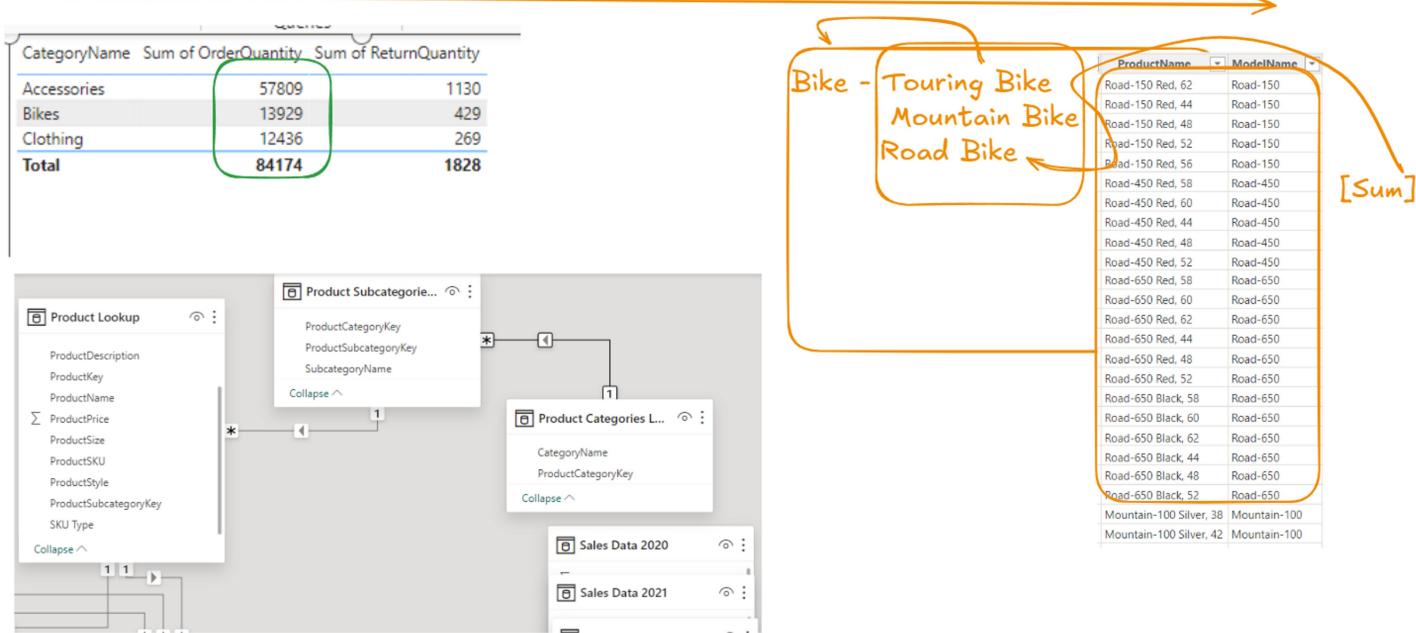
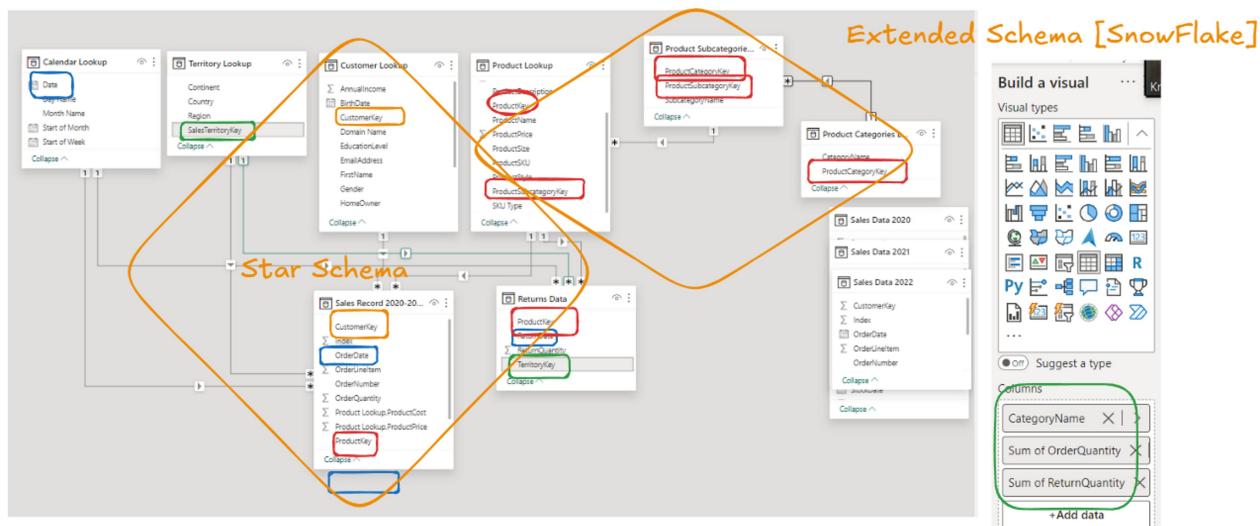


