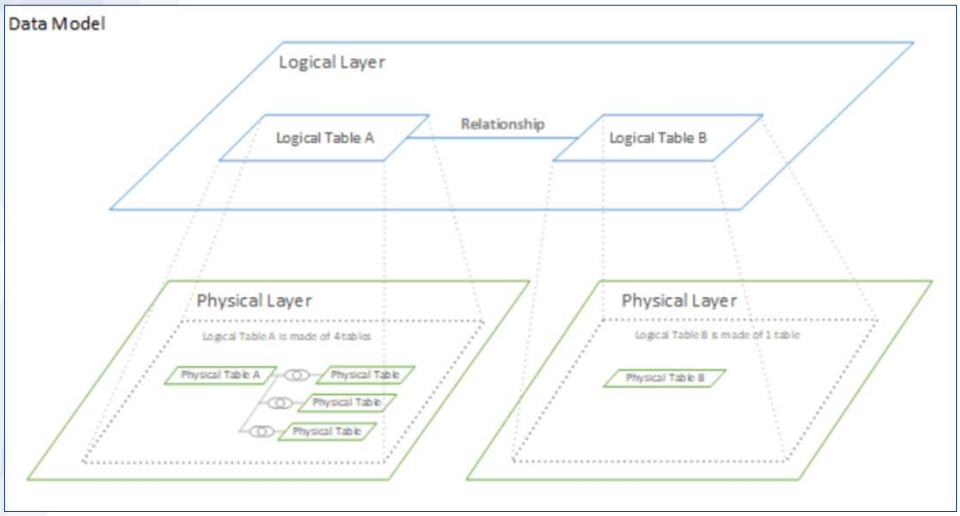
Lab 3:
Tableau Data Modelling

Strathmore
UNIVERSITY

Tableau Data Model



Since the Tableau version 2020.2 the Tableau data model consists of a logical and Physical layer that provides more options for combining data using schemas to fit your needs.

The Logical Layer: tables are linked using relationships and remain distinct (normalised), not merged in the data source. Joins between tables are determined during runtime/visualisation

The Physical Layer: tables are combined using joins and unions to create a single, flattened table (de-normalised for analysis).

Tableau supports the two main data model schemas using in data warehouses (star and snowflake) as well as single-table that contains a mix of dimensions and measures.
→ *If the information about relationships is encoded in your DW Tableau will automatically use this information to configure the relationships.*

Data Example one

Strathmore UNIVERSITY

Orders

	A	B	C	D	N
1	Row ID	Order ID	Order Date	Ship Date	Product ID
2	6315	CA-2018-100762	24/11/2018	29/11/2018	OFF-AR-10000380
3	6316	CA-2018-100762	24/11/2018	29/11/2018	OFF-LA-10003930
4	6317	CA-2018-100762	24/11/2018	29/11/2018	OFF-PA-10001815
5	6318	CA-2018-100762	24/11/2018	29/11/2018	OFF-PA-10004082
6	9462	CA-2018-100867	19/10/2018	24/10/2018	TEC-PH-10004922

Returns

	Returned	Order ID
2	Yes	CA-2018-100762
3	Yes	CA-2018-100762
4	Yes	CA-2018-100762
5	Yes	CA-2018-100762
6	Yes	CA-2018-100867
7	Yes	CA-2018-102652
8	Yes	CA-2018-102652
9	Yes	CA-2018-102652
10	Yes	CA-2018-102652

People

Regional Manager	Region
Sadie Pawthorne	West
Chuck Magee	East
Roxanne Rodriguez	Central
Fred Suzuki	South

To demonstrate the functionality we will use the test Superstore Orders workbook that has three worksheet.

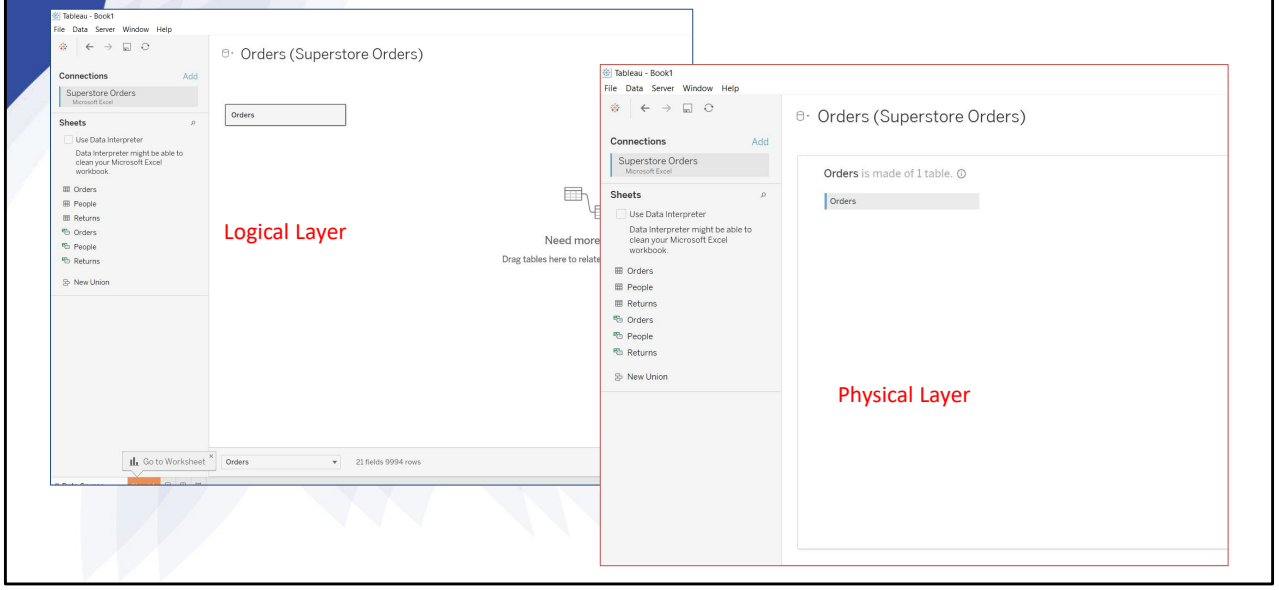
The **Orders** worksheet has Order details and the products in the order. One order may map to multiple products

The **Returns** worksheet that provides an dimension of whether a given product was returned. However, the returns are mapped to a given order (Order ID) rather than the product (Product).

The **People** worksheet that provides the managers for the four sales regions.

Not an ideal due to the lack of normalisation, which would cause a lot of referential integrity problems which we will discuss. But provides a good data for differentiating the functionality of logical and physical data models.

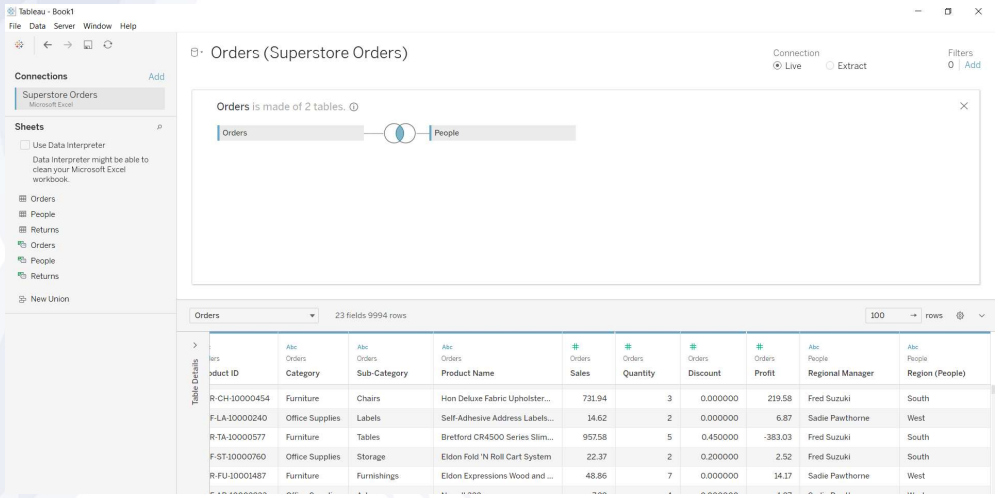
The Physical Data Model



1. Connect to the Superstore Orders excel workbook in the LAB 3 folder
2. Drag the Orders table to the workspace
3. The Orders table is loaded to the logical layer by default. To access the physical layer double click on the table **Orders** and it will drill down to the physical layer.
4. To know you are in the right layer the Orders table will have a blue line on the left of the container.

