LP

SALES REPORT

FY2022-2023

Analytics Team
Lakshmi Priya Diwakar
Data Analyst

Objective

Key Goals:

- Extract and clean retail orders data.
- Transform data using Pandas.
- Load data into PostgreSQL database.
- Write SQL queries to answer ad-hoc business requests.

Data Cleaning and Connecting to our database

Data Source:

- Retail orders dataset for FY2022-2023 from Client A.

Cleaning Steps:

- Renaming Columns: Ensured consistency.
- Creating Calculated Columns: Added discount, profit, sales price.

Database Design:

- Data Type Conversion: Ensured compatibility with SQL types.
- Index Creation: Improved query performance.

Connecting Python to PostgreSQL Db

Tools and Libraries:

- Pandas for data transformation.
- SQLAlchemy for database connection.
- PostgreSQL for data storage and querying.

Process:

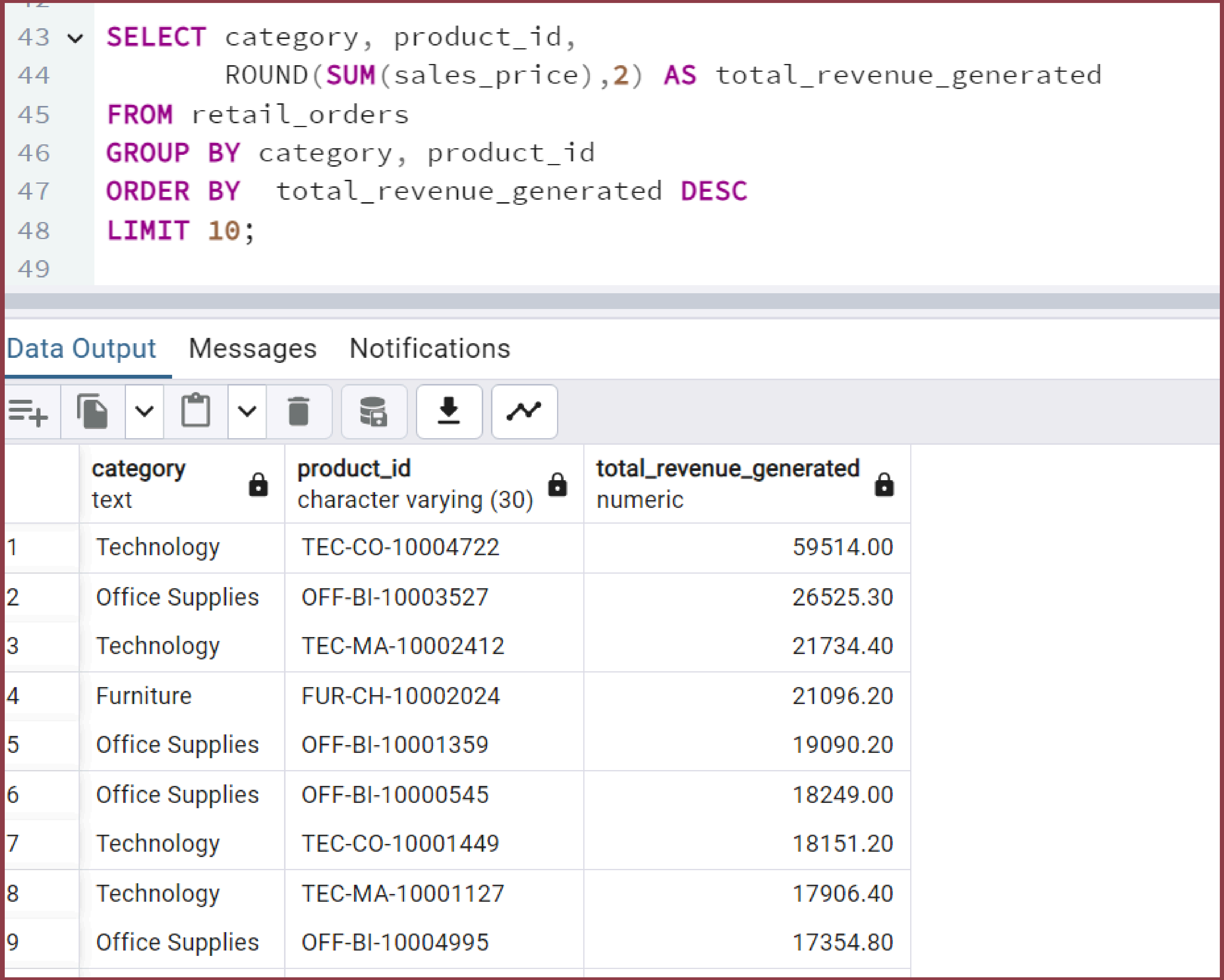
- Load cleaned data into PostgreSQL.
- Optimize database for performance.

Business Questions

Business Questions Overview

- Highest Revenue Products (2022 and 2023)
- Top 5 Highest Selling Products by Region (2022 and 2023)
- Monthly Sales Growth Comparison (2022 vs 2023)
- Highest Selling Months for Each Product Category
- Dominant Sales Category by Month and Year
- Highest Profit Growth Sub-category (2022 vs 2023)
- Highest Sales Growth Percentage Sub-category (2022 vs 2023)

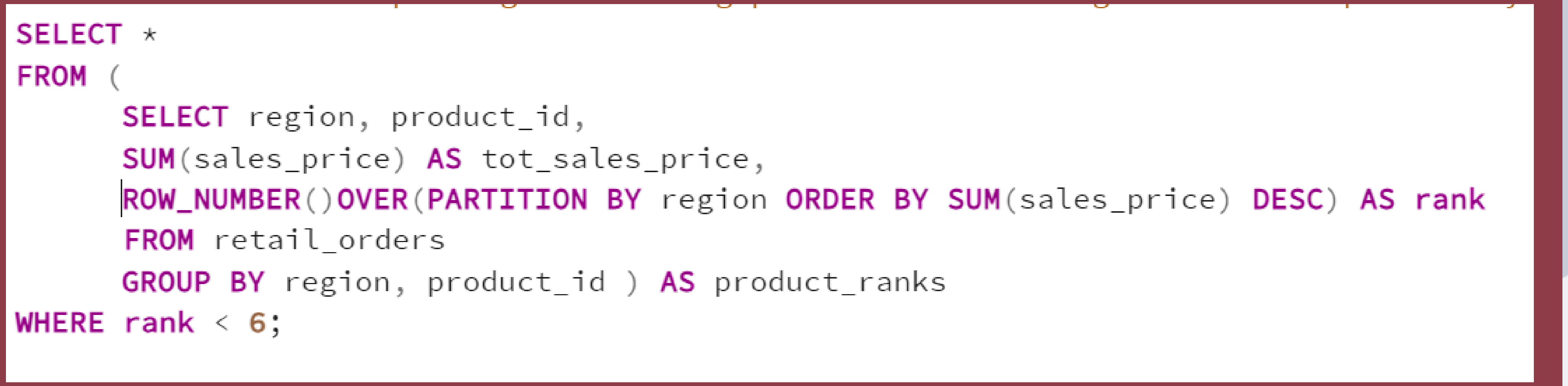
Q1. Which products have generated the highest revenue in the past two years (2022 and 2023)



