, find the cars that they drive

```
SELECT P.Name, R.Car
FROM Payroll AS P
     LEFT OUTER JOIN Regist AS R
     ON P.UserID = R.UserID;
```

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

Regist

UserID	Car
123	Charger
567	Civic
567	Pinto

Recap: Outer Join

For each employee, find the cars that they drive

```
SELECT P.Name, R.Car
FROM Payroll AS P
     LEFT OUTER JOIN Regist AS R
     ON P.UserID = R.UserID;
```

Name	Car
Jack	Charger
Magda	Civic
Magda	Pinto
Allison	NULL
Dan	NULL

NULL means "unknown" or "missing"

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

Regist

UserID	Car
123	Charger
567	Civic
567	Pinto

Recap: Outer Join

- LEFT OUTER JOIN
  - Add missing tuples from the LEFT
- RIGHT OUTER JOIN
  - Add missing tuples from the RIGHT
- FULL OUTER JOIN
  - Add missing tuples from both

Let’s discuss NULLs…

NULLs

NULLs in SQL

A NULL value means missing, or unknown, or undefined, or inapplicable

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	NULL	60000
567	Magda	Prof	90000
789	Dan	Prof	NULL
432	NULL	Prof	NULL

NULLs in SQL

A NULL value means missing, or unknown, or undefined, or inapplicable

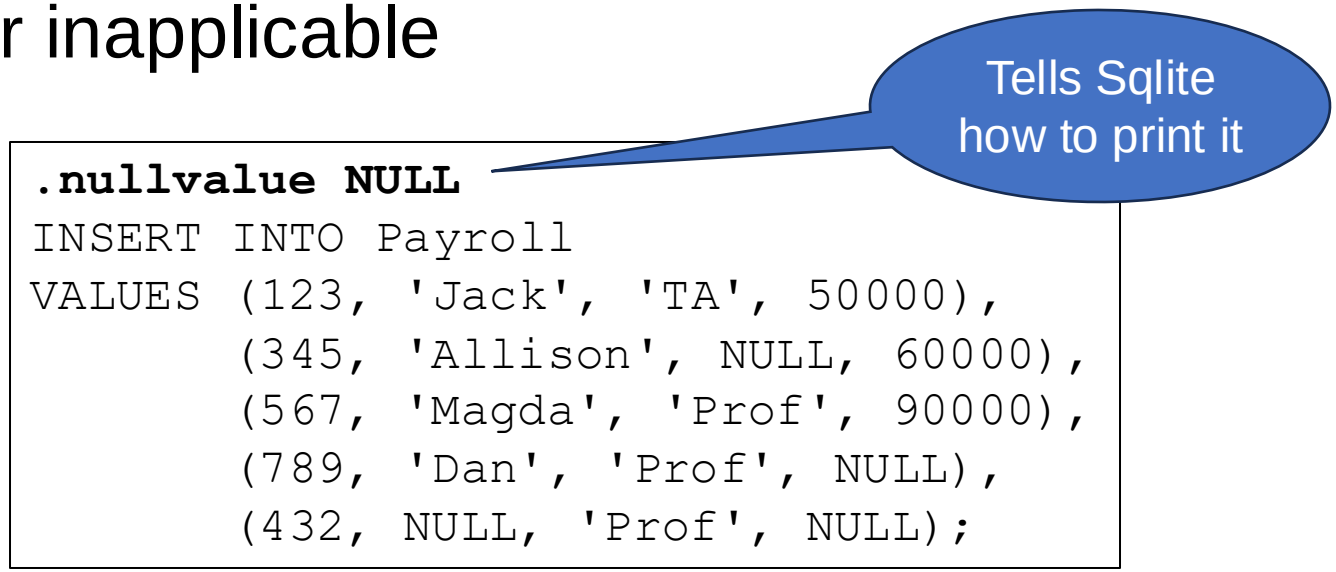
```
.nullvalue NULL
INSERT INTO Payroll
VALUES (123, 'Jack', 'TA', 50000),
       (345, 'Allison', NULL, 60000),
       (567, 'Magda', 'Prof', 90000),
       (789, 'Dan', 'Prof', NULL),
       (432, NULL, 'Prof', NULL);
```

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	NULL	60000
567	Magda	Prof	90000
789	Dan	Prof	NULL
432	NULL	Prof	NULL

NULLs in SQL

A NULL value means missing, or unknown, or undefined, or inapplicable



Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	NULL	60000
567	Magda	Prof	90000
789	Dan	Prof	NULL
432	NULL	Prof	NULL

NULLs in SQL

A NULL value means missing, or unknown, or undefined, or inapplicable

Complications:

- Expressions with NULLs?
- Conditions with NULLs?

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	NULL	60000
567	Magda	Prof	90000
789	Dan	Prof	NULL
432	NULL	Prof	NULL

Expressions with NULLs

If any term is NULL, the entire expression is NULL

Give everyone a 10% raise

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	NULL	60000
567	Magda	Prof	90000
789	Dan	Prof	NULL
432	NULL	Prof	NULL

Expressions with NULLs

If any term is NULL, the entire expression is NULL

Give everyone a 10% raise

```
SELECT Name, Salary*1.1 as NewSalary
FROM Payroll;
```

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	NULL	60000
567	Magda	Prof	90000
789	Dan	Prof	NULL
432	NULL	Prof	NULL

Expressions with NULLs

If any term is NULL, the entire expression is NULL

Give everyone a 10% raise

```
SELECT Name, Salary*1.1 as NewSalary
FROM Payroll;
```

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	NULL	60000
567	Magda	Prof	90000
789	Dan	Prof	NULL
432	NULL	Prof	NULL

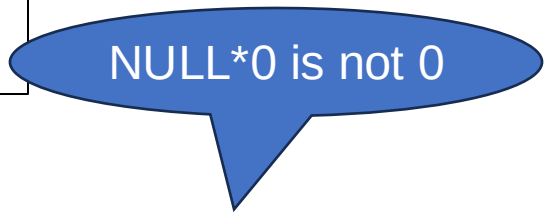
Name	NewSalary
Jack	55000
Allison	66000
Magda	99000
Dan	NULL
NULL	NULL

Expressions with NULLs

If any term is NULL, the entire expression is NULL

Everybody works for free!

```
SELECT Name, Salary*0 as NewSalary
FROM Payroll;
```



Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	NULL	60000
567	Magda	Prof	90000
789	Dan	Prof	NULL
432	NULL	Prof	NULL

Name	NewSalary
Jack	0
Allison	0
Magda	0
Dan	NULL
NULL	NULL

Expressions with NULLs

If any term is NULL, the entire expression is NULL

Everybody works for free!

```
SELECT Name, 0 as NewSalary
FROM Payroll;
```

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	NULL	60000
567	Magda	Prof	90000
789	Dan	Prof	NULL
432	NULL	Prof	NULL

Name	NewSalary
Jack	0
Allison	0
Magda	0
Dan	0
NULL	0

Conditions with NULLs

How should NULLs affect conditions in WHERE?

