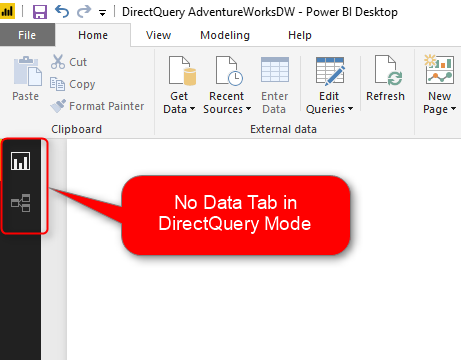
**Direct Query Mode**

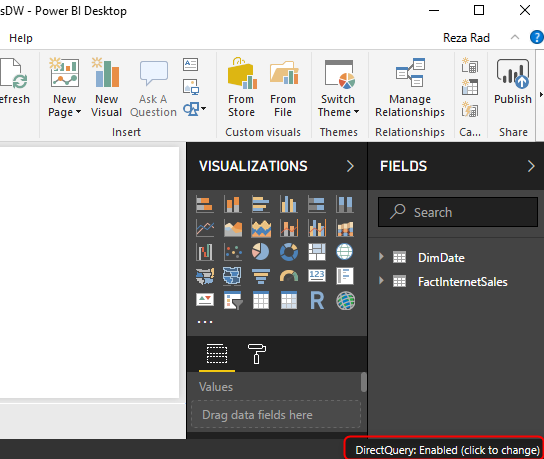
**There is no Data Tab in Direct Query Mode**

One of the very first things you will notice in the DirectQuery Mode is that there is no Data Tab (the middle tab on the left-hand side navigations of Power BI.



The Data tab shows you the data in the Power BI model. However, with DirectQuery, there is no data stored in the model.

Also at the bottom right side of the Power BI Desktop, you will notice that there is a note about DirectQuery connection.

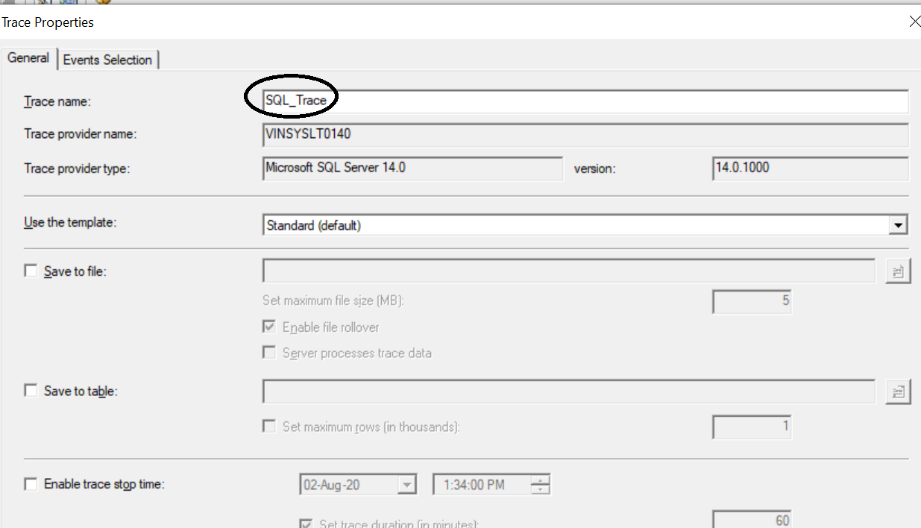


## How DirectQuery Works

With DirectQuery enabled; every time you see a visualization, Power BI sends a query to the data source, and the result of that comes back. You can check this process in SQL Profiler. SQL Profiler is a tool that you can use to capture queries sent to a SQL Server database

Demo:

1. Get connected to **SQL Server Profiler** and start a new **trace** from File 🡪 New Trace, get connected.
2. Give the trace name as **SQL\_Trace**

