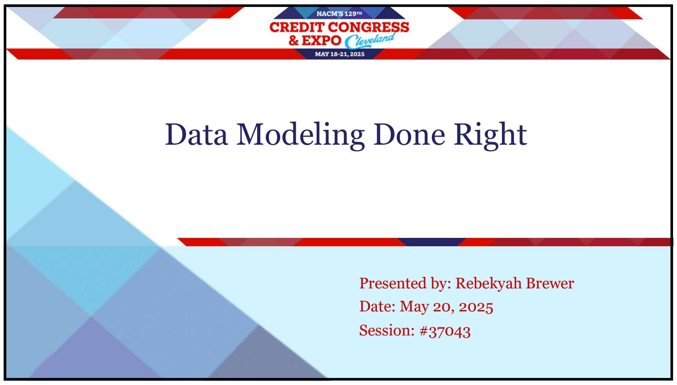
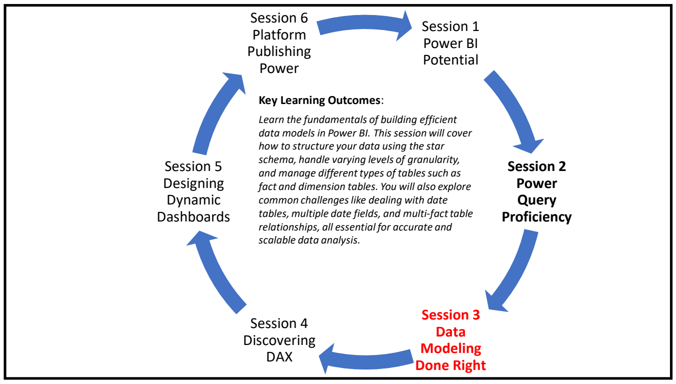


NACM'S 129TH CREDIT CONGRESS & EXPO Cleveland MAY 18-21, 2025

Data Modeling Done Right

Presented by: Rebekyah Brewer
Date: May 20, 2025
Session: #37043



Session 6
Platform
Publishing
Power

Session 1
Power BI
Potential

Session 2
Power
Query
Proficiency

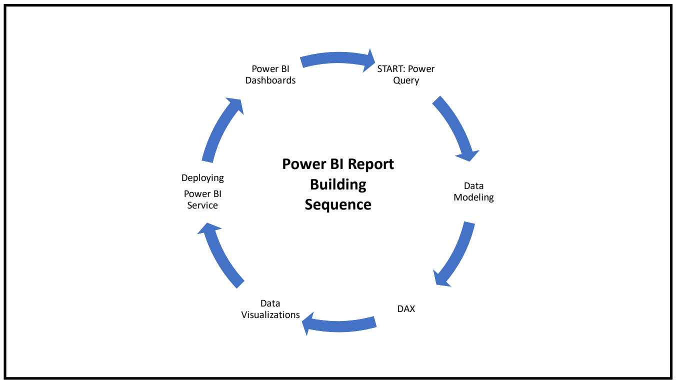
Session 3
Data
Modeling
Done Right

Session 4
Discovering
DAX

Session 5
Designing
Dynamic
Dashboards

Key Learning Outcomes:

Learn the fundamentals of building efficient data models in Power BI. This session will cover how to structure your data using the star schema, handle varying levels of granularity, and manage different types of tables such as fact and dimension tables. You will also explore common challenges like dealing with date tables, multiple date fields, and multi-fact table relationships, all essential for accurate and scalable data analysis.



Power BI Report Building Sequence

START: Power Query

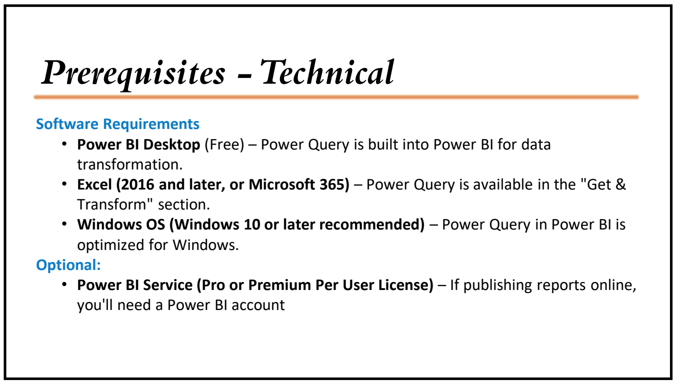
Data Modeling

DAX

Data Visualizations

Deploying Power BI Service

Power BI Dashboards



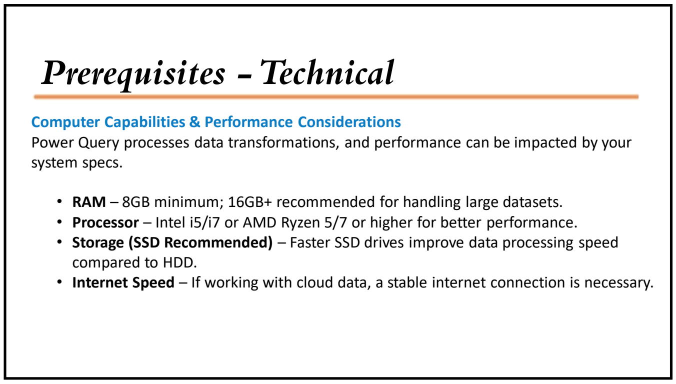
Prerequisites – Technical

Software Requirements

- **Power BI Desktop** (Free) – Power Query is built into Power BI for data transformation.
- **Excel (2016 and later, or Microsoft 365)** – Power Query is available in the "Get & Transform" section.
- **Windows OS (Windows 10 or later recommended)** – Power Query in Power BI is optimized for Windows.

Optional:

- **Power BI Service (Pro or Premium Per User License)** – If publishing reports online, you'll need a Power BI account

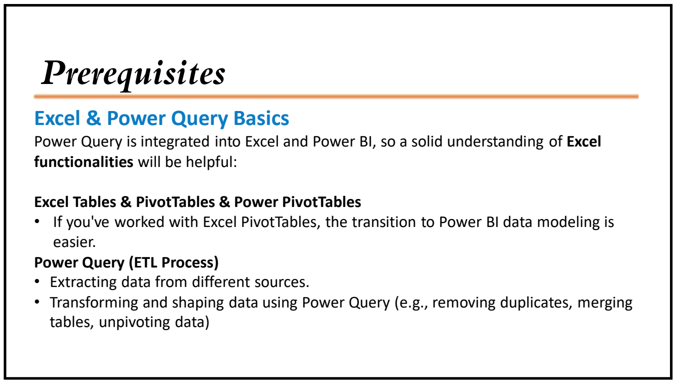


Prerequisites – Technical

Computer Capabilities & Performance Considerations

Power Query processes data transformations, and performance can be impacted by your system specs.

- **RAM** – 8GB minimum; 16GB+ recommended for handling large datasets.
- **Processor** – Intel i5/i7 or AMD Ryzen 5/7 or higher for better performance.
- **Storage (SSD Recommended)** – Faster SSD drives improve data processing speed compared to HDD.
- **Internet Speed** – If working with cloud data, a stable internet connection is necessary.



Prerequisites

Excel & Power Query Basics

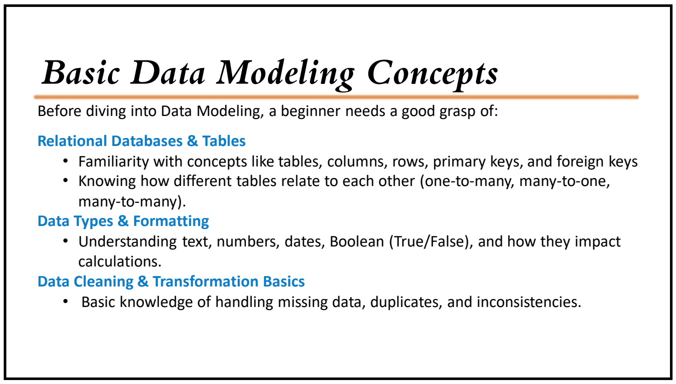
Power Query is integrated into Excel and Power BI, so a solid understanding of **Excel functionalities** will be helpful:

Excel Tables & PivotTables & Power PivotTables

- If you've worked with Excel PivotTables, the transition to Power BI data modeling is easier.

Power Query (ETL Process)

- Extracting data from different sources.
- Transforming and shaping data using Power Query (e.g., removing duplicates, merging tables, unpivoting data)



Basic Data Modeling Concepts

Before diving into Data Modeling, a beginner needs a good grasp of:

Relational Databases & Tables

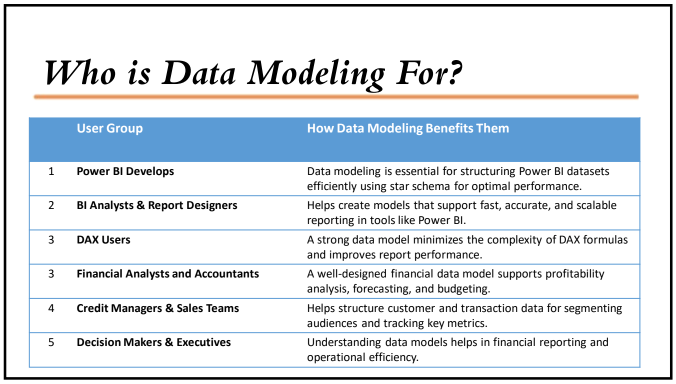
- Familiarity with concepts like tables, columns, rows, primary keys, and foreign keys
- Knowing how different tables relate to each other (one-to-many, many-to-one, many-to-many).

Data Types & Formatting

- Understanding text, numbers, dates, Boolean (True/False), and how they impact calculations.

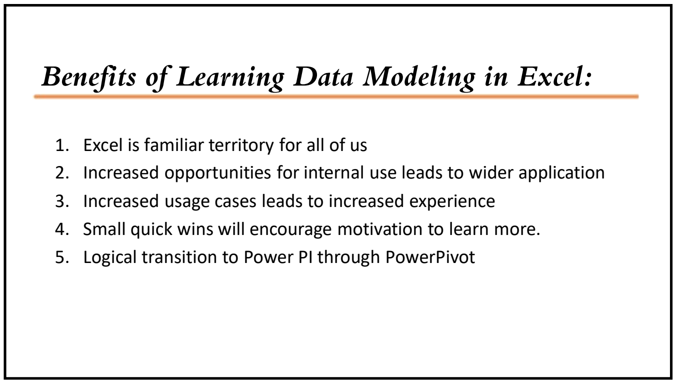
Data Cleaning & Transformation Basics

- Basic knowledge of handling missing data, duplicates, and inconsistencies.



Who is Data Modeling For?

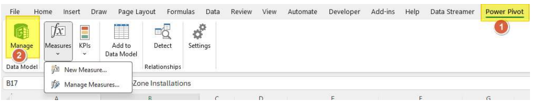
	User Group	How Data Modeling Benefits Them
1	**Power BI Develops**	Data modeling is essential for structuring Power BI datasets efficiently using star schema for optimal performance.
2	**BI Analysts & Report Designers**	Helps create models that support fast, accurate, and scalable reporting in tools like Power BI.
3	**DAX Users**	A strong data model minimizes the complexity of DAX formulas and improves report performance.
3	**Financial Analysts and Accountants**	A well-designed financial data model supports profitability analysis, forecasting, and budgeting.
4	**Credit Managers & Sales Teams**	Helps structure customer and transaction data for segmenting audiences and tracking key metrics.
5	**Decision Makers & Executives**	Understanding data models helps in financial reporting and operational efficiency.



Benefits of Learning Data Modeling in Excel:

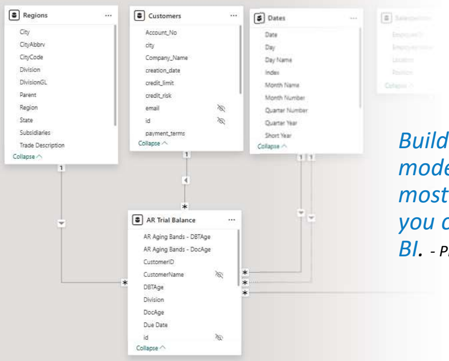
1. Excel is familiar territory for all of us
2. Increased opportunities for internal use leads to wider application
3. Increased usage cases leads to increased experience
4. Small quick wins will encourage motivation to learn more.
5. Logical transition to Power PI through PowerPivot

Loading Power Pivot

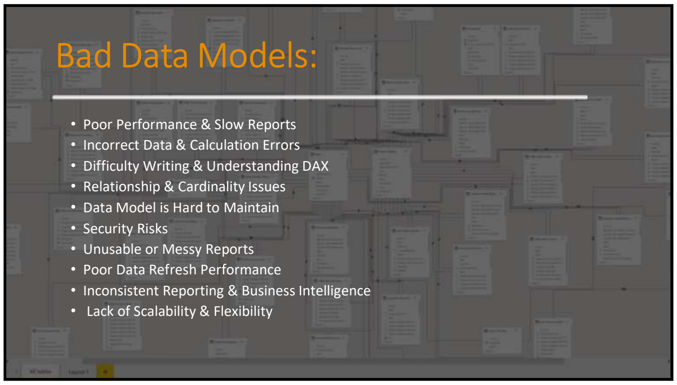


Why is a Data Model Important?

