Jens Dittrich

Example

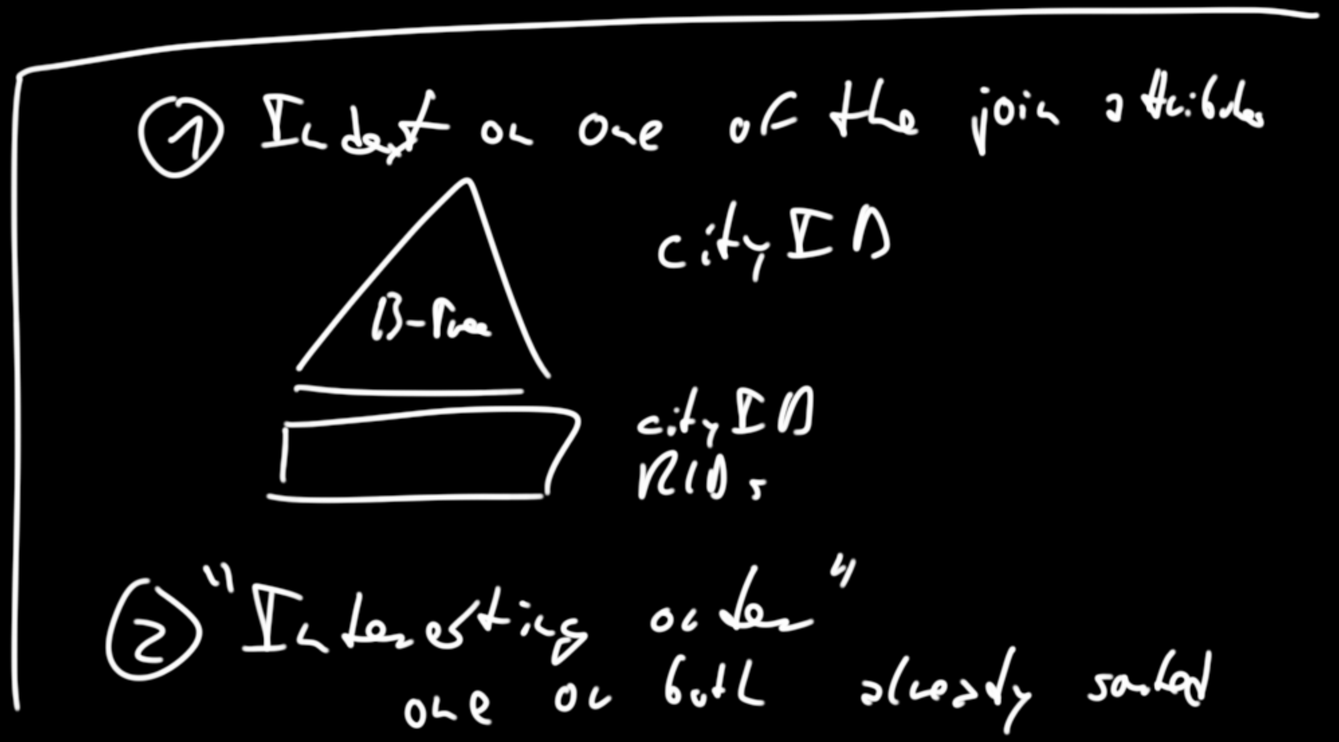
Customers		
name	**street**	**cityID**
peter	minstreet	0
steve	macstreet	1
mike	longstreet	9
tim	unistreet	9
hans	msstreet	5
jens	shortstreet	1
frank	minstreet	0
olaf	macstreet	9
stefan	unistreet	0
alekh	unistreet	7
felix	macstreet	5
jorge	minstreet	9

Cities_Dictionary	
cityID	**city**
5	berlin
1	cuppertino
9	saarbruecken
3	paris
0	new york
7	london

Sort Both Tables on Join Attributes

Customers		
name	**street**	**cityID**
frank	minstreet	0
peter	minstreet	0
stefan	unistreet	0
jens	shortstreet	1
steve	macstreet	1
felix	macstreet	5
hans	msstreet	5
alekh	unistreet	7
jorge	minstreet	9
mike	longstreet	9
olaf	macstreet	9
tim	unistreet	9

Cities_Dictionary	
cityID	**city**
0	new york
1	cuppertino
3	paris
5	berlin
7	london
9	saarbruecken

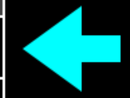


Values Pointed to are Equal!