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	NULL	60000
567	Magda	Prof	90000
789	Dan	Prof	NULL
432	NULL	Prof	NULL

Conditions with NULLs

How should NULLs affect conditions in WHERE?

```
SELECT Name
FROM Payroll
WHERE Job = 'TA';
```

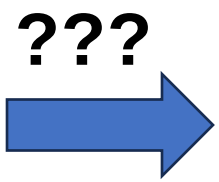
Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	NULL	60000
567	Magda	Prof	90000
789	Dan	Prof	NULL
432	NULL	Prof	NULL

Conditions with NULLs

How should NULLs affect conditions in WHERE?

```
SELECT Name
FROM Payroll
WHERE Job = 'TA';
```



Name	Job
Jack	TA
Allison??	???

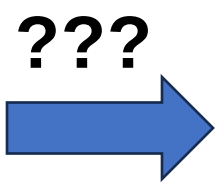
Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	NULL	60000
567	Magda	Prof	90000
789	Dan	Prof	NULL
432	NULL	Prof	NULL

Conditions with NULLs

How should NULLs affect conditions in WHERE?

```
SELECT Name
FROM Payroll
WHERE Job = 'TA';
```



Name	Job
Jack	TA
~~Allison??~~	~~???~~

Not included:
SQL uses
3 valued logic.

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	NULL	60000
567	Magda	Prof	90000
789	Dan	Prof	NULL
432	NULL	Prof	NULL

Three-valued Logic

false = 0; unknown = 0.5; true = 1

Three-valued Logic

false = 0; unknown = 0.5; true = 1

x AND y = min(x,y);
x OR y = max(x,y);
not x = 1-x

Three-valued Logic

false = 0; unknown = 0.5; true = 1

x AND y = min(x,y);
x OR y = max(x,y);
not x = 1-x

What are these conditions?

- true AND unknown = min(1, 0.5) = unknown

Three-valued Logic

false = 0; unknown = 0.5; true = 1

x AND y = min(x,y);
x OR y = max(x,y);
not x = 1-x

What are these conditions?

- true AND unknown = unknown

Three-valued Logic

false = 0; unknown = 0.5; true = 1

x AND y = min(x,y);
x OR y = max(x,y);
not x = 1-x

What are these conditions?

- true AND unknown = unknown
- true OR unknown =

Three-valued Logic

false = 0; unknown = 0.5; true = 1

x AND y = min(x,y);
x OR y = max(x,y);
not x = 1-x

What are these conditions?

- true AND unknown = unknown
- true OR unknown = true

Three-valued Logic

false = 0; unknown = 0.5; true = 1

x AND y = min(x,y);
x OR y = max(x,y);
not x = 1-x

What are these conditions?

- true AND unknown = unknown
- true OR unknown = true
- unknown AND false =

Three-valued Logic

false = 0; unknown = 0.5; true = 1

x AND y = min(x,y);
x OR y = max(x,y);
not x = 1-x

What are these conditions?

- true AND unknown = unknown
- true OR unknown = true
- unknown AND false = false

Three-valued Logic

false = 0; unknown = 0.5; true = 1

x AND y = min(x,y);
x OR y = max(x,y);
not x = 1-x

What are these conditions?

- true AND unknown = unknown
- true OR unknown = true
- unknown AND false = false

A = value
A < value
A > value

Three-valued Logic

false = 0; unknown = 0.5; true = 1

x AND y = min(x,y);
x OR y = max(x,y);
not x = 1-x

What are these conditions?

- true AND unknown = unknown
- true OR unknown = true
- unknown AND false = false

A = value
A < value
A > value

When A is NULL
then unknown

Three-valued Logic

What does this query return?

```
SELECT *
FROM Payroll
WHERE Job != 'Prof'
   or Salary > 80000;
```

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	NULL	60000
567	Magda	Prof	90000
789	Dan	Prof	NULL
432	NULL	Prof	NULL

True

Three-valued Logic

What does this query return?

```
SELECT *
FROM Payroll
WHERE Job != 'Prof'
   or Salary > 80000;
```

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	NULL	60000
567	Magda	Prof	90000
789	Dan	Prof	NULL
432	NULL	Prof	NULL

True

Unknown

Three-valued Logic

What does this query return?

```
SELECT *
FROM Payroll
WHERE Job != 'Prof'
   or Salary > 80000;
```

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	NULL	60000
567	Magda	Prof	90000
789	Dan	Prof	NULL
432	NULL	Prof	NULL

True

Unknown

True

Three-valued Logic

What does this query return?

```
SELECT *
FROM Payroll
WHERE Job != 'Prof'
   or Salary > 80000;
```

Payroll

UserID	Name	Job	Salary	
123	Jack	TA	50000	**True**
345	Allison	NULL	60000	**Unknown**
567	Magda	Prof	90000	**True**
789	Dan	Prof	NULL	**Unknown**
432	NULL	Prof	NULL	**Unknown**

Three-valued Logic

What does this query return?

```
SELECT *
FROM Payroll
WHERE Job != 'Prof'
   or Salary > 80000;
```

UserID	Name	Job	Salary
123	Jack	TA	50000
567	Magda	Prof	90000

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	NULL	60000
567	Magda	Prof	90000
789	Dan	Prof	NULL
432	NULL	Prof	NULL

