RETAIL ORDER ANALYSIS

Presented by

Chirag Modha

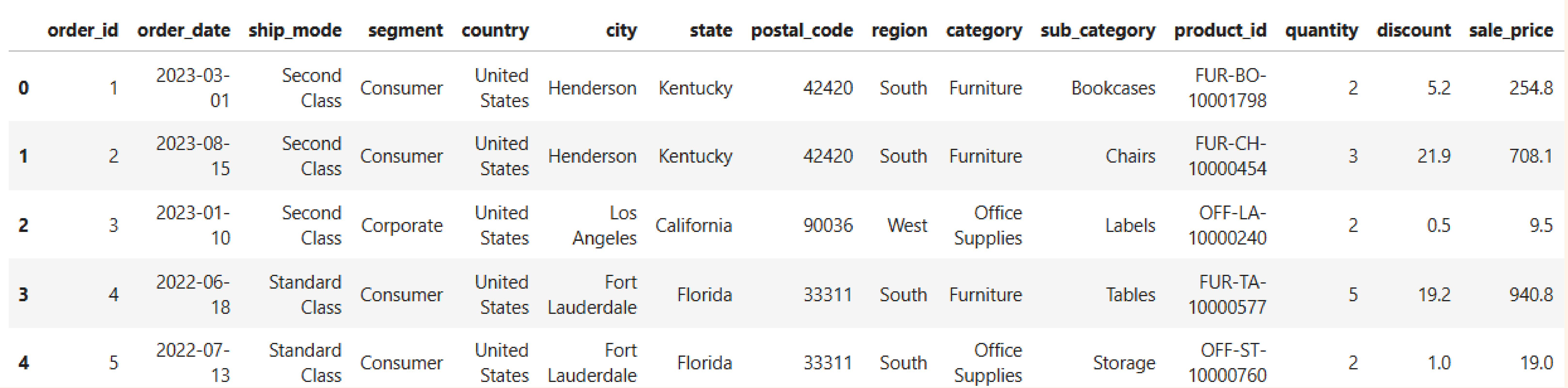
CONTENTS

1. Overview
2. Schema
3. Question Answered using SQL

OVERVIEW

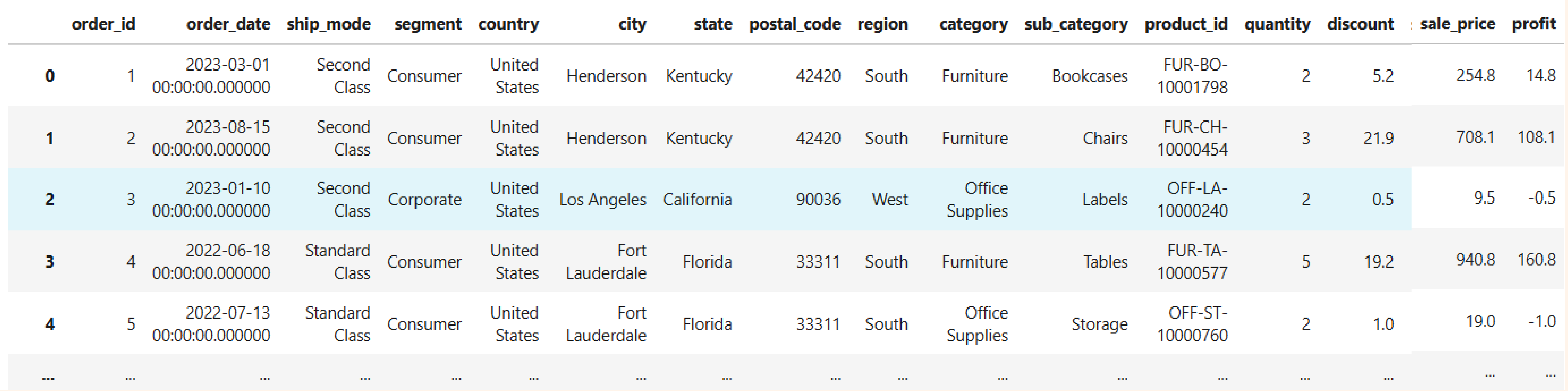
- The following Retail Order Analysis was done in Jupyter Notebook using Python and SQL. The dataset was provided by Ankit Bansal on Kaggle
- Dataset Link: https://www.kaggle.com/datasets/ankitbansal06/retail-orders
- The dataset was extracted with the help of kaggle api, data cleaning and processing with done in Jupyter Notebook with help of Python and loaded into SQL where the analysis was completed.

