**What is a Data Model?**

In short, a data model is the collection of

* loaded tables of data
  + with one or more columns that have been loaded for a reason (don’t load any columns from your source into your model unless you have a reason).
  + with one or more rows of data that have been loaded for a reason (don’t load rows from your source into your model unless you have a reason).
* relationships between the loaded tables
* the measures (formulas) written that apply the business rules to the raw data to extract business insights (e.g., sell price minus cost price is profit), including
  + the formulas themselves
  + the formatting applied to the formulas
  + the business names given to the formulas

All of the above make up what is referred to as “the data model” in Power BI (also for Analysis Services Tabular, Power Pivot for Excel and no doubt countless other BI tools).

**Why Do I Need a Model?**

Power BI is a **data model based BI reporting tool;**not all BI tools are data model based. One example of a non-data model based BI tool is SQL Server Reporting Services (SSRS). **SSRS is a non-data model based BI tool**. Non-data model based BI reporting tools require the report writer to first generate a query to fetch the data from a database (typically SQL Server) and return the results of that query to SSRS so the results can be rendered in a report. Many non-data model based reporting tools have a user interface that helps with the generation of the query (if you want to use it), or you can also use a scripting language like T-SQL to fetch the data that you need for each report.

A second example is Excel. **Traditional** **Excel is also a non-data model based BI tool**. By traditional Excel, I mean Excel the spreadsheet tool without the modern BI add-ins of Power Query and Power Pivot. In the case of traditional Excel, the user loads the data into the spreadsheet and then logically relates and aggregates the data using Excel formulas and builds a report (often on a new sheet) to summarise and present the results (the report).

Now don’t confuse “a tool” here with “a data model based tool”.  Excel is definitely a tool – it is a very flexible tool that lets you build virtually any report without being a programmer.  In fact I think Excel is probably the best and most popular BI tool ever invented.  But it is not a data model based tool – because it doesn’t have a data model.

There is nothing wrong with using a non-data model based BI tool – it is just that it is a different approach to using a data model based tool. The biggest “issue” with non-data model based reporting tools is that every time you need a new report, you have to start again, often from scratch. Each report has a single purpose with very little reusability or extensibility.

The benefits of a data model based tool like Power BI are:

* the author of “the data model” **builds a reusable capability (the data model) that can be used to solve the current reporting requirements as well as (often) future requirements** without the need to go back and write further queries to retrieve a new subset of data.
* The author can often be a business user (normally with good Excel skills) – **they don’t have to be a professionally trained database administrator or SQL professional.**
* The model is “conceptual” in nature, **supported by a user friendly interface that let’s you “build” the model logically with the minimum of coding**. Note: **You will need to do some coding** (writing formulas) for a good model, but no harder than you average Excel spreadsheet [once you have the skills](https://exceleratorbi.com.au/learn-power-bi-from-an-industry-expert/).

**Data Modelling in Power BI**

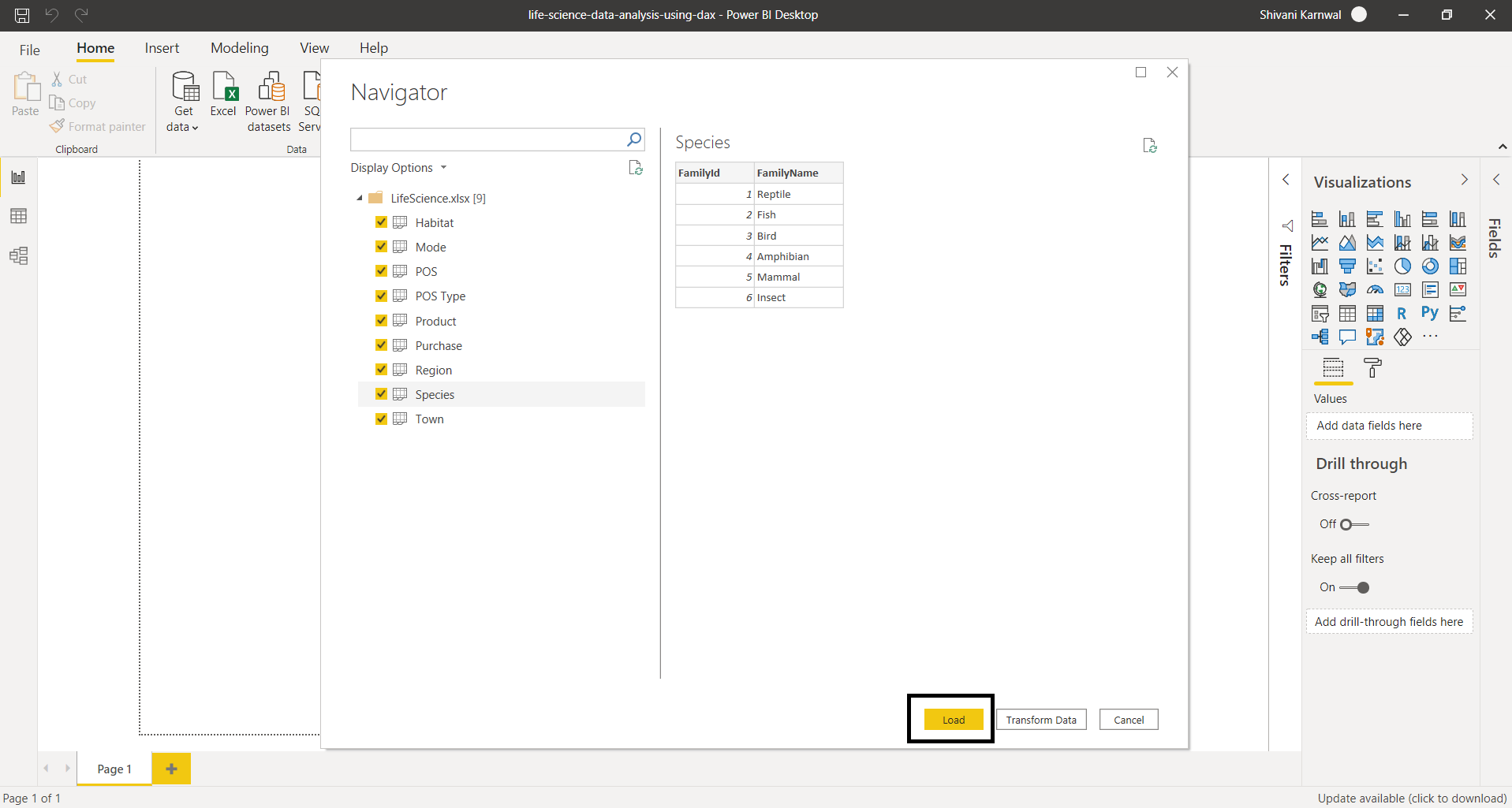
Data Modelling is one of the features used to connect multiple data sources in BI tool using a relationship. A relationship defines how data sources are connected with each other and you can create interesting data visualizations on multiple data sources.

