

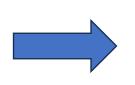
CSE 344: Intro to Data Management Outer Joins, NULL

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

Outer Joins

For each employee, find the cars that they drive

```
SELECT p.name, r.car
FROM payroll AS p
    JOIN regist AS r
    ON p.user_id = r.user_id;
```



| name | car |
|-------|---------|
| Jack | Charger |
| Magda | Civic |
| Magda | Pinto |

payroll

| user_id | name | job | salary |
|---------|---------|------|--------|
| 123 | Jack | TA | 50000 |
| 345 | Allison | TA | 60000 |
| 567 | Magda | Prof | 90000 |
| 789 | Dan | Prof | 100000 |

| user_id | car |
|---------|---------|
| 123 | Charger |
| 567 | Civic |
| 567 | Pinto |

For each employee, find the cars that they drive

SELECT p.name, r.car
FROM payroll AS p
 JOIN regist AS r
 ON p.user_id = r.user_id;



| name | car |
|-------|---------|
| Jack | Charger |
| Magda | Civic |
| Magda | Pinto |

payroll

| user_id | name | job | salary |
|---------|---------|------|--------|
| 123 | Jack | TA | 50000 |
| 345 | Allison | TA | 60000 |
| 567 | Magda | Prof | 90000 |
| 789 | Dan | Prof | 100000 |

| user_id | car |
|---------|---------|
| 123 | Charger |
| 567 | Civic |
| 567 | Pinto |

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.user_id = r.user_id;
```

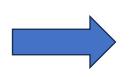
payroll

| user_id | name | job | salary |
|---------|---------|------|--------|
| 123 | Jack | TA | 50000 |
| 345 | Allison | TA | 60000 |
| 567 | Magda | Prof | 90000 |
| 789 | Dan | Prof | 100000 |

| user_id | car |
|---------|---------|
| 123 | Charger |
| 567 | Civic |
| 567 | Pinto |

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.user_id = r.user_id;



| name | car |
|---------|---------|
| Jack | Charger |
| Magda | Civic |
| Magda | Pinto |
| Allison | NULL |
| Dan | NULL |

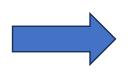
payroll

| user_id | name | job | salary |
|---------|---------|------|--------|
| 123 | Jack | TA | 50000 |
| 345 | Allison | TA | 60000 |
| 567 | Magda | Prof | 90000 |
| 789 | Dan | Prof | 100000 |

| user_id | car |
|---------|---------|
| 123 | Charger |
| 567 | Civic |
| 567 | Pinto |

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.user_id = r.user_id;



| r |
|---|
| |
| |
| |
| |
| |
| |

payroll

| user_id | name | job | salary |
|---------|---------|------|--------|
| 123 | Jack | TA | 50000 |
| 345 | Allison | TA | 60000 |
| 567 | Magda | Prof | 90000 |
| 789 | Dan | Prof | 100000 |

regist

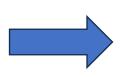
| user_id | car |
|---------|---------|
| 123 | Charger |
| 567 | Civic |
| 567 | Pinto |

NULL means "unknown" or "missing"

January 13, 2025

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.user_id = r.user_id;
```



| name | car |
|---------|---------|
| Jack | Charger |
| Magda | Civic |
| Magda | Pinto |
| Allison | NULL |
| Dan | NULL |

Left outer join:

- Perform the join <u>with the ON clause</u>
- 2. Add all missing tuples from LEFT
- 3. Check the <u>WHERE clause</u> (if present) payroll

| user_id | name | job | salary |
|---------|---------|------|--------|
| 123 | Jack | TA | 50000 |
| 345 | Allison | TA | 60000 |
| 567 | Magda | Prof | 90000 |
| 789 | Dan | Prof | 100000 |

| _ | |
|---------|---------|
| user_id | car |
| 123 | Charger |
| 567 | Civic |
| 567 | Pinto |

