Super Store’s Product Sales

SQL PROJECT

Yuvraj Giri

17 sep, 2024

Super Store's Product Sales

INTRODUCTION

"The SuperStore dataset comprises a comprehensive sales record from a superstore, containing 9,994 entries across 19 distinct fields. The dataset includes order details, anonymized customer information, product specifics, and financial metrics. This report analyzes various aspects of the dataset to extract meaningful insights

Super Store's Product Sales

QUESTIONS

01 Total Sales per Region

02 Top 5 Customers by Total Purchases

03 Numbers of Orders per Product Category

04 Total Discount Given by Region

05 Average Profit Margin by Product Category

06 Count of Orders in Each Year

07 Customer with the Most Orders

08 Top Regions by Total Profit

09 Most Popular Product in Each Region

10 Month with the Highest Sales in Each Year

QUESTIONS

11 Identify Customers with No Purchases in the last 6 Months ending 2022

12 Calculate Most Repeated Customers

13 Identify Products with Consistently Increasing Sales

14 Customer Lifetime Value (CLV) Calculation of Top 10 Customer

15 Profitability Analysis by Order Size

16 Predict High-Value Customers

INTRODUCING MYSELF

My name is Yuvraj Giri. My aim is to be a Data Analyst. Currently I'm learning SQL and have upper intermediate skills. This is my first Project of sales data. I've included various questions on this project which will be beneficial for making data driven business decisions and extracting meaningful insights from this.

Super Store's Product Sales

01

```sql
-- Total Sales per Region
SELECT
    region,
    ROUND(SUM(sales)::NUMERIC,2)
    AS Total_Sales
FROM Sales
GROUP BY 1
ORDER BY 2 DESC;

```

	region character varying (30)	total_sales numeric
1	West	725457.82
2	East	678781.24
3	Central	501239.89
4	South	391721.91

Super Store's Product Sales

02

```sql
-- Top 5 Customers by Total Puchases
SELECT
    customer,
    ROUND(SUM(sales)::NUMERIC,2)
    AS "Total_Purchase"
FROM Sales
GROUP BY 1
ORDER BY 2 DESC
LIMIT 5;
```

	customer character varying (25)	Total_Purchase numeric
1	Sean Miller	25043.05
2	Tamara Chand	19052.22
3	Raymond Buch	15117.34
4	Tom Ashbrook	14595.62
5	Adrian Barton	14473.57

Super Store's Product Sales

03

```sql
-- Number of Orders per Product Category
SELECT
    category,
    COUNT(order_id)
    AS Total_Orders
FROM Sales
GROUP BY 1
ORDER BY 2 DESC;
```

	category character varying (30)	total_orders bigint
1	Office Supplies	6026
2	Furniture	2121
3	Technology	1847

04

```sql
-- Total Discount Given by Region
SELECT
    region,
    ROUND(SUM(discount)::NUMERIC,2)
    AS Total_Discount
FROM Sales
GROUP BY 1
ORDER BY 2 DESC;
```

	region character varying (30)	total_discount numeric
1	Central	558.34
2	East	414.00
3	West	350.20
4	South	238.55

05

```sql
-- Average Profit  Margin by Product Category
SELECT
    category,
    ROUND(AVG(profit_margin)::NUMERIC,2)
    AS Average_Profit_Margin
FROM Sales
GROUP BY 1
ORDER BY 2 DESC;
```

	category character varying (30)	average_profit_margin numeric
1	Technology	0.16
2	Office Supplies	0.14
3	Furniture	0.04

Super Store's Product Sales

06

```sql
-- Top 5 Customers with the Most Orders
SELECT
    customer,
    COUNT(order_id)
    AS Total_Orders
FROM Sales
GROUP BY 1
ORDER BY 2 DESC
LIMIT 5;
```

	customer character varying (25)	total_orders bigint
1	William Brown	37
2	Matt Abelman	34
3	Paul Prost	34
4	John Lee	34
5	Jonathan Doherty	32

