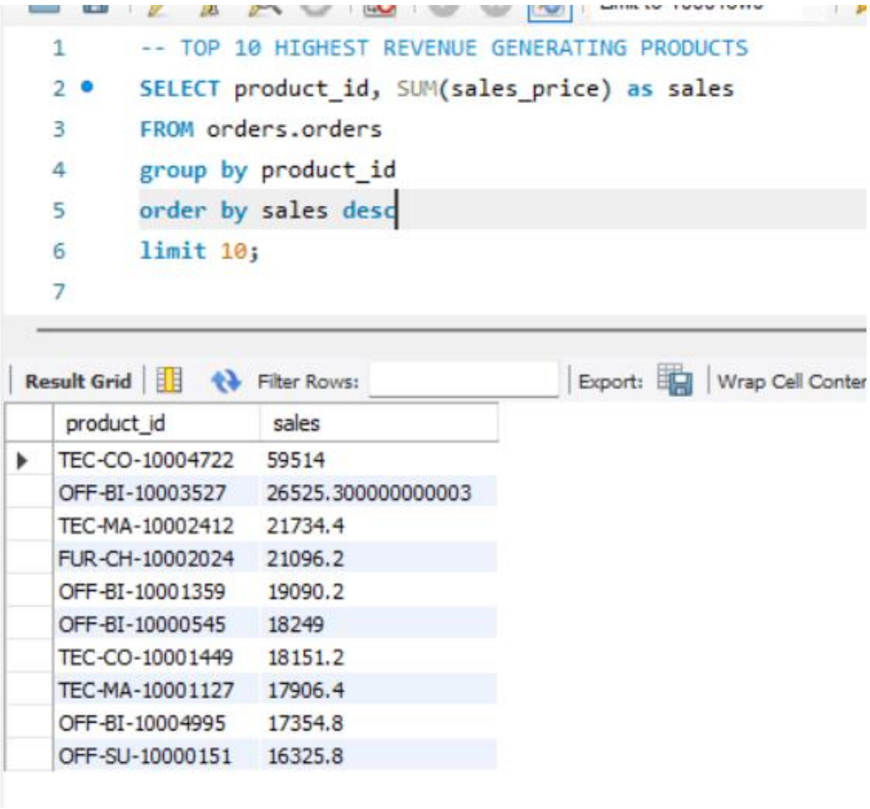
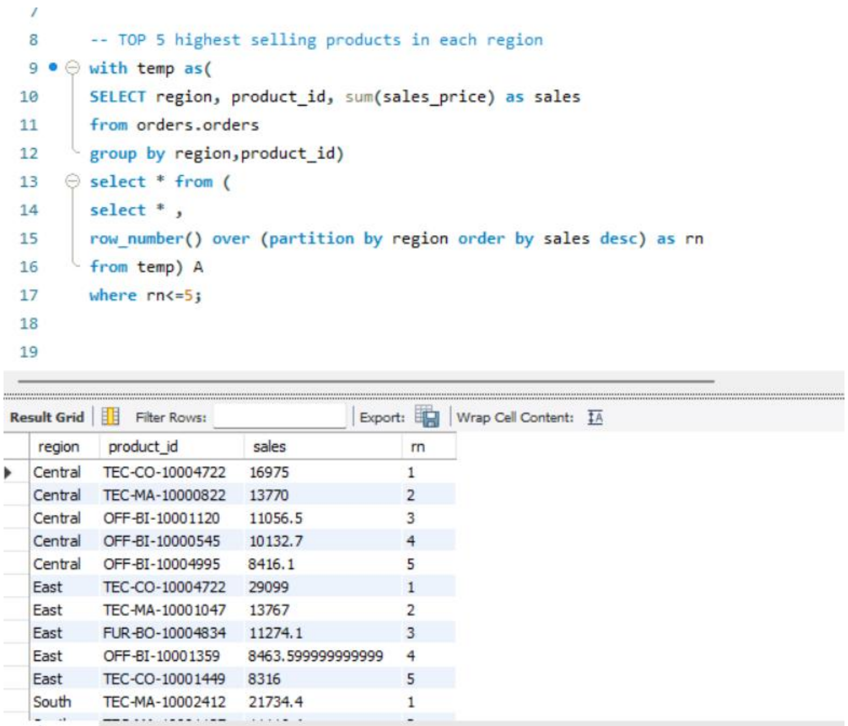
For getting top 10 highest revenue generating products:



```sql
-- TOP 10 HIGHEST REVENUE GENERATING PRODUCTS
SELECT product_id, SUM(sales_price) as sales
FROM orders.orders
group by product_id
order by sales desc
limit 10;
```

product_id	sales
TEC-CO-10004722	59514
OFF-BI-10003527	26525.300000000003
TEC-MA-10002412	21734.4
FUR-CH-10002024	21096.2
OFF-BI-10001359	19090.2
OFF-BI-10000545	18249
TEC-CO-10001449	18151.2
TEC-MA-10001127	17906.4
OFF-BI-10004995	17354.8
OFF-SU-10000151	16325.8

