

# Mastering the Query Features of Power BI Desktop



# Agenda

- Deciding What To Measure
- Managing Queries, Datasources and Credentials
- Working with the Query Editor Window
- Designing Queries to Generate a Star Schema



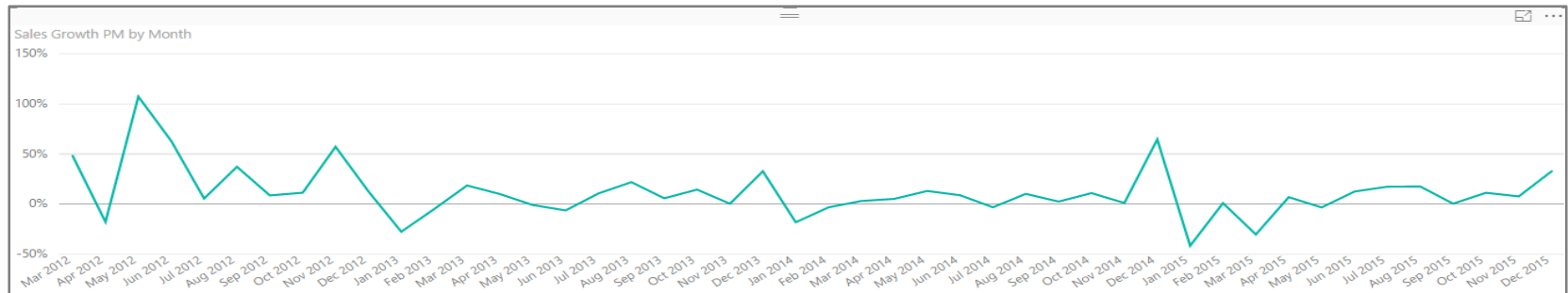
# Data Discovery

- Data can live in a variety of sources
  - Files (e.g. CSV file, Excel workbook)
  - OLTP Databases
  - OLAP Databases
  - SharePoint Lists and Document Libraries
  - Azure-based services
  - Online services & SaaS applications



# Deciding What To Measure

- You Must Determine Measurable Objectives
  - Financial (revenue, expenses, profit margin, etc.)
  - Business processes efficiency
  - Customer Satisfaction Levels



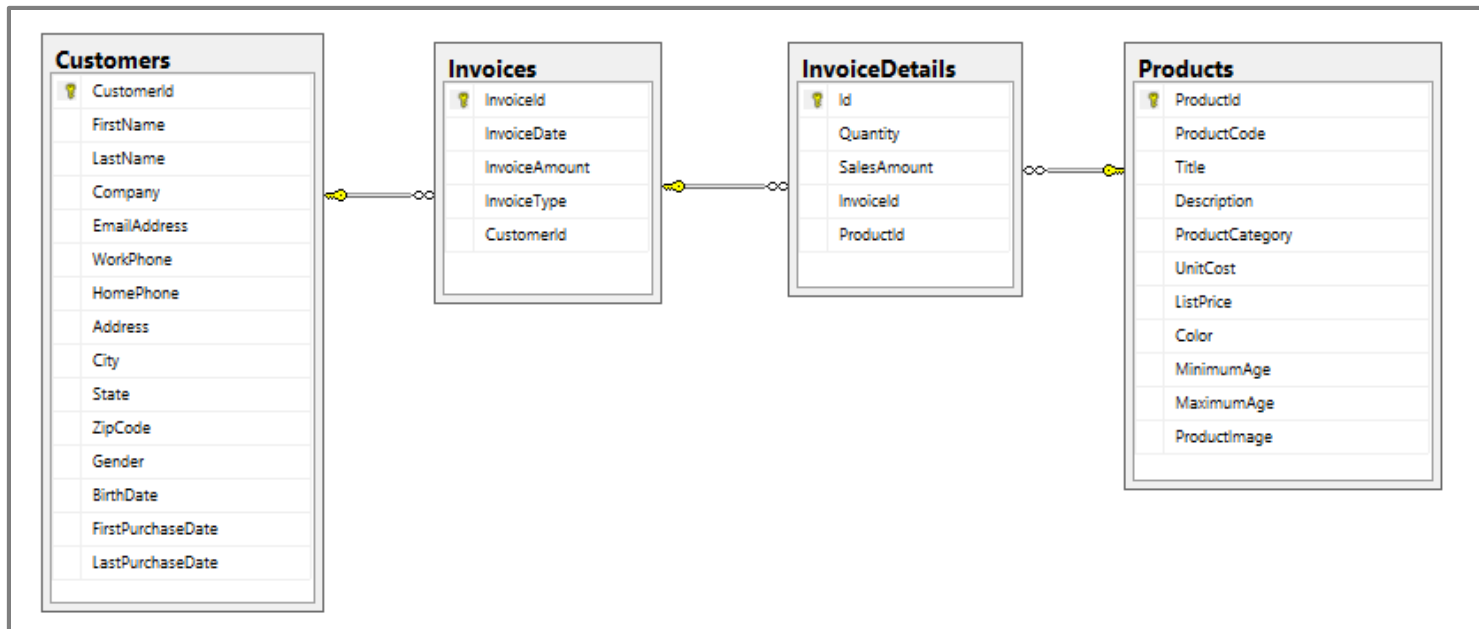
# Defining Grain Statements

- Grain statements should be defined in initial design phase
  - Grain statements helps determine requirements for BI queries
  - Grain statements can be created & understood by business users
- Example grain statements for BI project at Wingtip Toys
  - What was the total sales revenue over the last 4 years?
  - What was the sales revenue by year, quarter and month?
  - What was the sales revenue by region, state, city and zip code?
  - What was the sales revenue by category, subcategory and product?
  - What was the growth in sales revenue from month to month in 2013?
  - What was profit margin for each product by year, quarter and month?
  - Have their been any products with significantly decreasing profit margin?