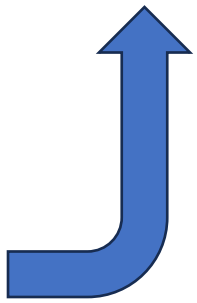
True

Unknown

True

Unknown

Unknown



Three-valued Logic

NULLs are the nightmare of query optimizers

```
SELECT *
FROM Payroll
WHERE Job !='Prof' or Job ='Prof';
```

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	NULL	60000
567	Magda	Prof	90000
789	Dan	Prof	NULL
432	NULL	Prof	NULL

Three-valued Logic

NULLs are the nightmare of query optimizers

```
SELECT *
FROM Payroll
WHERE Job !='Prof' or Job ='Prof';
```

Should return everyone, but…

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	NULL	60000
567	Magda	Prof	90000
789	Dan	Prof	NULL
432	NULL	Prof	NULL

Three-valued Logic

NULLs are the nightmare of query optimizers

```
SELECT *
FROM Payroll
WHERE Job !='Prof' or Job ='Prof';
```

Should return everyone, but…

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	NULL	60000
567	Magda	Prof	90000
789	Dan	Prof	NULL
432	NULL	Prof	NULL

…we are missing Allison!

Three-valued Logic

NULLs are the nightmare of query optimizers

```
SELECT *
FROM Payroll
WHERE Job !='Prof' or Job ='Prof' or Job isNull;
```

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	NULL	60000
567	Magda	Prof	90000
789	Dan	Prof	NULL
432	NULL	Prof	NULL

Now we get everyone!

Discussion

- NULL: convenient way to represent missing values
- Need 3-valued logic
- However, leads to huge complications for the optimizer, and even counterintuitive query behavior
- Better avoid NULLs if possible
- One exception: LEFT OUTER Joins. Let's revisit

Outer Joins Revisited: ON v.s. WHERE

```
SELECT P.Name, R.Car
FROM Payroll AS P
   LEFT OUTER JOIN Regist AS R
   ON P.UserID = R.UserID
      AND R.Car = 'Charger';
```

1. Perform the join with the ON clause
2. Add all missing tuples from LEFT
3. Check the WHERE clause (if any)

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000

Regist

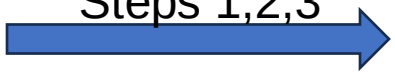
UserID	Car
123	Charger
567	Civic
567	Pinto

Outer Joins Revisited: ON v.s. WHERE

```
SELECT P.Name, R.Car
FROM Payroll AS P
    LEFT OUTER JOIN Regist AS R
    ON P.UserID = R.UserID
        AND R.Car = 'Charger';
```

1. Perform the join with the ON clause
2. Add all missing tuples from LEFT
3. Check the WHERE clause (if any)

Steps 1,2,3



Name	Car
Jack	Charger
Allison	NULL
Magda	NULL

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000

Regist

UserID	Car
123	Charger
567	Civic
567	Pinto

Outer Joins Revisited: ON v.s. WHERE

1. Perform the join with the ON clause
2. Add all missing tuples from LEFT
3. Check the WHERE clause (if any)

```
SELECT P.Name, R.Car
FROM Payroll AS P
    LEFT OUTER JOIN Regist AS R
    ON P.UserID = R.UserID
       AND R.Car = 'Charger';
```

Steps 1,2,3

Name	Car
Jack	Charger
Allison	NULL
Magda	NULL

```
SELECT P.Name, R.Car
FROM Payroll AS P
    LEFT OUTER JOIN Regist AS R
    ON P.UserID = R.UserID
WHERE R.Car = 'Charger';
```

What differs if we place R.Car='Charger' in the WHERE clause?

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000

Regist

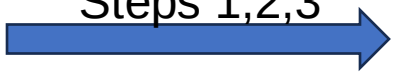
UserID	Car
123	Charger
567	Civic
567	Pinto

Outer Joins Revisited: ON v.s. WHERE

```
SELECT P.Name, R.Car
FROM Payroll AS P
    LEFT OUTER JOIN Regist AS R
    ON P.UserID = R.UserID
       AND R.Car = 'Charger';
```

1. Perform the join with the ON clause
2. Add all missing tuples from LEFT
3. Check the WHERE clause (if any)

Steps 1,2,3



Name	Car
Jack	Charger
Allison	NULL
Magda	NULL

```
SELECT P.Name, R.Car
FROM Payroll AS P
    LEFT OUTER JOIN Regist AS R
    ON P.UserID = R.UserID
WHERE R.Car = 'Charger';
```

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000

Regist

UserID	Car
123	Charger
567	Civic
567	Pinto

Outer Joins Revisited: ON v.s. WHERE

1. Perform the join with the ON clause
2. Add all missing tuples from LEFT
3. Check the WHERE clause (if any)

```
SELECT P.Name, R.Car
FROM Payroll AS P
    LEFT OUTER JOIN Regist AS R
    ON P.UserID = R.UserID
       AND R.Car = 'Charger';
```

Steps 1,2,3

Name	Car
Jack	Charger
Allison	NULL
Magda	NULL

```
SELECT P.Name, R.Car
FROM Payroll AS P
    LEFT OUTER JOIN Regist AS R
    ON P.UserID = R.UserID
WHERE R.Car = 'Charger';
```

Steps 1,2

Name	Car
Jack	Charger
Allison	NULL
Magda	Civic
Magda	Pinto

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000

Regist

UserID	Car
123	Charger
567	Civic
567	Pinto

Outer Joins Revisited: ON v.s. WHERE

1. Perform the join with the ON clause
2. Add all missing tuples from LEFT
3. Check the WHERE clause (if any)

```
SELECT P.Name, R.Car
FROM Payroll AS P
    LEFT OUTER JOIN Regist AS R
    ON P.UserID = R.UserID
       AND R.Car = 'Charger';
```

Steps 1,2,3

Name	Car
Jack	Charger
Allison	NULL
Magda	NULL

```
SELECT P.Name, R.Car
FROM Payroll AS P
    LEFT OUTER JOIN Regist AS R
    ON P.UserID = R.UserID
WHERE R.Car = 'Charger';
```

