



Power BI L200 Data Modeling

Instructor: Roshan Oganiya
Email: roganiya@dstrat.com



An aerial night view of a city, likely Singapore, showing a dense urban landscape with numerous high-rise buildings and a complex network of roads and highways. The image is overlaid with a network of glowing white lines and nodes, suggesting data connectivity or a digital infrastructure. A prominent light trail from a train or subway line runs vertically through the center of the image. A solid green square is positioned on the left side of the image.

Making sense of **your data**

We Love Data

- 20 years of experience in **analytics & business intelligence**.
- Based in the GTA & service clients worldwide.
- Award winning Microsoft Partner with 70+ Employees.

Microsoft
Partner



Gold Application Development
Gold Cloud Platform
Gold Datacenter
Gold Data Analytics
Gold Data Platform

Microsoft Partner Network

IMPACT
AWARDS

2016 WINNER



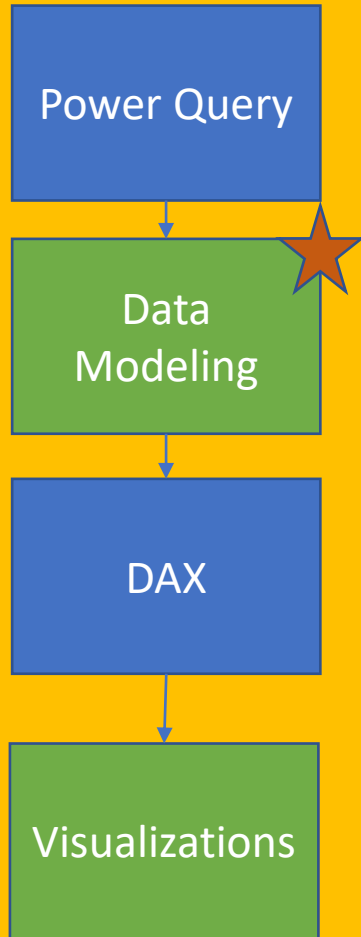
DIMENSIONAL
STRATEGIES INC

Our clients

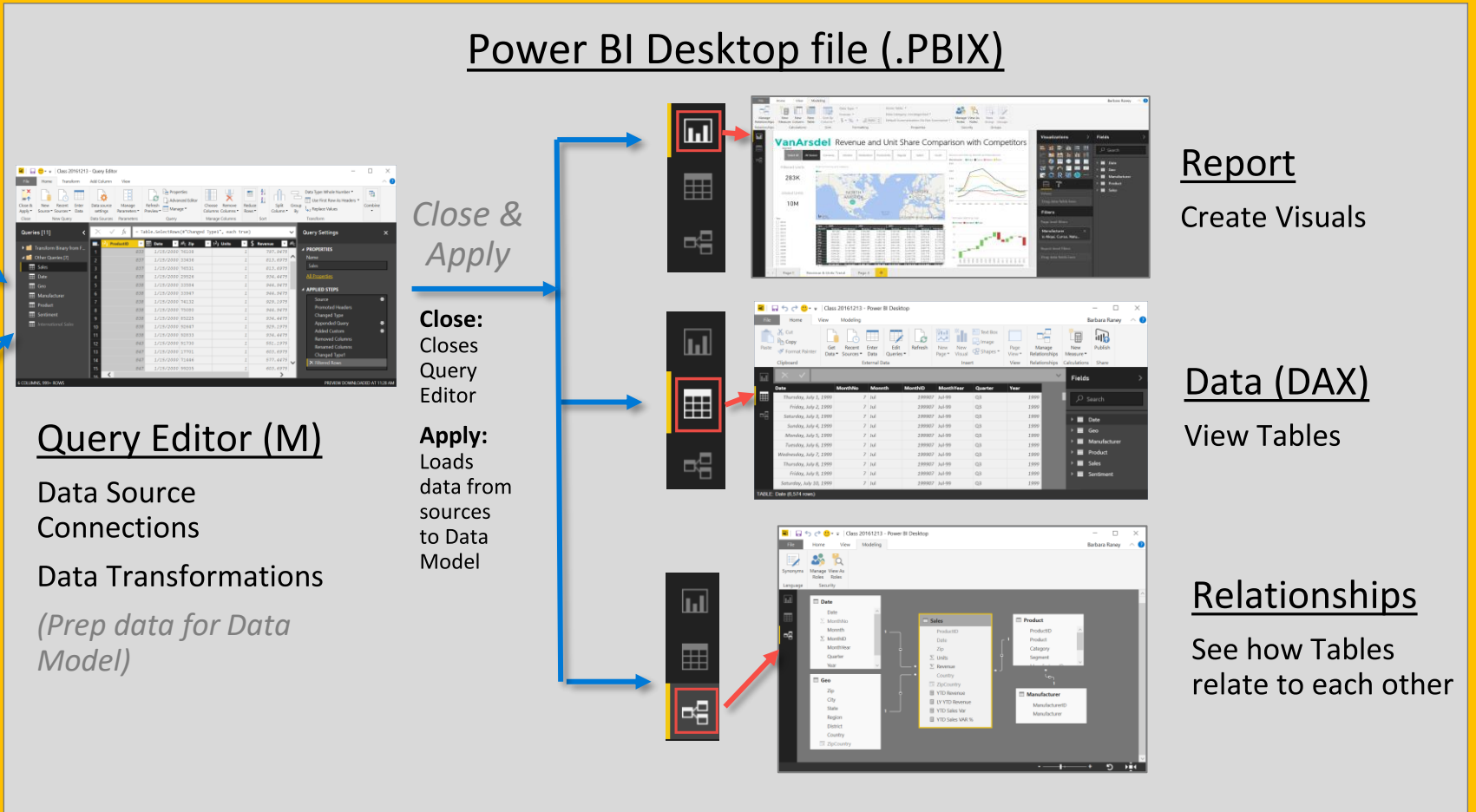
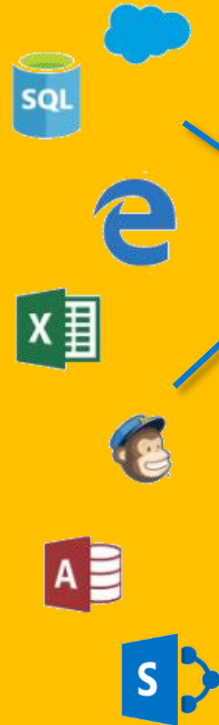


Report Development Flow

Every component is equally important to produce robust solution



Data Sources



COURSE OBJECTIVES

By the end of this course, you will be able to:

- Understand basic concepts of Data Modeling
- Understand the consequences of data model design decisions
- Understand concepts of calculated columns and measures

Agenda

9:00 - 9:15 Initial remarks and Introduction to the course

Section A

9:15 - 10:15 Intro to Data Preparation

Section B

10:15 - 11:00 Data Model Schemas, Normalization, Calculated Columns and Measures

11:00 - 11:15 Break

11:15 - 11:45 Lab 1

Section C

11:45 - 12:15 Data Storage in Power BI

12:15 - 12:30 Best Practices, Q&A

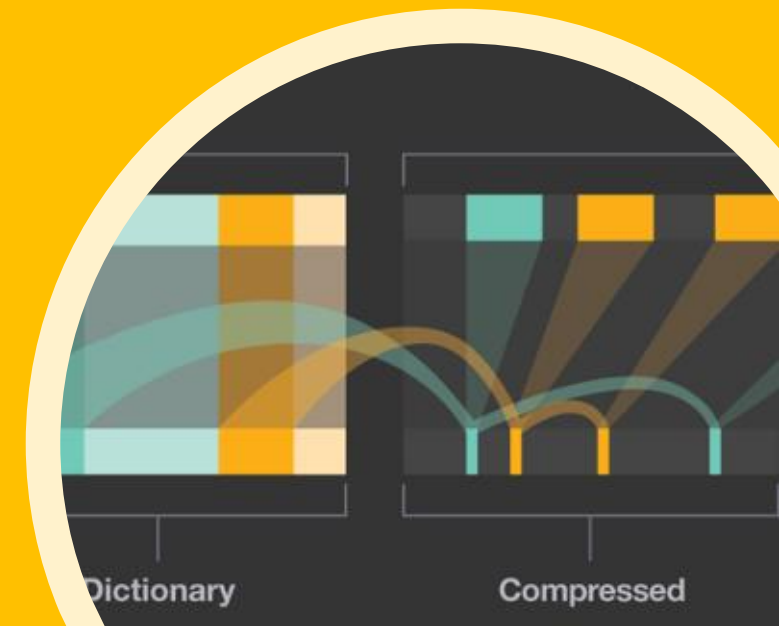
Section A

Intro to Data Preparation



Why Prepare our data?