TIP: To create clarity when viewing your data model, add the finest grain table to the data source canvas first, and then relating all other tables to that first table.

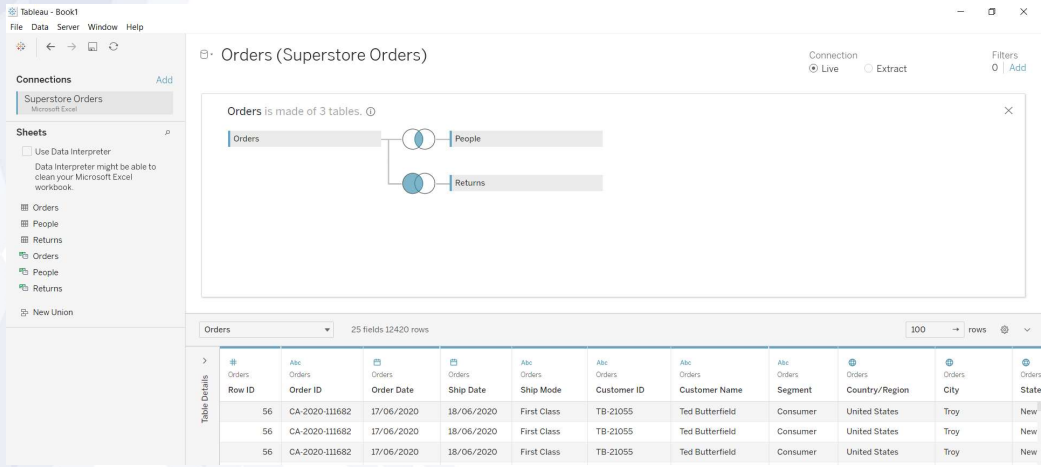
The Physical Data Model



Lets create the data model for Orders that will allow us to answer questions about manager performance

1. Join the People to the Orders dataset so that we can get information on who are the regional managers
    I. Drag the People dataset to the Physical Layer
    II. Since both tables have a Region dimension, Tableau automatically generates a join between the two tables.
    III. An inner join is automatically selected, because the tables have a one to one relationship.
    IV. Notice the number of rows is the same as the original, **9,994** rows.

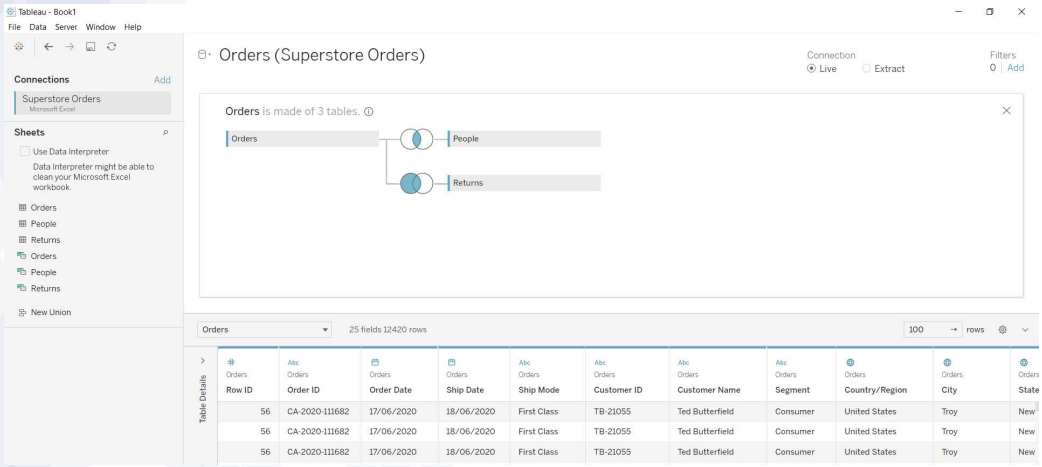
The Physical Data Model



Lets create the data model for Orders that will allow us to answer questions about the orders return ratio/return rate

1. Join the Returns to the Orders dataset so that we can get information on which orders were returned
    I. Drag the Returns dataset to the Physical Layer
    II. Since both tables have an Order ID dimension, Tableau automatically generates a join between the two tables.
    III. An inner join is automatically selected, However we are interested in analysing our orders and determining the return rate, therefore we shall convert this to a left-join
    IV. Notice the number of rows. -> we have **2,426** new rows from the original **9,994** rows

The Physical Data Model

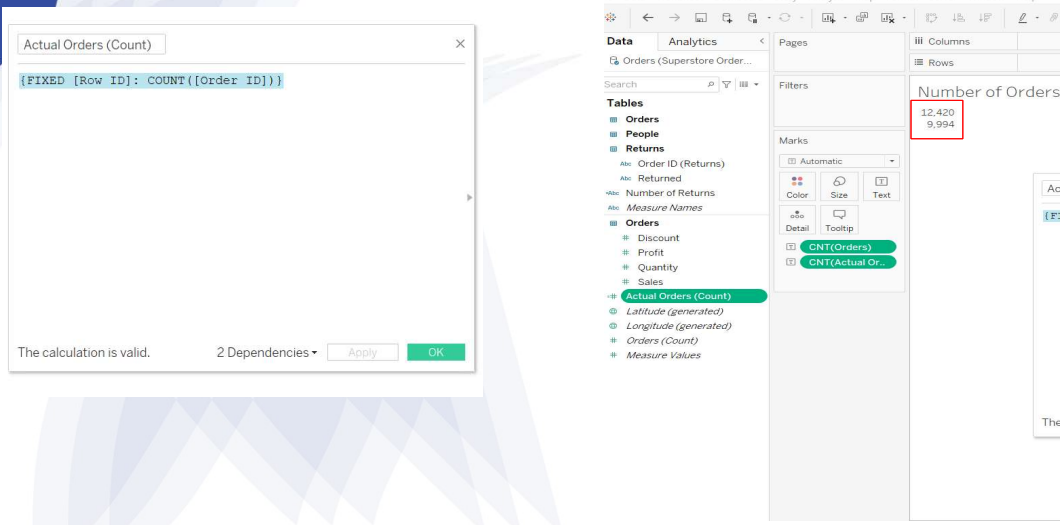


The Problem:

- The returns table tells you which **product** was returned at the **Order ID** level. Therefore, if multiple products were returned you have multiple OrderIDs (one for each product retuned)
- Additionally in the Orders table we have multiple rows per Order ID in the case where multiple products were purchased.

