**Power BI & SQL Integration Documentation**

Steps to Import **Chocolates** Dataset in MySQL Workbench

1. Download the .sql file provided (chocolates.sql)

After you have the file,

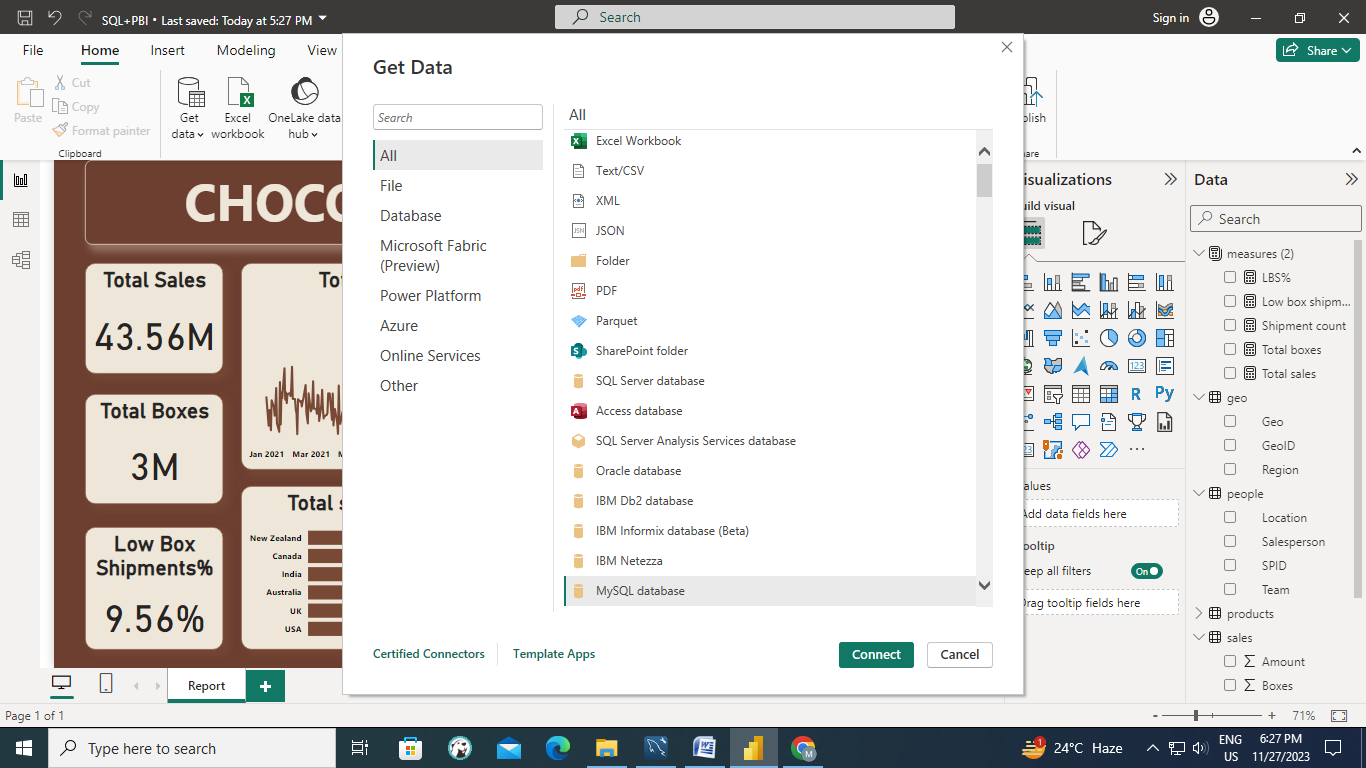
* Open MySQL Workbench, login if necessary
* Click on the “server administration” tab (see illustration, click to expand)
* Click on “Data Import/Restore”
* Select the option “Import from self-contained file”
* Specify the path of the downloaded chocolates.sql file
* Start import

At the end of these steps, MySQL should have the database. You can see that from “Schemas” tab on the workbench.

