

# SECTION 2: Data Analytics

---

## Section Summary: Data Analytics (SQL in SSMS)

This section of your project demonstrates how SQL was used within Microsoft SQL Server Management Studio (SSMS) to analyze cleaned retail order data. The analysis explores revenue, profit, and sales performance across multiple dimensions including products, categories, regions, and time periods.

---

## Description of the Process

The analytics phase focused on extracting insights through SQL queries designed to answer key business questions, including:

1. Top Revenue Products – Identified the 10 highest revenue-generating SKUs and compared their total revenue, units sold, and profit margins to assess product performance balance between volume and profitability.
  2. Profitability Analysis – Compared the top 10 products by profit to highlight discrepancies between high sales and high margins, revealing strategic opportunities for pricing and promotion.
  3. Subcategory Performance – Compared revenue and profit across subcategories to find which product lines drive consistent profitability versus those that rely on discounts or volume.
  4. Temporal Trends – Conducted month-over-month and year-over-year comparisons (2022 vs. 2023) to observe sales growth, seasonality, and demand patterns across time.
  5. Regional Insights – Analyzed the top-performing products and categories by region to uncover geographic demand patterns and performance variation.
- 

## Findings Summary

- Product Insights:
    - TEC-CO-100004722 led in revenue (\$59,514) but had a low margin (~9.5%), indicating volume-driven rather than profit-driven success.
    - TEC-MA-10002412, though lower in revenue, achieved the highest margin (~16.7%), suggesting a premium product segment.
  - Category/Subcategory Insights:
    - Phones led total revenue, but Chairs and Binders had stronger profit efficiency.
    - Machines and Bookcases underperformed in margin, possibly due to high costs or discounts.
  - Regional & Temporal Trends:
    - Consistent monthly growth across regions from 2022 to 2023.
    - Region-specific top sellers suggest localized consumer behavior.
    - Seasonal peaks indicate potential for optimizing marketing and inventory during high-demand months.
-

# Query Execution & Insights

## Description:

This section presents the SQL queries executed in Microsoft SQL Server Management Studio (SSMS), along with screenshots of query results and corresponding insights. Each query addresses a specific business question, followed by a summary of key findings and interpretations based on the output.

### Total Sales Summary:

	total_sales_2022	total_sales_2023	pct_growth_22_23	total_sales_combined
1	1095590.20	1120268.50	2.25	2215858.70

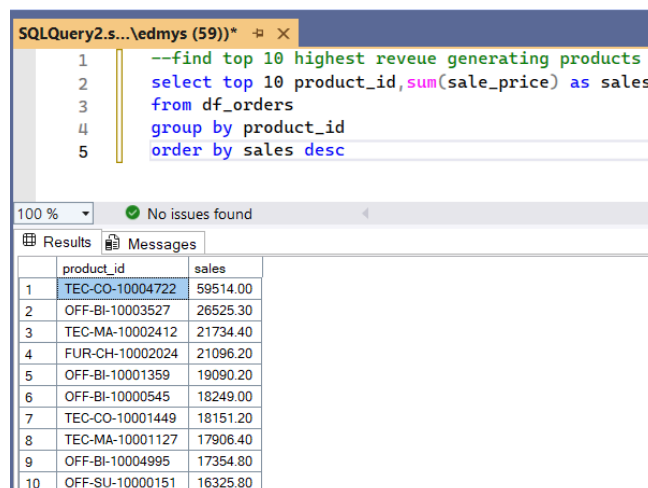
### Regional Sales Summary:

	region	sales_2022	sales_2023	pct_growth	total_sales
1	West	370693.80	329164.80	-11.20	699858.60
2	East	310009.20	344310.80	11.06	654320.00
3	Central	226980.20	256688.40	13.09	483668.60
4	South	187907.00	190104.50	1.17	378011.50

	region	market_share_2022	market_share_2023	pct_growth_22_23	combined_market_share
1	West	33.84	29.38	-4.46	31.58
2	East	28.30	30.73	2.43	29.53
3	Cent...	20.72	22.91	2.19	21.83
4	South	17.15	16.97	-0.18	17.06

