

Data Warehouse Development and ETL Skills Showcase

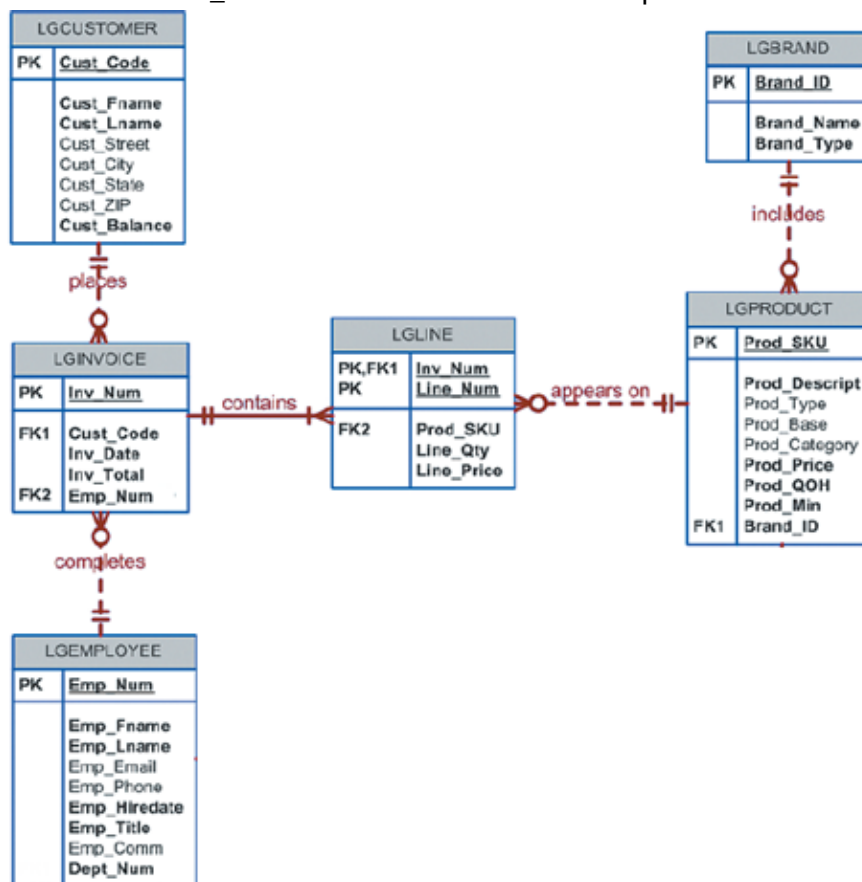
This project demonstrates my ability to design, implement, and optimize a data warehouse using SQL and Tableau Prep. The goal was to create an efficient system to track and analyze orders and sales statistics for a fictional company, “Large.” Key objectives included summarizing sales data by location (city and state levels), time (month, quarter, and year), product, brand, and vendor.

Through this project, I showcased my skills in database schema design, data cleaning and transformation, and SQL query writing for reporting. Below is a detailed overview of the project:

1. Data Warehouse Schema Design

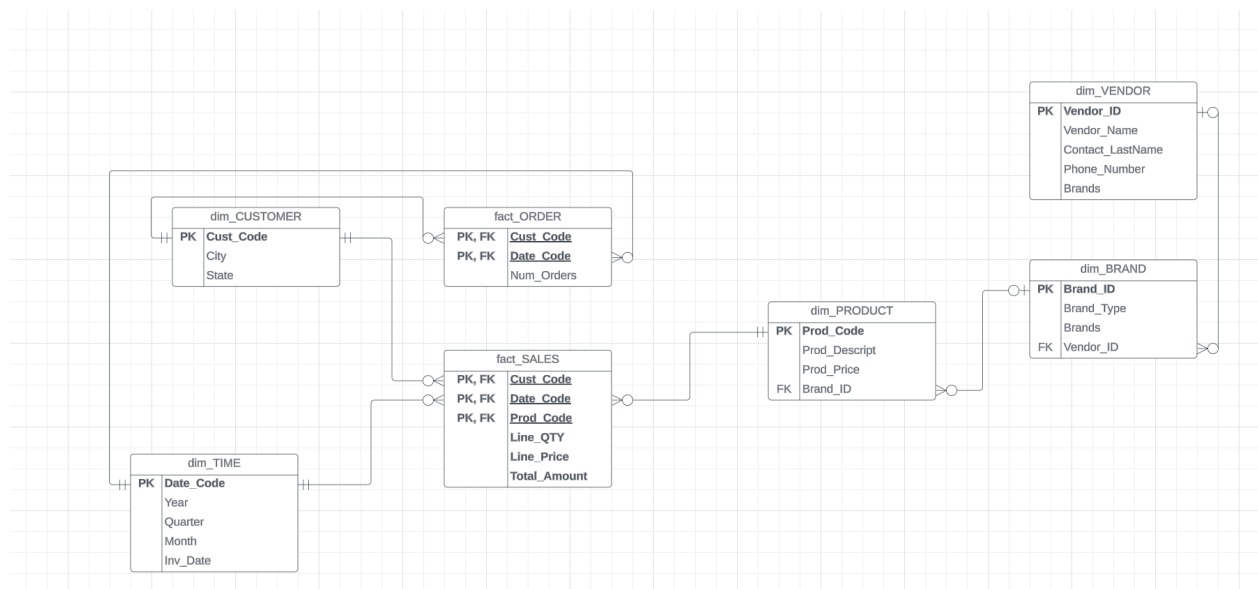
To meet the needs of tracking orders and sales, I designed a star schema for the data warehouse. My design included the following components:

- **Fact Tables:** fact_ORDER: Contains detailed order information.
fact_SALES: Stores sales data, including units sold and dollar amounts.
- **Dimension Tables:** dim_CUSTOMER, dim_PRODUCT, dim_BRAND, dim_VENDOR, and dim_TIME: These tables hold descriptive data for their respective attributes.



Schema Design Approach

- I implemented a **snowflake schema** to normalize dimension tables such as dim_BRAND, dim_VENDOR, and dim_CUSTOMER. This approach reduces redundancy and ensures that each table holds unique and atomic attributes, facilitating easier data updates and maintenance.
- I chose to **maintain separate fact tables** (fact_ORDER and fact_SALES) to represent different business processes and levels of aggregation. This separation improves query performance and allows for targeted analysis based on specific requirements.
- This design enhances data integrity, efficiency, and scalability for handling large datasets.



2. Data Cleaning and Transformation with Tableau Prep

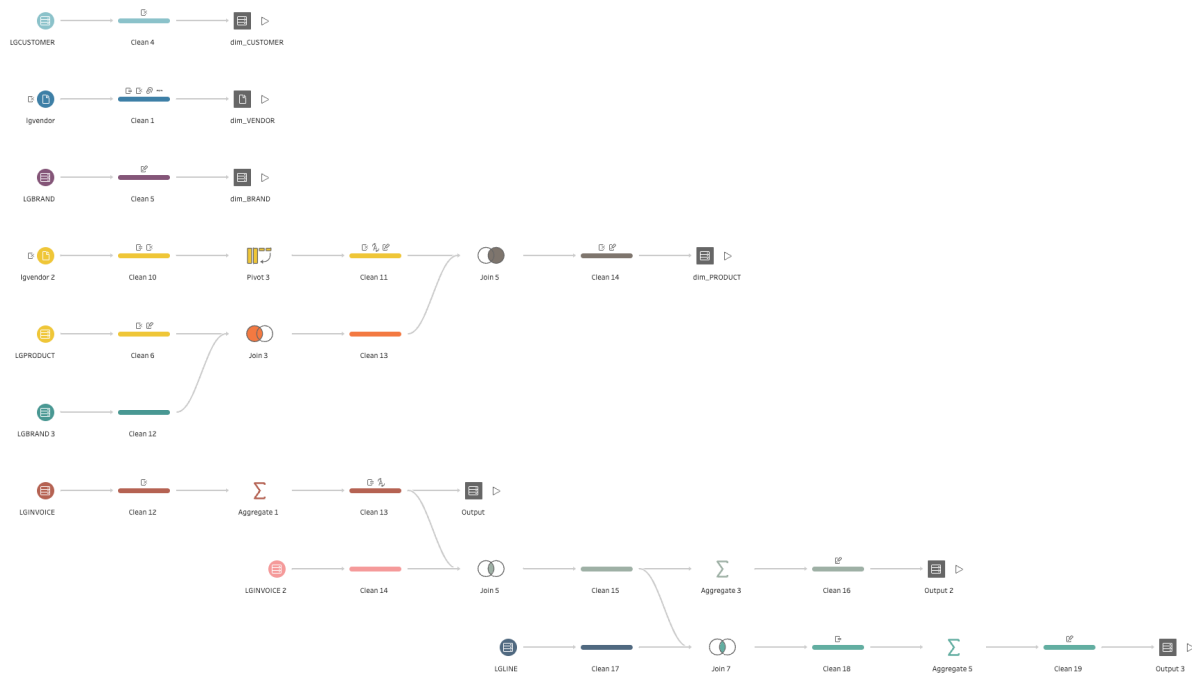
A crucial aspect of this project was the integration of vendor data, stored externally in a CSV file (lgvendor.csv), with the operational database. To achieve this, I used **Tableau Prep** to clean and transform the data before loading it into the data warehouse.

