End to end data analytics project using python and SQL. We will use Kaggle API to download the dataset and to data processing and cleaning using pandas and load the data into sql server. Lastly we will answer some interesting questions using SQL.

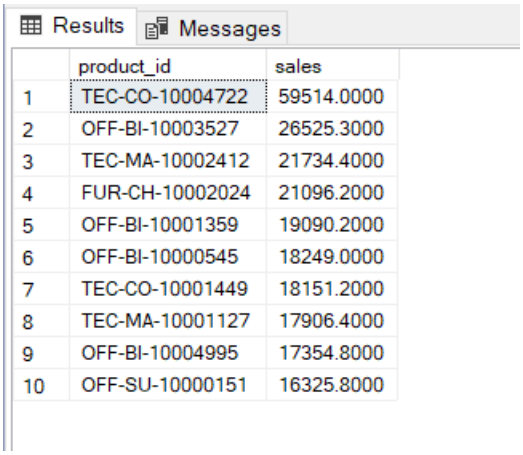
Questions;

1. find top 10 highest revenue generating products

SQL Code

```
SELECT top 10 product_id,sum(sale_price) AS sales
FROM df_orders
GROUP BY product_id
ORDER BY sales desc;
```

Result



Results | Messages

	product_id	sales
1	TEC-CO-10004722	59514.0000
2	OFF-BI-10003527	26525.3000
3	TEC-MA-10002412	21734.4000
4	FUR-CH-10002024	21096.2000
5	OFF-BI-10001359	19090.2000
6	OFF-BI-10000545	18249.0000
7	TEC-CO-10001449	18151.2000
8	TEC-MA-10001127	17906.4000
9	OFF-BI-10004995	17354.8000
10	OFF-SU-10000151	16325.8000

2. find top 5 highest selling products in each region

Code

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

Results

Results | Messages

	region	product_id	sales
1	Central	TEC-CO-10004722	16975.0000
2	Central	TEC-MA-10000822	13770.0000
3	Central	OFF-BI-10001120	11056.5000
4	Central	OFF-BI-10000545	10132.7000
5	Central	OFF-BI-10004995	8416.1000
6	East	TEC-CO-10004722	29099.0000
7	East	TEC-MA-10001047	13767.0000
8	East	FUR-BO-10004834	11274.1000
9	East	OFF-BI-10001359	8463.6000
10	East	TEC-CO-10001449	8316.0000
11	South	TEC-MA-10002412	21734.4000
12	South	TEC-MA-10001127	11116.4000
13	South	OFF-BI-10001359	8053.2000
14	South	TEC-MA-10004125	7840.0000
15	South	OFF-BI-10003527	7391.4000
16	West	TEC-CO-10004722	13440.0000
17	West	OFF-SU-10000151	12592.3000
18	West	FUR-CH-10001215	9604.0000
19	West	OFF-BI-10003527	7804.8000
20	West	TEC-AC-10003832	7722.7000

3. find month over month growth comparison for 2022 and 2023 sales eg : jan 2022 vs jan 2023

Code

```
with cte as (
select year(order_date) as order_year,month(order_date) as order_month,
sum(sale_price) as sales
from df_orders
group by year(order_date),month(order_date)
--order 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
```

Results

Results | Messages

	order_month	sales_2022	sales_2023
1	1	94712.5000	88632.6000
2	2	90091.0000	128124.2000
3	3	80106.0000	82512.3000
4	4	95451.6000	111568.6000
5	5	79448.3000	86447.9000
6	6	94170.5000	68976.5000
7	7	78652.2000	90563.8000
8	8	104808.0000	87733.6000
9	9	79142.2000	76658.6000
10	10	118912.7000	121061.5000
11	11	84225.3000	75432.8000
12	12	95869.9000	102556.1000

4. for each category which month had highest sales

Code

```
with cte as (
select category,year(order_date) as order_year,month(order_date) as order_month
, sum(sale_price) as sales
from df_orders
group by category,year(order_date),month(order_date)
--order by category,format(order_date,'yyyyMM')
)
select category, order_year,order_month,sales from (
select *,
row_number() over(partition by category order by sales desc) as rn
from cte
) a
where rn=1
```

Result

Results | Messages

	category	order_year	order_month	sales
1	Furniture	2022	10	42888.9000
2	Office Supplies	2023	2	44118.5000
3	Technology	2023	10	53000.1000

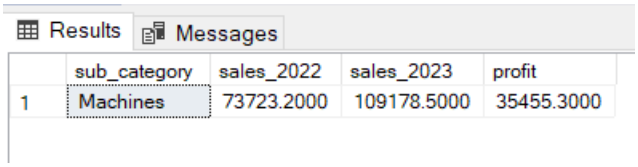
5. which sub category had highest growth by profit in 2023 compare to 2022

Code

```
with cte as (
select sub_category,year(order_date) as order_year,
sum(sale_price) as sales
from df_orders
group by sub_category,year(order_date)
--order by year(order_date),month(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 top 1 *
,(sales_2023-sales_2022) as profit
from  cte2
order by (sales_2023-sales_2022) desc
```

Result



	sub_category	sales_2022	sales_2023	profit
1	Machines	73723.2000	109178.5000	35455.3000