For each employee, find the cars that they drive

| SELECT p.name, r.car | |
|--------------------------------------|---|
| FROM payroll AS p | |
| LEFT OUTER JOIN regist AS | r |
| <pre>ON p.user_id = r.user_id;</pre> | |
| | |

| <i></i> | |
|---------|---------|
| name | car |
| Jack | Charger |
| Magda | Civic |
| Magda | Pinto |
| Allison | NULL |
| Dan | NULL |

Left outer join:

- 1. Perform the join with the ON clause
- 2. Add all missing tuples from LEFT
- 3. Check the <u>WHERE clause</u> (if present) payroll

| user_id | name | job | salary |
|---------|---------|------|--------|
| 123 | Jack | TA | 50000 |
| 345 | Allison | TA | 60000 |
| 567 | Magda | Prof | 90000 |
| 789 | Dan | Prof | 100000 |

ON, WHERE differ (next lecture)

| user_id | car |
|---------|---------|
| 123 | Charger |
| 567 | Civic |
| 567 | Pinto |

Outer Joins

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

NULLs

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

payroll

| user_id | name | job | salary |
|---------|---------|------|--------|
| 123 | Jack | TA | 50000 |
| 345 | Allison | NULL | 60000 |
| 567 | Magda | Prof | 90000 |
| 789 | Dan | Prof | NULL |
| 432 | NULL | Prof | NULL |

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

payroll

| user_id | name | job | salary |
|---------|---------|------|--------|
| 123 | Jack | TA | 50000 |
| 345 | Allison | NULL | 60000 |
| 567 | Magda | Prof | 90000 |
| 789 | Dan | Prof | NULL |
| 432 | NULL | Prof | NULL |

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

Tells Sqlite how to print it

.nullvalue NULL

| user_id | name | job | salary |
|---------|---------|------|--------|
| 123 | Jack | TA | 50000 |
| 345 | Allison | NULL | 60000 |
| 567 | Magda | Prof | 90000 |
| 789 | Dan | Prof | NULL |
| 432 | NULL | Prof | NULL |

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

Complications:

- Expressions with NULLs?
- Conditions with NULLs?

| user_id | name | job | salary |
|---------|---------|------|--------|
| 123 | Jack | TA | 50000 |
| 345 | Allison | NULL | 60000 |
| 567 | Magda | Prof | 90000 |
| 789 | Dan | Prof | NULL |
| 432 | NULL | Prof | NULL |

If any term is NULL, the entire expression is NULL

Give everyone a 10% raise

| user_id | name | job | salary |
|---------|---------|------|--------|
| 123 | Jack | TA | 50000 |
| 345 | Allison | NULL | 60000 |
| 567 | Magda | Prof | 90000 |
| 789 | Dan | Prof | NULL |
| 432 | NULL | Prof | NULL |

If any term is NULL, the entire expression is NULL

Give everyone a 10% raise

```
SELECT name, salary*1.1 AS new_salary
FROM payroll;
```

| user_id | name | job | salary |
|---------|---------|------|--------|
| 123 | Jack | TA | 50000 |
| 345 | Allison | NULL | 60000 |
| 567 | Magda | Prof | 90000 |
| 789 | Dan | Prof | NULL |
| 432 | NULL | Prof | NULL |

If any term is NULL, the entire expression is NULL

Give everyone a 10% raise

```
SELECT name, salary*1.1 AS new_sal
FROM payroll;
```

payroll

| user_id | name | job | salary |
|---------|---------|------|--------|
| 123 | Jack | TA | 50000 |
| 345 | Allison | NULL | 60000 |
| 567 | Magda | Prof | 90000 |
| 789 | Dan | Prof | NULL |
| 432 | NULL | Prof | NULL |



| name | new_sal |
|---------|---------|
| Jack | 55000 |
| Allison | 66000 |
| Magda | 99000 |
| Dan | NULL |
| NULL | NULL |