Data Cleaning Steps:

- **Phone Number Standardization:** Combined the Area_Code and Phone_Number columns to create a single unified phone number field.
- **State Name Normalization:** Ensured consistency by replacing state names with the appropriate abbreviations, as the column initially contained a mix of both.

Data Transformation and Merging:

- I used Tableau Prep to join the cleaned vendor data with the operational database (Large) and loaded the merged dataset into the data warehouse (LargeDW). This process highlights my ability to manage and transform large datasets using Tableau Prep, without relying on direct SQL queries for data extraction.



Clean 16 Fields 9 rows

Filter Values...

Identify Duplicate Rows

Rename Fields...

Create Calculated Field...

Changes (12)

Calculated Field

FullPhoneNumber

LEFT(STR([AREA_CODE]), 3) + MID(STR([AREA_CODE]), 4, 3) + "-"

Remove Field

AREA_CODE

Remove Field

PHONE_NUMBER

Group Values

STATE

"Tennessee" replaced by "TN"

Group Values

STATE

"Georgia" replaced by "GA"

Rename Field

Phone_Number

From [FullPhoneNumber] to [Phone_Number]

Rename Field

Vendor_ID

From [Vendor_ID] to [Vendor_ID]

Rename Field

Vendor_Name

From [VENDOR_NAME] to [Vendor_Name]

Rename Field

Contact_LastName

From [CONTACT_LASTNAME] to [Contact_LastName]

Rename Field

State

Vendor_ID 9

21225

21226

21231

21344

22567

23119

24004

24288

25443

Vendor_Name 9

B&K, Inc.

Brackman Bros.

Bryson, Inc.

D&E Supply

Dome Supply

Gomez Bros.

ORDVA, Inc.

Randssets Ltd.

SuperLo, Inc.