07

```sql
-- Count of Orders in Each Year
SELECT
    EXTRACT(YEAR FROM order_date)
    AS Year,
    COUNT(order_id)
    AS Total_Orders
FROM Sales
GROUP BY 1
ORDER BY 1;
```

	year numeric	total_orders bigint
1	2019	1993
2	2020	2102
3	2021	2587
4	2022	3312

08

```sql
-- Top Regions by Total Profit
SELECT
    region,
    ROUND(SUM(profit)::NUMERIC,2)
    AS Total_Profit
FROM Sales
GROUP BY 1
ORDER BY 2 DESC
LIMIT 1;
```

	region character varying (30)	total_profit numeric
1	West	108418.45

Super Store's Product Sales

09

```sql
-- Most Popular Product in Each Region
WITH my as (SELECT
    region,
    product_name,
    COUNT(product_name) as Total_Orders,
DENSE_RANK()
OVER( PARTITION BY region
ORDER BY COUNT(product_name) DESC ) AS rank
FROM Sales
GROUP BY 1,2)

SELECT
    region,
    product_name,
    Total_orders
FROM my
WHERE rank=1
```

	region character varying (30)	product_name character varying (200)	total_orders bigint
1	Central	Staple envelope	13
2	Central	Easy-staple paper	13
3	Central	Staples	13
4	East	Staple envelope	17
5	South	Staples	9
6	South	Easy-staple paper	9
7	West	Staples	13

Super Store's Product Sales

10

```sql
--Month with the Highest Sales in Each Year
WITH my AS (SELECT
    EXTRACT(YEAR FROM order_date) AS Year,
    EXTRACT(MONTH FROM order_date)AS Month,
    ROUND(SUM(sales)::NUMERIC,2)AS Total_Sales,
DENSE_RANK() OVER(PARTITION BY EXTRACT(YEAR FROM order_date)
ORDER BY ROUND(SUM(sales)::NUMERIC,2) DESC)
AS rank
FROM Sales
GROUP BY 1,2)

SELECT
    Year,
    Month,
    Total_Sales
FROM my
WHERE rank=1
```

	year numeric	month numeric	total_sales numeric
1	2019	9	81777.35
2	2020	11	75972.56
3	2021	12	96999.04
4	2022	11	118447.83

Super Store's Product Sales

11

```sql
-- Identify Customers with No Purchases in the Last 6 Months ending 2022
SELECT
    customer
FROM Sales
WHERE order_date
NOT BETWEEN '2022-07-01' AND '2022-12-31'
```

	customer character varying (25)
1	Darren Powers
2	Phillina Ober
3	Phillina Ober
4	Phillina Ober
5	Mick Brown
6	Jack O'Briant
7	Lycoris Saunders
8	Maria Etezadi

Total rows: 1000 of 7872 Query complete 00:00:00.880 Ln 2, Col 1

12

```sql
-- Calculate Most Repeated Customer
SELECT
    customer,
    COUNT(customer) AS Repeat_count
FROM Sales
GROUP BY 1
ORDER BY 2 DESC
LIMIT 1
```

	customer character varying (25)	repeat_count bigint
1	William Brown	37

Super Store's Product Sales

13

```sql
-- Identify Products with Consistently Increasing Sales
WITH total_sales_1 AS (SELECT
    EXTRACT(YEAR FROM order_date) AS Year,
    Product_name,
    ROUND(SUM(sales)::NUMERIC,2) AS Total_sales
FROM Sales
GROUP BY 1,2
ORDER BY 1),

last_sales AS (SELECT
    Year,
    Product_name,
    Total_sales,
LAG(Total_sales)
OVER( PARTITION BY Product_name ORDER BY Year) AS Last_year_total
FROM total_sales_1)

SELECT
    Product_name
    FROM last_sales
WHERE Total_sales > Last_year_total
GROUP BY 1
HAVING COUNT(Year) = (SELECT COUNT(DISTINCT Year)-1 FROM total_sales_1)
```

