

[BITE-SIZED POWER QUERY]

Clean pairwise matches in Power Query

What's a pairwise match?

Simple example with 2 columns

Ignoring order, the set of values in columns 1 and 2 on rows 1 and 2 are the same.

ID	column1	column2
1	blue	red
2	red	blue
3	yellow	blue
4	yellow	red

Simple example with 2 columns

One way of cleaning these rows is to select the first occurring row in each match and use it to replace the others

ID	column1	column2
1	blue	red
2	red	blue
3	yellow	blue
4	yellow	red

ID	column1	column2
1	blue	red
2	blue	red
3	yellow	blue
4	yellow	red

A Power Query solution

Example solution

```
let
    Data = Excel.CurrentWorkbook()[[Name="TwoColumnExample"]][Content],

    ComparisonColumns = {"column1","column2"},

    AddRecord = Table.AddColumn(Data, "Record", each _),

    AddSortedComparisonValues = Table.AddColumn(
        AddRecord,
        "SortedComparisonValues",
        each List.Sort(
            Record.FieldValues(
                Record.SelectFields(_, ComparisonColumns)
            )
        ),

    GetReplacement = Table.Group(
        AddSortedComparisonValues,
        "SortedComparisonValues",
        {
            {"ReplacementRecord",
                each Table.First(Table.SelectColumns(_, ComparisonColumns))}
        },

    AddReplacementRecord = Table.AddColumn(
        AddSortedComparisonValues,
        "ReplacementRecord",
        each GetReplacement{
            [SortedComparisonValues]=[SortedComparisonValues]
        }[ReplacementRecord]
    ),

    RecordMerge = Table.AddColumn(
        AddReplacementRecord,
        "NewRecord",
        each [Record] & [ReplacementRecord]
    ),

    Result = Table.FromRecords(RecordMerge[NewRecord])
in
    Result
```

Example solution - create a sorted list of values from the comparison columns

```
let
    Data = Excel.CurrentWorkbook(){[Name="TwoColumnExample"]}[Content],
    ComparisonColumns = {"column1", "column2"},
    AddRecord = Table.AddColumn(Data, "Record", each _),
    AddSortedComparisonValues = Table.AddColumn(
        AddRecord,
        "SortedComparisonValues",
        each List.Sort(
            Record.FieldValues(
                Record.SelectFields(_, ComparisonColumns)
            )
        ),
    ),
```

Pull the source data in and define the columns we want to compare as a list

On each row, create a sorted list of the values in the comparison columns

Store each original record in a new column for use later

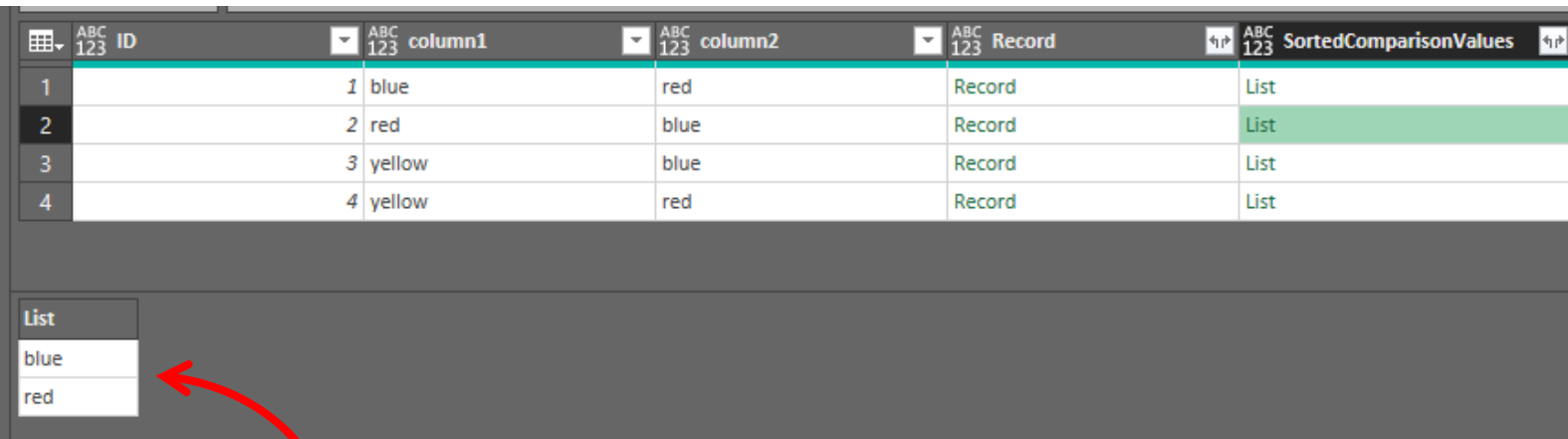
Example solution - create a sorted list of values from the comparison columns

```
let
    Source = Excel.CurrentWorkbook(){[Name="TwoColumnExample"]}[Content],

    ComparisonColumns = {"column1","column2"},

    AddRecord = Table.AddColumn(Data, "Record", each _),

    AddSortedComparisonValues = Table.AddColumn(
        AddRecord,
        "SortedComparisonValues",
        each List.Sort(
            Record.FieldValues(
                Record.SelectFields(_, ComparisonColumns)
            )
        ),
    ),
```