- Power BI is powerful enough to compile and analyze data, but..
- If the data is not prepared properly, these compilations will be slower and reduce the report's analytical efficiency
- Data needs to cater to the technology of the compression engine being used by PBI to develop a robust data model



What is a Data model?

A Power BI Data Model is a collection of tables with relationships which enable your business users to easily understand and explore their data to get business insights.

Why is it important to have a Good Data model?

- Improves understandability of the data
- Increases performance of dependent processes and systems
- Increases resilience to change

The Technology behind Power BI

The VertiPaq Engine:

Columnar Database Engine - Columns & Segments

How many distinct products sold in 2017-Q1 , only Product and Date columns are used

Compresses data to distinct values (Encoding)

In-Memory mode for tabular architecture

Pro Tip – Have your Queries/Tables be as “Narrow” as possible

VertiPaq Engine

Columnar Database

Row Based Database

First Name	Last Name	Sales
John	Smith	\$10
Jane	Doe	\$25
Hardy	B	\$35

- Stores **each row separately** (like a separate file)
- Retrieving multiple columns from a single row is fast
- Retrieving multiple rows from a single column is slower

PBI - Columnar Database

First Name	Last Name	Sales
John	Smith	\$10
Jane	Doe	\$25
Hardy	B	\$35

- Stores **each column separately** (like a separate file)
- Retrieving multiple columns from a single row is slower
- Retrieving multiple rows from a single column is faster
- **Columnar databases are well suited for analytics**

In-Memory Database

PBI uses In-Memory Database where Data stored in **RAM (in memory)** when the file is open

Power BI compresses data to conserve space in RAM

Dashboard in a Day Class Data

Sales Fact	145.0 MB
Dimensions	7.0 MB
Int'l Sales	128.0 MB
Total Data	280.0 MB

DIAD Complete Data Model

Data Model	13.0 MB
-------------------	----------------

Almost 21X
Compression!!

Queries ONLY – No Data Loaded

Query Metadata	14 KB
-----------------------	--------------

Entities

Dimension Table:

Contain **descriptive** information used to slice and dice data from Fact Tables (eg: branch_name, branch_type)

Also holds Relationship/Key Fields used to connect the **dimension** to the **fact table** (eg: branch_key)

Wider tables with small amount of rows

Fact Table:

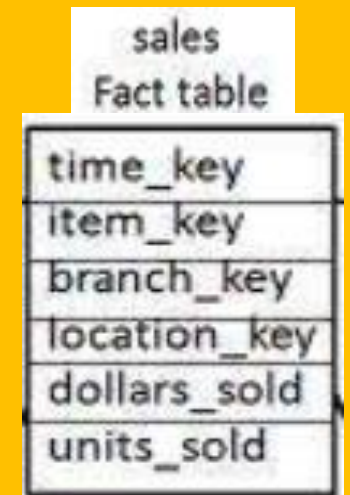
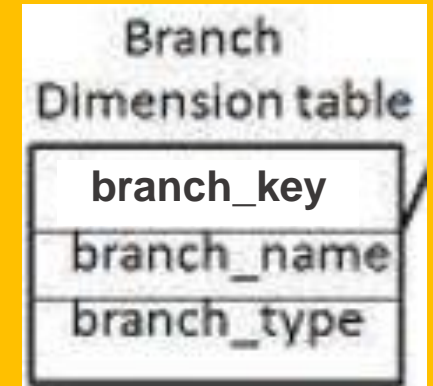
Contain **facts/details** which are fields used as values in a visualization (eg: dollars_sold, units_sold)

Also holds Relationship/Key fields used to connect the **dimension** to the **fact table** (eg: time_key, item_key, branch_key, location_key)

Narrow tables with large amount of rows

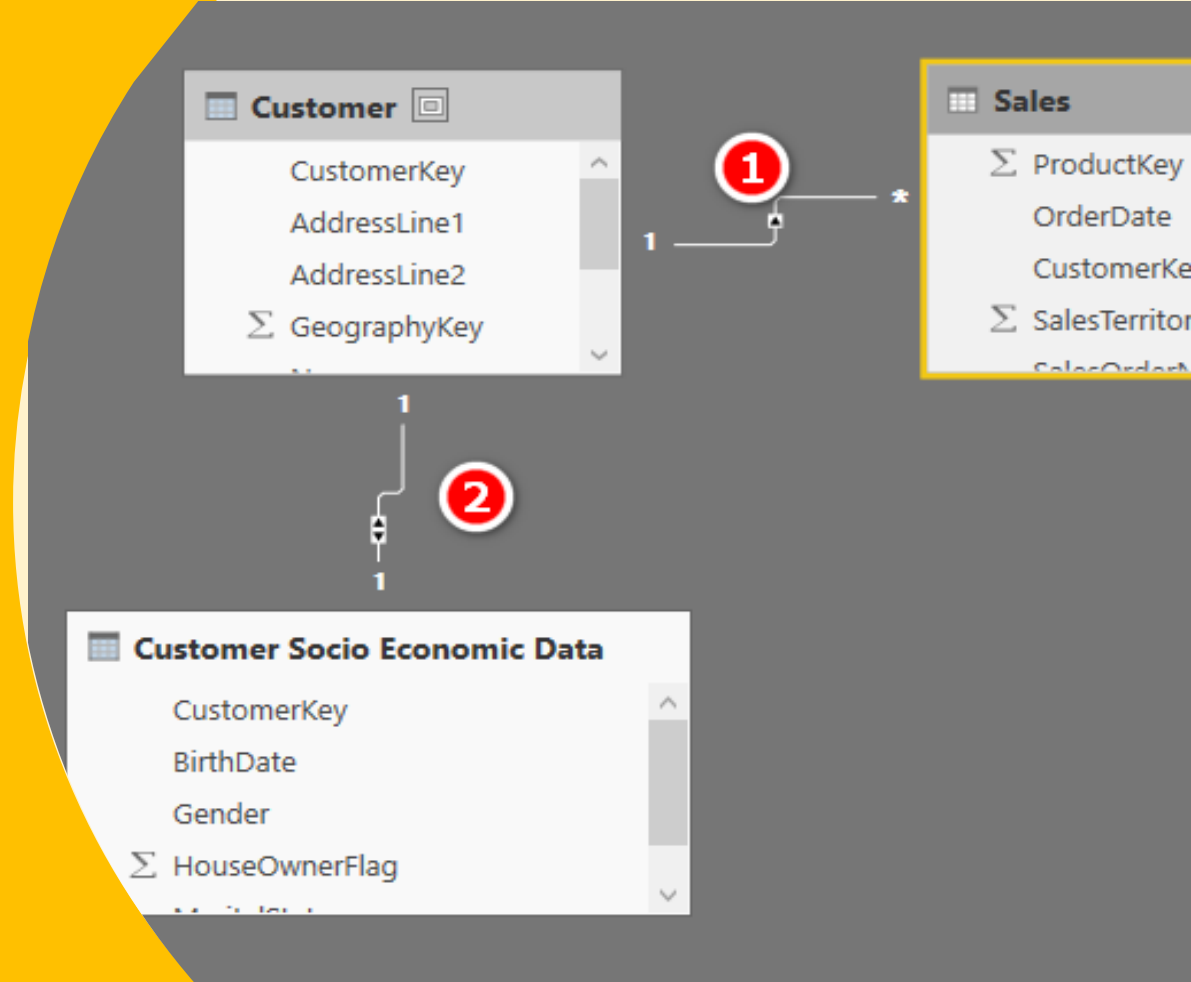
Golden Rule:

Avoid using a single table that includes everything (both facts and dimensions)



Relationships

- Connections between a 2 tables (usually fact & Dim tables) using columns from each are called **Relationships**
- Once you have two tables connected, you can work with the data in both as if they were in a single table
- A Relationship is analogous to how an Excel VLOOKUP function brings two tables together
- Power BI automatically sets the Cardinality, Cross Filter Direction and Active relationship when you load queries onto PBI.



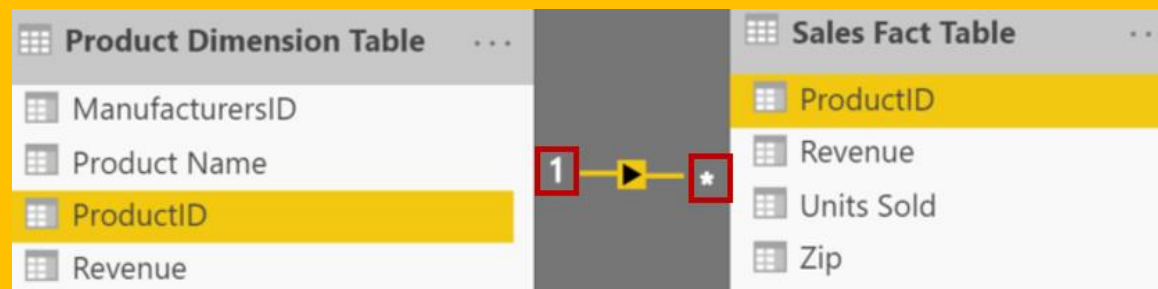
Cardinality

One to One (1 : 1) relationship

- Takes place when you connect columns with the same, distinct values.
- For such a relationship, you can **merge** the two tables together in Power Query Editor and disable loading the original to avoid redundancy

One to Many (1 : *) relationship

- The most common type of cardinality used
- Takes place when you connect a field with unique values to another table with the same field but repeating values



Cardinality

Many to Many (* : *) relationship

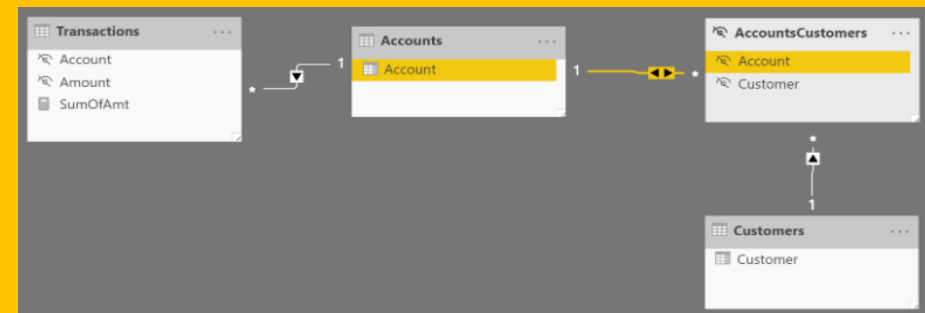
- Takes place when there are multiple records of the same value in the joining field of the two tables being joined.
- Considered to be a weak relationship; causes a lot of issues. Can be resolved by creating a shared dimension and creating one to many relationships with the shared dimension.
- **Avoid** Many to Many relationships when possible as it is laborious to maintain

Cross Filter Direction

The direction of a relationship is called the cross-filter direction as it sets up the way a filter propagates through your data

Uni -Directional Relationship

- Used when a dimension table filters through fact table data as the filter direction moves from the dimension to the fact table with the connecting field (ProductID)



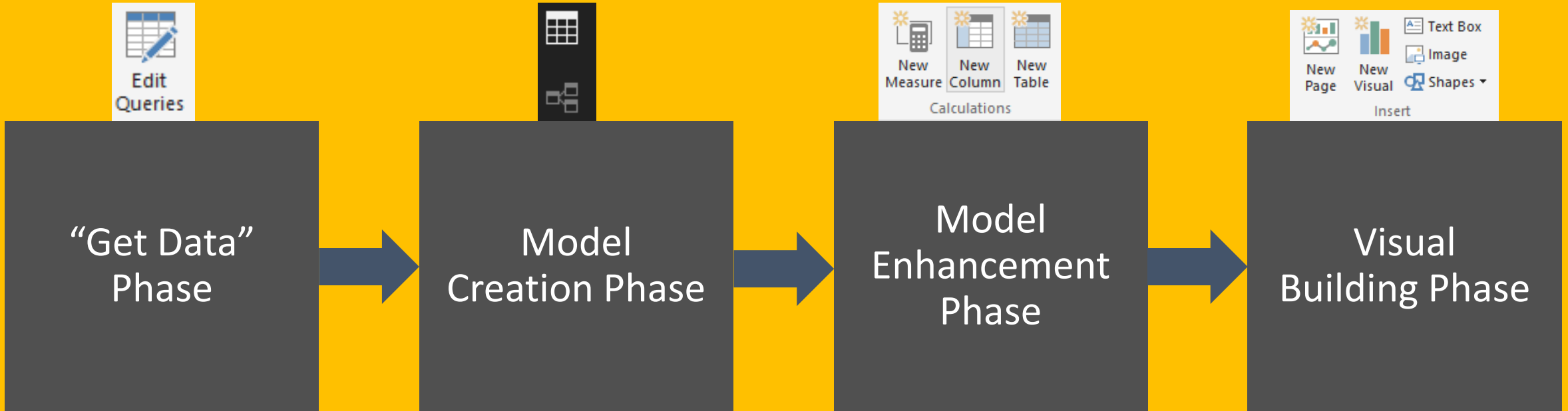
Bi-Directional Relationship

- Allow you to pass filters in both directions
 - This is different than Many to Many
 - There is a significant performance penalty for Bi-Directional filtering