With the modelling feature, you can build custom calculations on the existing tables and these columns can be directly presented into Power BI visualizations. This allows businesses to define new metrics and to perform custom calculations for those metrics.

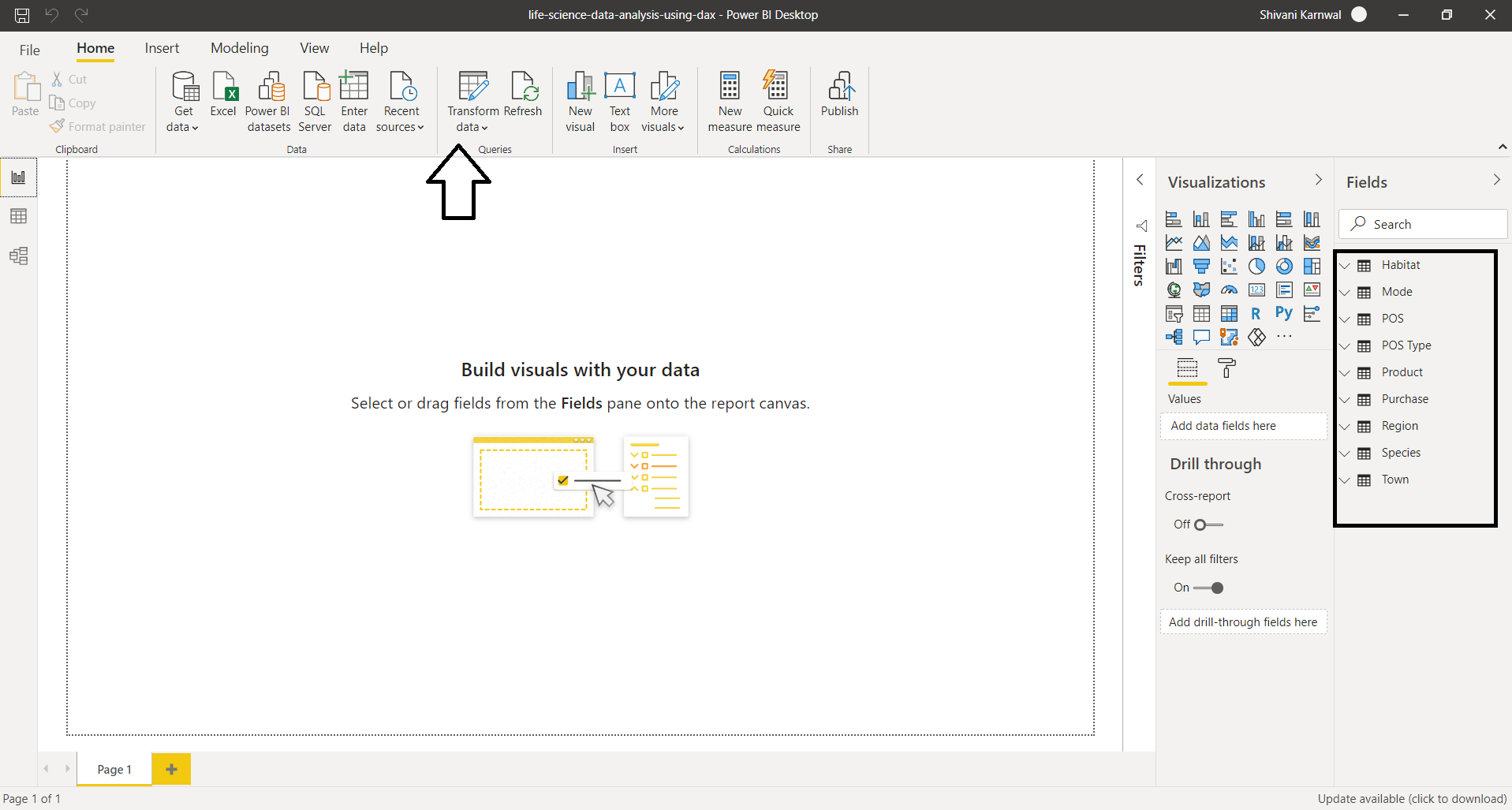
In Power BI, you set the relationship between two objects. To set the relationship, you have to drag a line between the common columns. You can also view the “Relationship” in a data model in Power BI.

