SQL

Sales Data Analysis

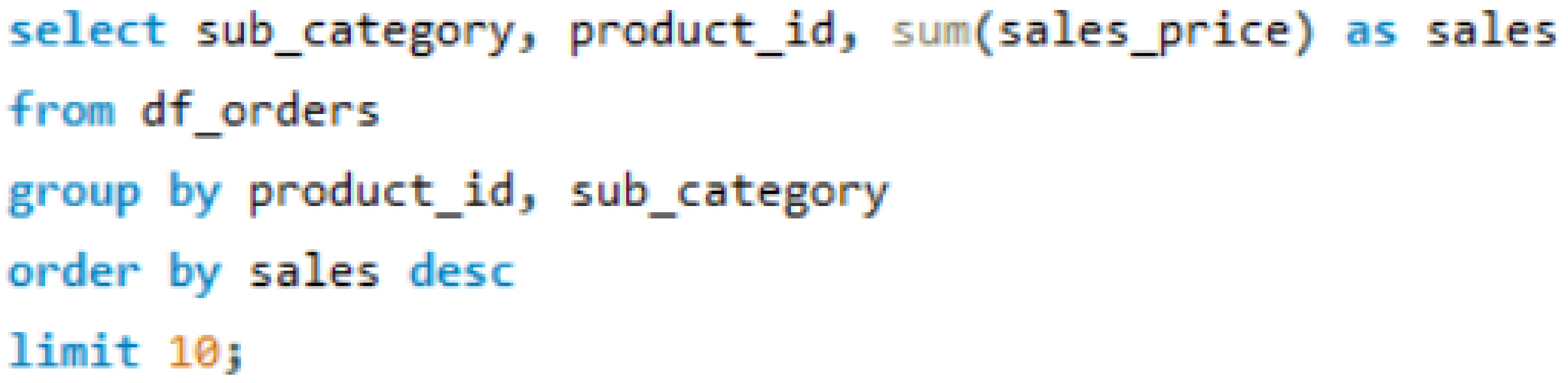
By Rishav Anand

Methodology:

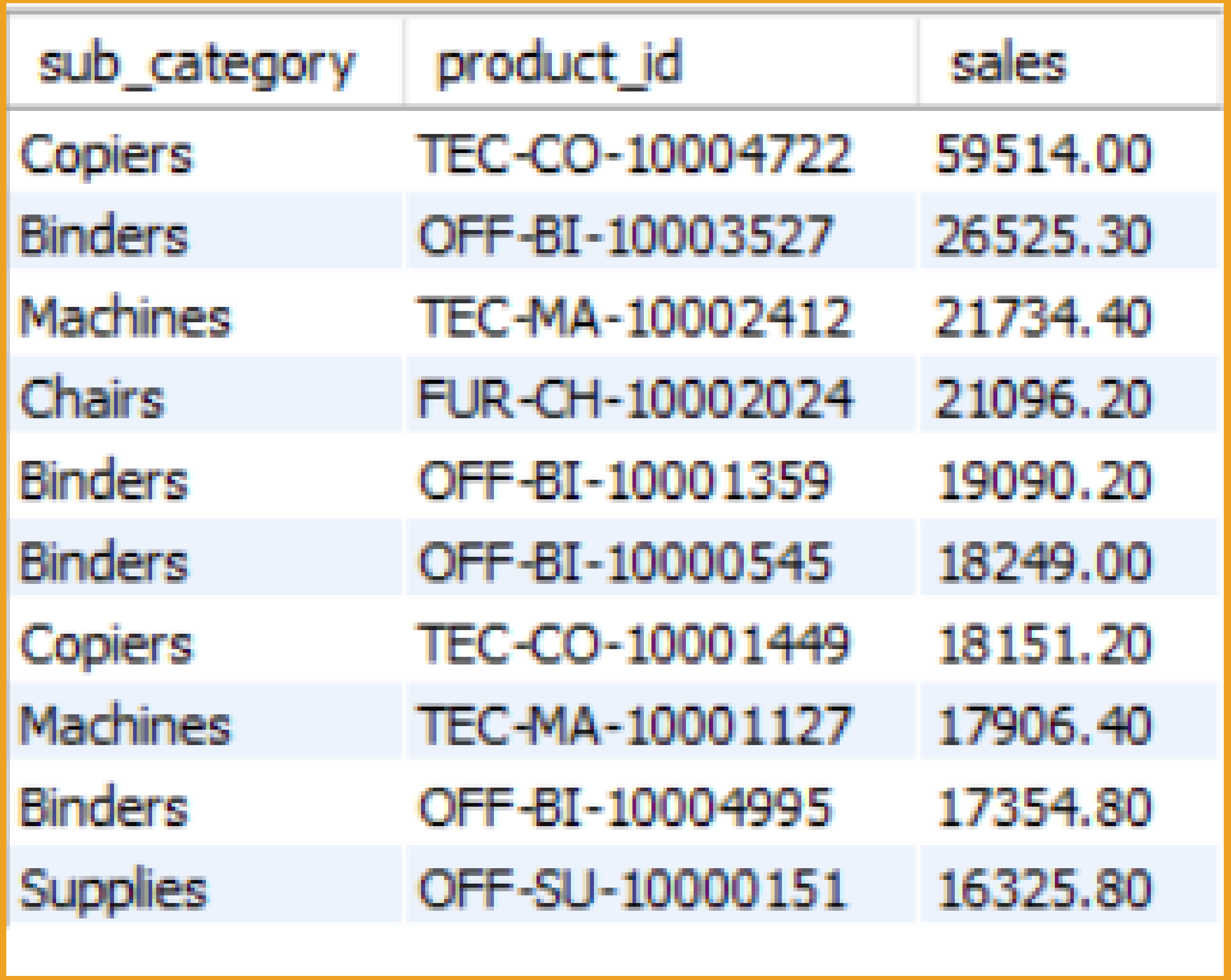
- Created a MySQL database and relevant tables to store the cleaned data. Appended the cleaned data into the MySQL database using Python.

