CS5226 Lecture 3
Query Tuning II

Query Tuning II

- Reordering groupby & select
- Reordering groupby & join

Reordering group by & select

- Customer (cust#, cname, country)
- Order (order#, cust#, date, totalprice)

```
select cust#, date, total
from   (select cust#, date, sum(totalprice) as total
        from   Order
        group by cust#, date)
where  date >= '2013-01-01'
```

```
select cust#, date, sum(totalprice) as total
from   Order
where  date >= '2013-01-01'
group by cust#, date
```

Reordering group by & select

- Let *p* denote the predicate "date $\geq$ 2013-01-01"

Order

order#	cust#	date	totalprice
200	2	2012-12-05	3000
203	2	2012-12-05	6000
180	3	2012-11-08	4000
500	3	2013-02-01	1000
400	4	2013-01-13	2000
610	4	2013-01-13	5000
701	4	2013-01-20	3000

$\sigma_{\mathbf{p}}(\mathbf{Order})$

order#	cust#	date	totalprice
500	3	2013-02-01	1000
400	4	2013-01-13	2000
610	4	2013-01-13	5000
701	4	2013-01-20	3000

$\mathcal{G}_{\{\mathbf{cust\#},\mathbf{date}\},\mathbf{sum}(\mathbf{totalprice})}\ (\mathbf{Order})$

cust#	date	total
2	2012-12-05	9000
3	2012-11-08	4000
3	2013-02-01	1000
4	2013-01-13	7000
4	2013-01-20	3000

$\sigma_{\mathbf{p}}(\mathcal{G}_{\{\mathbf{cust\#},\mathbf{date}\},\mathbf{sum}(\mathbf{totalprice})}\ (\mathbf{Order}))$

cust#	date	total
3	2013-02-01	1000
4	2013-01-13	7000
4	2013-01-20	3000

$\mathcal{G}_{\{\mathbf{cust\#},\mathbf{date}\},\mathbf{sum}(\mathbf{totalprice})}\ (\sigma_{\mathbf{p}}(\mathbf{Order}))$

cust#	date	total
3	2013-02-01	1000
4	2013-01-13	7000
4	2013-01-20	3000

Reordering group by & select

$$\sigma_p(\mathcal{G}_{A,F}\ (R)) \ = \ \mathcal{G}_{A,F}\ (\sigma_p(R))$$

iff

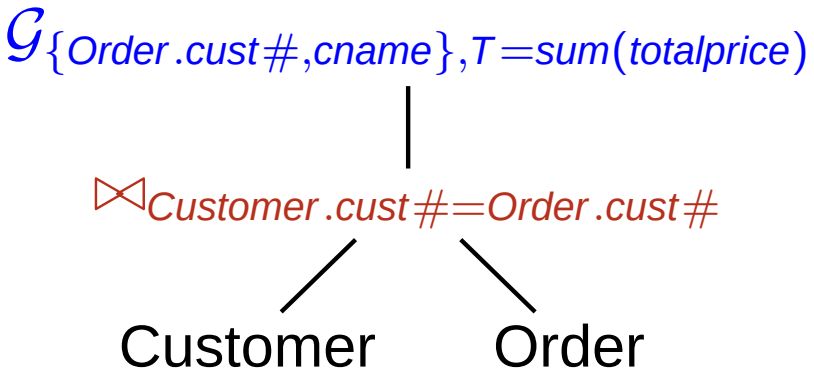
$$A \rightarrow \ columns(p)$$

Pushing group by below join

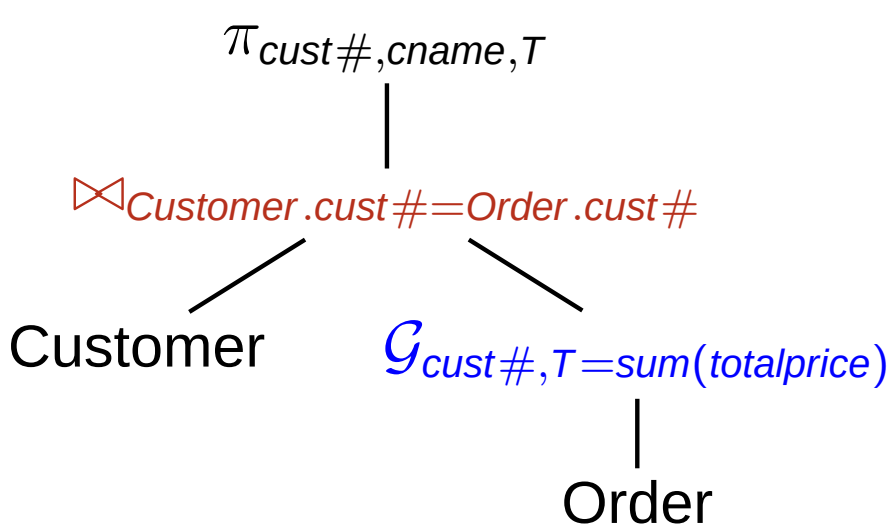
```
select   c.cust#, cname, sum(o.totalprice) as T
from     Customer c join Order o
         on c.cust# = o.cust#
group by c.cust#, cname
```

```
select c.cust#, cname, T
from   customer c,
       (select   cust#, sum(totalprice) as T
        from     Order
        group by cust#
       ) as o
where  c.cust# = o.cust#
```

