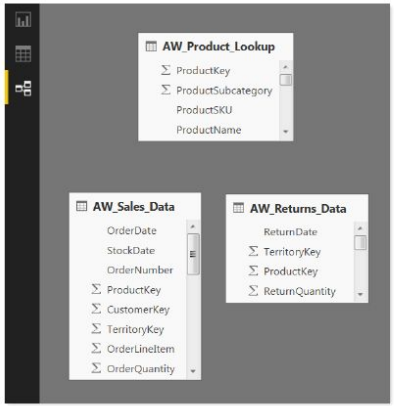
Power BI training

Power BI

Turn your data into Impact!

Power BI

What is a data model?

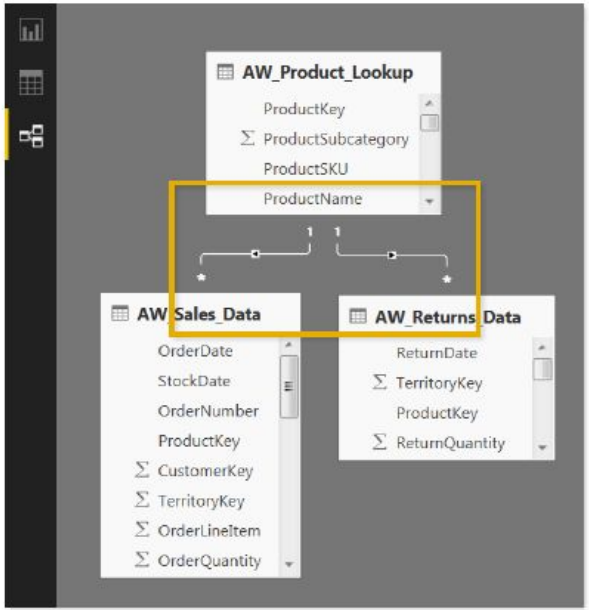


This is not a Data Model

Tables are not connected using Primary or Foreign Key

ProductName	OrderQuantity	ReturnQuantity
All-Purpose Bike Stand	84,174	1,828
AWC Logo Cap	84,174	1,828
Bike Wash - Dissolver	84,174	1,828
Cable Lock	84,174	1,828
Chain	84,174	1,828
Classic Vest, L	84,174	1,828
Classic Vest, M	84,174	1,828
Classic Vest, S	84,174	1,828
Fender Set - Mountain	84,174	1,828
Total	**84,174**	**1,828**

This is Data Model



Tables are connected using Relationships between Primary or Foreign Key.

Related Tables can used data from each other and generate output like -

ProductName	OrderQuantity	ReturnQuantity
All-Purpose Bike Stand	234	8
AWC Logo Cap	4,151	46
Bike Wash - Dissolver	1,706	25
Classic Vest, L	182	4
Classic Vest, M	182	7
Classic Vest, S	157	8
Fender Set - Mountain	3,960	54
Half-Finger Gloves, L	840	18
Half-Finger Gloves, M	918	16
Total	**84,174**	**1,828**

INTRODUCTION OF DATA TABLES - PREPARATION

Connect the below Data Tables with Power BI and check for the data types -

File Name (Original)	Changed Name (in Power BI)
1. dimCptCode.csv	1. CPTCode_LookUp.csv
2. dimDate.csv	2. dimDate.csv
3. DimDiagnosisCode.csv	3. DiagnosisCode_LookUp.csv
4. DimHospital.csv	4. Hospital_LookUp.csv
5. DimPatient.csv	5. Patient_LookUp.csv
6. DimPayer.csv	6. Payer_LookUp.csv
7. DimPhysician.csv	7. Physician_LookUp.csv
8. DimSpeciality.csv	8. Speciality_LookUp.csv
9. DimTransaction.csv	9. Transaction_LookUp.csv
10. FactTable.csv	10. FactTable.csv

IMPORTANT - Make sure that the name of the Primary Key columns in the 'Look Up' Tables and Fact Tables are same for easy identification which creating the Data Model. Also make sure that in the Data Model Section no relationship is created by PBI automatically. If created remove these relations as we are going to learn how to create the relationship.

Power BI

DATABASE NORMALIZATION

Normalization is the process of organizing the tables and columns in a 'Relational' database. The purpose is to 'eliminate redundancy' and 'preserve data integrity'. This helps reducing the table sizes and improve the processing speed especially with big data. Benefits of normalization are -

a) **Eliminate redundant data** to decrease table sizes and improve processing speed and efficiency
b) **Minimize errors and anomalies** when you modify data (i.e. insert update or delete records)
c) **Simplify queries** and structure the database for meaningful analysis

Important - Each table should serve as **distinct** and **specific purpose** (i.e. product information, dates, sales transactions, payer details, provider details, hospital details etc.)