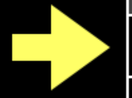
R

Customers		
name	**street**	**cityID**
frank	minstreet	0
peter	minstreet	0
stefan	unistreet	0
jens	shortstreet	1
steve	macstreet	1
felix	macstreet	5
hans	msstreet	5
alekh	unistreet	7
jorge	minstreet	9
mike	longstreet	9
olaf	macstreet	9
tim	unistreet	9

PR



PS



S

Cities_Dictionary	
cityID	**city**
0	new york
1	cuppertino
3	paris
5	berlin
7	london
9	saarbruecken

Customers NATURAL JOIN Cities_Dictionary			
name	**street**	**cityID**	**city**

Found a Join Result

move left pointer

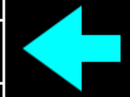
Customers		
name	**street**	**cityID**
frank	minstreet	0
peter	minstreet	0
stefan	unistreet	0
jens	shortstreet	1
steve	macstreet	1
felix	macstreet	5
hans	msstreet	5
alekh	unistreet	7
jorge	minstreet	9
mike	longstreet	9
olaf	macstreet	9
tim	unistreet	9

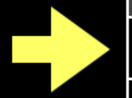
Cities_Dictionary	
cityID	**city**
0	new york
1	cuppertino
3	paris
5	berlin
7	london
9	saarbruecken

Customers NATURAL JOIN Cities_Dictionary			
name	**street**	**cityID**	**city**
frank	minstreet	0	new york

Values Pointed to are Equal!

Customers		
name	**street**	**cityID**
frank	minstreet	0
peter	minstreet	0
stefan	unistreet	0
jens	shortstreet	1
steve	macstreet	1
felix	macstreet	5
hans	msstreet	5
alekh	unistreet	7
jorge	minstreet	9
mike	longstreet	9
olaf	macstreet	9
tim	unistreet	9



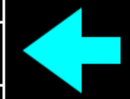


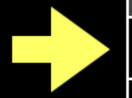
Cities_Dictionary	
cityID	**city**
0	new york
1	cuppertino
3	paris
5	berlin
7	london
9	saarbruecken

Customers NATURAL JOIN Cities_Dictionary			
name	**street**	**cityID**	**city**
frank	minstreet	0	new york

Found a Join Result

Customers		
name	**street**	**cityID**
frank	minstreet	0
peter	minstreet	0
stefan	unistreet	0
jens	shortstreet	1
steve	macstreet	1
felix	macstreet	5
hans	msstreet	5
alekh	unistreet	7
jorge	minstreet	9
mike	longstreet	9
olaf	macstreet	9
tim	unistreet	9



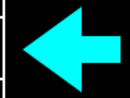


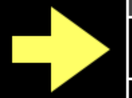
Cities_Dictionary	
cityID	**city**
0	new york
1	cuppertino
3	paris
5	berlin
7	london
9	saarbruecken

Customers NATURAL JOIN Cities_Dictionary			
name	**street**	**cityID**	**city**
frank	minstreet	0	new york
peter	minstreet	0	new york

Values Pointed to are Equal!