## Leading Revenue and Profit

### Top 10 highest revenue generating products



The screenshot shows the SQL Server Management Studio interface. At the top, a query window titled 'SQLQuery2.s... \edmys (59))' contains the following SQL code:

```
--find top 10 highest reveue generating products
select top 10 product_id,sum(sale_price) as sales
from df_orders
group by product_id
order by sales desc
```

Below the query window, the 'Results' tab is active, displaying the output of the query. The results are as follows:

	product_id	sales
1	TEC-CO-10004722	59514.00
2	OFF-BI-10003527	26525.30
3	TEC-MA-10002412	21734.40
4	FUR-CH-10002024	21096.20
5	OFF-BI-10001359	19090.20
6	OFF-BI-10000545	18249.00
7	TEC-CO-10001449	18151.20
8	TEC-MA-10001127	17906.40
9	OFF-BI-10004995	17354.80
10	OFF-SU-10000151	16325.80

The result shows:

- The highest product is TEC-CO-100004722 with \$59,514 in sales.
- The remaining top products range roughly \$26.5k → \$16.3k.
- Several top entries share a prefix pattern (TEC-, OFF-, FUR-), suggesting different categories (e.g., Tech, Office, Furniture), useful for later slicing.

Quick analysis / takeaways:

- Concentration check: There's a steep drop from #1 (~\$59.5k) to #2 (~\$26.5k). That hints at a power-law/long-tail pattern where one SKU dominates (generate large %). Worth quantifying the % of total revenue these top 10 represent.

Follow up: Check for high concentration of single products.

```
--
79 -- Calculate revenue share of the top 10 products
80 WITH product_sales AS (
81     SELECT
82         product_id,
83         SUM(sale_price) AS total_sales
84     FROM df_orders
85     GROUP BY product_id
86 ),
87 ranked_products AS (
88     SELECT
89         product_id,
90         total_sales,
91         RANK() OVER (ORDER BY total_sales DESC) AS sales_rank
92     FROM product_sales
93 ),
94 summary AS (
95     SELECT
96         SUM(total_sales) AS total_revenue
97     FROM product_sales
98 )
99 SELECT
100     r.product_id,
101     r.total_sales,
102     ROUND(r.total_sales / s.total_revenue * 100, 2) AS pct_of_total_revenue
103 FROM ranked_products r
104 CROSS JOIN summary s
105 WHERE r.sales_rank <= 10
106 ORDER BY r.total_sales DESC;
```

	product_id	total_sales	pct_of_total_revenue
1	TEC-CO-10004722	59514.00	2.690000
2	OFF-BI-10003527	26525.30	1.200000
3	TEC-MA-10002412	21734.40	0.980000
4	FUR-CH-10002024	21096.20	0.950000
5	OFF-BI-10001359	19090.20	0.860000
6	OFF-BI-10000545	18249.00	0.820000
7	TEC-CO-10001449	18151.20	0.820000
8	TEC-MA-10001127	17906.40	0.810000
9	OFF-BI-10004995	17354.80	0.780000
10	OFF-SU-10000151	16325.80	0.740000

Analysis:

There is no evidence of a power-law (high concentration of product). Rather, percentages are spread more evenly across products.

Further Questions:

- SKU vs. variant risk: Using product\_id implies SKU-level aggregation. If variants exist (size/color), confirm that product\_id is the right grain.
  - Confirm that some products with different product\_id do not need to be grouped together as different variants of the same product.
- Revenue vs. units: A product can rank high from high price or high volume. Pair this with units\_sold to avoid misleading conclusions.
- Gross vs. net: If sale\_price is post-discount but pre-returns, consider excluding returns/cancellations or subtracting them to reflect net revenue.
  - Need information on returns/cancellations to calculate more accurate profit/revenue.
- Check patterns by year/season for trend analysis.