**Importing the dataset**

Browse the dataset and select all the tables. Then click on LOAD.

****

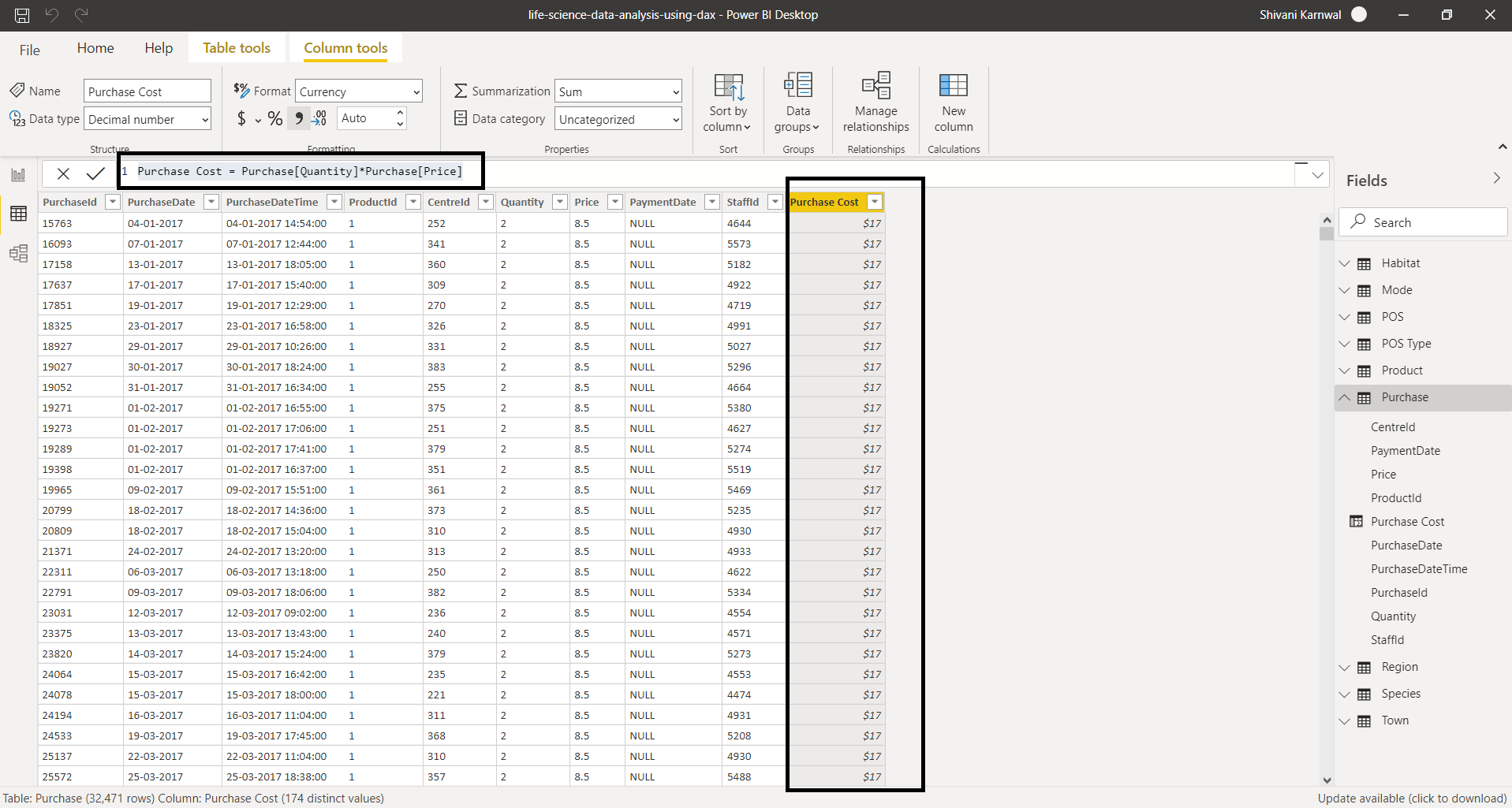
You will see that all the tables will be listed in the report view and then click on transform data to open power query editor.