Steps 1,2

Name	Car
Jack	Charger
Allison	NULL
Magda	Civic
Magda	Pinto

Step 3

Name	Car
Jack	Charger

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000

Regist

UserID	Car
123	Charger
567	Civic
567	Pinto

Outer Joins Revisited: ON v.s. WHERE

```
SELECT P.Name, R.Car
FROM Payroll AS P
    LEFT OUTER JOIN Regist AS R
    ON P.UserID = R.UserID
       AND R.Car = 'Charger';
```

1. Perform the join with the ON clause
2. Add all missing tuples from LEFT
3. Check the WHERE clause (if any)

Steps 1,2,3

Name	Car
Jack	Charger
Allison	NULL
Magda	NULL

```
SELECT P.Name, R.Car
FROM Payroll AS P
    LEFT OUTER JOIN Regist AS R
    ON P.UserID = R.UserID
WHERE R.Car = 'Charger';
```

Steps 1,2

Name	Car
Jack	Charger
Allison	NULL
Magda	Civic
Magda	Pinto

Step 3

Name	Car
Jack	Charger

ON, WHERE differ

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000

Regist

UserID	Car
123	Charger
567	Civic
567	Pinto

Outer Joins

- Now understand OUTER JOINs really well!

Aggregates

Aggregates

- Aggregate: many values to one value

Aggregates

- Aggregate: many values to one value

- Aggregates in SQL:
  - sum(1, 4, 3, 4) = 1+4+3+4 = 12

Aggregates

- Aggregate: many values to one value

- Aggregates in SQL:
  - sum(1, 4, 3, 4) = 1+4+3+4 = 12
  - max(1, 4, 3, 4) = 4
  - min(1, 4, 3, 4) = 1

Aggregates

- Aggregate: many values to one value

- Aggregates in SQL:
  - sum(1, 4, 3, 4) = 1+4+3+4 = 12
  - max(1, 4, 3, 4) = 4
  - min(1, 4, 3, 4) = 1
  - count(1, 4, 3, 4) = 4

Aggregates

- Aggregate: many values to one value

- Aggregates in SQL:
  - sum(1, 4, 3, 4) = 1+4+3+4 = 12
  - max(1, 4, 3, 4) = 4
  - min(1, 4, 3, 4) = 1
  - count(1, 4, 3, 4) = 4
  - avg(1, 4, 3, 4) = 3

Aggregates

- Aggregate: many values to one value

- Aggregates in SQL:
  - sum(1, 4, 3, 4) = 1+4+3+4 = 12
  - max(1, 4, 3, 4) = 4
  - min(1, 4, 3, 4) = 1
  - count(1, **4**, 3, **4**) = 4
  - avg(1, **4**, 3, **4**) = 3

COUNT

How many records are in Payroll?

```
SELECT count(*) as C
FROM Payroll;
```

C
4

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

Regist

UserID	Car
123	Charger
567	Civic
567	Pinto

COUNT

How many records are in Payroll?

```
SELECT count(*)
FROM Payroll;
```

count(*)
4

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

Regist

UserID	Car
123	Charger
567	Civic
567	Pinto

COUNT

How many records are in Payroll?

```
SELECT count(*)
FROM Payroll;
```

count(*)
4

How many cars are in the database?

```
SELECT count(*)
FROM Regist;
```

…
3

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

Regist

UserID	Car
123	Charger
567	Civic
567	Pinto

COUNT

How many records are in Payroll?

```
SELECT count(*)
FROM Payroll;
```

count(*)
4

How many cars are in the database?

```
SELECT count(*)
FROM Regist;
```

…
3

How many TA's are there?

```
SELECT count(*)
FROM Payroll
WHERE Job='TA';
```

…
2

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

Regist

UserID	Car
123	Charger
567	Civic
567	Pinto

COUNT

How many records are in Payroll?

```
SELECT count(*)
FROM Payroll;
```

count(*)
4

How many cars are in the database?

```
SELECT count(*)
FROM Regist;
```

…
3

How many TA's are there?

```
SELECT count(*)
FROM Payroll
WHERE Job='TA';
```

…
2

How many people have salary > 55000?

```
SELECT count(*)
FROM Payroll
WHERE Salary>55000;
```

…
3

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

Regist

UserID	Car
123	Charger
567	Civic
567	Pinto

SUM, MIN, MAX, AVG

What is the sum of all salaries?

```
SELECT sum(Salary)
FROM Payroll;
```

sum(…)
300000

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

Regist

UserID	Car
123	Charger
567	Civic
567	Pinto

SUM, MIN, MAX, AVG

What is the sum of all salaries?

```
SELECT sum(Salary)
FROM Payroll;
```

sum(…)
300000

What is the average salary?

```
SELECT avg(Salary)
FROM Payroll;
```

avg(…)
75000

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

Regist

UserID	Car
123	Charger
567	Civic
567	Pinto

SUM, MIN, MAX, AVG

What is the sum of all salaries?

```
SELECT sum(Salary)
FROM Payroll;
```

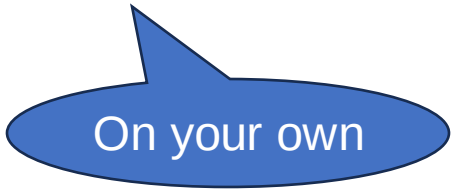
sum(…)
300000

What is the average salary?

```
SELECT avg(Salary)
FROM Payroll;
```

avg(…)
75000

What is the smallest salary?
What is the largest salary?



Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

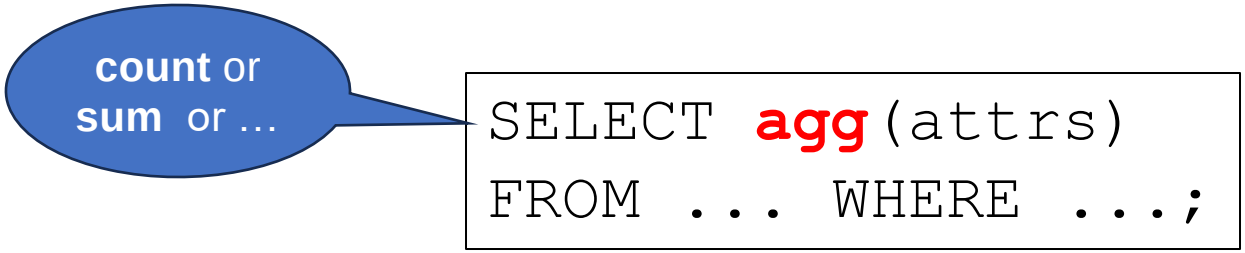
Regist

UserID	Car
123	Charger
567	Civic
567	Pinto

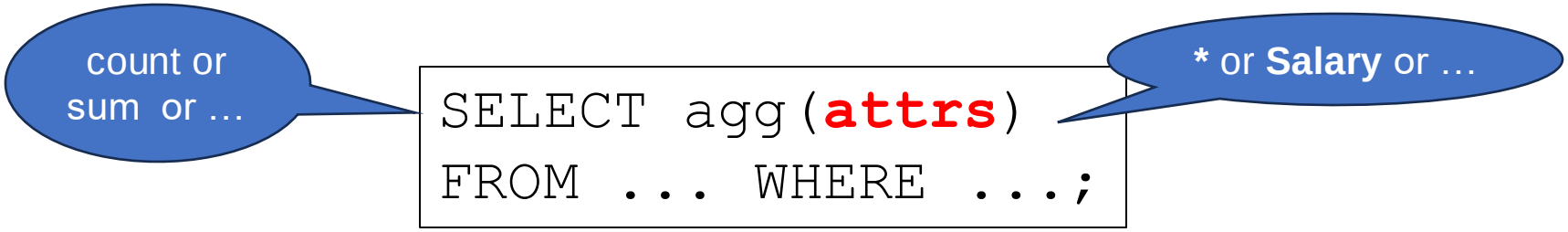
Semantics

```
SELECT agg(attrs)
FROM ... WHERE ...;
```

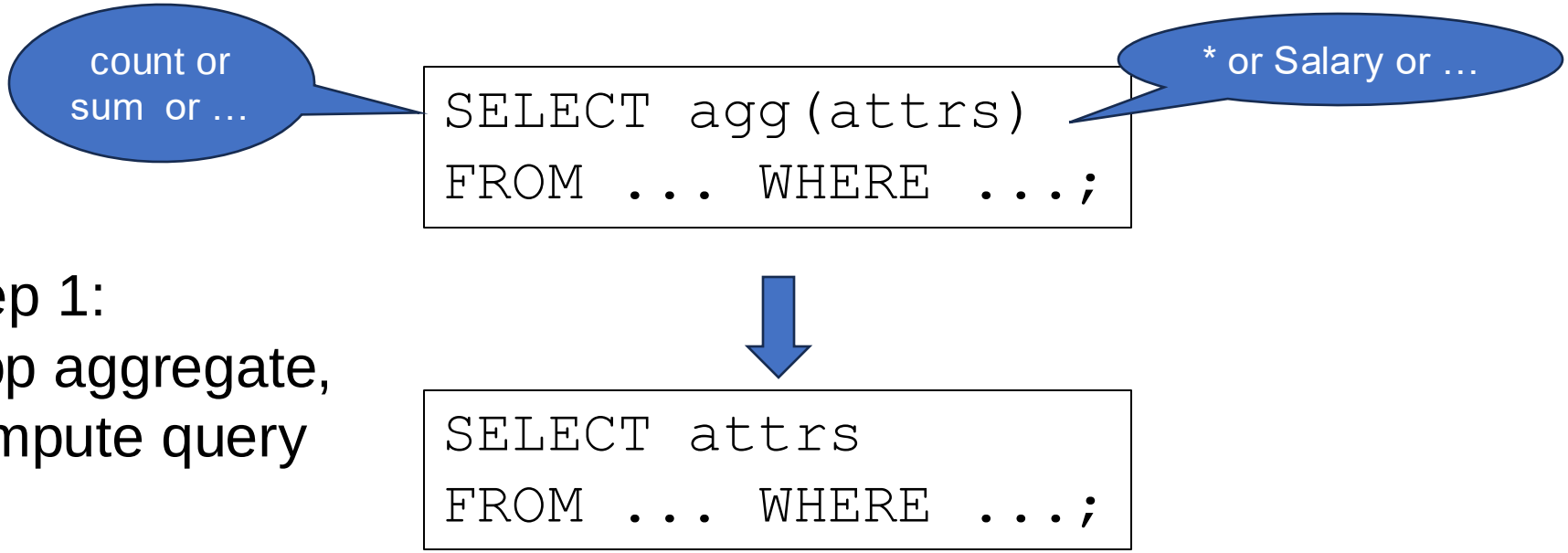
Semantics



Semantics



Semantics


count or
sum or …
SELECT agg(attrs)
FROM ... WHERE ...;
* or Salary or …
Step 1:
drop aggregate,
compute query
SELECT attrs
FROM ... WHERE ...;