Example 1



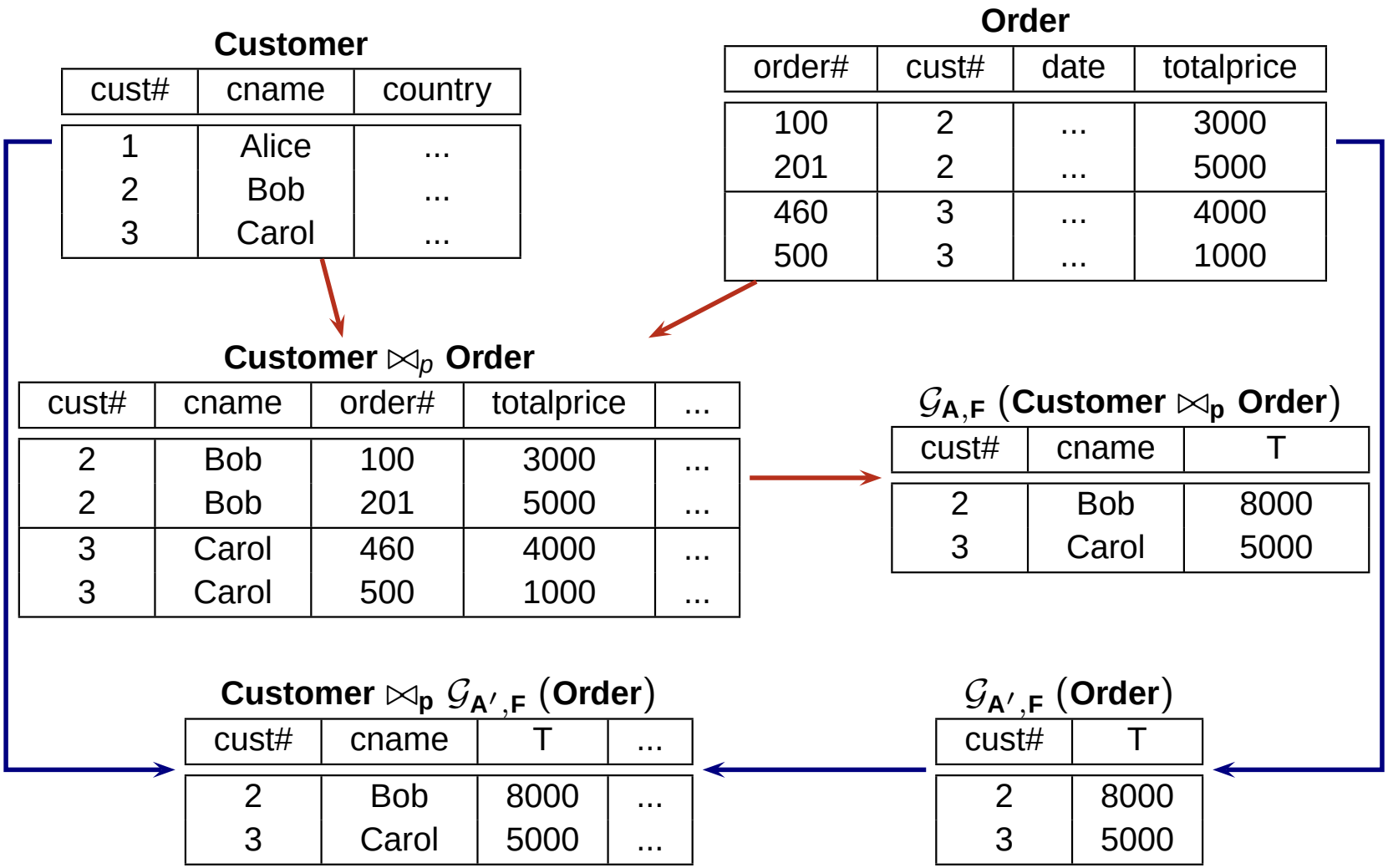
(a) Group by after join



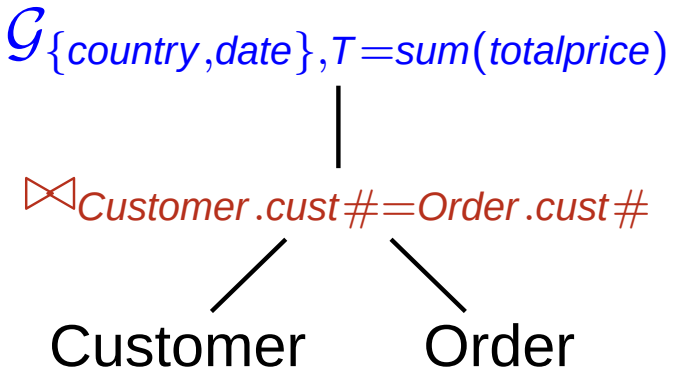
(b) Group by before join

Example 1

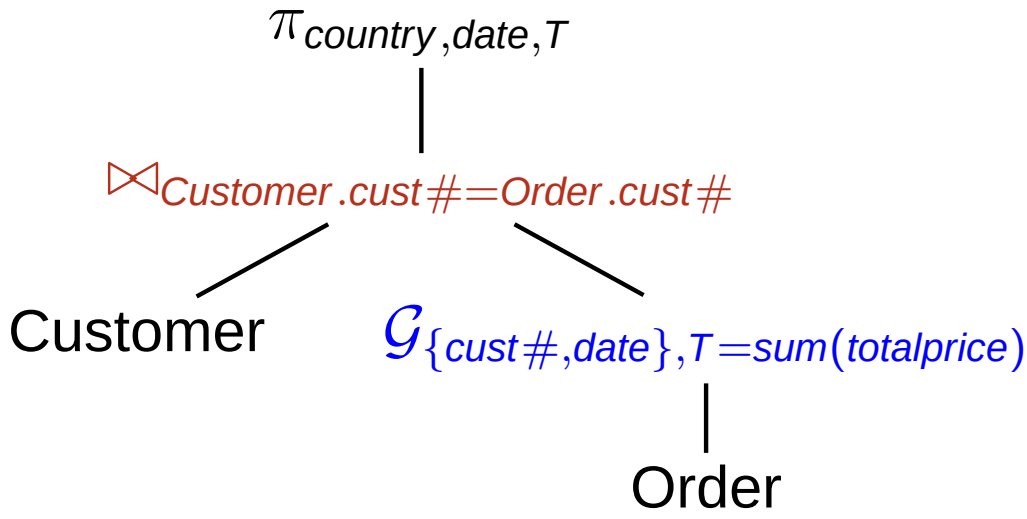
$$A = \{cust\#, cname\}, A' = \{cust\#\}, columns(p) = \{cust\#\}$$



Example 2



(a) Group by after join



(b) Group by before join

Example 2

$A = \{country, date\}, A' = \{cust\#, date\}, columns(p) = \{cust\#\}$

Customer

cust#	cname	country
1	Alice	UK
2	Bob	US
3	Carol	US

Order

order#	cust#	date	totalprice
100	2	2012-12-10	3000
201	2	2013-01-04	5000
460	3	2012-12-10	4000
500	3	2013-01-04	1000

Customer $\bowtie_p$ Order

cust#	country	date	totalprice	...
2	US	2012-12-10	3000	...
3	US	2012-12-10	4000	...
2	US	2013-01-04	5000	...
3	US	2013-01-04	1000	...

$\mathcal{G}_{A,F}$ (**Customer $\bowtie_p$ Order**)

country	date	T
US	2012-12-10	7000
US	2013-01-04	6000

Customer $\bowtie_p$ $\mathcal{G}_{A',F}$ (Order)

country	date	T	...
US	2012-12-10	3000	...
US	2013-01-04	5000	...
US	2012-12-10	4000	...
US	2013-01-04	1000	...