Section B

Data Model Schemas, Normalization, DAX Calculated Columns and Measures

Phases in Building a Power BI Desktop File



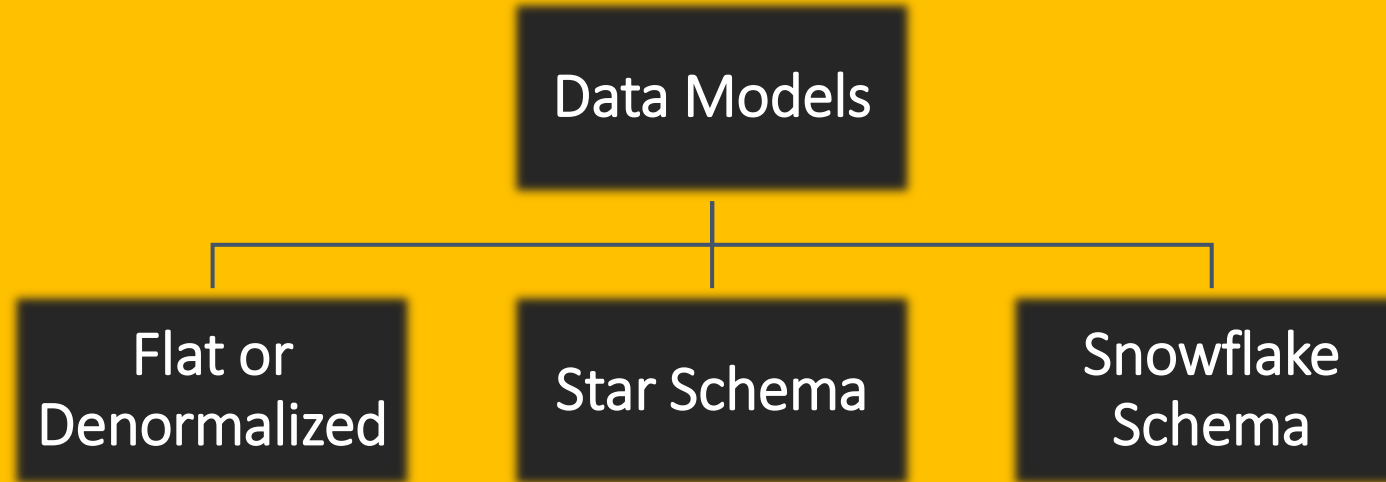
Create Query in
"M"

Compress data,
auto detect
relationships
(Automatically
done by Tool)

Add calculated
columns,
measures, add
missing
relationships

Evaluate
Measures and
build each visual

Data Model Brings Facts and Dimensions Together



Note: This is not an exhaustive list but are the most common model types used by Power BI.

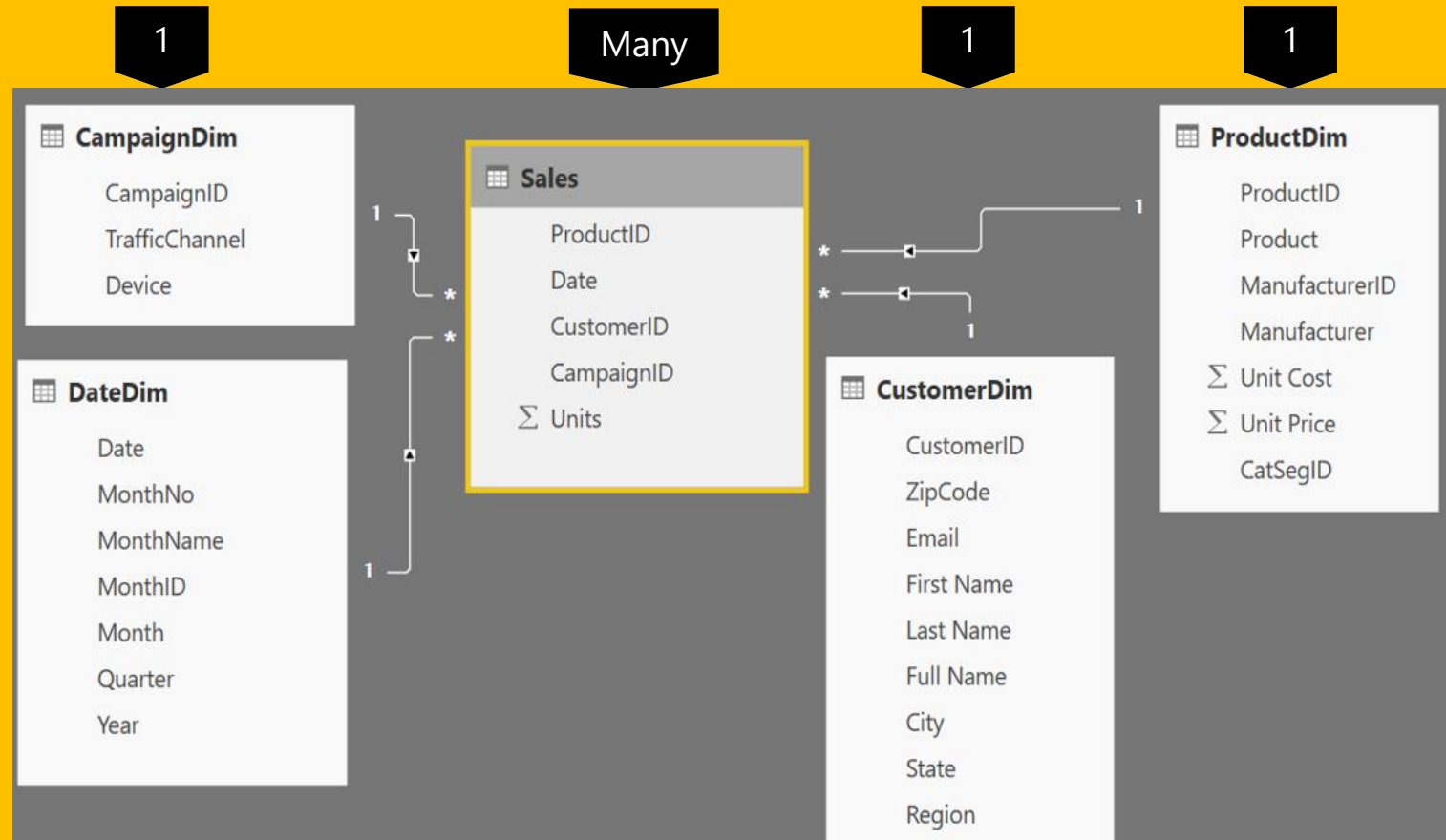
Flat or Denormalized Schema

- All attributes for model exist in a single table
- Highly inefficient
- Model has extra copies of data > slow performance
- Size of a flat table can blow up quickly as data model becomes complex

ProductID	Product	Date	CustomerID	Email	Last Name	First Name	Full Name	CampaignID	Units	CatSegID
1	676 Maximus UC-41	9/25/2011	70283	Farrah.Kent@xyza.com	Kent	Farrah	Farrah Kent	22	1	10
2	585 Maximus UC-50	3/24/2014	70283	Farrah.Kent@xyza.com	Kent	Farrah	Farrah Kent	15	1	10
3	585 Maximus UC-50	11/30/2014	138334	Martha.Mcclain@xyza...	Mcclain	Martha	Martha Mcclain	8	1	10
4	585 Maximus UC-50	6/21/2015	27193	Hedda.Mcintosh@xyza...	Mcintosh	Hedda	Hedda Mcintosh	22	1	10
5	585 Maximus UC-50	1/6/2013	238970	Lunea.Walker@xyza.com	Walker	Lunea	Lunea Walker	21	1	10
6	585 Maximus UC-50	3/22/2013	182241	Upton.Page@xyza.com	Page	Upton	Upton Page	17	1	10
7	449 Maximus UM-54	9/25/2011	195385	Drake.Wells@xyza.com	Wells	Drake	Drake Wells	22	1	4
8	449 Maximus UM-54	9/30/2014	168009	Wallace.Bender@xyza...	Bender	Wallace	Wallace Bender	17	1	4
9	449 Maximus UM-54	8/12/2014	110391	Astra.Erickson@xyza...	Erickson	Astra	Astra Erickson	20	1	4
10	449 Maximus UM-54	4/16/2014	49327	Echo.Bradley@xyza.com	Bradley	Echo	Echo Bradley	7	1	4
11	449 Maximus UM-54	2/28/2013	65952	Yoko.Gross@xyza.com	Gross	Yoko	Yoko Gross	17	1	4
12	449 Maximus UM-54	6/6/2013	97	Yoshi.Grant@xyza.com	Grant	Yoshi	Yoshi Grant	10	1	4
13	449 Maximus UM-54	5/14/2013	56757	Brian.Carrillo@xyza...	Carrillo	Brian	Brian Carrillo	10	1	4
14	449 Maximus UM-54	4/9/2015	248715	Mark.Hewitt@xyza.com	Hewitt	Mark	Mark Hewitt	19	1	4
15	449 Maximus UM-54	4/28/2013	248715	Mark.Hewitt@xyza.com	Hewitt	Mark	Mark Hewitt	8	1	4
16	449 Maximus UM-54	3/28/2014	240831	Oscar.Avila@xyza.com	Avila	Oscar	Oscar Avila	18	1	4
17	449 Maximus UM-54	2/26/2014	201004	Duncan.Mcintosh@xyza...	Mcintosh	Duncan	Duncan Mcintosh	19	1	4
18	615 Maximus UC-80	5/14/2012	212645	Jacob.Santiago@xyza...	Santiago	Jacob	Jacob Santiago	22	1	10
19	615 Maximus UC-80	5/14/2012	70666	Hilary.Collier@xyza...	Collier	Hilary	Hilary Collier	22	1	10
20	615 Maximus UC-80	5/14/2012	114459	Chester.Mitchell@xyz...	Mitchell	Chester	Chester Mitche...	22	1	10
21	615 Maximus UC-80	5/14/2012	221670	Sage.Yang@xyza.com	Yang	Sage	Sage Yang	22	1	10
22	615 Maximus UC-80	6/3/2012	168009	Wallace.Bender@xyza...	Bender	Wallace	Wallace Bender	22	1	10
23	615 Maximus UC-80	6/3/2012	154439	Iliana.Dunlap@xyza.c...	Dunlap	Iliana	Iliana Dunlap	22	1	10
24	615 Maximus UC-80	6/4/2012	191391	Joelle.Lee@xyza.com	Lee	Joelle	Joelle Lee	22	1	10