**Improved top 10 revenue query with useful context columns added:**

```

SQLQuery2...edmys (59)) * - x
114 | -- Top 10 revenue products with useful context (uses only columns in your table)
115 | SELECT TOP 10
116 |     product_id,
117 |     SUM(sale_price) AS revenue, -- total sales $
118 |     COUNT(*) AS orders, -- number of order lines
119 |     SUM(quantity) AS units, -- total units
120 |     SUM(ISNULL(discount, 0)) AS total_discount, -- $ discount (if any NULLs)
121 |     SUM(ISNULL(profit, 0)) AS profit, -- total profit $
122 |     CAST(SUM(sale_price) / NULLIF(SUM(quantity), 0) AS DECIMAL(18,2)) AS avg_selling_price, -- ASP
123 |     CAST(SUM(ISNULL(profit,0)) / NULLIF(SUM(sale_price),0) AS DECIMAL(18,4)) AS profit_margin -- % of revenue
124 | FROM df_orders
125 | GROUP BY product_id
126 | ORDER BY revenue DESC; -- or ORDER BY profit DESC if you want top by profit
127 |

```

This query groups all orders by product ID to summarize performance at the SKU level, calculating each product's total revenue, number of orders, total units sold, total discounts, total profit, average selling price, and profit margin, then sorts the results so the highest-revenue products appear first.

### Insights:

- Top Product Dominance:
  - TEC-CO-10004722 leads by a large margin: \$59,514 in sales vs. \$26,525 for #2.
  - → It contributes roughly 2.2x more revenue than the next product.
  - However, its profit margin (9.5%) is lower than several others.
  - It's a high-revenue but possibly low-margin driver, worth keeping for traffic but not as a profit engine.
- High-margin vs. high-volume contrast:
  - TEC-MA-10002412 (3rd place) has a 16.7% margin, the highest among the top 10.
  - It sold fewer units (6) but produced nearly \$21.7K revenue, meaning a high ASP (\$3,622).
  - These may be specialty or luxury items worth more focused marketing.

## Further Analysis: Comparing Profit and Revenue

### Top 10 products by highest profit.

SQLQuery2.s...edmys (59))

```
-- Top 10 most profitable products (uses only columns in your table)
SELECT TOP 10
    product_id,
    SUM(profit) AS total_profit,           -- total profit earned
    SUM(sale_price) AS revenue,           -- total revenue
    SUM(quantity) AS units_sold,         -- total units sold
    COUNT(*) AS total_orders,            -- total order lines
    SUM(ISNULL(discount,0)) AS total_discount, -- total discount amount
    CAST(SUM(profit) / NULLIF(SUM(sale_price),0) AS DECIMAL(18,4)) AS profit_margin, -- % margin
    CAST(SUM(sale_price) / NULLIF(SUM(quantity),0) AS DECIMAL(18,2)) AS avg_selling_price -- avg $
FROM df_orders
GROUP BY product_id
ORDER BY total_profit DESC;
```

100 % 7 0

	product_id	total_profit	revenue	units_sold	total_orders	total_discount	profit_margin	avg_selling_price
1	TEC-CO-10004722	5644.00	59514.00	20	5	2086.00	0.0948	2975.70
2	TEC-MA-10002412	3624.40	21734.40	6	1	905.60	0.1668	3622.40
3	OFF-BI-10003527	3435.30	26525.30	31	10	914.70	0.1295	855.65
4	TEC-CO-10001449	2631.20	18151.20	38	8	688.80	0.1450	477.66
5	FUR-CH-10002024	2246.20	21096.20	39	8	763.80	0.1065	540.93
6	OFF-BI-10001359	2080.20	19090.20	37	11	729.80	0.1090	515.95
7	OFF-BI-10000545	1959.00	18249.00	48	9	781.00	0.1073	380.19
8	OFF-BI-10001120	1695.70	15505.70	13	3	374.30	0.1094	1192.75
9	OFF-BI-10004995	1654.80	17354.80	27	6	615.20	0.0954	642.77
10	FUR-BO-10004834	1614.10	15024.10	24	5	585.90	0.1074	626.00

- High sales don't equal high profit — the top seller has one of the lowest margins.
- Premium items earn more profit despite lower sales volume.
- Office products drop in rank due to heavy discounting and thinner margins.
- Profit ranking reveals true value drivers, not just sales leaders.

