



Microsoft Power BI

Шевцов Василий Викторович,
директор ДИТ РУДН, shevtsov_vv@rudn.university

Правила форматирования

Необязательные правила форматирования

ИмяФункции()

ИмяФункции(Параметр1)

ИмяФункции(Параметр1;Параметр2)

ИмяФункции(Параметр1;Параметр2;...;ПараметрN)

Параметры функций

- значения (число, текст, дата)
- 'Таблица'
- 'Таблица'[Столбец]
- [Мера]
- вложенная функция

Необязательные правила форматирования

Мера = Функция1(Функция2(Функция4([Мера]));Функция3(Функция5();
Функция6('Таблица'; 'Таблица'[Столбец])))

Функция1 – 2 параметра

Функция2 – 1 параметр

Функция3 – 2 параметра

Функция4 – 1 параметр

Функция5 – 0 параметров

Функция6 – 2 параметра

Необязательные правила форматирования

1. Имя столбца записывается полностью 'Таблица'[Столбец]
2. Имя меры записывается кратко [Мера]
3. Имя вычисляемого объекта на первой строке, код на остальных

Сумма Прибыли =

...

код DAX

...

Необязательные правила форматирования

4. Если используются операторы, пробелы до и после операторов

$$= 5 * 5 / 2$$

Необязательные правила форматирования

5. Если в формуле более одной функции, каждая функция на новой строке

=

Функция 1

+ Функция 2

Необязательные правила форматирования

6. Если в функции более одного параметра, каждый параметр на новой строке

=

```
Функция (  
    Параметр 1;  
    Параметр 2  
)
```


Необязательные правила форматирования

7. Если в функции 1 параметр и он состоит из вложенной функции, то разбиваем DAX код на строки

```
=  
Функция 1(  
    Функция 2()  
)
```

Необязательные правила форматирования

8. Если в функции 1 параметр и он состоит просто из какого-либо значения или ссылки на таблицу, столбец и меру, то код пишем в 1 строку

= Функция (Значение)

Необязательные правила форматирования

9. Если в формуле есть операторы и код должен быть разбит на строки, то каждая новая строка начинается с оператора

=

'Таблица'[Столбец]

+ [Мера 1]

- [Мера 2]

Необязательные правила форматирования

10. Открывающая круглая скобка параметров всегда находится в той же строке, где и вызов функции

```
= Функция (  
    Параметр 1;  
    Параметр 2  
)
```

Необязательные правила форматирования

11. Закрывающая круглая скобка параметров всегда находится в новой строке, под первой буквой вызова функции

=

Функция (
 Параметр 1;
 Параметр 2
)

Необязательные правила форматирования

12. Параметры, находящиеся в новых строках пишутся с отступом в 4 пробела (1 табуляция) от первой буквы имени функции

```
= Функция (  
    Параметр 1;  
    Параметр 2  
)
```

Необязательные правила форматирования

13. Точка с запятой, разделяющая параметры находится на той же строке, что и предыдущий параметр

```
= Функция (  
    Параметр 1;  
    Параметр 2  
)
```

Необязательные правила форматирования

Горячие клавиши:

- ALT+ENTER — перенос курсора на начало новой строки
- SHIFT+ENTER — перенос курсора на новую строку с автоматическим отступом в 4 символа

Необязательные правила форматирования

Мера = Функция1(Функция2(Функция4([Мера]));Функция3(Функция5();
Функция6('Таблица'; 'Таблица'[Столбец])))

Мера =
Функция1(
 Функция2(
 Функция4([Мера])
);
 Функция3(
 Функция5();
 Функция6(
 'Таблица';
 'Таблица'[Столбец]
)
)
)

Функция1 – 2 параметра
Функция2 – 1 параметр
Функция3 – 2 параметра
Функция4 – 1 параметр
Функция5 – 0 параметров
Функция6 – 2 параметра

VAR и RETURN

VAR и RETURN

VAR Name 1 = Код Переменной 1

VAR Name 2 = Код Переменной 2

RETURN

Name 1 * Name 2

Где:

- Name — имя создаваемой переменной
- Код Переменной — любая DAX формула, характеризующая саму переменную
- VAR — служебное слово (специальная функция), всегда предшествующая объявлению имени переменной при ее создании
- RETURN — служебное слово (специальная функция), оканчивающая создание (объявление) блока переменных, после него идет основной код формулы
- Name 1 * Name 2 — основной код формулы (меры, вычисляемого столбца или таблицы)

Переменные:

- визуально упрощают сложный код формулы в DAX
- уменьшают время расчета сложного кода, который имеет повторяемые участки, так как переменная рассчитывается только 1 раз и ей можно заменить повторяемые участки кода в основной формуле
- «запоминают контекст строки», в котором переменная была вычислена

VAR и RETURN

```
1 Столбец =  
2   VAR f1=[2015]  
3   VAR f2=[2016]  
4   RETURN  
5   DIVIDE(f1;f2)
```



2015	2016	Столбец
1550	1553	0,998068254990341
1226	1221	1,004095004095
1397	1390	1,00503597122302
2333	2335	0,999143468950749
1030	1023	1,00684261974585
1010	1014	0,996055226824458
651	648	1,00462962962963
1120	1123	0,997328584149599
1156	1156	1

В сложном коде могут встречаться одни и те же части, повторяемые несколько раз. И использование переменных позволяет не только сократить длину кода, написав этот код один раз в переменной, и далее просто ссылаться на имя этой переменной, так и сократить время вычисления этой большой формулы. Так как повторяемый код прописанный переменной вычисляется только один раз, а далее, при вызове имени переменной из основного кода, просто уже будет возвращаться вычисленный ранее результат переменной.

IF()

Синтаксис

IF (Условие; Выражение 1; Выражение 2)

Где:

- Условие — логическое условие, результат которого равен либо значению TRUE, либо — FALSE
- Выражение 1 — выражение, которое вступит в работу, если результат проверки условия будет равен TRUE
- Выражение 2 — выражение, которое вступит в работу, если результат проверки условия будет равен FALSE

✕ ✓		div = [col1]/[col2]	
col1	col2	div	
10	1	10	
20	0	∞	
30	3	10	
40		∞	
50	5	10	
60	6	10	

✕ ✓		div = IF(ISERROR([col1]/[col2]);BLANK();[col1]/[col2])	
col1	col2	div	
10	1	10	
20	0		
30	3	10	
40			
50	5	10	
60	6	10	

Применение для управления интерфейсом

Визуальный элемент ограничивает область данных

Изменение видимости области данных доступно в мере

В зависимости от ограничения видимости данных меняется алгоритм вычислений

Год
2005
2010
2011
2012
2013
2014
2015
2016



- ☐ 2005
- ☐ 2010
- ☐ 2011
- ☐ 2012
- ☒ 2013
- ☐ 2014
- ☐ 2015
- ☐ 2016

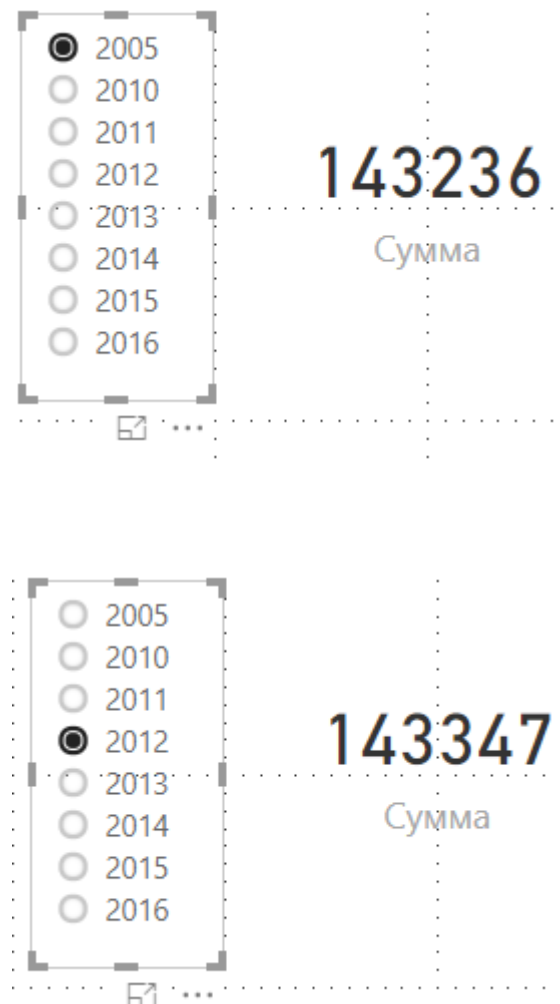


```
IF(  
VALUES('Год'[Год])=2005;  
SUM('Численность населения'[2005]);  
SUM('Численность населения'[2010])  
)
```