ABC 123	ID	ABC 123	column1	ABC 123	column2	ABC 123	Record	ABC 123	SortedComparisonValues
1		1	blue		red		Record		List
2		2	red		blue		Record		List
3		3	yellow		blue		Record		List
4		4	yellow		red		Record		List

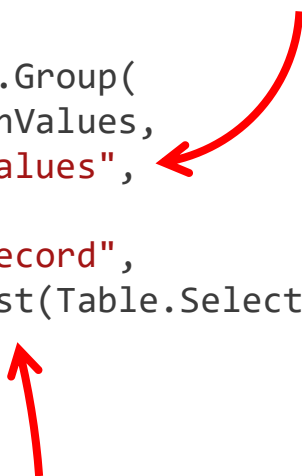
List
blue
red

Sorted list of values
from the comparison
columns

Example solution - Get a replacement record

Group by the column containing the sorted comparison values

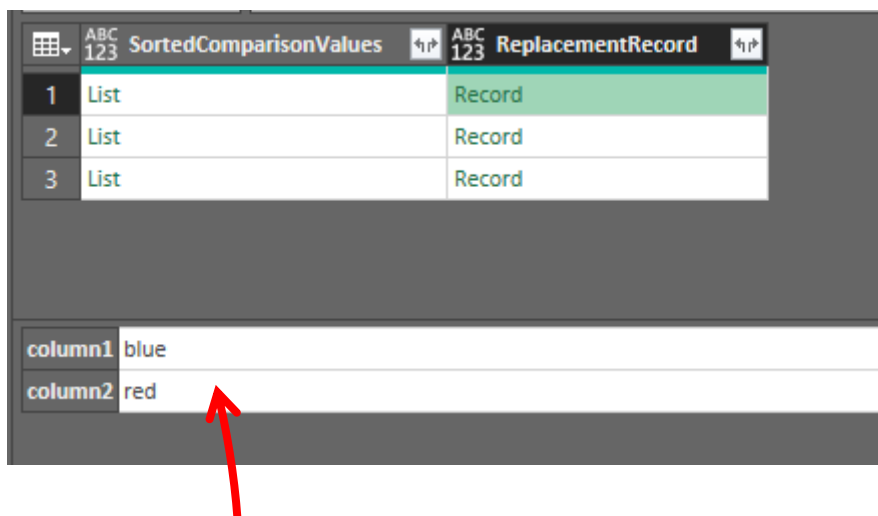
```
GetReplacement = Table.Group(  
    AddSortedComparisonValues,  
    "SortedComparisonValues",  
    {  
        {"ReplacementRecord",  
         each Table.First(Table.SelectColumns(_, ComparisonColumns))}  
    },  
    ),
```



From each group defined by the SortedComparisonValues, select the first record

Example solution - Get a replacement record

```
GetReplacement = Table.Group(  
    AddSortedComparisonValues,  
    "SortedComparisonValues",  
    {  
        {"ReplacementRecord",  
         each Table.First(Table.SelectColumns(_, ComparisonColumns))}  
    },  
    ),
```



	SortedComparisonValues	ReplacementRecord
1	List	Record
2	List	Record
3	List	Record

column1	column2
blue	red

The first 2 rows in the source data contain the same value in the SortedComparisonValues list.

The first record in the group defined by that list has column1="blue" and column2="red"

Example solution - Add the replacement record to the main table

Add a column to the main table called 'ReplacementRecord'



```
AddReplacementRecord = Table.AddColumn(  
    AddSortedComparisonValues,  
    "ReplacementRecord",  
    each GetReplacement{  
        [SortedComparisonValues=[SortedComparisonValues]]  
    }[ReplacementRecord]  
),
```



From the previous Group step, select the row whose SortedComparisonValues list column matches the same column of the current row of the main table.


