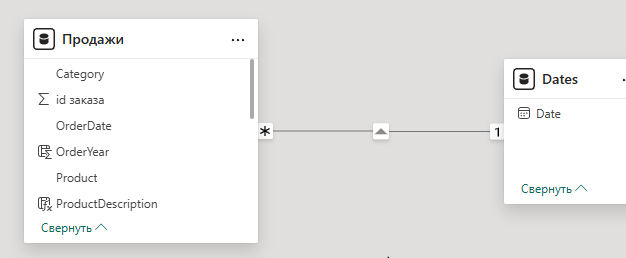
### **Открываю ранее созданный отчет (в части1)**

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]))

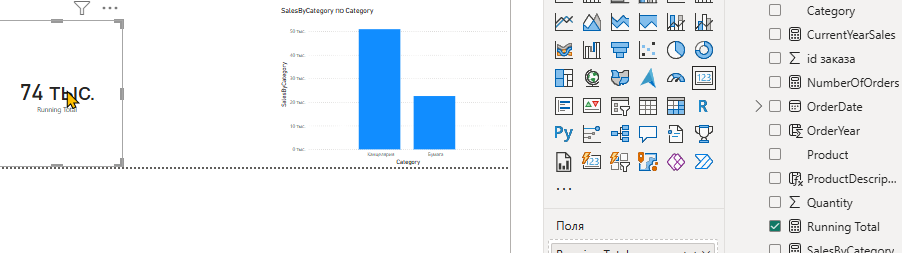


### **Упр. - Расчет нарастающего итога (Running Total) с учетом фильтров**

* 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]))

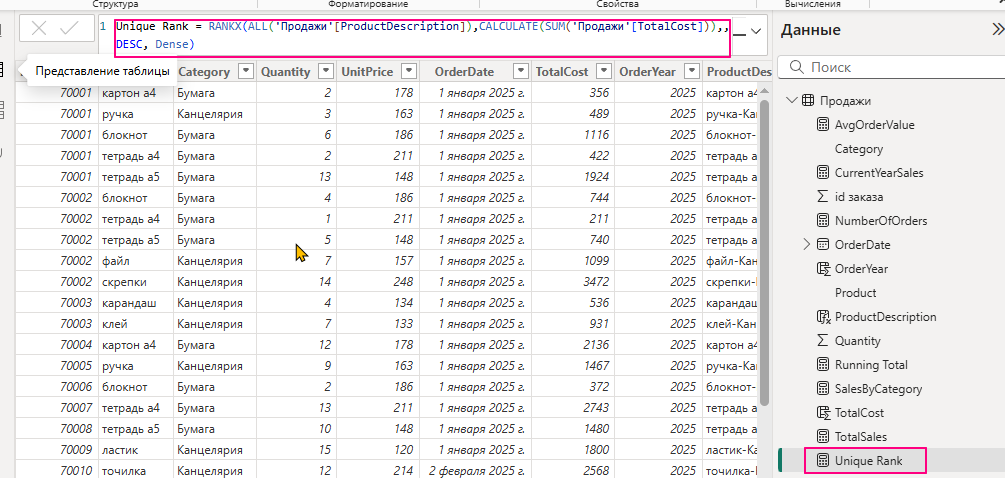


### **Упр. - Ранжирование с учетом повторений**

* 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)



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

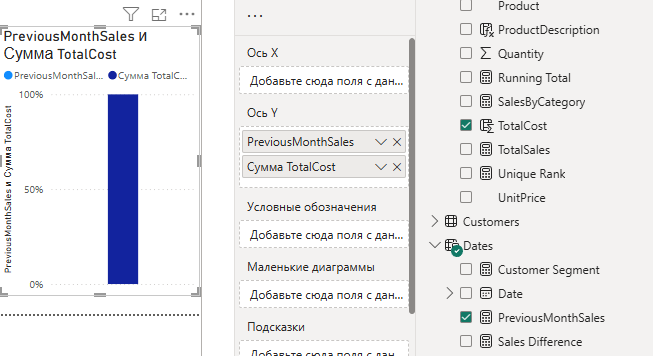
* Для анализа трендов полезно сравнивать значения в соседних строках:

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]