18

If any term is NULL, the entire expression is NULL

```
SELECT name, salary*0 AS new_sal
FROM payroll;
```

payroll

| user_id | name | job | salary |
|---------|---------|------|--------|
| 123 | Jack | TA | 50000 |
| 345 | Allison | NULL | 60000 |
| 567 | Magda | Prof | 90000 |
| 789 | Dan | Prof | NULL |
| 432 | NULL | Prof | NULL |



| name | new_sal |
|---------|---------|
| Jack | 0 |
| Allison | 0 |
| Magda | 0 |
| Dan | NULL |
| NULL | NULL |

If any term is NULL, the entire expression is NULL

```
SELECT name, salary*0 AS new_sal
FROM payroll;
```

NULL*0 is not 0

payroll

| user_id | name | job | salary |
|---------|---------|------|--------|
| 123 | Jack | TA | 50000 |
| 345 | Allison | NULL | 60000 |
| 567 | Magda | Prof | 90000 |
| 789 | Dan | Prof | NULL |
| 432 | NULL | Prof | NULL |



| name | new_sal |
|---------|---------|
| Jack | 0 |
| Allison | 0 |
| Magda | 0 |
| Dan | NULL |
| NULL | NULL |

20

If any term is NULL, the entire expression is NULL

SELECT name, 0 AS new_sal
FROM payroll;

payroll

| user_id | name | job | salary |
|---------|---------|------|--------|
| 123 | Jack | TA | 50000 |
| 345 | Allison | NULL | 60000 |
| 567 | Magda | Prof | 90000 |
| 789 | Dan | Prof | NULL |
| 432 | NULL | Prof | NULL |





| name | new_sal |
|---------|---------|
| Jack | 0 |
| Allison | 0 |
| Magda | 0 |
| Dan | 0 |
| NULL | 0 |

21

How should NULLs affect conditions in WHERE?

payroll

| user_id | name | job | salary |
|---------|---------|------|--------|
| 123 | Jack | TA | 50000 |
| 345 | Allison | NULL | 60000 |
| 567 | Magda | Prof | 90000 |
| 789 | Dan | Prof | NULL |
| 432 | NULL | Prof | NULL |

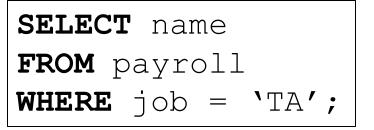
How should NULLs affect conditions in WHERE?

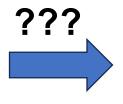
```
SELECT name
FROM payroll
WHERE job = 'TA';
```

payroll

| user_id | name | job | salary |
|---------|---------|------|--------|
| 123 | Jack | TA | 50000 |
| 345 | Allison | NULL | 60000 |
| 567 | Magda | Prof | 90000 |
| 789 | Dan | Prof | NULL |
| 432 | NULL | Prof | NULL |

How should NULLs affect conditions in WHERE?



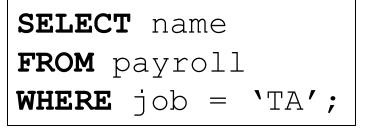


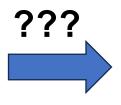
| name | job |
|-----------|-----|
| Jack | TA |
| Allison?? | ??? |

24

| user_id | name | job | salary |
|---------|---------|------|--------|
| 123 | Jack | TA | 50000 |
| 345 | Allison | NULL | 60000 |
| 567 | Magda | Prof | 90000 |
| 789 | Dan | Prof | NULL |
| 432 | NULL | Prof | NULL |

How should NULLs affect conditions in WHERE?





| name | job | |
|--|-----|--|
| Jack | TA | |
| Allison?? | ??? | |
| Not included: SQL uses 3 valued logic. | | |

25

payroll

| user_id | name | job | salary |
|---------|---------|------|--------|
| 123 | Jack | TA | 50000 |
| 345 | Allison | NULL | 60000 |
| 567 | Magda | Prof | 90000 |
| 789 | Dan | Prof | NULL |
| 432 | NULL | Prof | NULL |

false =
$$0$$
; unknown = 0.5 ; true = 1

```
false = 0; unknown = 0.5; true = 1

x \text{ AND } y = \min(x,y);

x \text{ OR } y = \max(x,y);

x \text{ not } x = 1-x
```

```
false = 0; unknown = 0.5; true = 1

x \text{ AND } y = \min(x,y);

x \text{ OR } y = \max(x,y);

x \text{ not } x = 1-x
```

What are these conditions?

