

## Отчет работы в DAX

Формирование справочника выходных и рабочих дней за 2021 год.

Запрос в Power Query расширенном редакторе. Подключение по API запросу к сайту Консультант Плюс:

Параметр начала сбора данных:

Имя

Описание

☒ Требуется

Тип

Предлагаемые значения

Текущее значение

Создание таблицы-примера, по подобию которой будет подтягиваться информация с сайта. Новый пустой запрос и вставляем в расширенном редакторе код:

let

```
TextYearParam = Text.From(YearParam)&(if YearParam=2020 then "b" else  
""),
```

```
Source =  
Lines.FromBinary(Web.Contents("http://www.consultant.ru/law/ref/calendar  
/proizvodstvennyye",[RelativePath = Text.From(TextYearParam) & "/?"])),
```

```
#"Converted to Table" = Table.FromList(Source, Splitter.SplitByNothing(),  
null, null, ExtraValues.Error),
```

```
#"Filtered Rows" = Table.SelectRows("Converted to Table", each  
Text.StartsWith([Column1], "#(tab)#(tab)#(tab)<td class="),
```

```

    #"Replaced Value" = Table.ReplaceValue("#Filtered
Rows", "</tr><tr>", "", Replacer.ReplaceText, {"Column1"}),

    #"Added Index" = Table.AddIndexColumn("#Replaced Value", "#Month",
1, 1),

    #"Added Prefix" = Table.TransformColumns("#Added Index",
{{"Column1", each Text.Split(_, "</td><td class="), type text}}),

    #"Expanded Custom" = Table.ExpandListColumn("#Added Prefix",
"Column1"),

    #"Filtered Rows1" = Table.SelectRows("#Expanded Custom", each not
Text.Contains([Column1], "inactively")),

    #"Replaced Value1" = Table.ReplaceValue("#Filtered
Rows1", "#(tab)#(tab)#(tab)<td
class=", "", Replacer.ReplaceText, {"Column1"}),

    #"Split Column by Delimiter" = Table.SplitColumn("#Replaced Value1",
"Column1", Splitter.SplitTextByEachDelimiter({">"}, QuoteStyle.Csv, false),
{"Day type", "Day"}),

    #"Extracted Text Before Delimiter" = Table.TransformColumns("#Split
Column by Delimiter", {"Day", each Text.BeforeDelimiter(_, "<"), type
text}}),

    #"Changed Type" = Table.TransformColumnTypes("#Extracted Text Before
Delimiter", {"Day", Int64.Type}),

    #"Added Custom" = Table.AddColumn("#Changed Type", "Year", each
YearParam, Int64.Type),

    #"Added Custom1" = Table.AddColumn("#Added Custom", "Date", each
#date([Year], [#"#Month"], [Day]), type date),

    #"Inserted Month Name" = Table.AddColumn("#Added Custom1",
"Month", each Date.MonthName([Date]), type text),

    #"Inserted Day Name" = Table.AddColumn("#Inserted Month Name",
"Day name", each Date.DayOfWeekName([Date]), type text)
in

    #"Inserted Day Name"

```

Создание функции сбора данных

Создать функцию

Введите имя новой функции.

Имя функции

Параметры: YearParam

OKОтмена

Создание еще два параметра «старт» и «конец» списка годов аналогично тому, как создавали параметр сбора данных:

Запросы [5]

- CalendarYear [3]
  - ExampleOfFunction
  - YearParam (2020)
  - CalendarYear
- Другие запросы [2]
  - StartYear (2010)
  - EndYear (2025)

Управление параметрами

Создать

YearParam
StartYear
EndYear

Имя

StartYear

Описание

☒ Требуется

Тип

Десятичное число

Предлагаемые значения

Любое значение

Текущее значение

2010

## Загрузка данных

Вводим в параметр StartYear 2020 год, а параметр EndYear можно оставить равным 2025.

Запросы [5]

- CalendarYear [3]
  - ExampleOfFunction
  - YearParam (2020)
  - CalendarYear
- Другие запросы [2]
  - StartYear (2020)
  - EndYear (2025)