Let us discuss an example (in excel)

date	product_id	quantity	product_brand	product_name	product_sku	product_weight
1/1/1997	869	5	Nationeel	Nationeel Grape Fruit Roll	52382137179	17
1/7/1997	869	2	Nationeel	Nationeel Grape Fruit Roll	52382137179	17
1/3/1997	1	4	Washington	Washington Berry Juice	90748583674	8.39
1/1/1997	1472	3	Fort West	Fort West Fudge Cookies	37276054024	8.28
1/6/1997	1472	2	Fort West	Fort West Fudge Cookies	37276054024	8.28
1/5/1997	2	4	Washington	Washington Mango Drink	96516502499	7.42
1/1/1997	76	4	Red Spade	Red Spade Sliced Chicken	62054644227	18.1
1/1/1997	76	2	Red Spade	Red Spade Sliced Chicken	62054644227	18.1
1/5/1997	3	2	Washington	Washington Strawberry Drink	58427771925	13.1
1/7/1997	3	2	Washington	Washington Strawberry Drink	58427771925	13.1
1/1/1997	320	3	Excellent	Excellent Cranberry Juice	36570182442	16.4

When you do not normalize you end up with tables with duplicate information. This really starts impacting the efficiency especially when when you are dealing with big data

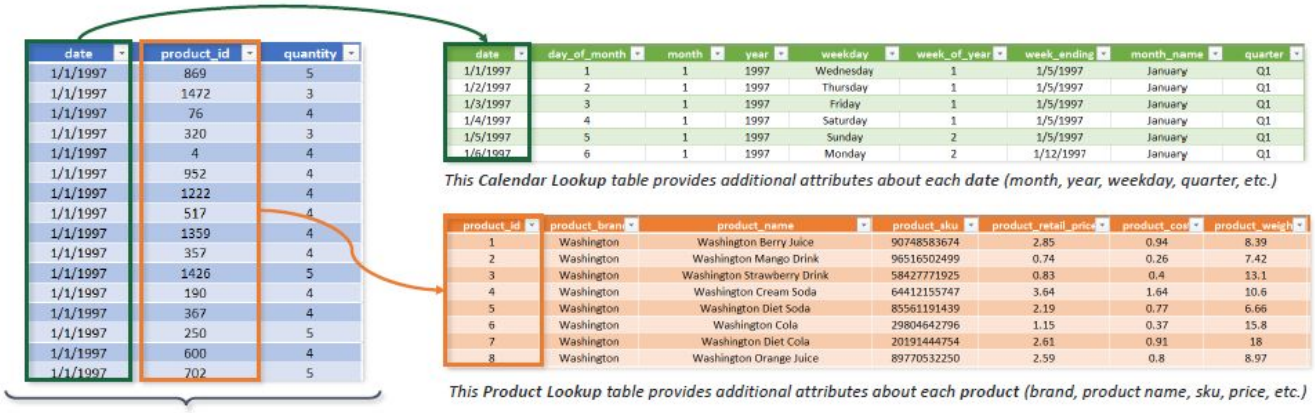
Power BI

DATA TABLES AND LOOK-UP TABLES

Data Model contains 2 types of tables -

Data Tables (are fact tables) - Contains numbers, values, transactions (facts). They are granular tables with 'ID' or 'Key' fields which is used to create relationship between them and the 'Look up Tables' e.g. Sales table,, Returns table

Look Up Tables (also called dimension table) contains descriptions and details which are usually text based attributes about dimensions in the data e.g - Customer, Products, Region etc.



date	product_id	quantity
1/1/1997	869	5
1/1/1997	1472	3
1/1/1997	76	4
1/1/1997	320	3
1/1/1997	4	4
1/1/1997	952	4
1/1/1997	1222	4
1/1/1997	517	4
1/1/1997	1359	4
1/1/1997	357	4
1/1/1997	1426	5
1/1/1997	190	4
1/1/1997	367	4
1/1/1997	250	5
1/1/1997	600	4
1/1/1997	702	5

This Data Table contains "quantity" values, and connects to lookup tables via the "date" and "product_id" columns

date	day_of_month	month	year	weekday	week_of_year	week_ending	month_name	quarter
1/1/1997	1	1	1997	Wednesday	1	1/5/1997	January	Q1
1/2/1997	2	1	1997	Thursday	1	1/5/1997	January	Q1
1/3/1997	3	1	1997	Friday	1	1/5/1997	January	Q1
1/4/1997	4	1	1997	Saturday	1	1/5/1997	January	Q1
1/5/1997	5	1	1997	Sunday	2	1/5/1997	January	Q1
1/6/1997	6	1	1997	Monday	2	1/12/1997	January	Q1

This Calendar Lookup table provides additional attributes about each date (month, year, weekday, quarter, etc.)

product_id	product_brand	product_name	product_sku	product_retail_price	product_cost	product_weight
1	Washington	Washington Berry Juice	90748583674	2.85	0.94	8.39
2	Washington	Washington Mango Drink	96516502499	0.74	0.26	7.42
3	Washington	Washington Strawberry Drink	58427771925	0.83	0.4	13.1
4	Washington	Washington Cream Soda	64412155747	3.64	1.64	10.6
5	Washington	Washington Diet Soda	85561191439	2.19	0.77	6.66
6	Washington	Washington Cola	29804642796	1.15	0.37	15.8
7	Washington	Washington Diet Cola	20191444754	2.61	0.91	18
8	Washington	Washington Orange Juice	89770532250	2.59	0.8	8.97

This Product Lookup table provides additional attributes about each product (brand, product name, sku, price, etc.)