- Executed advanced SQL queries to answer key business questions. Queries focused on total sales per product, monthly sales trends, and identifying top-selling products.

find top 10 highest revenue generating products

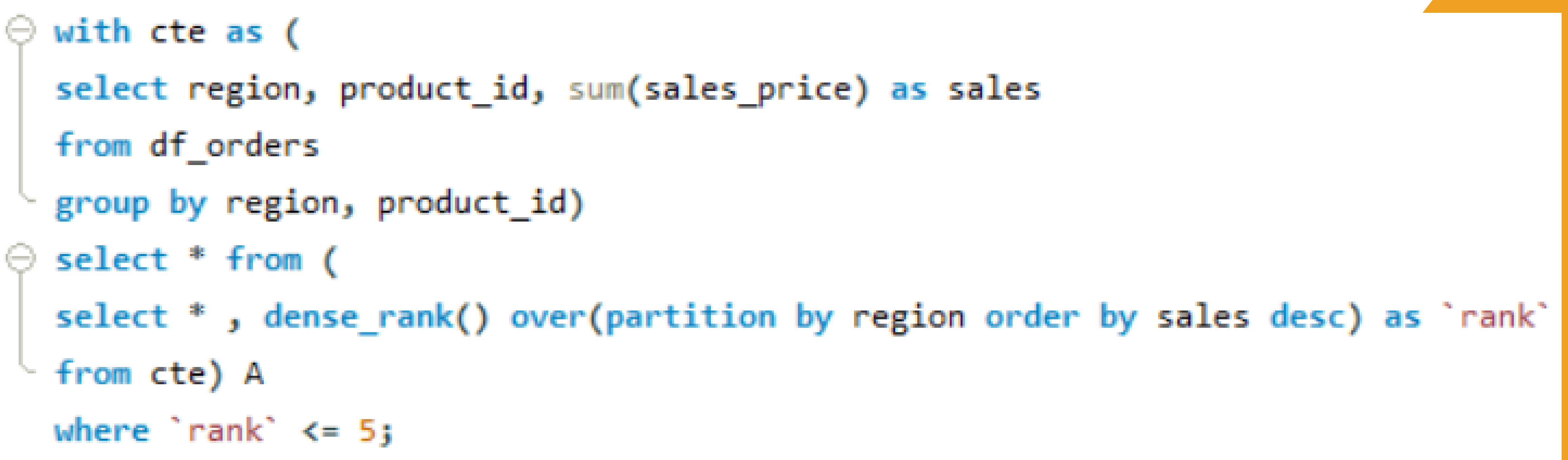


```
select sub_category, product_id, sum(sales_price) as sales
from df_orders
group by product_id, sub_category
order by sales desc
limit 10;
```