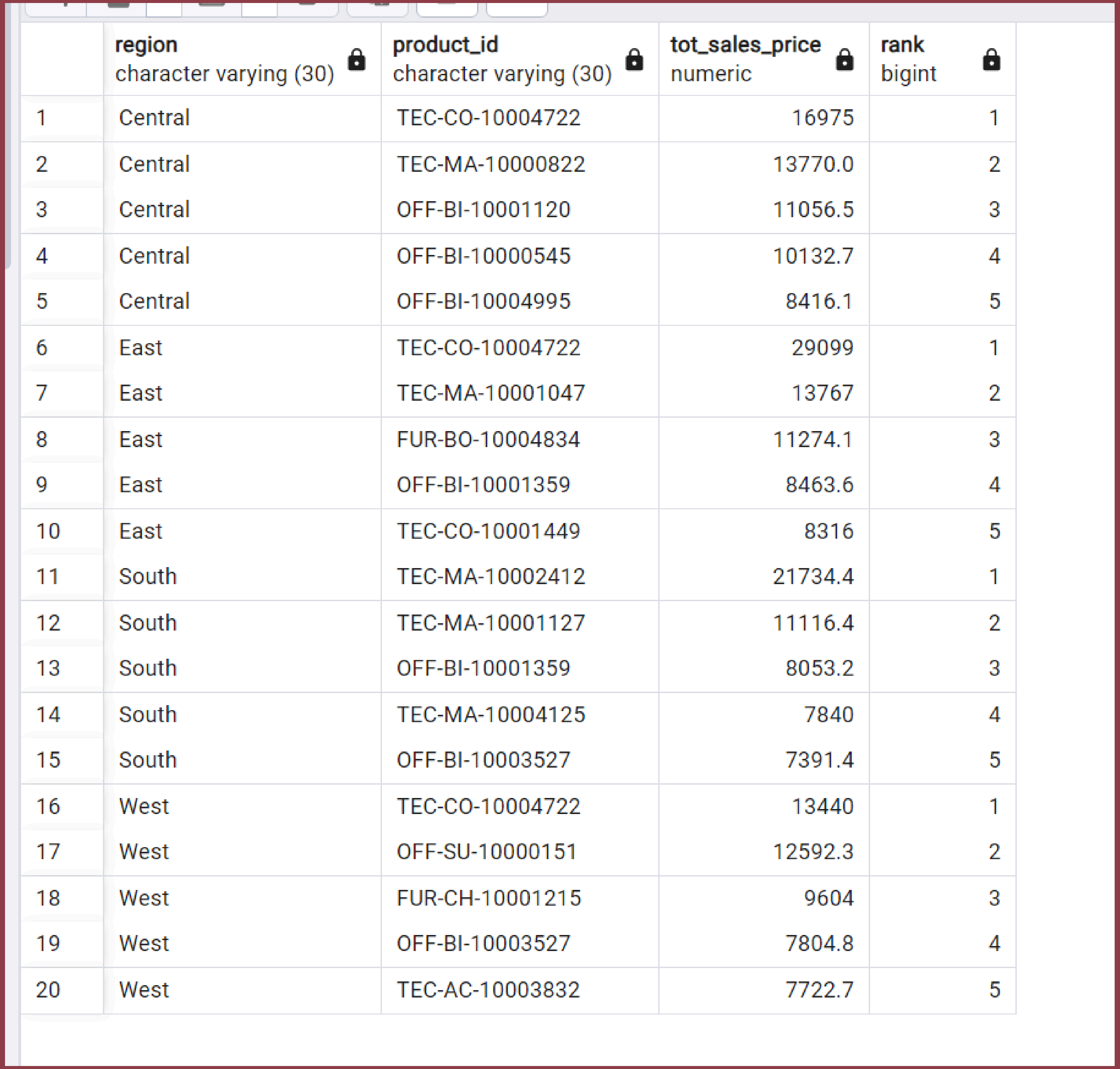
```sql
SELECT category, product_id,
       ROUND(SUM(sales_price),2) AS total_revenue_generated
FROM retail_orders
GROUP BY category, product_id
ORDER BY  total_revenue_generated DESC
LIMIT 10;
```

	category text	product_id character varying (30)	total_revenue_generated numeric
1	Technology	TEC-CO-10004722	59514.00
2	Office Supplies	OFF-BI-10003527	26525.30
3	Technology	TEC-MA-10002412	21734.40
4	Furniture	FUR-CH-10002024	21096.20
5	Office Supplies	OFF-BI-10001359	19090.20
6	Office Supplies	OFF-BI-10000545	18249.00
7	Technology	TEC-CO-10001449	18151.20
8	Technology	TEC-MA-10001127	17906.40
9	Office Supplies	OFF-BI-10004995	17354.80

Q2. What are the top 5 highest selling products in each region over the past two years