$\mathcal{G}_{A',F}$ (**Order**)

cust#	date	T
2	2012-12-10	3000
2	2013-01-04	5000
3	2012-12-10	4000
3	2013-01-04	1000

Pushing group by below join

$\mathcal{G}_{A,F}\ (S\ \bowtie_p\ R)$ can be rewritten as

$$\pi_{A,F}(S\ \bowtie_p\ (\mathcal{G}_{A',F}\ R))$$

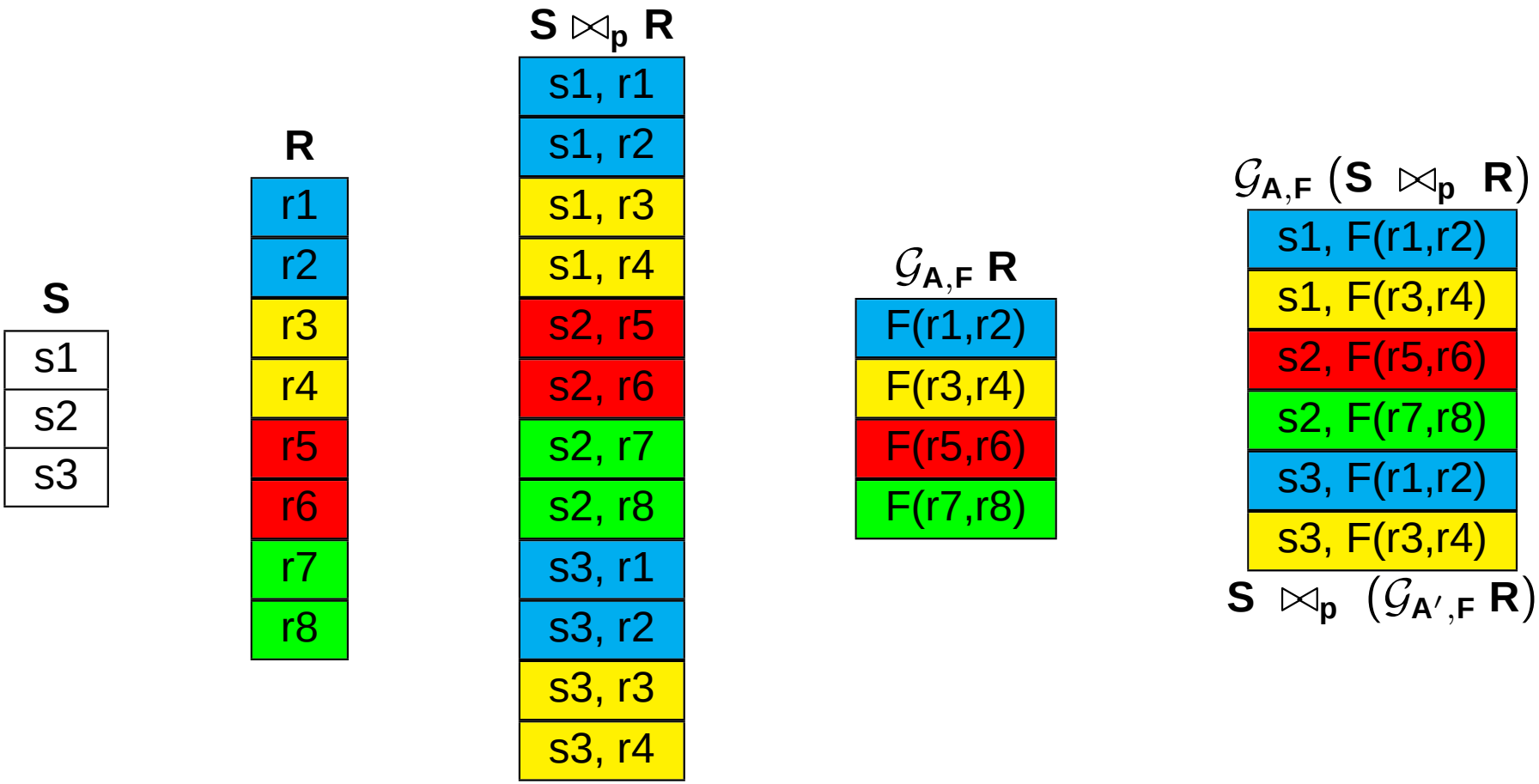
where

$$A' = A\ \cup\ columns(p) - columns(S)$$

iff the following conditions hold:

1. $columns(p) \cap columns(R) \subseteq A$,
2. $key(S) \subseteq A$, and
3. $columns(F) \subseteq columns(R)$

Pushing group by below join (cont.)



Tuples with same color have same $A \cap columns(R)$ values

Revisiting Example 1

- $\mathcal{G}_{A,T=sum(totalprice)}\,(Customer \bowtie_{cust\#} Order)$
  where $A = \{cust\#, cname\}$
- $\pi_{cust\#,cname,T}(Customer \bowtie_{cust\#} (\mathcal{G}_{A',T=sum(totalprice)}\, Order))$
  where $A' = \{cust\#\}$

- S = Customer
- $key(S) = cust\#$
- R = Order
- $columns(p) = \{Customer.cust\#, Order.cust\#\}$
- $columns(F) = \{Order.totalprice\}$

Revisiting Example 2

- $\mathcal{G}_{A,T=sum(totalprice)}\,(Customer \bowtie_{cust\#} Order)$
  where $A = \{country, date\}$
- $\pi_{country,date,T}(Customer \bowtie_{cust\#} (\mathcal{G}_{A',T=sum(totalprice)}\ Order))$
  where $A' = \{date, cust\#\}$

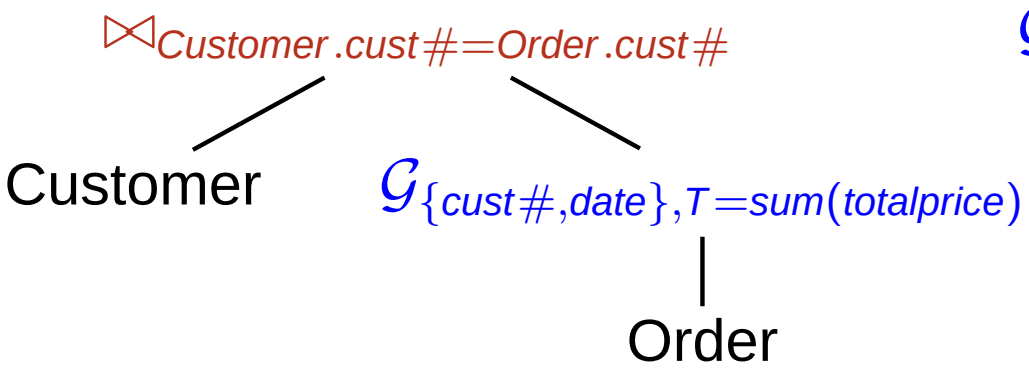
- S = Customer
- $key(S) = cust\#$
- R = Order
- $columns(p) = \{Customer.cust\#, Order.cust\#\}$
- $columns(F) = \{Order.totalprice\}$