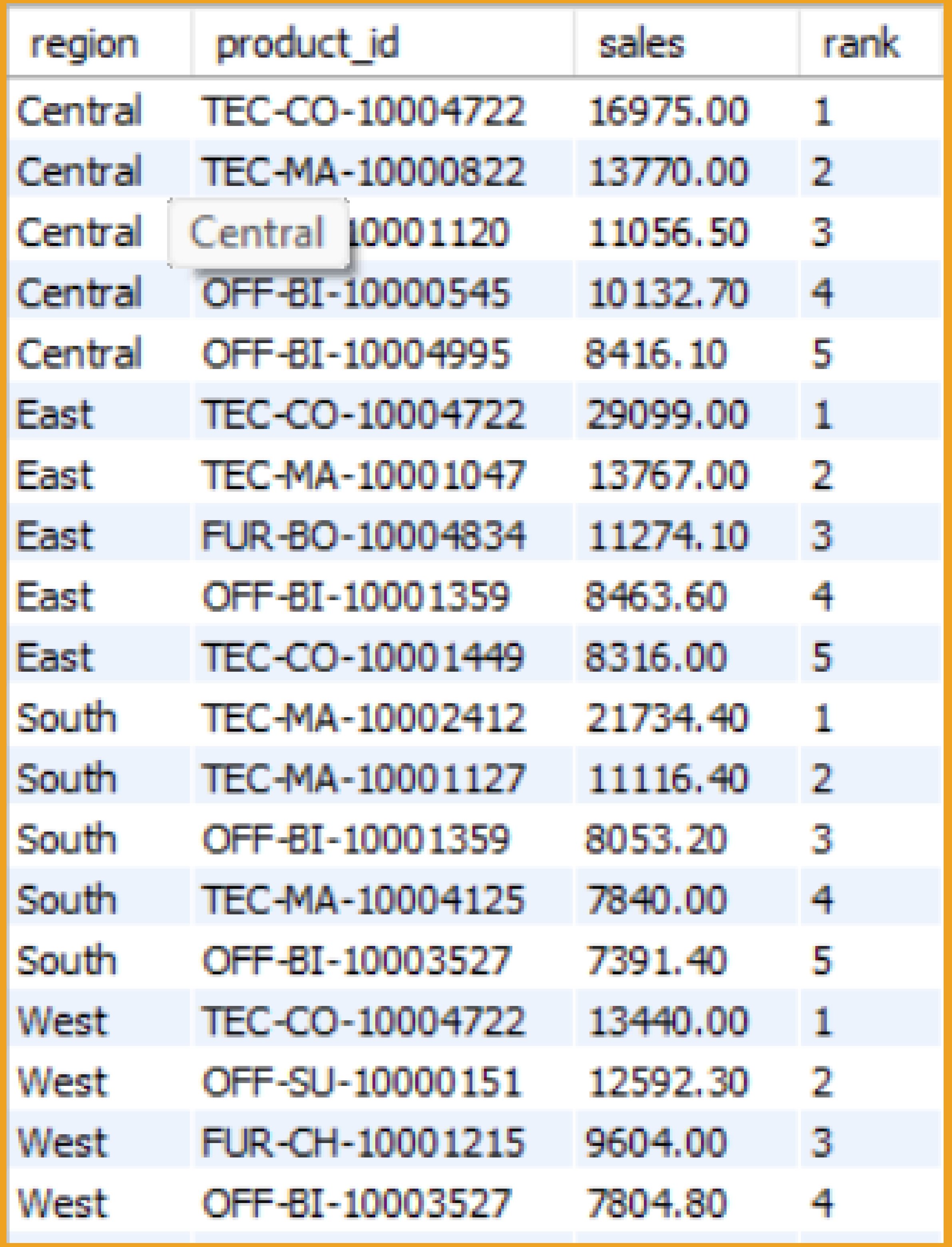


sub_category	product_id	sales
Copiers	TEC-CO-10004722	59514.00
Binders	OFF-BI-10003527	26525.30
Machines	TEC-MA-10002412	21734.40
Chairs	FUR-CH-10002024	21096.20
Binders	OFF-BI-10001359	19090.20
Binders	OFF-BI-10000545	18249.00
Copiers	TEC-CO-10001449	18151.20
Machines	TEC-MA-10001127	17906.40
Binders	OFF-BI-10004995	17354.80
Supplies	OFF-SU-10000151	16325.80

