[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

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

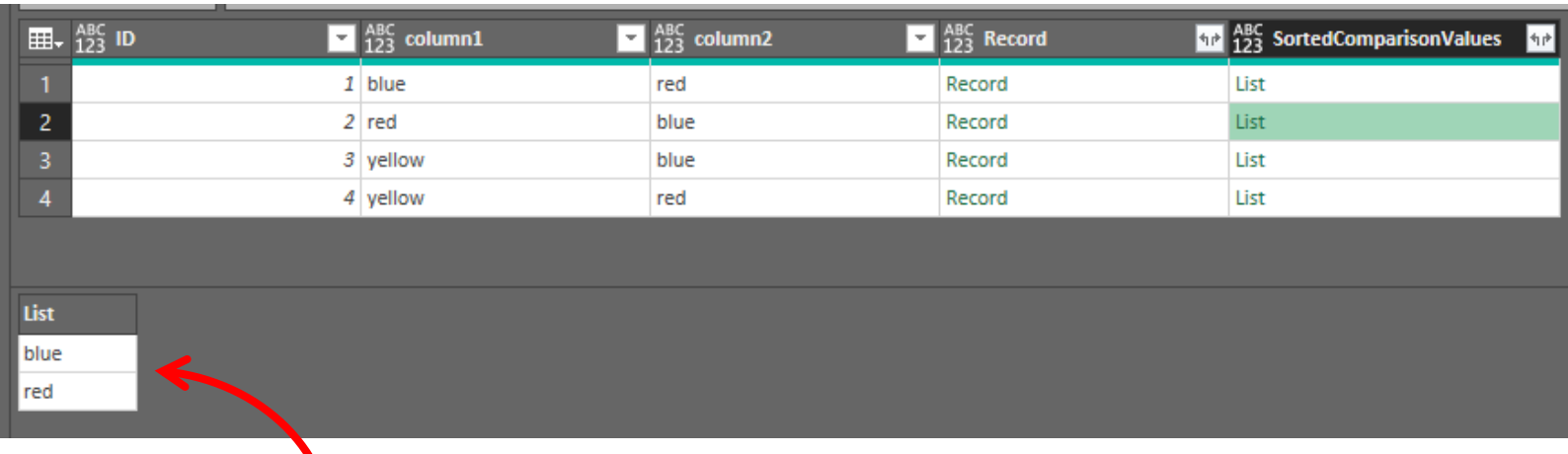
```
let
    Data = Excel.CurrentWorkbook(){[Name="TwoColumnExample"]}[Content],

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

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

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

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)
                )
            )
    ),
```



	ID	column1	column2	Record	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

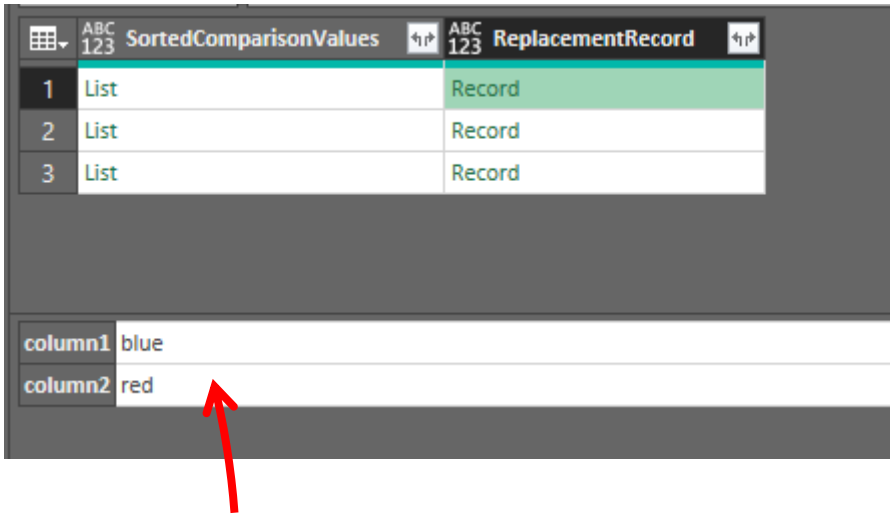
Group by the column containing the sorted comparison values

```
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))}
    }
),
```



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'