Return the ReplacementRecord from the Group step into the new column.

Example solution - Add the replacement record to the main table

```
AddReplacementRecord = Table.AddColumn(  
    AddSortedComparisonValues,  
    "ReplacementRecord",  
    each GetReplacement(  
        [SortedComparisonValues]=[SortedComparisonValues]  
    )[ReplacementRecord]  
),
```

	ABC 123 ID	ABC 123 column1	ABC 123 column2	ABC 123 Record	ABC 123 SortedComparisonValues	ABC 123 ReplacementRecord
1	1	blue	red	Record	List	Record
2	2	red	blue	Record	List	Record
3	3	yellow	blue	Record	List	Record
4	4	yellow	red	Record	List	Record

column1	blue
column2	red




Because rows 1 and 2 have the same sorted list in the SortedComparisonValues column, they share the same row in the GetReplacement group step.

And because of this, the ReplacementRecord for row 2 shows the values from the first record of that group – i.e. the values from row 1.

Example solution - Merge the replacement with the original record

We merge the ReplacementRecord with the original Record added at the beginning. The effect is to overwrite the values in column1 and column2

```
RecordMerge = Table.AddColumn(  
    AddReplacementRecord,  
    "NewRecord",  
    each [Record] & [ReplacementRecord]  
)
```



```
Result = Table.FromRecords(RecordMerge[NewRecord])  
in  
Result
```

Finally we convert the new record into the output table



Example solution - Merge the replacement with the original record

```
RecordMerge = Table.AddColumn(  
    AddReplacementRecord,  
    "NewRecord",  
    each [Record] & [ReplacementRecord]  
)
```

```
Result = Table.FromRecords(RecordMerge[NewRecord])
```

in

Result

ABC 123	ID	ABC 123	column1	ABC 123	column2	ABC 123	Record	ABC 123	SortedComparisonValues	ABC 123	ReplacementRecord	ABC 123	NewRecord
1		1	blue		red		Record		List		Record		Record
2		2	red		blue		Record		List		Record		Record
3		3	yellow		blue		Record		List		Record		Record
4		4	yellow		red		Record		List		Record		Record

ID	2
column1	blue
column2	red

The NewRecord for row 2 now shows the selected values from row 1

</

Saved as a flexible function

Convert the query into a function and parameterize the replacement selection rule

By passing the ReplacementFunction as a function parameter, we can greatly increase the flexibility of this function

```
(  
  Data as table,  
  ComparisonColumns as list,  
  ReplacementFunction as function  
) as table =>  
let
```

```
// steps as shown previously
```