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

```
false = 0; unknown = 0.5; true = 1

x \text{ AND } y = \min(x,y);

x \text{ OR } y = \max(x,y);

x \text{ not } x = 1-x
```

What are these conditions?

true AND unknown = unknown

```
false = 0; unknown = 0.5; true = 1

x \text{ AND } y = \min(x,y);

x \text{ OR } y = \max(x,y);

x \text{ not } x = 1-x
```

What are these conditions?

- true AND unknown = unknown
- true OR unknown =

```
false = 0; unknown = 0.5; true = 1

x \text{ AND } y = \min(x,y);

x \text{ OR } y = \max(x,y);

x \text{ not } x = 1-x
```

What are these conditions?

true AND unknown = unknown

■ true OR unknown = true

```
false = 0; unknown = 0.5; true = 1

x \text{ 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 = 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

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

false =
$$0$$
; unknown = 0.5 ; true = 1

$$x ext{ AND } y = min(x,y);$$

 $x ext{ OR } y = max(x,y);$
 $not ext{ } x = 1-x$

What are these conditions?

true AND unknown = unknown

true OR unknown = true

unknown AND false = false

A = value

A < value

A > value

35

When A is NULL then unknown

What does this query return?

```
SELECT *
FROM payroll
WHERE job != 'Prof'
OR salary > 80000;
```

payroll

| user_id | name | job | salary |
|---------|---------|------|--------|
| 123 | Jack | TA | 50000 |
| 345 | Allison | NULL | 60000 |
| 567 | Magda | Prof | 90000 |
| 789 | Dan | Prof | NULL |
| 432 | NULL | Prof | NULL |

True

36

What does this query return?

```
SELECT *
FROM payroll
WHERE job != 'Prof'
OR salary > 80000;
```

payroll

| user_id | 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

What does this query return?

```
SELECT *
FROM payroll
WHERE job != 'Prof'
OR salary > 80000;
```

payroll

| user_id | 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

38

True

What does this query return?

```
SELECT *
FROM payroll
WHERE job != 'Prof'
OR salary > 80000;
```

payroll

| user_id | 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 |

January 13, 2025 Aggregates

What does this query return?

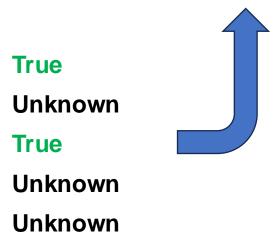
```
SELECT *
FROM payroll
WHERE job != 'Prof'
OR salary > 80000;
```

| user_id | name | job | salary |
|---------|-------|------|--------|
| 123 | Jack | TA | 50000 |
| 567 | Magda | Prof | 90000 |

40

payroll

| user_id | name | job | salary |
|---------|---------|------|--------|
| 123 | Jack | TA | 50000 |
| 345 | Allison | NULL | 60000 |
| 567 | Magda | Prof | 90000 |
| 789 | Dan | Prof | NULL |
| 432 | NULL | Prof | NULL |

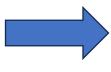


NULLs are the nightmare of query optimizers

```
SELECT *
FROM payroll
WHERE job != 'Prof' OR job = 'Prof';
```

payroll

| user_id | name | job | salary |
|---------|---------|------|--------|
| 123 | Jack | TA | 50000 |
| 345 | Allison | NULL | 60000 |
| 567 | Magda | Prof | 90000 |
| 789 | Dan | Prof | NULL |
| 432 | NULL | Prof | NULL |



NULLs are the nightmare of query optimizers