Таблицы 'Год' и 'Численность населения' не связаны между собой!

Реализация

```
1 Сумма = IF(  
2   VALUES('Год'[Год])=2005;  
3   SUM('Численность населения'[2005]);  
4   IF(  
5     VALUES('Год'[Год])=2010;  
6     SUM('Численность населения'[2010]);  
7     IF(  
8       VALUES('Год'[Год])=2011;  
9       SUM('Численность населения'[2011]);  
10      IF(  
11        VALUES('Год'[Год])=2012;  
12        SUM('Численность населения'[2012]);  
13        IF(  
14          VALUES('Год'[Год])=2013;  
15          SUM('Численность населения'[2013]);  
16          IF(  
17            VALUES('Год'[Год])=2014;  
18            SUM('Численность населения'[2014]);  
19            IF(  
20              VALUES('Год'[Год])=2015;  
21              SUM('Численность населения'[2015]);  
22              SUM('Численность населения'[2016])  
23            )  
24          )  
25        )  
26      )  
27    )  
28  )  
29 )
```



Switch()

Синтаксис

SWITCH (Выражение; Значение 1; Результат 1; Значение 2; Результат 2; Значение N; Результат N; Иначе)

Где:

- Выражение — вычисляемое выражение, результат которого сравнивается с заранее подготовленными значениями Значение — заранее подготовленное значение
- Результат — заранее подготовленный результат, который возвратится, если результат вычисления выражения будет равен соответствующему заранее подготовленному значению
- Иначе — заранее подготовленный результат, который возвратится, если результат вычисления выражения не будет равен ни одному заранее подготовленному значению

Switch()

```
Месяц = SWITCH (
    [col2];
    1; "Январь";
    2; "Февраль";
    3; "Март";
    4; "Апрель";
    5; "Май";
    6; "Июнь";
    7; "Июль";
    8; "Август";
    9; "Сентябрь";
    10; "Октябрь";
    11; "Ноябрь";
    "Декабрь" )
```

col1	col2	div	Месяц
10	1	10	Январь
20	0		Декабрь
30	3	10	Март
40			Декабрь
50	5	10	Май
60	6	10	Июнь

Switch()

```
Квартал = SWITCH(  
    TRUE();  
    [col2]<=3;"1 квартал";  
    [col2]<=6;"2 квартал";  
    [col2]<=9;"3 квартал";  
    [col2]<=12;"4 квартал"  
)
```

col1	col2	div	Месяц	Квартал	
Данные	10	1	10	Январь	1 квартал
	20	0		Декабрь	1 квартал
	30	3	10	Март	1 квартал
	40			Декабрь	1 квартал
	50	5	10	Май	2 квартал
	60	6	10	Июнь	2 квартал

Динамический выбор меры

Создание таблицы



	Название	Сортировка	*
1	Алгоритм 1	1	
2	Алгоритм 2	2	
3	Алгоритм 3	3	
*			

Имя:

Загрузить

Изменить

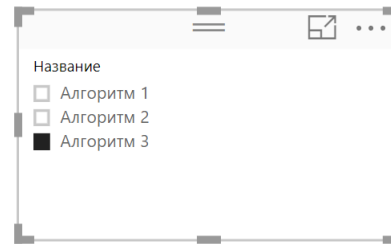
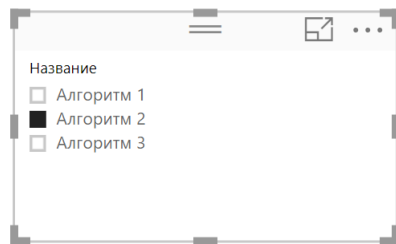
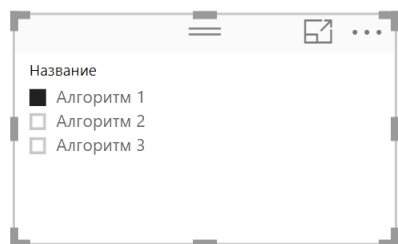
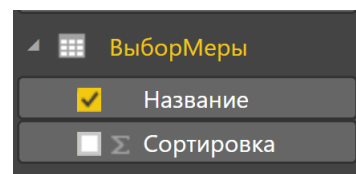
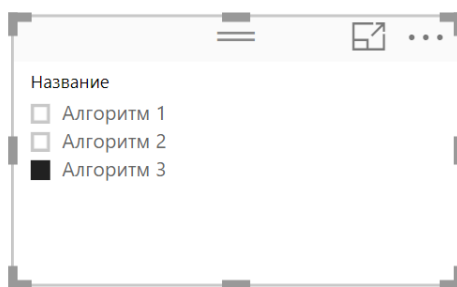
Отмена

col1	col2	div	Месяц	Квартал
10	1	10	Январь	1 квартал
20	0		Декабрь	1 квартал
30	3	10	Март	1 квартал
40			Декабрь	1 квартал
50	5	10	Май	2 квартал
60	6	10	Июнь	2 квартал

Мера1 = SUM('tab1'[col1])
Мера2 = AVERAGE('tab1'[col1])
Мера3 = MIN('tab1'[col1])

Динамический выбор меры

```
Мера_выбор = SWITCH(  
    TRUE();  
    VALUES('ВыборМеры'[Название])="Алгоритм 1";[Мера1];  
    VALUES('ВыборМеры'[Название])="Алгоритм 2";[Мера2];  
    VALUES('ВыборМеры'[Название])="Алгоритм 3";[Мера3];  
    BLANK()  
)
```



210,00

Мера_выбор

35,00

Мера_выбор

10,00

Мера_выбор



CONTAINS

Синтаксис

Возвращает значение true, если значения для всех столбцов, на которые имеются ссылки, существуют или содержатся в этих столбцах; в противном случае возвращает значение false.

```
CONTAINS(  
<table>, <columnName>, <value>[, <columnName>, <value>]...  
)
```

Параметры

table - Любое выражение DAX, возвращающее таблицу данных.

columnName - Имя существующего столбца в стандартном синтаксисе DAX. Не может быть выражением.

value - Любое выражение DAX, возвращающее скалярное значение, поиск которого выполняется в столбце columnName. Выражение вычисляется только один раз перед его передачей в список аргументов.

Пример

Мера = CONTAINS('tab1';tab1[col1];20)

col1	col2	div	Месяц	Квартал
10	1	10	Январь	1 квартал
20	0		Декабрь	1 квартал
30	3	10	Март	1 квартал
40			Декабрь	1 квартал
50	5	10	Май	2 квартал
60	6	10	Июнь	2 квартал

True
Мера

☒ ☐ Мера = CONTAINS('tab1';tab1[col1];20;tab1[col2];0;tab1[Месяц];"Декабрь")

col1	col2	div	Месяц	Квартал
10	1	10	Январь	1 квартал
20	0		Декабрь	1 квартал
30	3	10	Март	1 квартал
40			Декабрь	1 квартал
50	5	10	Май	2 квартал
60	6	10	Июнь	2 квартал

True
Мера

☐ ☒ Мера = CONTAINS('tab1';tab1[col1];20;tab1[col2];0;tab1[Месяц];"Март")

col1	col2	div	Месяц	Квартал
10	1	10	Январь	1 квартал
20	0		Декабрь	1 квартал
30	3	10	Март	1 квартал
40			Декабрь	1 квартал
50	5	10	Май	2 квартал
60	6	10	Июнь	2 квартал

False
Мера

ISONORAFTER

ISONORAFTER

ISONORAFTER(

<scalar_expression>, <scalar_expression>[, sort_order [,
<scalar_expression>, <scalar_expression>[, sort_order]]...)