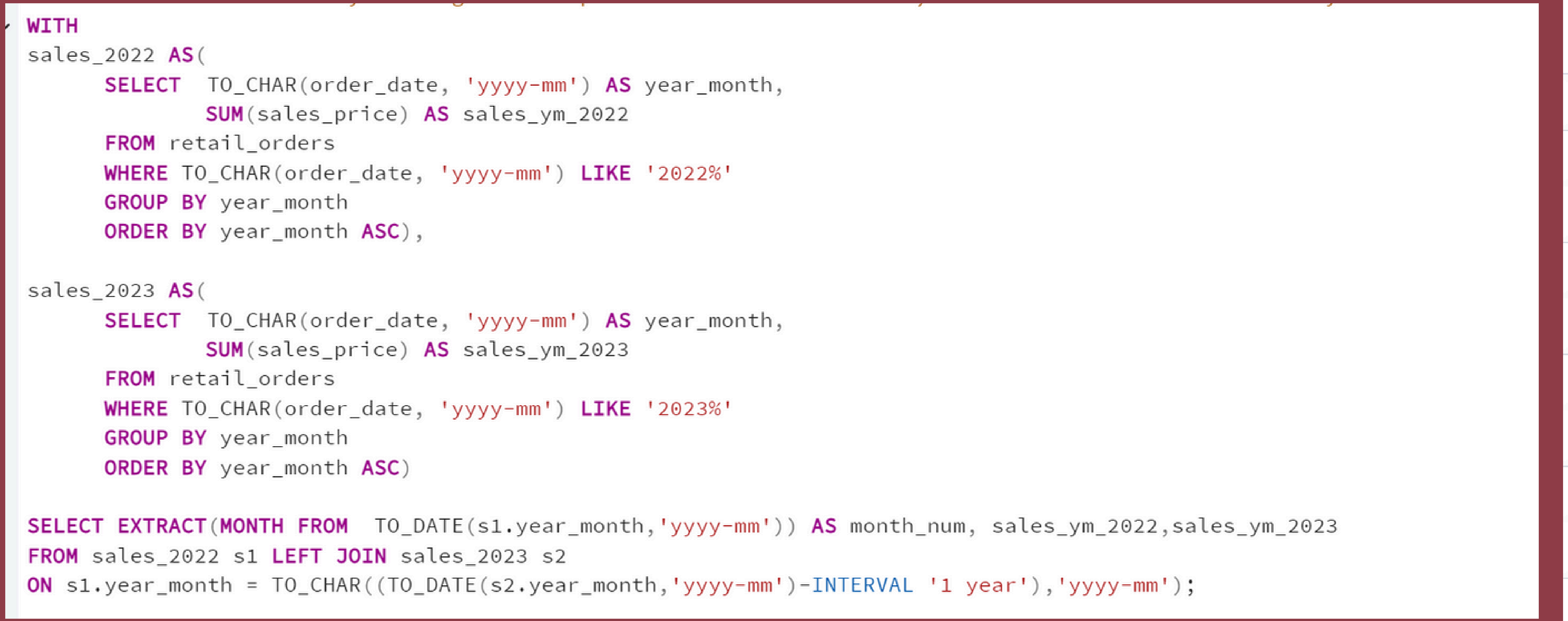


```
SELECT *
FROM (
      SELECT region, product_id,
      SUM(sales_price) AS tot_sales_price,
      ROW_NUMBER()OVER(PARTITION BY region ORDER BY SUM(sales_price) DESC) AS rank
      FROM retail_orders
      GROUP BY region, product_id ) AS product_ranks
WHERE rank < 6;
```

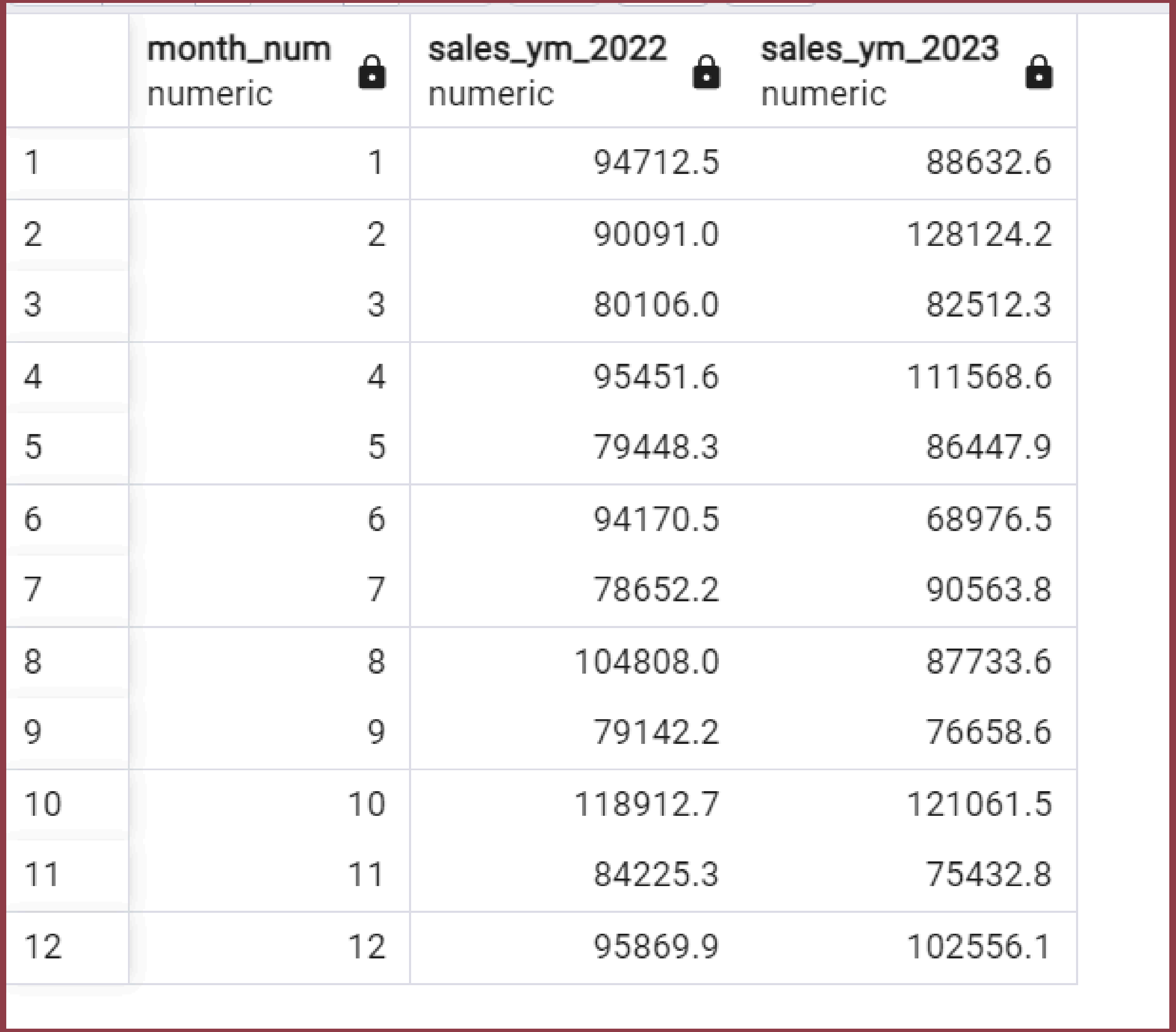


	region character varying (30)	product_id character varying (30)	tot_sales_price numeric	rank bigint
1	Central	TEC-CO-10004722	16975	1
2	Central	TEC-MA-10000822	13770.0	2
3	Central	OFF-BI-10001120	11056.5	3
4	Central	OFF-BI-10000545	10132.7	4
5	Central	OFF-BI-10004995	8416.1	5
6	East	TEC-CO-10004722	29099	1
7	East	TEC-MA-10001047	13767	2
8	East	FUR-BO-10004834	11274.1	3
9	East	OFF-BI-10001359	8463.6	4
10	East	TEC-CO-10001449	8316	5
11	South	TEC-MA-10002412	21734.4	1
12	South	TEC-MA-10001127	11116.4	2
13	South	OFF-BI-10001359	8053.2	3
14	South	TEC-MA-10004125	7840	4
15	South	OFF-BI-10003527	7391.4	5
16	West	TEC-CO-10004722	13440	1
17	West	OFF-SU-10000151	12592.3	2
18	West	FUR-CH-10001215	9604	3
19	West	OFF-BI-10003527	7804.8	4
20	West	TEC-AC-10003832	7722.7	5