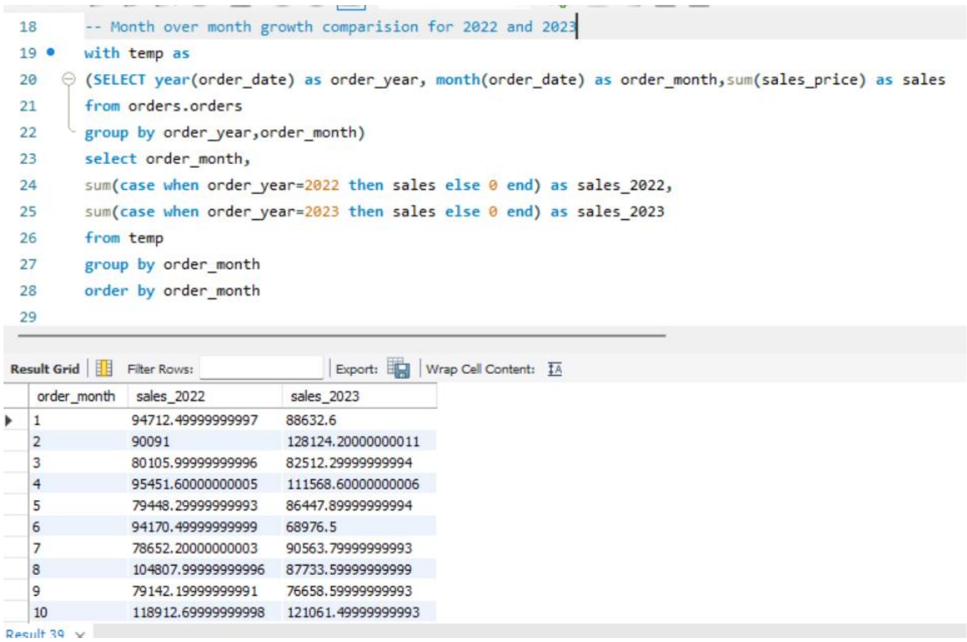
For getting top5 highest selling products in each region



```sql
-- TOP 5 highest selling products in each region
with temp as(
SELECT region, product_id, sum(sales_price) as sales
from orders.orders
group by region,product_id)
select * from (
select * ,
row_number() over (partition by region order by sales desc) as rn
from temp) A
where rn<=5;
```

region	product_id	sales	rn
Central	TEC-CO-10004722	16975	1
Central	TEC-MA-10000822	13770	2
Central	OFF-BI-10001120	11056.5	3
Central	OFF-BI-10000545	10132.7	4
Central	OFF-BI-10004995	8416.1	5
East	TEC-CO-10004722	29099	1
East	TEC-MA-10001047	13767	2
East	FUR-BO-10004834	11274.1	3
East	OFF-BI-10001359	8463.599999999999	4
East	TEC-CO-10001449	8316	5
South	TEC-MA-10002412	21734.4	1

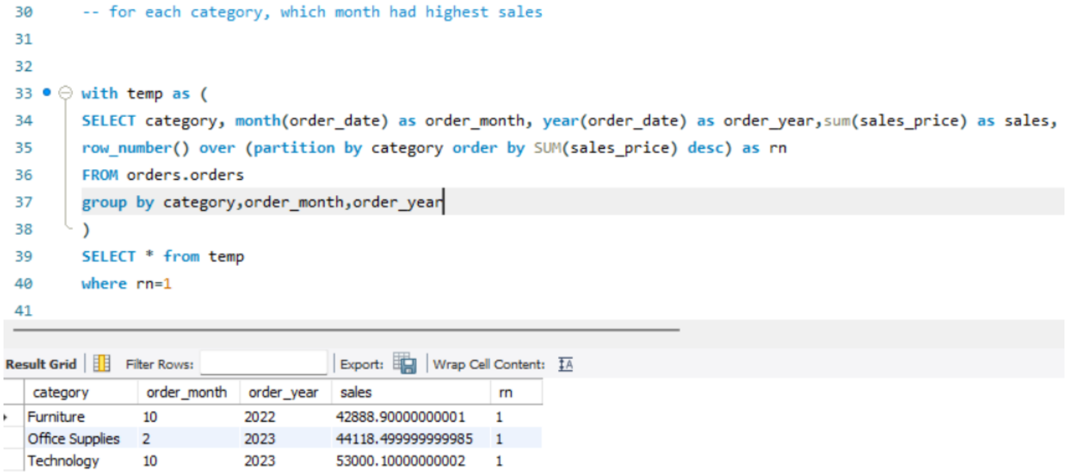
Month over month growth comparision for 2022 and 2023



```sql
-- Month over month growth comparision for 2022 and 2023
with temp as
(SELECT year(order_date) as order_year, month(order_date) as order_month,sum(sales_price) as sales
from orders.orders
group by order_year,order_month)
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 temp
group by order_month
order by order_month
```