- **Optimized Performance –** Reduces redundancy and improves efficiency.
- **Scalability –** Allows easy expansion of data as your model grows.
- **Accurate Insights –** Prevents errors and inconsistencies.
- **Better Data Relationships –** Enables complex queries and drill-downs.
- **Better DAX Performance & Simplicity –** Makes DAX Measures easier to write and troubleshoot.

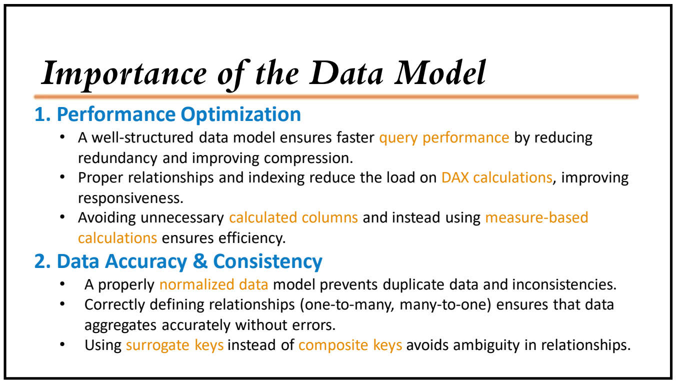


Building a good data model is one of the most important things you can do in Power BI. - Pragmatic Works



Bad Data Models:

- Poor Performance & Slow Reports
- Incorrect Data & Calculation Errors
- Difficulty Writing & Understanding DAX
- Relationship & Cardinality Issues
- Data Model is Hard to Maintain
- Security Risks
- Unusable or Messy Reports
- Poor Data Refresh Performance
- Inconsistent Reporting & Business Intelligence
- Lack of Scalability & Flexibility



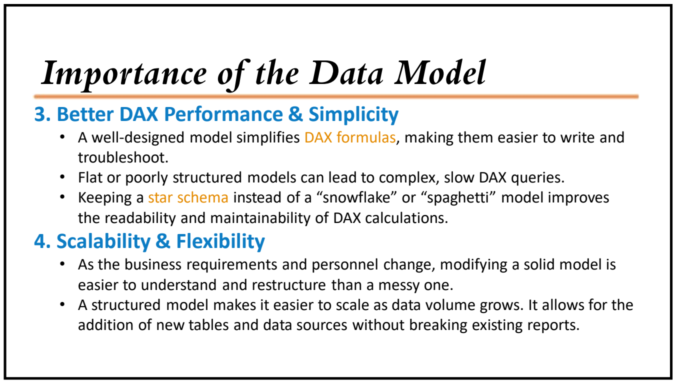
Importance of the Data Model

1. Performance Optimization

- A well-structured data model ensures faster query performance by reducing redundancy and improving compression.
- Proper relationships and indexing reduce the load on DAX calculations, improving responsiveness.
- Avoiding unnecessary calculated columns and instead using measure-based calculations ensures efficiency.

2. Data Accuracy & Consistency

- A properly normalized data model prevents duplicate data and inconsistencies.
- Correctly defining relationships (one-to-many, many-to-one) ensures that data aggregates accurately without errors.
- Using surrogate keys instead of composite keys avoids ambiguity in relationships.



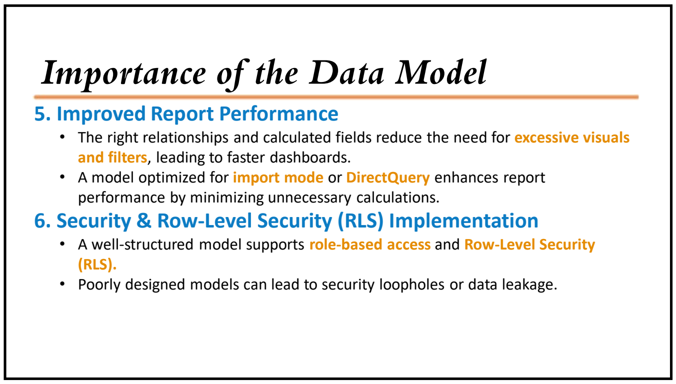
Importance of the Data Model

3. Better DAX Performance & Simplicity

- A well-designed model simplifies DAX formulas, making them easier to write and troubleshoot.
- Flat or poorly structured models can lead to complex, slow DAX queries.
- Keeping a star schema instead of a "snowflake" or "spaghetti" model improves the readability and maintainability of DAX calculations.

4. Scalability & Flexibility

- As the business requirements and personnel change, modifying a solid model is easier to understand and restructure than a messy one.
- A structured model makes it easier to scale as data volume grows. It allows for the addition of new tables and data sources without breaking existing reports.



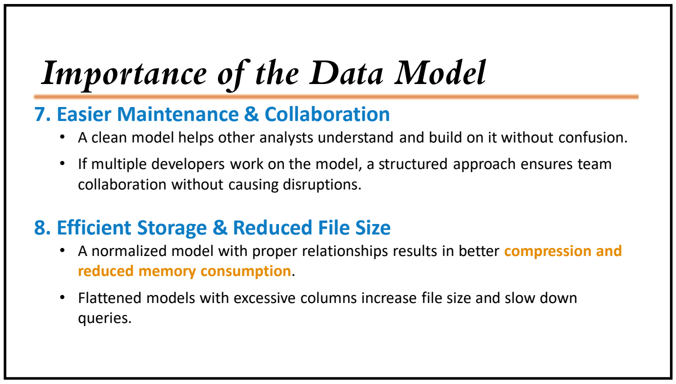
Importance of the Data Model

5. Improved Report Performance

- The right relationships and calculated fields reduce the need for **excessive visuals and filters**, leading to faster dashboards.
- A model optimized for **import mode** or **DirectQuery** enhances report performance by minimizing unnecessary calculations.

6. Security & Row-Level Security (RLS) Implementation

- A well-structured model supports **role-based access** and **Row-Level Security (RLS).**
- Poorly designed models can lead to security loopholes or data leakage.



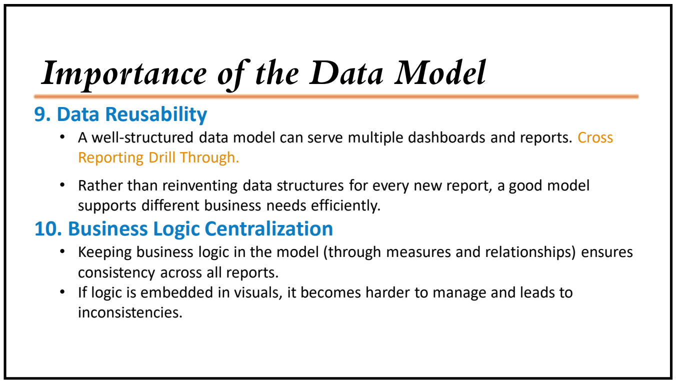
Importance of the Data Model

7. Easier Maintenance & Collaboration

- A clean model helps other analysts understand and build on it without confusion.
- If multiple developers work on the model, a structured approach ensures team collaboration without causing disruptions.

8. Efficient Storage & Reduced File Size

- A normalized model with proper relationships results in better **compression and reduced memory consumption**.
- Flattened models with excessive columns increase file size and slow down queries.



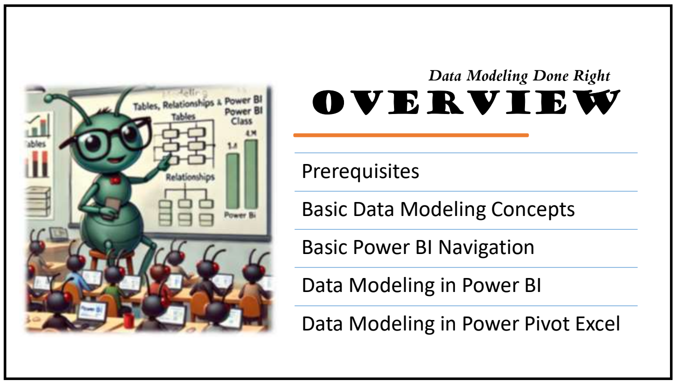
Importance of the Data Model

9. Data Reusability

- A well-structured data model can serve multiple dashboards and reports. Cross Reporting Drill Through.
- Rather than reinventing data structures for every new report, a good model supports different business needs efficiently.

10. Business Logic Centralization

- Keeping business logic in the model (through measures and relationships) ensures consistency across all reports.
- If logic is embedded in visuals, it becomes harder to manage and leads to inconsistencies.


Tables, Relationships & Power BI
Tables
Power BI Class
Relationships
Power BI

Data Modeling Done Right

OVERVIEW

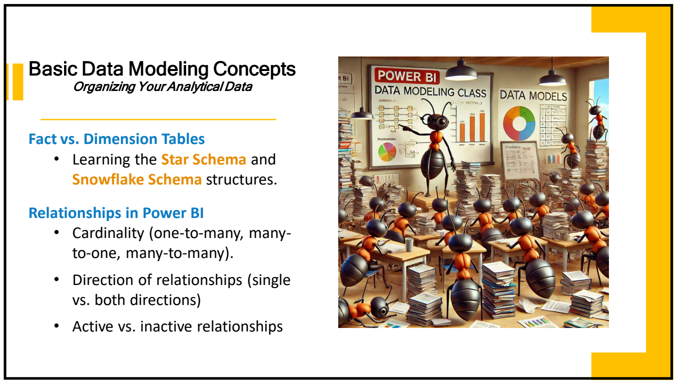
Prerequisites

Basic Data Modeling Concepts

Basic Power BI Navigation

Data Modeling in Power BI

Data Modeling in Power Pivot Excel


POWER BI
DATA MODELING CLASS
DATA MODELS

Basic Data Modeling Concepts

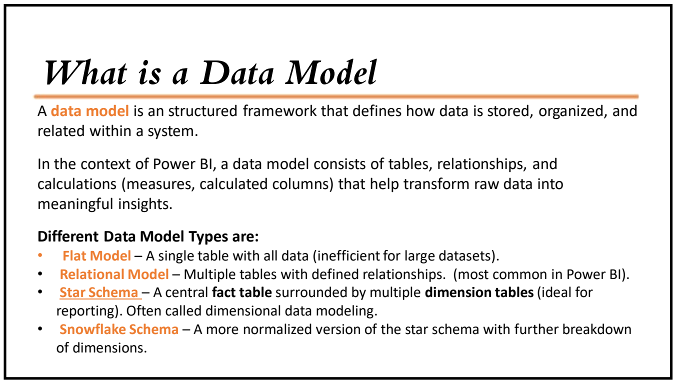
Organizing Your Analytical Data

Fact vs. Dimension Tables

- Learning the **Star Schema** and **Snowflake Schema** structures.

Relationships in Power BI

- Cardinality (one-to-many, many-to-one, many-to-many).
- Direction of relationships (single vs. both directions)
- Active vs. inactive relationships



What is a Data Model

A data model is an structured framework that defines how data is stored, organized, and related within a system.

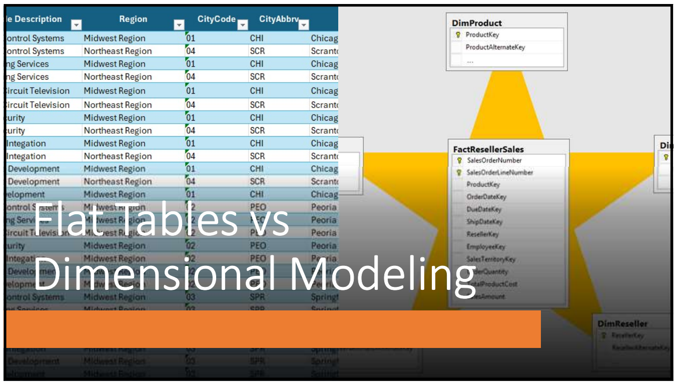
In the context of Power BI, a data model consists of tables, relationships, and calculations (measures, calculated columns) that help transform raw data into meaningful insights.

Different Data Model Types are:

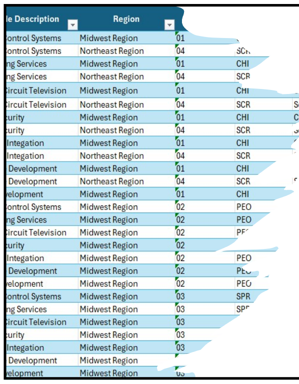
- Flat Model – A single table with all data (inefficient for large datasets).
- Relational Model – Multiple tables with defined relationships. (most common in Power BI).
- Star Schema – A central **fact table** surrounded by multiple **dimension tables** (ideal for reporting). Often called dimensional data modeling.
- Snowflake Schema – A more normalized version of the star schema with further breakdown of dimensions.

Key Components of Dimensional Data Modeling

- **Fact Tables** – Store measurable numerical data (e.g., sales, revenue, credit limits).
- **Dimension Tables** – Store descriptive data (e.g., customers, products, regions)
- **Relationships** – Define how dimensions relate to facts (typically one-to-many)
- **Surrogate Keys** – Unique identifiers used in dimension tables instead of **Natural keys**.
- **Hierarchies** – Logical drill-down structures (e.g., Year → Quarter → Month → Day).