Requirement 1: Create the following segment in terms of Purchase - Affordable (if< 10) - Expensive (if >10)

Firstly, we need to find purchase cost by creating a new column in Purchase Table. Go to Data view & click on New Column for Purchase Table. Name the new column as Purchase cost . Write the Dax expression for it.

Purchase Cost = Purchase[Quantity]\*Purchase[Price]

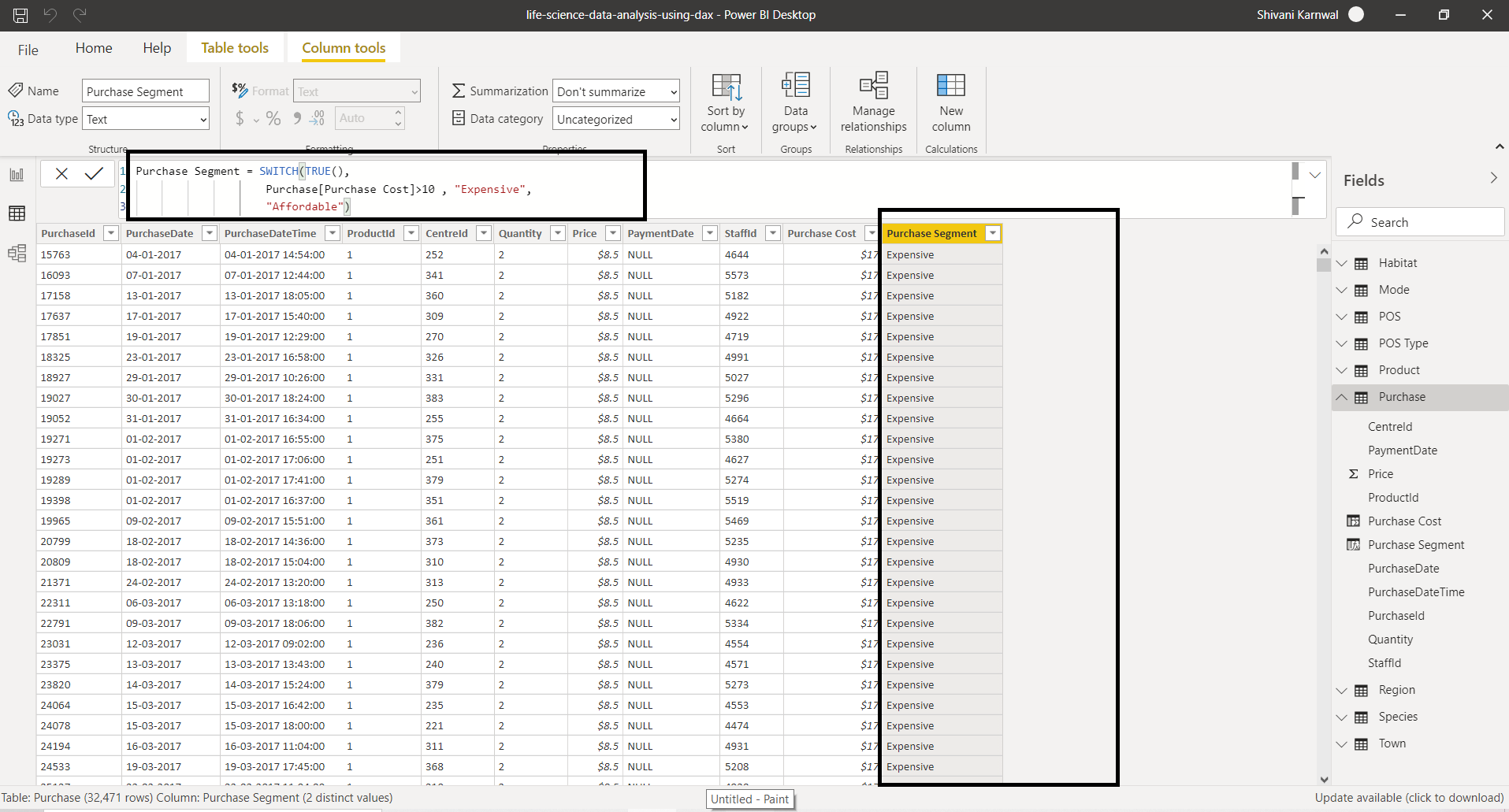


Now we can write the Dax expression for our requirement. We will add new column and write the following DAX for it.