Contact_LastName 8

Anderson

Browning

Flushing

Hakford

Ortega

Singh

Smith

Smithson

Phone_Number 9

615-223-3234

615-228-1410

615-228-3245

615-889-2546

615-898-1234

901-678-1419

901-678-3998

904-215-8995

904-227-0093

State 4

FL

GA

KY

TN

Brands 9

BUSTERS

FORESTERS BEST, STUTTENFU

HOME COMFORT, OLDE TYME C

HOMESTEADER FINEST, BINDE

LONG HAUL

PRIME OF LIFE, LE MODE

REGAL HOME

VALU-MATTE

YOUR HOME HELPER

Vendor_ID	Vendor_Name	Contact_LastName	Phone_Number	State	Brands
24004	Brackman Bros.	Browning	615-228-1410	TN	VALU-MATTE
21225	Bryson, Inc.	Smithson	615-223-3234	TN	FORESTERS BEST, STUTTENFURST
24288	ORDVA, Inc.	Hakford	615-898-1234	TN	YOUR HOME HELPER
21231	D&E Supply	Singh	615-228-3245	TN	HOMESTEADER FINEST, BINDER PRIME
22567	Dome Supply	Smith	901-678-1419	GA	BUSTERS
25443	B&K, Inc.	Smith	904-227-0093	FL	PRIME OF LIFE, LE MODE
21226	SuperLo, Inc.	Flushing	904-215-8995	FL	REGAL HOME
21344	Gomez Bros.	Ortega	615-889-2546	KY	HOME COMFORT, OLDE TYME QUALITY
23119	Randssets Ltd.	Anderson	901-678-3998	GA	LONG HAUL

3. Reporting with SQL Queries

Once the data warehouse was built and populated, I created two views in **phpMyAdmin** to provide meaningful insights into sales performance. The queries I developed focused on summarizing sales data by vendor and customer location:

1. **View:** vw_Sales_By_Vendor_State

Purpose: This query generates a report summarizing the total quantity of products sold and the total sales amount in dollars, grouped by vendor and customer state. The WITH ROLLUP feature provides subtotals at the state level. This view allows businesses to assess vendor performance across different regions, identifying key markets.

Query:

```
CREATE VIEW vw_Sales_By_Vendor_State AS
SELECT v.Vendor_Name, c.Cust_State AS Customer_State,
       SUM(s.Line_QTY) AS Total_Quantity_Sold,
       SUM(s.Total_Amount) AS Total_Dollar_Amount_Sold
FROM fact_SALES s
JOIN dim_PRODUCT p ON s.Prod_Code = p.Prod_Code
JOIN dim_VENDOR v ON p.Vendor_ID = v.Vendor_ID
JOIN dim_CUSTOMER c ON s.Cust_Code = c.Cust_Code
GROUP BY v.Vendor_Name, c.Cust_State WITH ROLLUP;
```

2. **View:** vw_Orders_By_Quarter_State

Purpose: This query calculates the total number of orders placed, broken down by quarter (including year) and customer state. It uses WITH ROLLUP to add subtotals for each quarter, allowing analysts to track order trends over time and across regions. It is useful for identifying seasonal patterns and regional demand fluctuations.

Query:

```
CREATE VIEW vw_Orders_By_Quarter_State AS
SELECT t.Year, t.Quarter, c.Cust_State AS Customer_State,
       COUNT(*) AS Total_Orders
FROM fact_ORDER o
JOIN dim_CUSTOMER c ON o.Cust_Code = c.Cust_Code
JOIN dim_TIME t ON o.Date_Code = t.Date_Code
GROUP BY t.Year, t.Quarter, c.Cust_State WITH ROLLUP;
```

The screenshot shows a database management interface. On the left is a tree view of the database structure. The main panel on the right displays the results of a query executed against the view 'vw_orders_by_quarter_state'.

Query: `SELECT * FROM `vw_orders_by_quarter_state``

Results:

Year	Quarter	Customer_State	Total_Orders
2015	1	AL	19
2015	1	CT	6
2015	1	FL	25
2015	1	GA	16
2015	1	IN	31
2015	1	KY	21
2015	1	MA	19
2015	1	MD	32
2015	1	ME	7
2015	1	MI	15
2015	1	MS	12
2015	1	NC	31
2015	1	NH	9
2015	1	NJ	15
2015	1	NY	75
2015	1	OH	40
2015	1	PA	83
2015	1	RI	7
2015	1	SC	16
2015	1	TN	9
2015	1	VA	24
2015	1	VT	13
2015	1	WV	34
2015	1	NULL	559
2015	2	AL	23

3. **View:** vw_Orders_Sales_By_City_State

Purpose: This view aggregates the total number of orders, total quantity sold, and total sales revenue by customer city and state. It gives businesses insights into which cities and states generate the most sales, helping optimize marketing and operational strategies in those locations.