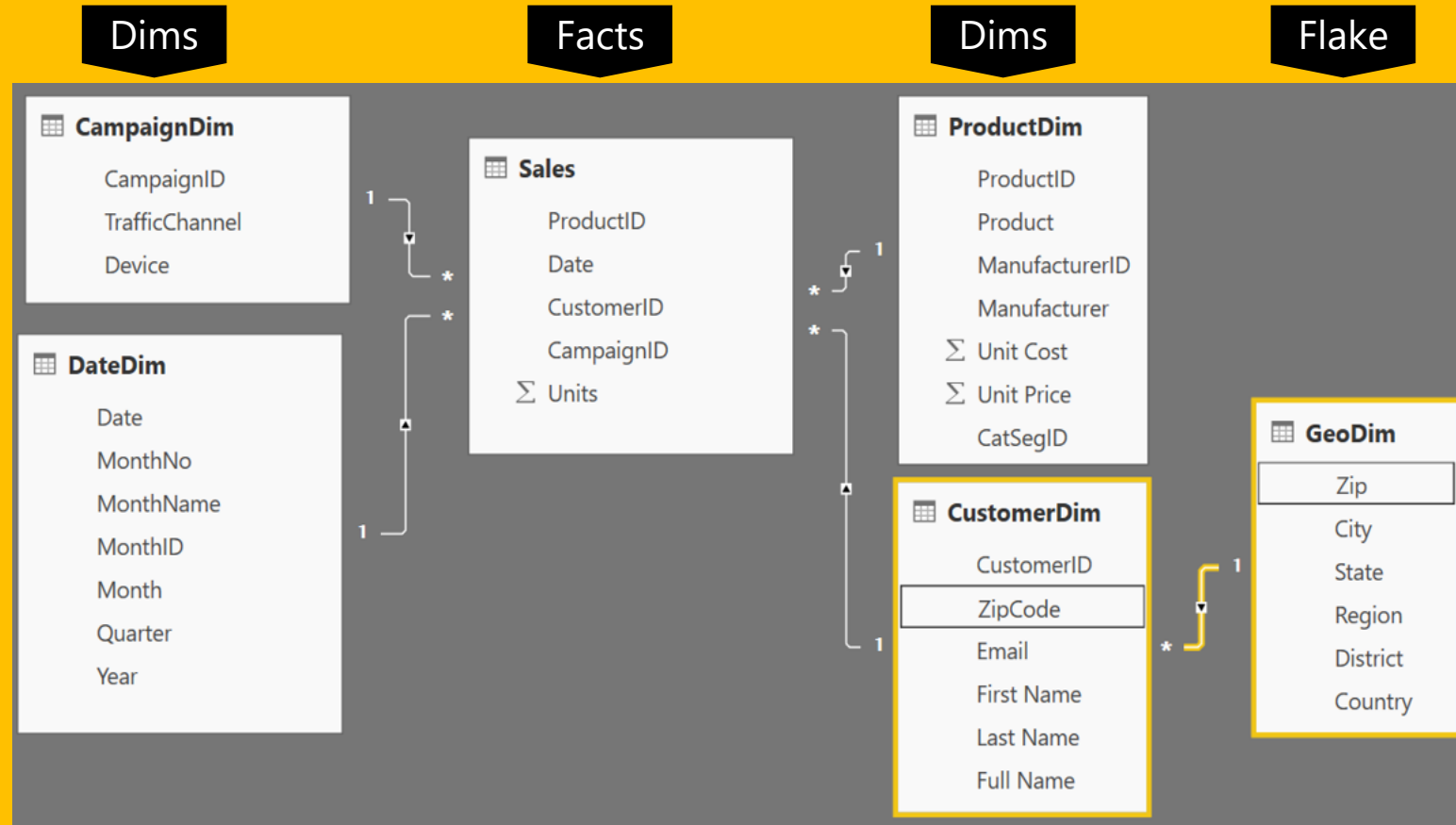
Star Schema

- Simple, easy to understand, fewer joins
- Comprises of a single Fact table in the middle branching outwards to connect to various dimension tables
- Fact table is the “Many” side of the (one to many) relationship
- Consumes more space than the snowflake schema (not always a bad thing as Power BI is powerful enough)



Snowflake Schema

- Dimension tables are **Normalized** in Snowflake schema
- Dimensions “snowflake” off of other Dimensions
- Dim or Fact tables can be the “Many” side of the relationship



