

[544] Spark: Grouping and Joining

Tyler Caraza-Harter

Outline

Views and Tables

Grouping

Joining

Tables and Views

orig.parquet

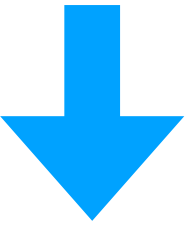
X	Y
A	1
B	2
A	3
C	4

```
df = spark.read.format("parquet").load("orig.parquet").where("X = 'A'")
```

```
df.write.saveAsTable("mytable")
```

```
df.createTempView("myview")
```

mytable
(parquet files in HDFS)



X	Y
A	1
A	3



X	Y
description of how to get data on demand	

myview
(a query with a name)

mytable vs. myview

- which one is faster to create?
- which one takes less space?
- which one is faster if we sum up the Y column?

Demos...

Outline

Views and Tables

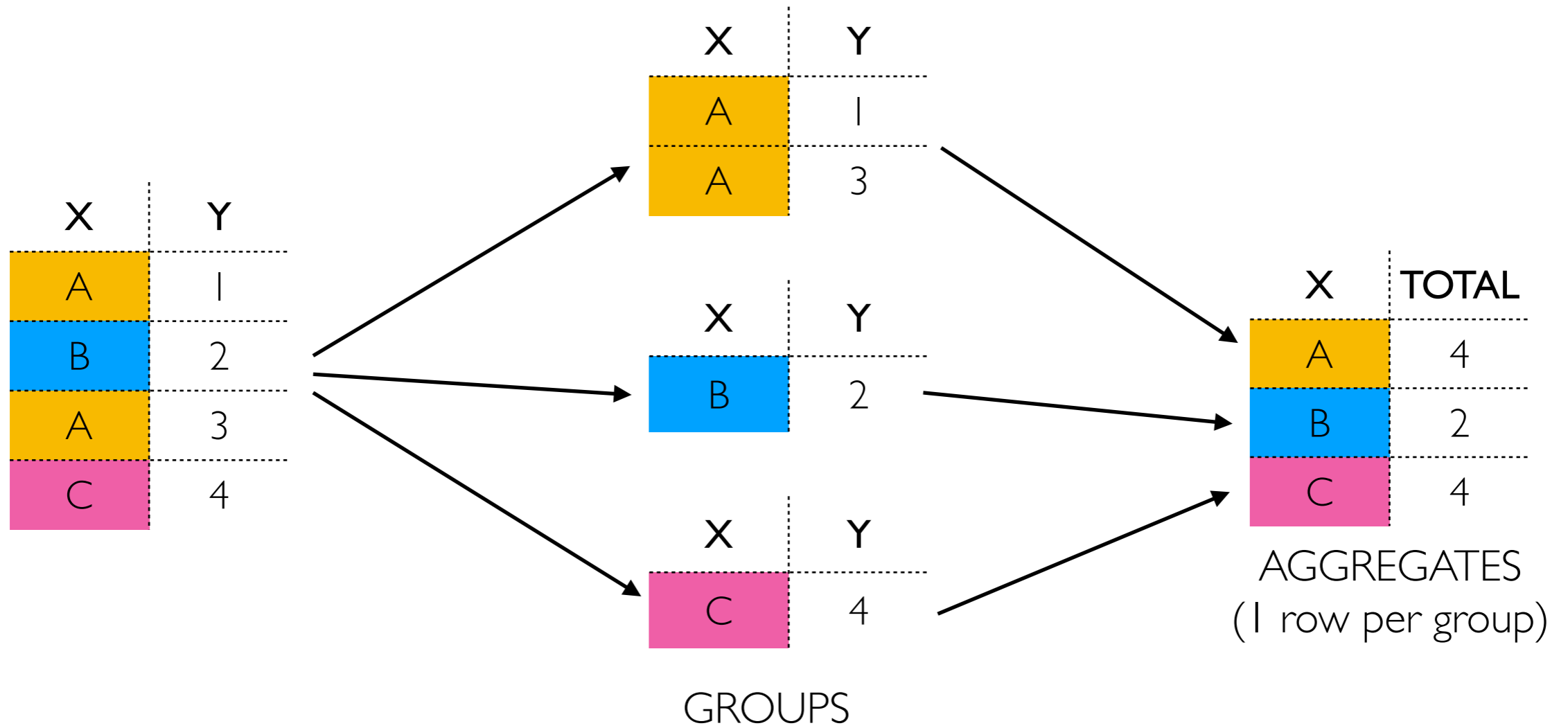
Grouping

Joining

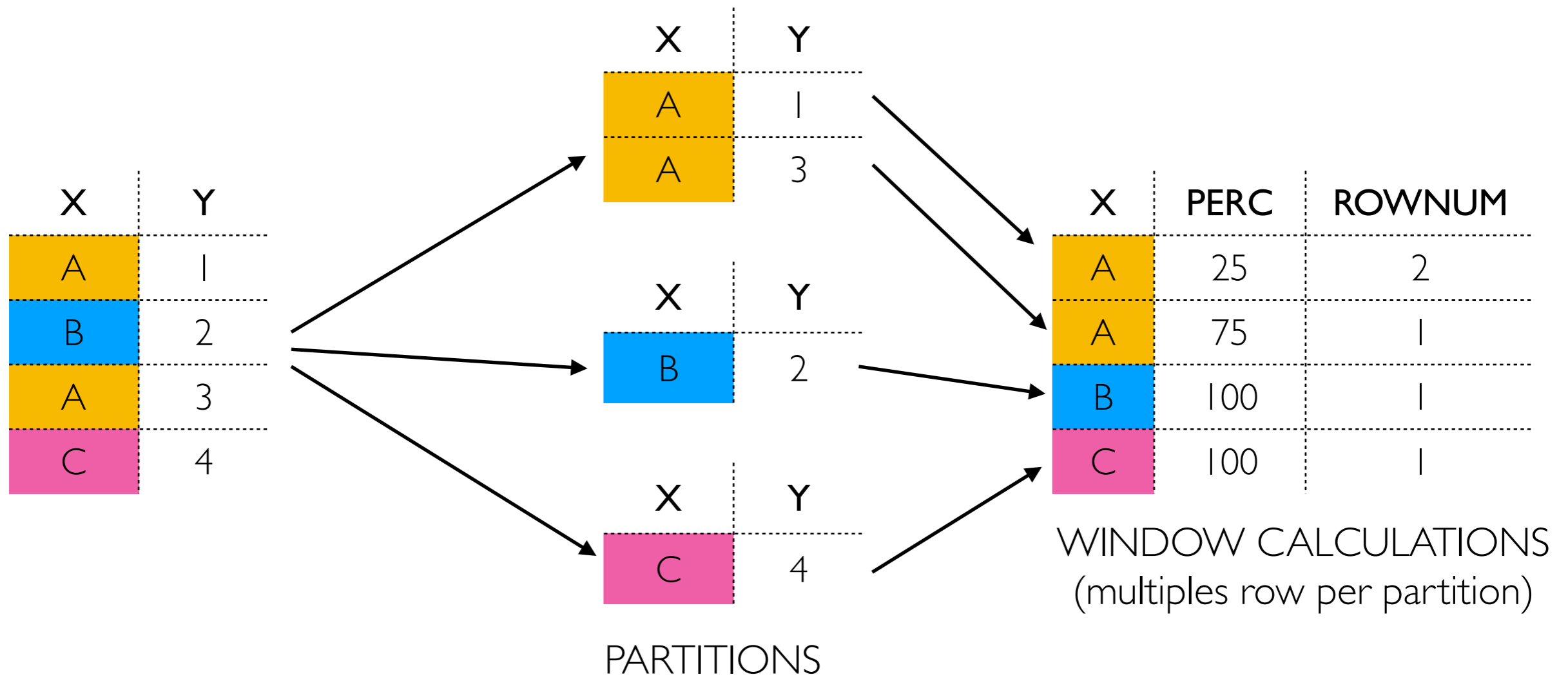
DISTINCT



GROUPS, AGGREGATES



PARTITIONS, WINDOW FUNCTIONS



Nested/chained grouping

X	Y
A	1
B	3
A	2
B	3

→

X	Y	Row Count
A	1	1
B	3	2
A	2	1

→

X	Y Categories	Row Count
A	2	2
B	1	2

Multiple grouping levels

- SQL uses nested queries
- DataFrames can chain multiple groupbys together

TopHat

Demos...