Query:

-- Orders and Sales by City and State

```
CREATE VIEW vw_Orders_Sales_By_City_State AS
SELECT c.Cust_City, c.Cust_State,
```

```

SUM(o.Num_Orders) AS Total_Orders,
SUM(s.Line_QTY) AS Total_Quantity_Sold,
SUM(s.Total_Amount) AS Total_Dollar_Amount_Sold
FROM fact_ORDER o
JOIN dim_CUSTOMER c ON o.Cust_Code = c.Cust_Code
JOIN fact_SALES s ON o.Cust_Code = s.Cust_Code
GROUP BY c.Cust_City, c.Cust_State WITH ROLLUP;

```

Extra options

Cust_City	Cust_State	Total_Orders	Total_Quantity_Sold	Total_Dollar_Amount_Sold
Abbot Village	ME	56	184	3090.96
Abbot Village	NULL	56	184	3090.96
Abel	AL	30	81	1642.59
Abel	NULL	30	81	1642.59
Acton	ME	4	6	91.94
Acton	NULL	4	6	91.94
Adamsville	RI	12	42	713.18
Adamsville	NULL	12	42	713.18
Adelphi	OH	80	265	5507.85
Adelphi	NULL	80	265	5507.85
Alafaya	FL	114	250	5108.25
Alafaya	NULL	114	250	5108.25
Albany	NY	72	228	4114.92
Albany	NULL	72	228	4114.92
Albion	NJ	36	108	1929.42
Albion	NULL	36	108	1929.42
Alexandria	VA	33	114	1936.56
Alexandria	NULL	33	114	1936.56
Alford	MA	6	18	301.82
Alford	NULL	6	18	301.82
Algonac	MI	10	20	702.20
Algonac	NULL	10	20	702.20
Allentown	PA	14	42	1004.00
Allentown	NULL	14	42	1004.00
Allred	TN	189	498	10268.22

>

☐ Show all

Number of rows: 25

Filter rows: Search this table

>



Show all

Number of rows:

25

Filter rows:

Search this table

4. **View:** vw_Orders_Sales_By_Month_Quarter_Year

Purpose: This query breaks down the total orders, total quantity sold, and total sales revenue by month, quarter, and year. It highlights sales trends at different time intervals, making it useful for time-based performance analysis, such as detecting seasonality or yearly growth patterns.

Query:

```
-- Orders and Sales by Month, Quarter, and Year
CREATE VIEW vw_Orders_Sales_By_Month_Quarter_Year AS
SELECT t.Year, t.Quarter, t.Month,
       COUNT(*) AS Total_Orders,
       SUM(s.Line_QTY) AS Total_Quantity_Sold,
       SUM(s.Total_Amount) AS Total_Dollar_Amount_Sold
FROM fact_ORDER o
JOIN dim_TIME t ON o.Date_Code = t.Date_Code
JOIN fact_SALES s ON o.Cust_Code = s.Cust_Code
GROUP BY t.Year, t.Quarter, t.Month WITH ROLLUP;
```


Show all

Number of rows: 25

Filter rows: Search this table

Extra options

Year	Quarter	Month	Total_Orders	Total_Quantity_Sold	Total_Dollar_Amount_Sold
2015	1	1	397	1188	23244.78
2015	1	2	5844	17858	339735.82
2015	1	3	5023	15232	292628.81
2015	1	NULL	11264	34278	655609.41
2015	2	4	4715	13968	264683.38
2015	2	5	4598	13718	264550.80
2015	2	6	3883	11719	223894.06
2015	2	NULL	13196	39405	753128.24
2015	3	7	3497	10502	200719.75
2015	3	8	3457	10425	197996.94
2015	3	9	3068	9145	176262.90
2015	3	NULL	10022	30072	574979.59
2015	4	10	2317	6966	134194.15
2015	4	11	2185	6457	121866.72
2015	4	12	1703	5224	98238.48
2015	4	NULL	6205	18647	354299.35
2015	NULL	NULL	40687	122402	2338016.59
2016	1	1	858	2646	51326.22
2016	1	NULL	858	2646	51326.22
2016	NULL	NULL	858	2646	51326.22
NULL	NULL	NULL	41545	125048	2389342.81

Show all

Number of rows: 25

Filter rows: Search this table

Query results operations

5. **View:** vw_Orders_Sales_By_Product

Purpose: This view aggregates order and sales data by product, providing the total number of orders, total quantity sold, and total revenue. It also includes the product description and brand ID, giving detailed insight into product-level performance, helping businesses to make informed decisions regarding product offerings.