Power BI - Data Modelling - p2



Star Schema VS Snowflake Schema → Which is connected with extended schema.

one Fact Table Connected multiple Dimension Table [Directly]



Queries		
CategoryName	Sum of OrderQuantity	Sum of ReturnQuantity
Accessories	57809	1130
Bikes	13929	429
Clothing	12436	269
Total	84174	1828

Extended Dimension Table [Sub-Dimension Table]
They do share the foreign key.

SubcategoryName	OrderQuantity	ReturnQuantity
Bike Racks	302	8
Bike Stands	234	8
Bottles and Cages	15106	288
Caps	4151	46
Cleaners	1706	25
Fenders	3960	54
Gloves	2644	49
Helmets	6034	188
Hydration Packs	695	25
Jerseys	3113	93
Mountain Bikes	4706	136
Road Bikes	7099	223
Shorts	944	40
Socks	1063	22
Tires and Tubes	29772	534
Touring Bikes	2124	70
Vests	521	19
Total	84174	1828

Use of Extended Schema to get the final result in Report View

Manage relationships

Active	From: Table (Column)	To: Table (Column)
<input checked="" type="checkbox"/>	Product Lookup (ProductSubcategoryKey)	Product Subcategories Lookup (ProductSubcategoryKey)
<input checked="" type="checkbox"/>	Product Subcategories Lookup (ProductCategoryKey)	Product Categories Lookup (ProductCategoryKey)
<input checked="" type="checkbox"/>	Returns Data (ProductKey)	Product Lookup (ProductKey)
<input checked="" type="checkbox"/>	Returns Data (ReturnDate)	Calendar Lookup (Date)
<input checked="" type="checkbox"/>	Returns Data (TerritoryKey)	Territory Lookup (SalesTerritoryKey)
<input checked="" type="checkbox"/>	Sales Record 2020-2022 (CustomerKey)	Customer Lookup (CustomerKey)
<input checked="" type="checkbox"/>	Sales Record 2020-2022 (OrderDate)	Calendar Lookup (Date)
<input checked="" type="checkbox"/>	Sales Record 2020-2022 (ProductKey)	Product Lookup (ProductKey)
<input checked="" type="checkbox"/>	Sales Record 2020-2022 (TerritoryKey)	Territory Lookup (SalesTerritoryKey)

RELATIONSHIPS VS. MERGED TABLES



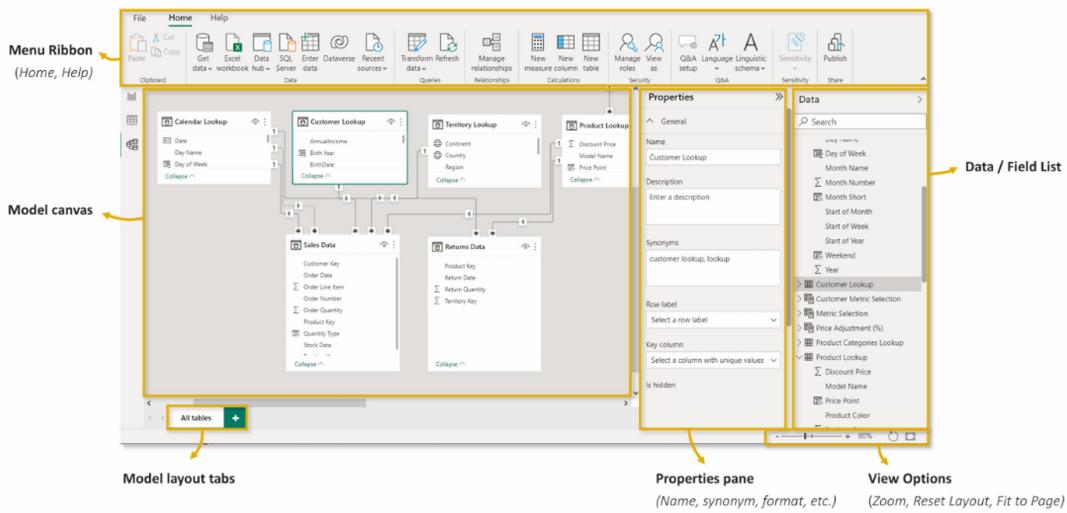
Can't I just merge queries or use lookup functions to pull everything into one single table?