Flat Tables vs Dimensional Modeling

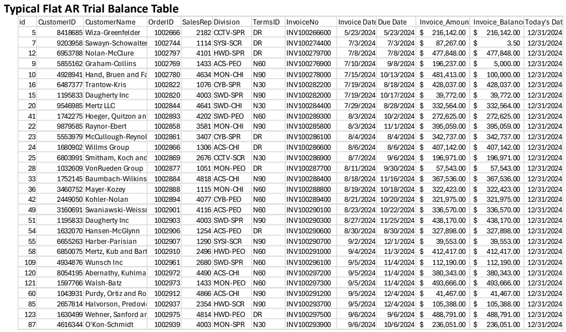


le Description	Region			
ontrol Systems	Midwest Region	01		
ontrol Systems	Northeast Region	04	SCR	
ng Services	Midwest Region	01	CHI	
ng Services	Northeast Region	04	SCR	
ircuit Television	Midwest Region	01	CHI	
ircuit Television	Northeast Region	04	SCR	Scran
urity	Midwest Region	01	CHI	Ch
urity	Northeast Region	04	SCR	Scr
Integration	Midwest Region	01	CHI	
Integration	Northeast Region	04	SCR	
Development	Midwest Region	01	CHI	
Development	Northeast Region	04	SCR	
elopment	Midwest Region	01	CHI	
ontrol Systems	Midwest Region	02	PEO	
ng Services	Midwest Region	02	PEO	
ircuit Television	Midwest Region	02	PE	
urity	Midwest Region	02		
Integration	Midwest Region	02	PEO	
Development	Midwest Region	02	PEO	
elopment	Midwest Region	02	PEO	
ontrol Systems	Midwest Region	03	SPR	
ng Services	Midwest Region	03	SPR	
ircuit Television	Midwest Region	03		
urity	Midwest Region	03		
Integration	Midwest Region	03		
Development	Midwest Region			
elopment	Midwest Region	03		

Flat Table also called a **denormalized table,** is a table that contains all relevant data in a single, wide structure without requiring joins to other tables. Each row in the table represents a complete dataset record, with all necessary attributes included as columns.

Characteristics of a Flat Table:

- **No Relationships or Joins Required** – All data is stored in a single table, removing the need for complex queries that join multiple tables. (No VLOOKUP/XLOOKUP, SUMIFS or INDEX/MATCH needed)
- **Redundant Data** – Since all information is in one place, some data fields (e.g., customer names, product descriptions) might be repeated across multiple rows.
- **Easy to Query** – Since all data is in one table, queries are typically simpler and run faster for basic reporting and analysis. All columns needed in one place.
- **Wide Table Structure** – The table may have many columns to store all necessary attributes.
- **Lack of Normalization** – Unlike a normalized relational database structure, flat tables do not follow principles like reducing redundancy through primary and foreign key relationships.



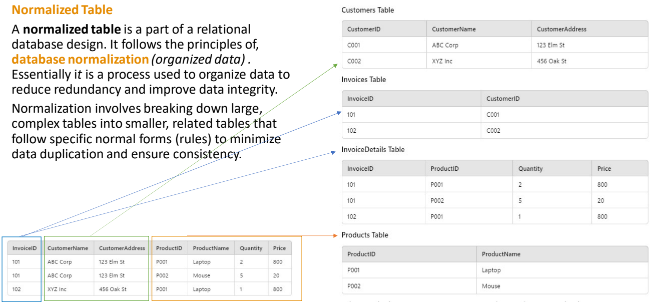
Typical Flat AR Trial Balance Table

id	CustomerID	CustomerName	OrderID	SalesRep	Division	TermsID	InvoiceNo	Invoice Date	Due Date	Invoice_Amoun	Invoice_Balanc	Today's Dat
5	8418685	Wiza-Greenfelder	1002666	2182	CCTV-SPR	DR	INV100266600	5/23/2024	5/23/2024	$ 216,142.00	$ 216,142.00	12/31/2024
7	9203958	Sawayn-Schowalter	1002744	1114	SYSI-SCR	DR	INV100274400	7/3/2024	7/3/2024	$ 87,267.00	$ 3.50	12/31/2024
12	6953788	Nolan-McClure	1002797	4101	HWD-SPR	DR	INV100279700	7/8/2024	7/8/2024	$ 477,848.00	$ 477,848.00	12/31/2024
9	5855162	Graham-Collins	1002769	1433	ACS-PEO	N60	INV100276900	7/10/2024	9/8/2024	$ 196,237.00	$ 5,000.00	12/31/2024
10	4928941	Hand, Bruen and Fa	1002780	4634	MON-CHI	N90	INV100278000	7/15/2024	10/13/2024	$ 481,413.00	$ 100,000.00	12/31/2024
16	6487377	Trantow-Kris	1002822	1076	CYB-SPR	N30	INV100282200	7/19/2024	8/18/2024	$ 428,037.00	$ 428,037.00	12/31/2024
15	1195833	Daugherty Inc	1002820	4003	SWD-SPR	N90	INV100282000	7/19/2024	10/17/2024	$ 39,772.00	$ 39,772.00	12/31/2024
20	9546985	Mertz LLC	1002844	4641	SWD-CHI	N30	INV100284400	7/29/2024	8/28/2024	$ 332,564.00	$ 332,564.00	12/31/2024
41	1742275	Hoeger, Quitzon an	1002893	4202	SWD-PEO	N60	INV100289300	8/3/2024	10/2/2024	$ 272,625.00	$ 272,625.00	12/31/2024
22	9879585	Raynor-Ebert	1002858	3581	MON-CHI	N90	INV100285800	8/3/2024	11/1/2024	$ 395,059.00	$ 395,059.00	12/31/2024
23	5553979	McCullough-Reynol	1002861	3407	CYB-SPR	DR	INV100286100	8/4/2024	8/4/2024	$ 342,737.00	$ 342,737.00	12/31/2024
24	1680902	Willms Group	1002866	1306	ACS-CHI	DR	INV100286600	8/6/2024	8/6/2024	$ 407,142.00	$ 407,142.00	12/31/2024
25	6803991	Smitham, Koch and	1002869	2676	CCTV-SCR	N30	INV100286900	8/7/2024	9/6/2024	$ 196,971.00	$ 196,971.00	12/31/2024
28	1032609	VonRueden Group	1002877	1051	MON-PEO	DR	INV100287700	8/11/2024	9/30/2024	$ 57,543.00	$ 57,543.00	12/31/2024
33	1752145	Baumbach-Wilkins	1002884	4818	ACS-CHI	N90	INV100288400	8/18/2024	11/16/2024	$ 367,536.00	$ 367,536.00	12/31/2024
36	3460752	Mayer-Kozey	1002888	1115	MON-CHI	N60	INV100288800	8/19/2024	10/18/2024	$ 322,423.00	$ 322,423.00	12/31/2024
42	2449050	Kohler-Nolan	1002894	4077	CYB-PEO	N60	INV100289400	8/21/2024	10/20/2024	$ 321,975.00	$ 321,975.00	12/31/2024
49	3160691	Swaniawski-Weissı	1002901	4116	ACS-PEO	N60	INV100290100	8/23/2024	10/22/2024	$ 336,570.00	$ 336,570.00	12/31/2024
51	1195833	Daugherty Inc	1002903	4003	SWD-SPR	N90	INV100290300	8/27/2024	11/25/2024	$ 438,170.00	$ 438,170.00	12/31/2024
54	1632070	Hansen-McGlynn	1002906	1254	ACS-PEO	DR	INV100290600	8/30/2024	8/30/2024	$ 327,898.00	$ 327,898.00	12/31/2024
55	6655263	Harber-Parisian	1002907	1290	SYSI-SCR	N90	INV100290700	9/2/2024	12/1/2024	$ 39,553.00	$ 39,553.00	12/31/2024
58	6850075	Mertz, Kub and Bart	1002910	2496	HWD-PEO	N60	INV100291000	9/4/2024	11/3/2024	$ 412,417.00	$ 412,417.00	12/31/2024
109	4934876	Wunsch Inc	1002961	2680	SWD-SPR	N60	INV100296100	9/5/2024	11/4/2024	$ 112,190.00	$ 112,190.00	12/31/2024
120	8054195	Abernathy, Kuhlma	1002972	4490	ACS-CHI	N60	INV100297200	9/5/2024	11/4/2024	$ 380,343.00	$ 380,343.00	12/31/2024
121	1597766	Walsh-Batz	1002973	1433	MON-PEO	N60	INV100297300	9/5/2024	11/4/2024	$ 493,666.00	$ 493,666.00	12/31/2024
60	1043931	Purdy, Ortiz and Ro	1002912	4866	ACS-CHI	N90	INV100291200	9/5/2024	12/4/2024	$ 41,467.00	$ 41,467.00	12/31/2024
85	2657814	Halvorson, Predovi	1002937	2354	HWD-SCR	N90	INV100293700	9/5/2024	12/4/2024	$ 105,388.00	$ 105,388.00	12/31/2024
123	1630499	Wehner, Sanford ar	1002975	4814	HWD-PEO	DR	INV100297500	9/6/2024	9/6/2024	$ 488,791.00	$ 488,791.00	12/31/2024
87	4616344	O'Kon-Schmidt	1002939	4003	MON-SPR	N30	INV100293900	9/6/2024	10/6/2024	$ 236,051.00	$ 236,051.00	12/31/2024

Normalized Table

A **normalized table** is a part of a relational database design. It follows the principles of, **database normalization** *(organized data)* . Essentially *it* is a process used to organize data to reduce redundancy and improve data integrity.

Normalization involves breaking down large, complex tables into smaller, related tables that follow specific normal forms (rules) to minimize data duplication and ensure consistency.



Customers Table

CustomerID	CustomerName	CustomerAddress
C001	ABC Corp	123 Elm St
C002	XYZ Inc	456 Oak St

Invoices Table

InvoiceID	CustomerID
101	C001
102	C002

InvoiceDetails Table

InvoiceID	ProductID	Quantity	Price
101	P001	2	800
101	P002	5	20
102	P001	1	800

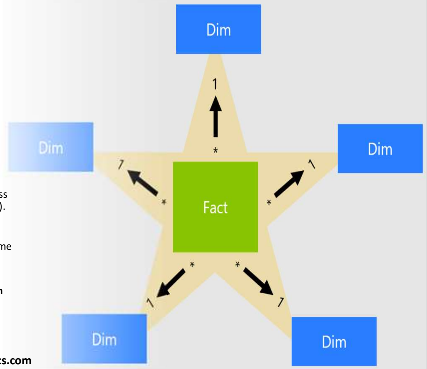
Products Table

ProductID	ProductName
P001	Laptop
P002	Mouse

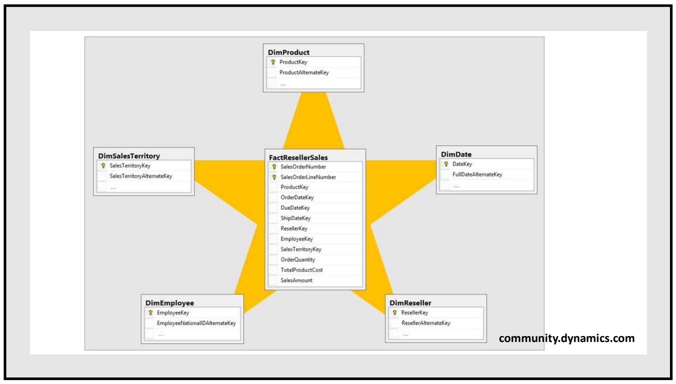
InvoiceID	CustomerName	CustomerAddress	ProductID	ProductName	Quantity	Price
101	ABC Corp	123 Elm St	P001	Laptop	2	800
101	ABC Corp	123 Elm St	P002	Mouse	5	20
102	XYZ Inc	456 Oak St	P001	Laptop	1	800

Star Schema

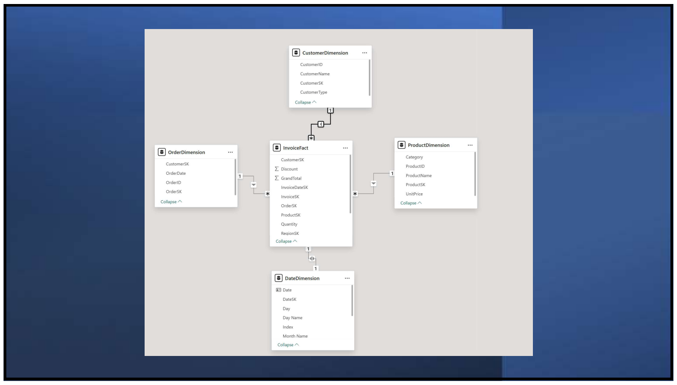
- **Dimensional modeling** is based on organizing data into two main types of tables:
- **Fact Tables** – Store quantitative business metrics (e.g., sales revenue, quantity sold).
- **Dimension Tables** – Store descriptive attributes that provide context to facts (e.g., customer details, product names, time periods).
- This structure enables **denormalization** (reducing complex joins) to speed up queries.