Query:

-- Orders and Sales by Product

```
CREATE VIEW vw_Orders_Sales_By_Product AS
SELECT p.Prod_Code, p.Prod_Descript, p.Brand_ID,
       SUM(o.Num_Orders) AS Total_Orders,
       SUM(s.Line_QTY) AS Total_Quantity_Sold,
       SUM(s.Total_Amount) AS Total_Dollar_Amount_Sold
FROM fact_SALES s
JOIN dim_PRODUCT p ON s.Prod_Code = p.Prod_Code
JOIN fact_ORDER o ON s.Cust_Code = o.Cust_Code
GROUP BY p.Prod_Code, p.Prod_Descript, p.Brand_ID;
```

Extra options						
Prod_Code	Prod_Descript	Brand_ID	Total_Orders	Total_Quantity_Sold	Total_Dollar_Amount_Sold	
1010-MIW	Dry Fall, Latex, Semi-Gloss (MPI Gloss Level 5)	28	227	738	16083.42	
1021-MTI	Elastomeric, Exterior, Industrial Grade, Water Bas...	35	114	336	21139.44	
1045-DUY	Traffic Marking Paint, Alkyd	35	163	513	3380.67	
1067-KBB	Primer, Galvanized Metal, Cementitious	30	208	599	8919.11	
1074-VVJ	Light Industrial Coating, Exterior, Water Based ('...	25	180	584	13601.36	
1078-QXL	Varnish, Interior, Semi-Gloss	23	179	442	6447.32	
1099-WFZ	Primer, Epoxy, Anti-Corrosive, for Metal	35	235	674	10103.26	
1143-RGX	Shellac	27	123	364	4340.36	
1150-MMR	Primer, Alkyd, Quick Dry, for Metal	35	150	524	6192.76	
1153-AWY	Polyurethane, Moisture Cured, Pigmented, Intermedi...	28	138	343	5141.57	
1198-STR	Latex, Exterior, Gloss (MPI Gloss Level 6)	28	161	521	12290.39	
1200-KBU	Stain, Semi-Transparent, for Interior Wood	25	146	369	7713.81	
1203-AIS	Fire Retardant Coating, Latex, Interior, Flat (ULC...	35	191	592	21306.08	
1336-FVM	Alkyd, Sanding Sealer, Clear	33	109	317	7313.09	
1403-TUY	Sealer, Water Based, for Concrete Floors	29	199	533	22870.67	
1433-MMY	Heat Resistant Coating, (Up to 205°C/402°F), MPI G...	27	167	509	12210.91	
1485-NNI	Primer Sealer, for Multicolor Systems	30	186	572	10791.03	
1504-LVK	Floor Paint, Latex, Low Gloss	33	163	465	10225.35	
1580-VCE	Latex, Interior, High Performance Architectural, (...)	29	193	549	18935.01	
1663-CDD	Light Industrial Coating, Exterior, Water Based, S...	25	167	535	12343.75	
1747-XNI	Paste, Wood Filler	33	185	504	2010.96	
1838-LZI	Latex, Exterior, High Build	25	226	761	9541.93	
1854-AFJ	Epoxy Deck Coating (Slip-Resistant)	30	181	574	12966.66	
1871-GWZ	Latex, Exterior, Low Sheen (MPI Gloss Level 3-4)	28	220	566	14124.34	
2006-ZBH	Latex, Exterior Flat (MPI Gloss Level 1)	35	141	448	4909.77	

> | ☐ Show all | Number of rows: 25 | Filter rows:

6. View: vw_Orders_Sales_By_Brand

Purpose: This query summarizes orders and sales by brand. It provides the total number of orders, quantity sold, and total sales revenue for each brand, enabling businesses to track brand performance and evaluate customer preferences at the brand level.