Q3. How does our monthly overall sales growth compare between 2022 and 2023

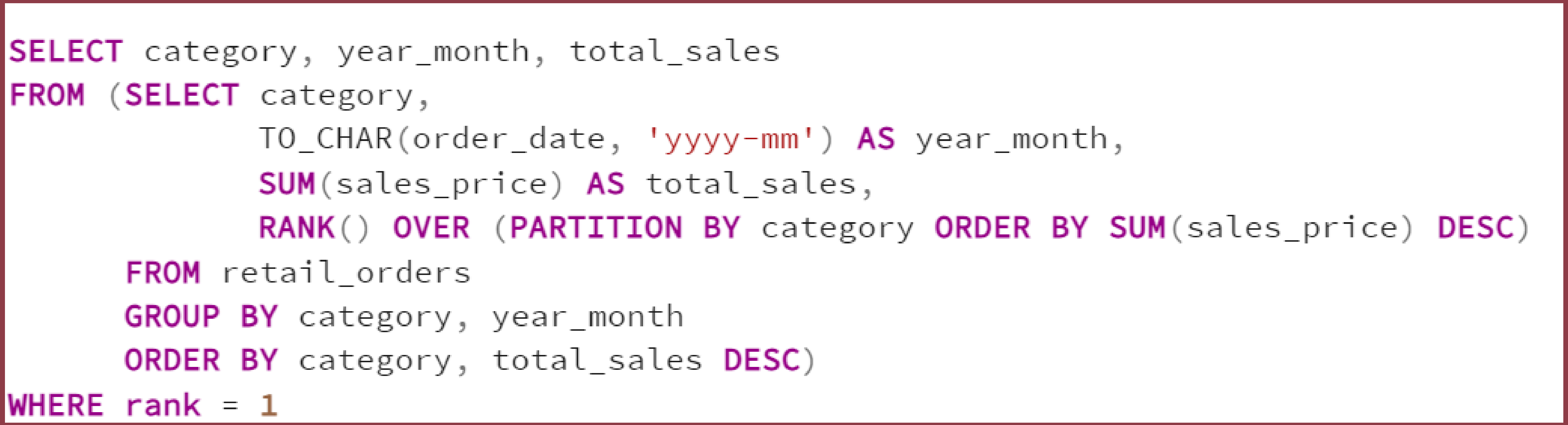


```
WITH
sales_2022 AS(
      SELECT  TO_CHAR(order_date, 'yyyy-mm') AS year_month,
              SUM(sales_price) AS sales_ym_2022
      FROM retail_orders
      WHERE TO_CHAR(order_date, 'yyyy-mm') LIKE '2022%'
      GROUP BY year_month
      ORDER BY year_month ASC),

sales_2023 AS(
      SELECT  TO_CHAR(order_date, 'yyyy-mm') AS year_month,
              SUM(sales_price) AS sales_ym_2023
      FROM retail_orders
      WHERE TO_CHAR(order_date, 'yyyy-mm') LIKE '2023%'
      GROUP BY year_month
      ORDER BY year_month ASC)

SELECT EXTRACT(MONTH FROM  TO_DATE(s1.year_month,'yyyy-mm')) AS month_num, sales_ym_2022,sales_ym_2023
FROM sales_2022 s1 LEFT JOIN sales_2023 s2
ON s1.year_month = TO_CHAR((TO_DATE(s2.year_month,'yyyy-mm')-INTERVAL '1 year'),'yyyy-mm');
```

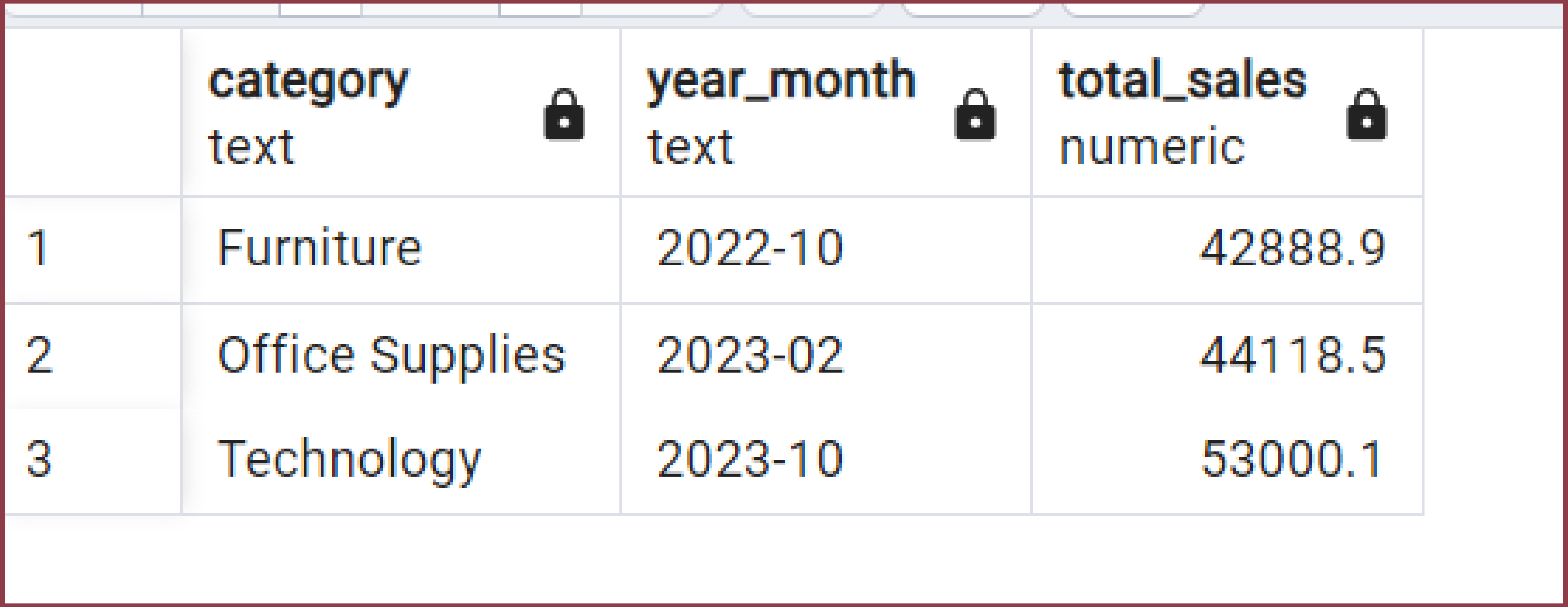


	month_num numeric	sales_ym_2022 numeric	sales_ym_2023 numeric
1	1	94712.5	88632.6
2	2	90091.0	128124.2
3	3	80106.0	82512.3
4	4	95451.6	111568.6
5	5	79448.3	86447.9
6	6	94170.5	68976.5
7	7	78652.2	90563.8
8	8	104808.0	87733.6
9	9	79142.2	76658.6
10	10	118912.7	121061.5
11	11	84225.3	75432.8
12	12	95869.9	102556.1