### Top 10 Subcategories: Profit vs Revenue

SQLQuery2.s...edmys (59))

```
-- Top 10 sub-categories ranked by total profit
SELECT TOP 10
    sub_category,
    SUM(profit) AS total_profit,           -- total profit earned
    SUM(sale_price) AS total_revenue,     -- total revenue
    SUM(quantity) AS total_units,         -- total units sold
    COUNT(*) AS total_orders,            -- number of order lines
    SUM(ISNULL(discount,0)) AS total_discount, -- total discount
    CAST(SUM(profit) / NULLIF(SUM(sale_price),0) AS DECIMAL(18,4)) AS avg_profit_margin,
    CAST(SUM(sale_price) / NULLIF(SUM(quantity),0) AS DECIMAL(18,2)) AS avg_selling_price
FROM df_orders
GROUP BY sub_category
ORDER BY total_profit DESC;
```

100 % 7 0

	sub_category	total_profit	total_revenue	total_units	total_orders	total_discount	avg_profit_margin	avg_selling_price
1	Chairs	29815.10	316825.10	2356	617	11404.90	0.0941	134.48
2	Phones	28368.30	318008.30	3289	889	11911.70	0.0892	96.69
3	Storage	19538.00	215908.00	3158	846	7712.00	0.0905	68.37
4	Binders	19208.60	196038.60	5974	1523	6971.40	0.0980	32.82
5	Tables	18591.20	199361.20	1241	319	7488.80	0.0933	160.65
6	Machines	18121.70	182901.70	440	115	6358.30	0.0991	415.69
7	Accessories	15444.60	161604.60	2976	775	5905.40	0.0956	54.30
8	Copiers	14498.90	144358.90	234	68	5171.10	0.1004	616.92
9	Bookcases	10496.10	110816.10	868	228	3953.90	0.0947	127.67
10	Appliances	10267.70	103697.70	1729	466	3702.30	0.0990	59.98

SQLQuery2.s... \edmys (59))\*

```

159 -- Top 10 sub-categories ranked by total revenue
160 SELECT TOP 10
161     sub_category,
162     SUM(sale_price) AS total_revenue,           -- total sales $
163     SUM(profit) AS total_profit,               -- total profit $
164     SUM(quantity) AS total_units,              -- total units sold
165     COUNT(*) AS total_orders,                 -- number of orders
166     SUM(ISNULL(discount,0)) AS total_discount, -- total discount $
167     CAST(SUM(profit) / NULLIF(SUM(sale_price),0) AS DECIMAL(18,4)) AS avg_profit_margin, -- profit %
168     CAST(SUM(sale_price) / NULLIF(SUM(quantity),0) AS DECIMAL(18,2)) AS avg_selling_price -- avg $
169 FROM df_orders
170 GROUP BY sub_category
171 ORDER BY total_revenue DESC;
172

```

100 % 7 0

Results Messages

	sub_category	total_revenue	total_profit	total_units	total_orders	total_discount	avg_profit_margin	avg_selling_price
1	Phones	318008.30	28368.30	3289	889	11911.70	0.0892	96.69
2	Chairs	316825.10	29815.10	2356	617	11404.90	0.0941	134.48
3	Storage	215908.00	19538.00	3158	846	7712.00	0.0905	68.37
4	Tables	199361.20	18591.20	1241	319	7488.80	0.0933	160.65
5	Binders	199038.60	19208.60	5974	1523	6971.40	0.0980	32.82
6	Machines	182901.70	18121.70	440	115	6358.30	0.0991	415.69
7	Accessories	161604.60	15444.60	2976	775	5905.40	0.0956	54.30
8	Copiers	144358.90	14498.90	234	68	5171.10	0.1004	616.92
9	Bookcases	110816.10	10496.10	868	228	3953.90	0.0947	127.67
10	Appliances	103697.70	10267.70	1729	466	3702.30	0.0990	59.98

## Take Aways:

- Revenue leaders aren't always profit leaders: Phones generated the highest sales volume, but Chairs produced the most profit due to stronger margins.
- Chairs and Binders are the most efficient categories, delivering higher profit per sale and indicating strong pricing or lower production costs.
- Machines and Bookcases show weak profit efficiency, suggesting high costs or aggressive discounting that erode margins.
- Mid-tier categories like Storage and Accessories provide balanced, reliable performance with steady revenue and moderate margins.
- Focusing on margin optimization and product mix strategy could significantly increase total profitability without needing to grow total sales volume.