ORIGINAL DATA



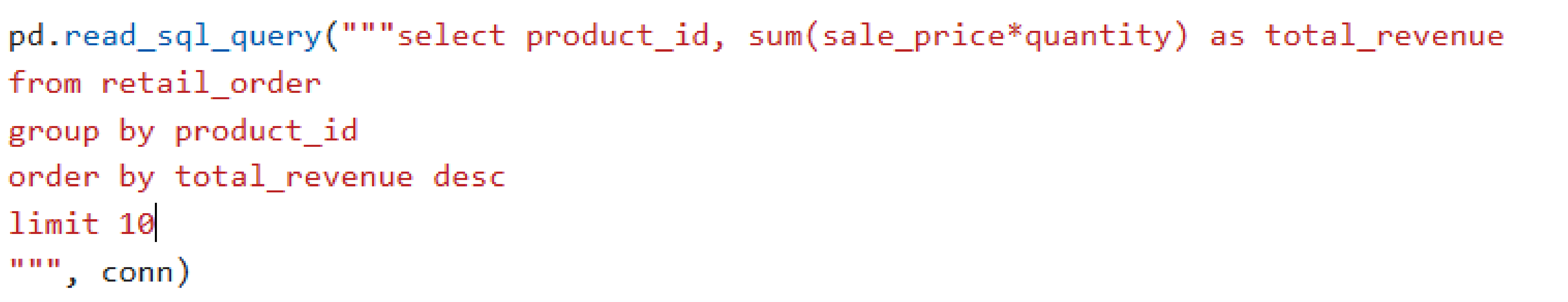
	order_id	order_date	ship_mode	segment	country	city	state	postal_code	region	category	sub_category	product_id	quantity	discount	sale_price
0	1	2023-03-01	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Bookcases	FUR-BO-10001798	2	5.2	254.8
1	2	2023-08-15	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Chairs	FUR-CH-10000454	3	21.9	708.1
2	3	2023-01-10	Second Class	Corporate	United States	Los Angeles	California	90036	West	Office Supplies	Labels	OFF-LA-10000240	2	0.5	9.5
3	4	2022-06-18	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Furniture	Tables	FUR-TA-10000577	5	19.2	940.8
4	5	2022-07-13	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Office Supplies	Storage	OFF-ST-10000760	2	1.0	19.0