find top 5 highest selling products in each region

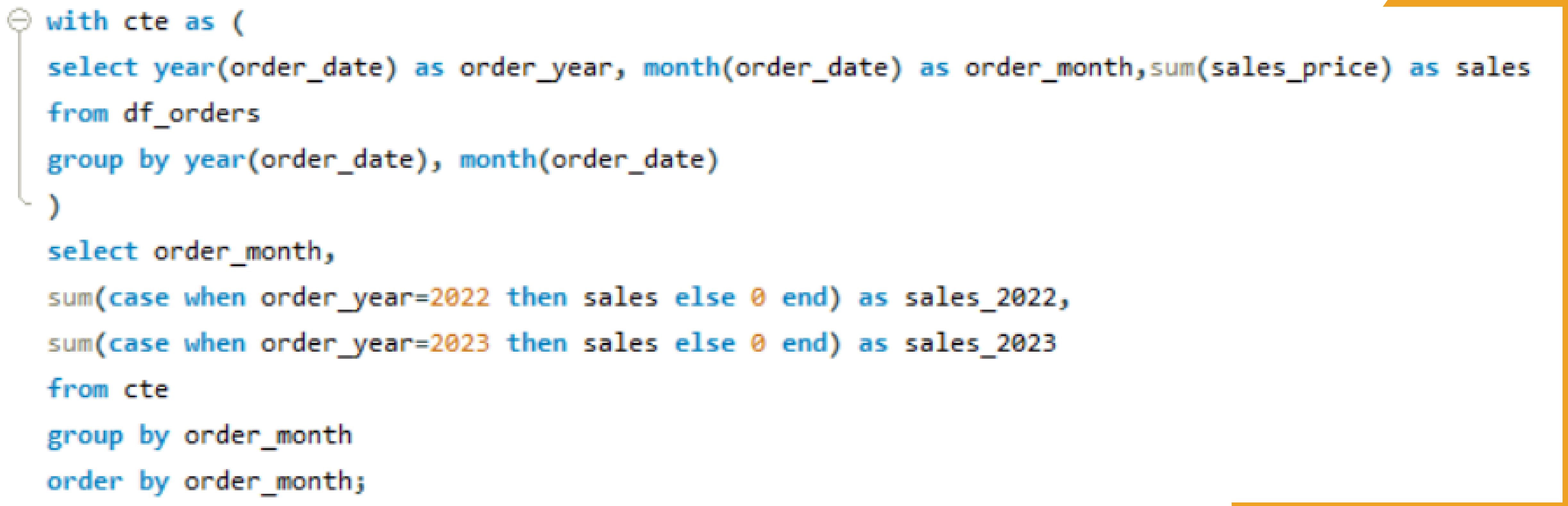


```
with cte as (
select region, product_id, sum(sales_price) as sales
from df_orders
group by region, product_id)
select * from (
select * , dense_rank() over(partition by region order by sales desc) as `rank`
from cte) A
where `rank` <= 5;
```

