

[BITE-SIZED POWER QUERY]

SIMPLIFIED TABLE JOINS

Merge these tables to add Category to the Sales rows

	A	B	C	D	E	F	G
1							
2		Table name: Sales				Table name: Category	
3		Product	Date	Amount		Product	Category
4		Apples	29-Jan-23	2585		Apples	Pome Fruit
5		Lemons	29-Jan-23	4626		Lemons	Citrus Fruit
6		Lemons	28-Jan-23	4680		Oranges	Citrus Fruit
7		Oranges	29-Jan-23	4928		Pears	Pome Fruit
8		Pears	29-Jan-23	3083			
9		Pears	28-Jan-23	4599			
10		Pears	27-Jan-23	2803			
11							

UI Approach

1 UI Merge defaults to `Table.NestedJoin`, adding a new column containing a Table on each row

2 We then need to expand the table column. If we de-select "Prefix new columns", a suffix is added to the new column names

let

```
Source = Table.NestedJoin(
    Sales, {"Product"},
    Category, {"Product"},
    "Category", JoinKind.LeftOuter),
```

```
#"Expanded Category" = Table.ExpandTableColumn(
    Source, "Category",
    {"Category"}, {"Category.1"}),
```

```
#"Renamed Columns" = Table.RenameColumns(
    #"Expanded Category",
    {"Category.1", "Category"})
```

in

```
#"Renamed Columns"
```

3 We then need to rename the column to tidy up

Table.RenameColumns(#"Expanded Category",{{"Category.1", "Category"}})

<div><div></div><div>ABC 123</div></div>	<div>Product</div>	<div><div></div><div>ABC 123</div></div>	<div>Date</div>	<div><div></div><div>ABC 123</div></div>	<div>Amount</div>	<div><div></div><div>ABC 123</div></div>	<div>Category</div>
1	Apples		1/29/2023 12:00:00 AM		2585		Pome Fruit
2	Lemons		1/29/2023 12:00:00 AM		4626		Citrus Fruit
3	Lemons		1/28/2023 12:00:00 AM		4680		Citrus Fruit
4	Oranges		1/29/2023 12:00:00 AM		4928		Citrus Fruit
5	Pears		1/29/2023 12:00:00 AM		3083		Pome Fruit
6	Pears		1/28/2023 12:00:00 AM		4599		Pome Fruit
7	Pears		1/27/2023 12:00:00 AM		2803		Pome Fruit

Table.Join

Table.Join defaults to **JoinKind.Inner**. Since the join columns are identical **and** have identical names, only the non-join columns from the right-table are added to the left table

```
let
    Source = Table.Join(
        Sales, {"Product"},
        Category, {"Product"})
in
    Source
```



No prefixes or suffixes are added so long as columns added from the right table are not named the same as any columns in the left table

✕ ✓ fx		= Table.Join(Sales, {"Product"}, Category, {"Product"})			
	ABC 123 Product	ABC 123 Date	ABC 123 Amount	ABC 123 Category	
1	Apples	1/29/2023 12:00:00 AM	2585	Pome Fruit	
2	Lemons	1/29/2023 12:00:00 AM	4626	Citrus Fruit	
3	Lemons	1/28/2023 12:00:00 AM	4680	Citrus Fruit	
4	Oranges	1/29/2023 12:00:00 AM	4928	Citrus Fruit	
5	Pears	1/29/2023 12:00:00 AM	3083	Pome Fruit	
6	Pears	1/28/2023 12:00:00 AM	4599	Pome Fruit	
7	Pears	1/27/2023 12:00:00 AM	2803	Pome Fruit	



If we use **JoinKind.LeftOuter**, both Product columns must be in the output because the left Product column may have rows without a match in the right Product column



Duplicate column names will produce an error

CategoryWithDate



```
let
    Source = Table.Join(
        Sales, {"Product"},
        CategoryWithDate, {"Product"})
in
    Source
```

Because Date is in both tables,
the join produces an error









Table.Join(
 Sales, {"Product"},
 CategoryWithDate, {"Product"})

 Expression.Error: A join operation cannot result in a table with duplicate column names ("Date").
Details:
[Type]

We can prefix all the columns in the right table by using `Table.PrefixColumns`

We must remember to add the prefix to the right-table's join column(s) list in `Table.Join`

```
let
    Source = Table.Join(
        Sales, {"Product"},
        Table.PrefixColumns( CategoryWithDate,
            "Category"), {"Category.Product"})
in
    Source
```

	Product	Date	Amount	Category.Product	Category.Category	Category.Date
1	Apples	1/29/2023 12:00:00 AM	2585	Apples	Pome Fruit	1/1/2023
2	Lemons	1/29/2023 12:00:00 AM	4626	Lemons	Citrus Fruit	1/1/2023
3	Lemons	1/28/2023 12:00:00 AM	4630	Lemons	Citrus Fruit	1/1/2023
4	Oranges	1/29/2023 12:00:00 AM	4928	Oranges	Citrus Fruit	1/1/2023
5	Pears	1/29/2023 12:00:00 AM	3083	Pears	Pome Fruit	1/1/2023
6	Pears	1/28/2023 12:00:00 AM	4599	Pears	Pome Fruit	1/1/2023
7	Pears	1/27/2023 12:00:00 AM	2803	Pears	Pome Fruit	1/1/2023

Unfortunately, we now have the `Category.Product` column in the result as well as the `Product` column

The category is also prefixed, even though it doesn't need to be.