DATA AFTER CLEANING AND TRANSFORMATION

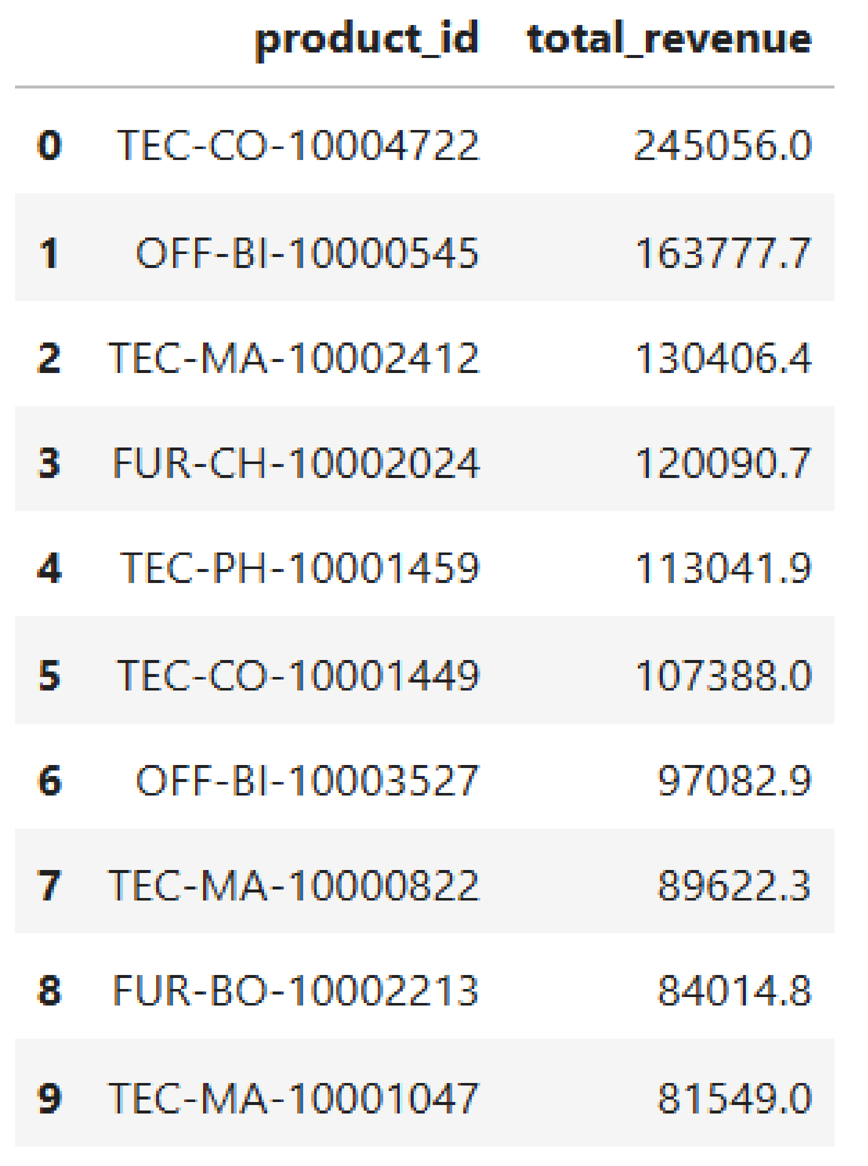


	order_id	order_date	ship_mode	segment	country	city	state	postal_code	region	category	sub_category	product_id	quantity	discount	sale_price	profit
0	1	2023-03-01 00:00:00.000000	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Bookcases	FUR-BO-10001798	2	5.2	254.8	14.8
1	2	2023-08-15 00:00:00.000000	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Chairs	FUR-CH-10000454	3	21.9	708.1	108.1
2	3	2023-01-10 00:00:00.000000	Second Class	Corporate	United States	Los Angeles	California	90036	West	Office Supplies	Labels	OFF-LA-10000240	2	0.5	9.5	-0.5