## Month and Year Revenue Comparison

Month over month growth comparison for 2022 and 2023 sales eg : jan 2022 vs jan 2023

SQLQuery2.s... \edmys (59))\*

```

20 --find month over month growth comparison for 2022 and 2023 sales eg : jan 2022 vs jan 2023
21 with cte as (
22     select year(order_date) as order_year, month(order_date) as order_month,
23     sum(sale_price) as sales
24     from df_orders
25     group by year(order_date), month(order_date)
26     --order by year(order_date), month(order_date)
27 )
28 select order_month
29     , sum(case when order_year=2022 then sales else 0 end) as sales_2022
30     , sum(case when order_year=2023 then sales else 0 end) as sales_2023
31 from cte
32 group by order_month
33 order by order_month

```

100 % No issues found Ln: 34

Results Messages

	order_month	sales_2022	sales_2023
1	1	94712.50	88632.60
2	2	90091.00	128124.20
3	3	80106.00	82512.30
4	4	95451.60	111568.60
5	5	79448.30	86447.90
6	6	94170.50	68976.50
7	7	78652.20	90563.80
8	8	104808.00	87733.60
9	9	79142.20	76658.60
10	10	118912.70	121061.50
11	11	84225.30	75432.80
12	12	95869.90	102556.10

Query executed successfully. localhost\SQLEXPRESS (16.0 ... LAPTOP-MG9Q3IIE\edmys ... e2e\_data\_project

For each category which month had highest sales, Order by category , format (order\_date,'yyyyMM')

SQLQuery2.s...edmys (59))

```

36 --for each category which month had highest sales
37 with cte as (
38     select category,format(order_date,'yyyyMM') as order_year_month
39         , sum(sale_price) as sales
40     from df_orders
41     group by category,format(order_date,'yyyyMM')
42     --order by category,format(order_date,'yyyyMM')
43 )
44 select * from (
45     select *,
46     row_number() over(partition by category order by sales desc) as rn
47 from cte
48 ) a
49 where rn=1

```

100 %

No issues found

Results

Messages

	category	order_year_month	sales	rn
1	Furniture	202210	42888.90	1
2	Office Supplies	202302	44118.50	1
3	Technology	202310	53000.10	1

Which sub category had highest growth by profit in 2023 compare to 2022

SQLQuery2.s...edmys (59))\*

```
55
56 --which sub category had highest growth by profit in 2023 compare to 2022
57 with cte as (
58     select sub_category,year(order_date) as order_year,
59     sum(sale_price) as sales
60     from df_orders
61     group by sub_category,year(order_date)
62     --order by year(order_date),month(order_date)
63 )
64 , cte2 as (
65     select sub_category
66     , sum(case when order_year=2022 then sales else 0 end) as sales_2022
67     , sum(case when order_year=2023 then sales else 0 end) as sales_2023
68     from cte
69     group by sub_category
70 )
71     select top 1 *
72     ,(sales_2023-sales_2022)
73     from cte2
74     order by (sales_2023-sales_2022) desc
```

100 %

No issues found

Results

Messages

	sub_category	sales_2022	sales_2023	(No column name)
1	Machines	73723.20	109178.50	35455.30

# Regional Sales Analysis:

## Top 5 highest selling products in each region

SQLQuery2.s...edmys (59))\*

7

--find top 5 highest selling products in each region

8

with cte as (

9

select region,product\_id,sum(sale\_price) as sales

10

from df\_orders

11

group by region,product\_id)