Therefore: *Tableau join causes duplicate rows because it does not know which returned Order ID belongs to the Order ID therefore joins the records in the Returns dataset to each row in the Order ID. For example Order ID CA-2020-111682 has seven products all of which were returned. On joining we have 49 rows because each product in* ***Orders*** *is joined with all rows in* ***Returns*** *table that corresponds to its Order ID. This is problematic as it inflates the sales of a given product.*

The Physical Data Model



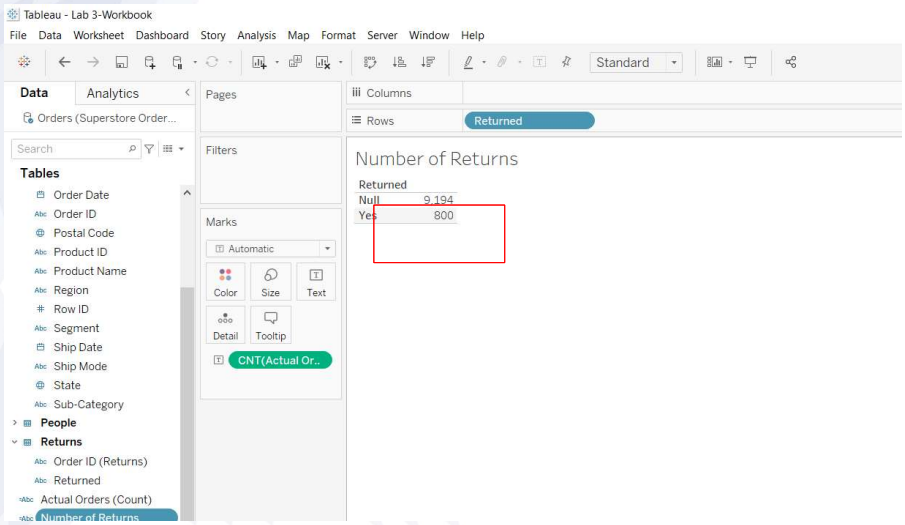
Solution:

- Outside Tableau: Normalise the dataset by creating a Product table with a unique Product ID table that can be used as an additional unique (foreign) key in the Returns dataset
- Tableau expressions to create a calculated field that uses a FIXED LOD (level of detail) expression to isolate one value for the most granular level of detail in the join

Number of Actual Orders

1. Select the dimension or set of dimensions that is unique for each row value in the data source is unique for each row value. In this case it the **Row ID**.
2. Analysis > Create Calculated Field
3. {FIXED [Row ID]: COUNT([Order ID])} > OK
4. Save as Actual Orders
5. Drag new measure to the text Mark.

The Physical Data Model

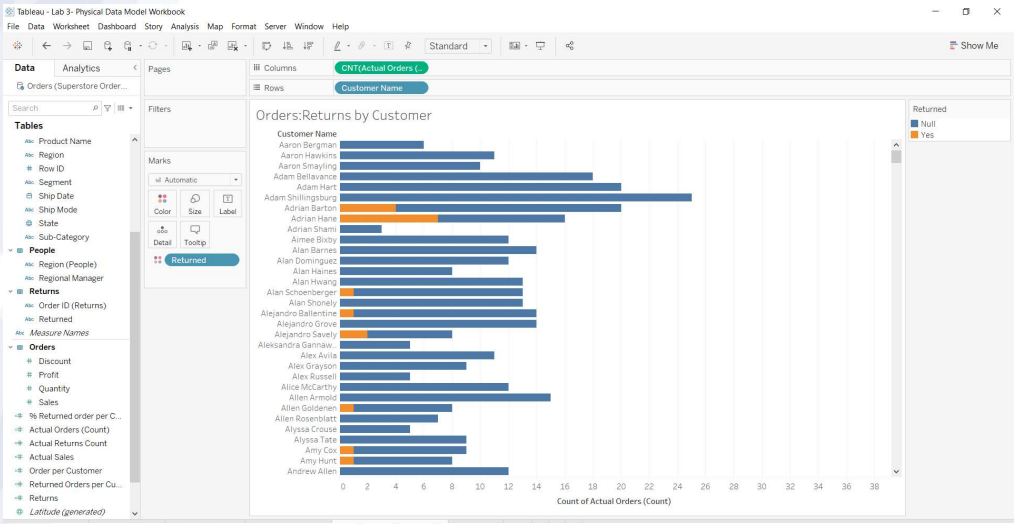


Using this field we have the same number of returns as in the excel worksheet.

But remember, This sheet also contained duplicate Order IDs for returns that had more than one product.

Therefore we need another level of calculation for the unique Orders with returns.

The Physical Data Model



These new calculated filed can now be used to analyse the ratio of orders to returns:

→ ***For instance we can visualise the order: return ration for each customer for promotions or for further customer service interventions.***

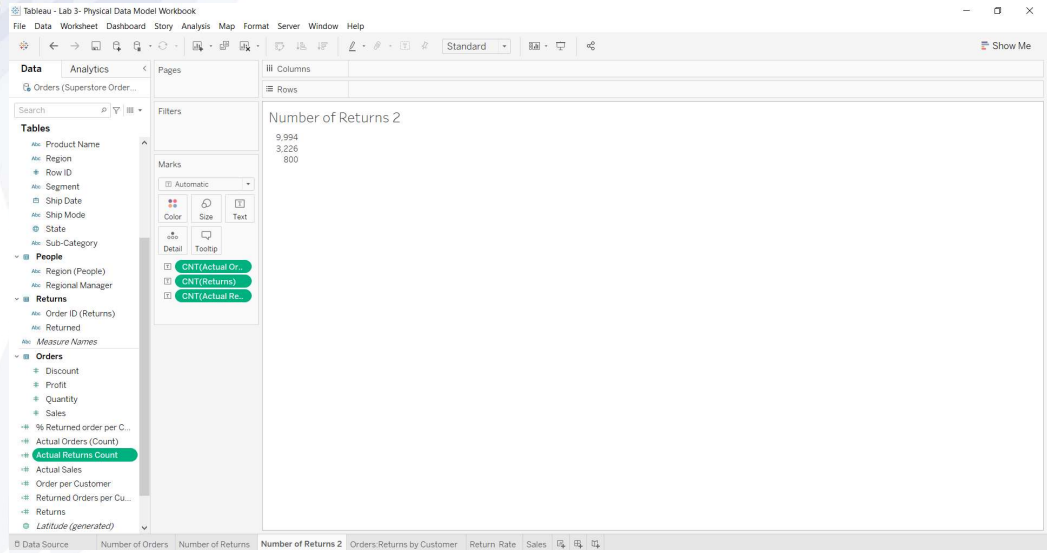
To do this:

1. Add customer name to the row shelf
2. For the columns, drag the new calculated field: Actual Orders (Count) to the column shelf

Now we have visualised the number of Actual Orders made by each customer

3. To visualise returns drag the Returned filed to the colour mark.

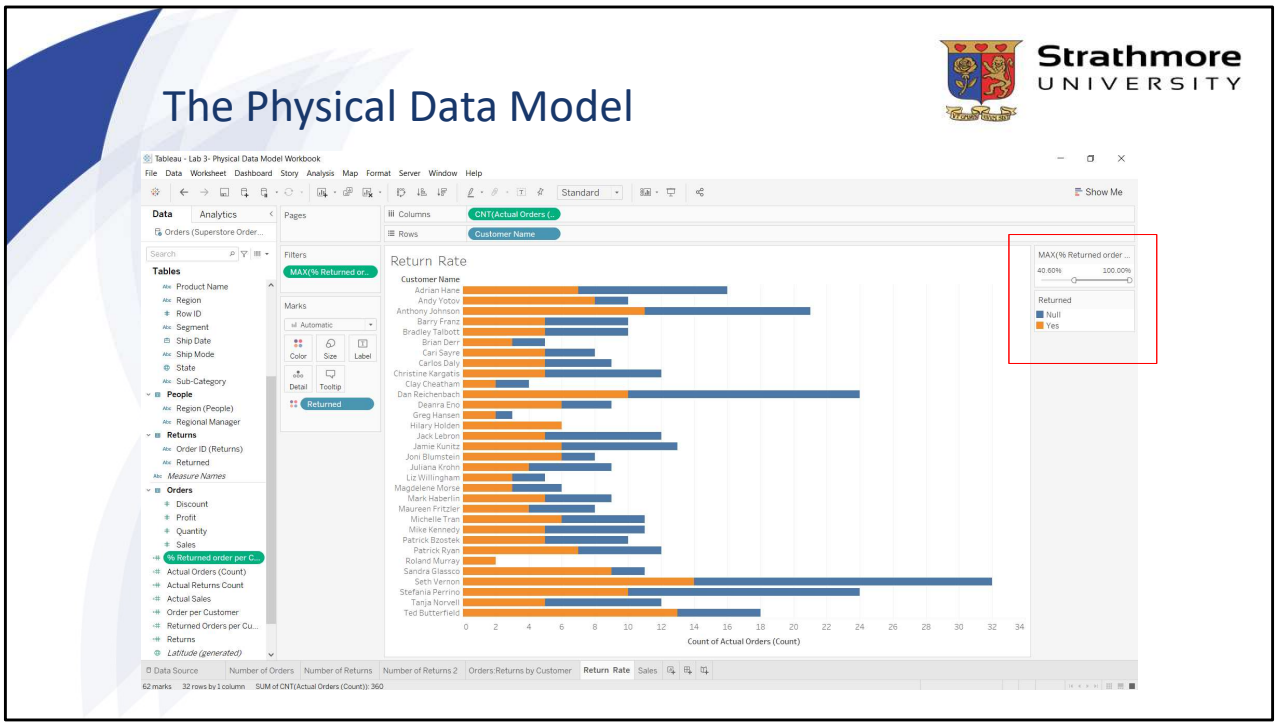
The Physical Data Model



To get the return rate for each customer we will need new calculations:

1. Create a calculate field to indicate whether an item was returned or not. (IIF([Returned]= "Yes",1,0) > 0
2. We want to get a unique count of returns. We can fix this at the row level just like we did with orders.
3. Analysis > Create Calculated Field > { FIXED [Row ID]: MIN([Returns])} > Name as Actual Returns(Count) > OK … ***If you use an aggregation of MIN, you will get one tally per Order ID.***
4. Drag new measure to the text Mark.
5. Change it to a count by selecting the drop down in the pill Select Measure> Count
6. The count of unique returns in this case is 800 products had been returned.

The Physical Data Model

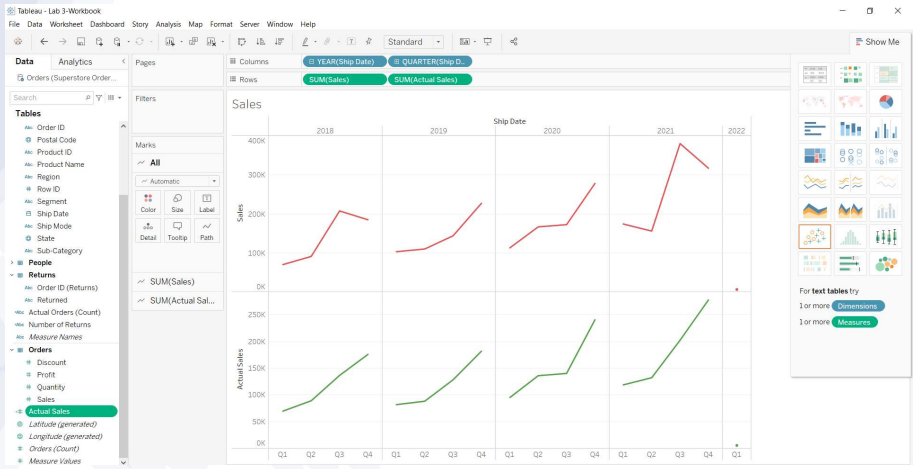


Calculating the % Returned for each customer requires more finessing.

1. Create a calculated field by selecting **Analyse** > **Create Calculated Field** > **{fixed [Customer Name]: COUNT([Actual Returns Count])/COUNT([Actual Orders (Count)])}**
2. This fixes the count of returns at the customer level to get returns for each customer. Name this field **as % Returned order per Customer** > **OK**
3. To format as a percentage, right click the new **% Returned order per Customer** field and **Default Properties** > **Number Format** > **Percentage**
4. Now we have the % Return Rate for each customer. It can be used as a filter to see which customers have a high return rate for further actioning.

- Duplicate the Orders: Return by customer and save as Return Rate
- Right click the new **% Returned order per Customer** and select **Show Filter**
- Adjust the filter to view customers of interest

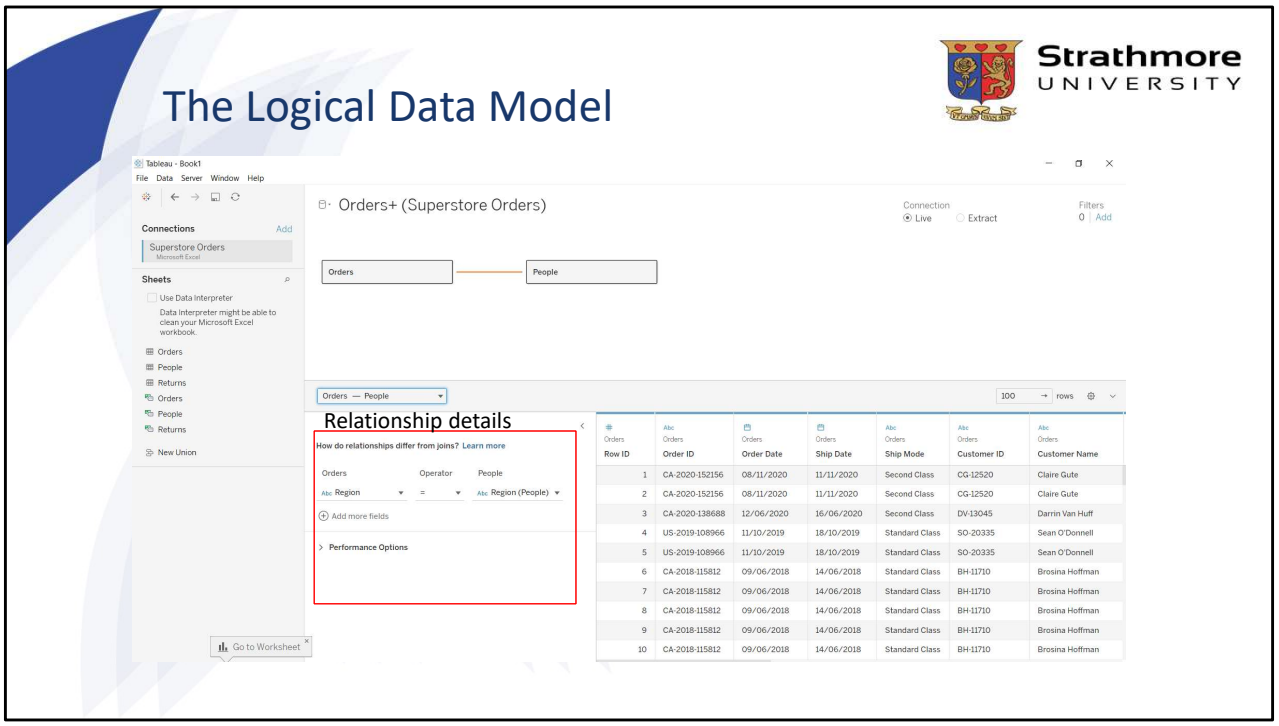
The Physical Data Model



Any other analysis needs a LOD definition to get accurate counts.

For instance: Time series of Sales will need:

1. New calculation for deduplicated sales
2. Follow the same procedure:
    1. Analysis > calculated field
    2. Enter the formula {FIXED [Row ID]: MIN([Sales])}
    3. Save as Actual Sales



Data modelling in the logical layer users “relationships” to link the different datasets.