Granularity & Multiple Fact Tables

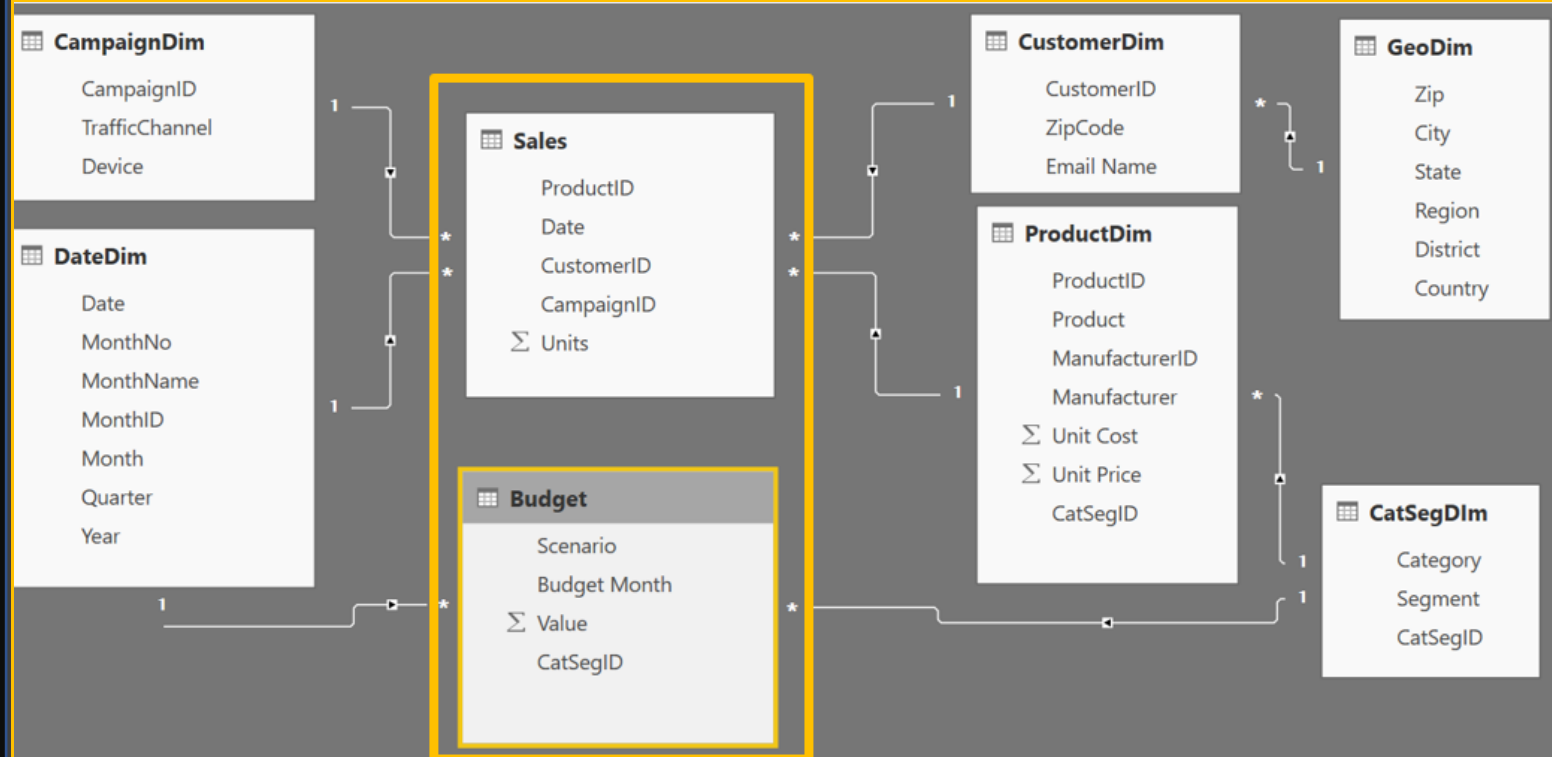
- Grain (granularity) measures the level of detail in a table

Example:

One row per order or per Item

Daily or Monthly date grain

- If your facts have **very different granularities**, split them into **Multiple Fact tables** & connect them to shared dimensions at the lowest common granularity.

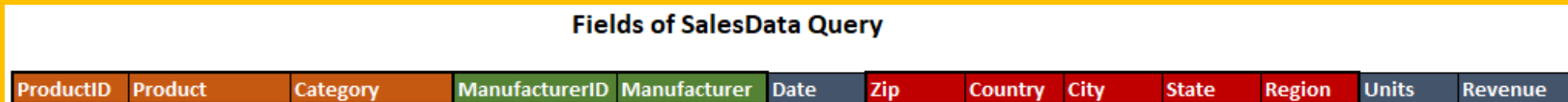


Sales (Daily by Product)

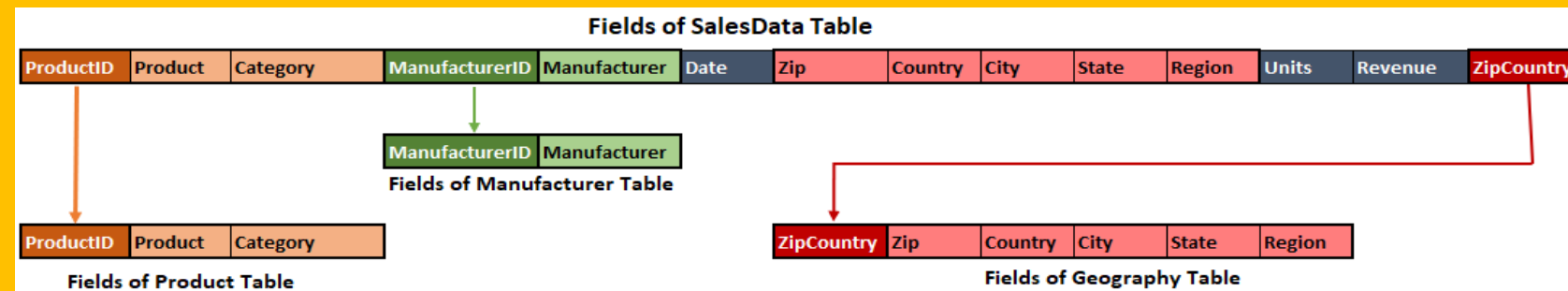
Budget (Monthly by Product Category & Product Segment)

Normalization

- Process of organizing database to make it more flexible by eliminating redundancy and inconsistent dependency
- Deals with creating separate tables for values that can apply to multiple records (dimension tables) and relating these tables with some sort of a foreign key.
 - This involves studying the dataset to see what fields can be grouped together to form dimension tables that could be used by other fact tables



- Next, try to figure out how the new dimension tables could be related to the fact table with the help of a simple or compound key



DAX Foundations

Calculated Columns and Measures are both written in the DAX Language

A Calculated Column is evaluated as a new column in the table in which it resides and will not change value until the underlying data is refreshed.

Measures are calculations which do not have a result until they are used in a visualization. They may use sums, averages, minimum or maximum values, counts, or more advanced calculations; and they change value in response to your interaction with your reports.

Calculated Column

Measure

What is a Calculated Column?

ProductID	Product	Category	Segment	ManufacturerID	Manufacturer	Unit Cost	Unit Price	Price Band
577	Maximus UC-42	Urban	Convenience	7	VanArsdel	74.73	102.37	High
578	Maximus UC-43	Urban	Convenience	7	VanArsdel	57.48	78.74	High
579	Maximus UC-44	Urban	Convenience	7	VanArsdel	96.96	132.82	High
580	Maximus UC-45	Urban	Convenience	7	VanArsdel	60.92	83.45	High
581	Maximus UC-46	Urban	Convenience	7	VanArsdel	101.54	139.10	High
582	Maximus UC-47	Urban	Convenience	7	VanArsdel	26.06	35.69	Medium
583	Maximus UC-48	Urban	Convenience	7	VanArsdel	40.18	55.05	High
584	Maximus UC-49	Urban	Convenience	7	VanArsdel	45.22	61.94	High

↓
Calculated Column

Pro Tip: Always refer to a calculated column by its full name -> **TableName[ColumnName]**

Best Practices – Calculated Columns

- Whenever possible, DAX helper columns should be avoided. Each “Helper Column” will consume RAM
- Create a calculated column in the Dim Table as opposed to in the Fact Table
- Move calculated columns to “M” if you can

What is a Measure?