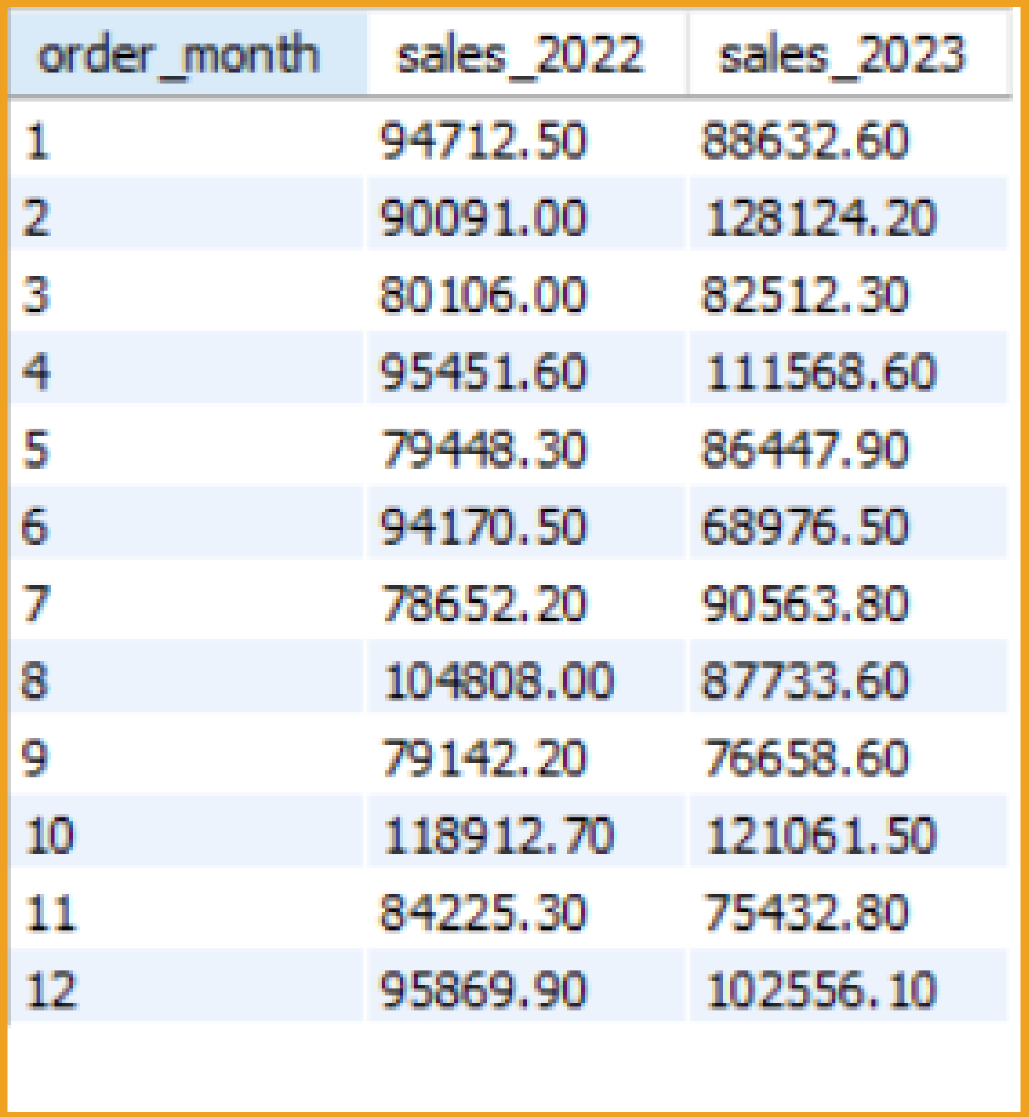


region	product_id	sales	rank
Central	TEC-CO-10004722	16975.00	1
Central	TEC-MA-10000822	13770.00	2
Central	Central 10001120	11056.50	3
Central	OFF-BI-10000545	10132.70	4
Central	OFF-BI-10004995	8416.10	5
East	TEC-CO-10004722	29099.00	1
East	TEC-MA-10001047	13767.00	2
East	FUR-BO-10004834	11274.10	3
East	OFF-BI-10001359	8463.60	4
East	TEC-CO-10001449	8316.00	5
South	TEC-MA-10002412	21734.40	1
South	TEC-MA-10001127	11116.40	2
South	OFF-BI-10001359	8053.20	3
South	TEC-MA-10004125	7840.00	4
South	OFF-BI-10003527	7391.40	5
West	TEC-CO-10004722	13440.00	1
West	OFF-SU-10000151	12592.30	2
West	FUR-CH-10001215	9604.00	3
West	OFF-BI-10003527	7804.80	4