```
SELECT *
FROM payroll
WHERE job != 'Prof' OR job = 'Prof';
```

payroll

| user_id | name | job | salary |
|---------|---------|------|--------|
| 123 | Jack | TA | 50000 |
| 345 | Allison | NULL | 60000 |
| 567 | Magda | Prof | 90000 |
| 789 | Dan | Prof | NULL |
| 432 | NULL | Prof | NULL |

Should return everyone, but...



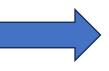
NULLs are the nightmare of query optimizers

```
SELECT *
FROM payroll
WHERE job != 'Prof' OR job = 'Prof';
```

payroll

| user_id | name | job | salary |
|---------|---------|------|--------|
| 123 | Jack | TA | 50000 |
| 345 | Allison | NULL | 60000 |
| 567 | Magda | Prof | 90000 |
| 789 | Dan | Prof | NULL |
| 432 | NULL | Prof | NULL |

Should return everyone, but...



...we are missing Allison!

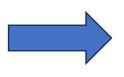
43

NULLs are the nightmare of query optimizers

```
SELECT *
FROM payroll
WHERE job != 'Prof' OR job = 'Prof' OR job IS NULL;
```

payroll

| user_id | 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
- NULL as "unknown" vs NULL as "known to be absent"
- 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

```
SELECT p.name, r.car
FROM payroll AS P
    LEFT OUTER JOIN regist AS R
    ON p.user_id = r.user_id
    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

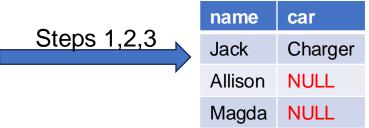
| user_id | name | job | salary |
|---------|---------|------|--------|
| 123 | Jack | TA | 50000 |
| 345 | Allison | TA | 60000 |
| 567 | Magda | Prof | 90000 |

regist

| user_id | car |
|------------------|---------|
| <mark>123</mark> | Charger |
| 567 | Civic |
| 567 | Pinto |

```
SELECT p.name, r.car
FROM payroll AS P
   LEFT OUTER JOIN regist AS R
   ON p.user_id = r.user_id
    AND r.car = 'Charger';
```

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



payroll

| user_id | name | job | salary |
|---------|---------|------|--------|
| 123 | Jack | TA | 50000 |
| 345 | Allison | TA | 60000 |
| 567 | Magda | Prof | 90000 |

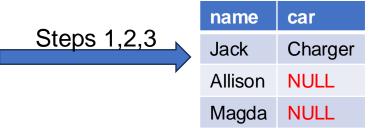
regist

| user_id | car |
|---------|---------|
| 123 | Charger |
| 567 | Civic |
| 567 | Pinto |

```
SELECT p.name, r.car
FROM payroll AS P
    LEFT OUTER JOIN regist AS R
    ON p.user_id = r.user_id
        AND r.car = 'Charger';
```

```
SELECT p.name, r.car
FROM payroll AS P
    LEFT OUTER JOIN regist AS R
    ON p.user_id = r.user_id
WHERE 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)



What differs if we place r.car='Charger' in the WHERE clause?

payroll

| user_id | name | job | salary |
|------------------|---------|------|--------|
| <mark>123</mark> | Jack | TA | 50000 |
| 345 | Allison | TA | 60000 |
| 567 | Magda | Prof | 90000 |

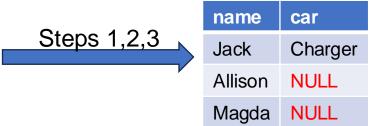
regist

| user_id | car |
|------------------|---------|
| <mark>123</mark> | Charger |
| <mark>567</mark> | Civic |
| <mark>567</mark> | Pinto |

```
SELECT p.name, r.car
FROM payroll AS P
    LEFT OUTER JOIN regist AS R
    ON p.user_id = r.user_id
    AND r.car = 'Charger';
```

```
SELECT p.name, r.car
FROM payroll AS P
    LEFT OUTER JOIN regist AS R
    ON p.user_id = r.user_id
WHERE 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

| user_id | name | job | salary |
|------------------|---------|------|--------|
| <mark>123</mark> | Jack | TA | 50000 |
| 345 | Allison | TA | 60000 |
| 567 | Magda | Prof | 90000 |