# Sample OLTP Database: WingtipSalesDB

- Online Transaction Processing (OLTP) System
  - Used for real-time data access and transaction-based data entry
  - Optimized for faster transactions (e.g. inserts, updates & deletes)
  - Tables normalized to reduce/eliminate redundancies
  - Table schemas can be hard for business users to understand



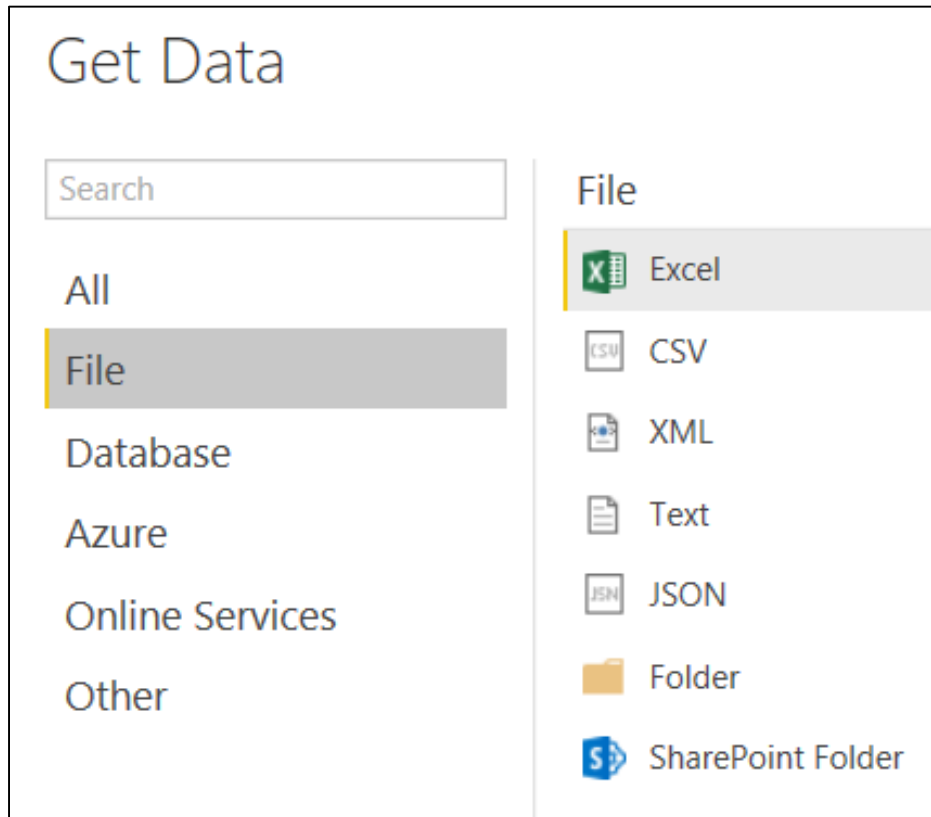
# Agenda

- ✓ Deciding What To Measure
- Managing Queries, Datasources and Credentials
  - Working with the Query Editor Window
  - Designing Queries to Generate a Star Schema



# File-based Data Sources

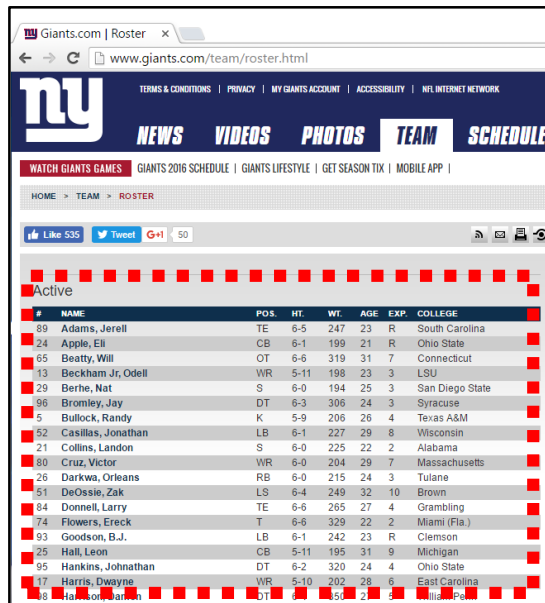
- Power BI Desktop supports common file types





# Working with Web Data Sources

- Many public websites publish data using HTML tables
- Power BI desktop can scrape data from tables in HTML pages



Active

#	NAME	POS.	HT.	WT.	AGE	EXP.	COLLEGE
89	Adams, Jerrell	TE	6-5	247	23	R	South Carolina
24	Apple, Eli	CB	6-1	199	21	R	Ohio State
65	Beatty, Will	OT	6-6	319	31	7	Connecticut
13	Beckham Jr, Odell	WR	5-11	198	23	3	LSU
29	Berhe, Nat	S	6-0	194	25	3	San Diego State
96	Bromley, Jay	DT	6-3	306	24	3	Syracuse
5	Bullock, Randy	K	5-9	206	26	4	Texas A&M
52	Casillas, Jonathan	LB	6-1	227	29	8	Wisconsin
21	Collins, Landon	S	6-0	225	22	2	Alabama
80	Cruz, Victor	WR	6-0	204	29	7	Massachusetts
26	Dar kwa, Orleans	RB	6-0	215	24	3	Tulane
51	DeOssie, Zak	LS	6-4	249	32	10	Brown
84	Donnell, Larry	TE	6-6	265	27	4	Grambling
74	Flowers, Ereck	T	6-6	329	22	2	Miami (Fla.)
93	Goodson, B.J.	LB	6-1	242	23	R	Clemson
25	Hall, Leon	CB	5-11	195	31	9	Michigan
55	Hankins, Johnathan	DT	6-2	320	24	4	Ohio State
17	Harris, Devyne	WR	5-10	202	28	6	East Carolina
16	Hunter, Jarron	DT	6-4	255	22	5	Illinois

From Web

Basic Advanced

Enter a Web page URL.