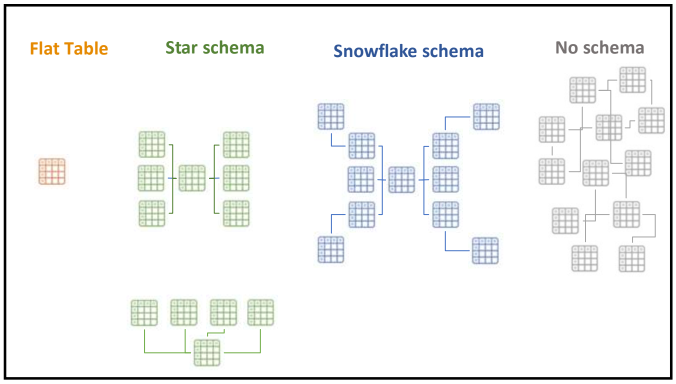


community.dynamics.com



community.dynamics.com





Flat Table

Star schema

Snowflake schema

No schema

1. One-to-Many (1:M) Relationship (Dimension to Fact)

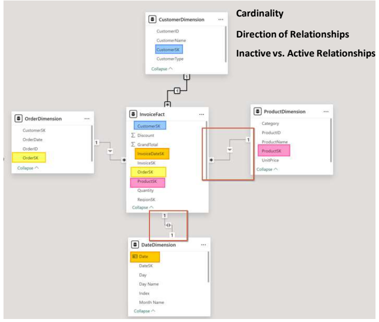
- Most common relationship in a star schema.
- Each Dimension Table has a Unique Value, Primary Key, Natural Key, Surrogate Key.
- The Fact table has a Foreign key referencing the dimension table that tie them together.
- One row in a dimension table can relate to many rows in the fact table.
- Dimension Tables filter Fact tables.
- One Way relationship.
    - Fact tables do not filter dimension tables. If a fact table filtered a dimension table, ambiguity is introduced. Think of a single row in an Excel spreadsheet that contained many rows, attempting to filter its own header row. It can be misleading.

For Example:

- - A Date dimension table has a unique Date Key.
- - The Sales fact table has a Date Key as a foreign key. (Several Date Column in fact)
- - One date in the Date table can relate to many sales transactions

2. Many-to-One (M:1) Relationship (Fact to Dimension)

- This is the reverse perspective of the one-to-many relationship.
- Each fact record refers to a single related dimension record or another table record.



Cardinality

Direction of Relationships

Inactive vs. Active Relationships

3. One-to-One (1:1) Relationship (Rare)

- This is less common in a star schema but may occur in cases where a dimension is split into multiple tables for performance or security reasons.

For Example:

A Customer table may store general information, while a Customer Details table stores sensitive information (like credit scores) in a separate, restricted table or if there is a junk table .

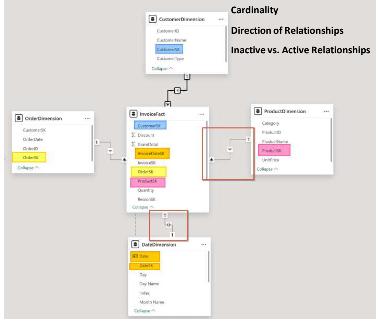
4. Many-to-Many (M:M) Relationships (Not Directly in Star Schema)

- Star schemas avoid direct many-to-many relationships. The Traditional star schema does not natively support **many-to-many relationships.**

These relationships are typically resolved using **bridge tables** (also called **junction tables**) in more complex data models.

For Example:

If Employees can be associated with multiple Roles, and each Role can belong to multiple Employees, a Role-Employee Bridge Table is used to break the many-to-many relationship into two one-to-many relationships.



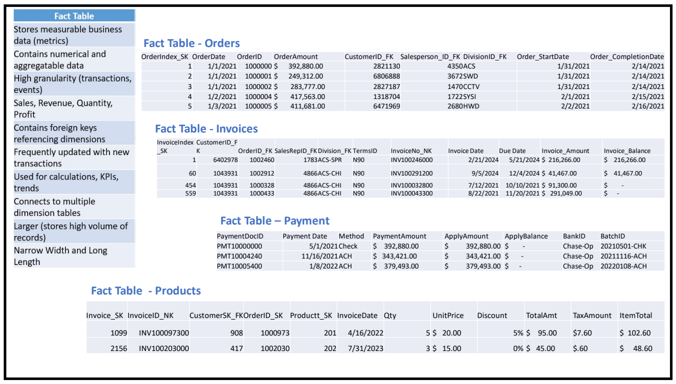
Cardinality

Direction of Relationships

Inactive vs. Active Relationships

Key Difference Between Fact & Dimensions Tables

Feature	Fact Table	Dimension Table
Purpose	Stores measurable business data (metrics)	Stores descriptive attributes (context)
Data Type	Contains numerical and aggregatable data	Contains categorical or descriptive data
Granularity	High granularity (transactions, events)	Lower granularity (business entities)
Examples	Sales, Revenue, Quantity, Profit	Product, Customer, Region, Date
Keys	Contains foreign keys referencing dimensions	Contains a primary key used in fact tables
Data Changes	Frequently updated with new transactions	Less frequently updated, usually static
Usage in BI	Used for calculations, KPIs, trends	Used for filtering, slicing, and dicing
Relationships	Connects to multiple dimension tables	Connected to fact tables using primary key
Data Size	Larger (stores high volume of records)	Smaller (stores descriptive data only)
Attributes	Narrow Width and Long Length	Wide Width, and Short Length



Fact Table
Stores measurable business data (metrics)
Contains numerical and aggregatable data
High granularity (transactions, events)
Sales, Revenue, Quantity, Profit
Contains foreign keys referencing dimensions
Frequently updated with new transactions
Used for calculations, KPIs, trends
Connects to multiple dimension tables
Larger (stores high volume of records)
Narrow Width and Long Length

Fact Table - Orders

OrderIndex_SK	OrderDate	OrderID	OrderAmount	CustomerID_FK	Salesperson_ID_FK	DivisionID_FK	Order_StartDate	Order_CompletionDate
1	1/1/2021	1000000	$ 392,880.00	2821130	4350	ACS	1/31/2021	2/14/2021
2	1/1/2021	1000001	$ 249,312.00	6806888	3672	SWD	1/31/2021	2/14/2021
3	1/1/2021	1000002	$ 283,777.00	2827187	1470	CCTV	1/31/2021	2/14/2021
4	1/2/2021	1000004	$ 417,563.00	1318704	1722	SYSI	2/1/2021	2/15/2021
5	1/3/2021	1000005	$ 411,681.00	6471969	2680	HWD	2/2/2021	2/16/2021

Fact Table - Invoices

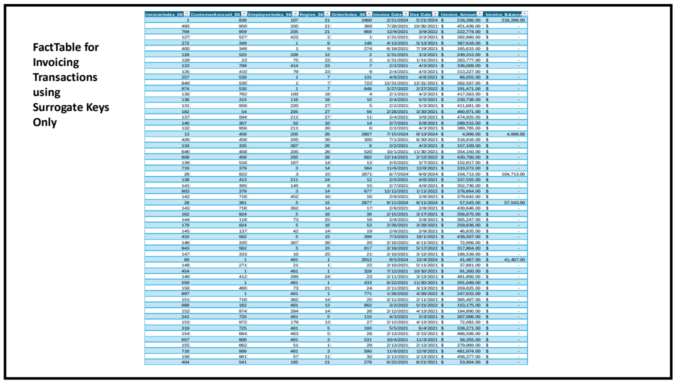
InvoiceIndex_SK	CustomerID_FK	OrderID_FK	SalesRepID_FK	Division_FK	TermsID	InvoiceNo_NK	Invoice Date	Due Date	Invoice_Amount	Invoice_Balance
1	6402978	1002460	1783	ACS-SPR	N90	INV100246000	2/21/2024	5/21/2024	$ 216,266.00	$ 216,266.00
60	1043931	1002912	4866	ACS-CHI	N90	INV100291200	9/5/2024	12/4/2024	$ 41,467.00	$ 41,467.00
454	1043931	1000328	4866	ACS-CHI	N90	INV100032800	7/12/2021	10/10/2021	$ 91,300.00	$ -
559	1043931	1000433	4866	ACS-CHI	N90	INV100043300	8/22/2021	11/20/2021	$ 291,049.00	$ -

Fact Table – Payment

PaymentDocID	Payment Date	Method	PaymentAmount	ApplyAmount	ApplyBalance	BankID	BatchID
PMT10000000	5/1/2021	Check	$ 392,880.00	$ 392,880.00	$ -	Chase-Op	20210501-CHK
PMT10004240	11/16/2021	ACH	$ 343,421.00	$ 343,421.00	$ -	Chase-Op	20211116-ACH
PMT10005400	1/8/2022	ACH	$ 379,493.00	$ 379,493.00	$ -	Chase-Op	20220108-ACH

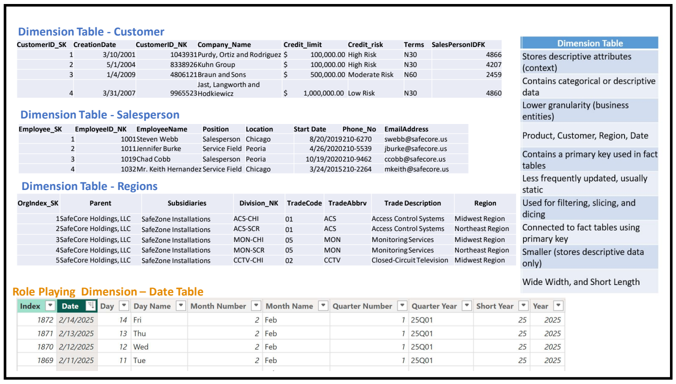
Fact Table - Products

Invoice_SK	InvoiceID_NK	CustomerSK_FK	OrderID_SK	Productt_SK	InvoiceDate	Qty	UnitPrice	Discount	TotalAmt	TaxAmount	ItemTotal
1099	INV100097300	908	1000973	201	4/16/2022	5	$ 20.00	5%	$ 95.00	$7.60	$ 102.60
2156	INV100203000	417	1002030	202	7/31/2023	3	$ 15.00	0%	$ 45.00	$.60	$ 48.60



FactTable for Invoicing Transactions using Surrogate Keys Only

InvoiceIndex_SK	CustomerAccount_SK	EmployeeIndex_SK	Region_SK	OrderIndex_SK	Invoice Date	Due Date	Invoice_Amoun	Invoice_Balance
1	838	107	21	2460	2/21/2024	5/21/2024	$ 216,266.00	$ 216,266.00
495	959	205	21	369	7/28/2021	10/26/2021	$ 451,439.00	$ -
794	959	205	21	668	12/9/2021	3/9/2022	$ 222,774.00	$ -
127	527	422	2	1	1/31/2021	3/2/2021	$ 392,880.00	$ -
272	349	1	9	146	4/13/2021	5/13/2021	$ 387,618.00	$ -
400	349	1	9	274	6/19/2021	7/19/2021	$ 165,615.00	$ -
128	525	326	12	2	1/31/2021	3/2/2021	$ 249,312.00	$ -
129	23	75	23	3	1/31/2021	1/31/2021	$ 283,777.00	$ -
133	799	414	23	7	2/2/2021	4/3/2021	$ 336,069.00	$ -
135	410	79	23	9	2/4/2021	4/5/2021	$ 313,227.00	$ -
257	530	1	7	131	4/8/2021	4/8/2021	$ 66,055.00	$ -
849	530	1	7	723	12/31/2021	12/31/2021	$ 262,507.00	$ -
974	530	1	7	848	2/27/2022	2/27/2022	$ 191,471.00	$ -
130	792	100	18	4	2/1/2021	4/2/2021	$ 417,563.00	$ -
136	315	116	18	10	2/4/2021	5/5/2021	$ 230,738.00	$ -
131	958	220	27	5	2/2/2021	5/3/2021	$ 411,681.00	$ -
182	54	205	27	56	2/28/2021	3/30/2021	$ 460,971.00	$ -
137	594	211	27	11	2/4/2021	3/6/2021	$ 474,925.00	$ -
140	307	52	10	14	2/7/2021	5/8/2021	$ 289,515.00	$ -
132	956	211	26	6	2/2/2021	4/3/2021	$ 389,785.00	$ -
13	458	205	26	2807	7/15/2024	9/13/2024	$ 4,906.00	$ 4,906.00
426	458	205	26	300	7/1/2021	8/30/2021	$ 318,816.00	$ -
134	335	367	26	8	2/2/2021	4/3/2021	$ 157,109.00	$ -
646	458	205	26	520	10/1/2021	11/30/2021	$ 354,150.00	$ -
808	458	205	26	682	12/14/2021	2/12/2022	$ 430,700.00	$ -
139	534	167	14	13	2/5/2021	3/7/2021	$ 152,917.00	$ -
710	379	3	14	584	11/6/2021	12/6/2021	$ 333,072.00	$ -
26	922	3	15	2871	8/7/2024	9/6/2024	$ 104,713.00	$ 104,713.00
138	413	211	24	12	2/5/2021	4/6/2021	$ 337,555.00	$ -
141	305	145	8	15	2/7/2021	4/8/2021	$ 352,736.00	$ -
803	379	3	14	677	12/12/2021	1/11/2022	$ 378,664.00	$ -
142	718	452	16	16	2/8/2021	2/8/2021	$ 379,642.00	$ -
28	381	5	15	2877	8/11/2024	8/11/2024	$ 57,543.00	$ 57,543.00
143	716	362	14	17	2/8/2021	2/8/2021	$ 430,646.00	$ -
162	924	5	16	36	2/15/2021	3/17/2021	$ 356,875.00	$ -
144	118	73	25	18	2/8/2021	2/8/2021	$ 365,247.00	$ -
179	924	5	16	53	2/26/2021	3/28/2021	$ 259,836.00	$ -
145	137	42	14	19	2/9/2021	2/9/2021	$ 46,835.00	$ -
432	562	5	15	306	7/3/2021	10/1/2021	$ 438,507.00	$ -
146	335	367	26	20	2/10/2021	4/11/2021	$ 72,956.00	$ -
943	562	5	15	817	2/16/2022	5/17/2022	$ 317,864.00	$ -
147	333	10	25	21	2/10/2021	3/12/2021	$ 186,539.00	$ -
60	1	481	1	2912	9/5/2024	12/4/2024	$ 41,467.00	$ 41,467.00
148	271	21	1	22	2/10/2021	5/11/2021	$ 37,881.00	$ -
454	1	481	1	328	7/12/2021	10/10/2021	$ 91,300.00	$ -
149	412	269	24	23	2/11/2021	3/13/2021	$ 481,800.00	$ -
559	1	481	1	433	8/22/2021	11/20/2021	$ 291,049.00	$ -
150	480	73	21	24	2/11/2021	3/13/2021	$ 359,825.00	$ -
897	1	481	1	771	1/26/2022	4/26/2022	$ 247,632.00	$ -
151	716	362	14	25	2/11/2021	2/11/2021	$ 365,497.00	$ -
988	182	481	13	862	3/2/2022	5/31/2022	$ 153,175.00	$ -
152	974	284	14	26	2/12/2021	4/13/2021	$ 184,890.00	$ -
241	725	481	5	115	4/3/2021	5/3/2021	$ 307,086.00	$ -
153	972	179	13	27	2/12/2021	4/13/2021	$ 72,082.00	$ -
319	725	481	5	193	5/5/2021	6/4/2021	$ 338,271.00	$ -
154	664	463	5	28	2/13/2021	3/15/2021	$ 466,506.00	$ -
657	906	481	3	531	10/4/2021	11/3/2021	$ 58,355.00	$ -
155	662	51	1	29	2/13/2021	2/13/2021	$ 379,969.00	$ -
716	906	481	3	590	11/8/2021	12/8/2021	$ 481,974.00	$ -
156	991	57	11	30	2/13/2021	2/13/2021	$ 456,377.00	$ -
404	541	185	21	278	6/22/2021	8/21/2021	$ 53,904.00	$ -