12

select \* from (

13

select \*

14

, row\_number() over(partition by region order by sales desc) as rn

15

from cte) A

16

where rn<=5

100 %

No issues found

Results

Messages

	region	product_id	sales	m
1	Central	TEC-CO-10004722	16975.00	1
2	Central	TEC-MA-10000822	13770.00	2
3	Central	OFF-BI-10001120	11056.50	3
4	Central	OFF-BI-10000545	10132.70	4
5	Central	OFF-BI-10004995	8416.10	5
6	East	TEC-CO-10004722	29099.00	1
7	East	TEC-MA-10001047	13767.00	2
8	East	FUR-BO-10004834	11274.10	3
9	East	OFF-BI-10001359	8463.60	4
10	East	TEC-CO-10001449	8316.00	5
11	South	TEC-MA-10002412	21734.40	1
12	South	TEC-MA-10001127	11116.40	2
13	South	OFF-BI-10001359	8053.20	3
14	South	TEC-MA-10004125	7840.00	4
15	South	OFF-BI-10003527	7391.40	5
16	West	TEC-CO-10004722	13440.00	1
17	West	OFF-SU-10000151	12592.30	2
18	West	FUR-CH-10001215	9604.00	3
19	West	OFF-BI-10003527	7804.80	4
20	West	TEC-AC-10003832	7722.70	5

## Top categories per region with metrics

```
-- Top 5 categories per region with performance metrics
WITH category_region_sales AS (
    SELECT
        region,
        category,
        SUM(sale_price) AS total_revenue,
        SUM(ISNULL(discount,0)) AS total_discount,
        SUM(ISNULL(profit,0)) AS total_profit,
        SUM(quantity) AS total_units,
        COUNT(*) AS total_orders,
        CAST(SUM(sale_price) / NULLIF(SUM(quantity),0) AS DECIMAL(10,2)) AS avg_selling_price
    FROM df_orders
    GROUP BY region, category
),
ranked AS (
    SELECT
        *,
        RANK() OVER (PARTITION BY region ORDER BY total_revenue DESC) AS rank_by_revenue
    FROM category_region_sales
)
SELECT
    region,
    category,
    total_revenue,
    total_discount,
    total_profit,
    avg_selling_price,
    total_units,
    total_orders
FROM ranked
WHERE rank_by_revenue <= 5
ORDER BY region, total_revenue DESC;
```

	region	category	total_revenue	total_discount	total_profit	avg_selling_price	total_units	total_orders
1	Central	Technology	164552.60	5877.40	15872.60	106.58	1544	420
2	Central	Office Supplies	161077.50	5832.50	14507.50	29.78	5409	1422
3	Central	Furniture	158038.50	5691.50	15998.50	86.50	1827	481
4	East	Technology	255581.80	9368.20	23881.80	131.61	1942	535
5	East	Furniture	200589.60	7480.40	17549.60	90.60	2214	601
6	East	Office Supplies	198148.60	7211.40	16788.60	30.66	6462	1712
7	South	Technology	143606.40	5153.60	14706.40	128.45	1118	293
8	South	Office Supplies	121263.90	4276.10	12233.90	31.91	3800	995
9	South	Furniture	113141.20	4078.80	11071.20	87.64	1291	332
10	West	Furniture	243571.40	8798.60	21861.40	90.35	2696	707
11	West	Technology	243132.70	8947.30	21972.70	104.13	2335	599
12	West	Office Supplies	213154.50	7615.50	18724.50	29.46	7235	1897