Semantics

count or sum or …

```
SELECT agg(attrs)
FROM ... WHERE ...;
```

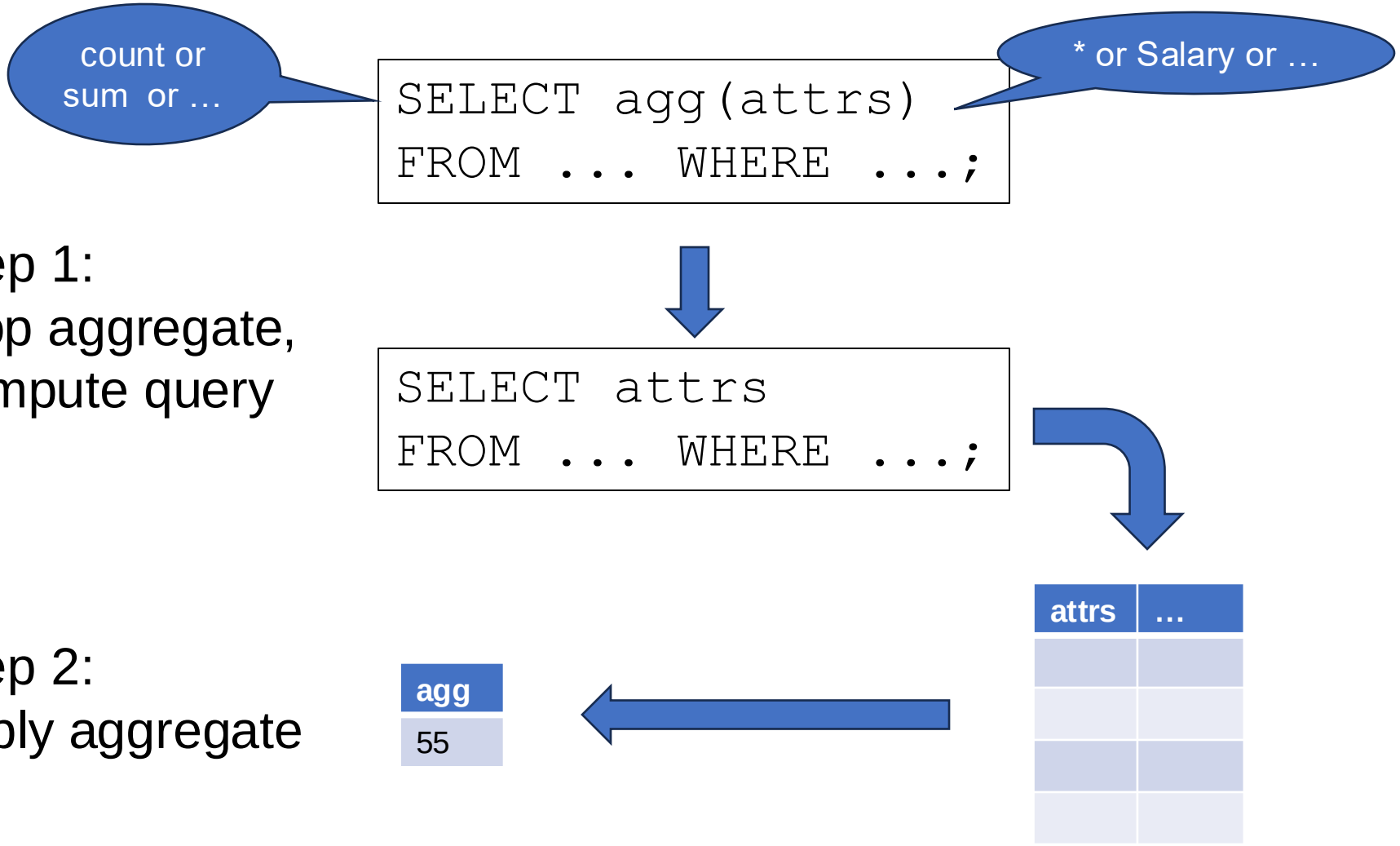
* or Salary or …

Step 1:
drop aggregate,
compute query

```
SELECT attrs
FROM ... WHERE ...;
```

attrs	…

Semantics


count or
sum or …
SELECT agg(attrs)
FROM ... WHERE ...;
* or Salary or …
Step 1:
drop aggregate,
compute query
SELECT attrs
FROM ... WHERE ...;
attrs
…
Step 2:
apply aggregate
agg
55

COUNT

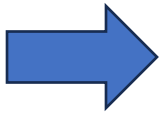
```
SELECT count(*)
FROM Payroll;
```

Payroll

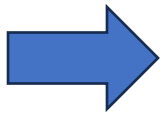
UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

COUNT

```
SELECT count(*)
FROM Payroll;
```



```
SELECT *
FROM Payroll;
```



...

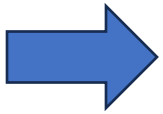
4

Payroll

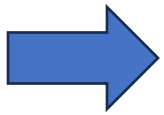
UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

COUNT

```
SELECT count(*)
FROM Payroll;
```



```
SELECT *
FROM Payroll;
```



...

4

How many job are there in this institution?

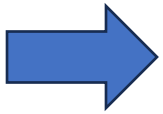
```
SELECT count(Job)
FROM Payroll;
```

Payroll

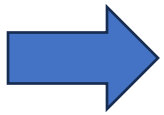
UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

COUNT

```
SELECT count(*)
FROM Payroll;
```



```
SELECT *
FROM Payroll;
```



...

4

How many job are there in this institution?

```
SELECT count(Job)
FROM Payroll;
```

```
SELECT Job
FROM Payroll;
```

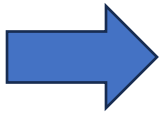
?

Payroll

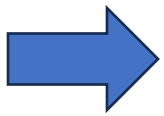
UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

COUNT

```
SELECT count(*)
FROM Payroll;
```



```
SELECT *
FROM Payroll;
```



...
4

How many job are there in this institution?

```
SELECT count(Job)
FROM Payroll;
```

```
SELECT Job
FROM Payroll;
```

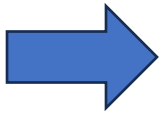
Job
TA
TA
Prof
Prof

Payroll

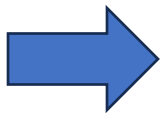
UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

COUNT

```
SELECT count(*)
FROM Payroll;
```



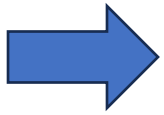
```
SELECT *
FROM Payroll;
```



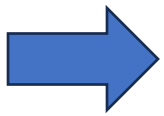
...
4

How many job are there in this institution?

```
SELECT count(Job)
FROM Payroll;
```



Job
TA
TA
Prof
Prof



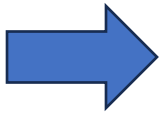
...
4

Payroll

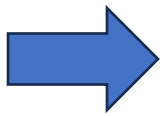
UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

COUNT

```
SELECT count(*)
FROM Payroll;
```



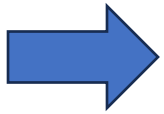
```
SELECT *
FROM Payroll;
```



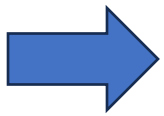
...
4

How many job are there in this institution?

```
SELECT count(Job)
FROM Payroll;
```