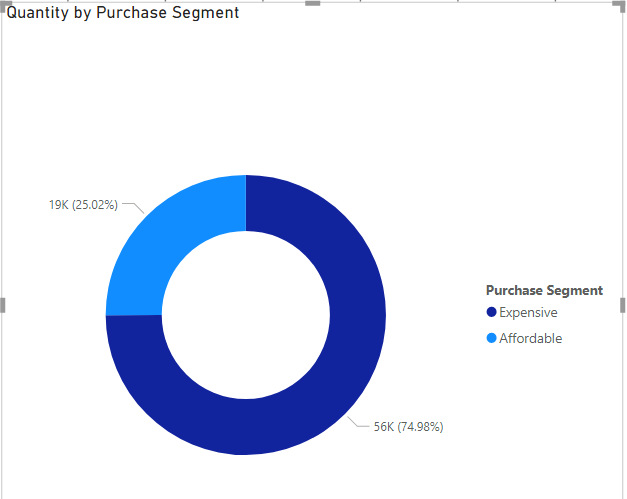
Purchase Segment = SWITCH(TRUE(),

Purchase[Purchase Cost]>=10,"Expensive",

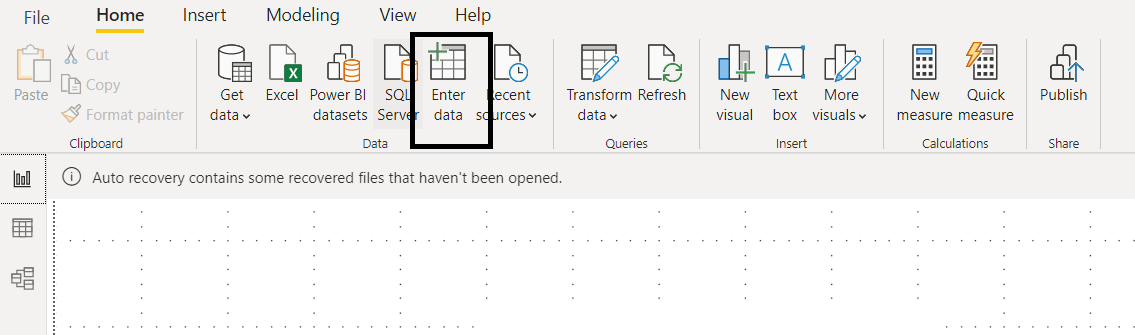
"Affordable")