- Measures are created using DAX
- Place your Measures on a Fact table for best results

ProductID	Date	CustomerID	CampaignID	Units	Sales Amount
666	2/24/12	58642	3	1	\$81.37
666	2/25/12	208515	3	1	\$81.37
666	7/12/12	164032	3	1	\$81.37
666	7/12/12	243676	3	1	\$81.37
406	6/12/16	31036	16	1	\$191.62
406	6/17/16	44688	16	1	\$191.62
406	6/17/16	108991	16	1	\$191.62

[Total Sales]=SUM(Sales[Sales Amount])

Tip: When referring to a measure in other calculations, refer to it without a Table name: **[MeasureName]**

Calculated Column vs. Measure: When to Use What

Rule of Thumb

Calculated Column – Use in Page, Report & Visual Filters as well as Slicers, Rows and Columns

Measures - Use in Values section

The image shows a screenshot of a Power BI report. On the left, there is a slicer for the 'Year' field, with options for 2010 through 2016. The year 2015 is selected. Below the slicer, a green arrow points to the word 'Slicer'. In the center, there is a table with the following data:

State	O1	O2	O3	O4	Total
VT	\$295.48	\$106.00	\$7.40	\$536.20	\$945.08
SD	\$1,449.57	\$1,717.00	\$1,269.22	\$3,000.62	\$7,436.41
DC	\$3,384.23	\$754.69	\$932.40	\$3,941.70	\$9,013.02
WY	\$1,433.65	\$2,550.38	\$3,087.02	\$3,762.88	\$10,833.93
ND	\$3,094.90	\$934.39	\$1,051.45	\$5,763.94	\$10,844.68
AK	\$1,094.00	\$2,889.09	\$3,288.21	\$4,365.64	\$11,636.94
MT	\$3,503.88	\$2,904.44	\$2,581.02	\$3,965.87	\$12,955.21
DE	\$5,688.76	\$2,344.29	\$1,206.45	\$5,849.41	\$15,088.91
HI	\$2,334.18	\$3,436.84	\$2,349.20	\$7,204.34	\$15,324.56
HI	\$3,284.68	\$4,434.03	\$3,105.51	\$7,158.20	\$17,982.42

Annotations on the table include: a green arrow pointing to the 'Total' column labeled 'Columns', a red arrow pointing to the 'Total' column labeled 'Values', and a green arrow pointing to the 'State' column labeled 'Rows'.

Designing good data models

Key takeaways to design a good Power BI Desktop data model

- RAM is precious !!!!!

Some Tips and tricks to save RAM and increase speed of model

- If a fact table contains an ID field which is unique for each record, **remove it** unless needed as a connector key
 - Ex. Transaction ID
- **Sort columns** before bringing them into a Power BI data model
- The DateTime data type is usually not needed, unless you are specifically using the Time component
 - If you really need Time, **try splitting Date & Time** into two columns - Reduces # of unique values

Star Schema – Good for most Data Models

Knowledge Check

1. What is a *data model* in the context of Power BI?

- *A data model is a collection of tables and relationships*

2. What are some advantages of a star schema over a flat or denormalized model?

- *Dimension tables save space by reducing the amount of data that needs to be repeated over and over in every row*
- *Relationships between tables can be leveraged for more complex measures*

3. How might you improve the performance of a Power BI model?

- *Try using a star schema instead of a flat or denormalized model*
- *Remove unnecessary columns*
- *Set appropriate data types*

Break

Lab 1

Section C

Data Storage in Power BI

Data Mode Types in Power BI

How can I tell what Data Model Type I have?

- **Live Connect** to SQL Analysis Services (SSAS) tabular
 - Report view only available
- **DirectQuery** to SQL or other relational source
 - Report & Relationship views available
- **Import data** into Power BI (creates a copy of the data)
 - Report, Data and Relationship views available



Live Connect



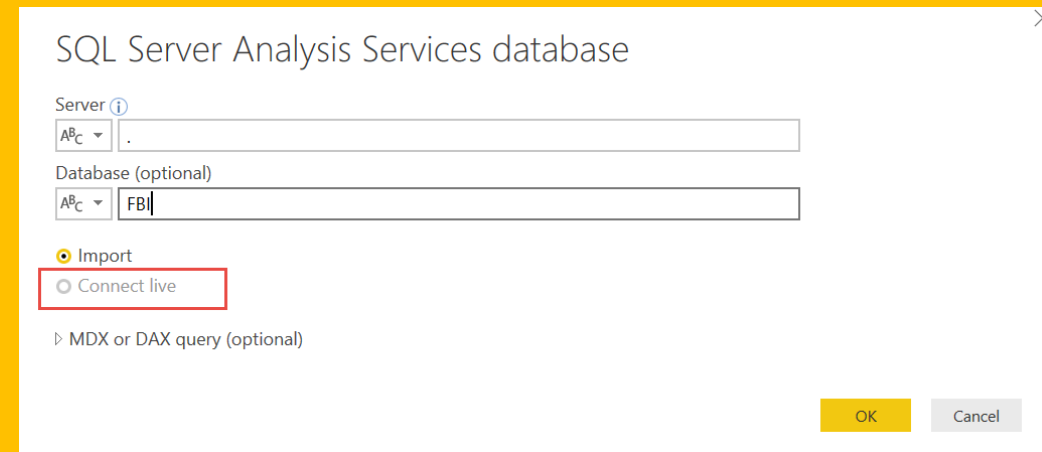
DirectQuery



Import

Connection: Live Connect

- Live Connect to Multidimensional or Tabular
 - On Premise or Azure
- Only a single connection will be made, and all modeling is done in the cube
- You can not add relationships or additional data source
- If allowed, you can add DAX measures



SQL Server Analysis Services database

Server ⓘ
ABC .

Database (optional)
ABC FBI

Import
 Connect live

▶ MDX or DAX query (optional)

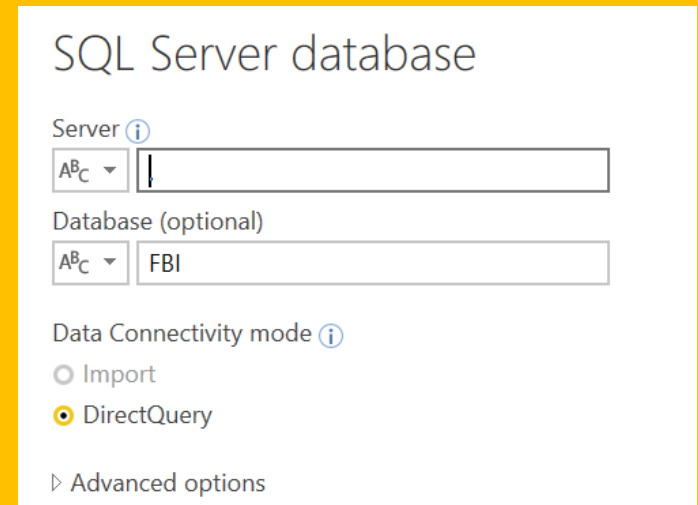
OK Cancel

Choosing storage mode: LiveConnect

- Mode used when Power BI model accessing an SQL Server Analysis Services (SSAS) data model
 - Can be a server (IaaS) or Azure Analysis Services (PaaS)
- When is it appropriate
 - Organization has already invested significantly in SSAS, have mature models
 - High level of control needed around partitions, data refresh, query scale-out and workload splitting
 - Granular auditing, monitoring and diagnostics
 - Models can't fit in Premium (P3+ models up to 10GB v. no overall model size limit in SSAS)

Connection: DirectQuery to Relational Source

- Direct Query to SQL or other relational source
 - On Premise or Azure
- Composite modeling is possible where some data sources are in Direct Query mode and a few are in Import mode
- You can add relationships and DAX



SQL Server database

Server ⓘ
ABC ▾ |

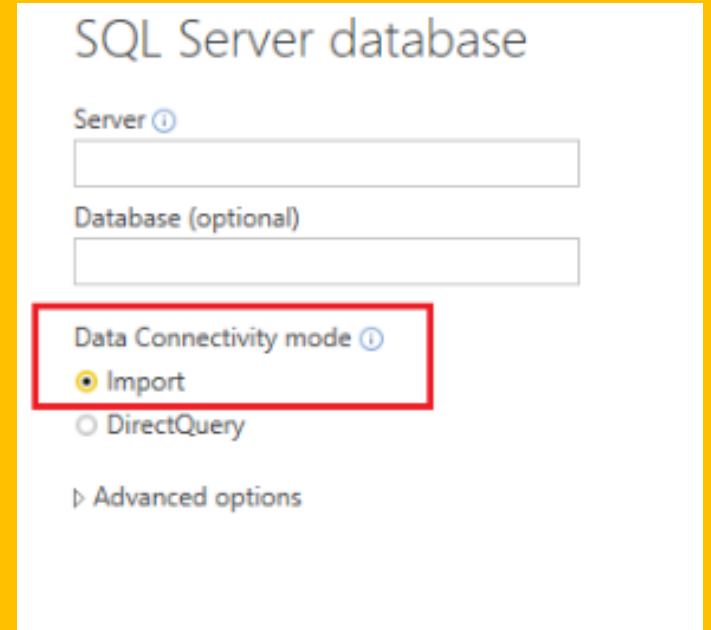
Database (optional)
ABC ▾ | FBI

Data Connectivity mode ⓘ
 Import
 DirectQuery

▸ Advanced options

Import Mode

- Most widely used connection and the default type when connecting to most sources.
- The connection will ingest/pull all the data from the source and make it a part of the PBI



SQL Server database

Server ⓘ

Database (optional)

Data Connectivity mode ⓘ

Import

DirectQuery

▸ Advanced options

Choosing storage mode: Import vs DirectQuery

Import is your first choice (all in memory = best speed, no DAX limits)

When is it inappropriate?