Outline

Views and Tables

Grouping

Joining

Joining

which bands did each guest at the festival see?

`INNER JOIN` on `visits.day = performances.day`

equi join



visits

guest_id	day
A	Tue
A	Mon
B	Tue
B	Wed
C	Wed

performances

band_id	day
X	Mon
X	Tue
Y	Tue

many-to-many relationship:

we join on day

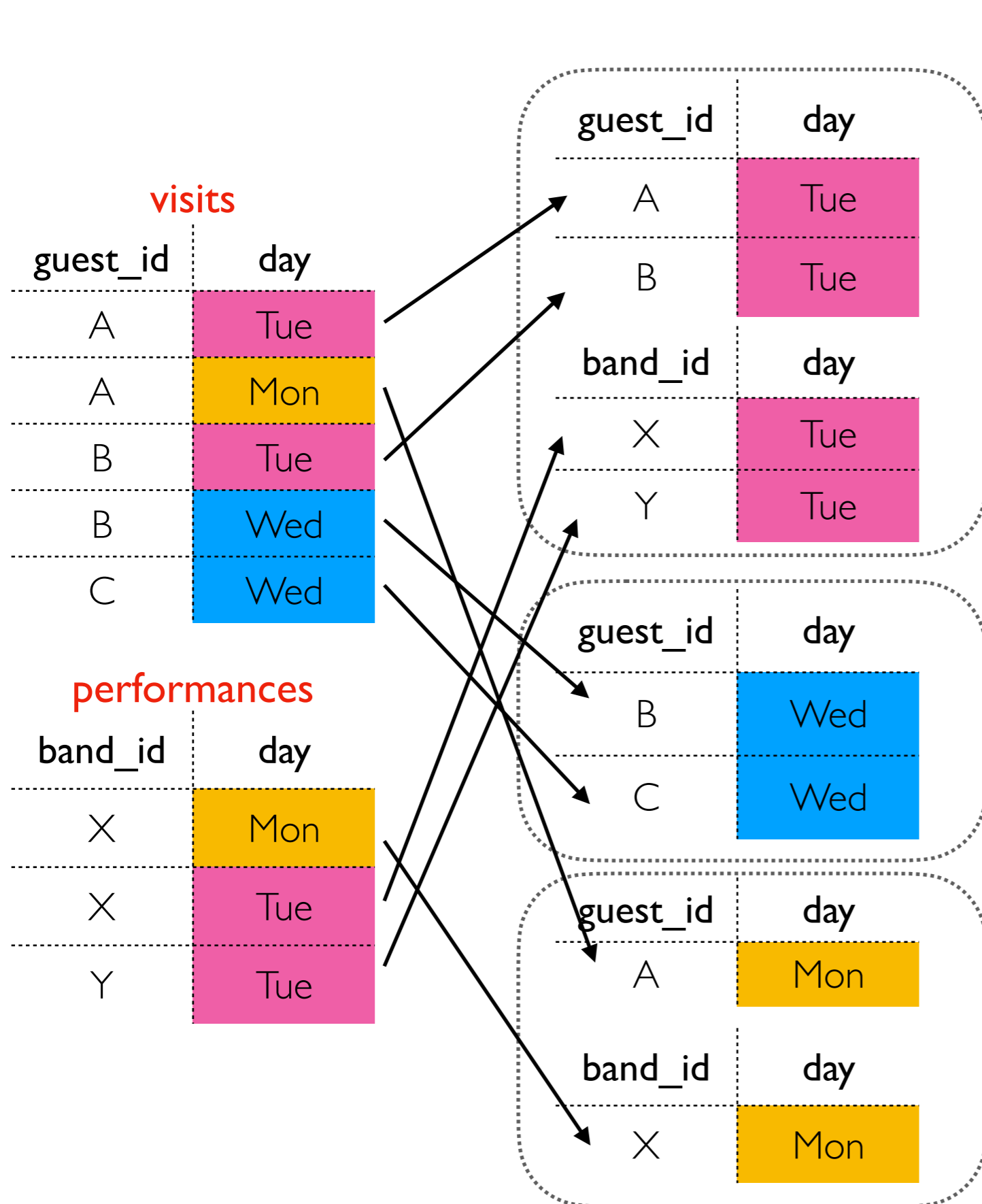
each day has many visits

each day has many performances

Joining

which bands did each guest at the festival see?