Q4. What are the highest selling months for each product category

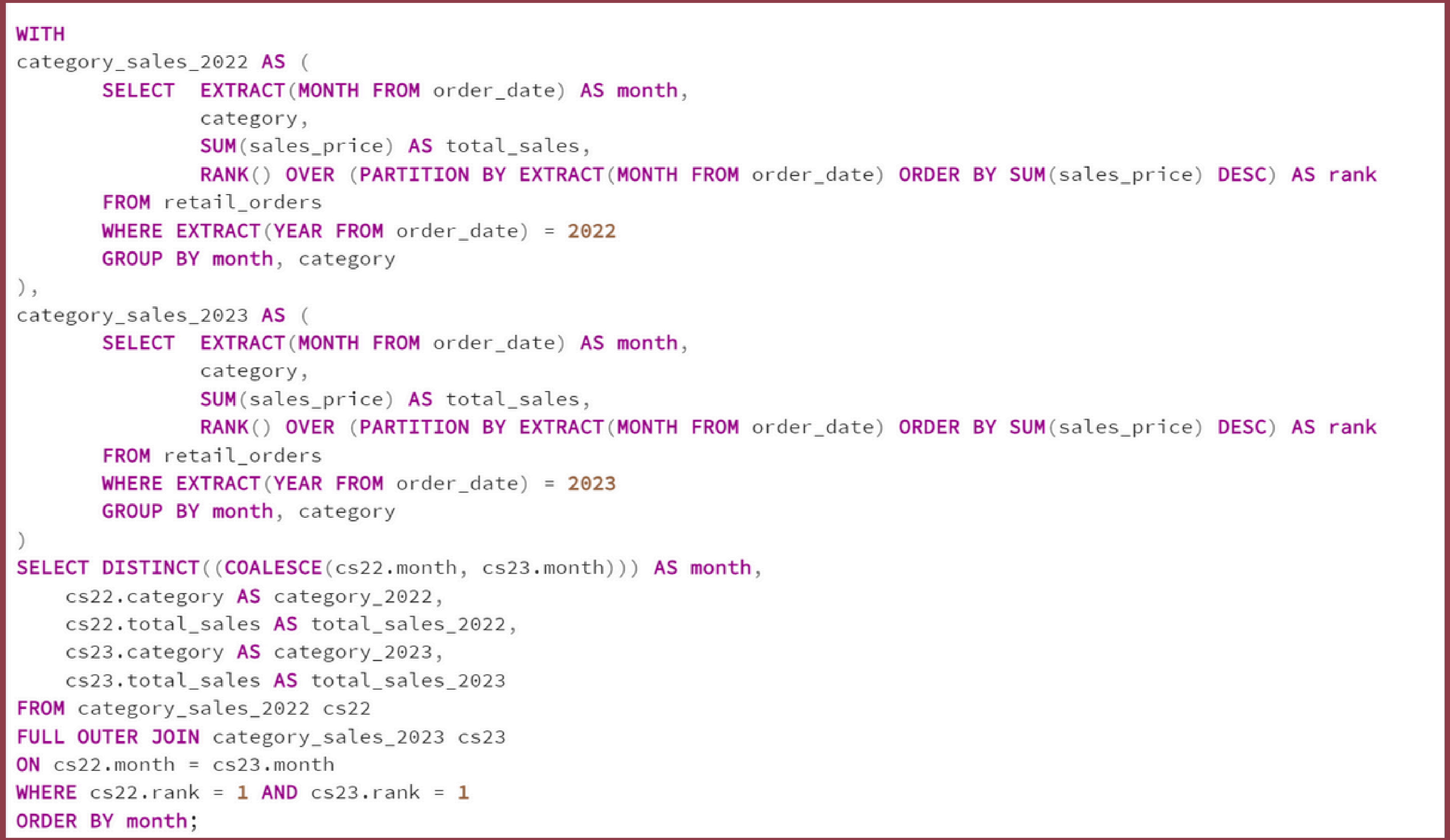


```
SELECT category, year_month, total_sales
FROM (SELECT category,
             TO_CHAR(order_date, 'yyyy-mm') AS year_month,
             SUM(sales_price) AS total_sales,
             RANK() OVER (PARTITION BY category ORDER BY SUM(sales_price) DESC)
      FROM retail_orders
      GROUP BY category, year_month
      ORDER BY category, total_sales DESC)
WHERE rank = 1
```