URL

Open file as

OK Cancel

Query Input

	#	Name	Pos.	Ht.	Wt.	Age	Exp.	College
1	89	Adams, Jerrell	TE	6-5	247	23	R	South Carolina
2	24	Apple, Eli	CB	6-1	199	21	R	Ohio State
3	65	Beatty, Will	OT	6-6	319	31	7	Connecticut
4	13	Beckham Jr, Odell	WR	5-11	198	23	3	LSU
5	29	Berhe, Nat	S	6-0	194	25	3	San Diego State
6	96	Bromley, Jay	DT	6-3	306	24	3	Syracuse

Query Output

	Number	Last Name	First Name	Weight	Height	Age	Experience	Position	Category	Side	College
1	89	Adams	Jerrell	247	77	23	0	Tight End	Backs and Receivers	Offense	South Carolina
2	84	Donnell	Larry	265	78	27	4	Tight End	Backs and Receivers	Offense	Grambling
3	45	Tye	Will	262	74	24	1	Tight End	Backs and Receivers	Offense	Stony Brook
4	24	Apple	Eli	199	73	21	0	Cornerback	Defensive Backs	Defense	Ohio State
5	25	Hall	Leon	195	71	31	9	Cornerback	Defensive Backs	Defense	Michigan
6	20	Jenkins	Janoris	198	70	27	5	Cornerback	Defensive Backs	Defense	North Alabama
7	41	Rodgers-Cromartie	Dominique	205	74	30	8	Cornerback	Defensive Backs	Defense	Tennessee State
8	65	Beatty	Will	319	78	31	7	Offensive Tackle	Offensive Line	Offense	Connecticut
9	13	Beckham Jr	Odell	198	71	23	3	Wide Receiver	Backs and Receivers	Offense	LSU
10	80	Cruz	Victor	204	72	29	7	Wide Receiver	Backs and Receivers	Offense	Massachusetts