INNER JOIN on visits.day = performances.day



equi join

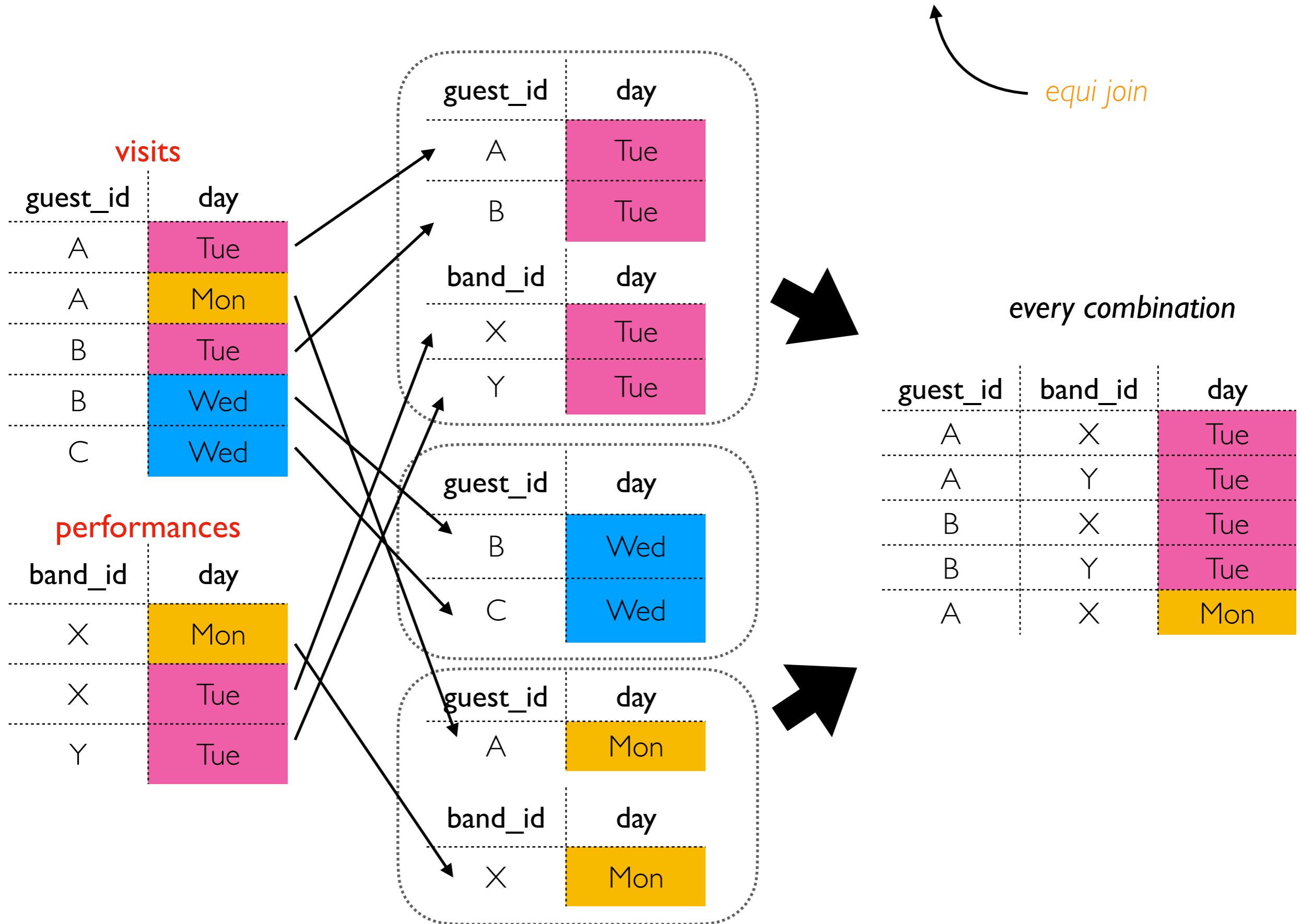
Joining is logically similar to grouping, but on two tables.