Pulling group by above join

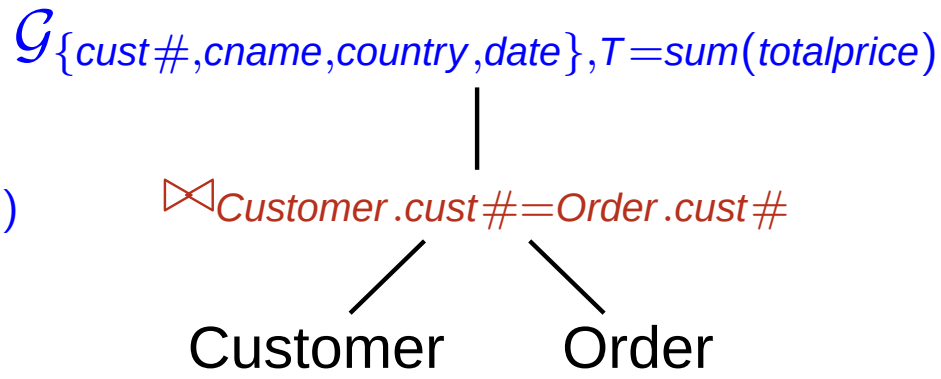
```
select c.cust#, cname, country, date, T
from   customer c,
       (select   cust#, date, sum(totalprice) as T
        from     Order
        group by cust#, date
       ) as o
where  c.cust# = o.cust#
```

```
select   c.cust#, cname, country, date,
         sum(o.totalprice) as T
from     Customer c join Order o
         on c.cust# = o.cust#
group by c.cust#, cname, country, date
```

Example 3



(a) Group by before join



(b) Group by after join

Example 3

$$A = \{cust\#, date\}, A' = \{cust\#, cname, country, date\}$$

Customer

cust#	cname	country
1	Alice	UK
2	Bob	US
3	Carol	US

Order

order#	cust#	date	totalprice
100	2	2012-12-10	3000
201	2	2013-01-04	5000
460	3	2012-12-10	4000
500	3	2013-01-04	1000

$\mathcal{G}_{\mathbf{A'},\mathbf{F}}$ (**Customer** $\bowtie_{\mathbf{p}}$ **Order**)

cust#	cname	country	date	totalprice
2	Bob	US	2012-12-10	3000
2	Bob	US	2013-01-04	5000
3	Carol	US	2012-12-10	4000
3	Carol	US	2013-01-04	1000

Customer $\bowtie_{\mathbf{p}}$ $\mathcal{G}_{\mathbf{A},\mathbf{F}}$ (**Order**)

$\mathcal{G}_{\mathbf{A},\mathbf{F}}$ (**Order**)

cust#	date	T
2	2012-12-10	3000
2	2013-01-04	5000
3	2012-12-10	4000
3	2013-01-04	1000

Pulling group by above join

$S \bowtie_p (\mathcal{G}_{A,F}\ R)$ can be rewritten as

$$\mathcal{G}_{A',F}\ (S \bowtie_p R)$$

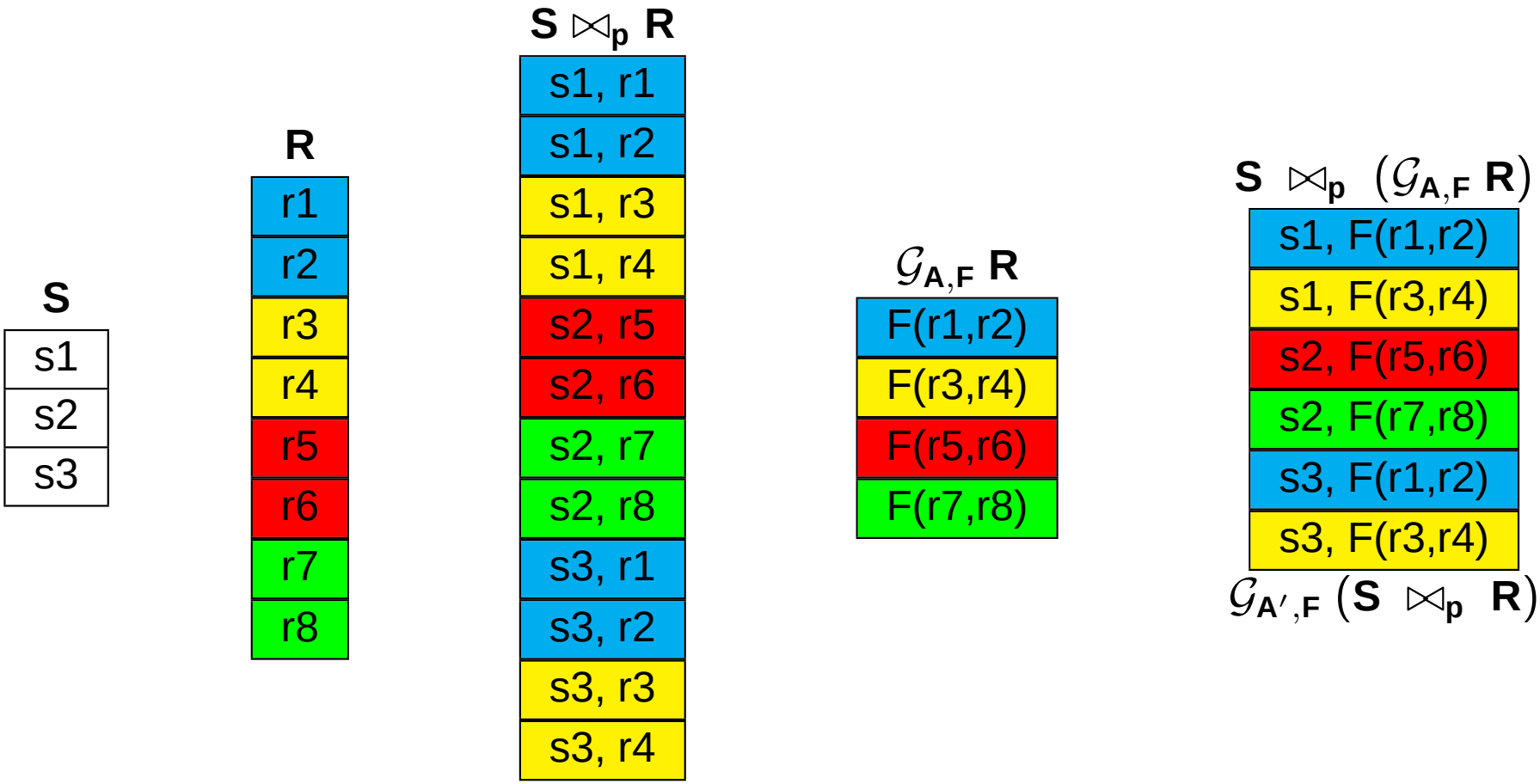
where

$$A' = A \cup columns(S)$$

iff the following conditions hold:

1. S has a key, and
2. p does not refer to aggregated columns in F; i.e., $columns(p) \cap columns(R) \subseteq A$

Pulling group by above join



Tuples with same color have same $A \cap columns(R)$ values

Pulling group by above join (cont.)