## Top selling products monthly per region:

```
-- Swap this to each of your top 3 in turn:
-- 'TEC-CO-10004722', 'OFF-BI-10003527', 'TEC-MA-10002412'
DECLARE @product_id VARCHAR(50) = 'OFF-BI-10003527';

WITH monthly AS (
    SELECT
        DATEFROMPARTS(YEAR(order_date), MONTH(order_date), 1) AS month_start,
        SUM(sale_price) AS revenue,
        SUM(quantity) AS units,
        COUNT(*) AS orders,
        CAST(SUM(sale_price) / NULLIF(SUM(quantity),0) AS DECIMAL(10,2)) AS avg_selling_price,
        SUM(ISNULL(discount,0)) AS total_discount,
        SUM(ISNULL(profit,0)) AS total_profit
    FROM df_orders
    WHERE product_id = @product_id
    AND YEAR(order_date) IN (2022, 2023)
    GROUP BY DATEFROMPARTS(YEAR(order_date), MONTH(order_date), 1)
)
SELECT
    month_start,
    revenue, units, orders, avg_selling_price, total_discount, total_profit
FROM monthly
ORDER BY month_start;
```

```
-- Swap this to each of your top 3 in turn:
-- 'TEC-CO-10004722', 'OFF-BI-10003527', 'TEC-MA-10002412'
DECLARE @product_id VARCHAR(50) = 'TEC-AC-10003832';

WITH base AS (
    SELECT
        DATEFROMPARTS(YEAR(order_date), MONTH(order_date), 1) AS month_start,
        region,
        SUM(sale_price) AS revenue
    FROM df_orders
    WHERE product_id = @product_id
    AND YEAR(order_date) IN (2022, 2023)
    GROUP BY DATEFROMPARTS(YEAR(order_date), MONTH(order_date), 1), region
)
SELECT
    month_start,
    ISNULL([Central],0) AS Central_rev,
    ISNULL([East],0) AS East_rev,
    ISNULL([South],0) AS South_rev,
    ISNULL([West],0) AS West_rev,
    (ISNULL([Central],0)+ISNULL([East],0)+ISNULL([South],0)+ISNULL([West],0)) AS total_rev
FROM base
PIVOT (SUM(revenue) FOR region IN ([Central],[East],[South],[West])) p
ORDER BY month_start;
```

```
-- Swap this to each of your top 3 in turn:
-- 'TEC-CO-10004722', 'OFF-BI-10003527', 'TEC-MA-10002412'
DECLARE @product_id VARCHAR(50) = 'OFF-BI-10003527';

SELECT
    DATEFROMPARTS(YEAR(order_date), MONTH(order_date), 1) AS month_start,
    region,
    SUM(sale_price) AS revenue,
    SUM(quantity) AS units,
    COUNT(*) AS orders
FROM df_orders
WHERE product_id = @product_id
AND YEAR(order_date) IN (2022, 2023)
GROUP BY DATEFROMPARTS(YEAR(order_date), MONTH(order_date), 1), region
ORDER BY month_start, revenue DESC;
```

- #1 Total Sales by month / Region / Units & Orders



	month_start	revenue	units	orders	avg_selling_price	total_discount	total_profit
1	2022-08-01	16975.00	5	1	3395.00	525.00	2975.00
2	2022-10-01	13440.00	4	1	3360.00	560.00	420.00
3	2022-11-01	9975.00	3	1	3325.00	525.00	735.00
4	2023-04-01	10976.00	4	1	2744.00	224.00	676.00
5	2023-06-01	8148.00	4	1	2037.00	252.00	838.00

	month_start	Central_rev	East_rev	South_rev	West_rev	total_rev
1	2022-08-01	16975.00	0.00	0.00	0.00	16975.00
2	2022-10-01	0.00	0.00	0.00	13440.00	13440.00
3	2022-11-01	0.00	9975.00	0.00	0.00	9975.00
4	2023-04-01	0.00	10976.00	0.00	0.00	10976.00
5	2023-06-01	0.00	8148.00	0.00	0.00	8148.00

	month_start	region	revenue	units	orders
1	2022-08-01	Central	16975.00	5	1
2	2022-10-01	West	13440.00	4	1
3	2022-11-01	East	9975.00	3	1
4	2023-04-01	East	10976.00	4	1
5	2023-06-01	East	8148.00	4	1

- #2 Total Sales by Month / Region / Units & Orders

	month_start	revenue	units	orders	avg_selling_price	total_discount	total_profit
1	2022-01-01	4826.00	5	1	965.20	254.00	766.00
2	2022-02-01	2989.00	3	1	996.33	61.00	309.00
3	2022-06-01	737.20	3	1	245.73	22.80	57.20
4	2022-12-01	989.40	1	1	989.40	30.60	169.40
5	2023-06-01	1468.80	6	1	244.80	61.20	48.80
6	2023-07-01	1989.40	2	1	994.70	40.60	349.40
7	2023-10-01	3670.30	3	2	1223.43	139.70	250.30
8	2023-12-01	9855.20	8	2	1231.90	304.80	1485.20