Customers		
name	**street**	**cityID**
frank	minstreet	0
peter	minstreet	0
stefan	unistreet	0
jens	shortstreet	1
steve	macstreet	1
felix	macstreet	5
hans	msstreet	5
alekh	unistreet	7
jorge	minstreet	9
mike	longstreet	9
olaf	macstreet	9
tim	unistreet	9



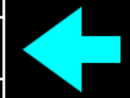


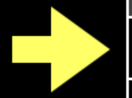
Cities_Dictionary	
cityID	**city**
0	new york
1	cuppertino
3	paris
5	berlin
7	london
9	saarbruecken

Customers NATURAL JOIN Cities_Dictionary			
name	**street**	**cityID**	**city**
frank	minstreet	0	new york
peter	minstreet	0	new york

Found a Join Result

Customers		
name	**street**	**cityID**
frank	minstreet	0
peter	minstreet	0
stefan	unistreet	0
jens	shortstreet	1
steve	macstreet	1
felix	macstreet	5
hans	msstreet	5
alekh	unistreet	7
jorge	minstreet	9
mike	longstreet	9
olaf	macstreet	9
tim	unistreet	9



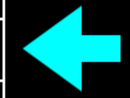


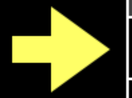
Cities_Dictionary	
cityID	**city**
0	new york
1	cuppertino
3	paris
5	berlin
7	london
9	saarbruecken

Customers NATURAL JOIN Cities_Dictionary			
name	**street**	**cityID**	**city**
frank	minstreet	0	new york
peter	minstreet	0	new york
stefan	unistreet	0	new york

Values Pointed to are NOT Equal!

Customers		
name	**street**	**cityID**
frank	minstreet	0
peter	minstreet	0
stefan	unistreet	0
jens	shortstreet	1
steve	macstreet	1
felix	macstreet	5
hans	msstreet	5
alekh	unistreet	7
jorge	minstreet	9
mike	longstreet	9
olaf	macstreet	9
tim	unistreet	9



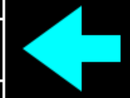


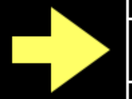
Cities_Dictionary	
cityID	**city**
0	new york
1	cuppertino
3	paris
5	berlin
7	london
9	saarbruecken

Customers NATURAL JOIN Cities_Dictionary			
name	**street**	**cityID**	**city**
frank	minstreet	0	new york
peter	minstreet	0	new york
stefan	unistreet	0	new york

Values Pointed to are Equal!

Customers		
name	**street**	**cityID**
frank	minstreet	0
peter	minstreet	0
stefan	unistreet	0
jens	shortstreet	1
steve	macstreet	1
felix	macstreet	5
hans	msstreet	5
alekh	unistreet	7
jorge	minstreet	9
mike	longstreet	9
olaf	macstreet	9
tim	unistreet	9



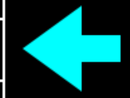


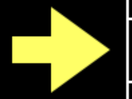
Cities_Dictionary	
cityID	**city**
0	new york
1	cuppertino
3	paris
5	berlin
7	london
9	saarbruecken

Customers NATURAL JOIN Cities_Dictionary			
name	**street**	**cityID**	**city**
frank	minstreet	0	new york
peter	minstreet	0	new york
stefan	unistreet	0	new york

Found a Join Result

Customers		
name	**street**	**cityID**
frank	minstreet	0
peter	minstreet	0
stefan	unistreet	0
jens	shortstreet	1
steve	macstreet	1
felix	macstreet	5
hans	msstreet	5
alekh	unistreet	7
jorge	minstreet	9
mike	longstreet	9
olaf	macstreet	9
tim	unistreet	9



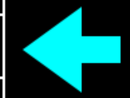


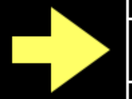
Cities_Dictionary	
cityID	**city**
0	new york
1	cuppertino
3	paris
5	berlin
7	london
9	saarbruecken

Customers NATURAL JOIN Cities_Dictionary			
name	**street**	**cityID**	**city**
frank	minstreet	0	new york
peter	minstreet	0	new york
stefan	unistreet	0	new york
jens	shortstreet	1	cuppertino

Values Pointed to are Equal!