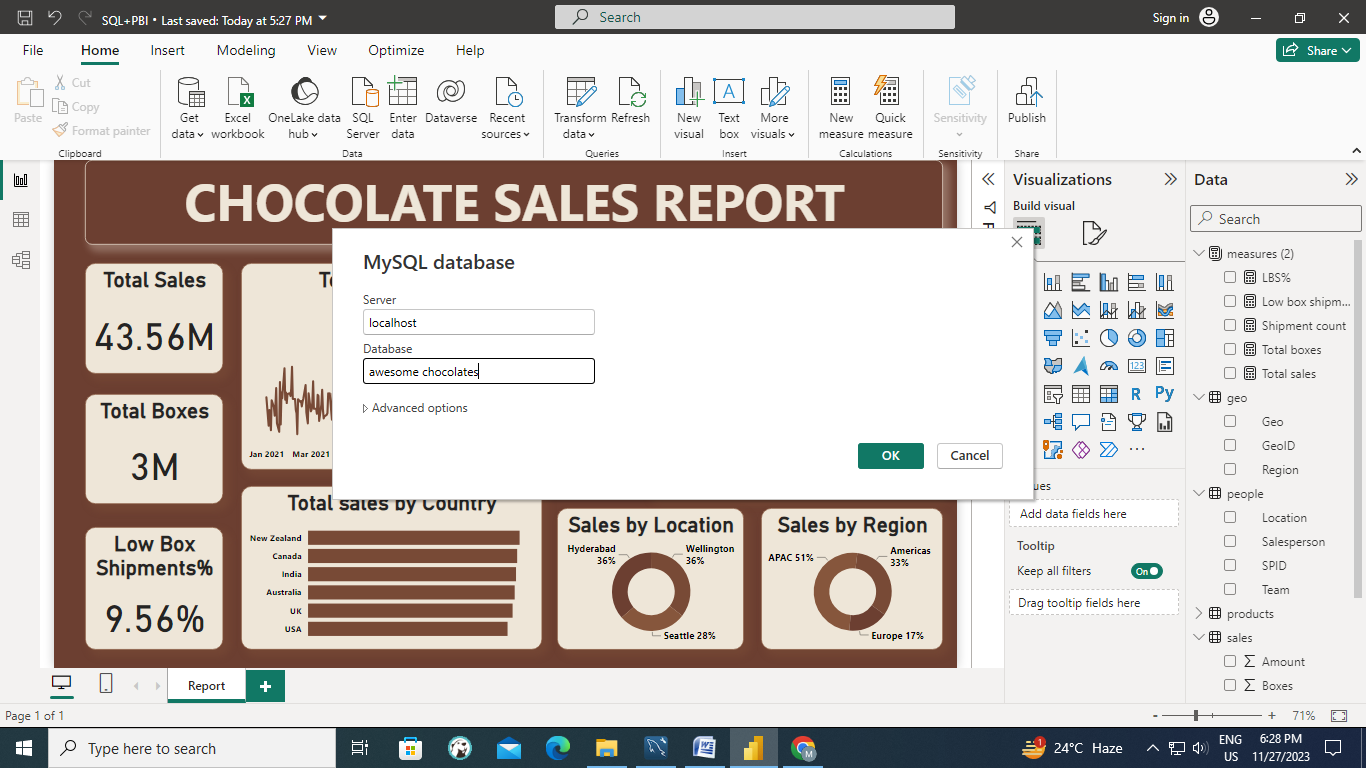
We have done some analysis of data, you can access it using queries.sql file.

Steps to Connect MySQL Database with Power BI

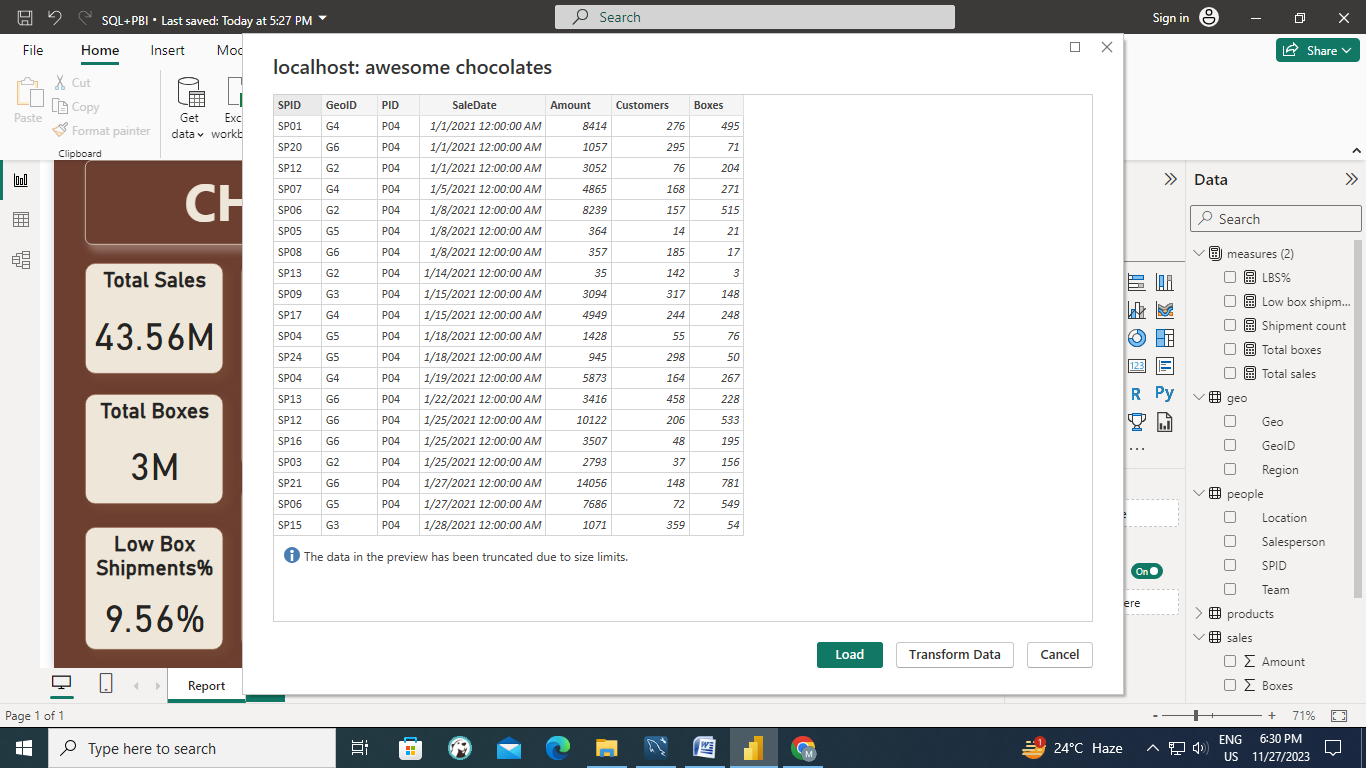
1. Go to **Get Data > More > MySQL Workbench**