Power BI

PRIMARY KEYS AND FOREIGN KEYS

date	product_id	quantity
1/1/1997	869	5
1/1/1997	1472	3
1/1/1997	76	4
1/1/1997	320	3
1/1/1997	4	4
1/1/1997	952	4
1/1/1997	1222	4
1/1/1997	517	4
1/1/1997	1359	4
1/1/1997	357	4
1/1/1997	1426	5
1/1/1997	190	4
1/1/1997	367	4
1/1/1997	250	5
1/1/1997	600	4
1/1/1997	702	5

date	day_of_month	month	year	weekday	week_of_year	week_ending	month_name	quarter
1/1/1997	1	1	1997	Wednesday	1	1/5/1997	January	Q1
1/2/1997	2	1	1997	Thursday	1	1/5/1997	January	Q1
1/3/1997	3	1	1997	Friday	1	1/5/1997	January	Q1
1/4/1997	4	1	1997	Saturday	1	1/5/1997	January	Q1
1/5/1997	5	1	1997	Sunday	2	1/5/1997	January	Q1
1/6/1997	6	1	1997	Monday	2	1/12/1997	January	Q1

product_id	product_[illegible]	product_name	product_sku	product_retail_pric[illegible]	product_cos[illegible]	product_weigh[illegible]
1	Washington	Washington Berry Juice	90748583674	2.85	0.94	8.39
2	Washington	Washington Mango Drink	96516502499	0.74	0.26	7.42
3	Washington	Washington Strawberry Drink	58427771925	0.83	0.4	13.1
4	Washington	Washington Cream Soda	64412155747	3.64	1.64	10.6
5	Washington	Washington Diet Soda	85561191439	2.19	0.77	6.66
6	Washington	Washington Cola	29804642796	1.15	0.37	15.8
7	Washington	Washington Diet Cola	20191444754	2.61	0.91	18
8	Washington	Washington Orange Juice	89770532250	2.59	0.8	8.97

Foreign Keys (FK) exist usually in Data (fact) tables*. They contain multiple instances and are mapped with the Primary keys from the 'Look up Tables'

Primary Keys (PK) exist usually in 'Lookup tables'*. They are Unique and are used with Foreign Keys to map the attributes/ details of the dimension

** Foreign and Primary keys can coexist in data tables also.*

Power BI

ACTIVITY ON UNDERSTANDING A DATA MODEL

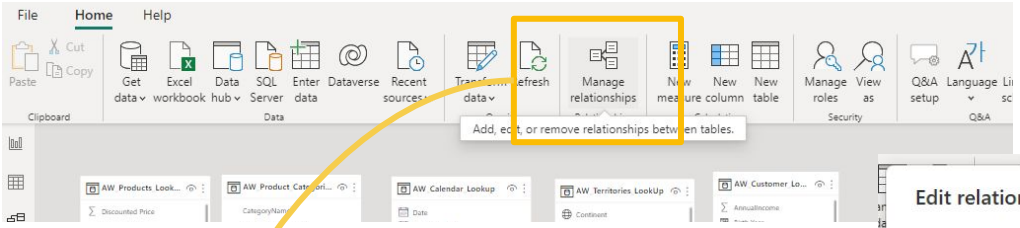
1. Go to Reports View
2. Open the Matrix template (Home -> Insert -> Matrix)
3. Click on this Icon on top of the template -> 

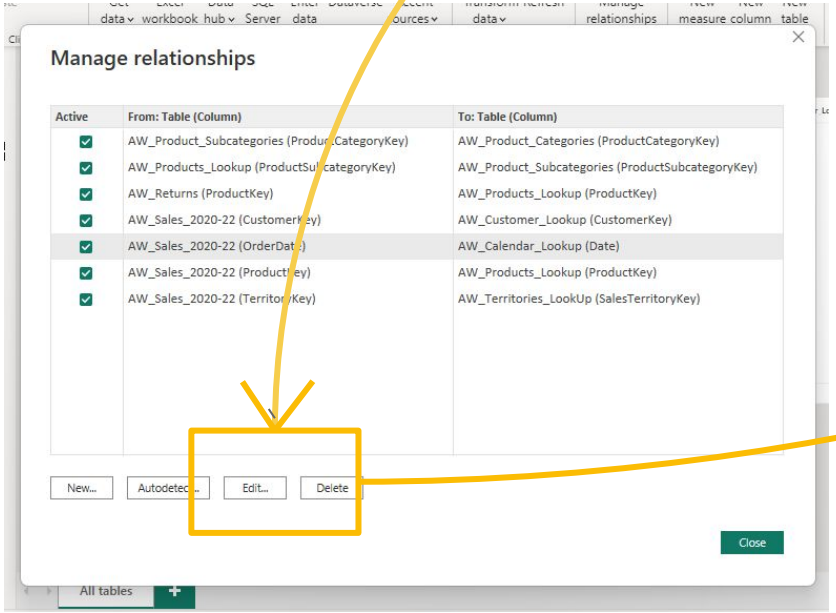
4. Bring the ‘HospitalName’ field from ‘Hospital_Lookup’ into the Rows section
5. Now bring the ‘CPTUnits’ from ‘Fact_Table’ into the Values section