	product_name character varying (200)
1	3M Hangers With Command Adhesive
2	Acco PRESSTEX Data Binder with Storage Hooks, Light Blue, 9 1/2" X 11"
3	Acco Suede Grain Vinyl Round Ring Binder
4	Astroparche Fine Business Paper
5	AT&T 841000 Phone
6	Avery Arch Ring Binders
7	Avery Durable Slant Ring Binders, No Labels
8	Belkin Premiere Surge Master II 8-outlet surge protector
9	Bush Andora Bookcase, Maple/Graphite Gray Finish
10	Cardinal EasyOpen D-Ring Binders
11	Conquest 14 Commercial Heavy-Duty Upright Vacuum, Collection System, Accessory ...
12	Eureka The Boss Cordless Rechargeable Stick Vac
13	GBC DocuBind P50 Personal Binding Machine
14	GBC Premium Transparent Covers with Diagonal Lined Pattern
15	Global Deluxe Steno Chair
16	Global Leather and Oak Executive Chair, Black
17	invisibleSHIELD by ZAGG Smudge-Free Screen Protector
18	Logitech Desktop MK120 Mouse and keyboard Combo
19	Logitech Wireless Headset h800
20	Logitech Wireless Marathon Mouse M705
21	Martin-Yale Premier Letter Opener
22	Office Star - Contemporary Task Swivel chair with Loop Arms, Charcoal
23	Pressboard Covers with Storage Hooks, 9 1/2" x 11", Light Blue
24	Pressboard Hanging Data Binders for Unburst Sheets

Total rows: 33 of 33 Query complete 00:00:00.290 Ln 19, Col 14

Super Store's Product Sales

14

```sql
-- Customer Lifetime Value (CLV) Calculaiton of Top 10 Customers
SELECT
    customer,
    (AVG(sales) * (COUNT(order_id) / COUNT(DISTINCT EXTRACT(YEAR FROM order_date))) *
    (MAX(EXTRACT(YEAR FROM order_date)) - MIN(EXTRACT(YEAR FROM order_date)) +1)
    * SUM(profit_margin)) AS Customer_Lifetime_Value
FROM Sales
GROUP BY 1
ORDER BY 2 DESC
limit 10
```

	customer character varying (25)	customer_lifetime_value double precision
1	Edward Hooks	97062.61453218765
2	Greg Tran	94532.4286791228
3	Pete Kriz	84579.86139020912
4	Karen Ferguson	82960.70767333332
5	John Lee	76057.284759247
6	Clay Ludtke	74586.14283000003
7	Sanjit Chand	74407.9618409091
8	Jonathan Doherty	57440.221247418915
9	Raymond Buch	57042.759154960884
10	Keith Dawkins	56167.93181379705

Super Store's Product Sales

15

```sql
-- Profitability Analysis by Order Size
SELECT
    CASE
    WHEN sales <= 8000 THEN 'Small Order'
    WHEN sales BETWEEN 8000 AND 16000 THEN 'Medium Order'
    WHEN sales > 16000 THEN 'Big Order'
    END AS Order_Size,
    COUNT(order_id) AS Total_Orders,
    ROUND(AVG(sales)::NUMERIC,2) AS Average_Sales,
    ROUND(SUM(sales)::NUMERIC,2) AS Total_Sales,
    ROUND(SUM(profit)::NUMERIC,2) AS Total_Profit,
    ROUND(SUM(discount)::NUMERIC,2) AS Total_Discount
FROM Sales
GROUP BY 1
ORDER BY 1 ASC
```

	order_size text	total_orders bigint	average_sales numeric	total_sales numeric	total_profit numeric	total_discount numeric
1	Big Order	2	20069.22	40138.43	6588.90	0.50
2	Medium Order	10	9764.00	97640.05	30510.28	1.00
3	Small Order	9982	216.33	2159422.38	249297.85	1559.59