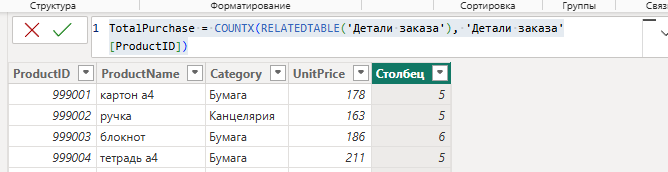


### **Упр. - Динамическая сегментация клиентов на основе перцентилей**

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])



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] )

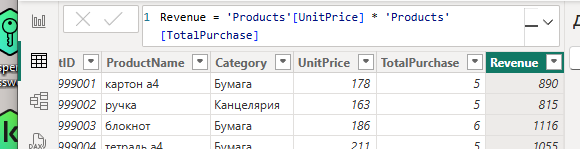
VAR TotalCost = SUM ( Sales[Cost] )

RETURN DIVIDE ( TotalRevenue - TotalCost, TotalRevenue, 0 )

РЕШЕНИЕ

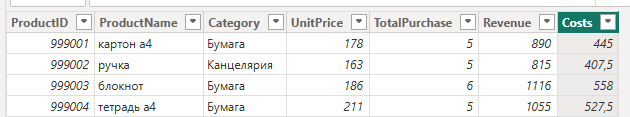
внесем в таблицу Products несколько вычисляемых полей с выручкой и себестоимостью

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



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

(себестоимость просто для примера такая - одинаковая для всех товаров



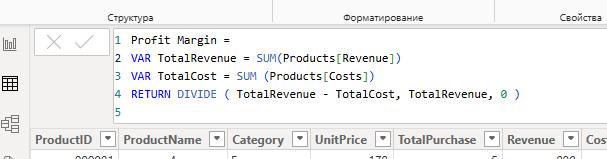
теперь рассчитаем маржу

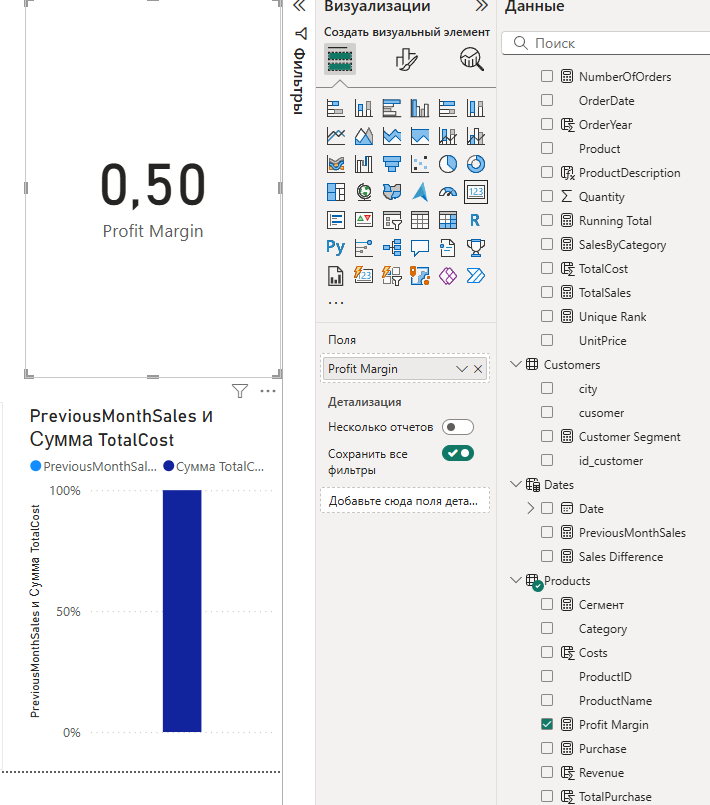
Profit Margin =

VAR TotalRevenue = SUM(Products[Revenue])

VAR TotalCost = SUM (Products[Costs])

RETURN DIVIDE ( TotalRevenue - TotalCost, TotalRevenue, 0 )





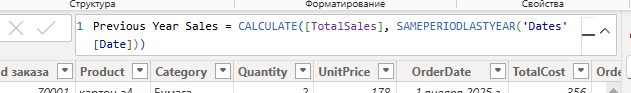
### **Упр. - Работа с временным интеллектом (Time Intelligence)**

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]))



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)