find month over month growth comparison for 2022 and 2023 sales

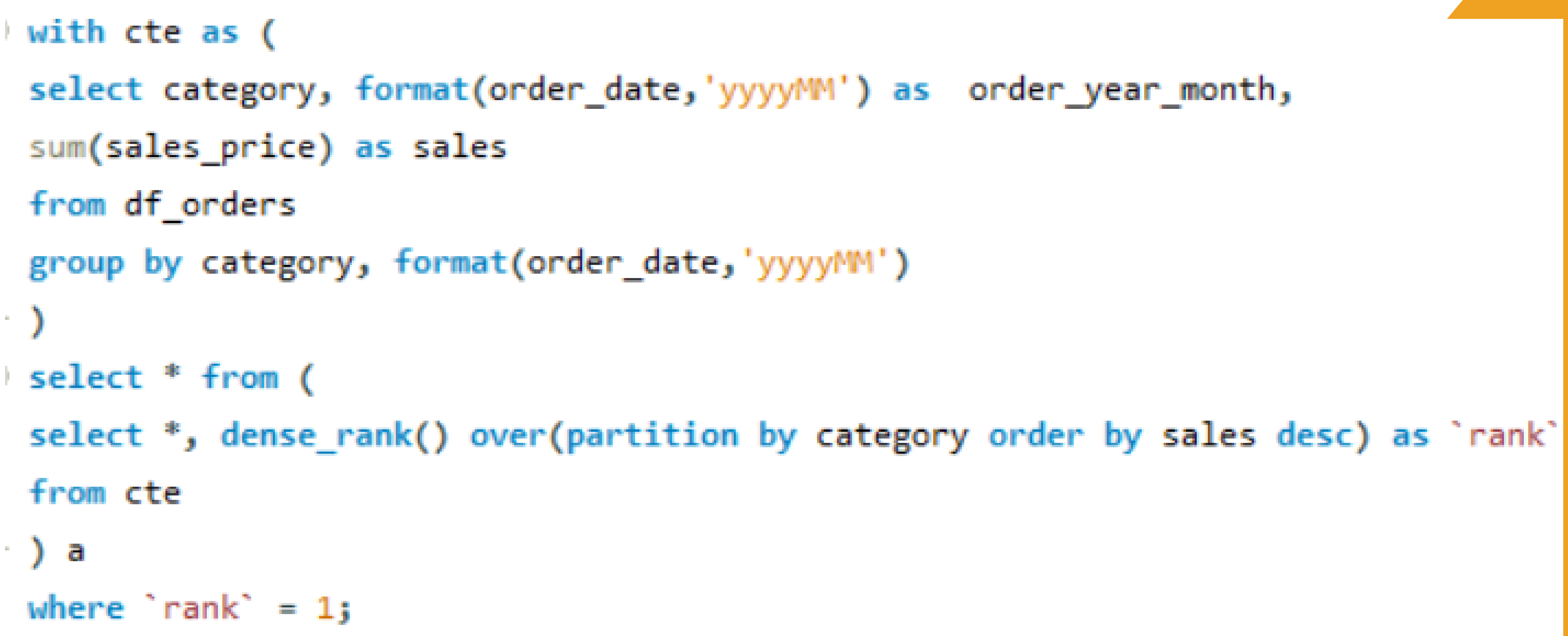


```
with cte as (
select year(order_date) as order_year, month(order_date) as order_month,sum(sales_price) as sales
from df_orders
group by year(order_date), month(order_date)
)
select order_month,
sum(case when order_year=2022 then sales else 0 end) as sales_2022,
sum(case when order_year=2023 then sales else 0 end) as sales_2023
from cte
group by order_month
order by order_month;
```