Original Fact Table fields			Attributes from Calendar Lookup table						Attributes from Product Lookup table				
date	product_id	quantity	day_of_month	month	year	weekday	month_name	quarter	product_brand	product_name	product_sku	product_weight	
1/1/1997	869	5	1	1	1997	Wednesday	January	Q1	Nationel	Nationel Grape Fruit Roll	52382137179	17	
1/1/1997	869	2	7	1	1997	Tuesday	January	Q1	Nationel	Nationel Grape Fruit Roll	52382137179	17	
1/3/1997	1	4	3	1	1997	Friday	January	Q1	Washington	Washington Berry Juice	90748583674	8.39	
1/1/1997	1472	3	1	1	1997	Wednesday	January	Q1	Fort West	Fort West Fudge Cookies	37276054024	8.28	
1/6/1997	1472	2	6	1	1997	Monday	January	Q1	Fort West	Fort West Fudge Cookies	37276054024	8.28	
1/5/1997	2	4	5	1	1997	Sunday	January	Q1	Washington	Washington Mango Drink	96516502499	7.42	
1/1/1997	76	4	1	1	1997	Wednesday	January	Q1	Red Spade	Red Spade Sliced Chicken	62054644227	18.1	
1/1/1997	76	2	1	1	1997	Wednesday	January	Q1	Red Spade	Red Spade Sliced Chicken	62054644227	18.1	
1/5/1997	3	2	5	1	1997	Sunday	January	Q1	Washington	Washington Strawberry Drink	58427771925	13.1	
1/7/1997	3	2	7	1	1997	Tuesday	January	Q1	Washington	Washington Strawberry Drink	58427771925	13.1	
1/1/1997	320	3	1	1	1997	Wednesday	January	Q1	Excellent	Excellent Cranberry Juice	36570182442	16.4	

You can, but it's extremely inefficient!

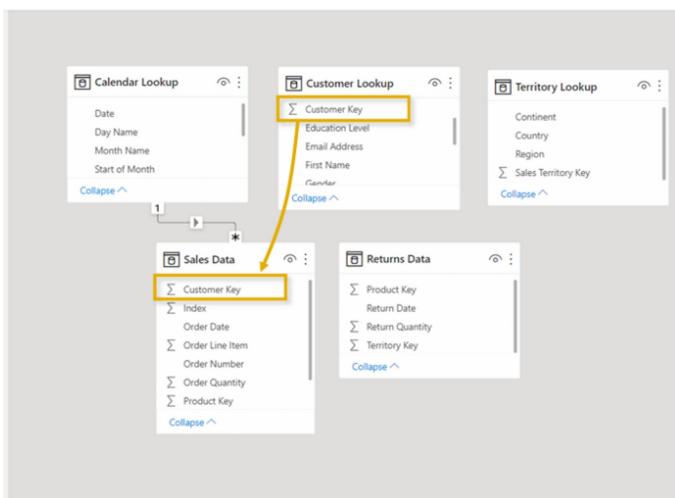
- Merging tables creates redundancy and often requires significantly more memory and processing power to analyze, compared to a relational model with multiple small tables.