3	4	2022-06-18 00:00:00.000000	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Furniture	Tables	FUR-TA-10000577	5	19.2	940.8	160.8
4	5	2022-07-13 00:00:00.000000	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Office Supplies	Storage	OFF-ST-10000760	2	1.0	19.0	-1.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

FIND TOP 10 HIGHEST REVENUE GENERATING PRODUCTS

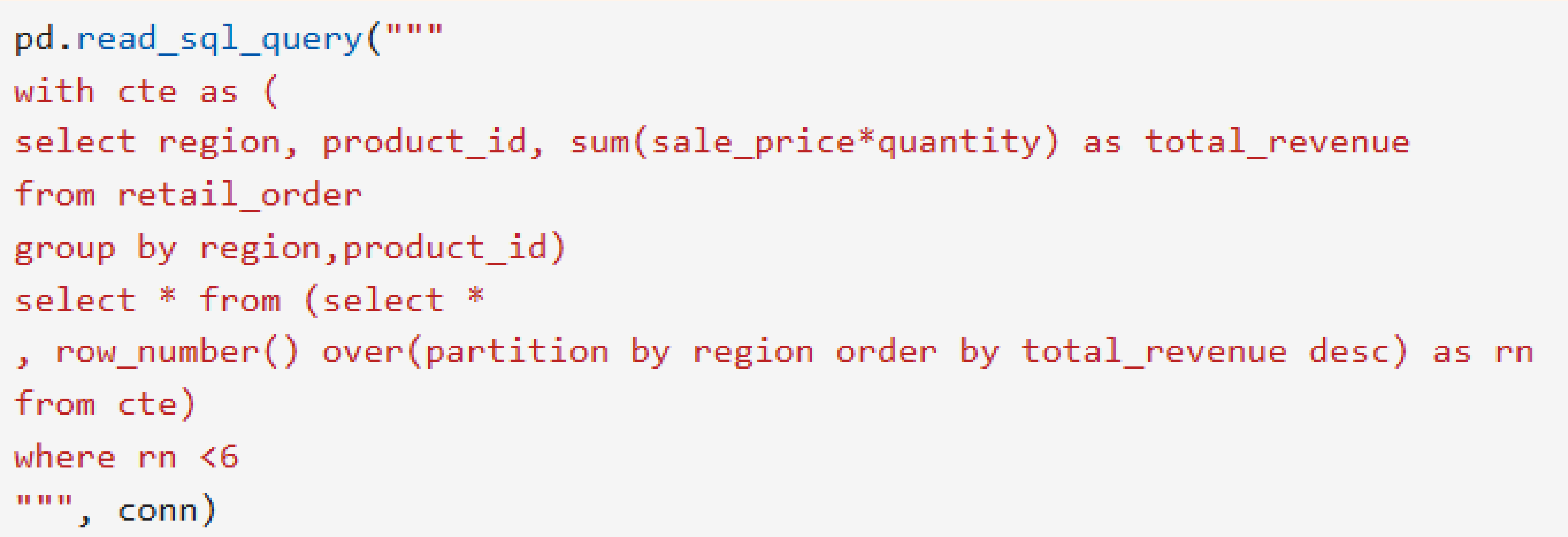


```
pd.read_sql_query("""select product_id, sum(sale_price*quantity) as total_revenue
from retail_order
group by product_id
order by total_revenue desc
limit 10
""", conn)
```