S

s1
s1
s2
s3

R

r1
r2
r3
r4
r5
r6
r7
r8

S $\bowtie_p$ R

s1, r1
s1, r1
s1, r2
s1, r2
s1, r3
s1, r3
s1, r4
s1, r4
s2, r5
s2, r6
s2, r7
s2, r8
s3, r1
s3, r2
s3, r3
s3, r4

$\mathcal{G}_{A,F}$ R

F(r1,r2)
F(r3,r4)
F(r5,r6)
F(r7,r8)

S $\bowtie_p$ ($\mathcal{G}_{A,F}$ R)

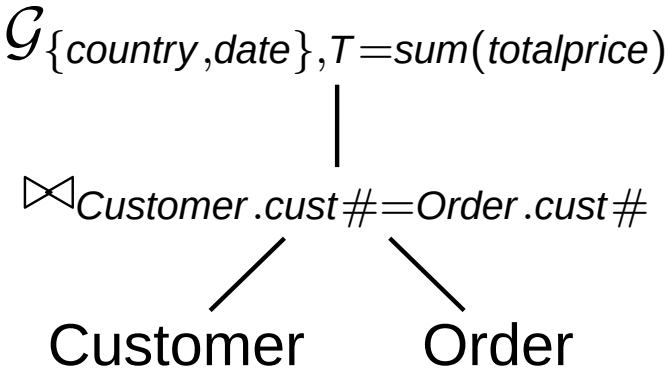
s1, F(r1,r2)
s1, F(r3,r4)
s1, F(r1,r2)
s1, F(r3,r4)
s2, F(r5,r6)
s2, F(r7,r8)
s3, F(r1,r2)
s3, F(r3,r4)

Suppose S does not have a key (i.e., S contains duplicates)

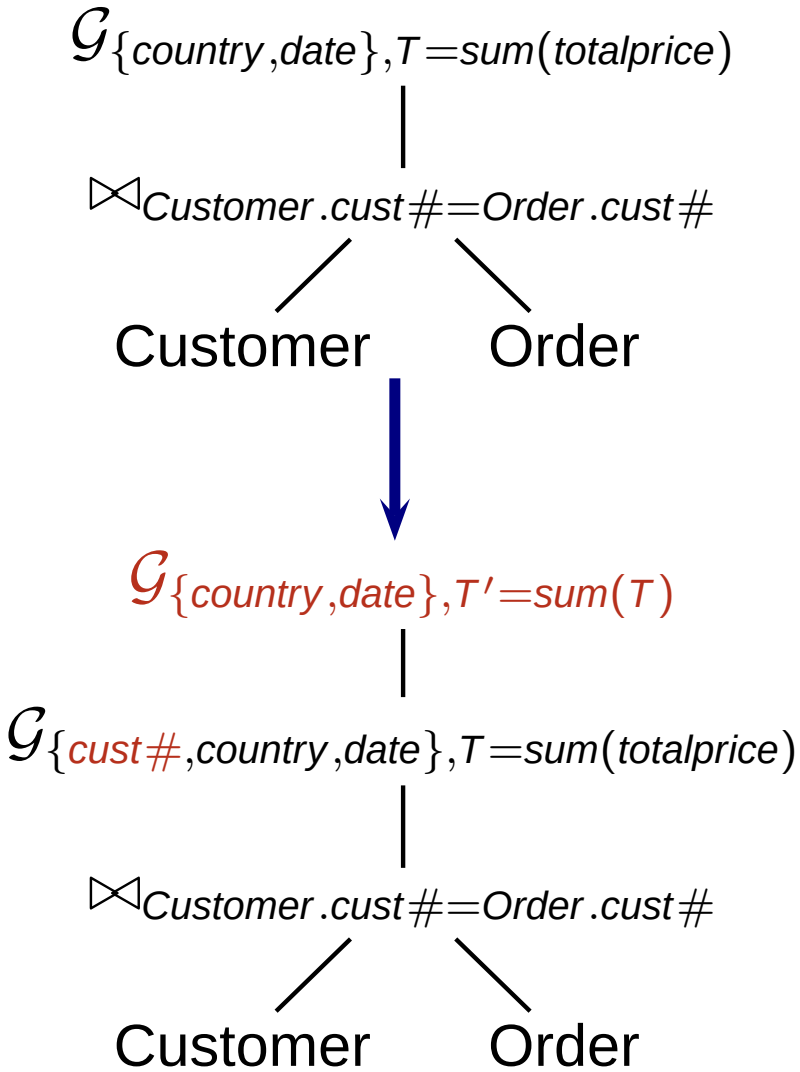
$\mathcal{G}_{A',F}$ (S $\bowtie_p$ R)

s1, F(r1,r1,r2,r2)
s1, F(r3,r3,r4,r4)
s2, F(r5,r6)
s2, F(r7,r8)
s3, F(r1,r2)
s3, F(r3,r4)