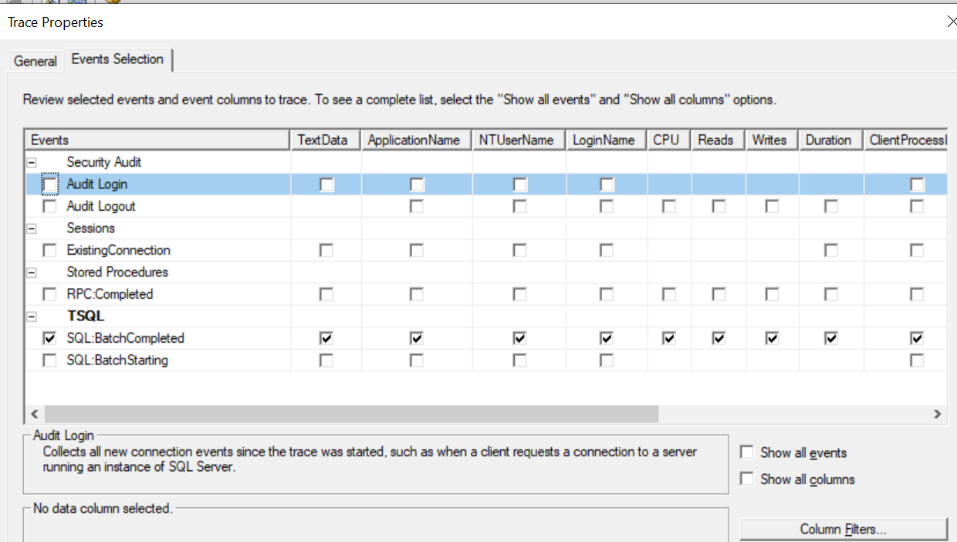
****

**Filtering trace results in SQL Profiler**

Click on the **Events Selection** tab

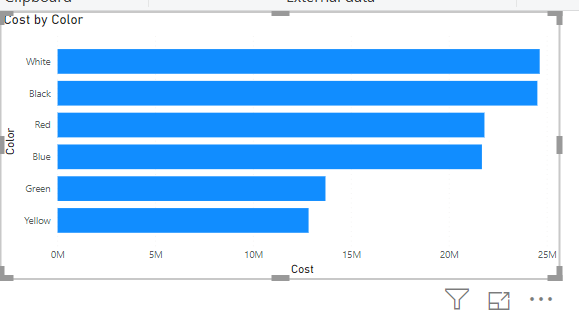
1) Deselect the check box of **SQL:BatchStarting**

2) Ensure only **SQL:BatchCompleted** is kept selected

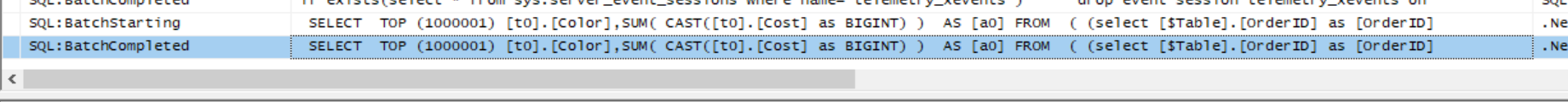


**3) Click on the Run button**

1. In PBI, get connected to SQL Server with of course **DirectQuery** option. Select the TShirts\_DB Database and then from that database select the table **SalesData**
2. Create a simple bar chart of Color and Cost as shown next 🡪



1. Immediately see the Profiler and observer the SQL query!!!



The complete query is as below:

SELECT

**TOP (1000001)** [t0].[Color],SUM(

CAST([t0].[Cost] as BIGINT)

)

AS [a0]

FROM

(

(select [$Table].[OrderID] as [OrderID],

[$Table].[OrderDate] as [OrderDate],

[$Table].[CustomerID] as [CustomerID],

[$Table].[SleeveLength] as [SleeveLength],

[$Table].[Color] as [Color],

[$Table].[Pattern] as [Pattern],

[$Table].[NeckStyle] as [NeckStyle],

[$Table].[Size] as [Size],

[$Table].[Price] as [Price],

[$Table].[Qty] as [Qty],

[$Table].[Cost] as [Cost]

from [dbo].[SalesData] as [$Table])