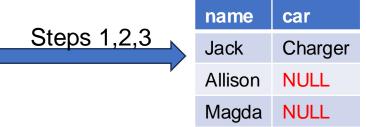
regist

| user_id | car |
|------------------|---------|
| 123 | Charger |
| <mark>567</mark> | Civic |
| <mark>567</mark> | Pinto |

```
SELECT p.name, r.car
FROM payroll AS P
    LEFT OUTER JOIN regist AS R
    ON p.user_id = r.user_id
    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)



| SELECT p.name, r.car | | | |
|-------------------------------------|--|--|--|
| FROM payroll AS P | | | |
| LEFT OUTER JOIN regist AS R | | | |
| <pre>ON p.user_id = r.user_id</pre> | | | |
| <pre>WHERE r.car = 'Charger';</pre> | | | |

| | name | car |
|-----------|---------|---------|
| Steps 1,2 | Jack | Charger |
| | Allison | NULL |
| | Magda | Civic |
| | Magda | Pinto |

payroll

| user_id | name | job | salary |
|------------------|---------|------|--------|
| <mark>123</mark> | Jack | TA | 50000 |
| 345 | Allison | TA | 60000 |
| 567 | Magda | Prof | 90000 |

regist

| user_id | car |
|------------------|---------|
| <mark>123</mark> | Charger |
| <mark>567</mark> | Civic |
| <mark>567</mark> | Pinto |

```
SELECT p.name, r.car
FROM payroll AS P
    LEFT OUTER JOIN regist AS R
    ON p.user_id = r.user_id
    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)



| SELECT p.name, r.car | | | |
|-------------------------------------|--|--|--|
| FROM payroll AS P | | | |
| LEFT OUTER JOIN regist AS R | | | |
| <pre>ON p.user_id = r.user_id</pre> | | | |
| <pre>WHERE r.car = 'Charger';</pre> | | | |

| | name | car | | | |
|-----------|---------|---------|------|------|---------|
| Steps 1,2 | Jack | Charger | 04 | 0 | |
| Steps 1,2 | Allison | NULL | Step | 3 | |
| | Magda | Civic | | name | car |
| | Magda | Pinto | Í | Jack | Charger |

payroll

| user_id | name | job | salary |
|----------------|---------|------|--------|
| 123 | Jack | TA | 50000 |
| 345 | Allison | TA | 60000 |
| 567 | Magda | Prof | 90000 |

regist

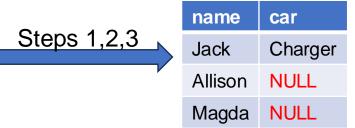
| user_id | car |
|------------------|---------|
| <mark>123</mark> | Charger |
| <mark>567</mark> | Civic |
| <mark>567</mark> | Pinto |

- SELECT p.name, r.car
 FROM payroll AS P
 LEFT OUTER JOIN regist AS R
 ON p.user_id = r.user_id
 AND r.car = 'Charger';
- 1. Perform the join with the ON clause
- Add all missing tuples from LEFT
- 3. Check the WHERE clause (if any)

car

Charger

52



| SELECT p.name, r.car | | | |
|------------------------------------|--|--|--|
| FROM payroll AS P | | | |
| LEFT OUTER JOIN regist AS R | | | |
| ON p.user_id = r.user_id | | | |
| WHERE r.car = 'Charger'; | | | |

| | name | car | | |
|-----------|---------|---------|--------|------|
| Steps 1,2 | Jack | Charger | Step 3 | |
| | Allison | NULL | | |
| | Magda | Civic | | name |
| | Magda | Pinto | ŕ | Jack |

payroll

123

345

user_id

job salary TA 50000

60000

ON, WHERE differ

567 Magda Prof 90000

TA

name

Jack

Allison

regist

| user_id | car |
|------------------|---------|
| <mark>123</mark> | Charger |
| <mark>567</mark> | Civic |
| <mark>567</mark> | Pinto |