- Extremely large data volumes
- Need near real-time access to data from source
- Considerable existing investment in external DW or OLAP (modelled, conformed, cleaned, calcs defined etc.). SSAS MD, SAP HANA and BW are common.
- Regulatory and data sovereignty requirements

Considerations

- How much source data, how compressible? Rule of thumb is 5x-10x
- Is Premium an option? (larger datasets supported there)
- Will blended architecture suffice? (Composite models, Aggregations for summary data)
- Some limits on DAX in DirectQuery mode (e.g. time intelligence)

Best Practices

Data Modeling

An inefficient model can completely slow down a report, even with very small data volumes

GOALS:

- Make the model as small as possible
 - There are valid reasons to bend this rule
- Schema supports the analysis
- Relationships are built purposefully and thoughtfully

Move calculations to the source

Scenario

- Many DAX calculated columns with high cardinality

Why is it undesired?

- Calculated columns don't compress as well as physical columns

Proposed Solution

- Perform calc in Power Query, ideally push down

Remove unused tables and columns

Scenario

- Model contains tables/columns that are not used for reporting/analysis or calculations

Why is it undesired?

- Increases model size
- Increases time to load into memory
- Increases refresh time
- May affect usability

Avoid high precision/cardinality columns

Scenario

- Model contains columns at a higher precision than needed for analysis e.g. datetime in milliseconds, weight to 6 decimal places
- Model contains columns that are highly unique

Why is it undesired?

- Less compression with high precision/cardinality
- Increases time to load into memory
- Increases refresh time

Proposed Solution

- Remove if not needed
- Reduce precision
- Split datetime into date and time

Use integers instead of strings

Why is it undesired?

- Strings use dictionary encoding, integers use run length encoding which is more efficient

Proposed Solution

- Check data types and set to integer if known to be numerical

Be careful with bi-directional relationships

Scenario

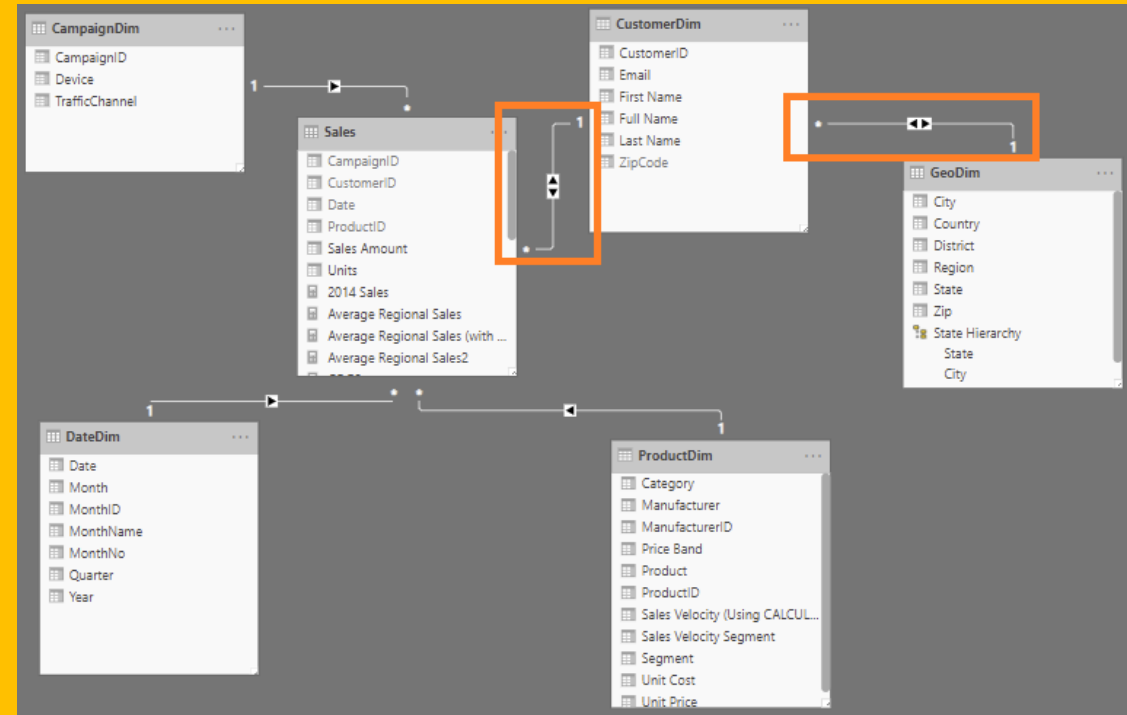
- Most relationships in the model are set to bi-directional

Why is it undesired?

- Applying filters/slicers traverses many relationships and can be slower
- Some filter chains unlikely to add business value

Proposed Solution

- Only use bi-di where the business scenario requires it



Set Default Summarization

Scenario

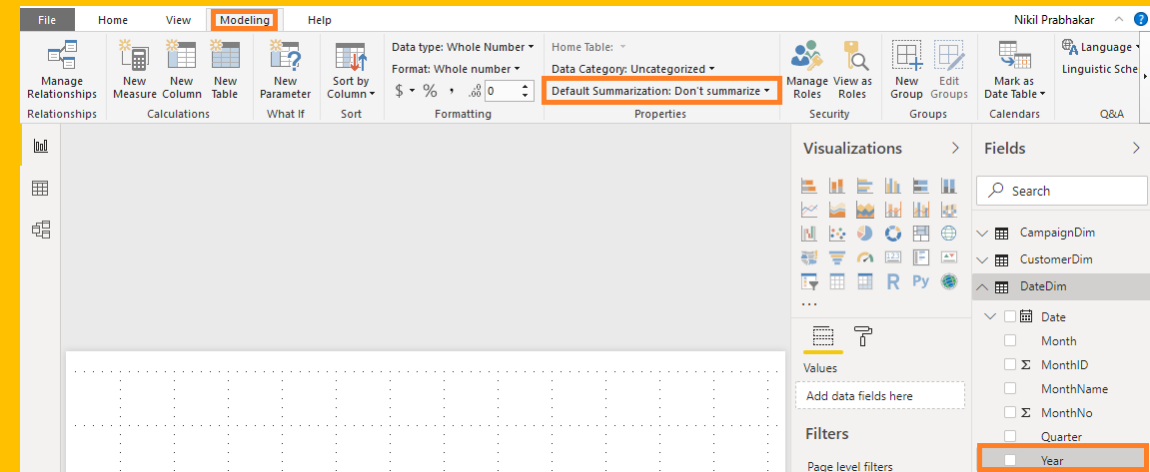
- Numeric columns in model that are purely informational (e.g. Account ID)
- Default summarization is Sum

Why is it undesired?

- Power BI will try to sum the number when dropped into visuals.
- Detailed tables/matrixes can be slower

Proposed Solution

- Set the default summarization to None



Q&A