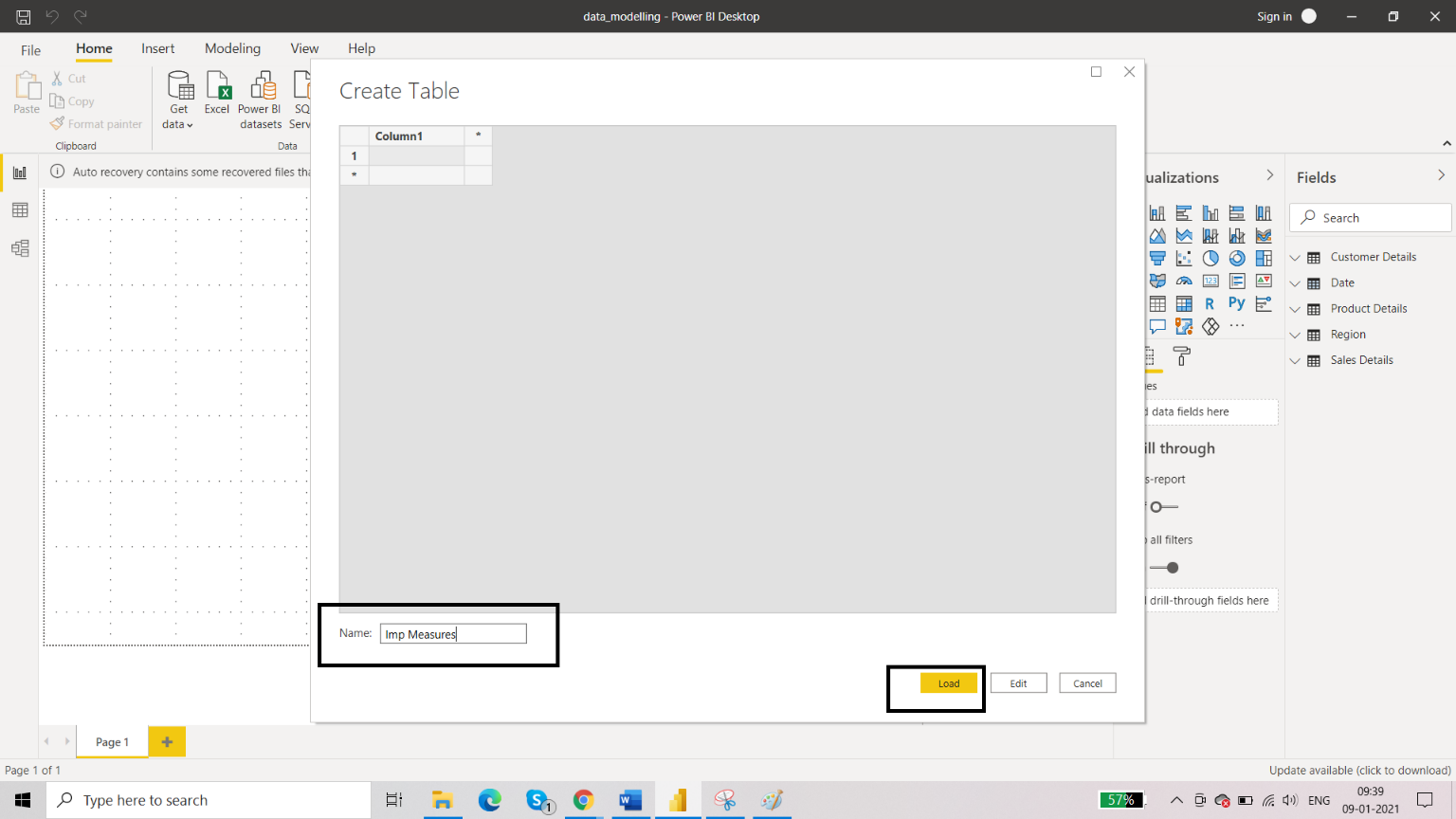


Requirement 2: Need a report on the quantities available on the basis of Animal 1. Affordable 2. Expensive

Requirement 3: Need to know how many animals do not belong to Air, Water & land

For this requirement, we will create 3 different measures. These measures will reside in a measure table. It is not a good practice to keep the measures in one table where they are created. To resolve this issue when there are many measures, we can create a measure table. So, a measure table is just like any other table in Power BI but it only contains measures.



DAX Functions: Insert 3 columns

NOT\_LAND = CALCULATE(COUNT(Habitat[EnvironmentId]),Habitat[EnvironmentId] <> 1)

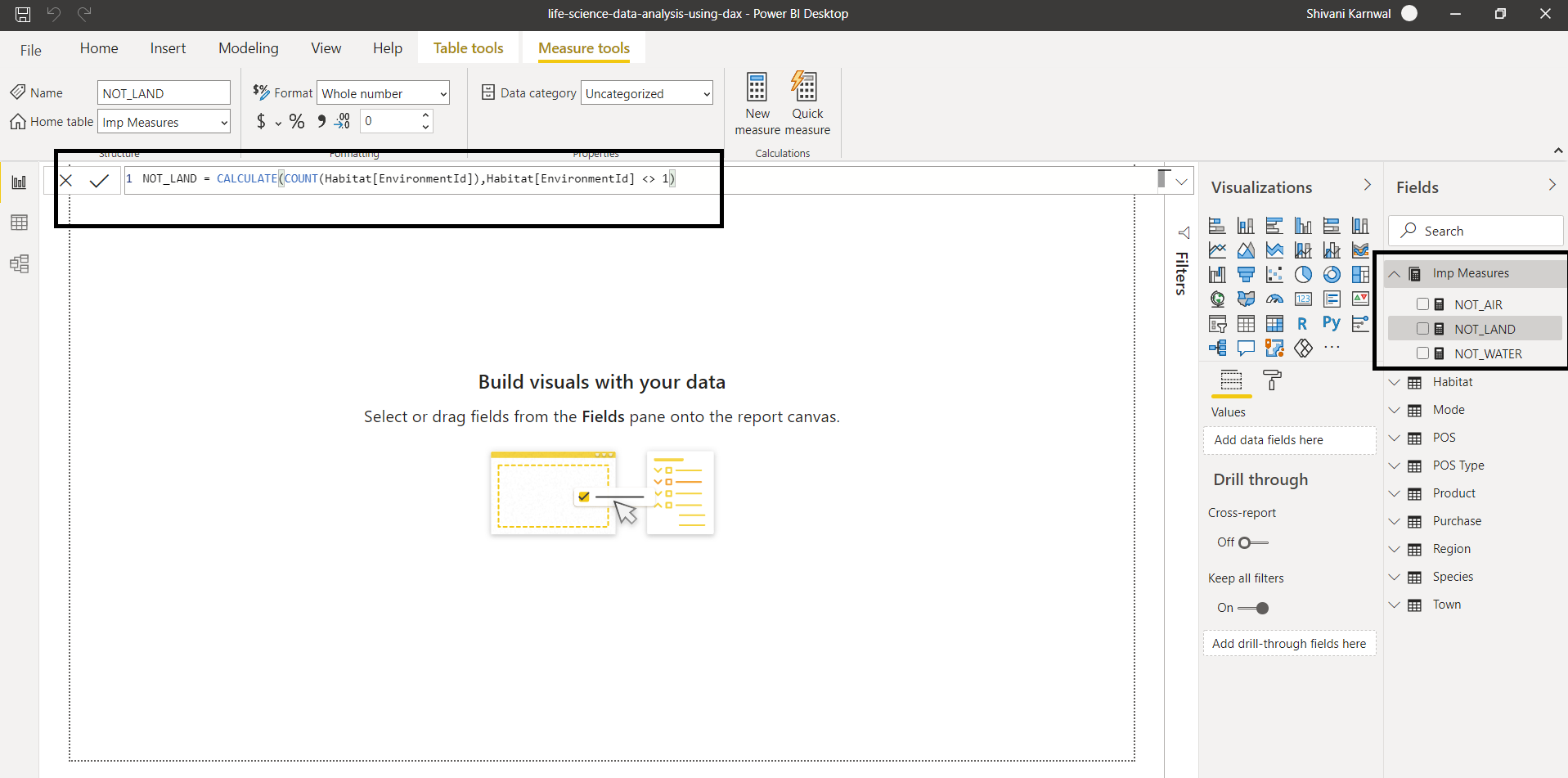
NOT\_AIR = CALCULATE(COUNT(Habitat[EnvironmentId]),Habitat[EnvironmentId] <> 2)