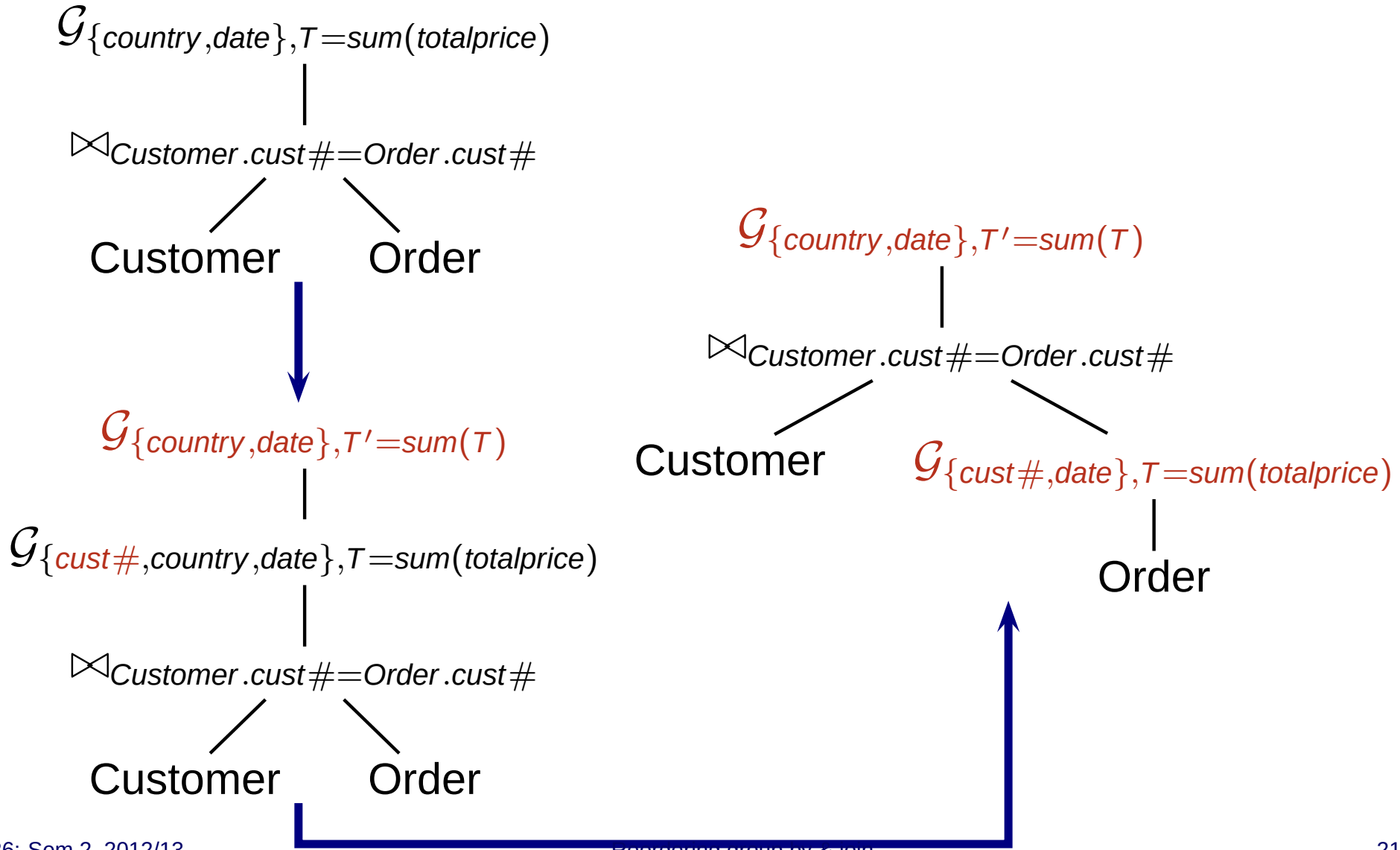
Revisiting Example 2



Revisiting Example 2



Revisiting Example 2



Revisiting Example 2

$$A = \{country, date\}, F = \{T = sum(totalprice)\},$$
$$A' = \{cust\#, date\}, F' = \{T' = sum(T)\}$$

Customer

cust#	cname	country
1	Alice	UK
2	Bob	US
3	Carol	US

Order

order#	cust#	date	totalprice
100	2	2012-12-10	3000
201	2	2013-01-04	5000
460	3	2012-12-10	4000
500	3	2013-01-04	1000

Customer $\bowtie_p$ Order

cust#	country	date	totalprice	...
2	US	2012-12-10	3000	...
3	US	2012-12-10	4000	...
2	US	2013-01-04	5000	...
3	US	2013-01-04	1000	...

$\mathcal{G}_{\mathbf{A},\mathbf{F}}$ (**Customer $\bowtie_p$ Order**)

country	date	T
US	2012-12-10	7000
US	2013-01-04	6000

$\mathcal{G}_{\mathbf{A},\mathbf{F'}}$ (**X**)

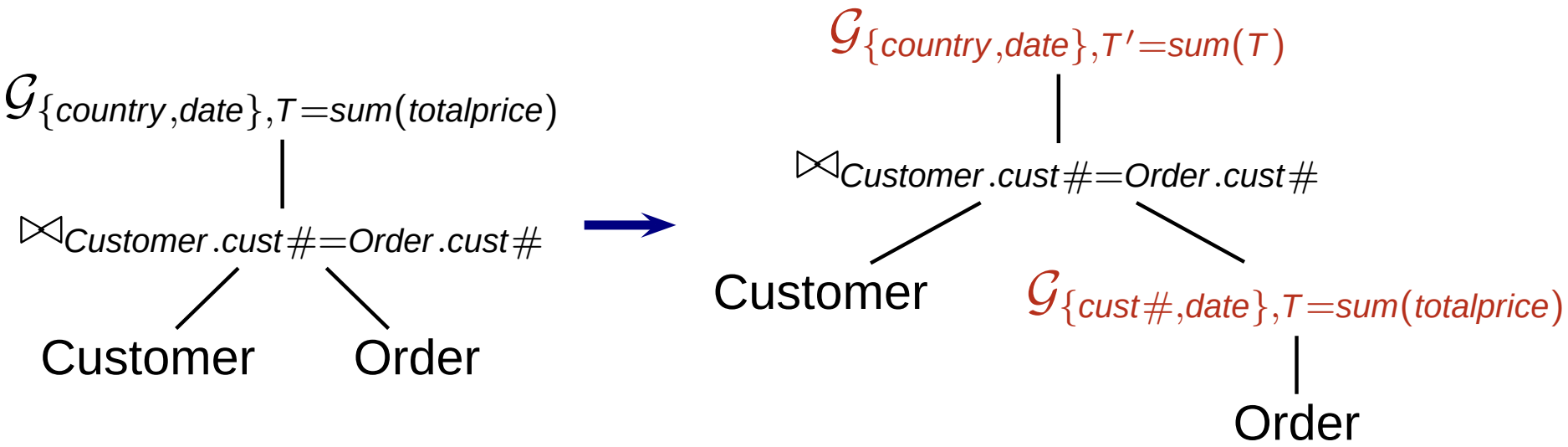
X = Customer $\bowtie_p$ $\mathcal{G}_{\mathbf{A'},\mathbf{F}}$ (Order)