Super Store's Product Sales

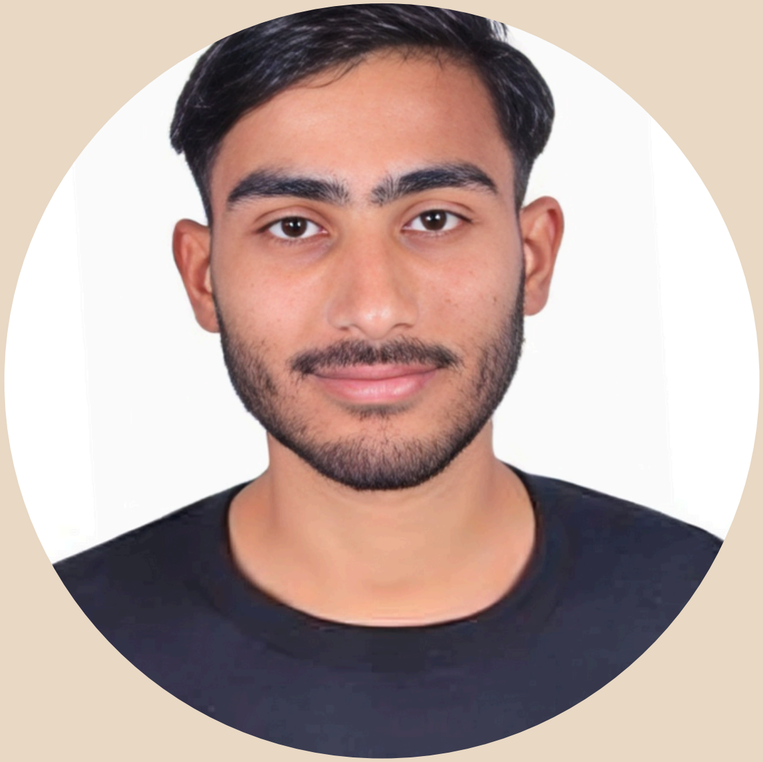
16

```sql
-- Predict High-Value Customers
SELECT
    customer,
    COUNT(order_id) AS Total_Orders,
    ROUND(AVG(sales)::NUMERIC,2) AS Average_Sales,
    ROUND(SUM(sales)::NUMERIC,2) AS Total_Sales,
    ROUND(SUM(profit)::NUMERIC,2) AS Total_Profit,
    ROUND(SUM(discount)::NUMERIC,2) AS Total_Discount
FROM Sales
GROUP BY 1
HAVING COUNT(order_id) > (SELECT COUNT(order_id)/COUNT(DISTINCT customer) FROM sales)
AND ROUND(SUM(sales)::NUMERIC,2) > (SELECT AVG(sales) FROM sales)
AND ROUND(SUM(profit)::NUMERIC,2) > (SELECT AVG(profit) FROM sales)
ORDER BY 2 DESC
```

	customer character varying (25)	total_orders bigint	average_sales numeric	total_sales numeric	total_profit numeric	total_discount numeric
1	William Brown	37	166.49	6160.10	714.33	7.60
2	Paul Prost	34	213.31	7252.61	1495.09	3.70
3	John Lee	34	288.23	9799.92	228.91	3.00
4	Matt Abelman	34	126.45	4299.16	1240.23	3.20
5	Chloris Kastensmidt	32	98.59	3154.86	141.28	7.50
6	Seth Vernon	32	358.47	11470.95	1199.42	5.00
7	Edward Hooks	32	322.22	10310.88	1393.52	2.30
8	Jonathan Doherty	32	237.84	7610.86	1050.27	2.40
9	Arthur Prichep	31	107.21	3323.56	579.31	2.90
10	Emily Phan	31	176.71	5478.06	144.96	6.12
11	Dean percer	29	248.23	7198.76	333.36	5.40
12	Sally Hughsby	29	117.48	3406.84	558.47	3.95
13	Greg Tran	29	407.59	11820.12	2163.43	2.90
14	Brian Moss	29	251.52	7294.19	2199.28	2.45
15	Ken Lonsdale	29	488.80	14175.23	806.86	5.80
16	Xylona Preis	28	84.81	2374.66	621.23	1.30
17	Clay Ludtke	28	388.59	10880.55	1933.78	3.20
18	Keith Dawkins	28	292.19	8181.26	3038.63	2.45
19	Chris Selesnick	28	98.36	2754.22	738.36	1.60
20	Kunst Miller	28	175.34	4909.47	745.77	2.50

Total rows: 297 of 297 Query complete 00:00:00.496 Ln 14, Col 16

THANK YOU.....



Yuvraj Giri