THE MODEL VIEW

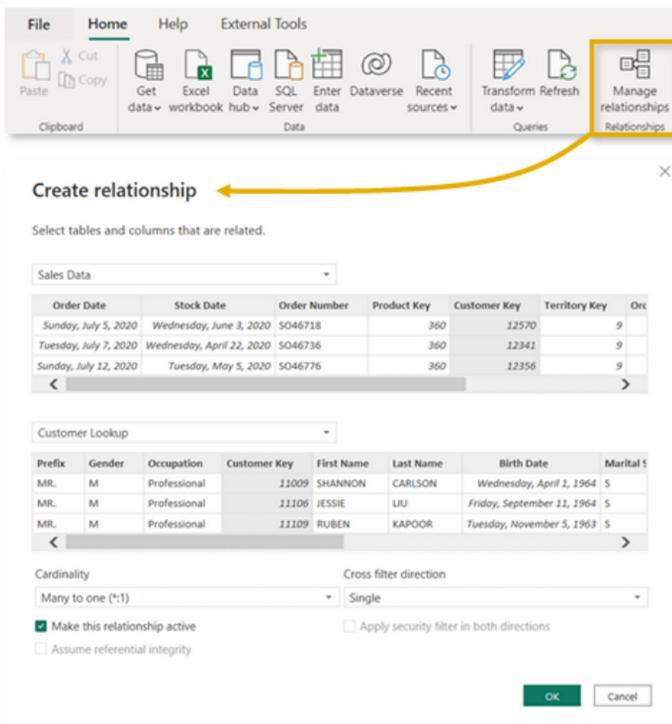


CREATING TABLE RELATIONSHIPS

OPTION 1: Click and drag to connect primary and foreign keys within the Model view



OPTION 2: Add or detect relationships using the Manage Relationships dialog box

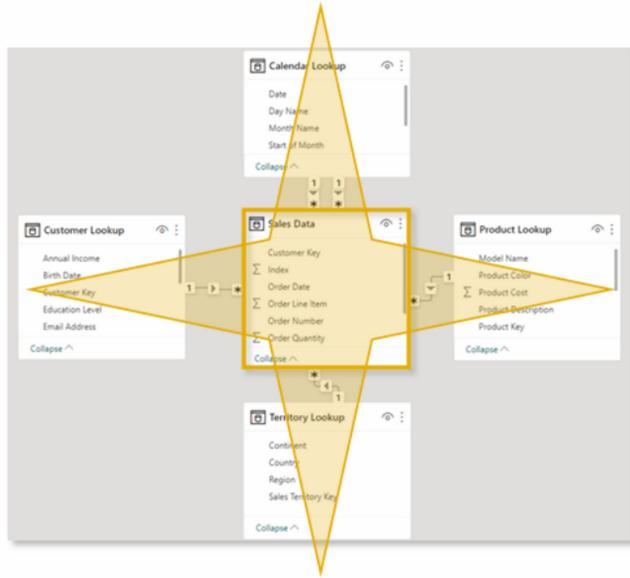


MANAGING & EDITING RELATIONSHIPS

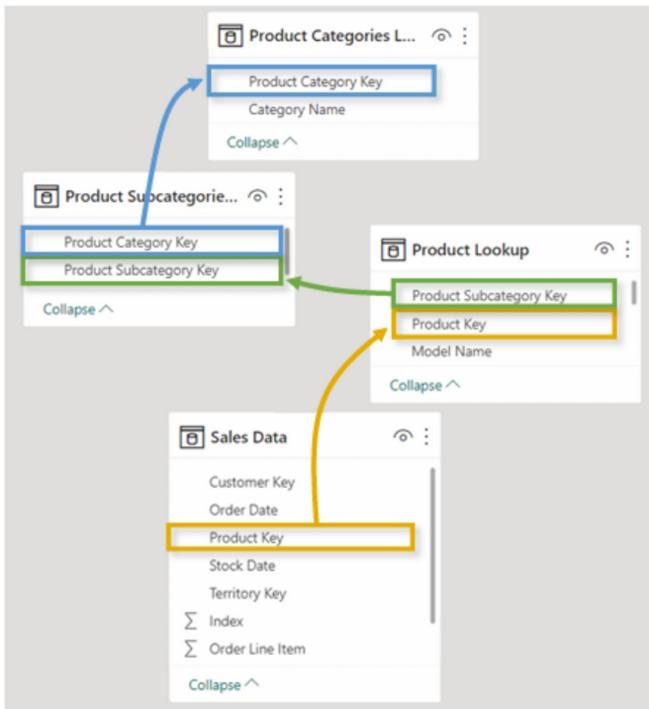
Launch the Manage Relationships dialog box or double-click a relationship to modify it

Editing tools allow you to activate or deactivate relationships and manage cardinality and filter direction—more on that soon!