Customers		
name	**street**	**cityID**
frank	minstreet	0
peter	minstreet	0
stefan	unistreet	0
jens	shortstreet	1
steve	macstreet	1
felix	macstreet	5
hans	msstreet	5
alekh	unistreet	7
jorge	minstreet	9
mike	longstreet	9
olaf	macstreet	9
tim	unistreet	9



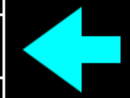


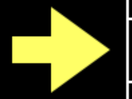
Cities_Dictionary	
cityID	**city**
0	new york
1	cuppertino
3	paris
5	berlin
7	london
9	saarbruecken

Customers NATURAL JOIN Cities_Dictionary			
name	**street**	**cityID**	**city**
frank	minstreet	0	new york
peter	minstreet	0	new york
stefan	unistreet	0	new york
jens	shortstreet	1	cuppertino

Found a Join Result

Customers		
name	**street**	**cityID**
frank	minstreet	0
peter	minstreet	0
stefan	unistreet	0
jens	shortstreet	1
steve	macstreet	1
felix	macstreet	5
hans	msstreet	5
alekh	unistreet	7
jorge	minstreet	9
mike	longstreet	9
olaf	macstreet	9
tim	unistreet	9



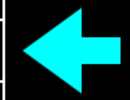


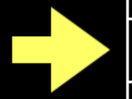
Cities_Dictionary	
cityID	**city**
0	new york
1	cuppertino
3	paris
5	berlin
7	london
9	saarbruecken

Customers NATURAL JOIN Cities_Dictionary			
name	**street**	**cityID**	**city**
frank	minstreet	0	new york
peter	minstreet	0	new york
stefan	unistreet	0	new york
jens	shortstreet	1	cuppertino
steve	macstreet	1	cuppertino

Values Pointed to are NOT Equal!

Customers		
name	**street**	**cityID**
frank	minstreet	0
peter	minstreet	0
stefan	unistreet	0
jens	shortstreet	1
steve	macstreet	1
felix	macstreet	5
hans	msstreet	5
alekh	unistreet	7
jorge	minstreet	9
mike	longstreet	9
olaf	macstreet	9
tim	unistreet	9



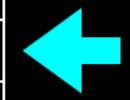


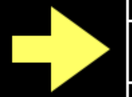
Cities_Dictionary	
cityID	**city**
0	new york
1	cuppertino
3	paris
5	berlin
7	london
9	saarbruecken

Customers NATURAL JOIN Cities_Dictionary			
name	**street**	**cityID**	**city**
frank	minstreet	0	new york
peter	minstreet	0	new york
stefan	unistreet	0	new york
jens	shortstreet	1	cuppertino
steve	macstreet	1	cuppertino

Values Pointed to are NOT Equal!

Customers		
name	**street**	**cityID**
frank	minstreet	0
peter	minstreet	0
stefan	unistreet	0
jens	shortstreet	1
steve	macstreet	1
felix	macstreet	5
hans	msstreet	5
alekh	unistreet	7
jorge	minstreet	9
mike	longstreet	9
olaf	macstreet	9
tim	unistreet	9



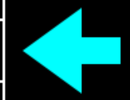


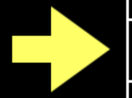
Cities_Dictionary	
cityID	**city**
0	new york
1	cuppertino
3	paris
5	berlin
7	london
9	saarbruecken

Customers NATURAL JOIN Cities_Dictionary			
name	**street**	**cityID**	**city**
frank	minstreet	0	new york
peter	minstreet	0	new york
stefan	unistreet	0	new york
jens	shortstreet	1	cuppertino
steve	macstreet	1	cuppertino

Values Pointed to are Equal!

Customers		
name	**street**	**cityID**
frank	minstreet	0
peter	minstreet	0
stefan	unistreet	0
jens	shortstreet	1
steve	macstreet	1
felix	macstreet	5
hans	msstreet	5
alekh	unistreet	7
jorge	minstreet	9
mike	longstreet	9
olaf	macstreet	9
tim	unistreet	9



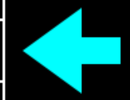


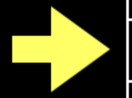
Cities_Dictionary	
cityID	**city**
0	new york
1	cuppertino
3	paris
5	berlin
7	london
9	saarbruecken