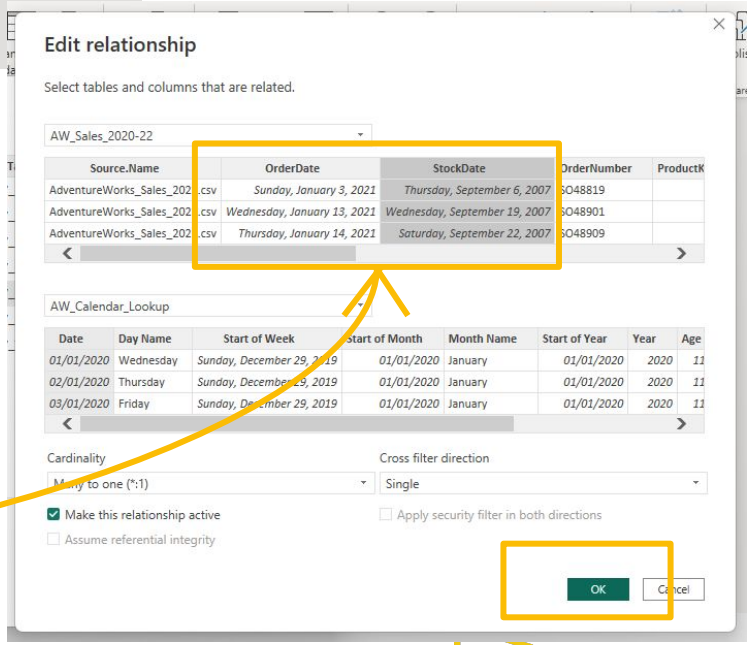
This created a Matrix.
Notice that as there is no relationship between ‘Hospital_Lookup’ table and ‘FactTable’ hence they cannot create a valid table showing the actual CPTUnits per Hospital..

Power BI

MANAGE RELATIONSHIP







Power BI

Exercise 1 - Creating Relationship

1. Place all the LookUp Tables on the top in the Data Mode section and Fact Table below them
2. Mark the Primary Key on each of the table using Properties section. PBI will create a small icon in front of the Primary Key that you identify
3. Now create Relationship between -
    - CPTCode_LookUp and FactTable
    - DiagnosisCode_LookUp and FactTable
    - DateTable and FactTable
    - Hospital_LookUp and FactTable
    - Patient_LookUp and FactTable
    - Payer_LookUp and FactTable
    - Transaction_LookUp and FactTable
    - Physician_LookUp and FactTable
    - CPTCode_LookUp and FactTable

Use slower way and also a quicker way.

Power BI

Exercise 2 - creating relationship

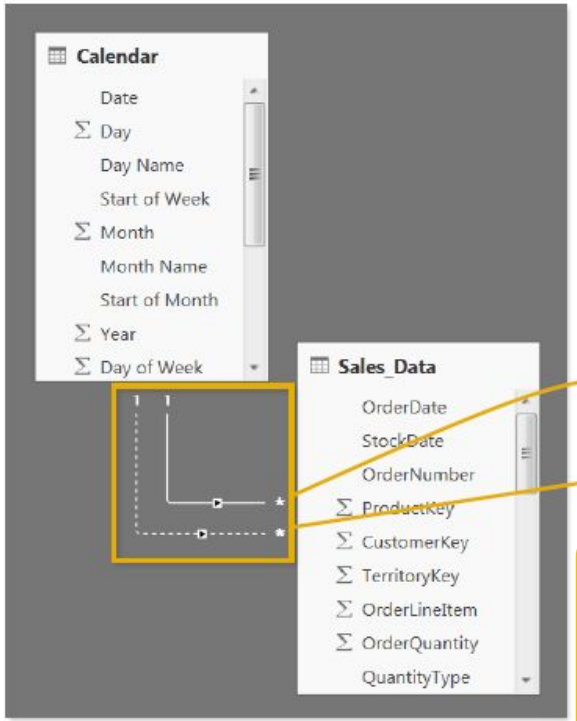
Also create relationship between -

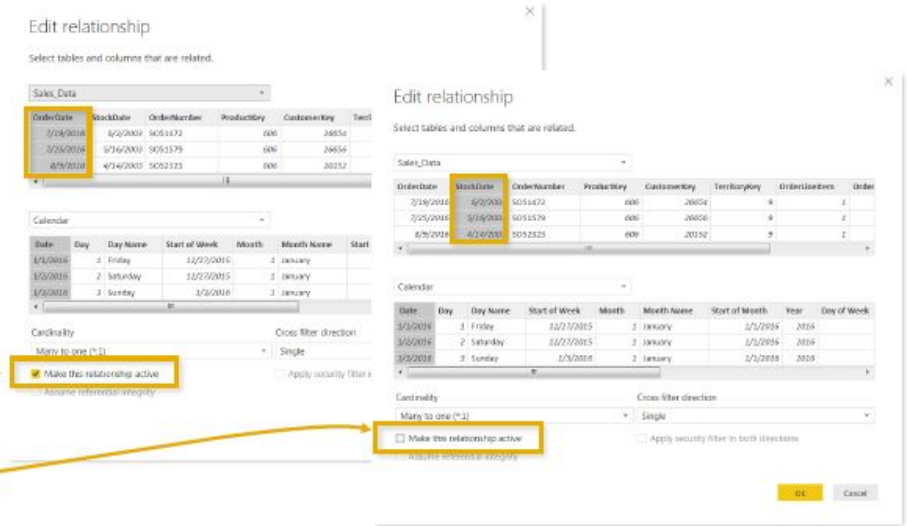
- Physician_LookUp and Speciality_LookUp

Use quicker way.