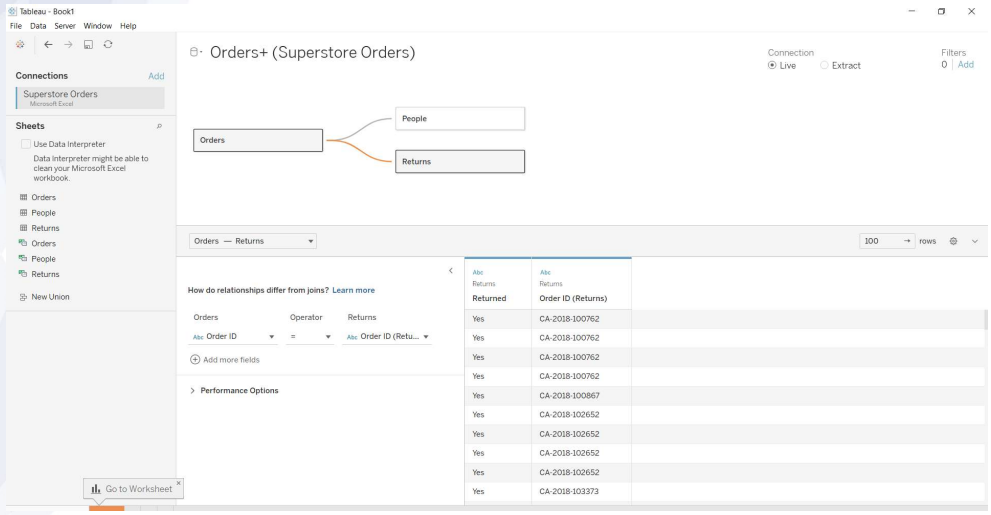
1. In a new workbook, connect to the same Superstore Excel Workbook
2. Drag the Orders table to the logical data layer canvas to begin building the model.

NOTE: *Drag the fact table into the model first and then relate the dimension tables to the fact table (in a star schema) or to other dimension tables (in a snowflake).*

3. Drag the People dataset to the workspace and notice a new link is created between the two datasets but instead of a join type (symbolised by a Venn diagram) you have a relationship defined (orange noodle).
4. When a table is added to the canvas, an orange line appears between the two tables indicating a relationship is being created. If you do not see the orange line, drag the dimensions table closer to the table till it appears.
5. Click on the noodle to view the relationship details in the Task pane below. Notice just like join, the relationship is created using the field the tables have in common. *You can add more fields to better define the relationship. You can also change the cardinality and the referential integrity of the relationship to improve performance (but it is best to leave this as default)*

The difference between the relationship and join is that the datasets are not merged to a single flat file. The tables remain independent at this point but a loose connection is defined to be used when Tableau generates the visualization-specific SQL statement.

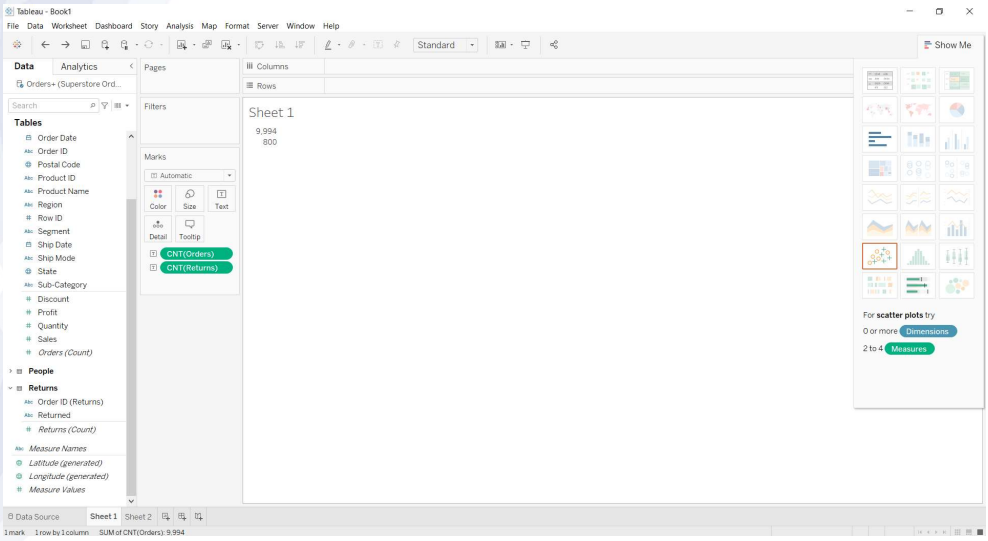
The Logical Data Model



Now for returns. We know that the returns data is liked to orders:

1. Drag and drop the Returns dataset to the workspace.
2. A new relationship is formed between Orders and Returns. Looking at the details in the Task pane, the relationship is formed based on Order ID which we expected. From this pane you can add more field to the relationship as required.
3. Notice again the datasets remain independent.
4. *To view the SQL generated for each operations we will activate the performance monitoring tool in Tableau.*
    1. *Help > Settings and Performance > Start Performance Recording*
5. The model is now done and we can start building a new visualisations. Click **Sheet 1** or Go to **Worksheet** to begin.

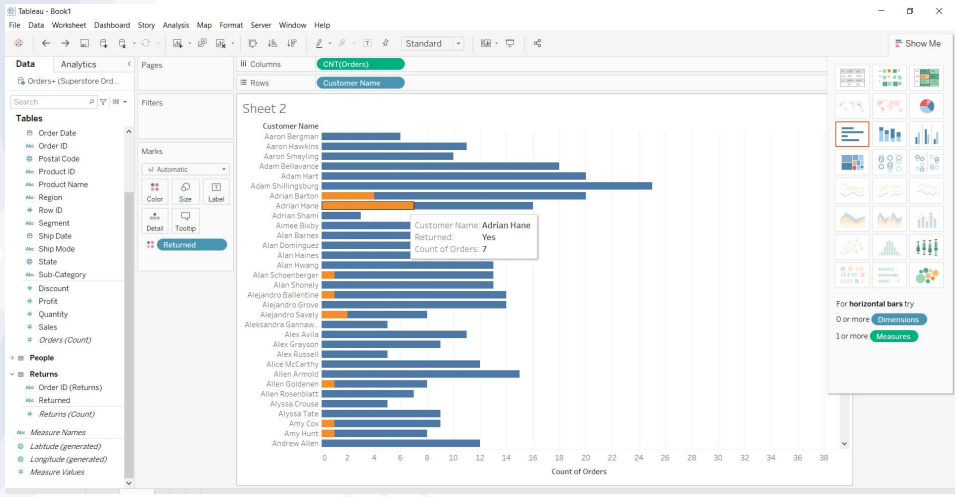
The Logical Data Model



The worksheet looks different:

- Each table has its own view with separate Measures and dimensions
- Calculations that use one dataset will appear under the specific dataset section.
- Calculations that use more than one dataset appear at the bottom of the view.
- If we drag the orders (count) to the Text Mark, we get the Actual count of Orders
- If we drag the Returns (count) to the Text Mark, we get the Actual count of Returns

The Logical Data Model

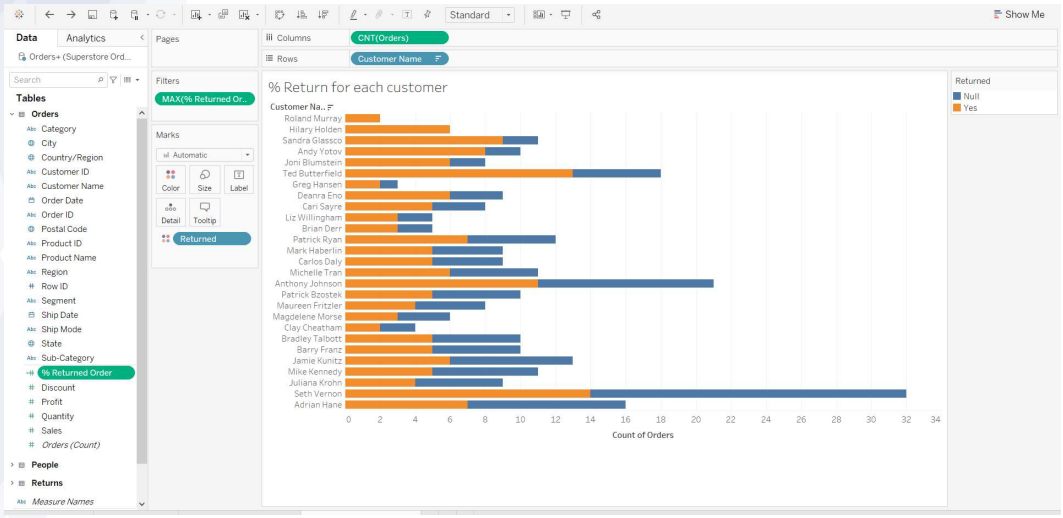


Visualising customer returns now is easier and more intuitive:

1. Drag Customer Name to the input shelve to get a list of customers
2. Drag Orders (count) to get the order counts by customers
3. Drag the Returned field to the Colour Marks to get the ratio of Orders: Returns visualised for each customer.