Job
TA
TA
Prof
Prof



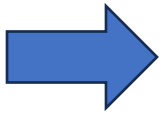
...
4

Payroll

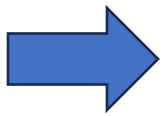
UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

COUNT

```
SELECT count(*)
FROM Payroll;
```



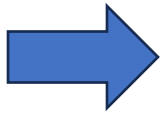
```
SELECT *
FROM Payroll;
```



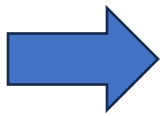
...
4

How many job are there in this institution?

```
SELECT count(Job)
FROM Payroll;
```



Job
TA
TA
Prof
Prof



...
4

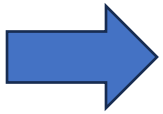
```
SELECT count(DISTINCT Job)
FROM Payroll;
```

Payroll

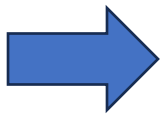
UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

COUNT

```
SELECT count(*)
FROM Payroll;
```



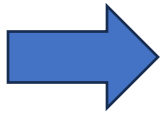
```
SELECT *
FROM Payroll;
```



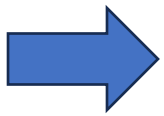
...
4

How many job are there in this institution?

```
SELECT count(Job)
FROM Payroll;
```



Job
TA
TA
Prof
Prof



...
4

```
SELECT count(DISTINCT Job)
FROM Payroll;
```

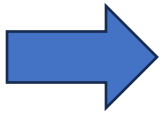
Job
TA
Prof

Payroll

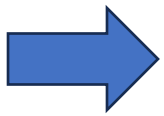
UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

COUNT

```
SELECT count(*)
FROM Payroll;
```



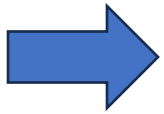
```
SELECT *
FROM Payroll;
```



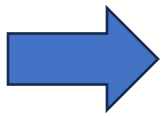
...
4

How many job are there in this institution?

```
SELECT count(Job)
FROM Payroll;
```



Job
TA
TA
Prof
Prof



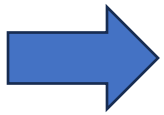
...
4

```
SELECT count(DISTINCT Job)
FROM Payroll;
```

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

Job
TA
Prof



...
2

Correct

Aggregates and NULLs

Aggregates ignore NULLs:

- Sum: same as 0
- Avg: NOT the same as 0
- Min/max: same as $+\infty, -\infty$
- Count: doesn't include them, but it's more subtle

Aggregates and NULLs

```
SELECT sum(Salary)
FROM Payroll;
```

sum(…)
200000

50000 + 60000 + 90000

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	NULL	60000
567	Magda	Prof	90000
789	Dan	Prof	NULL
432	NULL	Prof	NULL

Aggregates and NULLs

```
SELECT sum(Salary)
FROM Payroll;
```

sum(…)
200000

50000 + 60000 + 90000

```
SELECT avg(Salary)
FROM Payroll;
```

avg(…)
66667

NULLs are just ignored.
Just as you expected.

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	NULL	60000
567	Magda	Prof	90000
789	Dan	Prof	NULL
432	NULL	Prof	NULL

Discussion: Aggregates

- Semantics: two steps

- NULLs are ignored

Aggregates and Joins

Aggregates and Joins

- Joins combine records from multiple tables
- Aggregates: many values to one value
- Together they form a very powerful SQL tool

Aggregates and Joins

Find the average salary of people driving a Pinto

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

Regist

UserID	Car
123	Charger
123	Pinto
567	Civic
567	Pinto

Aggregates and Joins

Find the average salary of people driving a Pinto

```
SELECT avg(P.Salary)
FROM Payroll P, Regist R
WHERE P.UserID = R.UserID
  and R.Car = 'Pinto';
```

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

Regist

UserID	Car
123	Charger
123	Pinto
567	Civic
567	Pinto

Aggregates and Joins

Find the average salary of people driving a Pinto

```
SELECT avg(P.Salary)
FROM Payroll P, Regist R
WHERE P.UserID = R.UserID
  and R.Car = 'Pinto';
```

```
SELECT P.Salary
FROM Payroll P, Regist R
...;
```

Name	Salary
Jack	50000
Magda	90000

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

Regist

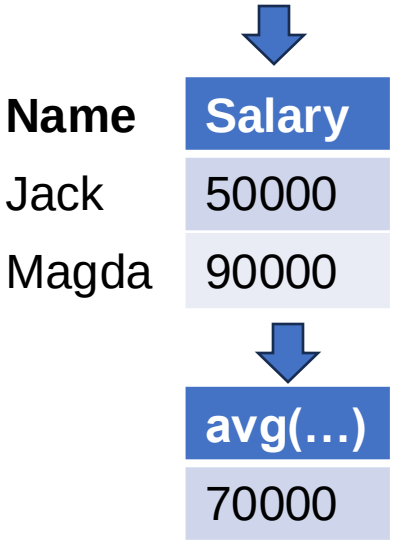
UserID	Car
123	Charger
123	Pinto
567	Civic
567	Pinto

Aggregates and Joins

Find the average salary of people driving a Pinto

```
SELECT avg(P.Salary)
FROM Payroll P, Regist R
WHERE P.UserID = R.UserID
  and R.Car = 'Pinto';
```

```
SELECT P.Salary
FROM Payroll P, Regist R
...;
```



Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

Regist

UserID	Car
123	Charger
123	Pinto
567	Civic
567	Pinto

Duplicates

- Need to watch for duplicates introduced when we join two tables
- Sometimes duplicates are easy to deal with, e.g. COUNT(DISTINCT …)
- Sometimes they are much harder to deal with, and we will discuss this in future lectures

Duplicates

How many people drive a car?