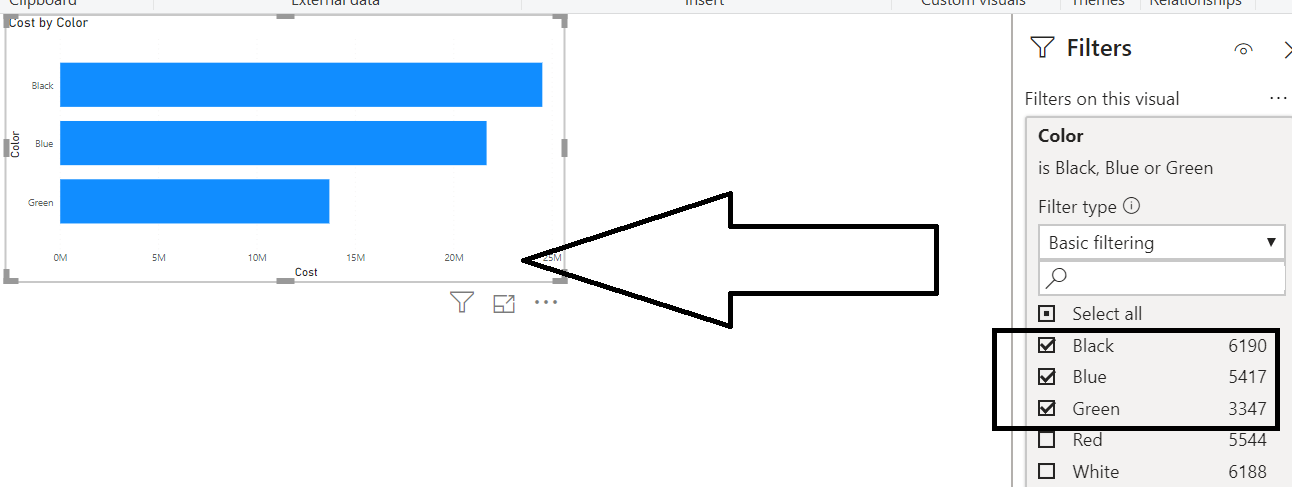
)

AS [t0]

GROUP BY [t0].[Color]

See the part **Top (1000001)**

1. Now in PBI apply a filter of Color as Black, Blue and Green as shown next



A new Profiler query will get generated as shown below:

SELECT

TOP (1000001) [t0].[Color],SUM(

CAST([t0].[Cost] as BIGINT)

)

AS [a0]

FROM

(

(select [$Table].[OrderID] as [OrderID],

[$Table].[OrderDate] as [OrderDate],

[$Table].[CustomerID] as [CustomerID],

[$Table].[SleeveLength] as [SleeveLength],

[$Table].[Color] as [Color],

[$Table].[Pattern] as [Pattern],

[$Table].[NeckStyle] as [NeckStyle],

[$Table].[Size] as [Size],

[$Table].[Price] as [Price],

[$Table].[Qty] as [Qty],

[$Table].[Cost] as [Cost]

from [dbo].[SalesData] as [$Table])

)

AS [t0]

**WHERE**

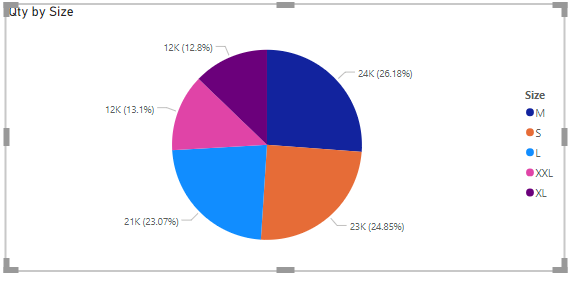
**(**

**([t0].[Color] IN ('Black','Blue','Green'))**

**)**

GROUP BY [t0].[Color]

1. Now in PBI place a pie chart and put Size and Qty in it as shown next 🡪



See the Profiler Query for the same pie chart 🡪

SELECT

TOP (1000001) [t0].[Size],SUM(

CAST([t0].[Qty] as BIGINT)

)

AS [a0]

FROM

(

(select [$Table].[OrderID] as [OrderID],

[$Table].[OrderDate] as [OrderDate],

[$Table].[CustomerID] as [CustomerID],

[$Table].[SleeveLength] as [SleeveLength],

[$Table].[Color] as [Color],

[$Table].[Pattern] as [Pattern],

[$Table].[NeckStyle] as [NeckStyle],

[$Table].[Size] as [Size],

[$Table].[Price] as [Price],

[$Table].[Qty] as [Qty],

[$Table].[Cost] as [Cost]

from [dbo].[SalesData] as [$Table])

)

AS [t0]

GROUP BY [t0].[Size]

**\*\*\* Note 🡪 The percentage contribution calculation is NOT done by SQL Server. It is done by DAX functions of PBI!!\*\*\***

**Imp point 🡪 One query, for each visualization. Even if two visualizations are showing the same thing, they still send two separate queries to the database.**