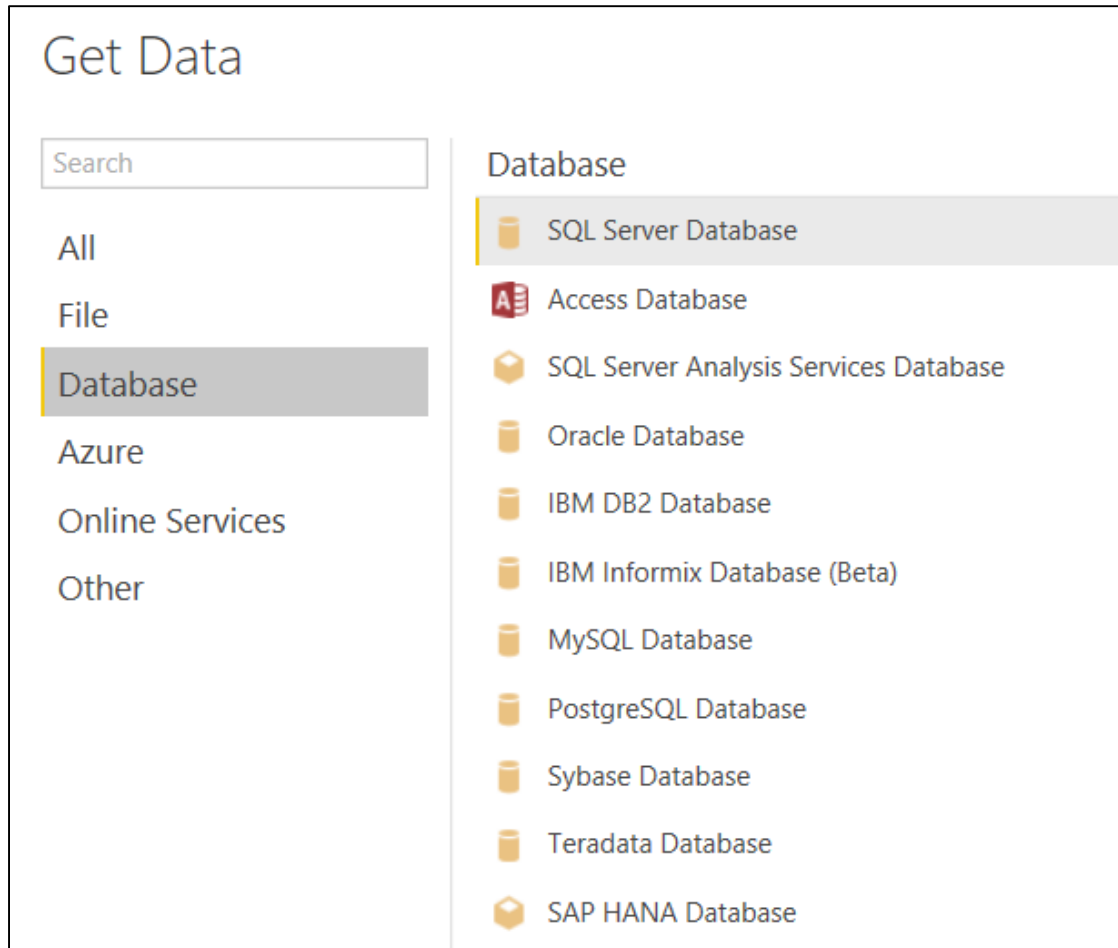


**DEMO**

# Scraping Data from a Web Page

# Supported Databases

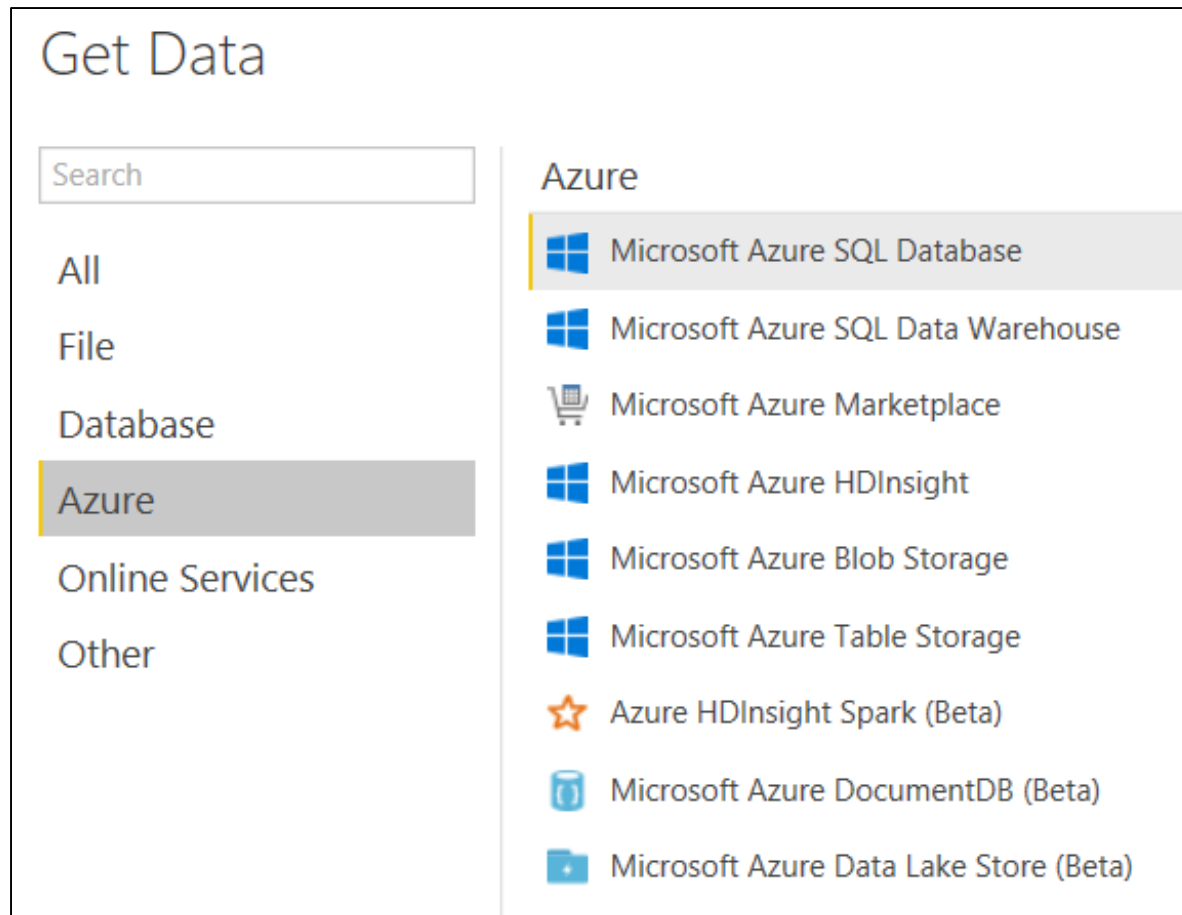
- Power BI Desktop supports many database systems





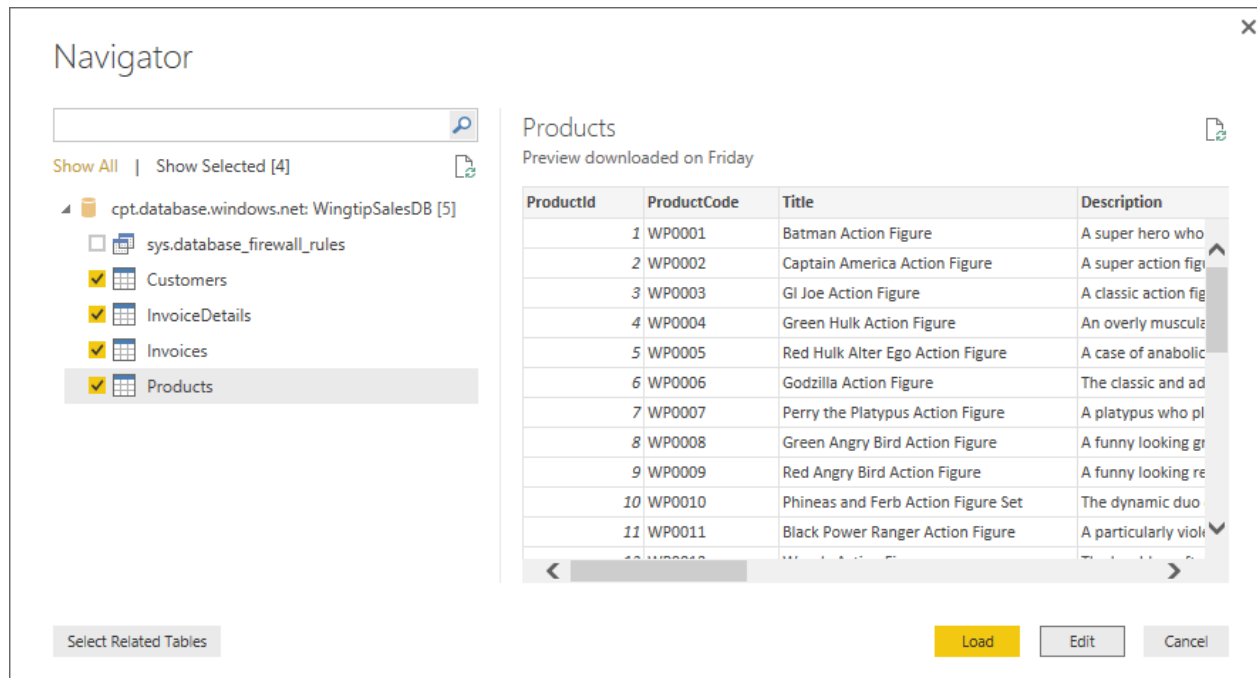
# Azure Data Sources

- Power BI Desktop supports many Azure data sources



# Selecting Tables from a SQL Database

- Power BI Desktop provides Navigator dialog
  - Allows you to select tables
  - Navigator understands existing table relationships
  - Clicking **Load** will run query and import data
  - Clicking **Edit** will open queries in Query Editor window