	month_start	Central_rev	East_rev	South_rev	West_rev	total_rev
1	2022-01-01	0.00	0.00	0.00	4826.00	4826.00
2	2022-02-01	0.00	2989.00	0.00	0.00	2989.00
3	2022-06-01	737.20	0.00	0.00	0.00	737.20
4	2022-12-01	0.00	0.00	0.00	989.40	989.40
5	2023-06-01	1468.80	0.00	0.00	0.00	1468.80
6	2023-07-01	0.00	0.00	0.00	1989.40	1989.40
7	2023-10-01	0.00	2438.40	1231.90	0.00	3670.30
8	2023-12-01	3695.70	0.00	6159.50	0.00	9855.20

	month_start	region	revenue	units	orders
1	2022-01-01	West	4826.00	5	1
2	2022-02-01	East	2989.00	3	1
3	2022-06-01	Central	737.20	3	1
4	2022-12-01	West	989.40	1	1
5	2023-06-01	Central	1468.80	6	1
6	2023-07-01	West	1989.40	2	1
7	2023-10-01	East	2438.40	2	1
8	2023-10-01	South	1231.90	1	1
9	2023-12-01	South	6159.50	5	1
10	2023-12-01	Central	3695.70	3	1

## Monthly sales per region

```
-- Monthly revenue pivoted by region (2022-2023)
WITH monthly AS (
  SELECT
    FORMAT(order_date, 'yyyy-MM') AS month, -- year-month format
    region,
    SUM(sale_price) AS total_revenue
  FROM df_orders
  WHERE YEAR(order_date) IN (2022, 2023)
  GROUP BY FORMAT(order_date, 'yyyy-MM'), region
)
SELECT
  month,
  ISNULL([Central], 0) AS Central,
  ISNULL([East], 0) AS East,
  ISNULL([South], 0) AS South,
  ISNULL([West], 0) AS West,
  (ISNULL([Central],0)+ISNULL([East],0)+ISNULL([South],0)+ISNULL([West],0)) AS Total
FROM monthly
PIVOT (
  SUM(total_revenue)
  FOR region IN ([Central],[East],[South],[West])
) AS p
ORDER BY month;
```

	month	Central	East	South	West	Total
1	2022-01	14668.80	28539.80	12328.40	39175.50	94712.50
2	2022-02	14206.50	38166.80	15928.80	21788.90	90091.00
3	2022-03	15099.40	27888.90	16189.60	20928.10	80106.00
4	2022-04	17639.10	23077.50	13387.90	41347.10	95451.60
5	2022-05	11312.00	20401.30	18014.20	29720.80	79448.30
6	2022-06	18839.00	31624.00	21300.90	22406.60	94170.50
7	2022-07	18538.10	20566.70	10432.50	29114.90	78652.20
8	2022-08	36161.00	28130.40	14271.20	26245.40	104808.00
9	2022-09	16255.60	18395.40	17231.90	27259.30	79142.20
10	2022-10	31812.70	21327.30	15753.00	50019.70	118912.70
11	2022-11	9316.60	30162.30	16109.50	28636.90	84225.30
12	2022-12	23131.40	21728.80	16959.10	34050.60	95869.90
13	2023-01	15843.20	29392.80	12876.20	30520.40	88632.60
14	2023-02	38276.10	35840.50	30086.20	23921.40	128124.20
15	2023-03	24758.70	16721.60	15684.10	25347.90	82512.30
16	2023-04	20506.80	43647.20	10965.10	36449.50	111568.60
17	2023-05	20210.10	17423.20	12405.00	36409.60	86447.90
18	2023-06	14135.50	24814.70	8647.40	21378.90	68976.50
19	2023-07	23821.40	30985.80	11167.90	24588.70	90563.80
20	2023-08	14477.20	34972.50	10767.90	27516.00	87733.60
21	2023-09	15455.30	19444.30	14857.90	26901.10	76658.60
22	2023-10	25322.30	36883.20	34703.60	24152.40	121061.50
23	2023-11	14593.90	23170.60	11300.30	26368.00	75432.80
24	2023-12	29287.90	31014.40	16642.90	25610.90	102556.10