To find matches, we need to bring portions of each table with the same day together to the same place.

Joining

which bands did each guest at the festival see?

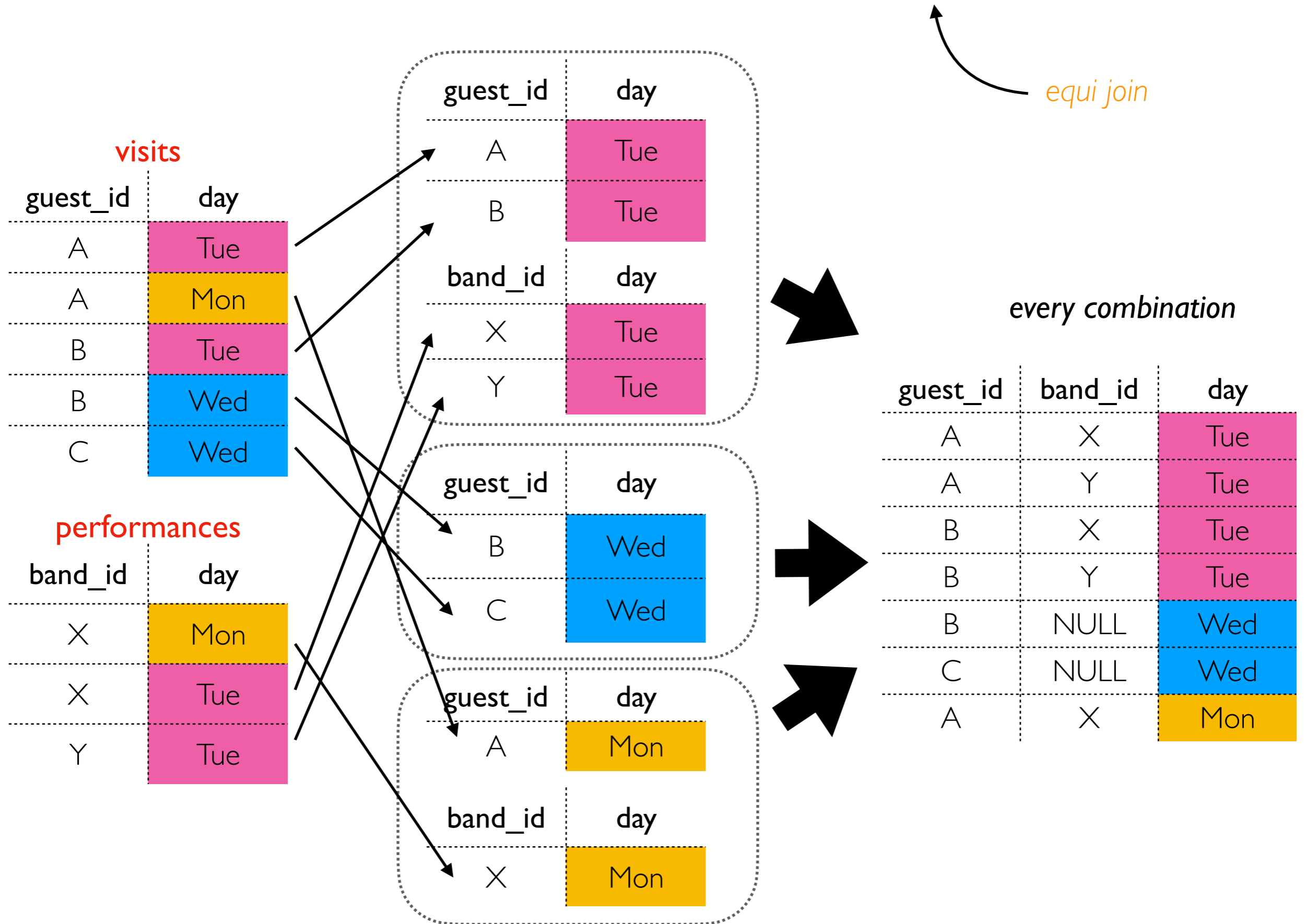
INNER JOIN on visits.day = performances.day



Joining

which guests never saw a performance?