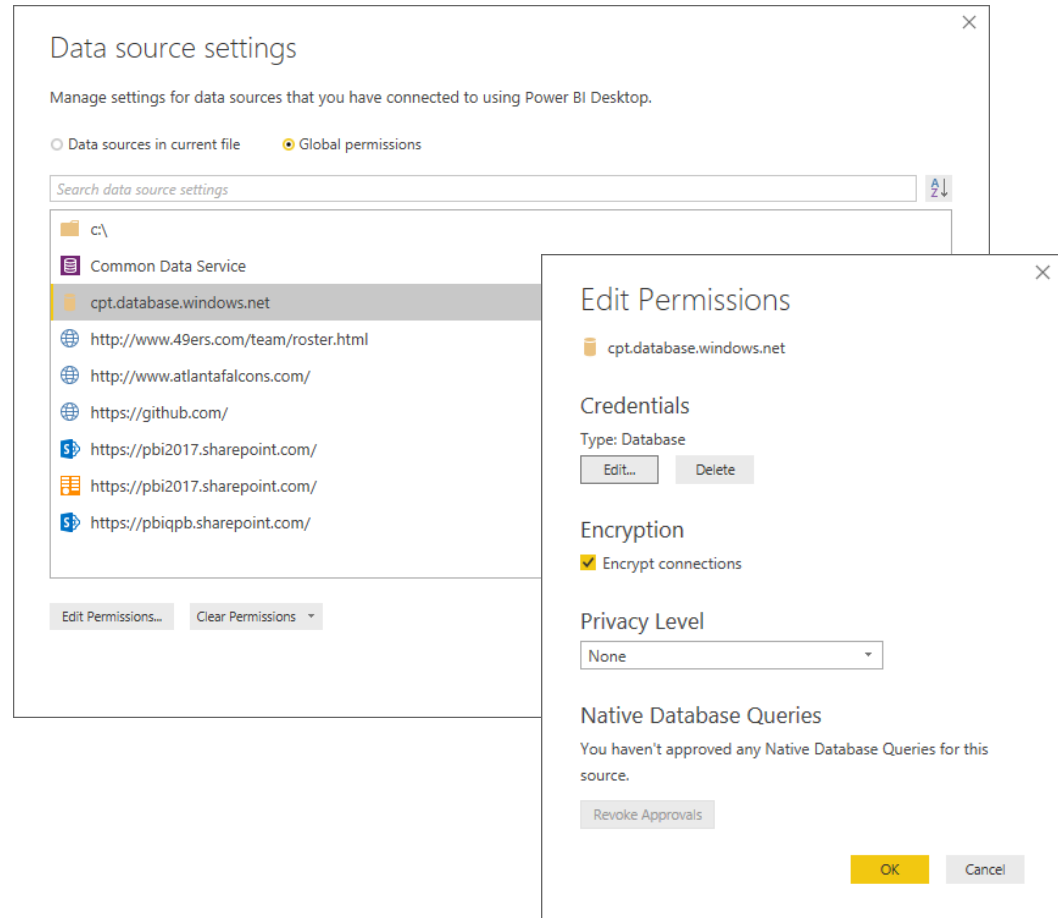
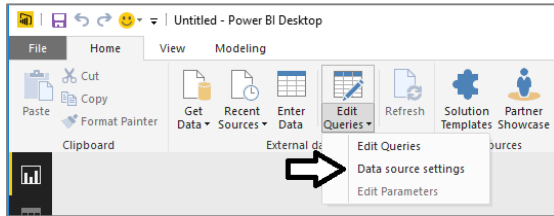


The background of the slide is a close-up, low-angle shot of a server rack. The rack is filled with numerous server units, each featuring a grid of small, glowing blue lights. The perspective is looking up the length of the rack, creating a sense of depth and scale. The lighting is predominantly blue, giving it a high-tech, digital feel.

**DEMO**

## **Creating Queries to Import Data from an Azure SQL Database**

# Managing Datasources and Credentials





# Agenda

- ✓ Deciding What To Measure
- ✓ Managing Queries, Datasources and Credentials
- Working with the Query Editor Window
- Designing Queries to Generate a Star Schema





# Query Steps

- A query is created as a sequence of steps
  - Each step is a parameterized operation on the data
  - Each step has formula which can be viewed/edited in formula bar
  - Query starts with Source step to extract data from a data source
  - Additional steps added to perform transform operations on data
  - You can replay query operations one by one by clicking on steps

The screenshot displays the Power BI Query Editor interface. At the top, the ribbon includes 'File', 'Home', 'Transform', 'Add Column', and 'View'. Below the ribbon, the 'Formula Bar' is active, showing the formula: `= Table.ReplaceValue("#Replaced Female Values","M","Male",Replacer.ReplaceText,`. A red dashed box highlights the formula bar, with a yellow callout box labeled 'step formula bar' pointing to it. On the left, the 'Queries [6]' pane lists 'Customers', 'Sales', 'Purchases', 'Products', 'SalesRegions', and 'SalesRegionsSort'. The main area shows a data table with columns: CustomerId, Customer, State, City, Zipcode, and Gender. The table contains 14 rows of data. On the right, the 'Query Settings' pane is open, showing the 'Properties' section with 'Name' set to 'Customers'. Below it, the 'Applied Steps' section is highlighted with a red dashed box and a yellow callout box labeled 'sequential list of steps for query'. The 'Applied Steps' list includes: Source, Navigation, Removed Other Columns, Merged Columns, Reordered Columns, Replaced Female Values, Replaced Male Values (which is selected with a mouse cursor), Changed Type, and Added Conditional Column.

CustomerId	Customer	State	City	Zipcode	Gender
1	Nina Diaz	CA	Eureka	95501	Female
2	Melinda Carter	CA	Napa	94558	Female
3	Pam Miller	CA	Napa	94558	Female
4	Merle Blackwell	CA	Sacramento	95823	Female
5	Ariel Hale	CA	Sacramento	95818	Male
6	Randy Carter	CA	Sacramento	95818	Male
7	Lillie Hinton	CA	Eureka	95501	Female
8	Ladonna Moody	CA	Napa	94559	Female
9	Buddy McKay	OR	Bend	97701	Male
10	Warren Sykes	CA	Sacramento	95818	Male
11	Jan Rutledge	OR	Portland	97216	Female
12	Dallas Lester	OR	Eugene	97402	Male
13	Matthew Zimmerman	OR	Portland	97220	Male
14	Sheryl Hernandez	CA	Sacramento	95823	Female

# Query Editor Ribbon Tabs

## Home tab



## Transform tab



## Add Column tab



## View tab



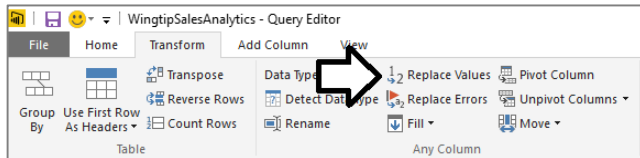
# Examples of Basic Power BI Desktop Steps

- Rename column
- Convert column type
- Format column values
- Reorder columns
- Replace column values
- Expanding related column
- Merging columns
- Splitting columns



# Replacing Values

- Used to substitute values during import



Replace Values

Replace one value with another in the selected columns.