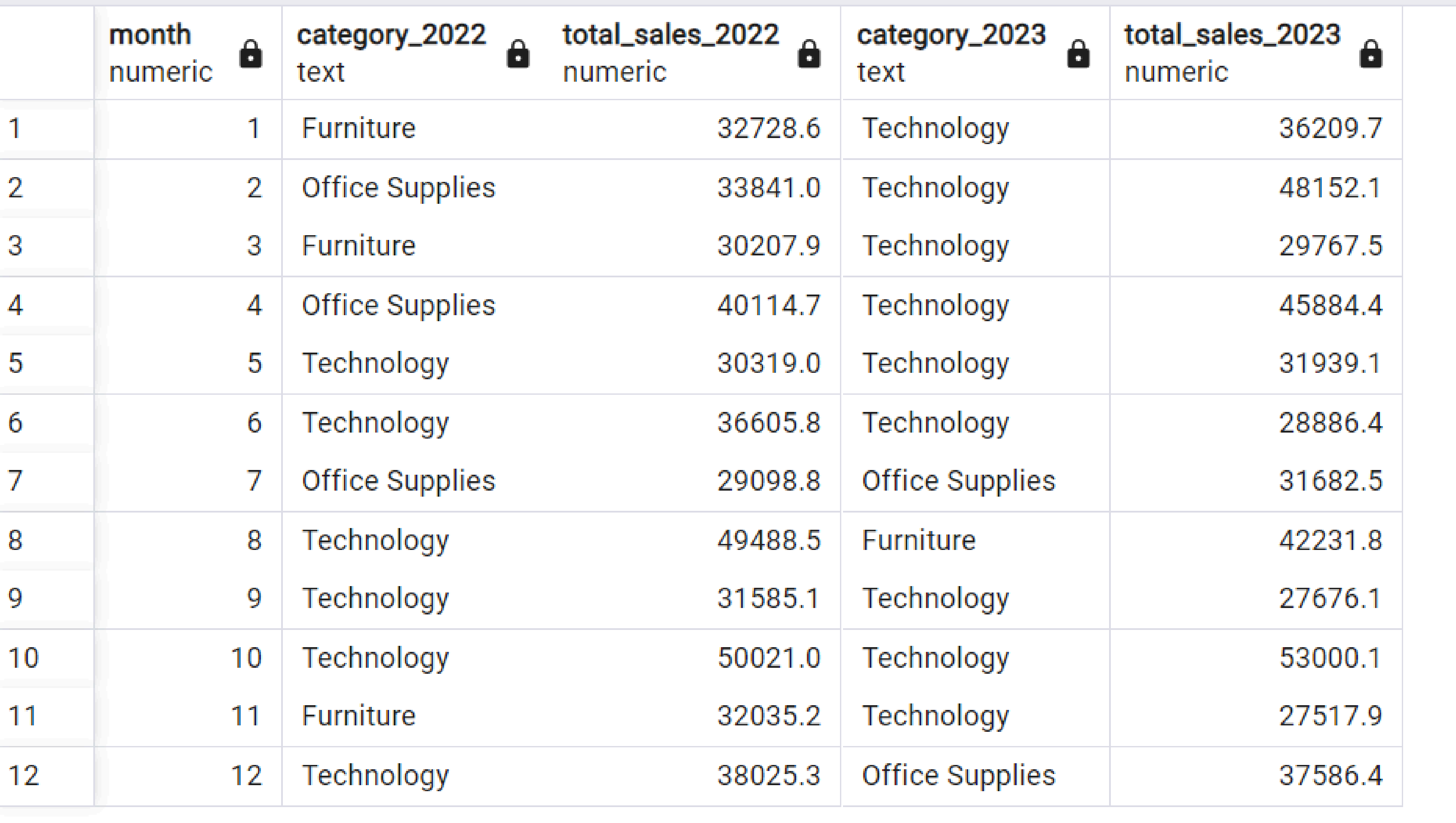


	category text	year_month text	total_sales numeric
1	Furniture	2022-10	42888.9
2	Office Supplies	2023-02	44118.5
3	Technology	2023-10	53000.1

Q5. Which category dominated sales for each month in each year

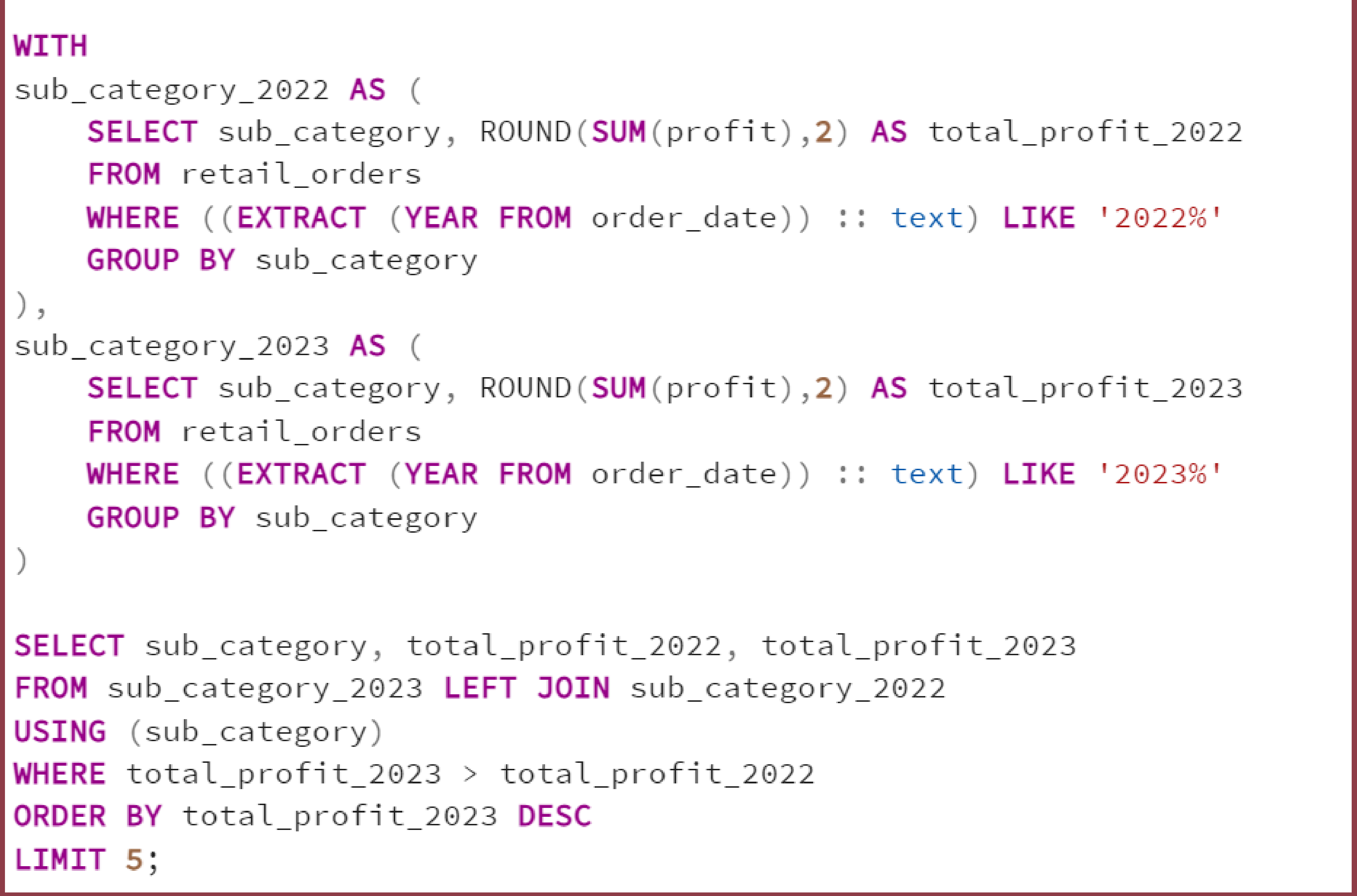


```sql
WITH
category_sales_2022 AS (
       SELECT  EXTRACT(MONTH FROM order_date) AS month,
               category,
               SUM(sales_price) AS total_sales,
               RANK() OVER (PARTITION BY EXTRACT(MONTH FROM order_date) ORDER BY SUM(sales_price) DESC) AS rank
       FROM retail_orders
       WHERE EXTRACT(YEAR FROM order_date) = 2022
       GROUP BY month, category
),
category_sales_2023 AS (
       SELECT  EXTRACT(MONTH FROM order_date) AS month,
               category,
               SUM(sales_price) AS total_sales,
               RANK() OVER (PARTITION BY EXTRACT(MONTH FROM order_date) ORDER BY SUM(sales_price) DESC) AS rank
       FROM retail_orders
       WHERE EXTRACT(YEAR FROM order_date) = 2023
       GROUP BY month, category
)
SELECT DISTINCT((COALESCE(cs22.month, cs23.month))) AS month,
    cs22.category AS category_2022,
    cs22.total_sales AS total_sales_2022,
    cs23.category AS category_2023,
    cs23.total_sales AS total_sales_2023
FROM category_sales_2022 cs22
FULL OUTER JOIN category_sales_2023 cs23
ON cs22.month = cs23.month
WHERE cs22.rank = 1 AND cs23.rank = 1
ORDER BY month;
```