Customers NATURAL JOIN Cities_Dictionary			
name	**street**	**cityID**	**city**
frank	minstreet	0	new york
peter	minstreet	0	new york
stefan	unistreet	0	new york
jens	shortstreet	1	cuppertino
steve	macstreet	1	cuppertino

Found a Join Result

Customers		
name	**street**	**cityID**
frank	minstreet	0
peter	minstreet	0
stefan	unistreet	0
jens	shortstreet	1
steve	macstreet	1
felix	macstreet	5
hans	msstreet	5
alekh	unistreet	7
jorge	minstreet	9
mike	longstreet	9
olaf	macstreet	9
tim	unistreet	9



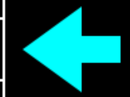


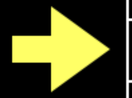
Cities_Dictionary	
cityID	**city**
0	new york
1	cuppertino
3	paris
5	berlin
7	london
9	saarbruecken

Customers NATURAL JOIN Cities_Dictionary			
name	**street**	**cityID**	**city**
frank	minstreet	0	new york
peter	minstreet	0	new york
stefan	unistreet	0	new york
jens	shortstreet	1	cuppertino
steve	macstreet	1	cuppertino
felix	macstreet	5	berlin

Values Pointed to are Equal!

Customers		
name	**street**	**cityID**
frank	minstreet	0
peter	minstreet	0
stefan	unistreet	0
jens	shortstreet	1
steve	macstreet	1
felix	macstreet	5
hans	msstreet	5
alekh	unistreet	7
jorge	minstreet	9
mike	longstreet	9
olaf	macstreet	9
tim	unistreet	9



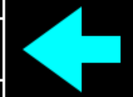


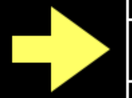
Cities_Dictionary	
cityID	**city**
0	new york
1	cuppertino
3	paris
5	berlin
7	london
9	saarbruecken

Customers NATURAL JOIN Cities_Dictionary			
name	**street**	**cityID**	**city**
frank	minstreet	0	new york
peter	minstreet	0	new york
stefan	unistreet	0	new york
jens	shortstreet	1	cuppertino
steve	macstreet	1	cuppertino
felix	macstreet	5	berlin

Found a Join Result

Customers		
name	**street**	**cityID**
frank	minstreet	0
peter	minstreet	0
stefan	unistreet	0
jens	shortstreet	1
steve	macstreet	1
felix	macstreet	5
hans	msstreet	5
alekh	unistreet	7
jorge	minstreet	9
mike	longstreet	9
olaf	macstreet	9
tim	unistreet	9



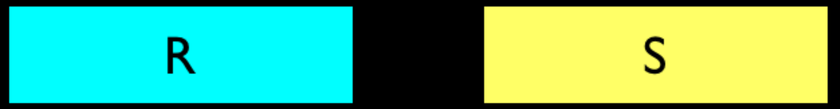


Cities_Dictionary	
cityID	**city**
0	new york
1	cuppertino
3	paris
5	berlin
7	london
9	saarbruecken

Customers NATURAL JOIN Cities_Dictionary			
name	**street**	**cityID**	**city**
frank	minstreet	0	new york
peter	minstreet	0	new york
stefan	unistreet	0	new york
jens	shortstreet	1	cuppertino
steve	macstreet	1	cuppertino
felix	macstreet	5	berlin
hans	msstreet	5	berlin

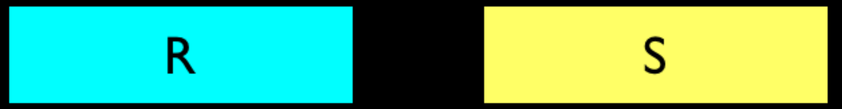
Sort-Merge Join

extensions: intervals, spatial, temporal



JP(r,s) := r.x == s.x //definition of the join predicate

Sort-Merge Join

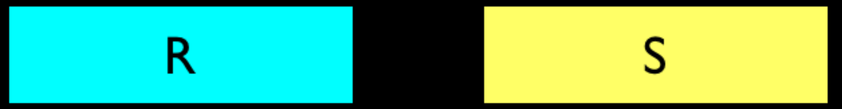



```
JP(r,s) := r.x == s.x                          //definition of the join predicate

SortMergeJoin( R, S, JP(r,s) ):
        sort(R on R.x);                        //sort R on join attribute x
        sort(S on S.x);                        //sort S on join attribute x
        Pointer PR = R[0];                     //initialize pointer to R
        Pointer PS = S[0];                     //initialize pointer to S
        do:                                    //start loop
            if PR.x == PS.x:                   //if values match
                output( (PR, PS) );            //output join result
```

How to project?