Value To Find  
M

Replace With  
Male

Advanced options

OK Cancel

Query Settings

PROPERTIES

Name  
Customers

All Properties

APPLIED STEPS

Source

Navigation

Removed Other Columns

Merged Columns

Reordered Columns

Replaced Value

Replaced Value1

ZipCode	Gender	BirthDate	FirstPurchase
90266	Male	2/24/1966 12:00:00 AM	1/28
78753	Male	2/3/1941 12:00:00 AM	1/28
80924	Male	10/20/1954 12:00:00 AM	1/28
79121	Female	11/4/1974 12:00:00 AM	1/28
79407	Male	8/14/1954 12:00:00 AM	1/28
85281	Male	6/11/1971 12:00:00 AM	1/28
79936	Male	4/25/1974 12:00:00 AM	1/28
78730	Male	10/2/1944 12:00:00 AM	1/28
78253	Female	10/26/1956 12:00:00 AM	1/28
90266	Male	10/2/1955 12:00:00 AM	1/28
85044	Male	1/11/1945 12:00:00 AM	1/28
95630	Female	2/1/1956 12:00:00 AM	1/28
98052	Male	5/13/1990 12:00:00 AM	1/28
97701	Female	5/14/1958 12:00:00 AM	1/28

Replace Values

Replace one value with another in the selected columns.

Value To Find  
F

Replace With  
Female











Advanced options

OK Cancel



# Converting Column Types

- Transform data to make it more reliable
  - Convert date-time column to date column
- Transform data to make it more efficient
  - Convert decimal to fixed decimal number for currency

 PurchaseDate		 Quantity		 SalesAmount		 ProductCost	
1/28/2012		1		2.95		1.2	Decimal Number
1/28/2012		6				\$	Fixed Decimal Number
1/28/2012		1		19.95		1 <sup>2</sup> / <sub>3</sub>	Whole Number
1/28/2012		5		249.75			Date/Time
1/28/2012		1		2.95			Date

- Beware: Conversion can have destructive effect on data



# Expanding Related Columns

- Used to pull data from related tables
  - Saves you from performing SQL joins or VLOOKUP

SalesAmount	Invoices	
119.8	Value	Value
29.95	Value	Value
59.9	Value	Value
399.6	Value	Value
29.9	Value	Value
59.8	Value	Value

Id	InvoiceId	ProductId	Quantity	SalesAmount	Invoices	Products
1	1	1				Value
2	2	1				Value
3	3	2				Value
4	4	3				Value
5	5	3				Value
6	6	3				Value
7	7	4				Value
8	8	5				Value
9	9	6				Value
10	10	6				Value
11	11	7				Value
12	12	7				Value
13	13	8				Value
14	14	9				Value

Search Columns to Expand

(Select All Columns)

☐ InvoiceId

☒ InvoiceDate

☐ InvoiceAmount

☐ InvoiceType

☒ CustomerId

☐ Customers

☐ InvoiceDetails

☐ Use original column name as prefix

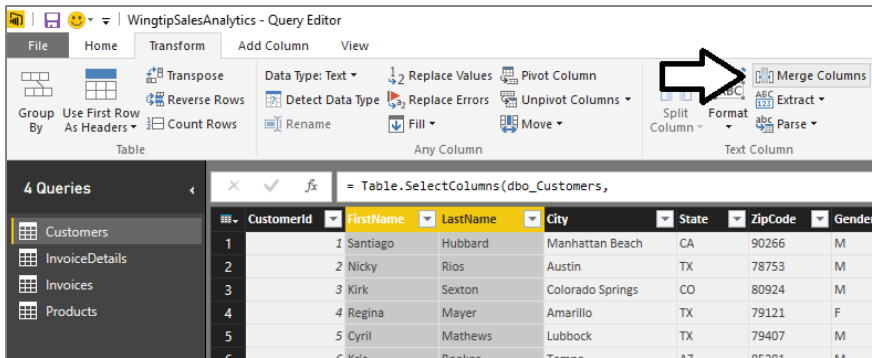
OK Cancel

Id	InvoiceId	ProductId	Quantity	SalesAmount	InvoiceDate	CustomerId	Products
1	1	1	22	4	119.8	1/28/2012 12:00:00 AM	1 Value
2	2	1	22	1	29.95	1/28/2012 12:00:00 AM	1 Value
3	3	2	22	2	59.9	1/28/2012 12:00:00 AM	2 Value
4	4	3	17	8	399.6	1/28/2012 12:00:00 AM	3 Value
5	5	3	18	2	29.9	1/28/2012 12:00:00 AM	3 Value
6	6	3	18	4	59.8	1/28/2012 12:00:00 AM	3 Value
7	7	4	16	1	2.95	1/28/2012 12:00:00 AM	4 Value



# Merging Columns

- Merge two columns into a single column



WingtipSalesAnalytics - Query Editor

File Home Transform Add Column View

Group By Use First Row As Headers Count Rows

Transpose Reverse Rows

Data Type: Text Replace Values Replace Errors

Pivot Column Unpivot Columns

Rename Fill Move

Split Column Format

Extract Parse

Merge Columns

Table

Any Column

Text Column

4 Queries

Customers InvoiceDetails Invoices Products

Customerid FirstName LastName City State ZipCode Gender

1 1 Santiago Hubbard Manhattan Beach CA 90266 M

2 2 Nicky Rios Austin TX 78753 M

3 3 Kirk Sexton Colorado Springs CO 80924 M

4 4 Regina Mayer Amarillo TX 79121 F

5 5 Cyril Mathews Lubbock TX 79407 M

6 6 Kris Booker Tempe AZ 85281 M

Merge Columns

Choose how to merge the selected columns.