1. Provide the credentials that is Username and password used in MySQL and name of database we want to analyze

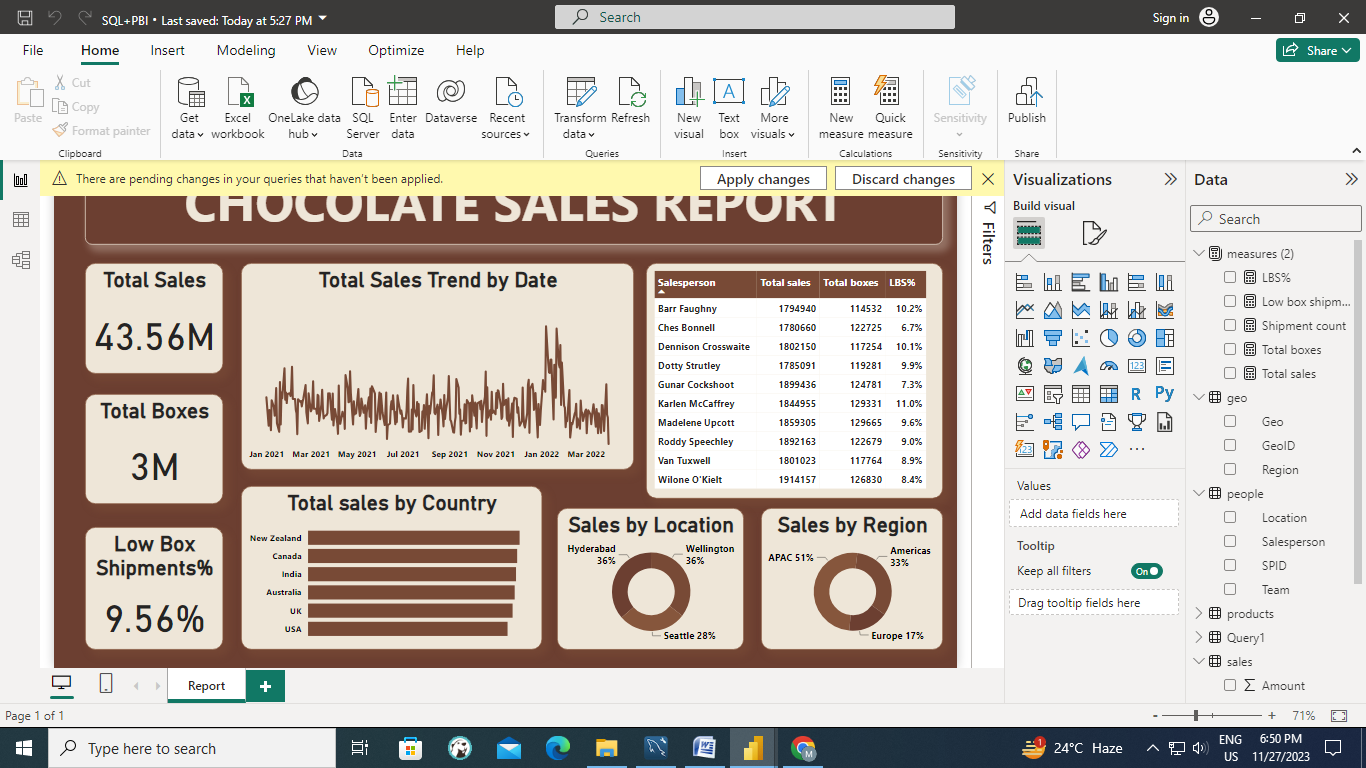


1. Select the tables you want to utilize and load the data in Power BI



Now we jump on to Power BI and start creating some measures and a basic report after analyzing the data.

And here’s the report.



Say the manager asks for the same report but for different products, then do we need to create the same report again and again for all products? The answer to this is, NO. In that case SQL queries comes in, we **parameterize** the SQL queries.