- скалярное выражение - Любое выражение, возвращающее скалярное значение, например ссылку на столбец, целое число или строковое значение. Обычно первый параметр является ссылкой на столбец, а второй-скалярным значением.
- порядок сортировки (необязательно) - порядок сортировки столбца. Может быть восходящим (ASC) или нисходящим (DEC). По умолчанию порядок сортировки - по возрастанию.

Функция ISONORAFTER возвращает TRUE, если любое из сравнений имеет значение TRUE, в противном случае возвращает FALSE.

ISONORAFTER

На основе порядка сортировки первый параметр сравнивается со вторым параметром. Если порядок сортировки по возрастанию, сравнение выполняется с первым параметром, который больше или равен второму параметру. Если порядок сортировки по убыванию, сравнение выполняется по второму параметру, меньшему или равному первому параметру.

= ISONORAFTER (2 , 0 , DESC , 2 , 1 , DESC):

эта формула DAX возвращает FALSE, потому что и 0 , и 1 не больше 2.

= ISONORAFTER (2 , 5 , DESC , 2 , 1 , DESC):

эта формула DAX возвращает TRUE, потому что 5 больше 2.

ISONORAFTER(0;1;ASC)=FALSE

ISONORAFTER(0;1;DESC)=TRUE

ISONORAFTER(0;1;DESC;0;1; DESC) =TRUE

ISONORAFTER(0;1;DESC;0;1;ASC)=TRUE

ISONORAFTER(0;1;DESC;1;0;DESC)=TRUE

ISONORAFTER(1;0;DESC;1;0;DESC)=FALSE

KEEPFILTERS

Различное поведение в CALCULATE()

```
EVALUATE ROW(  
    "$$ in WA"  
    , CALCULATE('Internet Sales'[Internet Total Sales]  
        , 'Geography'[State Province Code]="WA"  
    )  
    , "$$ in WA and OR"  
    , CALCULATE('Internet Sales'[Internet Total Sales]  
        , 'Geography'[State Province Code]="WA"  
        || 'Geography'[State Province Code]="OR"  
    )  
    , "$$ in WA and BC"  
    , CALCULATE('Internet Sales'[Internet Total Sales]  
        , 'Geography'[State Province Code]="WA"  
        || 'Geography'[State Province Code]="BC"  
    )  
    , "$$ in WA and OR ??"  
    , CALCULATE(  
        CALCULATE('Internet Sales'[Internet Total Sales]  
            , 'Geography'[State Province Code]="WA"  
            || 'Geography'[State Province Code]="OR"  
        )  
        , 'Geography'[State Province Code]="WA"  
        || 'Geography'[State Province Code]="BC"  
    )  
    , "$$ in WA !!"  
    , CALCULATE(  
        CALCULATE('Internet Sales'[Internet Total Sales]  
            , KEEPFILTERS('Geography'[State Province Code]="WA"  
                || 'Geography'[State Province Code]="OR"  
            )  
        )  
    , 'Geography'[State Province Code]="WA"  
    || 'Geography'[State Province Code]="BC"  
    )  
)
```