STAR & SNOWFLAKE SCHEMAS



A star schema is the simplest and most common type of data model, characterized by a single fact table surrounded by related dimension tables

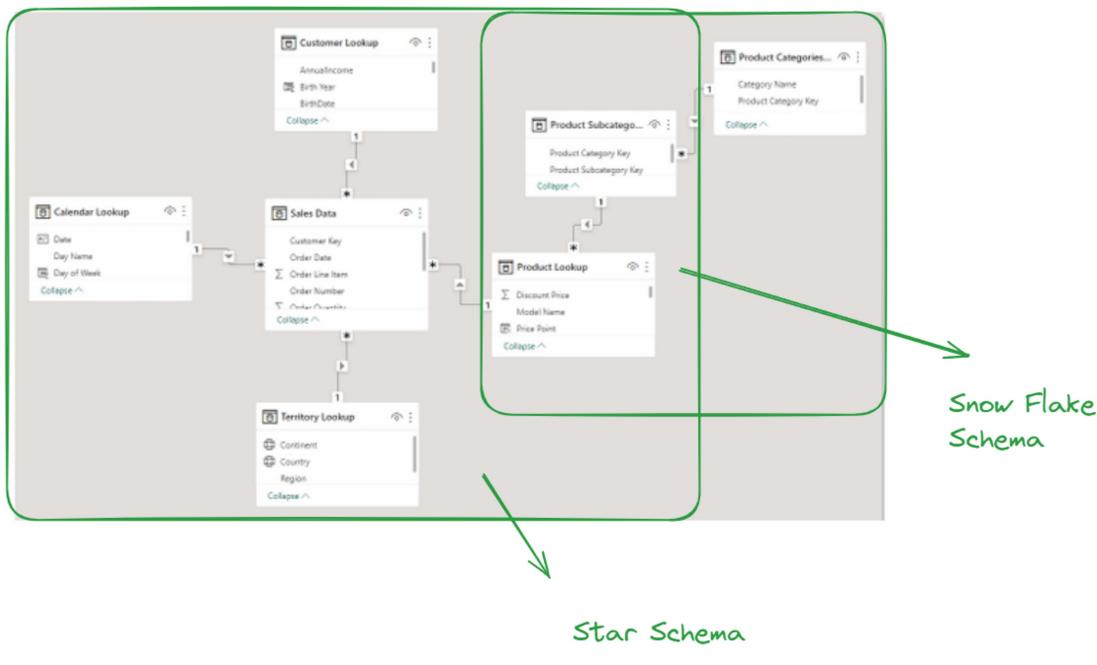


A snowflake schema is an extension of a star, and includes relationships between dimension tables and related sub-dimension tables

ASSIGNMENT: TABLE RELATIONSHIPS

1. Delete all existing table relationships.
2. Create a star schema by creating relationships between the Sales, Calendar, Customer, Product and Territories tables.
3. Connect all three product tables (Product, Subcategory, Category) in a snowflake schema.
4. Use the matrix visual to confirm that you can filter Order Quantity values using fields from each dimension table.

Solution Preview



ProductName	Sum of OrderQuantity	Sum of ReturnQuantity
All-Purpose Bike Stand	234	8
AWC Logo Cap	4151	46
Bike Wash - Dissolver	1700	25
Classic Vest, L	182	4
Classic Vest, M	182	7
Classic Vest, S	157	8
Fender Set - Mountain	3960	54
Half-Finger Gloves, L	840	18
Half-Finger Gloves, M	918	16
Half-Finger Gloves, S	886	15
Hitch Rack - 4-Bike	302	8
HL Mountain Tire	1305	49
HL Road Tire	795	28
Hydration Pack - 70 oz.	695	25
LL Mountain Tire	1560	39
LL Road Tire	1904	43
Long-Sleeve Logo Jersey, L	424	15
Long-Sleeve Logo Jersey, M	408	15
Total	84174	1828