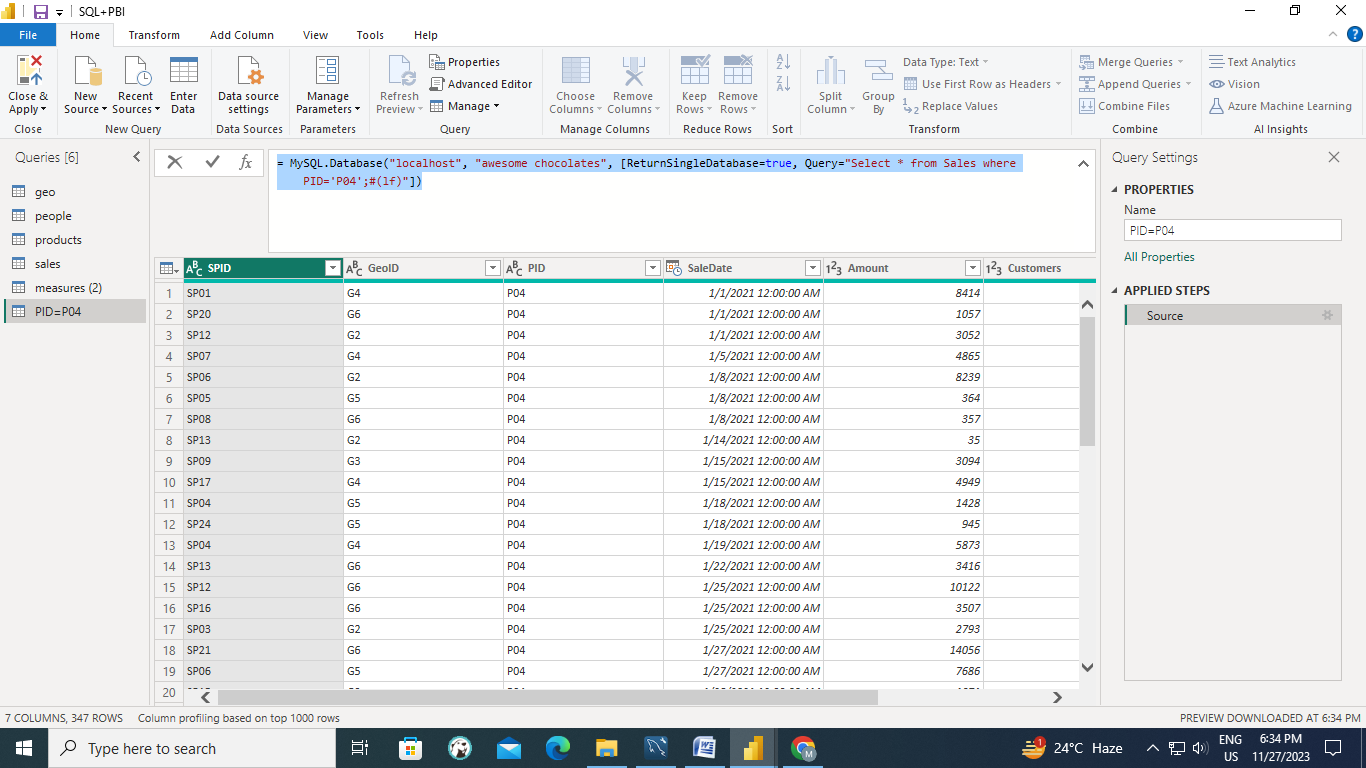
Yes we could have use page filters to resolve the same issue. But the end goal is not to fetch or load unnecessary data into Power BI.

So what we do is, we will go to Get Data and this time we write a SQL query there which gives us the data for some particular product.

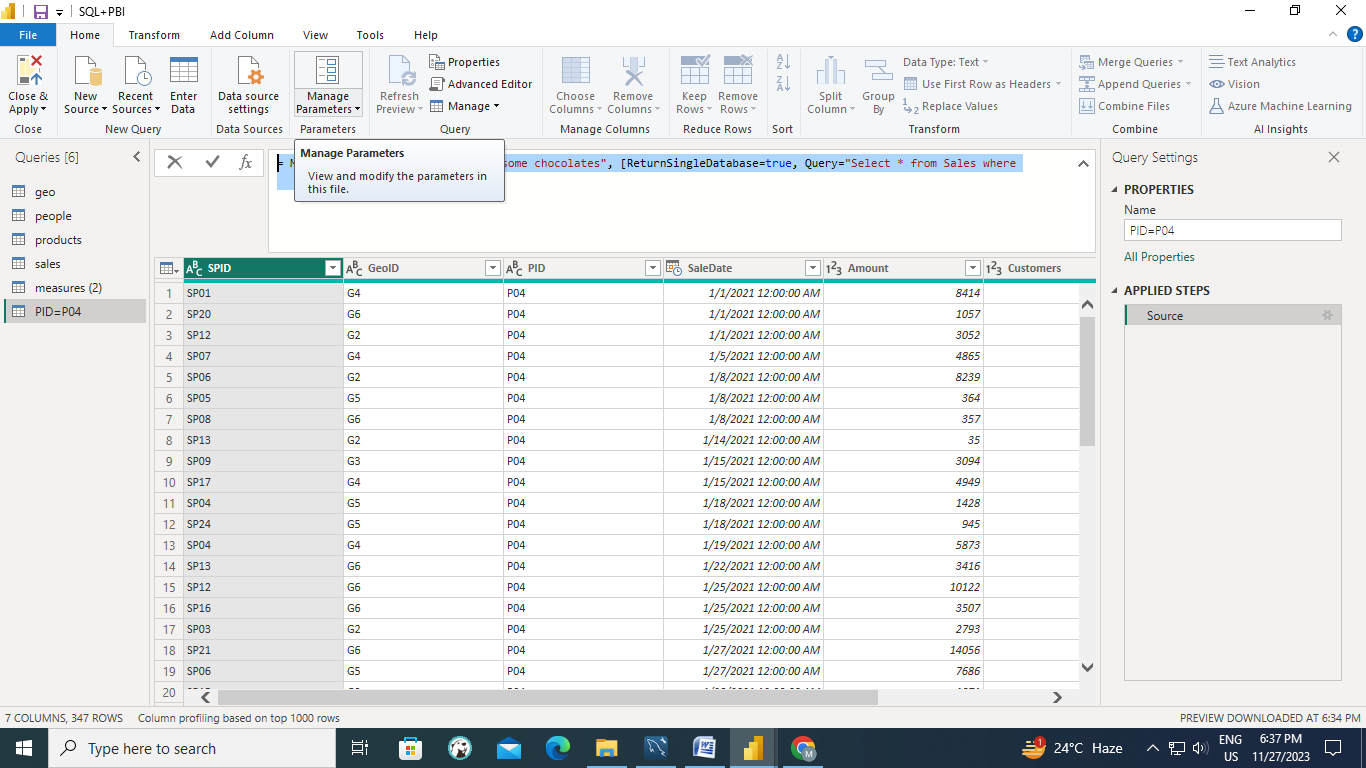
Select \* from Sales where PID='P04';

