

## Открываю ранее созданный отчет (в части1) упр\_Основные\_функции\_DAX.pbix

`Dates[Date]=Calendar (firstdate (Sales[Date]), LASTDATE(Sales[Date]))` - созданная таблица на основании таблицы 'Sales', которые между собой нужно связать "один ко многим"

`Dates = CALENDAR (FIRSTDATE ('Продажи' [OrderDate]), LASTDATE ('Продажи' [OrderDate]))`

Выберите взаимосвязанные таблицы и столбцы.

Из таблицы

Dates

Date
01.01.2025 0:00:00
02.01.2025 0:00:00
03.01.2025 0:00:00

В таблицу

Продажи

Category	id заказа	OrderDate	OrderYear	Product	ProductDescri...	Quantity
Бумага	70001	1 января 202...	2025	картон а4	картон а4-Бу...	2
Канцелярия	70001	1 января 202...	2025	ручка	ручка-Канце...	3
Бумага	70001	1 января 202...	2025	блокнот	блокнот-Бум...	6

Количество элементов

Один ко многим (1:\*)

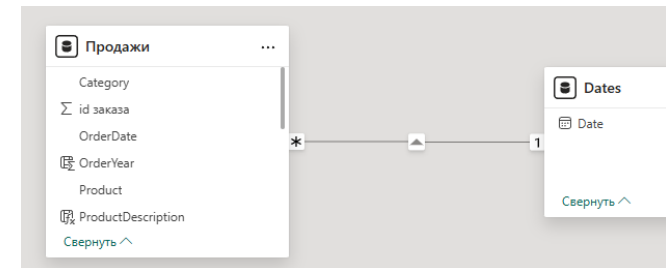
Направление перекрестной фильтрации

Простое

☒ Активировать связь

☐ Применить фильтр безопасности в обоих направлениях

☐ Предполагать целостность данных



### Упр. - Расчет нарастающего итога (Running Total) с учетом фильтров

- Обычный расчет `CALCULATE(SUM(Sales[Amount]), FILTER(ALL(Dates), Dates[Date] <= MAX(Dates[Date])))` не учитывает фильтры на странице отчета.
- Решение:  
Running Total = `CALCULATE ( SUM ( Sales[Amount] ), FILTER ( ALLSELECTED ( Dates ), Dates[Date] <= MAX ( Dates[Date] ) ) )`

Running Total = `CALCULATE (SUM ('Продажи' [TotalCost]), FILTER (ALLSELECTED ('Dates'), 'Dates' [Date] <= MAX ('Dates' [Date]))`



### Упр. - Ранжирование с учетом повторений

- Функция `RANKX` может возвращать одинаковые ранги для одинаковых значений. Для получения уникального ранга даже при повторениях:  
Unique Rank = `RANKX ( ALL ( Products[ProductKey] ), CALCULATE ( SUM ( Sales[SalesAmount] ) ), , DESC, DENSE )`

Unique Rank = `RANKX (ALL ('Продажи' [ProductDescription]), CALCULATE (SUM ('Продажи' [TotalCost])), , DESC, Dense)`

Структура		Форматирование		Свойства		Вычисления	
1 Unique Rank = RANKX(ALL('Продажи'[ProductDescription]),CALCULATE(SUM('Продажи'[TotalCost])),,DESC, Dense)						Данные	
Представление таблицы	Category	Quantity	UnitPrice	OrderDate	TotalCost	OrderYear	ProductDes
70001	картон а4	Бумага	2	178	1 января 2025 г.	356	2025
70001	ручка	Канцелярия	3	163	1 января 2025 г.	489	2025
70001	блокнот	Бумага	6	186	1 января 2025 г.	1116	2025
70001	тетрадь а4	Бумага	2	211	1 января 2025 г.	422	2025
70001	тетрадь а5	Бумага	13	148	1 января 2025 г.	1924	2025
70002	блокнот	Бумага	4	186	1 января 2025 г.	744	2025
70002	тетрадь а4	Бумага	1	211	1 января 2025 г.	211	2025
70002	тетрадь а5	Бумага	5	148	1 января 2025 г.	740	2025
70002	файл	Канцелярия	7	157	1 января 2025 г.	1099	2025
70002	скрепки	Канцелярия	14	248	1 января 2025 г.	3472	2025
70003	карандаш	Канцелярия	4	134	1 января 2025 г.	536	2025
70003	клей	Канцелярия	7	133	1 января 2025 г.	931	2025
70004	картон а4	Бумага	12	178	1 января 2025 г.	2136	2025
70005	ручка	Канцелярия	9	163	1 января 2025 г.	1467	2025
70006	блокнот	Бумага	2	186	1 января 2025 г.	372	2025
70007	тетрадь а4	Бумага	13	211	1 января 2025 г.	2743	2025
70008	тетрадь а5	Бумага	10	148	1 января 2025 г.	1480	2025
70009	ластик	Канцелярия	15	120	1 января 2025 г.	1800	2025
70010	точилка	Канцелярия	12	214	2 февраля 2025 г.	2568	2025