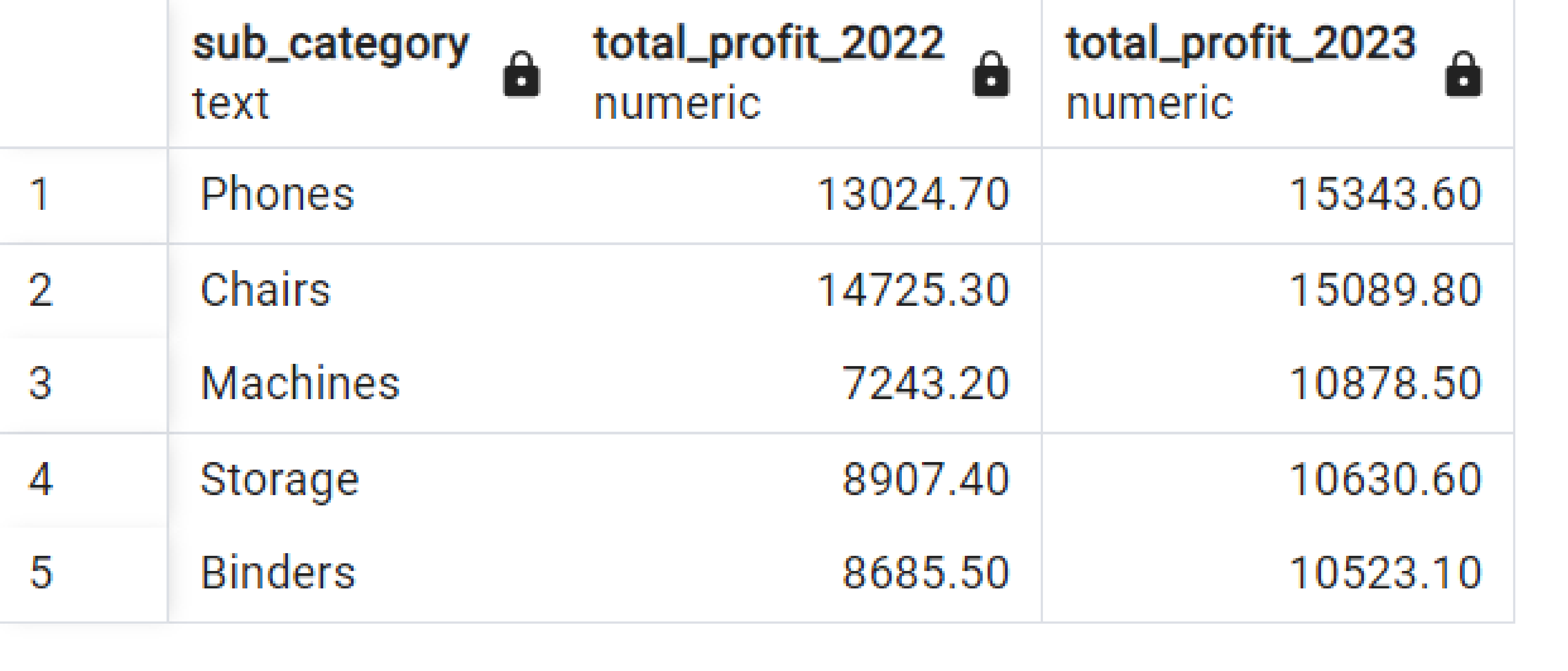


	month numeric	category_2022 text	total_sales_2022 numeric	category_2023 text	total_sales_2023 numeric
1	1	Furniture	32728.6	Technology	36209.7
2	2	Office Supplies	33841.0	Technology	48152.1
3	3	Furniture	30207.9	Technology	29767.5
4	4	Office Supplies	40114.7	Technology	45884.4
5	5	Technology	30319.0	Technology	31939.1
6	6	Technology	36605.8	Technology	28886.4
7	7	Office Supplies	29098.8	Office Supplies	31682.5
8	8	Technology	49488.5	Furniture	42231.8
9	9	Technology	31585.1	Technology	27676.1
10	10	Technology	50021.0	Technology	53000.1
11	11	Furniture	32035.2	Technology	27517.9
12	12	Technology	38025.3	Office Supplies	37586.4

Q6. Which sub-category (top 5) experienced the highest profit growth in 2023 compared to 2022