And we load the data only for **Product = “Raspberry Chocolate” (PID = P04)**

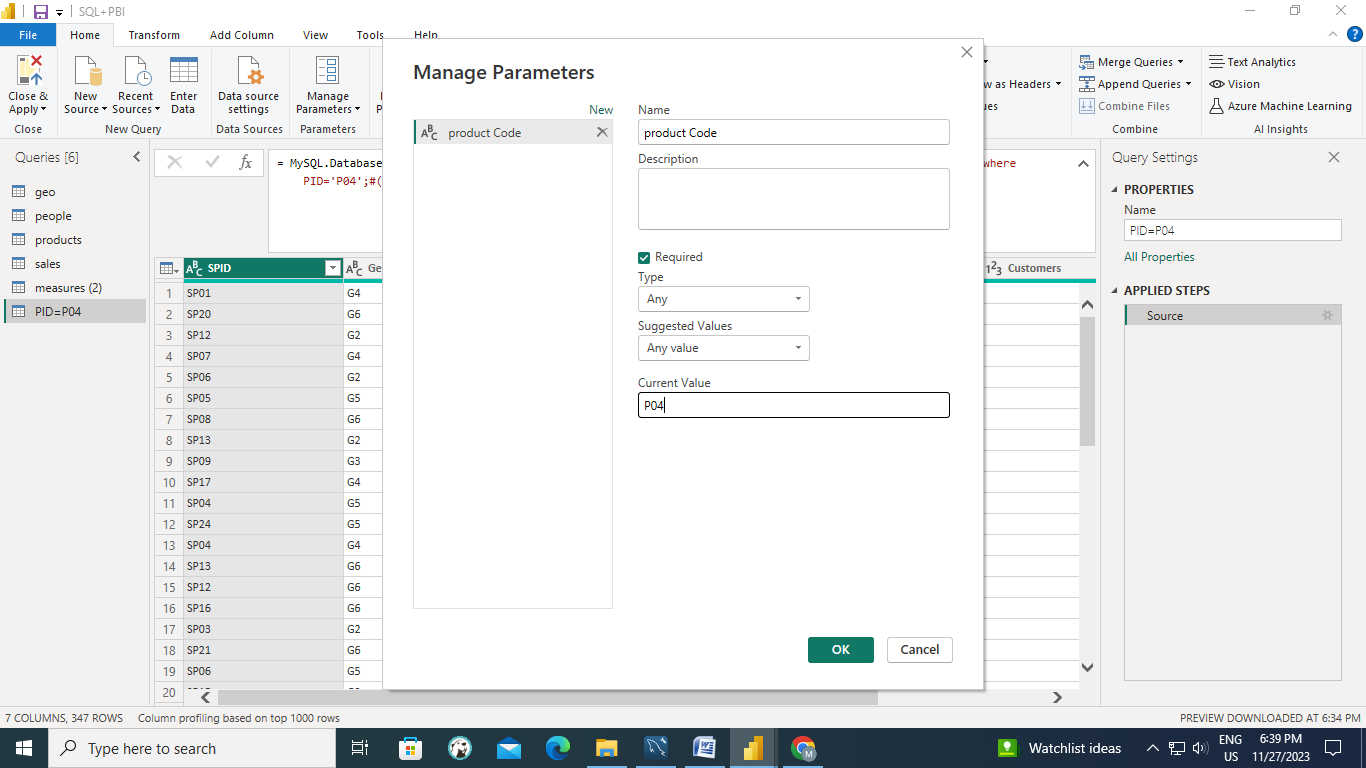
If we transform data, then we see this table is for PID=P04, now we want to parameterize it.