Joining tables using relationships makes it more intuitive and easier to handle complex joins without the need for LOD calculations.

The Logical Data Model



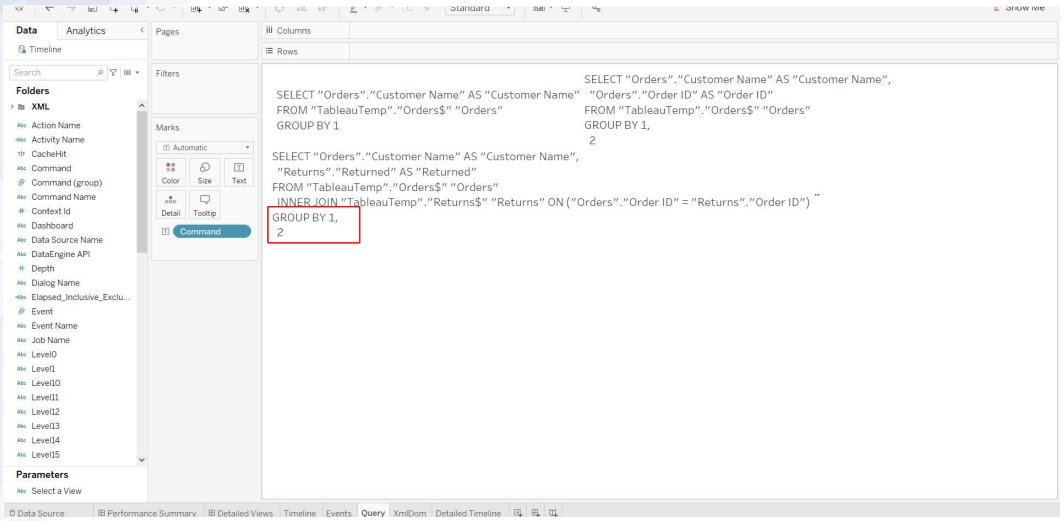
But it is not the last you will see of Fixed calculations. For instance, to calculate the ratio Order to Return per customer we fill need to fix the level of detail to Customer and not Order.

To do this we:

1. Create a calculated field
2. In the formula pane add { FIXED [Customer Name]:COUNT([Returns])/COUNT([Orders])} and click ok
3. Save this field as % Returned Orders
4. This value can be used to filter the view to get the customers with a 50% return rate for example.
    1. Drop the new % Returned orders in the Filter Card.
    2. Select the filtering method as Max
    3. You can use the range of values to select the maximum and minimum % Return Orders. More the range to a value of choice let say 50%-100% return rate.
    4. Click OK

You now have a view of customers who have more than 50% order Return Rate for actioning

The Logical Data Model



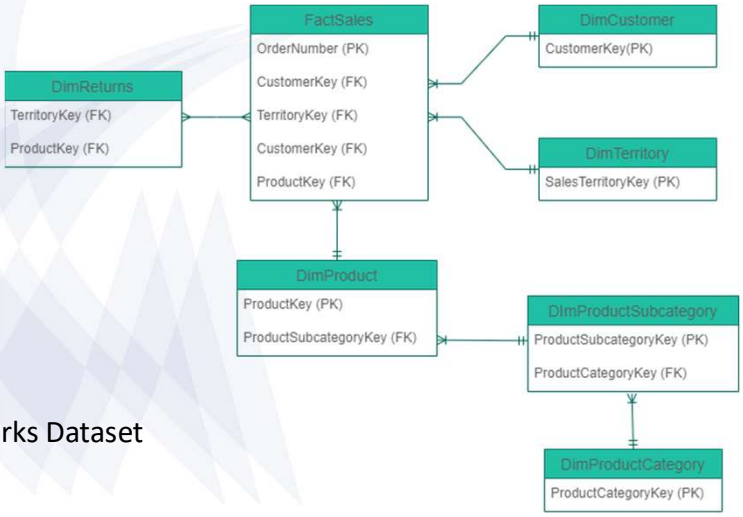
Stop the Recording to view the generated SQL statements

Help > Settings and Performance > Stop Performance Recording

- Tableau generates the SQL statement to group the required tables for each visualisations.

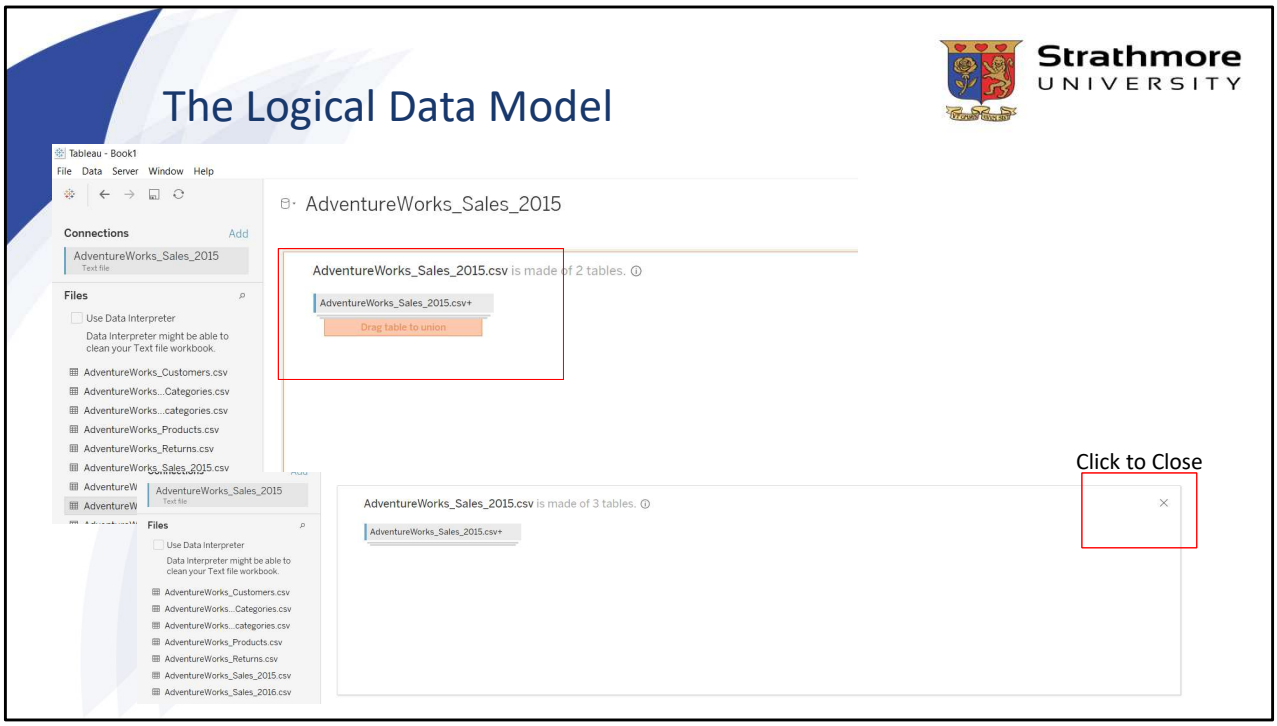
- ***The SQL does the group by that takes care of the issues with duplications***.

More Practice….