Индивидуальные фильтры
для каждой строки

Внешний фильтр,
накладываемый на все
строки

Различное поведение фильтров в CALCULATE()

```
EVALUATE ROW(  
    "$$ in WA"  
    , CALCULATE('Internet Sales'[Internet Total Sales]  
        , 'Geography'[State Province Code]="WA"  
    )  
    , "$$ in WA and OR"  
    , CALCULATE('Internet Sales'[Internet Total Sales]  
        , 'Geography'[State Province Code]="WA"  
        || 'Geography'[State Province Code]="OR"  
    )  
    , "$$ in WA and BC"  
    , CALCULATE('Internet Sales'[Internet Total Sales]  
        , 'Geography'[State Province Code]="WA"  
        || 'Geography'[State Province Code]="BC"  
    )  
    , "$$ in WA and OR ??"  
    , CALCULATE(  
        CALCULATE('Internet Sales'[Internet Total Sales]  
            , 'Geography'[State Province Code]="WA"  
            || 'Geography'[State Province Code]="OR"  
        )  
        , 'Geography'[State Province Code]="WA"  
        || 'Geography'[State Province Code]="BC"  
    )  
    , "$$ in WA !!"  
    , CALCULATE(  
        CALCULATE('Internet Sales'[Internet Total Sales]  
            , KEEPFILTERS('Geography'[State Province Code]="WA"  
                || 'Geography'[State Province Code]="OR"  
            )  
        , 'Geography'[State Province Code]="WA"  
        || 'Geography'[State Province Code]="BC"  
    )  
)
```

Column	Value
[\$\$ in WA]	\$ 2,467,248.34
[\$\$ in WA and OR]	\$ 3,638,239.88
[\$\$ in WA and BC]	\$ 4,422,588.44
[\$\$ in WA and OR ??]	\$ 3,638,239.88
[\$\$ in WA !!]	\$ 2,467,248.34

Совпадают первый и последний расчет.

В первом случае CALCULATE переопределяет внешние фильтры ("WA"||"BC") и заменяет их собственными фильтрами ("WA"). (также в 2,3,4 случаях. в 4 случае фильтры переопределяются на вложенные во второй CALCULATE).

В последнем случае внутри CALCULATE применяются совпадающие фильтры KEEPFILTERS ("WA"||"OR") и внешний ("WA"||"BC").
В итоге применяется только фильтр ("WA").

HASONEVALUE

Возвращает значение TRUE, если контекст для columnName был отфильтрован до одного конкретного значения. В противном случае - значение FALSE.

HASONEVALUE(<columnName>)

Эквивалентное выражение для HASONEVALUE() —
 $\text{COUNTROWS}(\text{VALUES}(\text{<columnName>})) = 1.$

Страна	Округ	Область	Год	Численность
Российская Федерация	Центральный федеральный округ	Белгородская область	2005	1512
Российская Федерация	Центральный федеральный округ	Белгородская область	2010	1532
Российская Федерация	Центральный федеральный округ	Белгородская область	2011	1536
Российская Федерация	Центральный федеральный округ	Белгородская область	2012	1541
Российская Федерация	Центральный федеральный округ	Белгородская область	2013	1544
Российская Федерация	Центральный федеральный округ	Белгородская область	2014	1548
Российская Федерация	Центральный федеральный округ	Белгородская область	2015	1550
Российская Федерация	Центральный федеральный округ	Белгородская область	2016	1553
Российская Федерация	Центральный федеральный округ	Белгородская область	2017	1550
Российская Федерация	Центральный федеральный округ	Белгородская область	2018	1548
Российская Федерация	Центральный федеральный округ	Брянская область	2005	1327
Российская Федерация	Центральный федеральный округ	Брянская область	2010	1275
Российская Федерация	Центральный федеральный округ	Брянская область	2011	1264

147 тыс.

Численность

Год

- ☐ 2005
- ☐ 2010
- ☐ 2011
- ☐ 2012
- ☐ 2013
- ☐ 2014
- ☐ 2015
- ☒ 2016
- ☐ 2017
- ☐ 2018

42

1 млн

Численность

Год

- ☐ 2005
- ☐ 2010
- ☐ 2011
- ☐ 2012
- ☐ 2013
- ☐ 2014
- ☐ 2015
- ☐ 2016
- ☐ 2017
- ☐ 2018

```
Мера = IF(  
    HASONEVALUE(  
        'Численность населения'[Год]  
    );  
    SUM('Численность населения'[Численность]);  
    "выберите год"  
)
```

146,80
ТЫС.