### Упр. - Вычисление разницы между текущей строкой и предыдущей

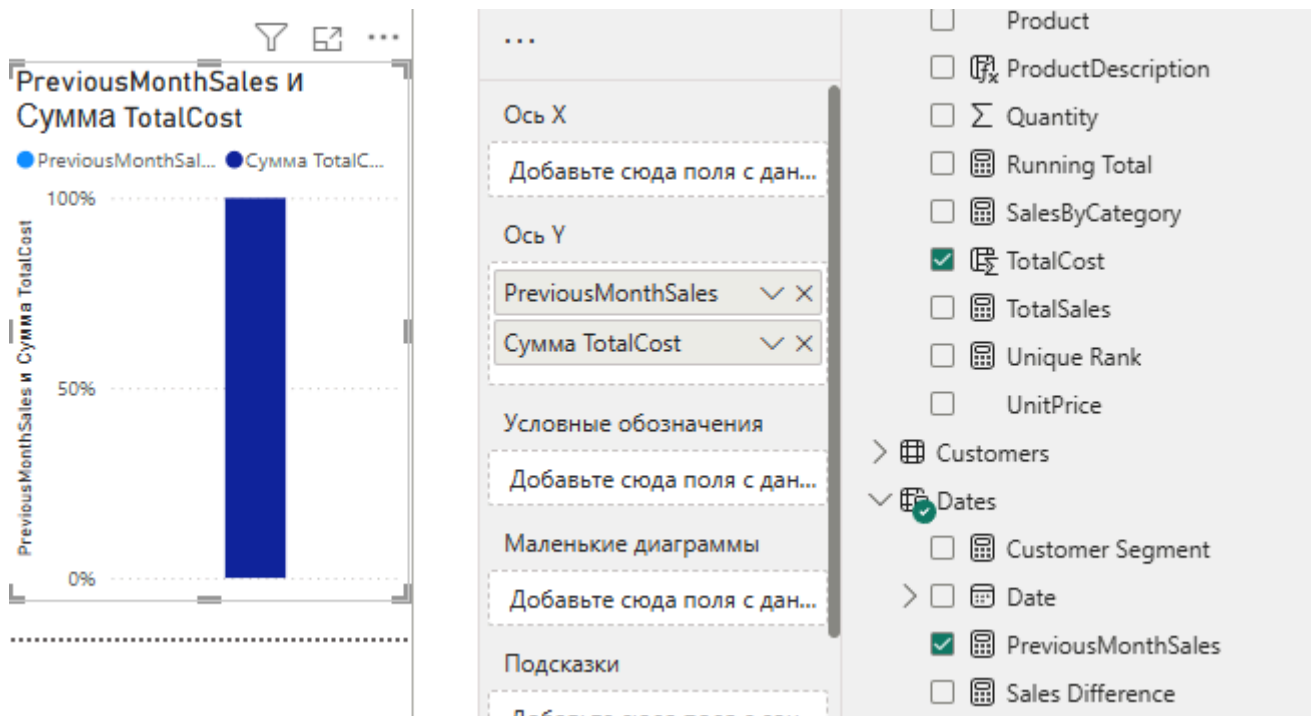
- Для анализа трендов полезно сравнивать значения в соседних строках:

Previous Month Sales = CALCULATE ( SUM ( Sales[SalesAmount] ), PREVIOUSMONTH ( Dates[Date] ) )

Sales Difference = [SalesAmount] - [Previous Month Sales]

PreviousMonthSales = CALCULATE(SUM('Продажи'[TotalCost]), PREVIOUSMONTH('Dates'[Date]) )

Sales Difference = [TotalSales] - [PreviousMonthSales]



### Упр. - Динамическая сегментация клиентов на основе перцентилей

Customer Segment = VAR Percentile = PERCENTILEX.INC ( Customers, Customers[TotalPurchases], 0.75 ) RETURN IF ( [TotalPurchases] >= Percentile, "High Value", "Regular" )

посчитаю товары

TotalPurchase = COUNTX(RELATEDTABLE('Детали заказа'), 'Детали заказа'[ProductID])

Структура		Форматирование		Сортировка	Группы	Связ
1 TotalPurchase = COUNTX(RELATEDTABLE('Детали заказа'), 'Детали заказа' [ProductID])						
ProductID	ProductName	Category	UnitPrice	Столбец		
999001	картон а4	Бумага	178	5		
999002	ручка	Канцелярия	163	5		
999003	блокнот	Бумага	186	6		
999004	тетрадь а4	Бумага	211	5		

Purchase = SUM(Products[TotalPurchase])

сегментирую товары

Сегмент = VAR Percentile = PERCENTILEX.INC ('Products', 'Products'[TotalPurchase], 0.75 ) RETURN IF ( [Purchase] >= Percentile, "High Value", "Regular" )

## Упр. - Использование переменных для повышения читаемости и производительности

Profit Margin =

VAR TotalRevenue = SUM ( Sales[Revenue] )

VAR TotalCost = SUM ( Sales[Cost] )

RETURN DIVIDE ( TotalRevenue - TotalCost, TotalRevenue, 0 )

### РЕШЕНИЕ

внесем в таблицу Products несколько вычисляемых полей с выручкой и себестоимостью

Revenue = 'Products'[UnitPrice] \* 'Products'[TotalPurchase]

1 Revenue = 'Products'[UnitPrice] * 'Products'[TotalPurchase]						
ProductID	ProductName	Category	UnitPrice	TotalPurchase	Revenue	
999001	картон а4	Бумага	178	5	890	
999002	ручка	Канцелярия	163	5	815	
999003	блокнот	Бумага	186	6	1116	
999004	тетрадь а4	Бумага	211	5	1055	

Costs = 'Products'[UnitPrice] \* 0.5 \* 'Products'[TotalPurchase]

(себестоимость просто для примера такая - одинаковая для всех товаров

ProductID	ProductName	Category	UnitPrice	TotalPurchase	Revenue	Costs
999001	картон а4	Бумага	178	5	890	445
999002	ручка	Канцелярия	163	5	815	407,5
999003	блокнот	Бумага	186	6	1116	558
999004	тетрадь а4	Бумага	211	5	1055	527,5

теперь рассчитаем маржу

Profit Margin =

VAR TotalRevenue = SUM(Products[Revenue])

VAR TotalCost = SUM ( Products[Costs] )

RETURN DIVIDE ( TotalRevenue - TotalCost, TotalRevenue, 0 )