Separator

Space

New column name (optional)

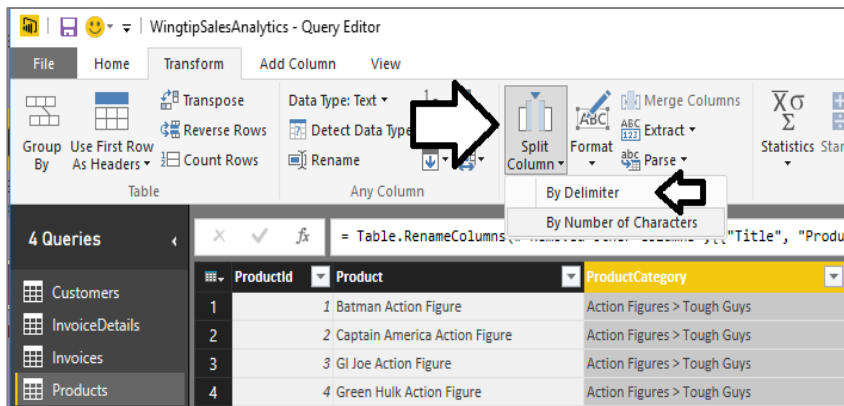
Customer

OK Cancel

	Customerid	Customer
1	1	Santiago Hubbard
2	2	Nicky Rios
3	3	Kirk Sexton
4	4	Regina Mayer
5	5	Cyril Mathews
6	6	Kris Booker
7	7	Tracy Christensen
8	8	Reed Glover

# Splitting Columns

- Split a single column up into two columns



Split Column by Delimiter

Specify the delimiter to split the text column.

Select or enter delimiter

--Custom--

>

Split

☐ At the left-most delimiter

☐ At the right-most delimiter

☒ At each occurrence of the delimiter

Advanced options

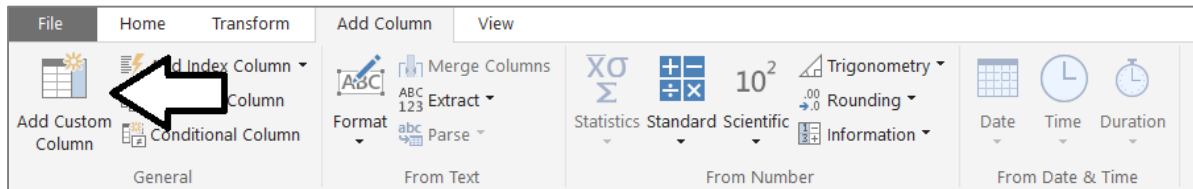
OK Cancel

ProductId	Product	Category	Subcategory
1	1 Batman Action Figure	Action Figures	Tough Guys
2	2 Captain America Action Figure	Action Figures	Tough Guys
3	3 GI Joe Action Figure	Action Figures	Tough Guys
4	4 Green Hulk Action Figure	Action Figures	Tough Guys
5	5 Red Hulk Alter Ego Action Figure	Action Figures	Tough Guys
6	6 Godzilla Action Figure	Action Figures	Tough Guys
7	7 Perry the Platypus Action Figure	Action Figures	Cute and Huggable
8	8 Green Angry Bird Action Figure	Action Figures	Cute and Huggable



# Adding a Custom Column

- Custom column provide custom logic
  - Logic must be written in M programming language



**Add Custom Column**

New column name:

Custom column formula:  

```
= if [FirstPurchaseDate]=[LastPurchaseDate]  
then "One-time Customer"  
else "Repeat Customer"
```

Available columns:  
CustomerId  
Customer  
State  
City  
ZipCode  
Gender  
BirthDate  
FirstPurchaseDate  
LastPurchaseDate  
CustomerType

<< Insert

[Learn about Power BI Desktop formulas](#)

✓ No syntax errors have been detected.

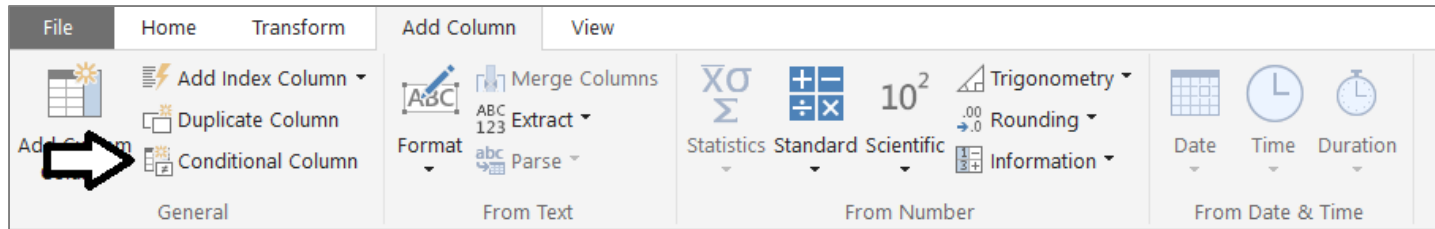
OK Cancel

FirstPurchaseDate	LastPurchaseDate	CustomerType
1/28/2012	1/28/2012	One-time Customer
1/28/2012	1/28/2012	One-time Customer
1/28/2012	1/28/2012	One-time Customer
1/28/2012	1/28/2012	One-time Customer
1/28/2012	1/28/2012	One-time Customer
1/28/2012	1/28/2012	One-time Customer
1/29/2012	11/22/2015	Repeat Customer
1/29/2012	10/2/2015	Repeat Customer
1/29/2012	1/29/2012	One-time Customer
1/29/2012	5/6/2015	Repeat Customer
1/29/2012	1/29/2012	One-time Customer



# Adding a Conditional Column

- Abstracts away need to write M code



**Add Conditional Column** [X]

Add a conditional column that is computed from the other columns or values.