NOT\_WATER = CALCULATE(COUNT(Habitat[EnvironmentId]),Habitat[EnvironmentId] <> 3)

NOT\_AIR = CALCULATE(COUNT('Product'[Animal]), FILTER('Mode',Mode[EnvironmentId] <>2))

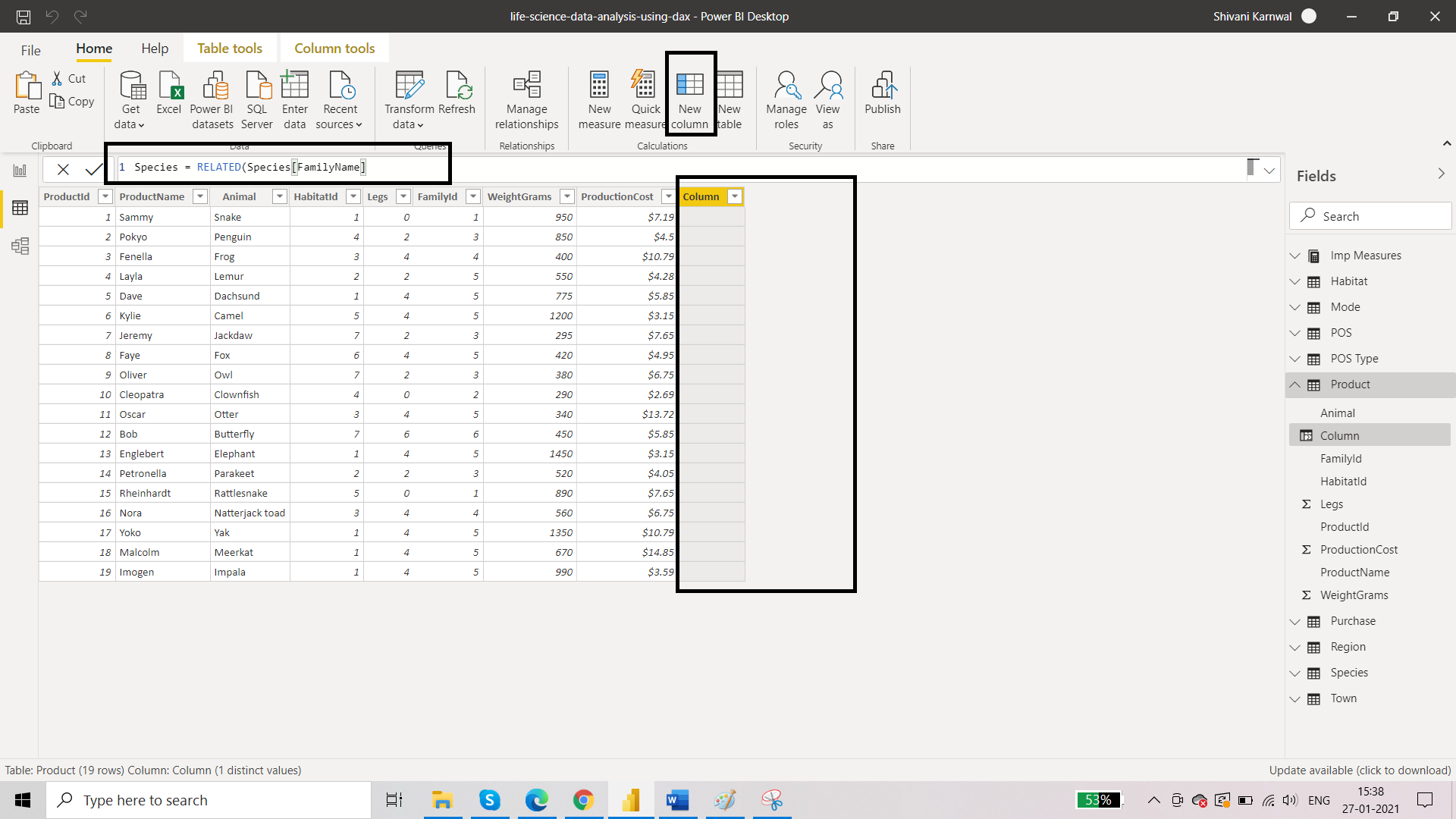
NOT\_LAND = CALCULATE(COUNT('Product'[Animal]), FILTER('Mode',Mode[EnvironmentId] <>1))