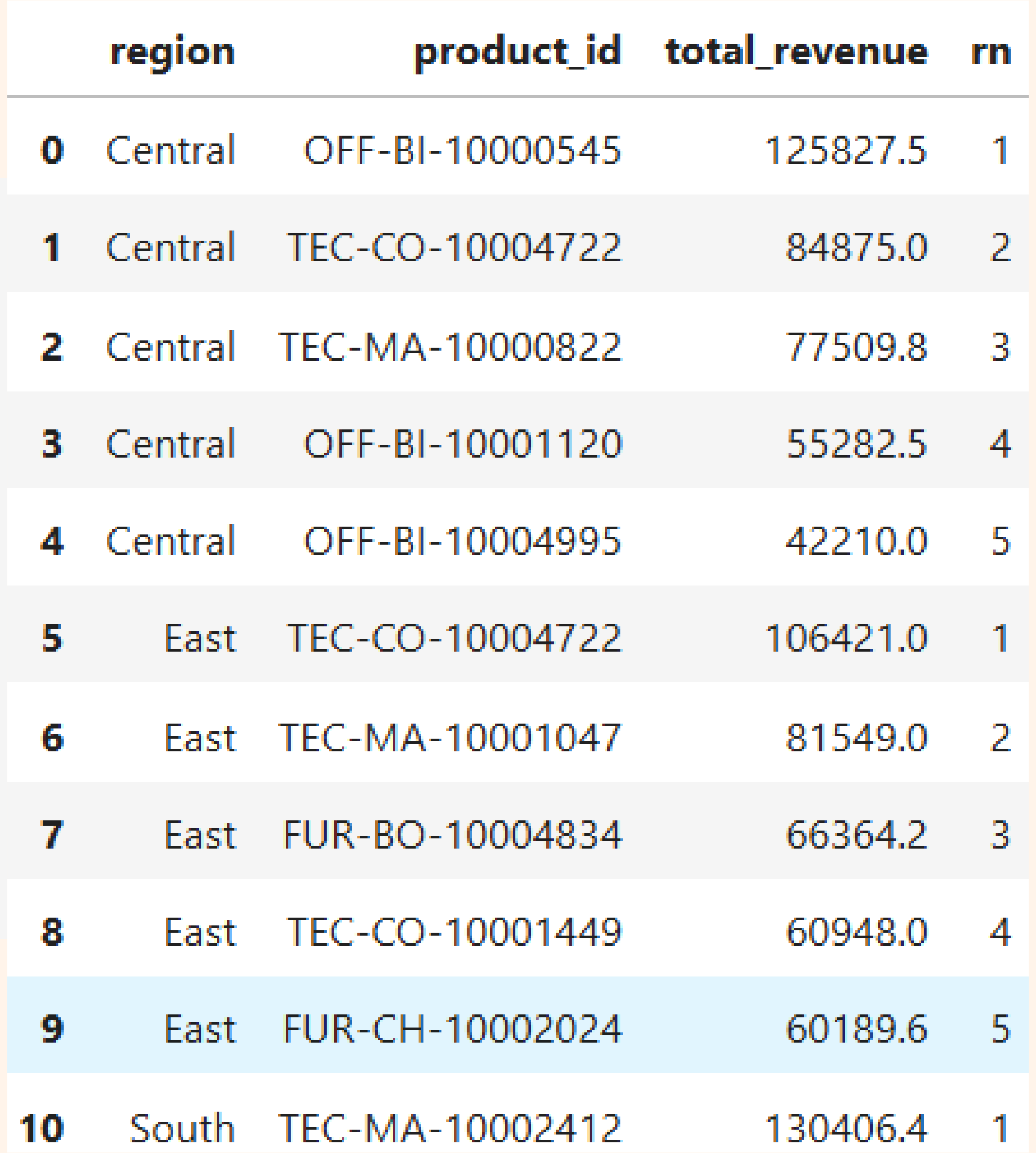


	product_id	total_revenue
0	TEC-CO-10004722	245056.0
1	OFF-BI-10000545	163777.7
2	TEC-MA-10002412	130406.4
3	FUR-CH-10002024	120090.7
4	TEC-PH-10001459	113041.9
5	TEC-CO-10001449	107388.0
6	OFF-BI-10003527	97082.9
7	TEC-MA-10000822	89622.3
8	FUR-BO-10002213	84014.8
9	TEC-MA-10001047	81549.0

FIND TOP 5 SELLING PRODUCTS IN EACH REGION

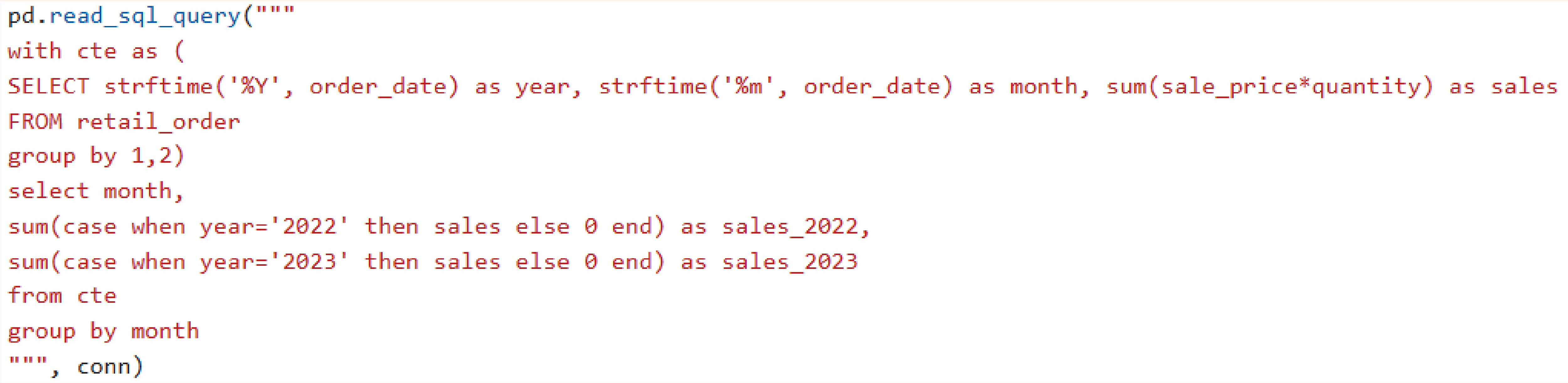


```
pd.read_sql_query("""
with cte as (
select region, product_id, sum(sale_price*quantity) as total_revenue
from retail_order
group by region,product_id)
select * from (select *
, row_number() over(partition by region order by total_revenue desc) as rn
from cte)
where rn <6
""", conn)
```