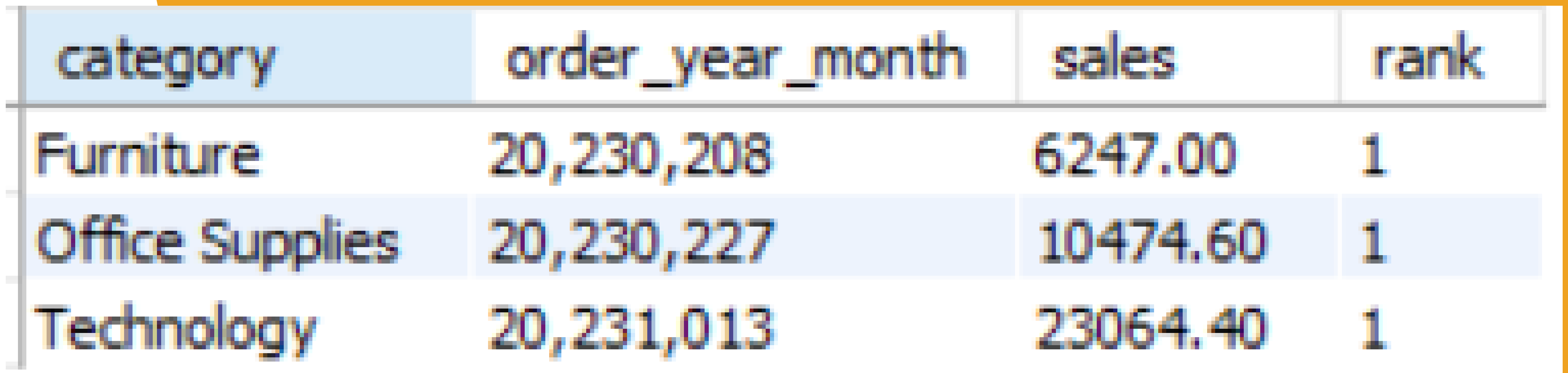


order_month	sales_2022	sales_2023
1	94712.50	88632.60
2	90091.00	128124.20
3	80106.00	82512.30
4	95451.60	111568.60
5	79448.30	86447.90
6	94170.50	68976.50
7	78652.20	90563.80
8	104808.00	87733.60
9	79142.20	76658.60
10	118912.70	121061.50
11	84225.30	75432.80
12	95869.90	102556.10

for each category which month has highest sales

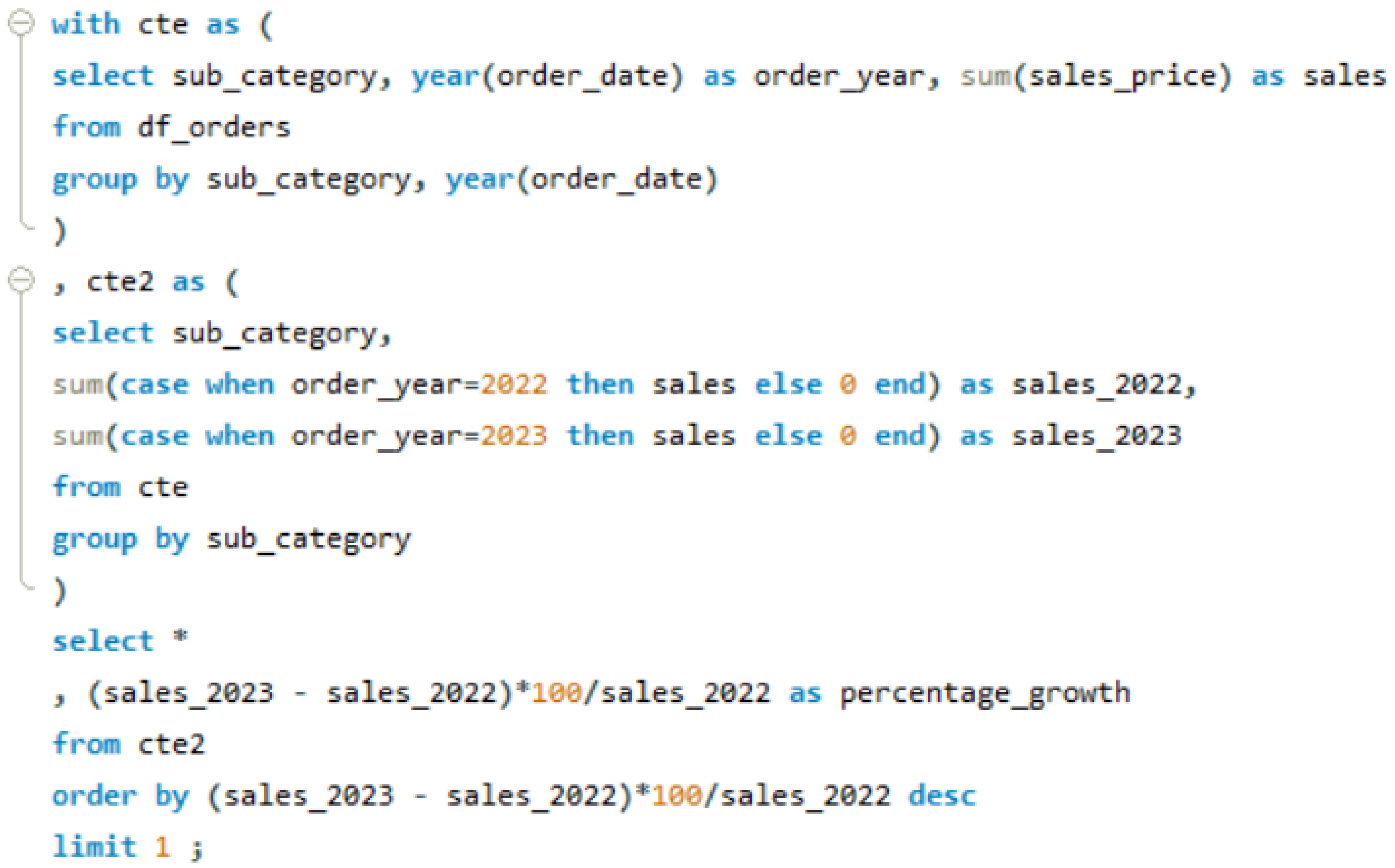


```
with cte as (
select category, format(order_date,'yyyyMM') as  order_year_month,
sum(sales_price) as sales
from df_orders
group by category, format(order_date,'yyyyMM')
)
select * from (
select *, dense_rank() over(partition by category order by sales desc) as `rank`
from cte
) a
where `rank` = 1;
```