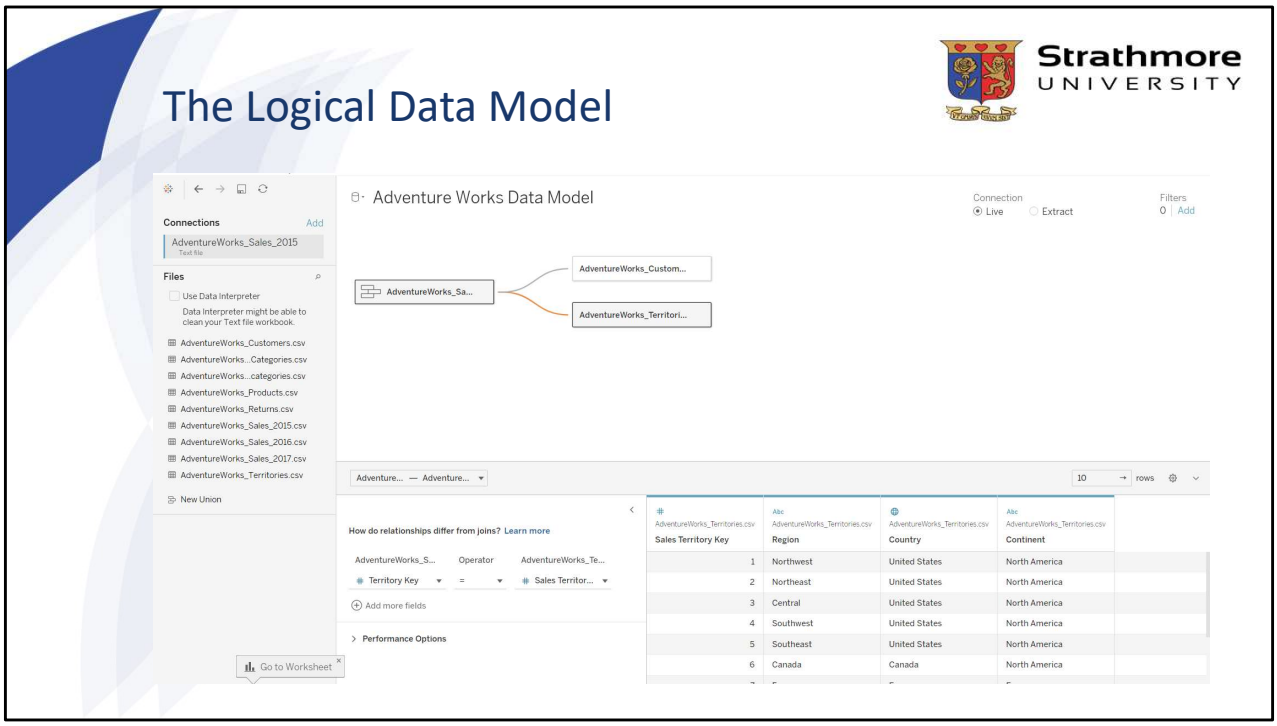
Adventure Works Dataset

The Logical Data Model



1. Create connection to the sales dataset and drag the Sales 2015 dataset onto the canvas. It is best practice to bring the fact table onto the canvas first
2. We need to combine the sales data for the three years
3. Double click to access the physical layer to perform a union
4. Drag the 2016 Sales dataset to the workspace and drag table to union
5. Do the same for the 2017 and 2018
6. Close the physical layer to exit back to the logical layer

The Logical Data Model



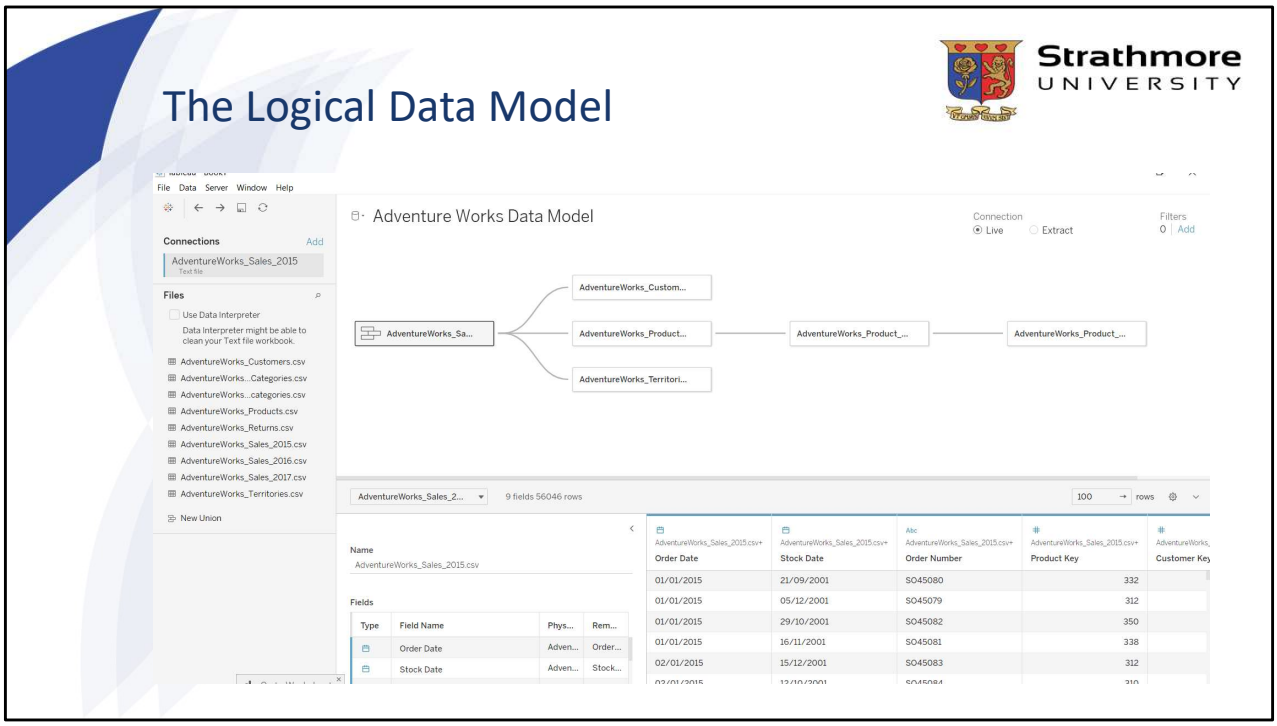
Create relationships between the Sales fact table and the Customer and Territory dimensions table

1. Drag the customer dataset onto the canvas.
2. Ensure that Tableau recognized the correct Keys in the **Edit Relationship** Window.
3. Repeat this process for the dimTerritory table.
4. Notice you will get an error because Tableau is not able to match the two keys needed for the relationship
5. Fix this by selecting the right key that is Territory Key in in the Sales and Sales Territory Key in the Territory table

If no constraints are detected, a Many-to-many relationship is created, and referential integrity is set to Some records match.
These default settings are a safe choice and provide the most a lot of flexibility for your data source. The default settings support full outer joins and optimize queries by aggregating table data before forming joins during analysis. All column and row data from each table becomes available for analysis.

Once complete the canvas should look like the one shown above.