Текущее значение

2020

Управление параметром

Чтобы вызвать функцию за период от StartYear до EndYear вставляем следующий код:

let

```
Список = {StartYear..EndYear},  
  
#"Преобразовано в таблицу" = Table.FromList(Список,  
Splitter.SplitByNothing(), null, null, ExtraValues.Error),  
  
#"Добавлен пользовательский объект" =  
Table.AddColumn(#"Преобразовано в таблицу", "Календарь", each  
CalendarYear([Column1])),  
  
#"Удаленные столбцы" = Table.RemoveColumns(#"Добавлен  
пользовательский объект",{"Column1"}),  
  
#"Развернутый элемент Календарь" =  
Table.ExpandTableColumn(#"Удаленные столбцы", "Календарь", {"Day  
type", "Day", "#Month", "Year", "Date", "Month", "Day name"}, {"Day type",  
"Day", "#Month", "Year", "Date", "Month", "Day name"})
```

in

```
#"Развернутый элемент Календарь"  
Календарь за весь период.
```

let

```
Список = {StartYear..EndYear},  
  
#"Преобразовано в таблицу" = Table.FromList(Список,  
Splitter.SplitByNothing(), null, null, ExtraValues.Error),  
  
#"Добавлен пользовательский объект" =  
Table.AddColumn(#"Преобразовано в таблицу", "Календарь", each  
CalendarYear([Column1])),  
  
#"Удаленные столбцы" = Table.RemoveColumns(#"Добавлен  
пользовательский объект",{"Column1"}),  
  
#"Развернутый элемент Календарь" =  
Table.ExpandTableColumn(#"Удаленные столбцы", "Календарь", {"Day  
type", "Day", "#Month", "Year", "Date", "Month", "Day name"}, {"Day type",  
"Day", "#Month", "Year", "Date", "Month", "Day name"}),
```

```
#"Строки с примененным фильтром" =  
Table.SelectRows(#"Развернутый элемент Календарь", each ([Year] =  
2021)),
```

```
#"Измененный тип" = Table.TransformColumnTypes(#"Строки с  
примененным фильтром",{{"Date", type date}})
```

in

#"Измененный тип"

### Расчетные таблицы:

Количество заказов =

SUMMARIZE(FILTER(orders,[cancel\_reason\_id]=5),orders[customer\_id],  
"Количество заказов", COUNT(orders[order\_num]))

Цена на товары = SUMMARIZE(FILTER(order\_rows, 'order\_rows'[Статус  
заказа] = "Доставлен"), order\_rows[order\_num], "Цена на товар",  
SUM(order\_rows[Цена заказа]))

### Расчетные столбцы:

праздники/выходные = IF('Calendar'[Day type]=BLANK() || 'Calendar'[Day  
type] = "preholiday","Рабочий", "Выходной")

### Меры:

Количество выходных дней =

CALCULATE(COUNT('Calendar'[праздники/выходные]),'Calendar'[праздник  
и/выходные]="Выходной")

Количество рабочих дней =

CALCULATE(COUNT('Calendar'[праздники/выходные]),'Calendar'[праздник  
и/выходные]="Рабочий")

Доля выходных дней = ([Количество выходных дней]/365)

Доля рабочих дней = ([Количество рабочих дней]/365)

Кол всех заказов = DISTINCTCOUNT(orders[order\_num])

Количество доставленных заказов =

CALCULATE(DISTINCTCOUNT(orders[order\_num]),'orders'[cancel\_reason\_id  
]=5)

Купон = RELATED(discounts[discount])

Скидка = RELATED('Ozon-Nov-Sale'[Скидка])

Купон на бй заказ = RELATED('Купон на бй заказ'[coupon])

Диван/Кресло = IF(LEFT('sku'[sku\_name],1)="Д", "Диван", "Кресло")

Доставлен =

CALCULATE(DISTINCTCOUNT(order\_rows[order\_num]),'order\_rows'[Статус  
заказа]="Доставлен")

```

Отменен =
CALCULATE(DISTINCTCOUNT(order_rows[order_num]),'order_rows'[Статус
заказа]="Отменен")

Возраст = 2021 - YEAR ('customers'[birth_date])

Возрастная_группа =
SWITCH(
    TRUE(),
    'customers'[Возраст] >= 20 && 'customers'[Возраст] <= 29, "20-29",
    'customers'[Возраст] >= 30 && 'customers'[Возраст] <= 39, "30-39",
    'customers'[Возраст] >= 40 && 'customers'[Возраст] <= 49, "40-49",
    'customers'[Возраст] >= 50 && 'customers'[Возраст] <= 59, "50-59",
    'customers'[Возраст] >= 60 && 'customers'[Возраст] <= 69, "60-69",
    'customers'[Возраст] >= 70 && 'customers'[Возраст] <= 79, "70-79",
    'customers'[Возраст] >= 80 && 'customers'[Возраст] <= 89, "80-89",
    "Менее 20"
)

Пол = IF(RIGHT([fio],1)="ч","Мужчина", "Женщина")

ВЫРУЧКА = ([Цена на товар]-[Купон])*(1-[Скидка])

Выручка 2 = ([Цена на товар]-([Купон]+[Купон 250]))*(1-[Скидка])

Город = IF(LEFT('Цена на товары'[order_num],1)="М", "Москва", "Санкт-
Петербург")

```