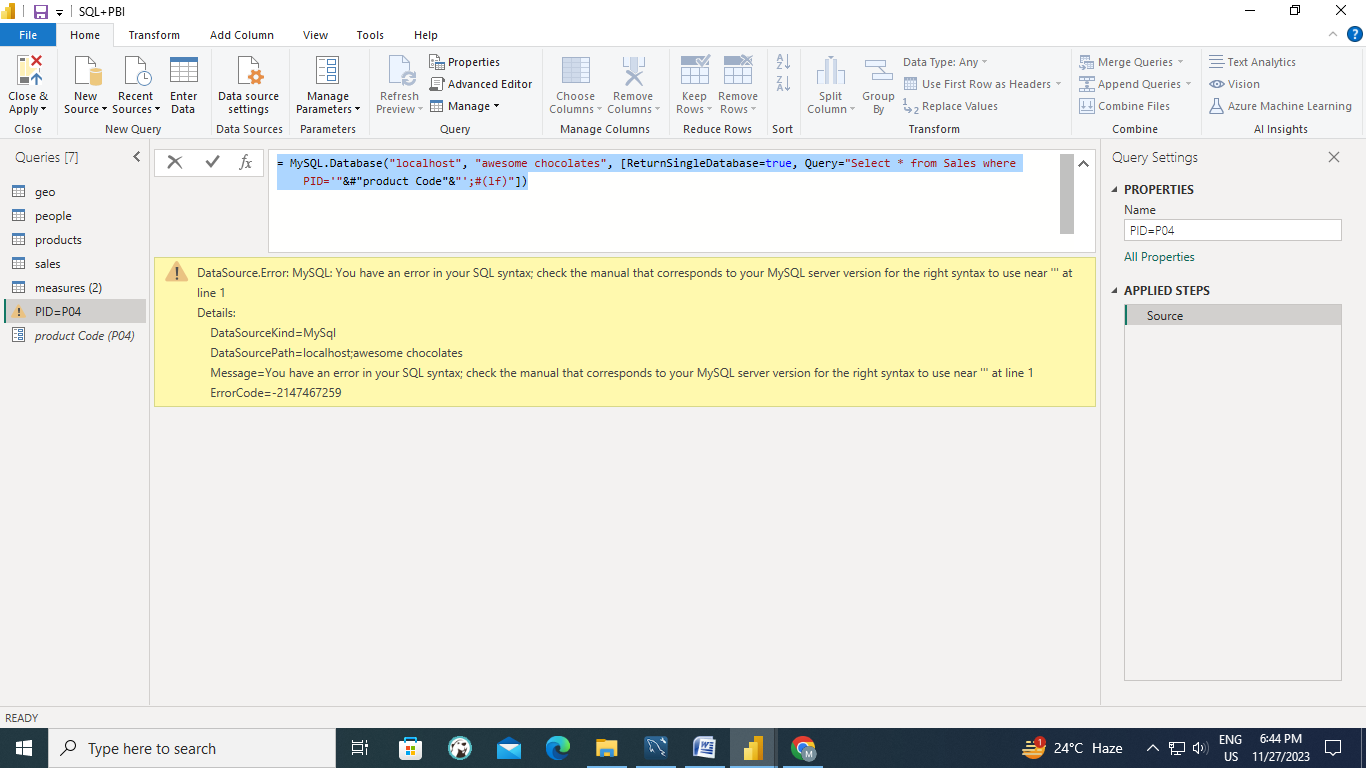
And to parameterize we Manage Parameters ribbon in Home tab



Then create a new parameter & set the current value to P04.



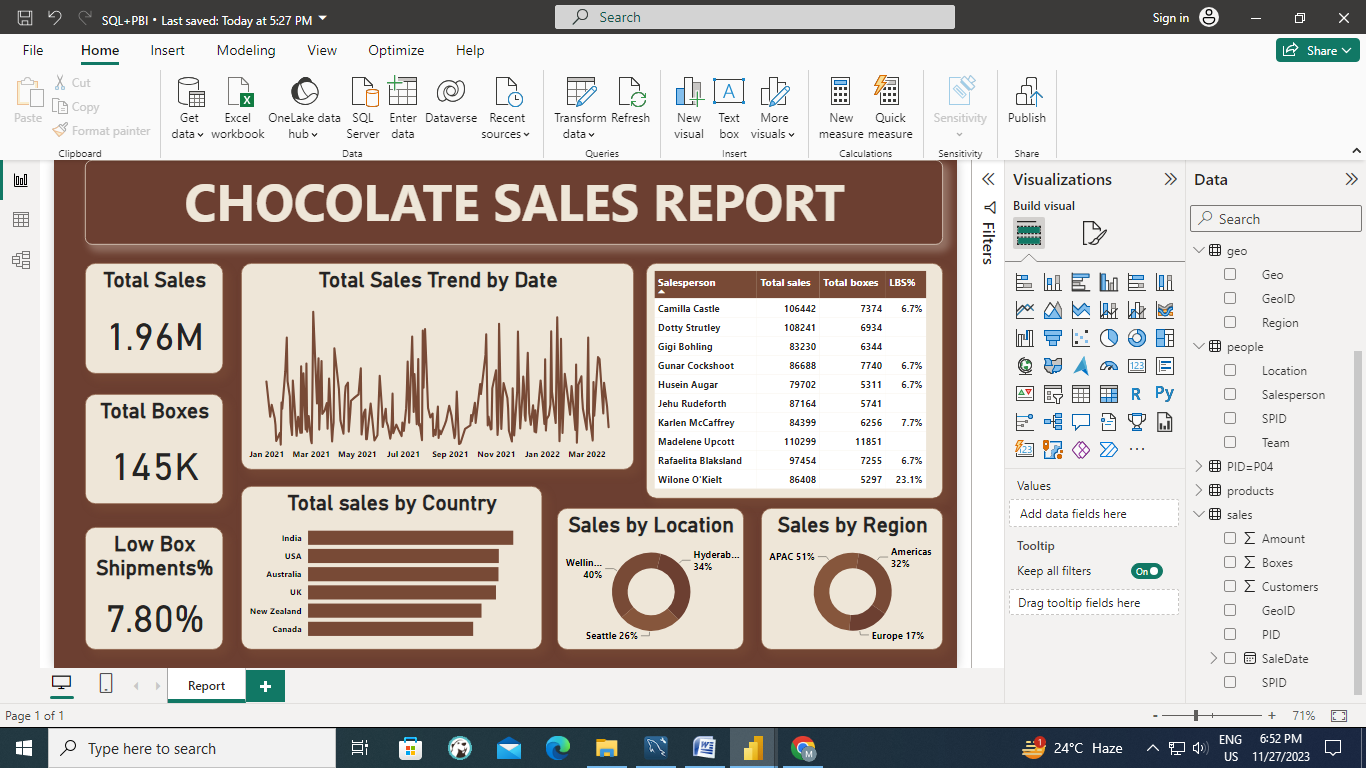
Now, change default value of the query to parameter.



Allow access.

Now on changing the parameter value, whole table will be changed.