Dimension Table - Customer

CustomerID_SK	CreationDate	CustomerID_NK	Company_Name	Credit_limit	Credit_risk	Terms	SalesPersonIDFK
1	3/10/2001	1043931	Purdy, Ortiz and Rodriguez	$ 100,000.00	High Risk	N30	4866
2	5/1/2004	8338926	Kuhn Group	$ 100,000.00	High Risk	N30	4207
3	1/4/2009	4806121	Braun and Sons	$ 500,000.00	Moderate Risk	N60	2459
4	3/31/2007	9965523	Jast, Langworth and Hodkiewicz	$ 1,000,000.00	Low Risk	N30	4860

Dimension Table - Salesperson

Employee_SK	EmployeeID_NK	EmployeeName	Position	Location	Start Date	Phone_No	EmailAddress
1	1001	Steven Webb	Salesperson	Chicago	8/20/2019	210-6270	swebb@safecore.us
2	1011	Jennifer Burke	Service Field	Peoria	4/26/2020	210-5539	jburke@safecore.us
3	1019	Chad Cobb	Salesperson	Peoria	10/19/2020	210-9462	ccobb@safecore.us
4	1032	Mr. Keith Hernandez	Service Field	Chicago	3/24/2015	210-2264	mkeith@safecore.us

Dimension Table - Regions

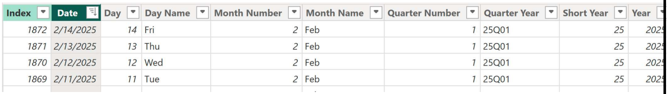
OrgIndex_SK	Parent	Subsidiaries	Division_NK	TradeCode	TradeAbbrv	Trade Description	Region
1	SafeCore Holdings, LLC	SafeZone Installations	ACS-CHI	01	ACS	Access Control Systems	Midwest Region
2	SafeCore Holdings, LLC	SafeZone Installations	ACS-SCR	01	ACS	Access Control Systems	Northeast Region
3	SafeCore Holdings, LLC	SafeZone Installations	MON-CHI	05	MON	Monitoring Services	Midwest Region
4	SafeCore Holdings, LLC	SafeZone Installations	MON-SCR	05	MON	Monitoring Services	Northeast Region
5	SafeCore Holdings, LLC	SafeZone Installations	CCTV-CHI	02	CCTV	Closed-Circuit Television	Midwest Region

Dimension Table
Stores descriptive attributes (context)
Contains categorical or descriptive data
Lower granularity (business entities)
Product, Customer, Region, Date
Contains a primary key used in fact tables
Less frequently updated, usually static
Used for filtering, slicing, and dicing
Connected to fact tables using primary key
Smaller (stores descriptive data only)
Wide Width, and Short Length

Role Playing Dimension – Date Table

Index	Date	Day	Day Name	Month Number	Month Name	Quarter Number	Quarter Year	Short Year	Year
1872	*2/14/2025*	*14*	Fri	*2*	Feb	*1*	25Q01	*25*	*2025*
1871	*2/13/2025*	*13*	Thu	*2*	Feb	*1*	25Q01	*25*	*2025*
1870	*2/12/2025*	*12*	Wed	*2*	Feb	*1*	25Q01	*25*	*2025*
1869	*2/11/2025*	*11*	Tue	*2*	Feb	*1*	25Q01	*25*	*2025*

Date Table – Time Intelligence



Index	Date	Day	Day Name	Month Number	Month Name	Quarter Number	Quarter Year	Short Year	Year
1872	2/14/2025	14	Fri	2	Feb	1	25Q01	25	2025
1871	2/13/2025	13	Thu	2	Feb	1	25Q01	25	2025
1870	2/12/2025	12	Wed	2	Feb	1	25Q01	25	2025
1869	2/11/2025	11	Tue	2	Feb	1	25Q01	25	2025

- **Continuous Date Range -** no missing oe duplicate dates, typically spans from the earliest to the latest transaction date in your dataset. To support future forecasts, it will need to extend beyond the current date (e.g., 2–3 years into the future).
- **One Row per Day** - Each row should represent a single date (no duplicates).
- **Must Include a Column of Type 'Date'** - At least one column formatted as Date (not DateTime)as the Primary Key.
- **Must be Mark as a Date TableIn Power BI** - Table Tools > Mark as Date Table > Select Date Column
- **Sort By Other Columns** – You will want to Sort Month Name by Month Number, so your visual months are in order.
- **Standard Date Attributes** - The table should include additional columns to support various time-based filtering and grouping: Year: Qtr, YYYY-MM, Full Month Name, Period, Weeks, Date Calculations, Supports Fiscal Year
- **No Relationships to Fact Tables with Duplicate Date Columns** - Only the Date column from the Date Table should be used to create relationships with fact tables. Avoid using multiple date fields from fact tables—use USERELATIONSHIP() in DAX for alternate date fields (e.g., Order Date vs. Invoice Date).

Date Table – DAX Method

```
DateTable =
ADDCOLUMNS (
   CALENDAR (DATE(2015,1,1), DATE(2030,12,31)),
   "Year", YEAR([Date]),
   "Month", FORMAT([Date], "MMMM"),
   "Month Number", MONTH([Date]),
   "Quarter", "Q" & FORMAT([Date], "Q"),
   "Day of Week", FORMAT([Date], "dddd"),
   "Day of Year", FORMAT([Date], "DDD"),
   "Week Number", WEEKNUM([Date])
)
```

Date Table – 'M' Method

```
let
Source = List.Dates,
#"Invoked FunctionSource" = Source(#date(2020, 1, 1), Duration.Days(DateTime.Date(DateTime.FixedLocalNow()) - #date(2020,1,1)), #duration(1, 0, 0, 0)),
#"Table from List" = Table.FromList(#"Invoked FunctionSource", Splitter.SplitByNothing(), null, null, ExtraValues.Error),
#"Added Index" = Table.AddIndexColumn(#"Table from List", "Index", 1, 1),
#"Renamed Columns" = Table.RenameColumns(#"Added Index",{{"Column1", "Date"}}),
#"Added Custom" = Table.AddColumn(#"Renamed Columns", "Year", each Date.Year([Date])),
#"Added Custom1" = Table.AddColumn(#"Added Custom", "Month Number", each Date.Month([Date])),
#"Added Custom2" = Table.AddColumn(#"Added Custom1", "Day", each Date.Day([Date])),
#"Added Custom3" = Table.AddColumn(#"Added Custom2", "Day Name", each Date.ToText([Date],"ddd")),
#"Added Custom4" = Table.AddColumn(#"Added Custom3", "Month Name", each Date.ToText([Date],"MMM")),
#"Reordered Columns" = Table.ReorderColumns(#"Added Custom4",{"Date", "Index", "Year", "Month Number", "Month Name", "Day", "Day Name"}),
#"Added Custom5" = Table.AddColumn(#"Reordered Columns", "Quarter Number", each Date.QuarterOfYear([Date])),
#"Duplicated Column" = Table.DuplicateColumn(#"Added Custom5", "Year", "Copy of Year"),
#"Renamed Columns1" = Table.RenameColumns(#"Duplicated Column",{{"Copy of Year", "Short Year"}}),
#"Changed Type" = Table.TransformColumnTypes(#"Renamed Columns1",{{"Short Year", type text}}),
#"Split Column by Position" = Table.SplitColumn(#"Changed Type","Short Year",Splitter.SplitTextByRepeatedLengths(2),{"Short Year.1", "Short Year.2"}),
#"Changed Type1" = Table.TransformColumnTypes(#"Split Column by Position",{{"Short Year.1", Int64.Type}, {"Short Year.2", Int64.Type}}),
#"Removed Columns" = Table.RemoveColumns(#"Changed Type1",{"Short Year.1"}),
#"Renamed Columns2" = Table.RenameColumns(#"Removed Columns",{{"Short Year.2", "Short Year"}}),
#"Added Custom6" = Table.AddColumn(#"Renamed Columns2", "Quarter Year", each Number.ToText([Short Year]) & "Q" & Number.ToText([Quarter Number],"00")),
#"Reordered Columns1" = Table.ReorderColumns(#"Added Custom6",{"Index", "Date", "Day", "Day Name", "Month Number", "Month Name", "Quarter Number",
"Quarter Year", "Short Year", "Year"}),
#"Changed Type2" = Table.TransformColumnTypes(#"Reordered Columns1",{{"Date", type date}, {"Day", Int64.Type}, {"Index", Int64.Type}, {"Month Number",
Int64.Type}, {"Quarter Number", Int64.Type}, {"Month Name", type text}, {"Quarter Year", type text}, {"Year", Int64.Type}})
in
#"Changed Type2"
```

Basic Power BI Navigation

Importing Data

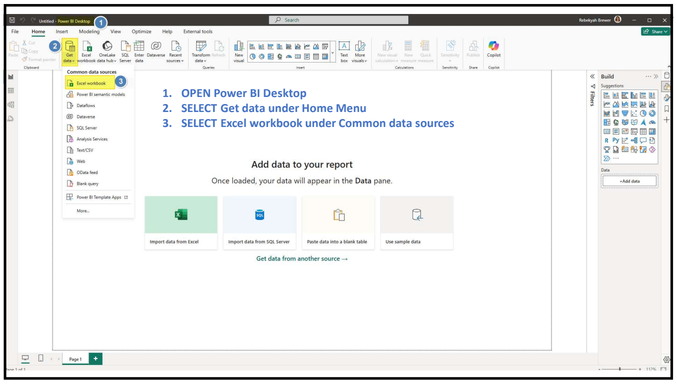
- Connecting Power BI to dataset.

Building Simple Data Model

- Connect our Data Model
- Create a Date Table
- Mark our Date Table

Building Simple Reports

- Dragging fields into visuals.
- Formatting charts and tables.



Common data sources

- Excel workbook
- Power BI semantic models
- Dataflows
- Dataverse
- SQL Server
- Analysis Services
- Text/CSV
- Web
- OData feed
- Blank query
- Power BI Template Apps
- More...

1. OPEN Power BI Desktop
2. SELECT Get data under Home Menu
3. SELECT Excel workbook under Common data sources

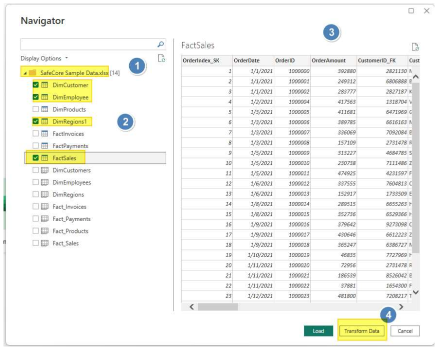
Add data to your report

Once loaded, your data will appear in the **Data** pane.