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

Regist

UserID	Car
123	Charger
567	Civic
567	Pinto

Duplicates

How many people drive a car?

```
SELECT count(*)
FROM Payroll P, Regist R
WHERE P.UserID = R.UserID;
```

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

Regist

UserID	Car
123	Charger
567	Civic
567	Pinto

Duplicates

How many people drive a car?

```
SELECT count(*)
FROM Payroll P, Regist R
WHERE P.UserID = R.UserID;
```

```
SELECT *
FROM ...;
```

UserID	Name	Job	Salary	UserID	Car
123	Jack	TA	50000	123	Charger
567	Magda	Prof	90000	567	Civic
567	Magda	Prof	90000	567	Pinto

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

Regist

UserID	Car
123	Charger
567	Civic
567	Pinto

Duplicates

How many people drive a car?

```
SELECT count(*)
FROM Payroll P, Regist R
WHERE P.UserID = R.UserID;
```

```
SELECT *
FROM ...;
```

UserID	Name	Job	Salary	UserID	Car
123	Jack	TA	50000	123	Charger
567	Magda	Prof	90000	567	Civic
567	Magda	Prof	90000	567	Pinto

count(*)
3

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

Regist

UserID	Car
123	Charger
567	Civic
567	Pinto

Duplicates

How many people drive a car?

```
SELECT count(DISTINCT UserID)
FROM Payroll P, Regist R
WHERE P.UserID = R.UserID;
```

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

Regist

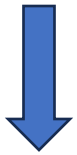
UserID	Car
123	Charger
567	Civic
567	Pinto

Duplicates

How many people drive a car?

```
SELECT count(DISTINCT UserID)
FROM Payroll P, Regist R
WHERE P.UserID = R.UserID;
```

```
SELECT DISTINCT UserID
FROM ...;
```



UserID
123
567

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

Regist

UserID	Car
123	Charger
567	Civic
567	Pinto

Duplicates

How many people drive a car?

```
SELECT count(DISTINCT UserID)
FROM Payroll P, Regist R
WHERE P.UserID = R.UserID;
```

```
SELECT DISTINCT UserID
FROM ...;
```

UserID
123
567

count(*)
2

Right!

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

Regist

UserID	Car
123	Charger
567	Civic
567	Pinto

Duplicates

What is the average salary of car drivers?

```
SELECT avg(P.Salary)
FROM Payroll P, Regist R
WHERE P.UserID = R.UserID;
```

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

Regist

UserID	Car
123	Charger
567	Civic
567	Pinto

Duplicates

What is the average salary of car drivers?

```
SELECT avg(P.Salary)
FROM Payroll P, Regist R
WHERE P.UserID = R.UserID;
```

```
SELECT P.Salary
FROM ...;
```

Name	Salary
Jack	50000
Magda	90000
Magda	90000

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

Regist

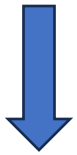
UserID	Car
123	Charger
567	Civic
567	Pinto

Duplicates

What is the average salary of car drivers?

```
SELECT avg(P.Salary)
FROM Payroll P, Regist R
WHERE P.UserID = R.UserID;
```

```
SELECT P.Salary
FROM ...;
```



Name	Salary
Jack	50000
Magda	90000
Magda	90000

avg(...)
76667

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

Regist

UserID	Car
123	Charger
567	Civic
567	Pinto

Duplicates

What is the average salary of car drivers?

```
SELECT avg(P.Salary)
FROM Payroll P, Regist R
WHERE P.UserID = R.UserID;
```

```
SELECT P.Salary
FROM ...;
```

Name	Salary
Jack	50000
Magda	90000
Magda	90000

Duplicate!

avg(…)
76667

Wrong!

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

Regist

UserID	Car
123	Charger
567	Civic
567	Pinto

Duplicates

What is the average salary of car drivers?

```
SELECT avg(DISTINCT P.Salary)
FROM Payroll P, Regist R
WHERE P.UserID = R.UserID;
```

Does DISTINCT fix it?

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

Regist

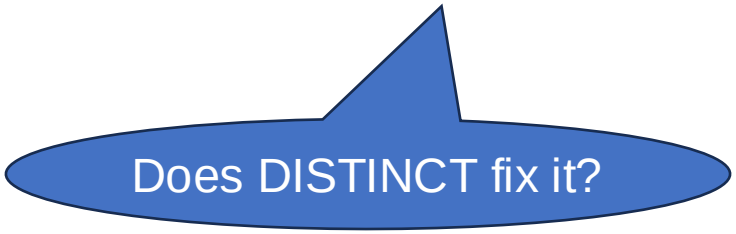
UserID	Car
123	Charger
567	Civic
567	Pinto

Duplicates

What is the average salary of car drivers?

```
SELECT avg(DISTINCT P.Salary)
FROM Payroll P, Regist R
WHERE P.UserID = R.UserID;
```

```
SELECT DISTINCT P.Salary
FROM ...;
```



Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	50000
567	Magda	Prof	90000
789	Dan	Prof	100000

Regist

UserID	Car
123	Charger
345	Tesla
567	Civic
567	Pinto

Duplicates

What is the average salary of car drivers?

```
SELECT avg(DISTINCT P.Salary)
FROM Payroll P, Regist R
WHERE P.UserID = R.UserID;
```

```
SELECT DISTINCT P.Salary
FROM ...;
```

Salary
50000
90000

avg(...)
70000

Wrong!

Does DISTINCT fix it?

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	50000
567	Magda	Prof	90000
789	Dan	Prof	100000

Regist

UserID	Car
123	Charger
345	Tesla
567	Civic
567	Pinto

Duplicates

What is the average salary of car drivers?

```
SELECT avg(DISTINCT P.Salary)
FROM Payroll P, Regi
WHERE P.UserID = R.U
```

```
SELECT DISTINCT P.Salary
FROM ...;
```

This query is harder to fix. We will discuss it on Friday

Does DISTINCT fix it?

avg(…)
70000

Wrong!

Salary
50000
90000

Correct answer: 63333

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	50000
567	Magda	Prof	90000
789	Dan	Prof	100000

Regist

UserID	Car
123	Charger
345	Tesla
567	Civic
567	Pinto

Summary for Today

- NULLs:
  - Once a NULL, always a NULL
  - 3-Valued Logic (3VL)
  - Outer-joins revisited

- Aggregates
  - sum, min, max, count, avg
  - Two steps semantics
  - Subtle interactions with joins, duplicates, nulls