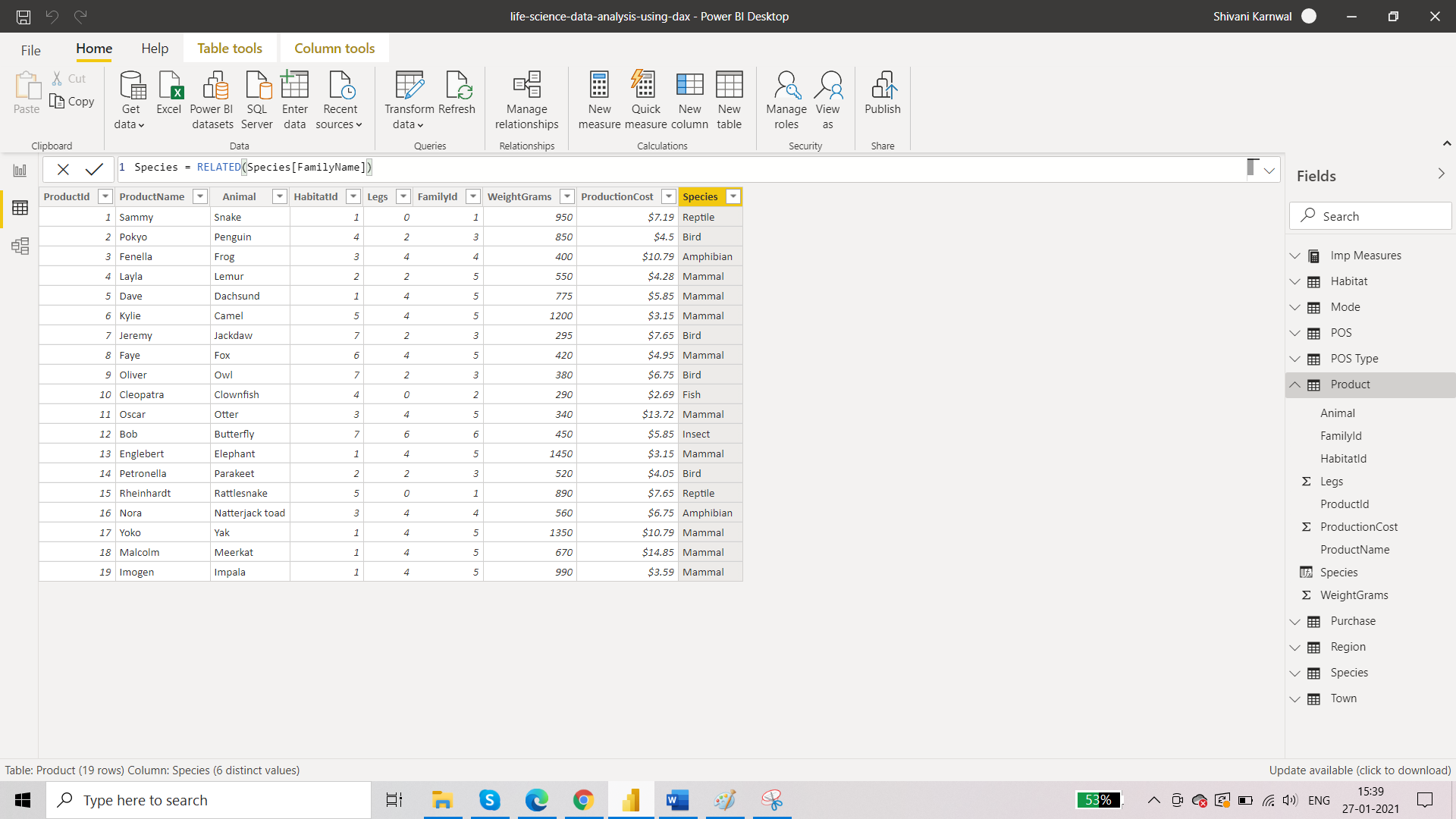
NOT\_WATER = CALCULATE(COUNT('Product'[Animal]), FILTER('Mode',Mode[EnvironmentId] <>3))



Requirement 4: Create a segment on the basis of Species by the referring Animals and create a column in the required table. (hint – Product Table)

* Go to Data view and Select the product table and click on new column.
* To create a segment on the basis of Species by the referring Animals we can use a DAX function called RELATED. Related function helps to return the related values in another table to the new column.
  + - * + Species = RELATED(Species[FamilyName])
* Product table add a new column (Species)





Requirement 4: Need to segment the towns on the basis of region.

* Go to Data view Select the Town table and click on new column.
* To create a segment on the basis of Region. we can use a DAX function called RELATED. Related function helps to return the related values in another table(Region Table) to the new column.