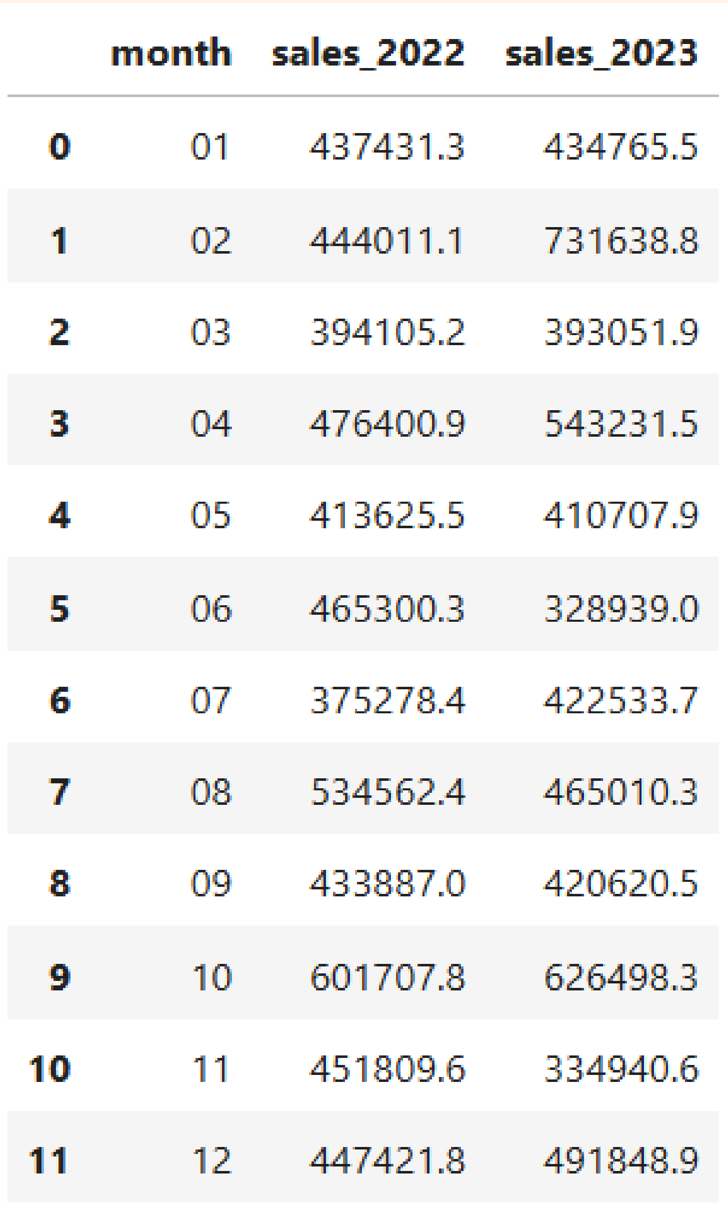


	region	product_id	total_revenue	rn
0	Central	OFF-BI-10000545	125827.5	1
1	Central	TEC-CO-10004722	84875.0	2
2	Central	TEC-MA-10000822	77509.8	3
3	Central	OFF-BI-10001120	55282.5	4
4	Central	OFF-BI-10004995	42210.0	5
5	East	TEC-CO-10004722	106421.0	1
6	East	TEC-MA-10001047	81549.0	2
7	East	FUR-BO-10004834	66364.2	3
8	East	TEC-CO-10001449	60948.0	4
9	East	FUR-CH-10002024	60189.6	5
10	South	TEC-MA-10002412	130406.4	1