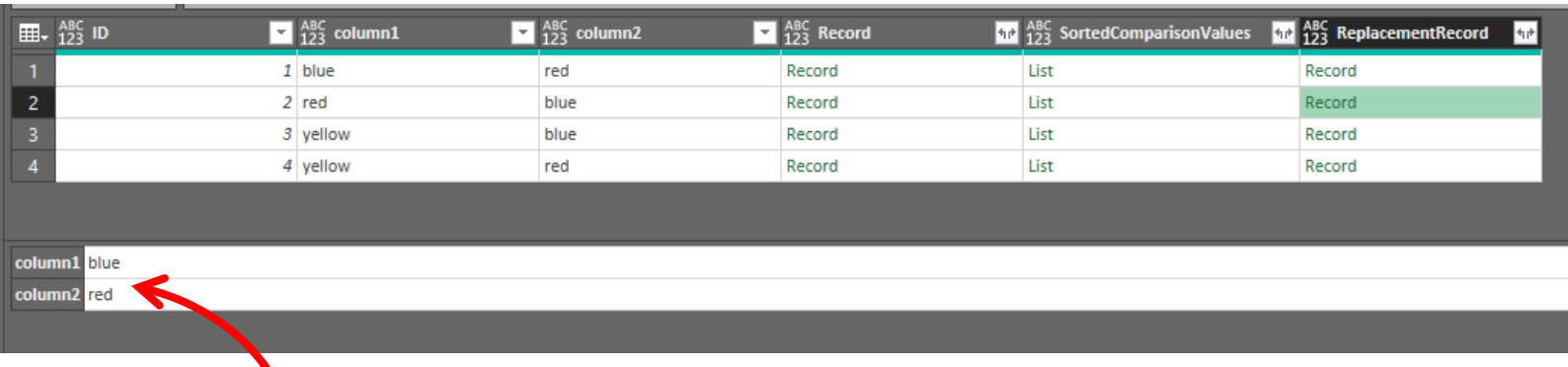
```
AddReplacementRecord = Table.AddColumn(
    AddSortedComparisonValues,
    "ReplacementRecord",
    each GetReplacement{
        [SortedComparisonValues=[SortedComparisonValues]]
        }[ReplacementRecord]
),
```

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]
),
```



	ID	column1	column2	Record	SortedComparisonValues	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

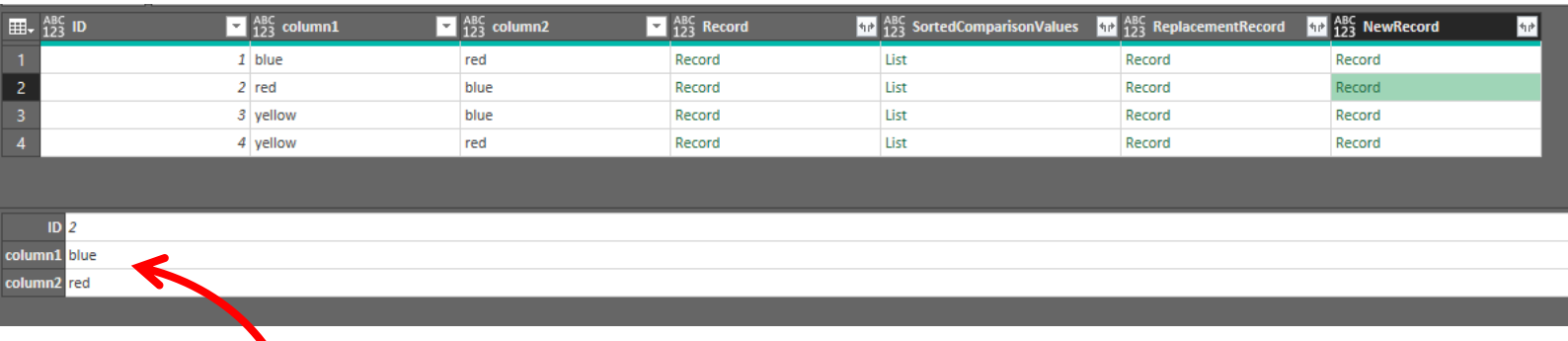
We merge the ReplacementRecord with the original Record added at the beginning. The effect is to overwrite the values in column1 and column2

```
    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

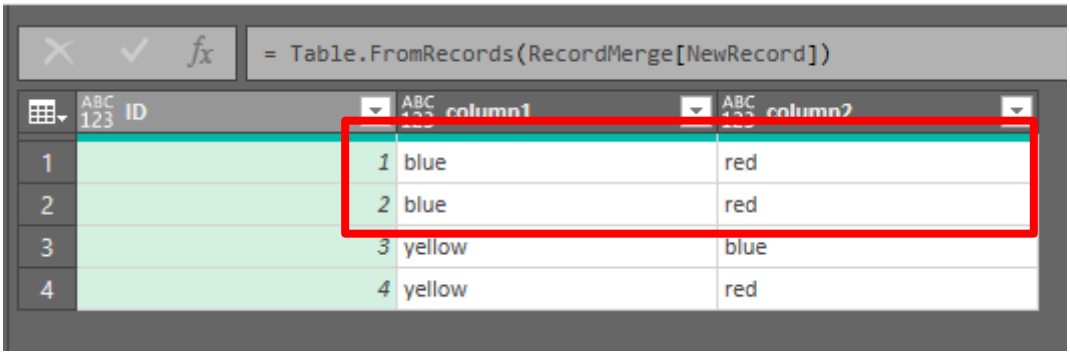
```
    RecordMerge = Table.AddColumn(
        AddReplacementRecord,
        "NewRecord",
        each [Record] & [ReplacementRecord]
    ),

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



	ID	column1	column2	Record	SortedComparisonValues	ReplacementRecord	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



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

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

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)
                    )
            }
        }
    ),

    // steps as shown previously
in
    Result
```

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

For more...

www.youtube.com/@flexyourdata