Import data from Excel | Import data from SQL Server | Paste data into a blank table | Use sample data

Get data from another source →

1. OPEN Example File: *SafeCore Sample Date.xlsx*
2. SELECT Tables to import:
   - DimCustomer
   - DimEmployee
   - DimRegions
   - FactSales
3. PREVIEW Sample Data
4. CLICK Transform Data to open Power Query



SafeCore Sample Data.xlsx [14]

- ☑ DimCustomer
- ☑ DimEmployee
- □ DimProducts
- ☑ DimRegions1
- □ FactInvoices
- □ FactPayments
- ☑ FactSales
- □ DimCustomers
- □ DimEmployees
- □ DimRegions
- □ Fact_Invoices
- □ Fact_Payments
- □ Fact_Products
- □ Fact_Sales

FactSales

OrderIndex_SK	OrderDate	OrderID	OrderAmount	CustomerID_FK	Cust
1	1/1/2021	1000000	392880	2821130	N
2	1/1/2021	1000001	249312	6806888	E
3	1/1/2021	1000002	283777	2827187	K
4	1/2/2021	1000004	417563	1318704	V
5	1/3/2021	1000005	411681	6471969	G
6	1/3/2021	1000006	389785	6616163	M
7	1/3/2021	1000007	336069	7092084	E
8	1/3/2021	1000008	157109	2731478	R
9	1/5/2021	1000009	313227	4684785	S
10	1/5/2021	1000010	230738	7111486	Z
11	1/5/2021	1000011	474925	4231597	F
12	1/6/2021	1000012	337555	7604813	C
13	1/6/2021	1000013	152917	1733509	E
14	1/8/2021	1000014	289515	6655263	H
15	1/8/2021	1000015	352736	6529366	H
16	1/9/2021	1000016	379642	9273098	C
17	1/9/2021	1000017	430646	6612223	Z
18	1/9/2021	1000018	365247	6386727	M
19	1/10/2021	1000019	46835	7727969	H
20	1/11/2021	1000020	72956	2731478	R
21	1/11/2021	1000021	186539	8526042	E
22	1/11/2021	1000022	37881	1654300	F
23	1/12/2021	1000023	481800	7208217	T

Step 1

Create Group Folders to organize your data sources:

- *Datasources*
- *Staging Table (Optional)*
- *DataModel*

Step 2

- Move your date source files to your Datasource folder
- Disable load for each

Step 3

- Reference each of your tables to create copies. (Right Click choose reference)
- Move to your DataModel folder
- Rename for user friendliness.

Step 4

- Set your data types.
- Rename your Columns to make them user friendly in natural language.
- Remove Extra Columns.
- Perform additional transformations as needed.

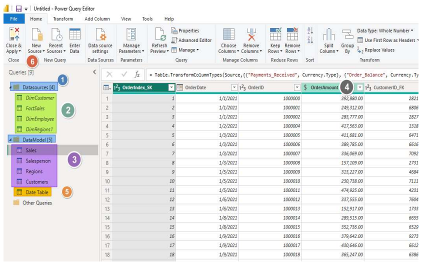
Step 5

- Create Your Date Table

(Use .txt file/Select Blank Query/Advanced Editor)

Step 6

- Click Close and Apply

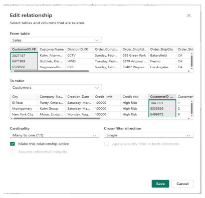


```
= Table.TransformColumnTypes(Source,{{"Payments_Received", Currency.Type}, {"Order_Balance", Currency.Ty
```

	OrderIndex_SK	OrderDate	OrderID	OrderAmount	CustomerID_FK
1	1	1/1/2021	1000000	392,880.00	2821
2	2	1/1/2021	1000001	249,312.00	6806
3	3	1/1/2021	1000002	283,777.00	2827
4	4	1/2/2021	1000004	417,563.00	1318
5	5	1/3/2021	1000005	411,681.00	6471
6	6	1/3/2021	1000006	389,785.00	6616
7	7	1/3/2021	1000007	336,069.00	7092
8	8	1/3/2021	1000008	157,109.00	2731
9	9	1/5/2021	1000009	313,227.00	4684
10	10	1/5/2021	1000010	230,738.00	7111
11	11	1/5/2021	1000011	474,925.00	4231
12	12	1/6/2021	1000012	337,555.00	7604
13	13	1/6/2021	1000013	152,917.00	1733
14	14	1/8/2021	1000014	289,515.00	6655
15	15	1/8/2021	1000015	352,736.00	6529
16	16	1/9/2021	1000016	379,642.00	9273
17	17	1/9/2021	1000017	430,646.00	6612
18	18	1/9/2021	1000018	365,247.00	6386

Create your star schema

1. Select DataModel
2. Select Manage relationships
3. Select Fact Table as "From". Your Dimension Table as "To" for each.
4. Verify your Cardinality is M:1
5. Verify Relationship Direction is Single.
6. Verify relationship is Active
7. Arrange Your tables as a Star Schema or as hierarchy.



Edit relationship

Select tables and columns that are related.

From table: Sales

CustomerID_FK	CustomerName	DivisionID_FK	Order_Compl...	Order_ShipAd...	Order_ShipCity	Order_Shi
2827187	Kuhn, Altenw...	CCTV	Sunday, Febr...	595 Green Park	Bakersfield	CA
6471969	Gottlieb, Kris ...	HWD	Tuesday, Febr...	6379 Arizona ...	Fresno	CA
6529366	Hegmann-Bo...	CYB	Sunday, Febr...	32407 Waywo...	Los Angeles	CA

To table: Customers

City	Company_Na...	Creation_Date	Credit_limit	Credit_risk	CustomerID_...	Customer
El Paso	Purdy, Ortiz a...	Saturday, Mar...	100000	High Risk	1043931	1
Montgomery	Kuhn Group	Saturday, Ma...	100000	High Risk	8338926	2
New York City	Morar, Lindgr...	Monday, Aug...	100000	High Risk	6499972	8

Cardinality: Many to one (*:1)

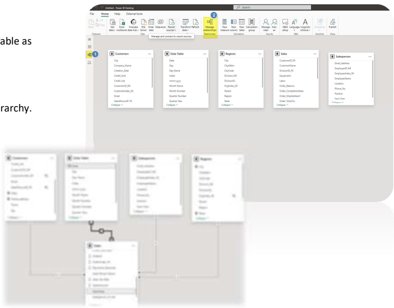
Cross-filter direction: Single

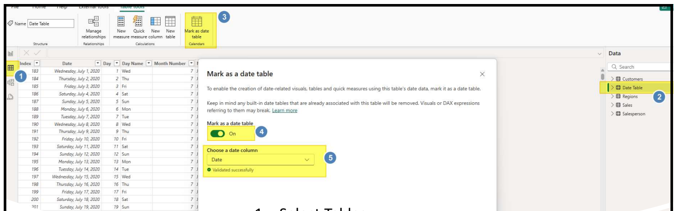
☑ Make this relationship active

☐ Apply security filter in both directions

☐ Assume referential integrity

Save Cancel





Index	Date	Day	Day Name	Month Number
183	Wednesday, July 1, 2020	1	Wed	7
184	Thursday, July 2, 2020	2	Thu	7
185	Friday, July 3, 2020	3	Fri	7
186	Saturday, July 4, 2020	4	Sat	7
187	Sunday, July 5, 2020	5	Sun	7
188	Monday, July 6, 2020	6	Mon	7
189	Tuesday, July 7, 2020	7	Tue	7
190	Wednesday, July 8, 2020	8	Wed	7
191	Thursday, July 9, 2020	9	Thu	7
192	Friday, July 10, 2020	10	Fri	7
193	Saturday, July 11, 2020	11	Sat	7
194	Sunday, July 12, 2020	12	Sun	7
195	Monday, July 13, 2020	13	Mon	7
196	Tuesday, July 14, 2020	14	Tue	7
197	Wednesday, July 15, 2020	15	Wed	7
198	Thursday, July 16, 2020	16	Thu	7
199	Friday, July 17, 2020	17	Fri	7
200	Saturday, July 18, 2020	18	Sat	7
201	Sunday, July 19, 2020	19	Sun	7
	Monday, July 20, 2020	20	Mon	7
	[illegible] 21, 2020	21	Tue	7
	[illegible] 22, 2020	[illegible]	Wed	7

Mark as a date table

To enable the creation of date-related visuals, tables and quick measures using this table's date data, mark it as a date table.

Keep in mind any built-in date tables that are already associated with this table will be removed. Visuals or DAX expressions referring to them may break. Learn more

Mark as a date table: On

Choose a date column: Date

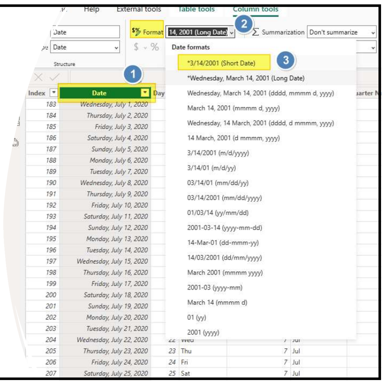
Validated successfully

Mark Your Date Table

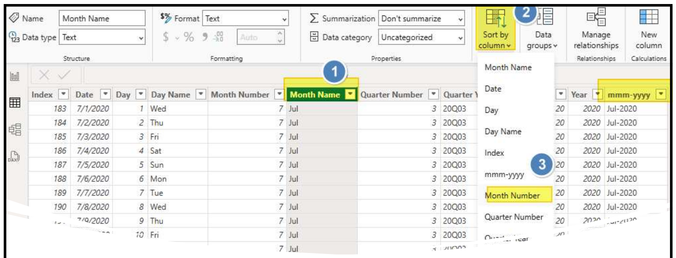
1. Select Tables
2. Select Date Table
3. Click 'Mark as date table' under Table tools menu
4. Select 'On'
5. Select your Primary Date Column. Must be consecutive.
6. Click Save

Format Your Date Table

- Select Date Column
- Select Column tools and Format
- Choose (Short Date)
- Repeat on other columns as needed.



Index	Date	Day			
183	Wednesday, July 1, 2020				
184	Thursday, July 2, 2020				
185	Friday, July 3, 2020				
186	Saturday, July 4, 2020				
187	Sunday, July 5, 2020				
188	Monday, July 6, 2020				
189	Tuesday, July 7, 2020				
190	Wednesday, July 8, 2020				
191	Thursday, July 9, 2020				
192	Friday, July 10, 2020				
193	Saturday, July 11, 2020				
194	Sunday, July 12, 2020				
195	Monday, July 13, 2020				
196	Tuesday, July 14, 2020				
197	Wednesday, July 15, 2020				
198	Thursday, July 16, 2020				
199	Friday, July 17, 2020				
200	Saturday, July 18, 2020				
201	Sunday, July 19, 2020				
202	Monday, July 20, 2020				
203	Tuesday, July 21, 2020				
204	Wednesday, July 22, 2020	22	Wed	7	Jul
205	Thursday, July 23, 2020	23	Thu	7	Jul
206	Friday, July 24, 2020	24	Fri	7	Jul
207	Saturday, July 25, 2020	25	Sat	7	Jul



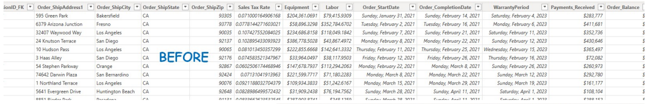
Name: Month Name
Data type: Text
Format: Text
Summarization: Don't summarize
Data category: Uncategorized

Index	Date	Day	Day Name	Month Number	Month Name	Quarter Number	Quarter Y		Year	mmm-yyyy
183	7/1/2020	1	Wed	7	Jul	3	20Q03	20	2020	Jul-2020
184	7/2/2020	2	Thu	7	Jul	3	20Q03	20	2020	Jul-2020
185	7/3/2020	3	Fri	7	Jul	3	20Q03	20	2020	Jul-2020
186	7/4/2020	4	Sat	7	Jul	3	20Q03	20	2020	Jul-2020
187	7/5/2020	5	Sun	7	Jul	3	20Q03	20	2020	Jul-2020
188	7/6/2020	6	Mon	7	Jul	3	20Q03	20	2020	Jul-2020
189	7/7/2020	7	Tue	7	Jul	3	20Q03	20	2020	Jul-2020
190	7/8/2020	8	Wed	7	Jul	3	20Q03	20	2020	Jul-2020
	7/9/2020	9	Thu	7	Jul	3	20Q03	20		
		10	Fri	7	Jul	3	20Q03			

Sort your Calendar Columns Correctly

1. Select a column that needs a specific order. i.e… Month Name or mmm-yyyy column. You do not want these to display alphabetical in visuals.
2. Select Sort by Column
3. Select the corresponding numeric value to use as sort order.
4. Create a visual to validate

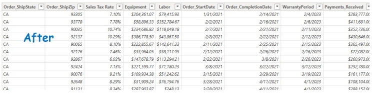
Format All Your Other Tables Data Types in Power BI



ionID_FK	Order_ShipAddress1	Order_ShipCity	Order_ShipState	Order_ShipZip	Sales Tax Rate	Equipment	Labor	Order_StartDate	Order_CompletionDate	WarrantyPeriod	Payments_Received	Order_Balance
	595 Green Park	Bakersfield	CA	93305	0.071000164906168	$204,361.0691	$79,415.9309	Sunday, January 31, 2021	Sunday, February 14, 2021	Saturday, February 4, 2023	$283,777	$
	6379 Arizona Junction	Fresno	CA	93778	0.0778144271603021	$58,896.3298	$352,784.6702	Tuesday, February 2, 2021	Tuesday, February 16, 2021	Monday, February 6, 2023	$411,681	$
	32407 Waywood Way	Los Angeles	CA	90035	0.107427552084025	$234,686.8158	$118,049.1842	Sunday, February 7, 2021	Sunday, February 21, 2021	Saturday, February 11, 2023	$352,736	$
	24 Knutson Terrace	San Diego	CA	92137	0.102895433093923	$386,778.5028	$43,867.4972	Monday, February 8, 2021	Monday, February 22, 2021	Sunday, February 12, 2023	$430,646	$
	10 Hudson Pass	Los Angeles	CA	90065	0.081013450357299	$222,855.6668	$142,641.3332	Thursday, February 11, 2021	Thursday, February 25, 2021	Wednesday, February 15, 2023	$365,497	$
	3 Haas Alley	San Diego	CA	92176	0.074583521347967	$33,964.0497	$38,117.9503	Friday, February 12, 2021	Friday, February 26, 2021	Thursday, February 16, 2023	$72,082	$
	54 Stephen Parkway	Orange	CA	92867	0.0602506174468946	$147,678.7937	$113,294.2063	Monday, February 22, 2021	Monday, March 8, 2021	Sunday, February 26, 2023	$260,973	$
	74642 Darwin Plaza	San Bernardino	CA	92424	0.07131041913963	$221,599.7717	$71,180.2283	Monday, March 8, 2021	Monday, March 22, 2021	Sunday, March 12, 2023	$292,780	$
	1 Northland Terrace	Los Angeles	CA	90076	0.0921188032704379	$109,934.3833	$51,242.6167	Monday, March 15, 2021	Monday, March 29, 2021	Sunday, March 19, 2023	$161,177	$
	5641 Evergreen Drive	Huntington Beach	CA	92648	0.0828986499572432	$31,909.2438	$76,194.7562	Sunday, March 28, 2021	Sunday, April 11, 2021	Saturday, April 1, 2023	$108,104	$

Tips:

- Use short dates
- Set all Currency & Percentage formats to 2 decimals instead of 'Auto' default.
- Select Numeric values that will not be aggregated to ' Don't Summarize' in Column Tools.
- You may need to return to Power Query to fix some columns data types that cannot be fixed in Power BI
- Set any data Categories for State, Address, Zip clms.
- 'Hide in Report View' Columns for Surrogate 'Keys' in Fact Tables and Dimensions.
- **'Hide in Report View' All Implicit Measure Fields, Fields that auto sum that you will create Explicit Measures for to replace unless you need them to filter.