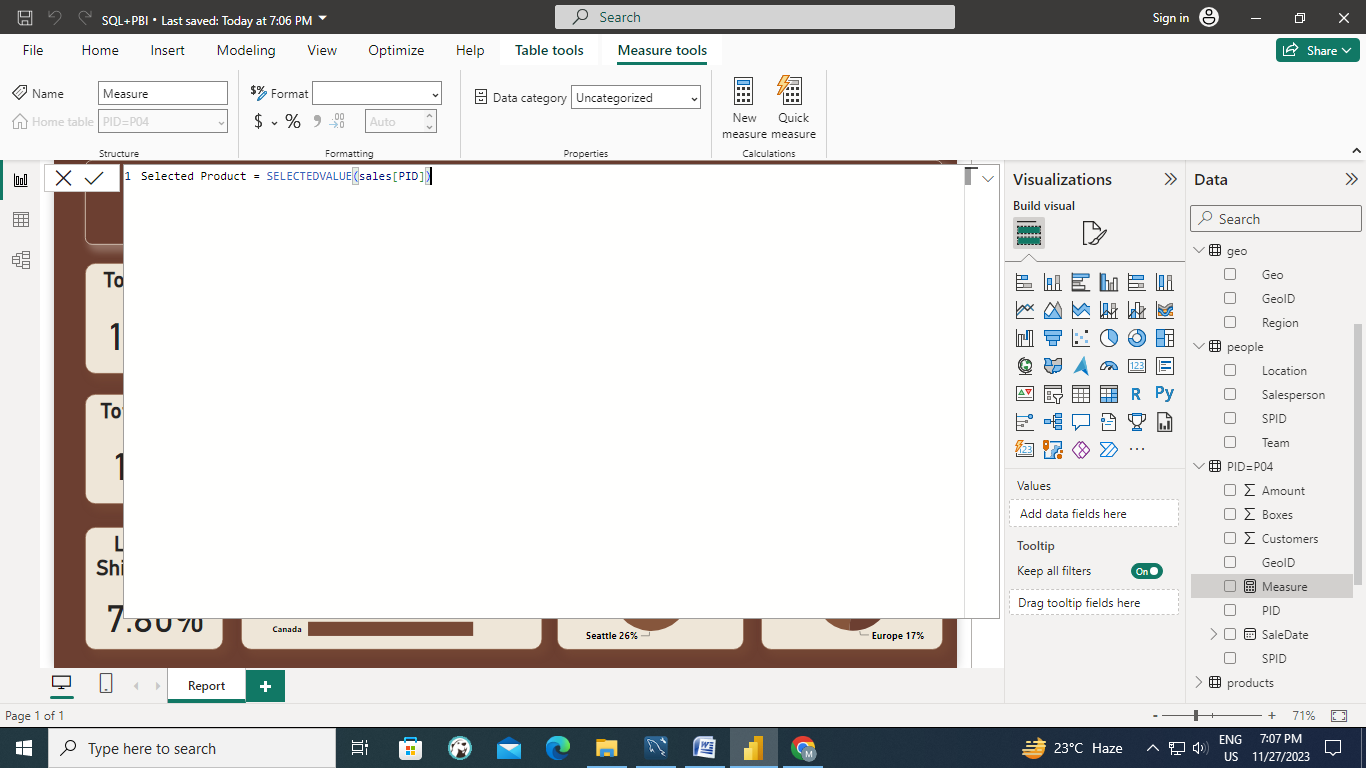
Copy this query, go to Sales table and paste the query in source. And the changes will now get reflected in report we created. And here’s the new report for PID=P06.



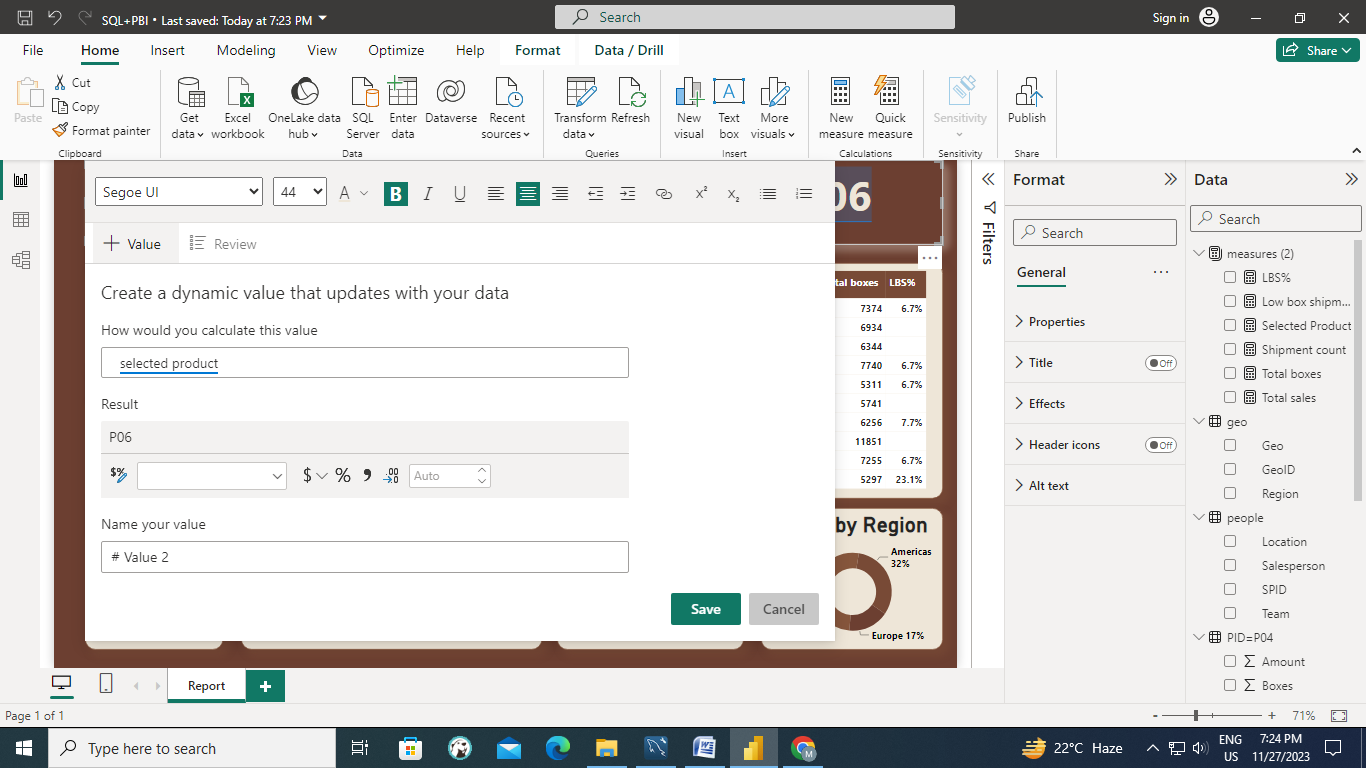
So, every time you need report for different products, just go to Power Query and change the parameter value and as a result of it, the changes will get reflected for that product.