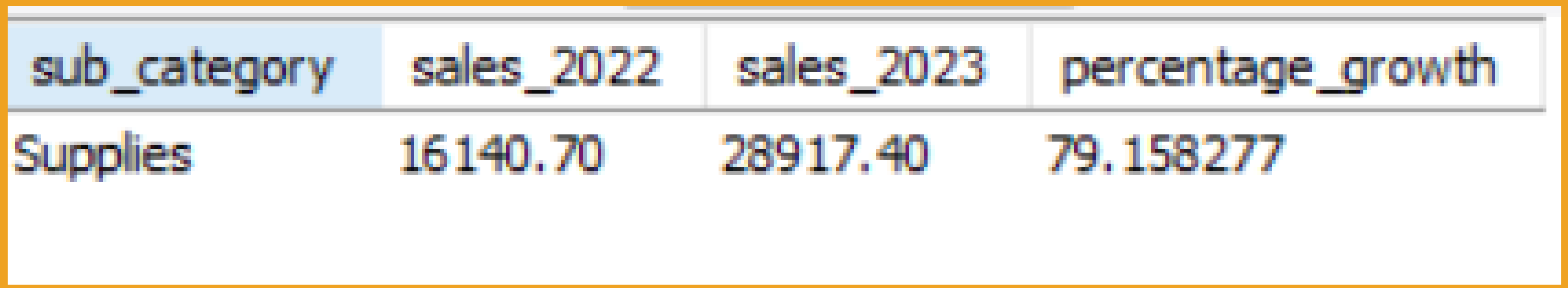


category	order_year_month	sales	rank
Furniture	20,230,208	6247.00	1
Office Supplies	20,230,227	10474.60	1
Technology	20,231,013	23064.40	1

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



```
with cte as (
select sub_category, year(order_date) as order_year, sum(sales_price) as sales
from df_orders
group by sub_category, year(order_date)
)
, cte2 as (
select sub_category,
sum(case when order_year=2022 then sales else 0 end) as sales_2022,
sum(case when order_year=2023 then sales else 0 end) as sales_2023
from cte
group by sub_category
)
select *
, (sales_2023 - sales_2022)*100/sales_2022 as percentage_growth
from cte2
order by (sales_2023 - sales_2022)*100/sales_2022 desc
limit 1 ;
```



sub_category	sales_2022	sales_2023	percentage_growth
Supplies	16140.70	28917.40	79.158277

Thank You