Power BI

ACTIVE VS INACTIVE RELATIONSHIP





Only one Relationship can be active at a time.
Sales Table has 2 Dates field (Order Date and Stock Date) but it can have a Relationship with Calendar table using only one of them at a time.
But you can switch between the active and inactive by deactivating the existing active relationship first and then activating the other

ACTIVITY ON ACTIVE AND INACTIVE RELATIONSHIP

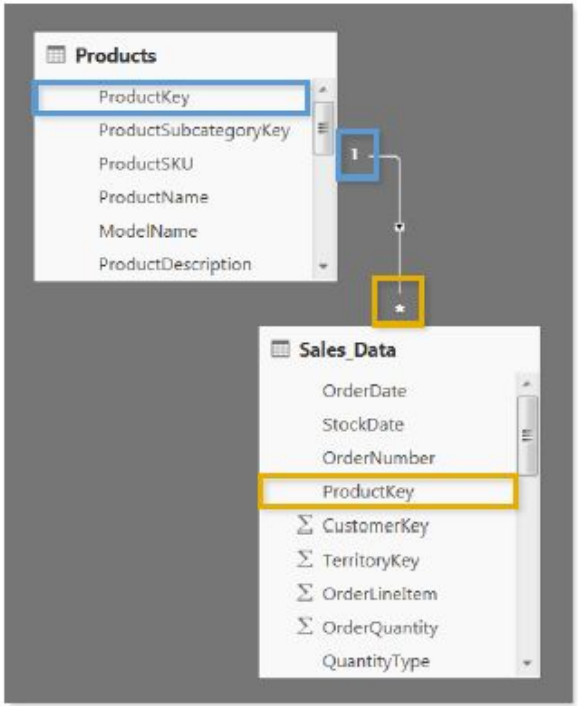
Create another relationship between ‘FactTable’ and ‘DateTable’ using ‘DateOfPosting’ in ‘FactTable’. (Note that this will be inactive as there is already an Relationship between these tables using ‘Order Date’ and Date.)

Now change the ‘Active’ status of the relationship between the tables to ‘Inactive’ and then activate the Relationship using ‘DateOfPosting’ and Date.

Then reverse the Relationship back to the original i.e. using ‘DateOfService’ (in FactTable) and Date (DateTable).

Power BI

CARDINALITY



Cardinality refers to the *uniqueness of values* in a column

- For our purposes, all relationships in the data model should follow a "**one-to-many**" cardinality; **one** instance of each *primary key*, but potentially **many** instances of each *foreign key*

*In this case, there is only **ONE instance of each ProductKey** in the Products table (noted by the "**1**"), since each row contains **attributes of a single product** (Name, SKU, Description, Retail Price, etc)*

*There are **MANY instances of each ProductKey** in the Sales_Data table (noted by the asterisk *****), since there are **multiple sales associated with each product***

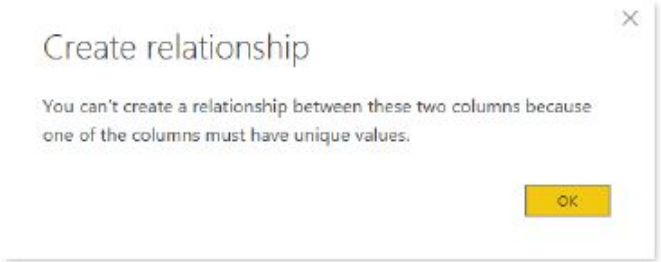
Power BI

CARDINALITY : MANY TO MANY

product_id	product_name	product_sku
4	Washington Cream Soda	64412155747
4	Washington Diet Cream Soda	81727382373
5	Washington Diet Soda	85561191439
7	Washington Diet Cola	20191444754
8	Washington Orange Juice	89770532250

date	product_id	transactions
1/1/2017	4	12
1/2/2017	4	9
1/3/2017	4	11
1/1/2017	5	16
1/2/2017	5	19
1/1/2017	7	11

Error Message



When we try to create a Relationship between the above tables we will get 'many-to-many' relationship error message.

In this case there is no way to determine the Sales (in blue table) belongs to which product

Power BI

CARDINALITY : ONE TO ONE

product_id	product_name	product_sku
4	Washington Cream Soda	64412155747
5	Washington Diet Soda	85561191439
7	Washington Diet Cola	20191444754
8	Washington Orange Juice	89770532250

product_id	product_price
4	$3.64
5	$2.19
7	$2.61
8	$2.59

Relationship between the above tables can be created using 'one-to-one' relationship but only that it is inefficient. We can always merge them can create a single table as we do not need a separate LookUp and Data table.