CategoryName	OrderQuantity	ReturnQuantity
Accessories	57809	1130
Bikes	13929	429
Clothing	12436	269
Total	84174	1828

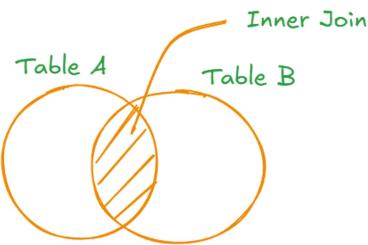
Sub categoryName	OrderQuantity	ReturnQuantity
Bike Racks	302	8
Bike Stands	234	8
Bottles and Cages	15106	288
Caps	4151	46
Cleaners	1706	25
Fenders	3960	54
Gloves	2644	49
Helmos	6034	188
Hydration Packs	695	25
Jerseys	3113	93
Mountain Bikes	4706	136
Road Bikes	7099	223
Shorts	944	40
Socks	1063	22
Tires and Tubes	29772	534
Touring Bikes	2124	70
Vests	521	19
Total	84174	1828

ACTIVE & INACTIVE RELATIONSHIPS

The diagram illustrates the configuration of relationships in Power BI:

- Relationships View:** Shows the relationship between the Sales Data table (Customer Key) and the Calendar table (Date). A relationship is active between Order Date and Date.
- Edit Relationship Dialog:** Shows the "Edit relationship" dialog for the Sales Data -> Calendar relationship. It lists Sales Data and Calendar Lookup tables. The "Order Date" column is selected in Sales Data, and the "Date" column is selected in Calendar Lookup. The cardinality is set to "Many to one (*:1)". The "Make this relationship active" checkbox is checked. Other options include "Cross filter direction" (set to "Single") and "Assume referential integrity".
- Properties Dialog:** Shows the "Properties" dialog for the same relationship. The "Make this relationship active" checkbox is also checked.
- Note:** A callout box states: "The Sales Data table contains two date fields (Order Date & Stock Date), but there can only be one active relationship to the Date key in the Calendar table."
- Relationships View (Bottom):** Shows multiple relationships between the Date, Sales Record, and other tables. A note indicates that the relationship to Stock Date is inactive, while the one to Order Date is active.

Joins in SQL :

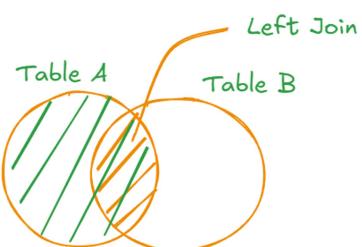


Relationship : [Cardinality]

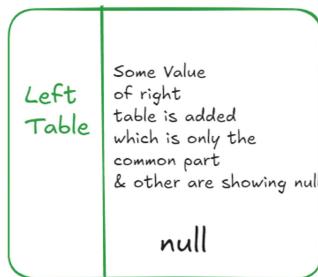
1:1 , 1:* , *:1 *;*

Dimensional Table [p.k] - Unique [1]

Fact Table [F.K] - Storing multiple instances [*]

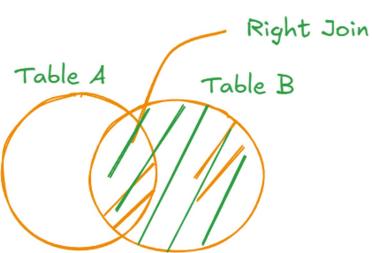


Left Join

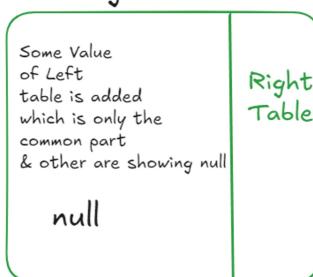


colID	colVal	columnID	columnVal
1	one	1	some value
2	two	NULL	NULL
3	three	3	blah blah blah
4	four	NULL	NULL
5	five	5	hello world