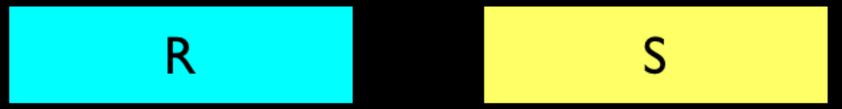
Sort-Merge Join




```
JP(r,s) := r.x == s.x                                  //definition of the join predicate

SortMergeJoin( R, S, JP(r,s) ):
        sort(R on R.x);                                //sort R on join attribute x
        sort(S on S.x);                                //sort S on join attribute x
        Pointer PR = R[0];                             //initialize pointer to R
        Pointer PS = S[0];                             //initialize pointer to S
        do:                                            //start loop
            if PR.x == PS.x:                           //if values match
                output( (PR, PS) );                    //output join result
            if PR.x <= PS.x:                           //find smaller join key (move left first!)
                PR++;                                  //move pointer to R forward
            else: PR.x > PS.x
                PS++;                                  //move pointer to S forward
```

Sort-Merge Join

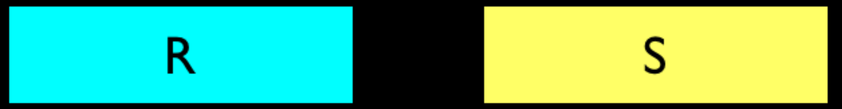



```
JP(r,s) := r.x == s.x                                   //definition of the join predicate

SortMergeJoin( R, S, JP(r,s) ):
        sort(R on R.x);                                 //sort R on join attribute x
        sort(S on S.x);                                 //sort S on join attribute x
        Pointer PR = R[0];                              //initialize pointer to R
        Pointer PS = S[0];                              //initialize pointer to S
        do:                                             //start loop
            if PR.x == PS.x:                            //if values match
                output( (PR, PS) );                     //output join result
            if PR.x <= PS.x:                            //find smaller join key (move left first!)
                PR++;                                   //move pointer to R forward
            else:
                PS++;                                   //move pointer to S forward
        while PR != R.end && PS != S.end                //both tables unprocessed
```

→LAB

Sort-Merge Join




```
JP(r,s) := r.x == s.x                          //definition of the join predicate

SortMergeJoin( R, S, JP(r,s) ):
        sort(R on R.x);                        //sort R on join attribute x
        sort(S on S.x);                        //sort S on join attribute x
        Pointer PR = R[0];                     //initialize pointer to R
        Pointer PS = S[0];                     //initialize pointer to S
        do:                                    //start loop
            if PR.x == PS.x:                   //if values match
                output( (PR, PS) );            //output join result
            if PR.x <= PS.x:                   //find smaller join key (move left first!)
                PR++;                          //move pointer to R forward
            else:
                PS++;                          //move pointer to S forward
        while PR != R.end && PS != S.end       //both tables unprocessed
         ...                                   //output join results...
            ...                                //...for non-empty table
```