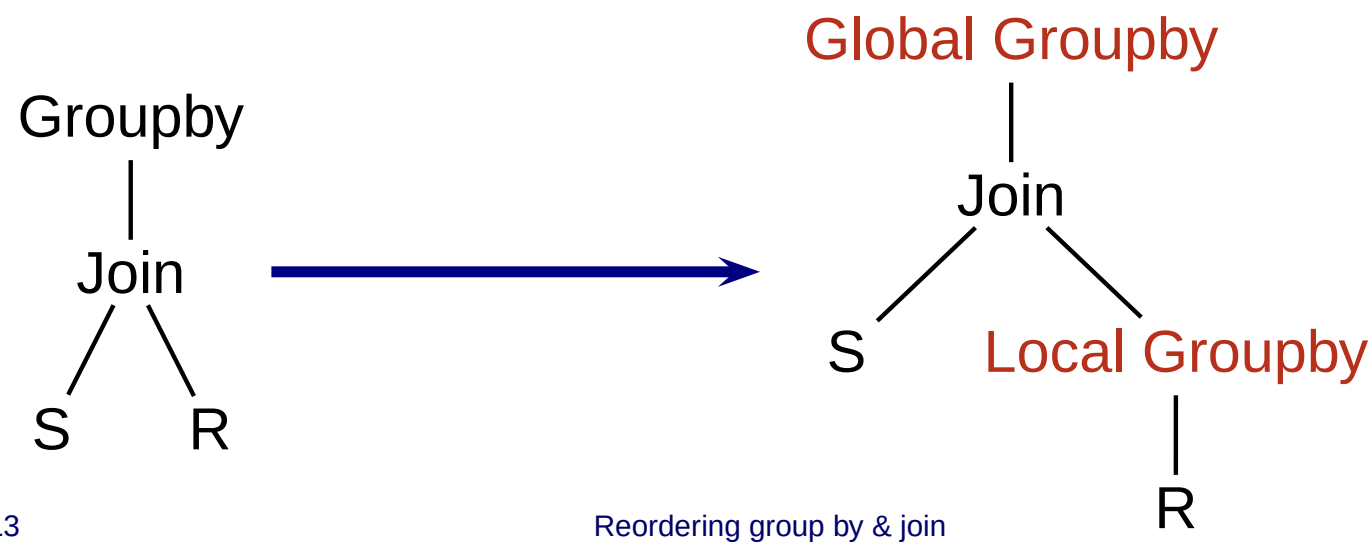
country	date	T	...
US	2012-12-10	3000	...
US	2013-01-04	5000	...
US	2012-12-10	4000	...
US	2013-01-04	1000	...

$\mathcal{G}_{\mathbf{A'},\mathbf{F}}$ (**Order**)

cust#	date	T
2	2012-12-10	3000
2	2013-01-04	5000
3	2012-12-10	4000
3	2013-01-04	1000

Local/Global group by





Local/Global group by (cont.)

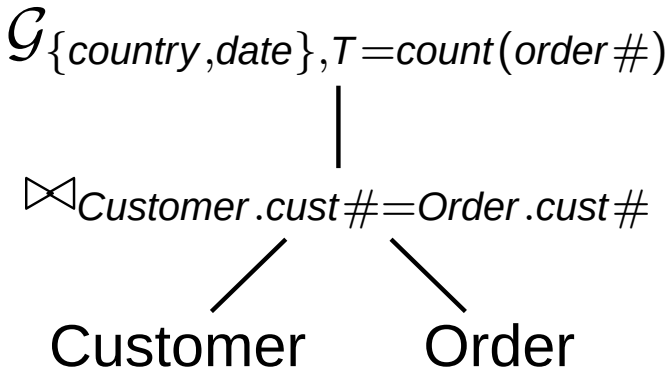
- Consider an aggregation function f on a relation R
- Let R be partitioned into n groups
  - $R = R_1 \cup \cdots \cup R_n$,
  - $R_i \cap R_j = \emptyset,\ \forall\, i, j \in [1, n], i \neq j$
- f_ℓ = a local aggregate function
- f_g = a global aggregate function

$$
\begin{aligned}
f(R) &= f(R_1 \cup \cdots \cup R_n) \\
&= f_g(f_\ell(R_1) \cup \cdots \cup f_\ell(R_n))
\end{aligned}
$$

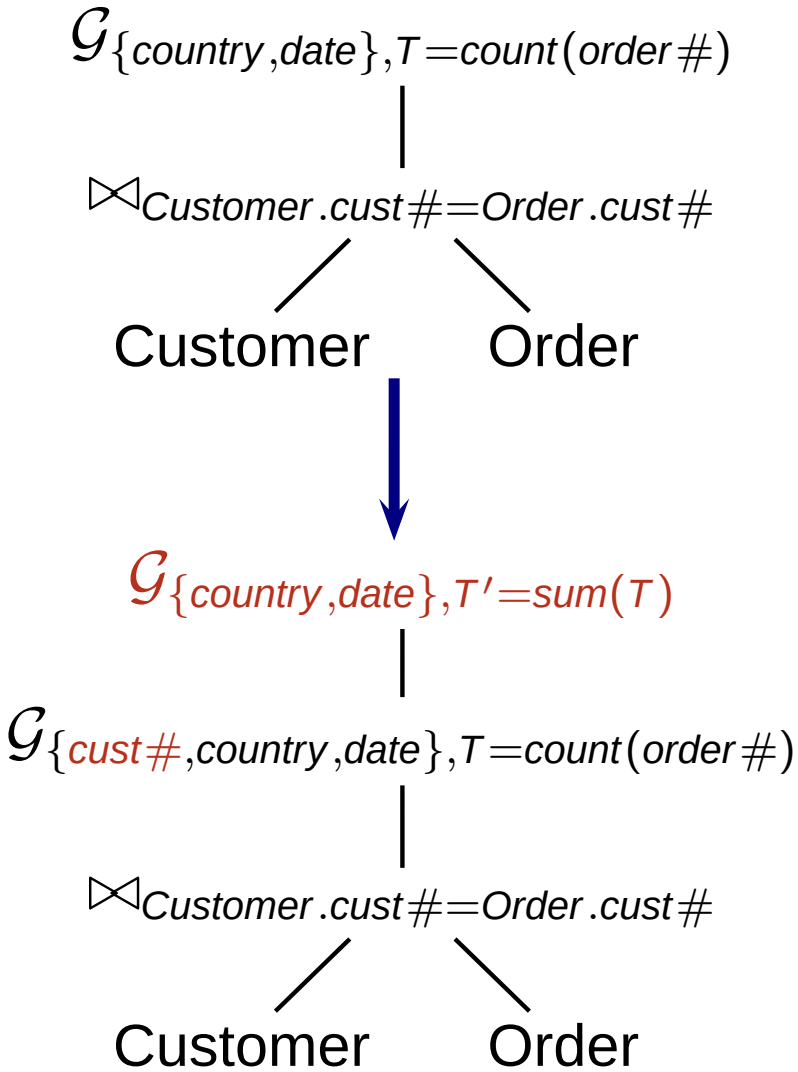
Local/Global group by (cont.)