Left Join



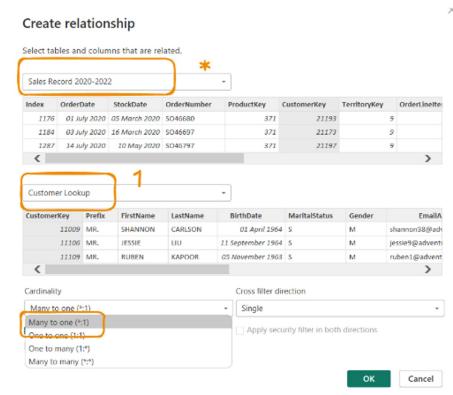
Right Join



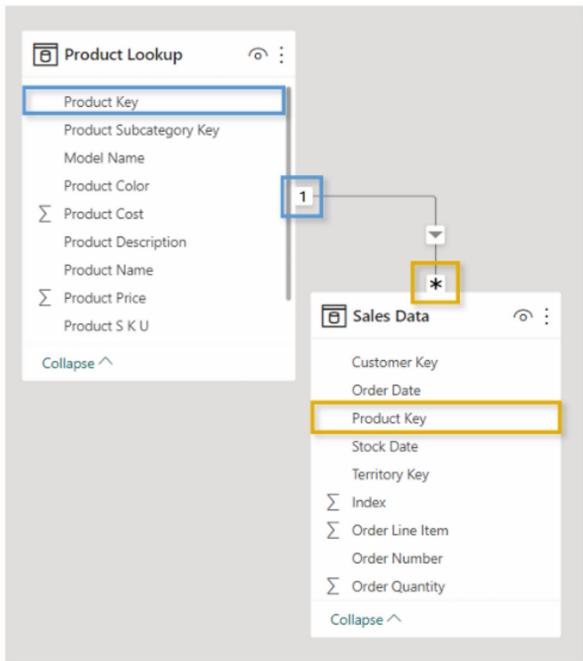
Right Outer Join			
TableA		TableB	
ID	Num1	ID	Num2
A	1	A	5
B	2	C	6
C	3	E	7
D	4	F	8

ID	Num1	Num 2
A	1	5
C	3	6
E	Null	7
F	Null	8

Right Join



RELATIONSHIP CARDINALITY



Cardinality refers to the uniqueness of values in a column.

- Ideally, all relationships in the data model should follow a one-to-many cardinality: one instance of each primary key, and many instances of each foreign key

In this example there is only ONE instance of each Product Key in the Product table (noted by a "1"), since each row contains attributes of a single product (name, SKU, description, price, etc.)

There are MANY instances of each Product Key in the Sales table (noted by an asterisk *), since there are multiple sales for each product

ONE-TO-ONE CARDINALITY

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

- Connecting the two tables above using `product_id` creates a one-to-one relationship, since each product ID only appears once in each table
- This isn't necessarily a "bad" relationship, but you can simplify the model by merging the tables into a single, valid dimension table

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

NOTE: this still respects the rules of normalization, since all rows are unique and capture product-specific attributes

MANY-TO-MANY CARDINALITY →

Product Lookup

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

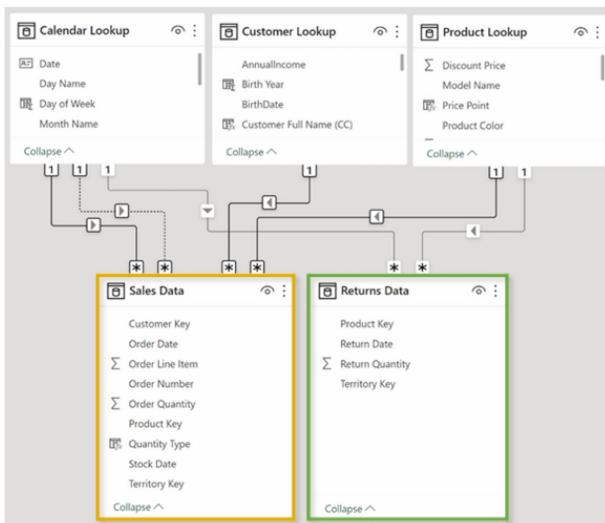
Sales

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

⚠ This relationship has cardinality Many-Many. This should only be used if it is expected that neither column (product_id and product_id) contains unique values, and that the significantly different behavior of Many-many relationships is understood. [Learn more](#)

- If we try to connect the tables above using product_id, we'll get a many-to-many relationship warning since there are multiple instances of product_id in both tables.
- Even if we force this relationship, how would we know which product was actually sold on each date – Cream Soda or Diet Cream Soda?