Query:

-- Orders and Sales by Brand

```

CREATE VIEW vw_Orders_Sales_By_Brand AS
SELECT b.Brand_ID, b.Brand_Name,
       SUM(o.Num_Orders) AS Total_Orders,
       SUM(s.Line_QTY) AS Total_Quantity_Sold,
       SUM(s.Total_Amount) AS Total_Dollar_Amount_Sold
FROM fact_ORDER o
JOIN dim_PRODUCT p ON o.Prod_Code = p.Prod_Code
JOIN dim_BRAND b ON p.Brand_ID = b.Brand_ID
JOIN fact_SALES s ON o.Cust_Code = s.Cust_Code
GROUP BY b.Brand_ID, b.Brand_Name;

```

7. **View:** vw_Orders_Sales_By_Vendor

Purpose: This query calculates total orders, total quantity sold, and total sales revenue, grouped by vendor. It enables a detailed performance analysis of vendors, allowing businesses to assess vendor contribution to overall sales and identify key vendor partnerships.

Query:

```

-- Orders and Sales by Vendor
CREATE VIEW vw_Orders_Sales_By_Vendor AS
SELECT v.Vendor_ID, v.Vendor_Name,
       SUM(o.Num_Orders) AS Total_Orders,
       SUM(s.Line_QTY) AS Total_Quantity_Sold,
       SUM(s.Total_Amount) AS Total_Dollar_Amount_Sold
FROM fact_ORDER o
JOIN dim_PRODUCT p ON o.Prod_Code = p.Prod_Code
JOIN dim_VENDOR v ON p.Vendor_ID = v.Vendor_ID
JOIN fact_SALES s ON o.Cust_Code = s.Cust_Code
GROUP BY v.Vendor_ID;

```

<div> <div>arter_state</div> <div>py_city</div> <div>py_mon</div> <div>py_prod</div> <div>for_state</div> </div>	Extra options			
	Vendor_Name	Customer_State	Total_Quantity_Sold	Total_Dollar_Amount_Sold
	B&K, Inc.	AL	140	2591.32
	B&K, Inc.	CT	61	1321.09
	B&K, Inc.	DE	11	175.49
	B&K, Inc.	FL	250	5106.93
	B&K, Inc.	GA	134	2005.46
	B&K, Inc.	IN	232	4170.78
	B&K, Inc.	KY	219	3542.76
	B&K, Inc.	MA	160	3100.80
	B&K, Inc.	MD	256	5151.34
	B&K, Inc.	ME	67	1274.13
	B&K, Inc.	MI	133	2231.37
	B&K, Inc.	MS	116	1986.04
	B&K, Inc.	NC	251	4375.09
	B&K, Inc.	NH	77	1364.83
	B&K, Inc.	NJ	159	2819.31
	B&K, Inc.	NY	543	10465.52
	B&K, Inc.	OH	279	5275.21
	B&K, Inc.	PA	603	11380.90
	B&K, Inc.	RI	31	508.30
	B&K, Inc.	SC	119	2288.89
	B&K, Inc.	TN	111	2266.69
	B&K, Inc.	VA	170	3167.15
	B&K, Inc.	VT	95	1905.75
	B&K, Inc.	WV	146	3296.89
	B&K, Inc.	NULL	4363	81772.04
	<div> <div>></div> <div><input type="checkbox"/> Show all</div> <div>Number of rows: 25</div> <div>Filter rows: <input type="text" value="Search this table"/></div> </div>			
	Query results operations			

Each of these views provides valuable insights into different facets of the business, ranging from vendor performance to product sales, and regional order trends. They form a comprehensive analytical framework for assessing sales, orders, and customer behavior across multiple dimensions, aiding decision-making in both strategic and operational contexts.

Conclusion

This project demonstrates my ability to design and build a data warehouse from scratch, clean and transform data using Tableau Prep, and write SQL queries for meaningful business insights. By combining these tools and techniques, I can create efficient data pipelines and deliver insightful reports that support decision-making in any organization.

Next Steps

Feel free to explore the project files, including the SQL queries and Tableau Prep flows, in the repository. If you have any questions or would like to discuss how I can contribute to your team, please don't hesitate to reach out!