Order_ShipState	Order_ShipZip	Sales Tax Rate	Equipment	Labor	Order_StartDate	Order_CompletionDate	WarrantyPeriod	Payments_Received
CA	93305	7.10%	$204,361.07	$79,415.93	1/31/2021	2/14/2021	2/4/2023	$283,777.00
CA	93778	7.78%	$58,896.33	$352,784.67	2/2/2021	2/16/2021	2/6/2023	$411,681.00
CA	90035	10.74%	$234,686.82	$118,049.18	2/7/2021	2/21/2021	2/11/2023	$352,736.00
CA	92137	10.29%	$386,778.50	$43,867.50	2/8/2021	2/22/2021	2/12/2023	$430,646.00
CA	90065	8.10%	$222,855.67	$142,641.33	2/11/2021	2/25/2021	2/15/2023	$365,497.00
CA	92176	7.46%	$33,964.05	$38,117.95	2/12/2021	2/26/2021	2/16/2023	$72,082.00
CA	92867	6.03%	$147,678.79	$113,294.21	2/22/2021	3/8/2021	2/26/2023	$260,973.00
CA	92424	7.13%	$221,599.77	$71,180.23	3/8/2021	3/22/2021	3/12/2023	$292,780.00
CA	90076	9.21%	$109,934.38	$51,242.62	3/15/2021	3/29/2021	3/19/2023	$161,177.00
CA	92648	8.29%	$31,909.24	$76,194.76	3/28/2021	4/11/2021	4/1/2023	$108,104.00

Date Table – DAX Method

```
DateTable =
ADDCOLUMNS (
    CALENDAR (DATE(2015,1,1), DATE(2030,12,31)),
    "Year", YEAR([Date]),
    "Month", FORMAT([Date], "MMMM"),
    "Month Number", MONTH([Date]),
    "Quarter", "Q" & FORMAT([Date], "Q"),
    "Day of Week", FORMAT([Date], "dddd"),
    "Day of Year", FORMAT([Date], "DDD"),
    "Week Number", WEEKNUM([Date])
)
```

PowerPivot Excel Comparison

Importing Data

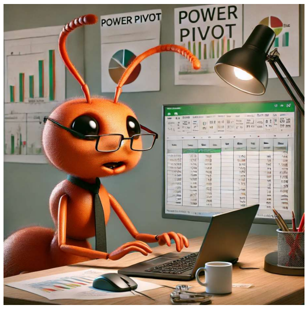
- Connecting Power BI to dataset.

Building Simple Data Model

- Connect our Data Model
- Create a Date Table
- Mark our Date Table

Building Simple Reports

- Dragging fields into visuals.
- Formatting charts and tables.

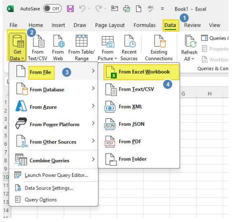


Benefits of Learning Power Pivot in Excel:

1. Excel is familiar territory for all of us
2. Increased opportunities for internal use leads to wider application
3. Increased usage cases leads to increased experience
4. Small quick wins will encourage motivation to learn more.
5. Logical transition to Power PI through PowerPivot

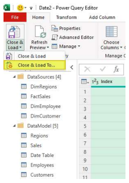
Loading Power Pivot

Step 1. Load Data into Power Query from Excel

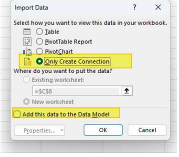


Step 2. Transform Data in Power Query as Needed for Dimension and Fact Tables then 'Close & Load To..' option.

Note Original Tables in DataSources have no option not to load.



Step 3. Select Only Create Connection. If working with just DataModel Tables select 'Add this data to the Data Model to bypass Step 4.


Import Data

Select how you want to view this data in your workbook.

○ Table
○ PivotTable Report
○ PivotChart
● Only Create Connection

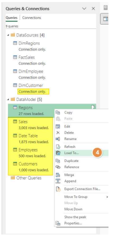
Where do you want to put the data?

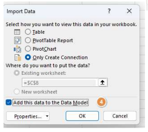
○ Existing worksheet:
=$C$8
○ New worksheet

□ Add this data to the Data Model

Loading Power Pivot

STEP 4. All your tables will be under 'Connection only' under Queries & Connections on Tata Tab. Select each of the Tables under your Data Model and click 'Load To:' and then select '"Add this data to Data Model." This avoided loading all tables and only the tables you need. For a visual copy, select 'Table' to create an Excel Table of your data.




Import Data

Select how you want to view this data in your workbook.

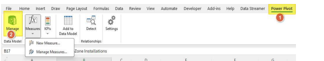
○ Table
○ PivotTable Report
○ PivotChart
● Only Create Connection

Where do you want to put the data?

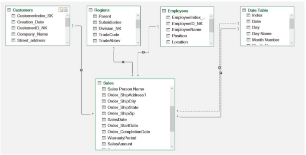
○ Existing worksheet:
=$C$8
○ New worksheet

☑ Add this data to the Data Model

STEP 5.Open Power Pivot to edit your tables in Power Pivot and create your data model. Click Manage to access Power Pivot area of Excel.

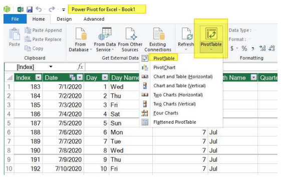


STEP 6. Set your date table, format your column values, create calculated columns. Connect your relationships for your Data Model.

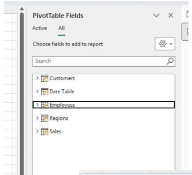


Loading Power Pivot

STEP 7. Under Home on Menu, Select PivotTable or PivotChart to begin creating your PowerPivot Table and PowerPivot Charts.



	Index	Date	Day	Day Name	th Name	Quarte
1	183	7/1/2020	1	Wed		
2	184	7/2/2020	2	Thu		
3	185	7/3/2020	3	Fri		
4	186	7/4/2020	4	Sat		
5	187	7/5/2020	5	Sun		
6	188	7/6/2020	6	Mon	7	Jul
7	189	7/7/2020	7	Tue	7	Jul
8	190	7/8/2020	8	Wed	7	Jul
9	191	7/9/2020	9	Thu	7	Jul
10	192	7/10/2020	10	Fri	7	Jul



STEP 8. Begin adding fields to your PowerPivot Table as you would a normal Pivot Table, except now you have Multi-tables.

STEP 9. Create your DAX measures if needed for additional calculations. (Sign up for next class: Discovering DAX!)

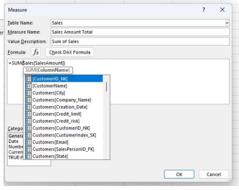

Measure

Table Name: Sales

Measure Name: Sales Amount Total

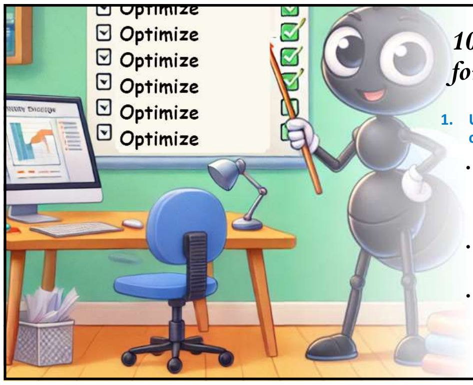
Value Description: Sum of Sales

Formula:

```
=SUM(Sales[SalesAmount])
```

Excel Power Pivot vs Power BI Considerations

Feature/Aspect	Power Pivot in Excel	Power BI
Purpose	Enhances Excel's data modeling and analysis capabilities Good for small scale, one time, ad-hoc analysis reports.	A full-fledged BI and reporting tool for analytics Power BI is better when the analysis needs to be repeated, refreshed, or shared with others dynamically.
Data Modeling	Supports data modeling with DAX and relationships	More advanced modeling, including DAX, M
Data Sources	Limited connectivity (Excel, SQL, Access, OData, etc.)	Extensive support (SQL, APIs, cloud sources, etc.)
User Interface	Excel-based, uses Power PivotTables & Power Query integration	Dedicated UI with visuals, slicers, dashboards
Visualization	Basic (PivotTables, PivotCharts)	Advanced (interactive dashboards, custom visuals) Creating interactive dashboards
Data Refresh	Manual or scheduled via Power Query and Excel features	Automated refresh via Power BI Service
Data Volume	Limited to Excel's memory constraints (~1M rows)	Can handle billions of rows with efficient compression
Sharing & Collaboration	Via Excel file sharing or OneDrive	Cloud-based sharing via Power BI Service
Security	Limited security controls (password, OneDrive settings)	Row-level security (RLS), role-based access
Advanced Features	Some support for DAX, Measures, KPIs	Advanced AI, predictive analytics, Python, R
Deployment	Local desktop use only	Can be deployed to cloud, Power BI Service, Power BI Report Server
Automation	Limited (Excel Macros, VBA, Power Automate)	Supports Power Automate, APIs, AI-driven insights
Updates & Support	Slow updates (dependent on Excel updates)	Frequent updates with new features monthly
Licensing	Included in Excel (with Power Pivot enabled)	Requires Power BI Free, Pro, or Premium



10 Best Practices for Data Modeling

1. **Use a Star Schema instead of a flat table**
   - A star schema consists of a central fact table surrounded by dimension tables, reducing redundancy and improving performance.
   - Avoid a snowflake schema unless necessary, as it increases complexity.
   - Power BI performs best with denormalized data structures optimized for reporting.

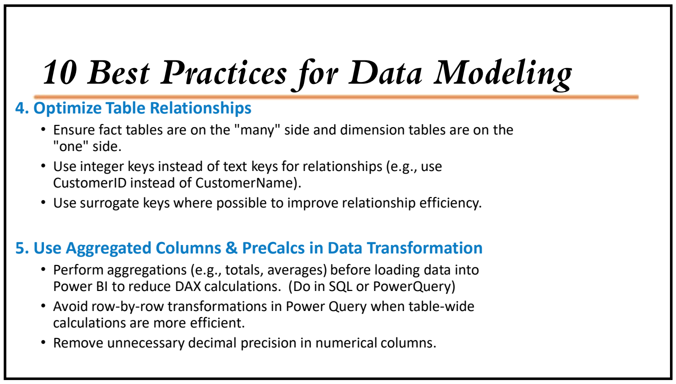
10 Best Practices for Data Modeling

2. Reduce Relationships & Cardinality Issues

- Use one-to-many relationships rather than many-to-many to prevent performance issues.
- Reduce high-cardinality columns in relationships (e.g., avoid using unique transaction IDs unless necessary).
- Avoid bidirectional filtering unless essential—it can introduce ambiguity and slow down calculations.

3. Keep the Model as Simple as Possible

- Remove unnecessary tables and columns that don't add value.
- Create a single source of truth by defining clear measures and relationships.
- Consolidate multiple fact tables if they serve the same purpose.



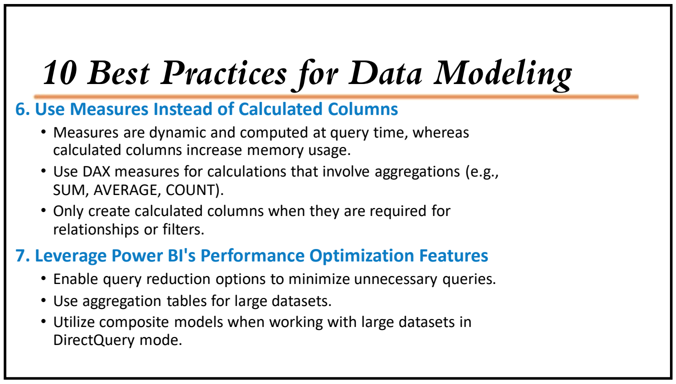
10 Best Practices for Data Modeling

4. Optimize Table Relationships

- Ensure fact tables are on the "many" side and dimension tables are on the "one" side.
- Use integer keys instead of text keys for relationships (e.g., use CustomerID instead of CustomerName).
- Use surrogate keys where possible to improve relationship efficiency.