Структура

Форматирование

Свойства

✕

✓

1 Profit Margin =

2 VAR TotalRevenue = SUM(Products[Revenue])

3 VAR TotalCost = SUM (Products[Costs])

4 RETURN DIVIDE ( TotalRevenue - TotalCost, TotalRevenue, 0 )

5

ProductID

ProductName

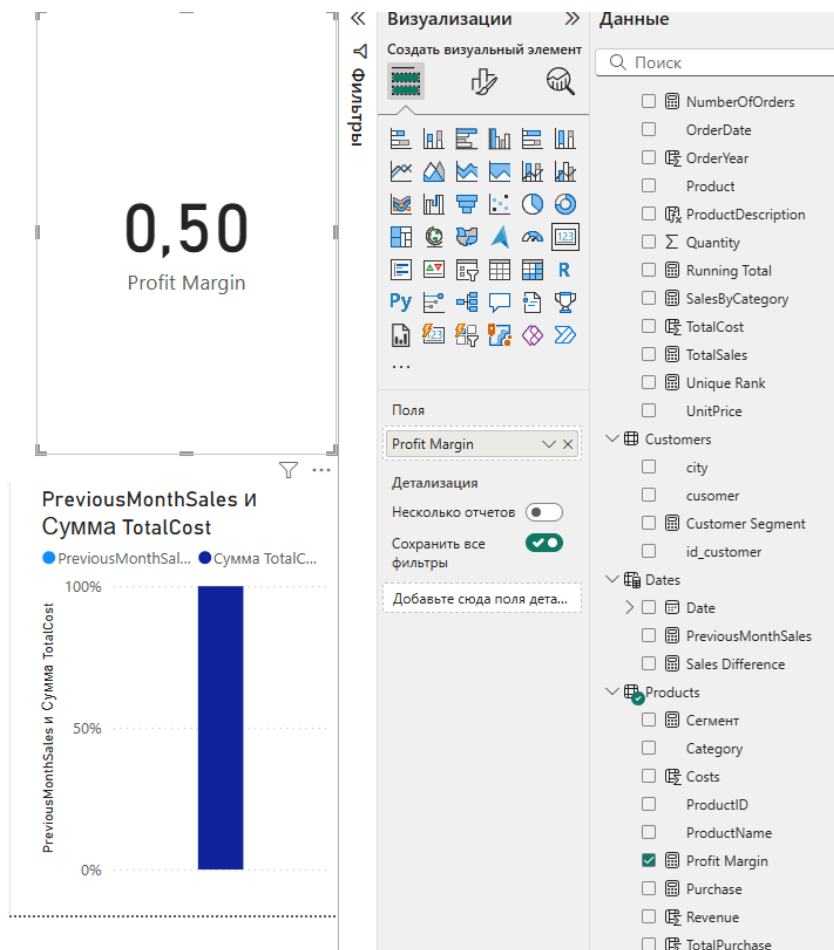
Category

UnitPrice

TotalPurchase

Revenue

Cos



## Упр. - Работа с временным интеллектом (Time Intelligence)

Расчет продаж за предыдущий год:

Previous Year Sales = CALCULATE([TotalSales], SAMEPERIODLASTYEAR(Dates[Date]))

в таблице только заказы за январь и февраль

Previous Year Sales = CALCULATE([TotalSales], SAMEPERIODLASTYEAR('Dates'[Date]))

Структура	Форматирование	Свойства
1 Previous Year Sales = CALCULATE([TotalSales], SAMEPERIODLASTYEAR('Dates'[Date]))		
id заказа	Product	Category
70001	картон а4	Бумага
	Quantity	UnitPrice
2	170	1 января 2025
	OrderDate	TotalCost
		256

Скользящее среднее за 3 месяца:

Moving Average = CALCULATE ( AVERAGEX ( DATESINPERIOD ( Dates[Date], LASTDATE ( Dates[Date] ), -3, MONTH ), [TotalSales] ) )

сделаю за 3 дня

Moving Average = CALCULATE ( AVERAGEX ( DATESINPERIOD ( 'Dates'[Date], LASTDATE ( 'Dates'[Date] ), 3, DAY ), [TotalSales] ) )

## Упр. - Использование X-функций для итераций

Top 5 Products = TOPN ( 5, SUMMARIZE ( Sales, Products[ProductName], "Total Sales", SUM ( Sales[SalesAmount] ) ), [Total Sales], DESC )

Top 5 Products = TOPN(5, SUMMARIZE('Продажи', 'Продажи'[Product], "Total Sales", SUM(Products[Revenue])), [Total Sales], DESC)

1 Top 5 Products = TOPN(5, SUMMARIZE('Продажи', 'Продажи'[Product], "Total Sales", SUM(Products[Revenue])), [Total Sales], DESC)	
Product	Total Sales
точилка	1284
блокнот	1116
тетрадь а4	1055
скрепки	992
картон а4	890