To eliminate the inefficiency, you could simply ***merge the two tables*** *into a single, valid lookup*

NOTE: *this still respects the laws of normalization, since all rows are unique and capture attributes related to the primary key*

product_id	product_name	product_sku	product_price
4	Washington Cream Soda	64412155747	$3.64
5	Washington Diet Soda	85561191439	$2.19
7	Washington Diet Cola	20191444754	$2.61
8	Washington Orange Juice	89770532250	$2.59

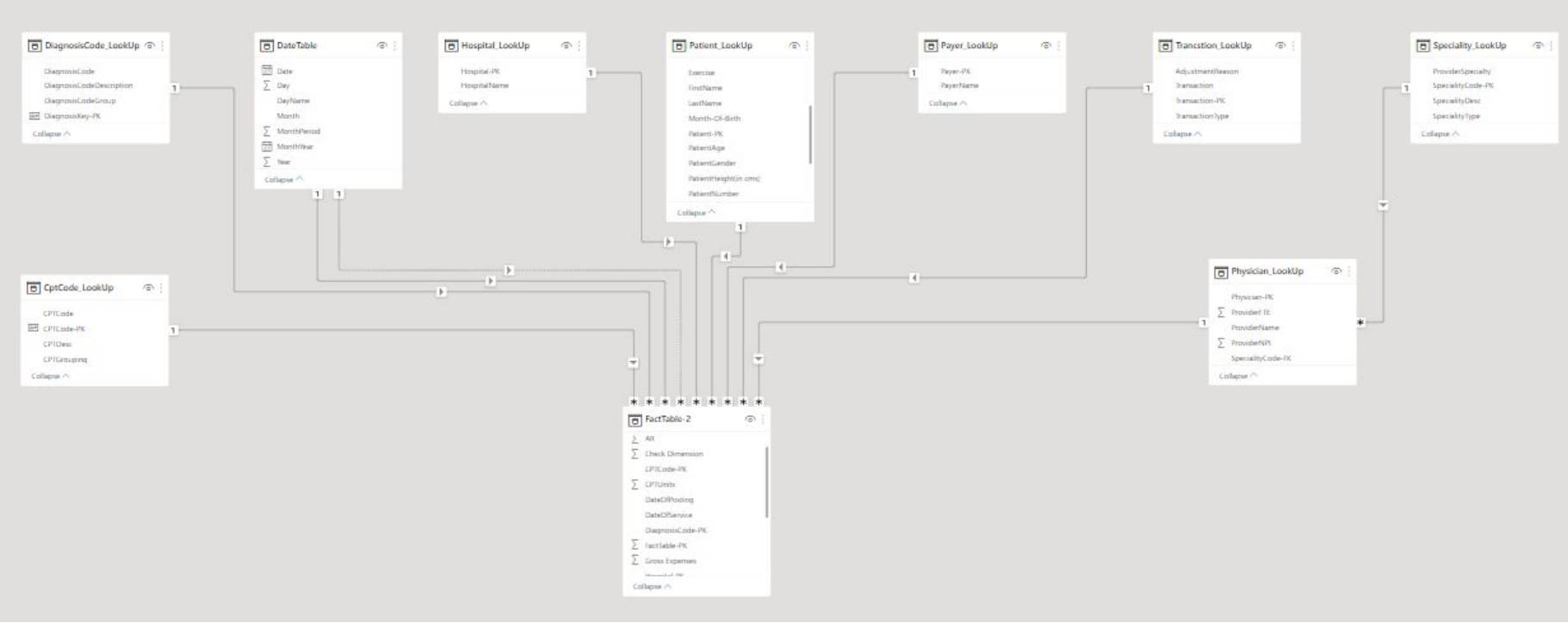
Power BI

ACTIVITY ON CARDINALITY OF RELATIONSHIP

Check the Cardinality of the Relationships created in the data model. Make sure the Relationships follow one to one Cardinality in all cases.

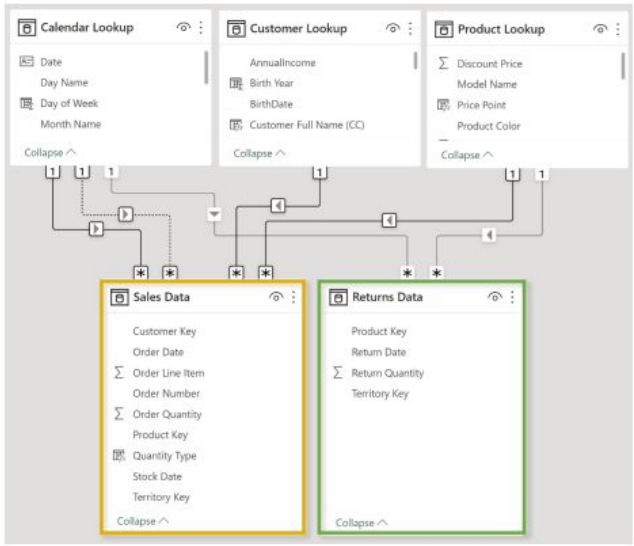
Power BI

OUR FINAL DATA MODEL - HEALTHCARE DATASET



Power BI

DEALING WITH MULTIPLE DATA TABLES



This model contains 2 Fact tables: Sales Data and Returns Data:

- You cannot create any 'direct' Relationship between Sales Table and Returns Table as there is no Primary/ Foreign Key arrangement between the two tables (or there is no 'One-to- Many' cardinality that exist)
- We can connect each fact table to the possible lookup tables. This allows us to filter both Sales and Returns data using fields from **'shared lookup tables'.**
- We can view Sales Orders and Returns Orders by Product as both the Fact Tables are linked with Product LookUp table, but we cannot view Returns by Customer since there is no Relationship that exists between Returns Table and Customer LookUp.

Important - When there are more than one Fact Tables, they should connect with each other indirectly through 'shared' LookUp Tables to be used together for visualization and analysis.

MULTIPLE DATA TABLES - ADVENTUREWORKS DATA

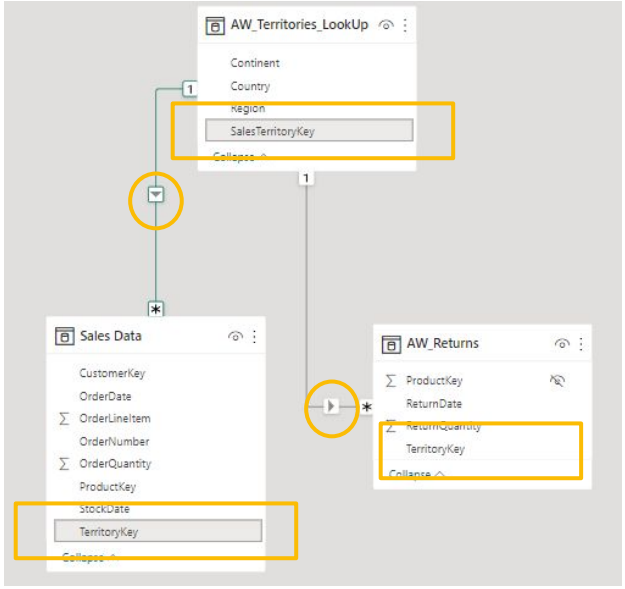
Power BI

Dataset - AdventureWorks Sales, Return, Customer, Product and Calendar data tables

Adventure Works Cycles* global manufacturing company. They have given you dataset related to Sales, Returns, Customer, Product and Calendar. You have to -

- Create a combined Sales Table for the years
- Identify the Fact Tables and LookUp Tables
- Rename them properly with 'LookUp' extension
- Identify the Primary Key in each LookUp Table
- In the Calendar Table - Add Year, Month, Day, Month Name, Week of the Year, Day of the Week columns
- Now create relationships (Data Model) between Fact Tables and LookUp Tables
- Can you create a relationship between Sales and Returns Table? Why or Why not?
- Create a Table in Reports View to Show the Sales and Returns for different Products.
- Create a Table in Reports View to Show the Sales and Returns for different Customers.

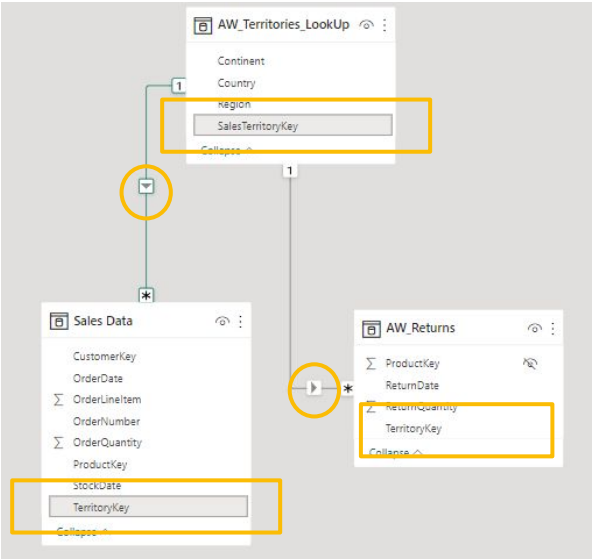
FILTER CONTEXT AND FLOW



- **'Filter Context'** is decided by the relationship created between the Fact Tables and LookUp Table. It is the Primary Key (from the LookUp Table) that is linked to the Foreign Key (of the Fact Table). **Using this 'Filter context' the data model is able to use them to map the corresponding data in the linked tables.**
  Eg - In the picture the Filter context is 'Territory Key' between Territory_LookUp and Sales Table and also between Territory_LookUp and Sales Table (shown in rectangle).

- **'Filter Flow'** means the direction in which the Filter Context will move or flows. It is represented by the arrow you see on the Relationship links. The 'Filter Context' flows from the LookUp Table in the direction of the Fact Table for a particular relationship. **It does not flow in the opposite direction.**
  Eg - In the picture the Filter flow is from is 'Territory_LookUp' Table to the Sales Table and the Returns Table (shown in circle).