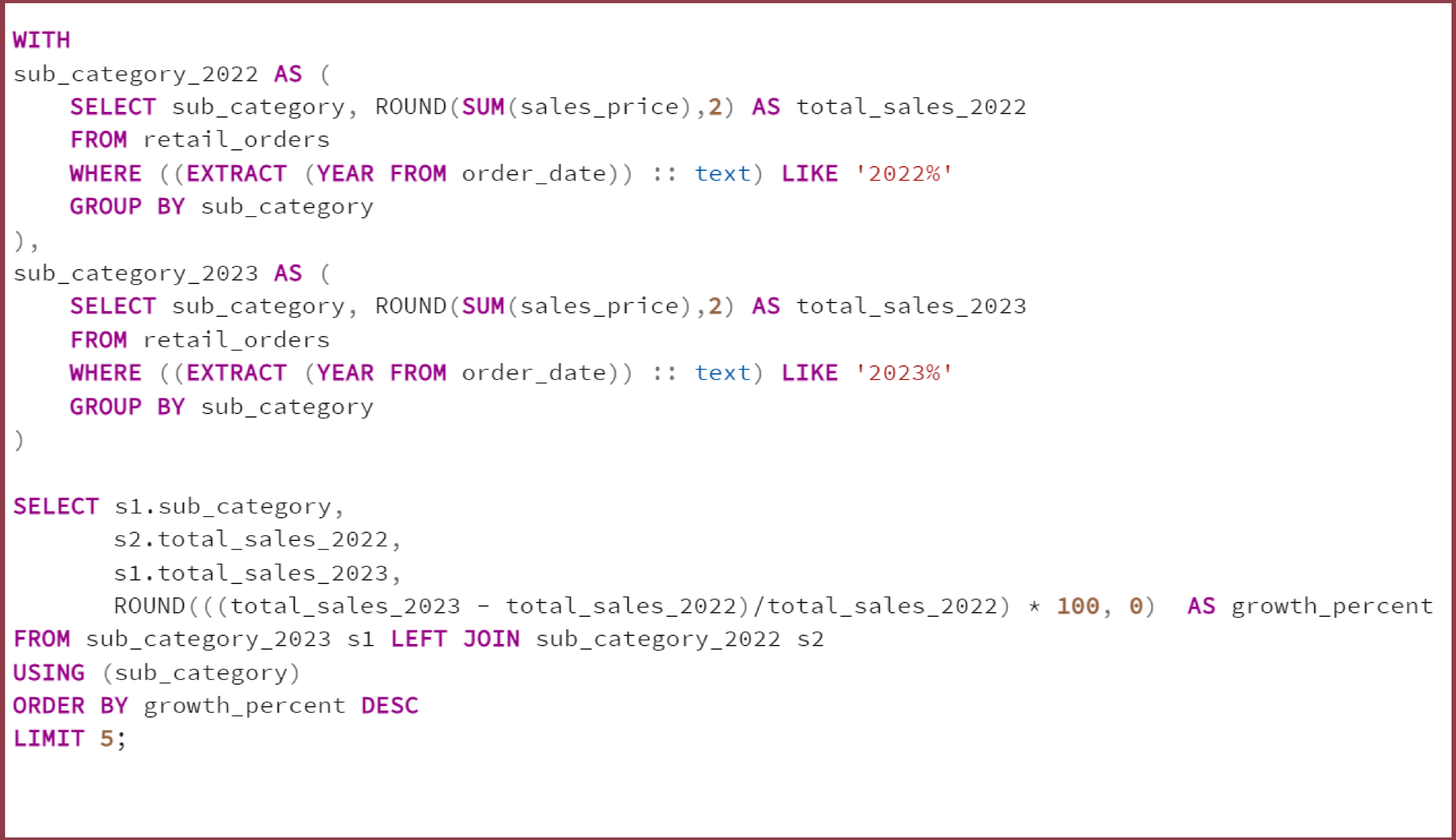


```sql
WITH
sub_category_2022 AS (
    SELECT sub_category, ROUND(SUM(profit),2) AS total_profit_2022
    FROM retail_orders
    WHERE ((EXTRACT (YEAR FROM order_date)) :: text) LIKE '2022%'
    GROUP BY sub_category
),
sub_category_2023 AS (
    SELECT sub_category, ROUND(SUM(profit),2) AS total_profit_2023
    FROM retail_orders
    WHERE ((EXTRACT (YEAR FROM order_date)) :: text) LIKE '2023%'
    GROUP BY sub_category
)

SELECT sub_category, total_profit_2022, total_profit_2023
FROM sub_category_2023 LEFT JOIN sub_category_2022
USING (sub_category)
WHERE total_profit_2023 > total_profit_2022
ORDER BY total_profit_2023 DESC
LIMIT 5;
```

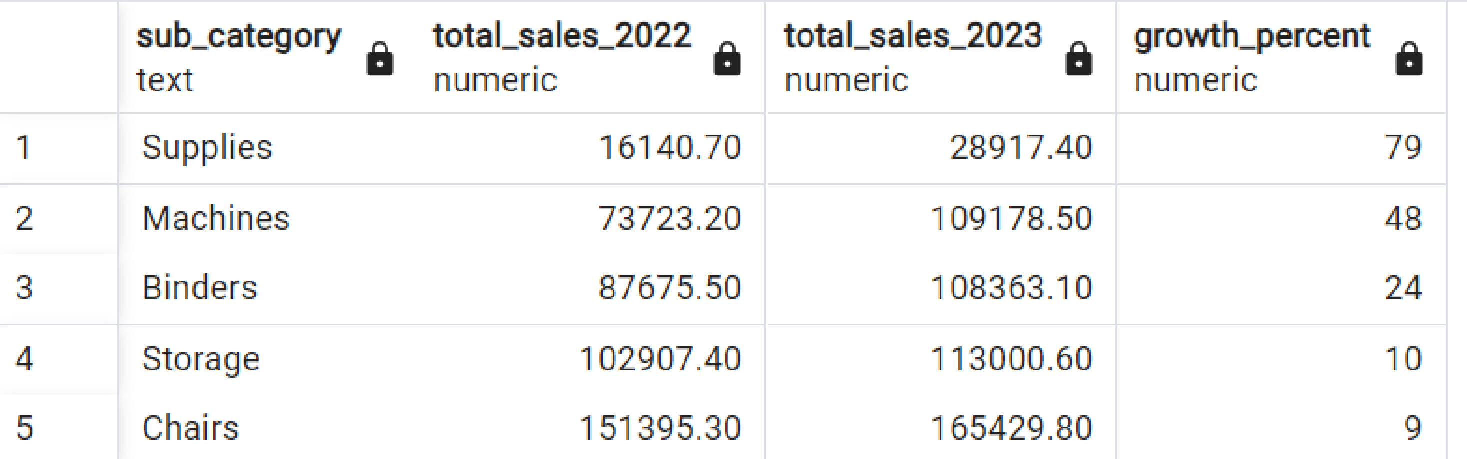


	sub_category text 🔒	total_profit_2022 numeric 🔒	total_profit_2023 numeric 🔒
1	Phones	13024.70	15343.60
2	Chairs	14725.30	15089.80
3	Machines	7243.20	10878.50
4	Storage	8907.40	10630.60
5	Binders	8685.50	10523.10

Q7. Which sub-category (top 5) had the highest growth percentage in sales in 2023 compared to 2022



```
WITH
sub_category_2022 AS (
    SELECT sub_category, ROUND(SUM(sales_price),2) AS total_sales_2022
    FROM retail_orders
    WHERE ((EXTRACT (YEAR FROM order_date)) :: text) LIKE '2022%'
    GROUP BY sub_category
),
sub_category_2023 AS (
    SELECT sub_category, ROUND(SUM(sales_price),2) AS total_sales_2023
    FROM retail_orders
    WHERE ((EXTRACT (YEAR FROM order_date)) :: text) LIKE '2023%'
    GROUP BY sub_category
)

SELECT s1.sub_category,
       s2.total_sales_2022,
       s1.total_sales_2023,
       ROUND(((total_sales_2023 - total_sales_2022)/total_sales_2022) * 100, 0)  AS growth_percent
FROM sub_category_2023 s1 LEFT JOIN sub_category_2022 s2
USING (sub_category)
ORDER BY growth_percent DESC
LIMIT 5;
```



	sub_category text	total_sales_2022 numeric	total_sales_2023 numeric	growth_percent numeric
1	Supplies	16140.70	28917.40	79
2	Machines	73723.20	109178.50	48
3	Binders	87675.50	108363.10	24
4	Storage	102907.40	113000.60	10
5	Chairs	151395.30	165429.80	9

Insights Gained

- Product Performance and Revenue Drivers.
- Regional Sales Trends and Opportunities.
- Seasonal Sales Patterns and Growth Trends.

Next Steps...

- Deep Dive into High-Performing Products.
- Segmented Regional Analysis.
- Time-Series Analysis for Sales Forecasting.

Thank You!