CONNECTING MULTIPLE FACT TABLES

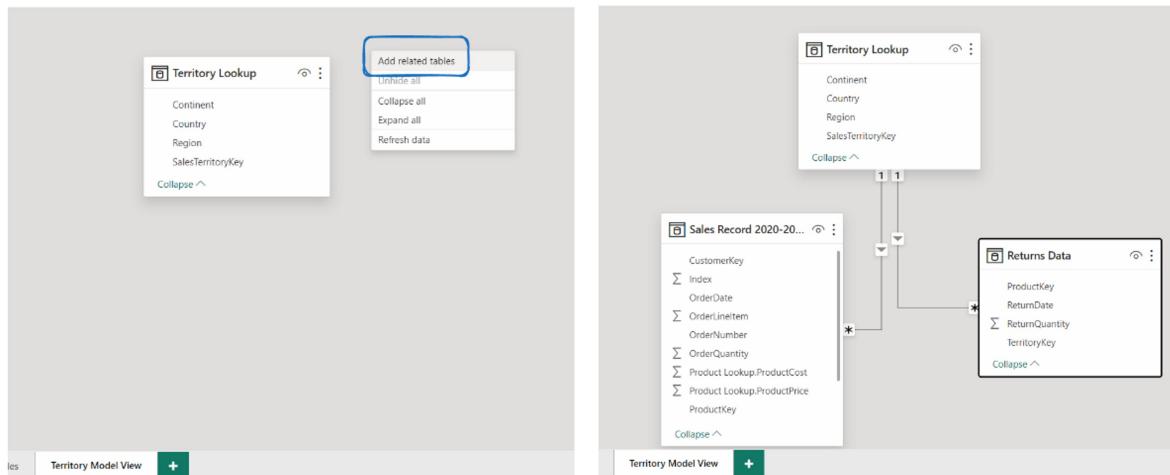
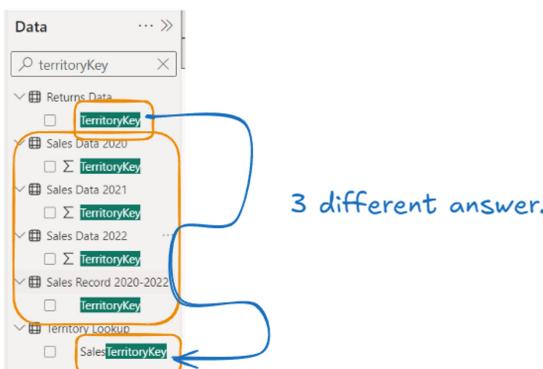


This model contains two fact tables:
Sales Data and Returns Data

- Since there is no primary/foreign key relationship, we can't connect them directly to each other.
- But we can connect each fact table to related lookups, which allows us to filter both sales and returns data using fields from any shared lookup tables.
- We can view orders and returns by product since both tables relate to Product Lookup, but we can't view returns by customer since no relationship exists.

Generally speaking, fact tables should connect through shared dimension tables, not directly to each other

Filter Flow :



Sales Data 2020

Sales Data 2021

Sales Data 2022

CustomerKey

Index

OrderDate

OrderLineItem

OrderNumber

StockDate

territoryKey

Returns Data

TerritoryKey

Sales Record 2020-2022

TerritoryKey

Territory Lookup

SalesTerritoryKey

Hide the unwanted Sales record. That are not useful.

Original Table [Territory Lookup]

SalesTerritoryKey	Sum of OrderQuantity	Sum of ReturnQuantity
1	12513	270
2	40	
3	30	
4	17191	362
5	49	1
6	10894	238
7	7862	186
8	7950	163
9	17951	404
10	9694	204
Total	84174	1828

Sales Record

TerritoryKey	Sum of OrderQuantity	Sum of ReturnQuantity
1	12513	1828
2	40	1828
3	30	1828
4	17191	1828
5	49	1828
6	10894	1828
7	7862	1828
8	7950	1828
9	17951	1828
10	9694	1828
Total	84174	1828

Returns Table

TerritoryKey	Sum of OrderQuantity	Sum of ReturnQuantity
1	84174	270
4	84174	362
5	84174	1
6	84174	238
7	84174	186
8	84174	163
9	84174	404
10	84174	204
Total	84174	1828

Just because there is no direct relation

Just because there is no direct relation