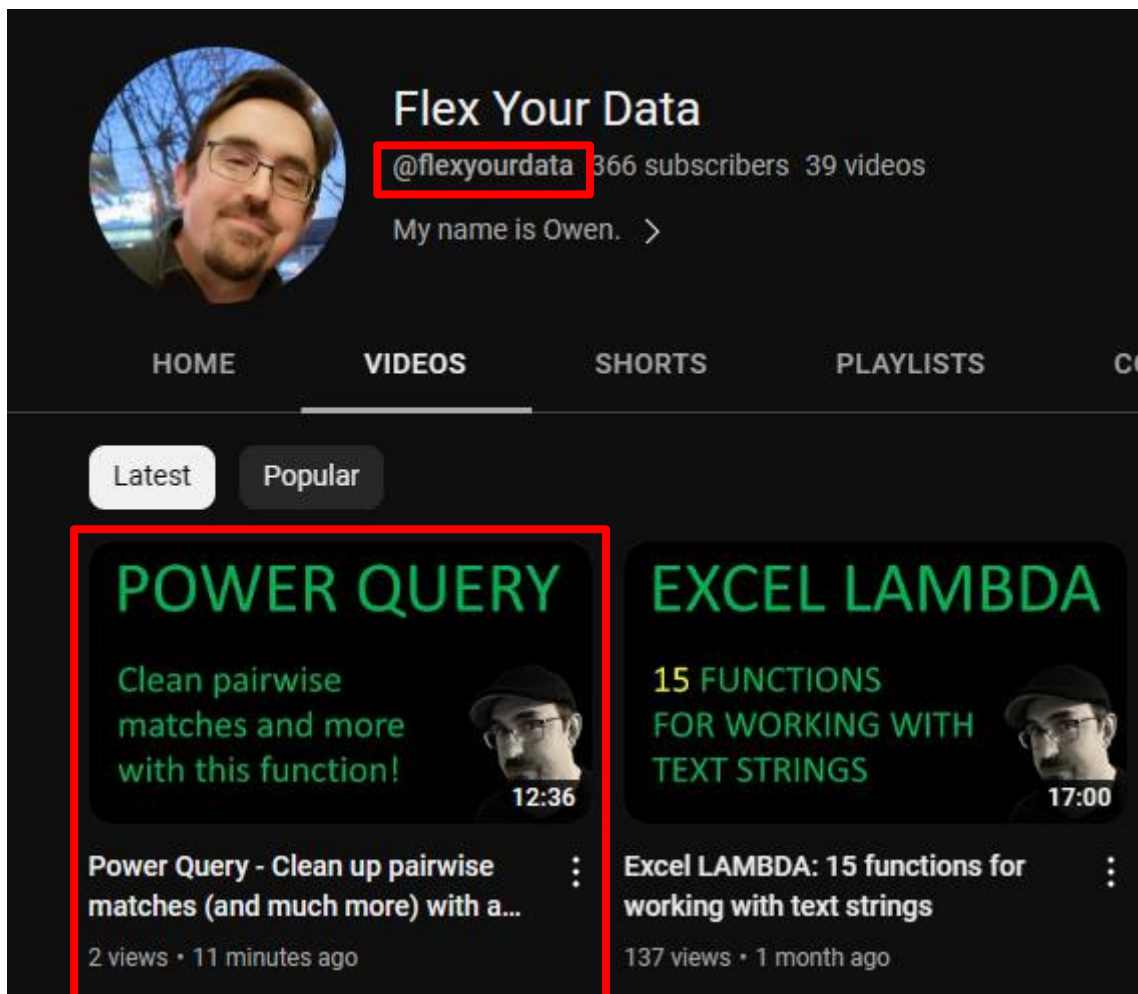
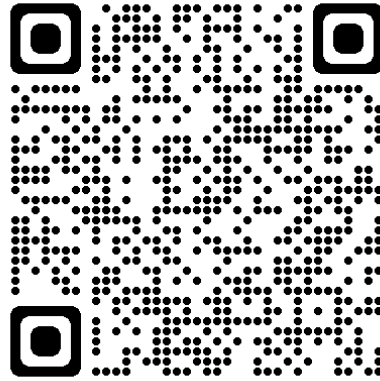
```
GetReplacement = Table.Group(  
  AddSortedComparisonValues,  
  "SortedComparisonValues",  
  {  
    {  
      "ReplacementRecord",  
      each ReplacementFunction(  
        Table.SelectColumns(_,  
          ComparisonColumns)  
      )  
    }  
  }  
,  
)
```

We replace Table.First with this arbitrary function, which can now be *any* function that accepts a table and returns a record

```
// steps as shown previously  
in  
Result
```


For more...

www.youtube.com/@flexyourdata



The image shows a screenshot of a YouTube channel page for "Flex Your Data". The channel name is "Flex Your Data" with the handle "@flexyourdata" highlighted in a red box. It has 366 subscribers and 39 videos. The bio says "My name is Owen. >". The navigation tabs are HOME, VIDEOS, SHORTS, PLAYLISTS, and COMMUNITY. The "VIDEOS" tab is selected. There are two filters: "Latest" and "Popular". Two video thumbnails are visible. The first video, titled "POWER QUERY", is highlighted with a red box. It has a description "Clean pairwise matches and more with this function!" and a duration of 12:36. The second video, titled "EXCEL LAMBDA", has a description "15 FUNCTIONS FOR WORKING WITH TEXT STRINGS" and a duration of 17:00.

Flex Your Data
@flexyourdata 366 subscribers 39 videos
My name is Owen. >

HOME VIDEOS SHORTS PLAYLISTS COMMUNITY

Latest Popular

POWER QUERY
Clean pairwise matches and more with this function!
12:36
Power Query - Clean up pairwise matches (and much more) with a...
2 views • 11 minutes ago

EXCEL LAMBDA
15 FUNCTIONS FOR WORKING WITH TEXT STRINGS
17:00
Excel LAMBDA: 15 functions for working with text strings
137 views • 1 month ago