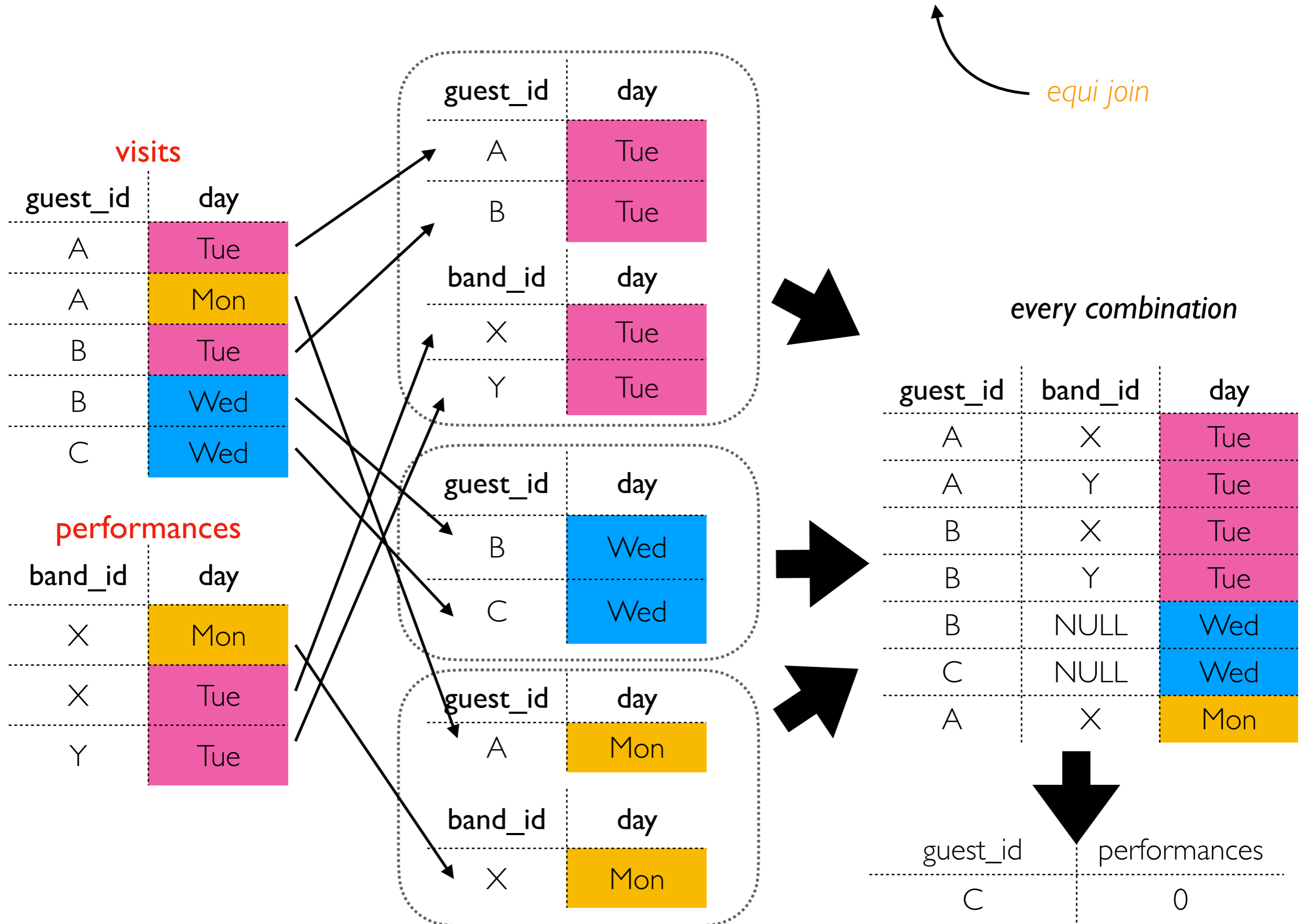
LEFT JOIN on `visits.day = performances.day`



Joining

which guests never saw a performance?

LEFT JOIN on `visits.day = performances.day`



Demos...