order_month	sales_2022	sales_2023
1	94712.49999999997	88632.6
2	90091	128124.20000000011
3	80105.99999999996	82512.29999999994
4	95451.60000000005	111568.60000000006
5	79448.29999999993	86447.89999999994
6	94170.49999999999	68976.5
7	78652.20000000003	90563.79999999993
8	104807.99999999996	87733.59999999999
9	79142.19999999991	76658.59999999993
10	118912.69999999998	121061.49999999993

For each category, which month had highest sales

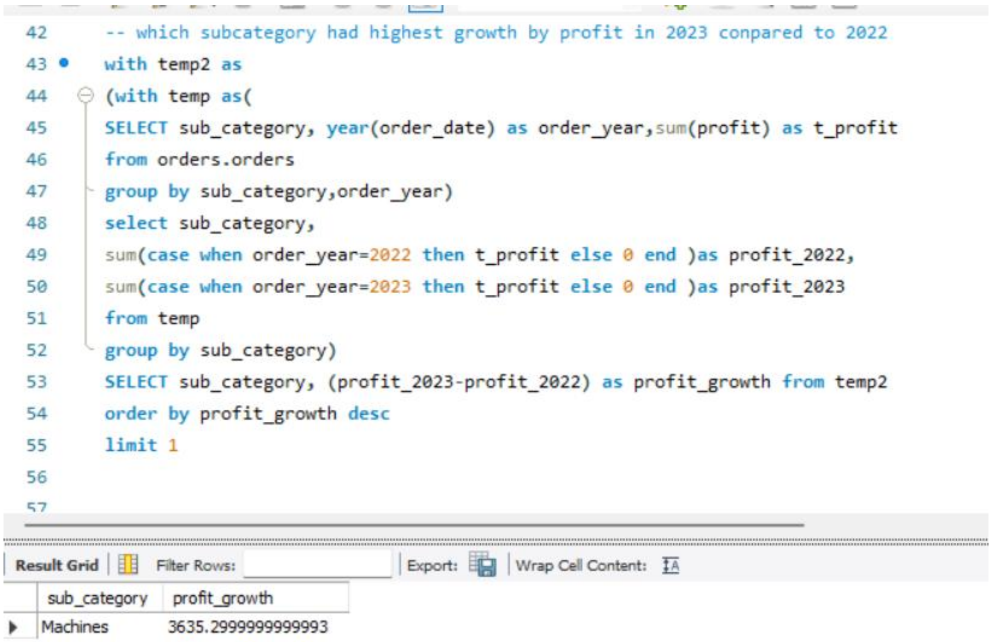


```sql
-- for each category, which month had highest sales


with temp as (
SELECT category, month(order_date) as order_month, year(order_date) as order_year,sum(sales_price) as sales,
row_number() over (partition by category order by SUM(sales_price) desc) as rn
FROM orders.orders
group by category,order_month,order_year
)
SELECT * from temp
where rn=1
```

category	order_month	order_year	sales	rn
Furniture	10	2022	42888.90000000001	1
Office Supplies	2	2023	44118.499999999985	1
Technology	10	2023	53000.10000000002	1

Which subcategory had highest growth by profit in 2023 compared to 2022



```sql
-- which subcategory had highest growth by profit in 2023 conpared to 2022
with temp2 as
(with temp as(
SELECT sub_category, year(order_date) as order_year,sum(profit) as t_profit
from orders.orders
group by sub_category,order_year)
select sub_category,
sum(case when order_year=2022 then t_profit else 0 end )as profit_2022,
sum(case when order_year=2023 then t_profit else 0 end )as profit_2023
from temp
group by sub_category)
SELECT sub_category, (profit_2023-profit_2022) as profit_growth from temp2
order by profit_growth desc
limit 1
```

sub_category	profit_growth
Machines	3635.2999999999993