Мера

Год

- ☐ 2005
- ☐ 2010
- ☐ 2011
- ☐ 2012
- ☐ 2013
- ☐ 2014
- ☐ 2015
- ☒ 2016
- ☐ 2017
- ☐ 2018

выберите
год

Мера

Год

- ☐ 2005
- ☐ 2010
- ☐ 2011
- ☐ 2012
- ☐ 2013
- ☐ 2014
- ☐ 2015
- ☐ 2016
- ☐ 2017
- ☐ 2018

IN {...}

IN {...}

```
1 RedOrBlack Sales OR :=  
2 CALCULATE (  
3     [Sales Amount],  
4     Products[Color] = "Red" || Products[Color] = "Black"  
5 )
```



```
1 RedOrBlack Sales IN :=  
2 CALCULATE (  
3     [Sales Amount],  
4     Products[Color] IN { "Red", "Black" }  
5 )
```

```
1 Other Colors 3 :=  
2 CALCULATE (  
3     [Sales Amount],  
4     NOT ( Products[Color] IN VALUES ( Products[Color] ) )  
5 )
```

IN {...}

Оператор IN позволяет сравнивать несколько столбцов в одной операции

```
1  ( 'Calendar'[Year], 'Calendar'[MonthName] )
2      IN {
3          ( 2005, "December" ),
4          ( 2006, "January" )
5      }
```

```
1  NY 2007 ALL :=
2  CALCULATE (
3      [Sales Amount],
4      FILTER (
5          ALL ( 'Calendar' ),
6          ( 'Calendar'[Year], 'Calendar'[MonthName] )
7              IN {
8                  ( 2006, "December" ),
9                  ( 2007, "January" )
10             }
11      )
12  )
```

TOPN()

Таблица =

```
TOPN(  
    3;  
    SUMMARIZE(  
        'Численность населения';  
        'Численность населения' [Округ];  
        "ЧисленностьОкруга";  
        SUM('Численность населения' [Численность])  
    );  
    [ЧисленностьОкруга];  
    DESC  
)
```

Округ ▾	ЧисленностьОкруга ▾
Сибирский федеральный округ	172379
Центральный федеральный округ	388544
Приволжский федеральный округ	297621

+ CONCATENATEX

Мера 2 =

CONCATENATEX(
TOPN(

3;

SUMMARIZE(

'Численность населения';

'Численность населения'[Округ];

"ЧисленностьОкруга";

SUM('Численность населения'[Численность])

);

[ЧисленностьОкруга];

DESC

);

[Округ] & " " & [ЧисленностьОкруга];

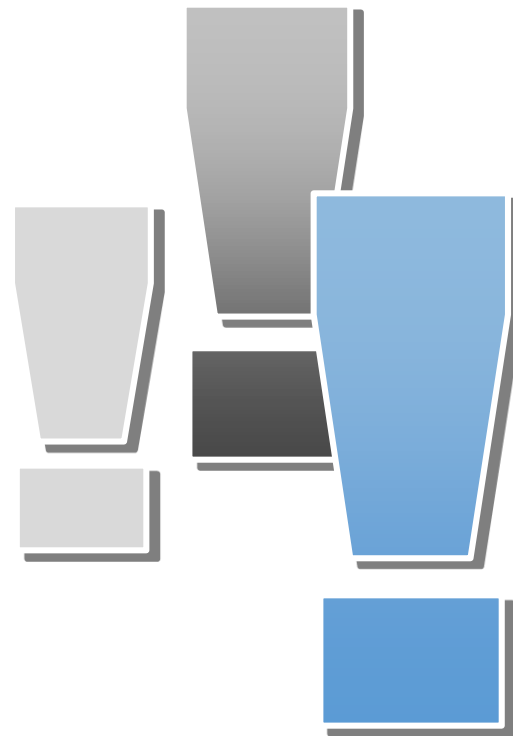
"," "

)

Сибирский федеральный округ
172379; Центральный
федеральный округ 388544;
Приволжский федеральный округ
297621

Мера 2

Спасибо за внимание!



Шевцов Василий Викторович

shevtsov_vv@rudn.university
+7(903)144-53-57