$\mathbf{f}$	$\mathbf{f}_{\ell}$	$\mathbf{f}_{\mathbf{g}}$
SUM	SUM	SUM
MIN	MIN	MIN
MAX	MAX	MAX
COUNT	COUNT	SUM

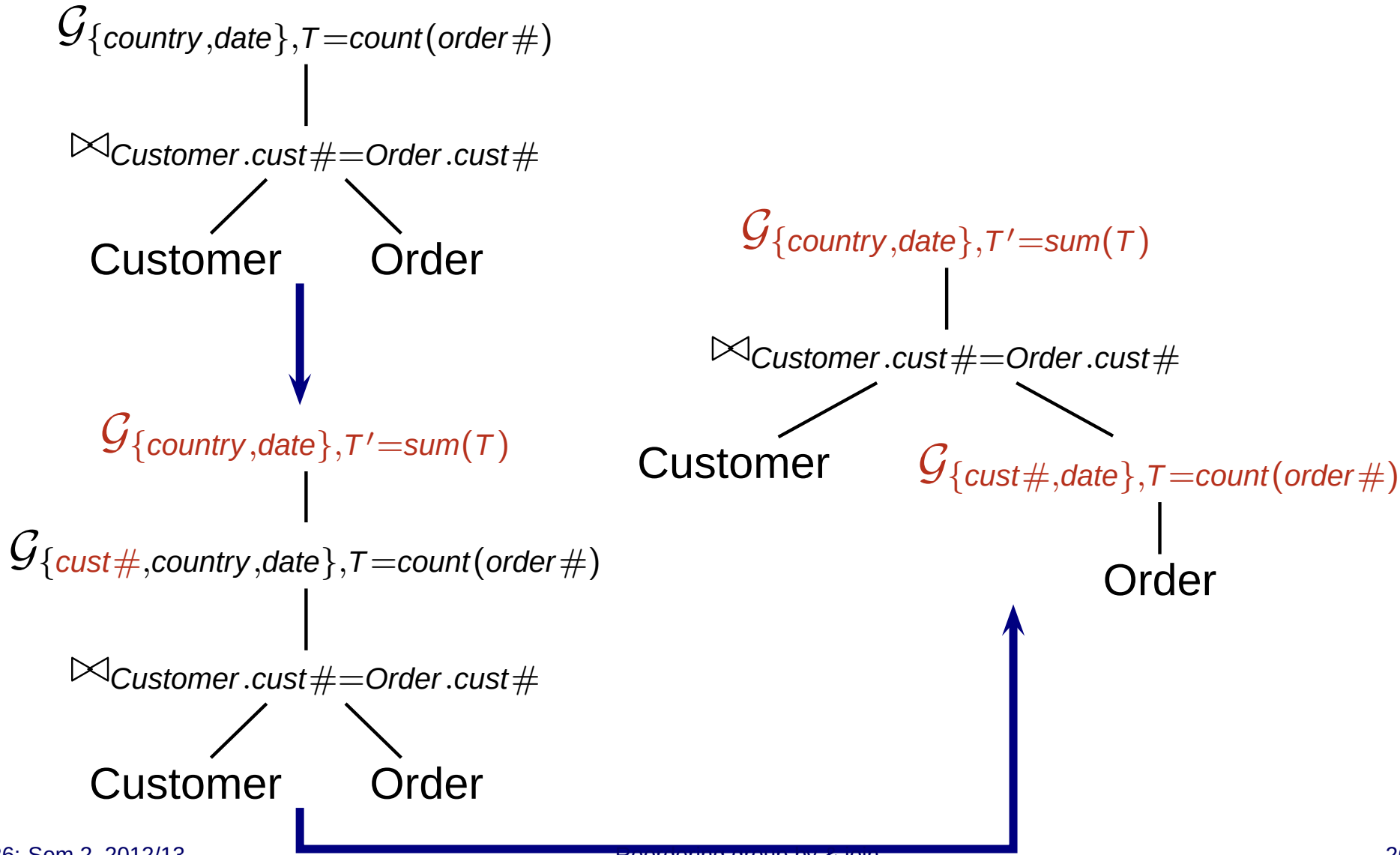
Example 4


$\mathcal{G}_{\{country,date\},T=count(order\#)}$
$\bowtie_{Customer.cust\#=Order.cust\#}$
Customer
Order

Example 4


$\mathcal{G}_{\{country,date\},T=count(order\#)}$
$\bowtie_{Customer.cust\#=Order.cust\#}$
Customer
Order
$\mathcal{G}_{\{country,date\},T'=sum(T)}$
$\mathcal{G}_{\{cust\#,country,date\},T=count(order\#)}$
$\bowtie_{Customer.cust\#=Order.cust\#}$
Customer
Order

Example 4


$\mathcal{G}_{\{country,date\},T=count(order\#)}$
$\bowtie_{Customer.cust\#=Order.cust\#}$
Customer
Order
$\mathcal{G}_{\{country,date\},T'=sum(T)}$
$\mathcal{G}_{\{cust\#,country,date\},T=count(order\#)}$
$\bowtie_{Customer.cust\#=Order.cust\#}$
Customer
Order
$\mathcal{G}_{\{country,date\},T'=sum(T)}$
$\bowtie_{Customer.cust\#=Order.cust\#}$
Customer
$\mathcal{G}_{\{cust\#,date\},T=count(order\#)}$
Order

Example 4

$$A = \{country, date\}, F = \{T = count(order\#)\},$$
$$A' = \{cust\#, date\}, F' = \{T' = sum(T)\}$$

Customer

cust#	cname	country
1	Alice	UK
2	Bob	US
3	Carol	US

Order

order#	cust#	date	totalprice
100	2	2013-01-04	3000
201	2	2013-01-04	5000
460	3	2012-12-10	4000
500	3	2013-01-04	1000

Customer $\bowtie_p$ Order

cust#	country	date	order#	...
3	US	2012-12-10	201	...
2	US	2013-01-04	100	...
2	US	2013-01-04	460	...
3	US	2013-01-04	500	...

$\mathcal{G}_{\mathbf{A},\mathbf{F}}$ (**Customer $\bowtie_p$ Order**)

country	date	T
US	2012-12-10	1
US	2013-01-04	3

$\mathcal{G}_{\mathbf{A},\mathbf{F}'}$ (**X**)

X = **Customer** $\bowtie_p$ $\mathcal{G}_{\mathbf{A}',\mathbf{F}}$ (**Order**)

country	date	T	...
US	2013-01-04	2	...
US	2012-12-10	1	...
US	2013-01-04	1	...

$\mathcal{G}_{\mathbf{A}',\mathbf{F}}$ (**Order**)

cust#	date	T
2	2013-01-04	2
3	2012-12-10	1
3	2013-01-04	1

References

Required Readings

- C.A. Galindo-Legaria, M.M. Joshi, *Orthogonal optimization of subqueries and aggregation*, SIGMOD 2001

Additional Readings

- W.P. Yan, P.A. Larson, *Performing group-by before join*, ICDE 1994
- W.P. Yan, P.A. Larson, *Eager aggregation and lazy aggregation*, VLDB 1995

Quiz 2

Rewrite the following query to eliminate the nested subquery such that the group by operation is performed as early as possible.

```
SELECT cust#
FROM   Customer c
WHERE  NOT EXISTS
       (SELECT 1
       FROM    Order o
       WHERE   c.cust# = o.cust#
       AND     o.date = '01-01-2013')
```