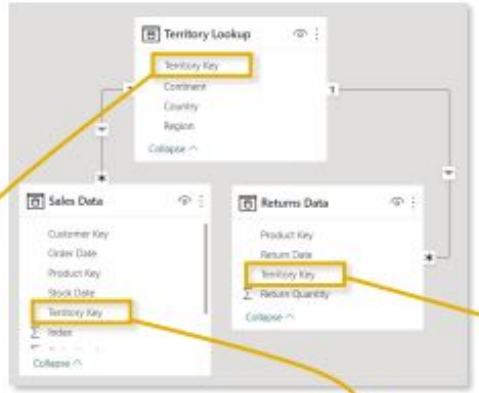
FILTER CONTEXT AND FLOW



- In this example there are 2 Data tables (Sales and Returns) and a Territory 'look up' table
- The arrows of each Relationship shows the 'filter direction'. They define the 'Filter Flow'. It points from the lookup side (1 side) of the relation to the data table side (many side indicated by '*')
- **When you filter a table, that filter context is passed along the 'downstream' tables (following the direction of the arrows)**
- **Filters cannot flow against the direction of the arrow i.e. they cannot flow upstream**

Best Practice - Always create a data model by placing the Lookup tables above the Data tables so it acts as a visual reminder that the filters flow downstream

FILTER FLOW



In this model, the only way to filter both **Sales** and **Returns** data by **Territory** is to use the **Territory Key** from the lookup table, which is upstream and related to both fact tables

- Filtering using Territory Key from the **Sales** table yields **incorrect Returns values**, since the filter context can't flow to any other table
- Filtering using Territory Key from the **Returns** table yields **incorrect Sales values**, and is limited to territories that exist in the returns table

TerritoryKey	OrderQuantity	ReturnQuantity
1	12,513	270
2	40	
3	30	
4	17,191	362
5	49	1
6	9,694	238
7	7,862	186
8	7,950	163
9	17,951	404
10	9,694	204
Total	**84,174**	**1,828**

*Filtering by **Territory Lookup**[Territory Key]*

TerritoryKey	OrderQuantity	ReturnQuantity
1	12,513	1,828
2	40	1,828
3	30	1,828
4	17,191	1,828
5	49	1,828
6	9,694	1,828
7	7,862	1,828
8	7,950	1,828
9	17,951	1,828
10	9,694	1,828
Total	**84,174**	**1,828**

*Filtering by **Sales Data**[Territory Key]*

TerritoryKey	OrderQuantity	ReturnQuantity
1	84,174	270
4	84,174	362
5	84,174	1
6	84,174	238
7	84,174	186
8	84,174	163
9	84,174	404
10	84,174	204
Total	**84,174**	**1,828**

*Filtering by **Returns Data**[Territory Key]*

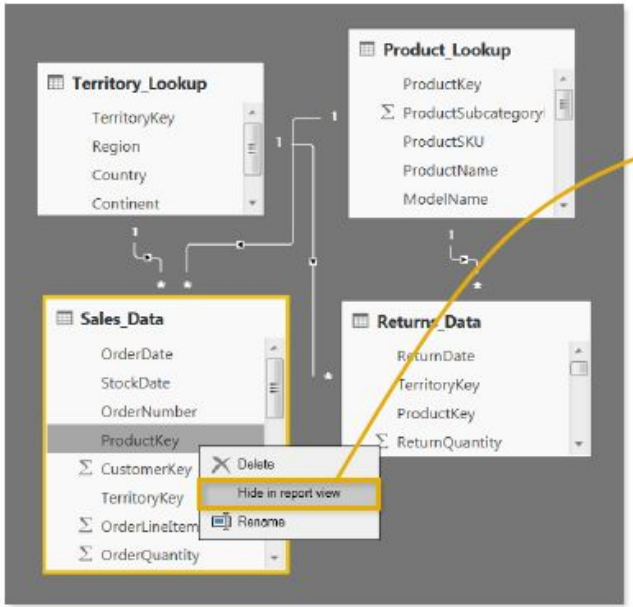
EXERCISE ON FILTER FLOW

Create relationship between Territory_lookUp table, Sales table and Returns table. Now create a Table using Matrix template in Reports view to show the ‘Order Quantity’ and ‘Returns Quantity’ using -

a) Territory Key from the ‘Territory_LookUp’ table
b) Territory Key from the ‘Sales Data’ table
c) Territory Key from the ‘Returns’ table

Power BI

HIDING FIELDS FROM REPORTS VIEW



Hiding fields from Report View makes them inaccessible from the Report tab (*although they can still be accessed within the **Data** or **Relationships** views*)

This is commonly used to prevent users from filtering using invalid fields, or to hide irrelevant metrics from view

Best Practice - Hide the 'Foreign key' columns in your data tables so that the users can filter using the lookup tables only

Power BI

EXERCISE ON HIDING THE FOREIGN KEYS

Hide all the 8 Foreign keys in the 'FactTable' -

- CPTCode-PK
- DateOfService
- DaignosisCode-PK
- Hospital-PK
- Patient-PK
- Payer-PK
- Physician-PK
- Transaction-PK

Power BI