The Logical Data Model

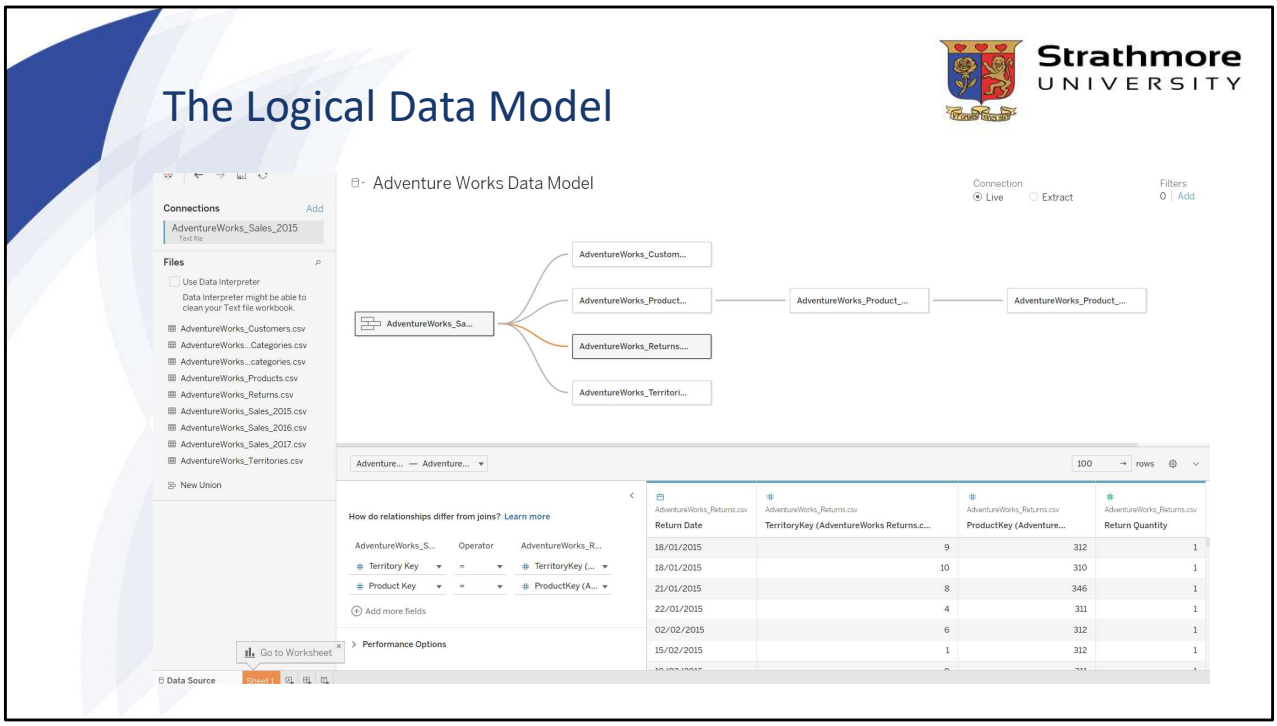


Lets create the relationship between the tables that make up the product hierarchy (Product, Product Sub Category, Product Category). Tableau recommends using relationships by default and only using joins when it is absolutely necessary.

1. Create a relationship between Product and the Sales fact table
2. Create a relationship between the Product Subcategory and Product
3. Create a relationship between Product Category and Product Subcategory

Once complete the canvas should look like the one shown above.

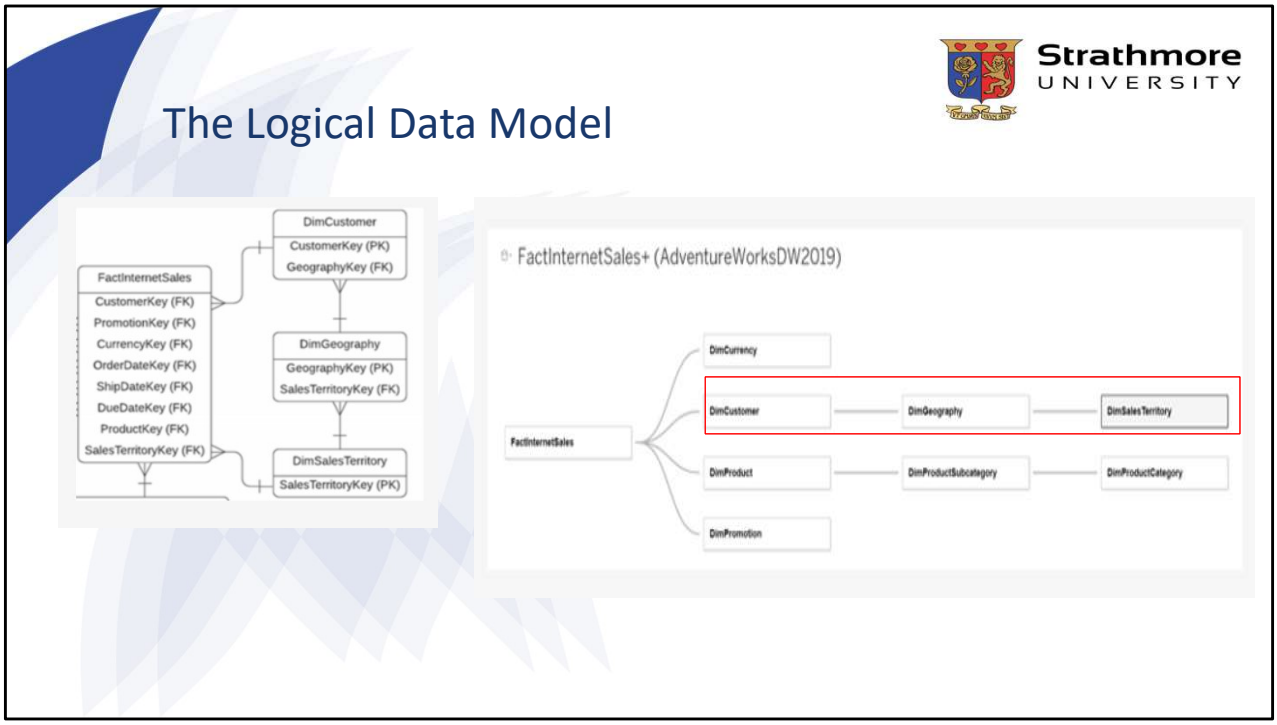
The Logical Data Model



Recall from the data model that the Returns dimension table has a composite key comprising of a the Territory Key and the Product Key

1. Drag the returns table to the canvas to create a relationship with the sales dataset
2. Select the appropriate fields in the Edit Relationship window as displayed above.

The Logical Data Model



Let say we had a geography table that connects to the Sales fact table and the Territory dimension table. The additional geography table allows for the analysis of the States making up each Territory.

To recreate this in the data model, we would have added the Geography table twice to connect to Sales and Sales territory. However this would cause confusion during analysis by having two geography tables. In this instance it is easier to leave out the direct connection to Sales and connect to geography instead.

Read more: https://technologyblog.rsmus.com/data-analytics/implementing-a-data-model-in-tableau-as-a-single-data-source/

Limitations Tableau Data Table

Strathmore UNIVERSITY

- Circular relationships are not supported
- Directly relating 3 or more fact tables on shared dimensions

In some use cases it is common to have multiple fact tables related to multiple shared dimension tables. For example, you might have two fact tables, Store Sales and Internet Sales, related to two common dimension tables, Date and Customer. Typically, such scenarios would require creating a circular relationship in your data model.

The solution is to duplicate the dimension table for each fact table **OR** You can approximate this type of model by merging some of the tables in the physical layer. For example, you might be able to union Store Sales and Internet Sales into a single table, which can then be related to Date and Customer **OR** you might be able to cross-join Date and Customer to create a single dimension table which can then be related to Store Sales and Internet Sales.

While it is possible to build this model in the logical layer, you might see unwanted results, unless you only use dimensions from a single table

Read more on the requirements: https://help.tableau.com/v2020.2/pro/desktop/en-us/datasource_datamodel.htm

More Resources

Strathmore UNIVERSITY

- Multiple fact tables - https://help.tableau.com/v2020.2/pro/desktop/en-us/datasource_datamodel.htm
- More capability- https://www.tableau.com/about/blog/2020/5/relationships-part-1-meet-new-tableau-data-model
- Watch and Learn - https://www.youtube.com/watch?v=75zvHrnuPr0 , https://www.youtube.com/watch?v=G_1uRWwX3Xg
- SQL enthusiasts - https://www.flerlagetwins.com/2020/05/tableau-data-model.html
- Powe BI vs Tableau data models - https://dataveld.com/2020/06/18/four-core-differences-between-the-tableau-and-power-bi-data-models/

Strathmore UNIVERSITY

Thank you!

Any Questions?