New column name  
Customer Type

	Column Name	Operator	Value		Output
If	FirstPurchaseDate	equals	LastPurchaseDate	Then	One-time Customer

+ Add Rule

Otherwise

Repeat Customer
-----------------

OK Cancel



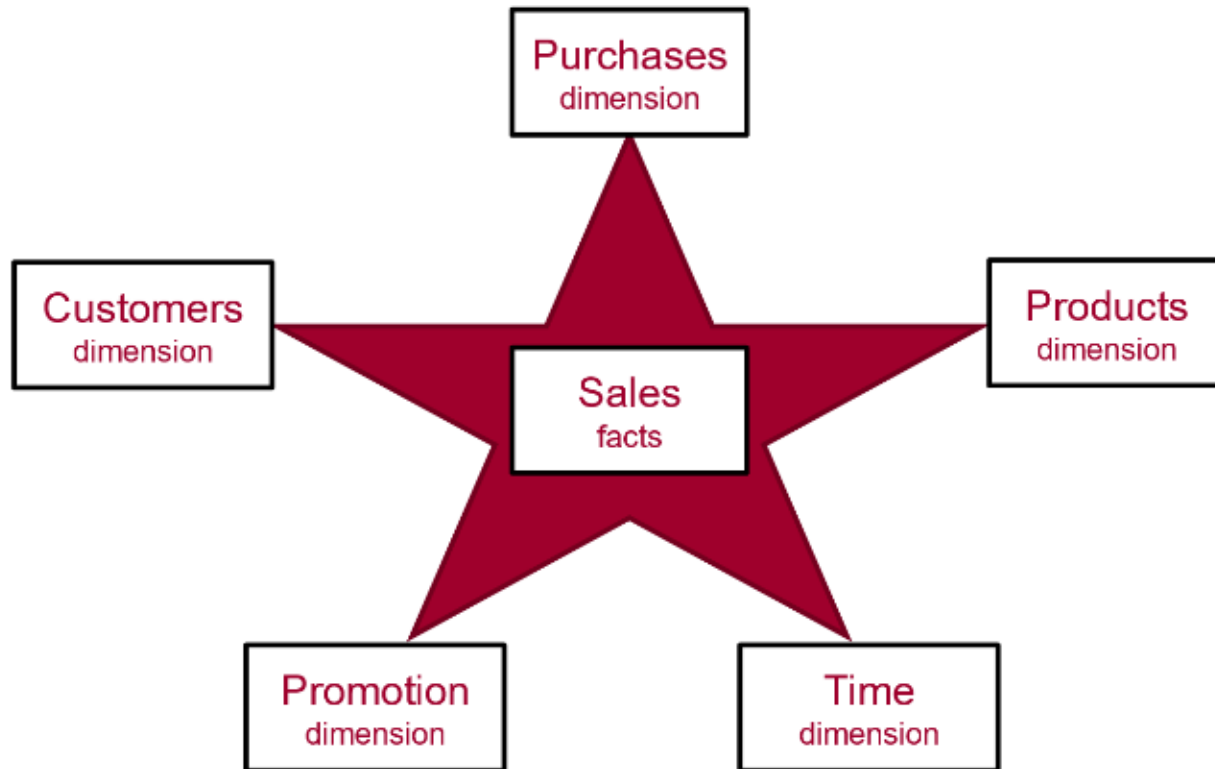
# Agenda

- ✓ Deciding What To Measure
- ✓ Managing Queries, Datasources and Credentials
- ✓ Working with the Query Editor Window
- Designing Queries to Generate a Star Schema



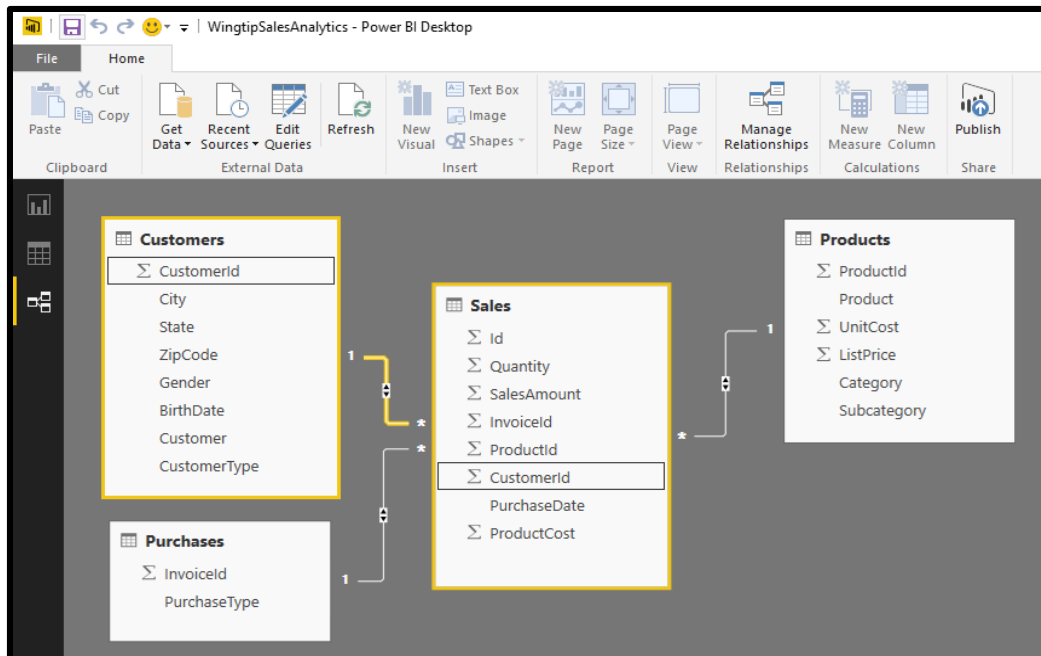
# Data Modeling using a Star Schema

- OLAP Modeling often based on Star Schema
  - Tables defined as fact tables or dimension tables
  - Fact tables related to dimension table using 1-to-many relationships



# Designing Queries to Build a Star Schema

- Converts OLTP Data Model to OLAP Data Model
  - Sales table is modeled as a OLAP Fact Table
  - Other tables are modeled as OLAP Dimension tables
  - Requires pulling CustomerId column into Sales table
  - All dimension tables should be directly related to fact table







**DEMO**

# Using Power BI Desktop to Import Data into a Star Schema

# Summary

- ✓ Deciding What To Measure
- ✓ Working with the Query Editor Window
- ✓ Managing Queries, Datasources and Credentials
- ✓ Designing Queries to Generate a Star Schema