Natural Join

FK

Customers		
name	**street**	**cityID**
frank	minstreet	0
peter	minstreet	0
stefan	unistreet	0
jens	shortstreet	1
steve	macstreet	1
felix	macstreet	5
hans	msstreet	5
alekh	unistreet	7
jorge	minstreet	9
mike	longstreet	9
olaf	macstreet	9
tim	unistreet	9

duplicates

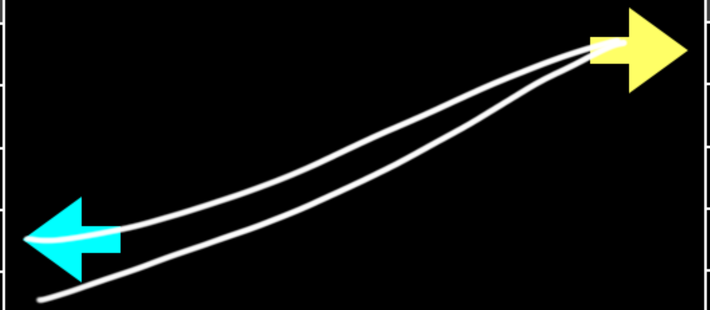
PK

Cities_Dictionary	
cityID	**city**
0	new york
1	cuppertino
3	paris
5	berlin
7	london
9	saarbruecken

no duplicates

Duplicates in both Columns!

Customers		
name	**street**	**cityID**
frank	minstreet	0
peter	minstreet	0
stefan	unistreet	0
jens	shortstreet	1
steve	macstreet	1
felix	macstreet	5
hans	msstreet	5
alekh	unistreet	7
jorge	minstreet	9
mike	longstreet	9
olaf	macstreet	9
tim	unistreet	9

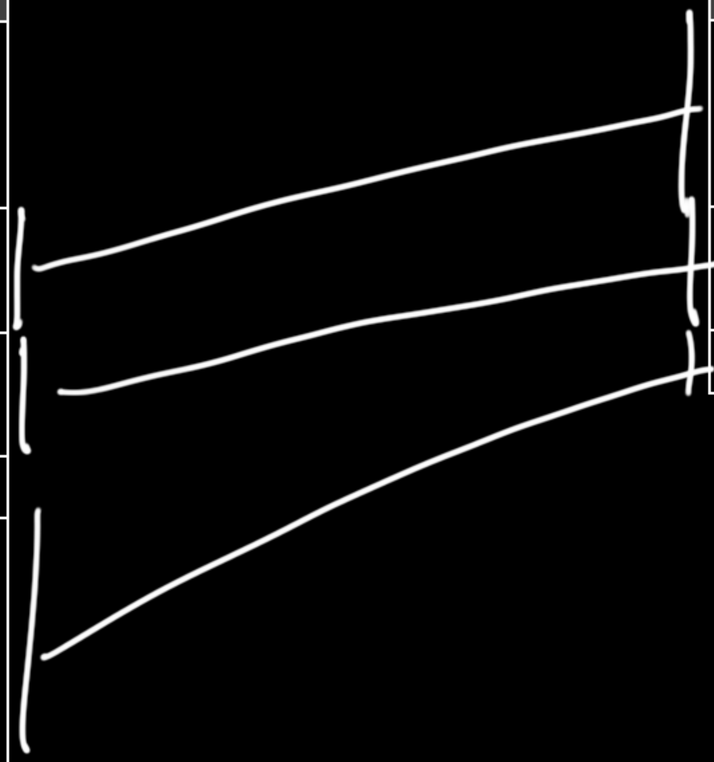


Cities_Dictionary	
cityNo	**city**
1	new york
1	cuppertino
1	paris
5	berlin
5	london
9	saarbruecken

Joins? CoGroups!

Customers		
name	**street**	**cityID**
frank	minstreet	0
peter	minstreet	
stefan	unistreet	
jens	shortstreet	1
steve	macstreet	
felix	macstreet	5
hans	msstreet	
alekh	unistreet	7
jorge	minstreet	9
mike	longstreet	
olaf	macstreet	
tim	unistreet	

Cities_Dictionary	
cityNo	**city**
1	new york
	cuppertino
	paris
5	berlin
	london
9	saarbruecken



1 : 2×3 = 6
5 : 2×2 = 4
9 : 4×1 = 4
→ 14 tuples

```
SELECT cityID
FROM Customers
GROUP BY cityID;
```

```
SELECT cityNo
FROM Cities_Dictionary
GROUP BY cityNo
```