5. Use Aggregated Columns & PreCalcs in Data Transformation

- Perform aggregations (e.g., totals, averages) before loading data into Power BI to reduce DAX calculations. (Do in SQL or PowerQuery)
- Avoid row-by-row transformations in Power Query when table-wide calculations are more efficient.
- Remove unnecessary decimal precision in numerical columns.



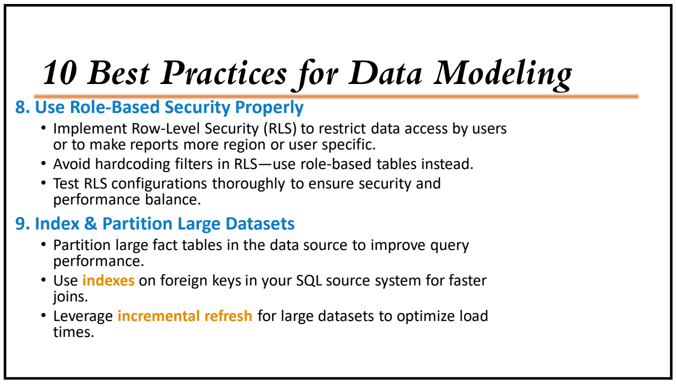
10 Best Practices for Data Modeling

6. Use Measures Instead of Calculated Columns

- Measures are dynamic and computed at query time, whereas calculated columns increase memory usage.
- Use DAX measures for calculations that involve aggregations (e.g., SUM, AVERAGE, COUNT).
- Only create calculated columns when they are required for relationships or filters.

7. Leverage Power BI's Performance Optimization Features

- Enable query reduction options to minimize unnecessary queries.
- Use aggregation tables for large datasets.
- Utilize composite models when working with large datasets in DirectQuery mode.



10 Best Practices for Data Modeling

8. Use Role-Based Security Properly

- Implement Row-Level Security (RLS) to restrict data access by users or to make reports more region or user specific.
- Avoid hardcoding filters in RLS—use role-based tables instead.
- Test RLS configurations thoroughly to ensure security and performance balance.

9. Index & Partition Large Datasets

- Partition large fact tables in the data source to improve query performance.
- Use **indexes** on foreign keys in your SQL source system for faster joins.
- Leverage **incremental refresh** for large datasets to optimize load times.

10 Best Practices for Data Modeling

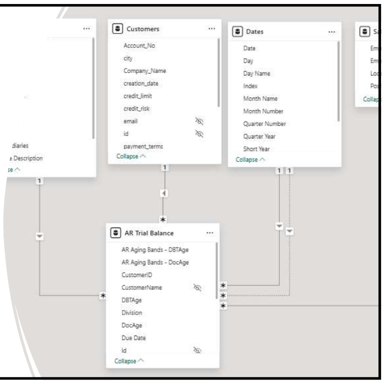
10. Document Your Data Model

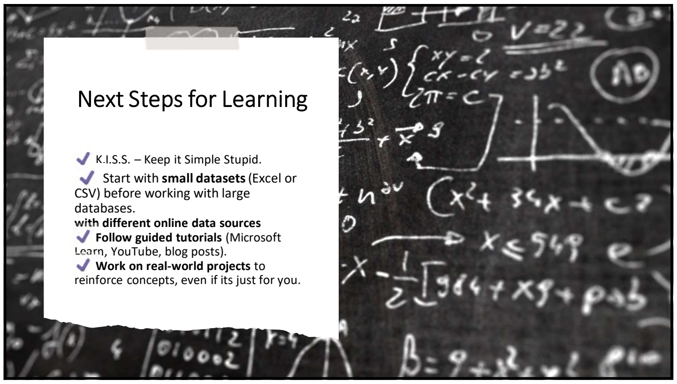
- Use clear naming conventions for tables, columns, and measures
- Create a **data dictionary** explaining key business logic.
- Add descriptions in Power BI for datasets, relationships, and measures.

Remember:

Good Data Models Provide:

- *Performance Optimization*
- *Data Accuracy & Consistency*
- *Better DAX Performance & Simplicity*
- *Scalability & Flexibility*
- *Improved Report Performance*
- *Security & Row-Level Security (RLS) Implementation*
- *Easier Maintenance & Collaboration*
- *Efficient Storage & Reduced Data Size*
- *Data Reusability*
- *Business Logic & Centralization*





Next Steps for Learning

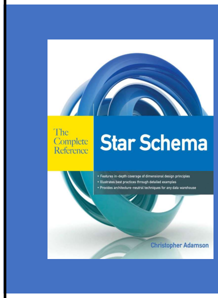
✓ K.I.S.S. – Keep it Simple Stupid.

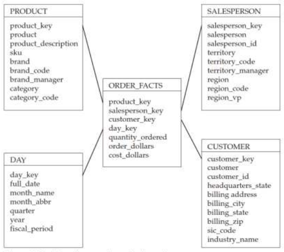
✓ Start with **small datasets** (Excel or CSV) before working with large databases.

with different online data sources

✓ **Follow guided tutorials** (Microsoft Learn, YouTube, blog posts).

✓ **Work on real-world projects** to reinforce concepts, even if its just for you.





PRODUCT
- product_key
- product
- product_description
- sku
- brand
- brand_code
- brand_manager
- category
- category_code

DAY
- day_key
- full_date
- month_name
- month_abbr
- quarter
- year
- fiscal_period

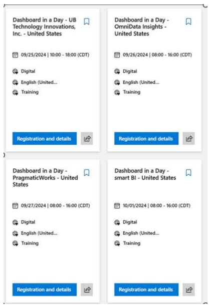
ORDER_FACTS
- product_key
- salesperson_key
- customer_key
- day_key
- quantity_ordered
- order_dollars
- cost_dollars

SALESPERSON
- salesperson_key
- salesperson
- salesperson_id
- territory
- territory_code
- territory_manager
- region
- region_code
- region_vp

CUSTOMER
- customer_key
- customer
- customer_id
- headquarters_state
- billing address
- billing_city
- billing_state
- billing_zip
- sic_code
- industry_name

Figure 1-5 A simple star schema for the orders process

Attend a Free 1 Day Event Workshop:
Microsoft Dashboard in a Day



Dashboard in a Day - UB Technology Innovations, Inc. - United States
09/25/2024 | 10:00 - 18:00 (CDT)
Digital
English (United...
Training
Registration and details

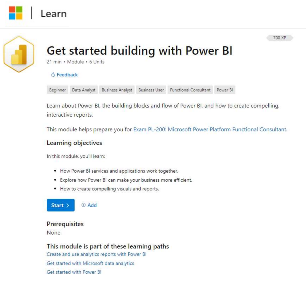
Dashboard in a Day - OmniData Insights - United States
09/26/2024 | 08:00 - 16:00 (CDT)
Digital
English (United...
Training
Registration and details

Dashboard in a Day - PragmaticWorks - United States
09/27/2024 | 08:00 - 16:00 (CDT)
Digital
English (United...
Training
Registration and details

Dashboard in a Day - smart BI - United States
10/01/2024 | 08:00 - 16:00 (CDT)
Digital
English (United...
Training
Registration and details

Pragmatic Works DAX Cheat Sheet for Beginners

- Hands-On, Practical Learning Experience
- Rapid Skill Acquisition
- Guided Instruction from Experts
- Structured Learning Agenda
- Real-World Application of Skills
- Access to Workshop Materials & Resources
- Networking Opportunities
- Personalized Feedback & Support
- Boosts Confidence with Power BI
- Preparation for Advanced Learning
- Cost-Effective Training Option
- Immediate Insight into Power BI's Capabilities
- Exposure to Power BI Service Features



Learn

700 XP

Get started building with Power BI

21 min • Module • 6 Units

Feedback

Beginner | Data Analyst | Business Analyst | Business User | Functional Consultant | Power BI

Learn about Power BI, the building blocks and flow of Power BI, and how to create compelling, interactive reports.

This module helps prepare you for Exam PL-200: Microsoft Power Platform Functional Consultant.

Learning objectives

In this module, you'll learn:

- How Power BI services and applications work together.
- Explore how Power BI can make your business more efficient.
- How to create compelling visuals and reports.

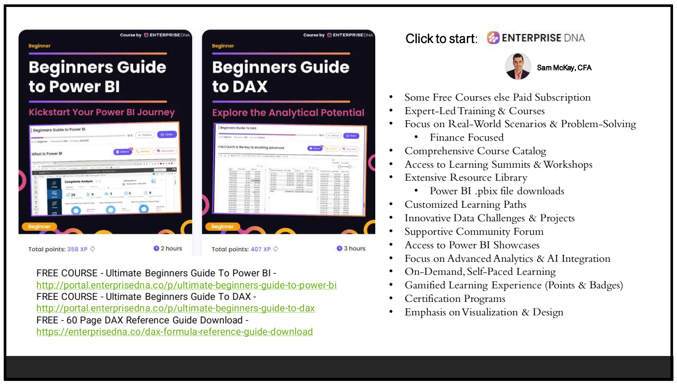
Start > ⊕ Add

Prerequisites
None

This module is part of these learning paths
Create and use analytics reports with Power BI
Get started with Microsoft data analytics
Get started with Power BI

Click to start: Microsoft Learn

- Free Access to High-Quality Content
- Structured Learning Paths
- Hands-on Labs and Interactive Exercises
- Official and Up-to-Date Content
- Integration with Certifications
- Gamified Learning Experiences (Points, Badges)
- Self-Paced Learning
- Community and Q&A Integration
- Comprehensive Coverage of Power BI Features
- Scenario-Based Learning Modules

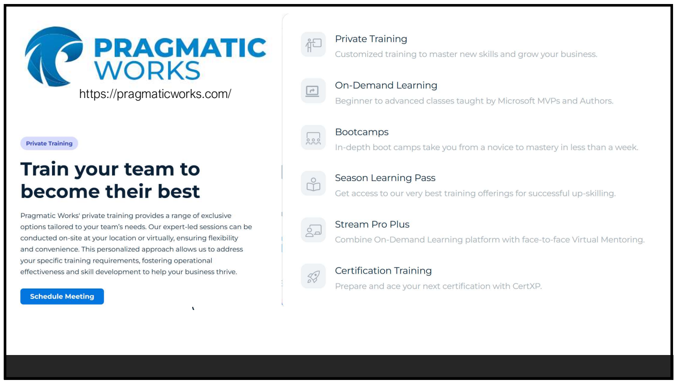


Click to start: ENTERPRISE DNA

Sam McKay, CFA

- Some Free Courses else Paid Subscription
- Expert-Led Training & Courses
- Focus on Real-World Scenarios & Problem-Solving
    - Finance Focused
- Comprehensive Course Catalog
- Access to Learning Summits & Workshops
- Extensive Resource Library
    - Power BI .pbix file downloads
- Customized Learning Paths
- Innovative Data Challenges & Projects
- Supportive Community Forum
- Access to Power BI Showcases
- Focus on Advanced Analytics & AI Integration
- On-Demand, Self-Paced Learning
- Gamified Learning Experience (Points & Badges)
- Certification Programs
- Emphasis on Visualization & Design

FREE COURSE - Ultimate Beginners Guide To Power BI -
http://portal.enterprisedna.co/p/ultimate-beginners-guide-to-power-bi
FREE COURSE - Ultimate Beginners Guide To DAX -
http://portal.enterprisedna.co/p/ultimate-beginners-guide-to-dax
FREE - 60 Page DAX Reference Guide Download -
https://enterprisedna.co/dax-formula-reference-guide-download



PRAGMATIC WORKS
https://pragmaticworks.com/

Private Training

Train your team to become their best

Pragmatic Works' private training provides a range of exclusive options tailored to your team's needs. Our expert-led sessions can be conducted on-site at your location or virtually, ensuring flexibility and convenience. This personalized approach allows us to address your specific training requirements, fostering operational effectiveness and skill development to help your business thrive.

Schedule Meeting

Private Training
Customized training to master new skills and grow your business.

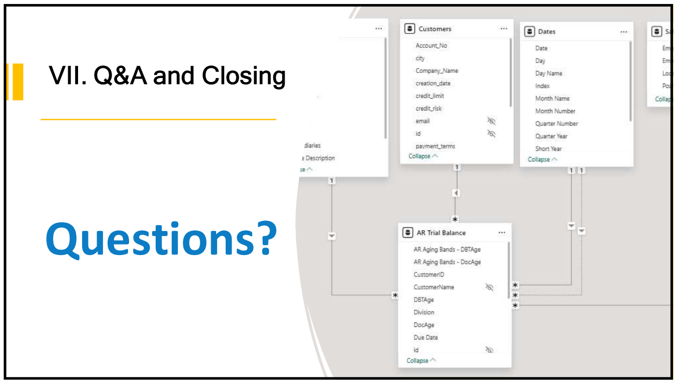
On-Demand Learning
Beginner to advanced classes taught by Microsoft MVPs and Authors.

Bootcamps
In-depth boot camps take you from a novice to mastery in less than a week.

Season Learning Pass
Get access to our very best training offerings for successful up-skilling.

Stream Pro Plus
Combine On-Demand Learning platform with face-to-face Virtual Mentoring.

Certification Training
Prepare and ace your next certification with CertXP.



VII. Q&A and Closing

Questions?