Custom function "SimpleJoin"

```
SimpleJoin =  
(  
  LeftTable as table,  
  JoinColumns as list,  
  RightTable as table,  
  optional JoinColumnsRight as list,  
  optional DuplicateColumnPrefix as text,  
  optional JoinKind as number  
) =>  
let (etc...)
```

- ☑ Only specify join columns once if join columns are the same in both tables
- ☑ Automatically handle column renaming in case of duplicate column names
- ☑ Only rename duplicate columns, leave non-duplicate columns alone
- ☑ Allow an optional column prefix for renamed duplicate columns (default "New ")
- ☑ Specify JoinKind when called (default Inner)

Table name: SalesMulti

Product	Date	Amount
Apples	1-Jan-23	2585
Lemons	1-Jan-23	4626
Lemons	1-Feb-23	4680
Oranges	1-Feb-23	4928
Pears	1-Jan-23	3083
Pears	1-Feb-23	4599
Pears	1-Dec-22	2803

Table name: MonthPrice

Product	Date	Price
Apples	1-Jan-23	24.57
Lemons	1-Jan-23	15.03
Oranges	1-Jan-23	22.14
Pears	1-Jan-23	23.16
Apples	1-Feb-23	18.24
Lemons	1-Feb-23	17.81
Oranges	1-Feb-23	23.41
Pears	1-Feb-23	24.35
Apples	1-Dec-22	18.88
Lemons	1-Dec-22	18.95
Oranges	1-Dec-22	18.14
Pears	1-Dec-22	22.68

A simple join with join columns only specified once. Adds Price to left table



</


Table name: SalesMulti

Product	Date	Amount
Apples	1-Jan-23	2585
Lemons	1-Jan-23	4626
Lemons	1-Feb-23	4680
Oranges	1-Feb-23	4928
Pears	1-Jan-23	3083
Pears	1-Feb-23	4599
Pears	1-Dec-22	2803

Table name: MonthPrice

Product	Date	Price
Apples	1-Jan-23	24.57
Lemons	1-Jan-23	15.03
Oranges	1-Jan-23	22.14
Pears	1-Jan-23	23.16
Apples	1-Feb-23	18.24
Lemons	1-Feb-23	17.81
Oranges	1-Feb-23	23.41
Pears	1-Feb-23	24.35
Apples	1-Dec-22	18.88
Lemons	1-Dec-22	18.95
Oranges	1-Dec-22	18.14
Pears	1-Dec-22	22.68

If JoinKind is not inner, the join columns are renamed according to the supplied prefix. Price is not renamed because it's not a duplicate



= SimpleJoin(SalesMulti, {"Product","Date"}, MonthPrice, null, "MonthPrice", JoinKind.LeftOuter)

	ABC 123 Product	ABC 123 Date	ABC 123 Amount	ABC 123 MonthPrice Product	ABC 123 MonthPrice Date	ABC 123 Price
1	Apples	1/1/2023 12:00:00 AM	2585	Apples	1/1/2023 12:00:00 AM	24.57
2	Lemons	1/1/2023 12:00:00 AM	4626	Lemons	1/1/2023 12:00:00 AM	15.03
3	Lemons	2/1/2023 12:00:00 AM	4680	Lemons	2/1/2023 12:00:00 AM	17.81
4	Oranges	2/1/2023 12:00:00 AM	4928	Oranges	2/1/2023 12:00:00 AM	23.41
5	Pears	1/1/2023 12:00:00 AM	3083	Pears	1/1/2023 12:00:00 AM	23.16
6	Pears	2/1/2023 12:00:00 AM	4599	Pears	2/1/2023 12:00:00 AM	24.35
7	Pears	12/1/2022 12:00:00 AM	2803	Pears	12/1/2022 12:00:00 AM	22.68

Table name: SalesMulti

Product	Date	Amount
Apples	1-Jan-23	2585
Lemons	1-Jan-23	4626
Lemons	1-Feb-23	4680
Oranges	1-Feb-23	4928
Pears	1-Jan-23	3083
Pears	1-Feb-23	4599
Pears	1-Dec-22	2803

Table name: MonthPrice

Product	Date	Price	Amount
Apples	1-Jan-23	24.57	0
Lemons	1-Jan-23	15.03	0
Oranges	1-Jan-23	22.14	0
Pears	1-Jan-23	23.16	0
Apples	1-Feb-23	18.24	0
Lemons	1-Feb-23	17.81	0
Oranges	1-Feb-23	23.41	0
Pears	1-Feb-23	24.35	0
Apples	1-Dec-22	18.88	0
Lemons	1-Dec-22	18.95	0
Oranges	1-Dec-22	18.14	0
Pears	1-Dec-22	22.68	0

Amount in the right table is renamed because it's a duplicate. Price is not. Because JoinKind is not specified, default is Inner and Product and Date from the right table are not returned



= SimpleJoin(SalesMulti, {"Product", "Date"}, MonthPrice, null, "MonthPrice")

	ABC 123 Product	ABC 123 Date	ABC 123 Amount	ABC 123 Price	ABC 123 MonthPrice Amount
1	Apples	1/1/2023 12:00:00 AM	2585	24.57	0
2	Lemons	1/1/2023 12:00:00 AM	4626	15.03	0
3	Lemons	2/1/2023 12:00:00 AM	4680	17.81	0
4	Oranges	2/1/2023 12:00:00 AM	4928	23.41	0
5	Pears	1/1/2023 12:00:00 AM	3083	23.16	0
6	Pears	2/1/2023 12:00:00 AM	4599	24.35	0
7	Pears	12/1/2022 12:00:00 AM	2803	22.68	0

For a walkthrough of how
SimpleJoin works and to grab
the code:

<https://www.youtube.com/@flexyourdata>