Okay, but this report doesn’t tell me that which product it has been created for. And we can do that by displaying parameter dynamically on Power BI report.

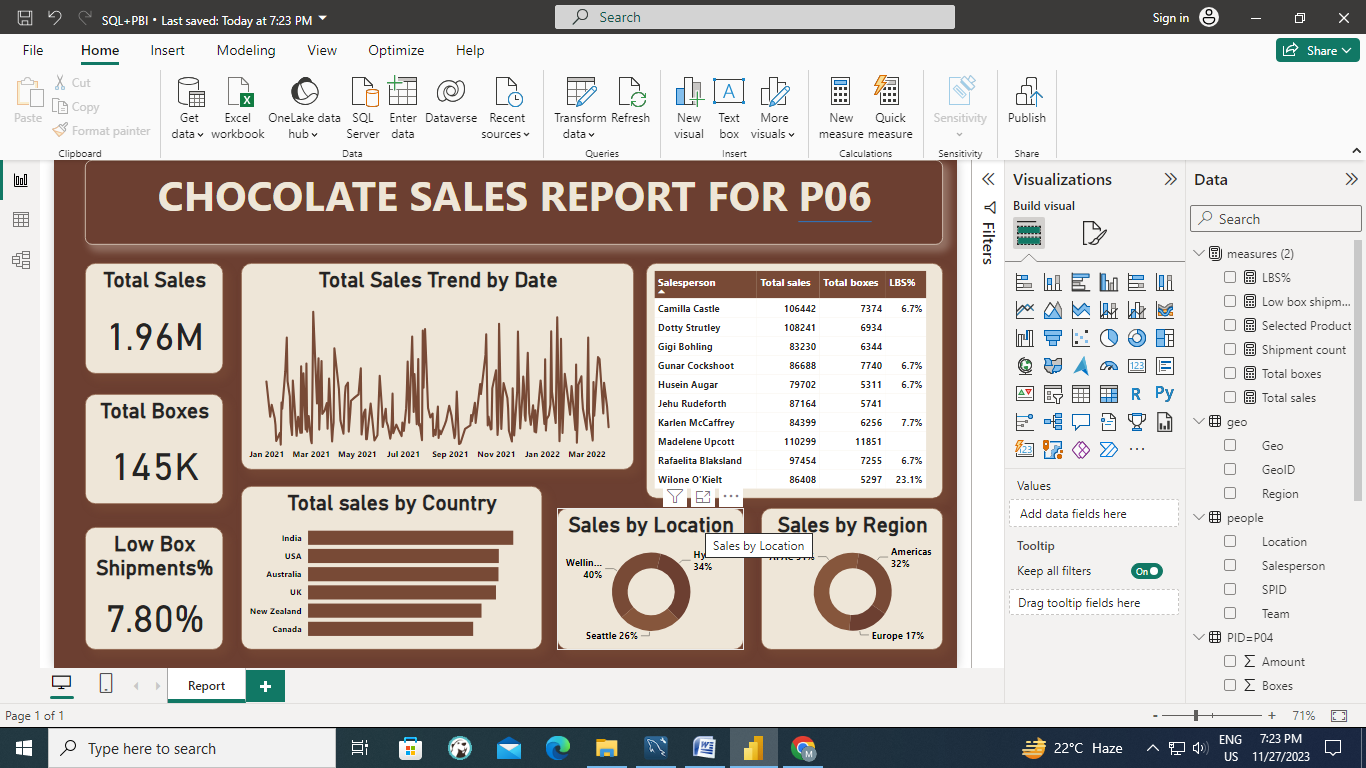
For that, we create a measure.



Now, click + to add aprameter



And finally we can see dynamic parameters on the report.



This was the journey from SQL to Power BI, creating parameters and making them appear dynamically on Power BI report.