FIND TOP 5 SELLING PRODUCTS IN EACH REGION

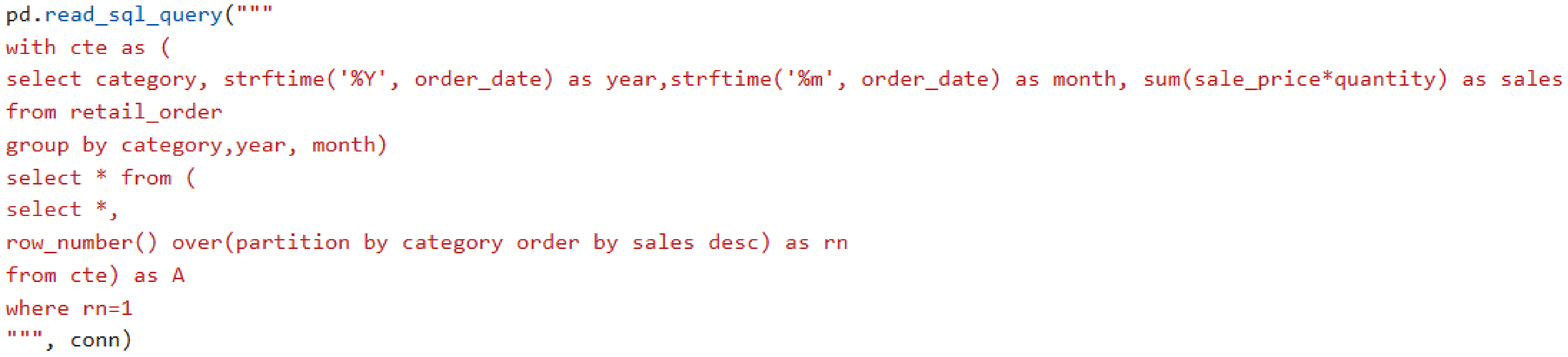


```
pd.read_sql_query("""
with cte as (
SELECT strftime('%Y', order_date) as year, strftime('%m', order_date) as month, sum(sale_price*quantity) as sales
FROM retail_order
group by 1,2)
select month,
sum(case when year='2022' then sales else 0 end) as sales_2022,
sum(case when year='2023' then sales else 0 end) as sales_2023
from cte
group by month
""", conn)
```

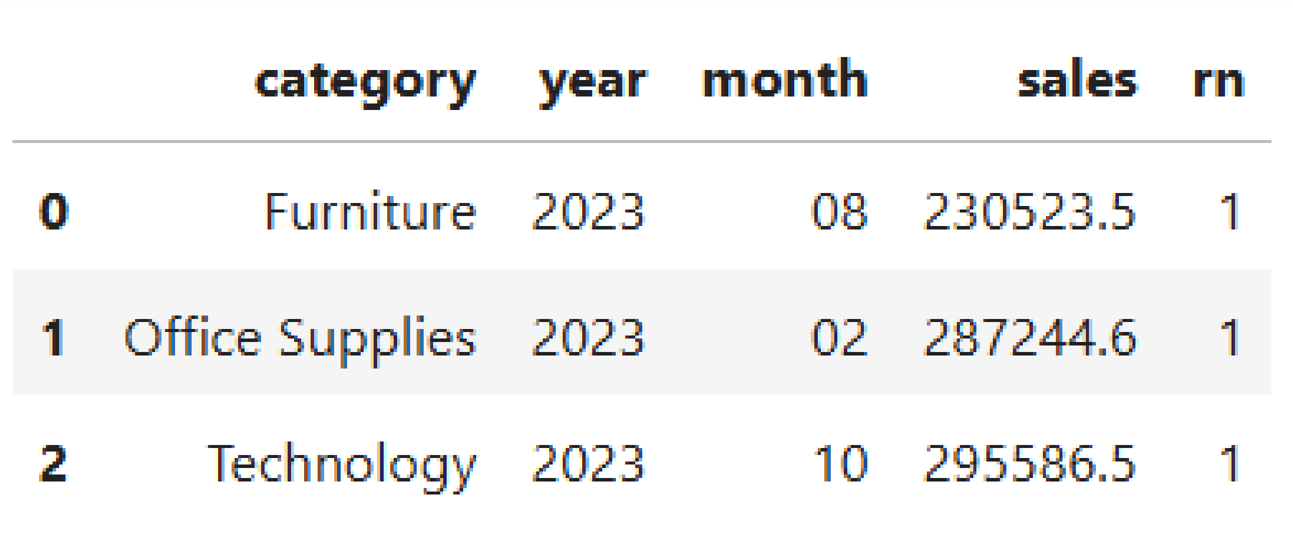


	month	sales_2022	sales_2023
0	01	437431.3	434765.5
1	02	444011.1	731638.8
2	03	394105.2	393051.9
3	04	476400.9	543231.5
4	05	413625.5	410707.9
5	06	465300.3	328939.0
6	07	375278.4	422533.7
7	08	534562.4	465010.3
8	09	433887.0	420620.5
9	10	601707.8	626498.3
10	11	451809.6	334940.6
11	12	447421.8	491848.9

FOR EACH CATEGORY WHICH MONTH HAD HIGHEST SALES

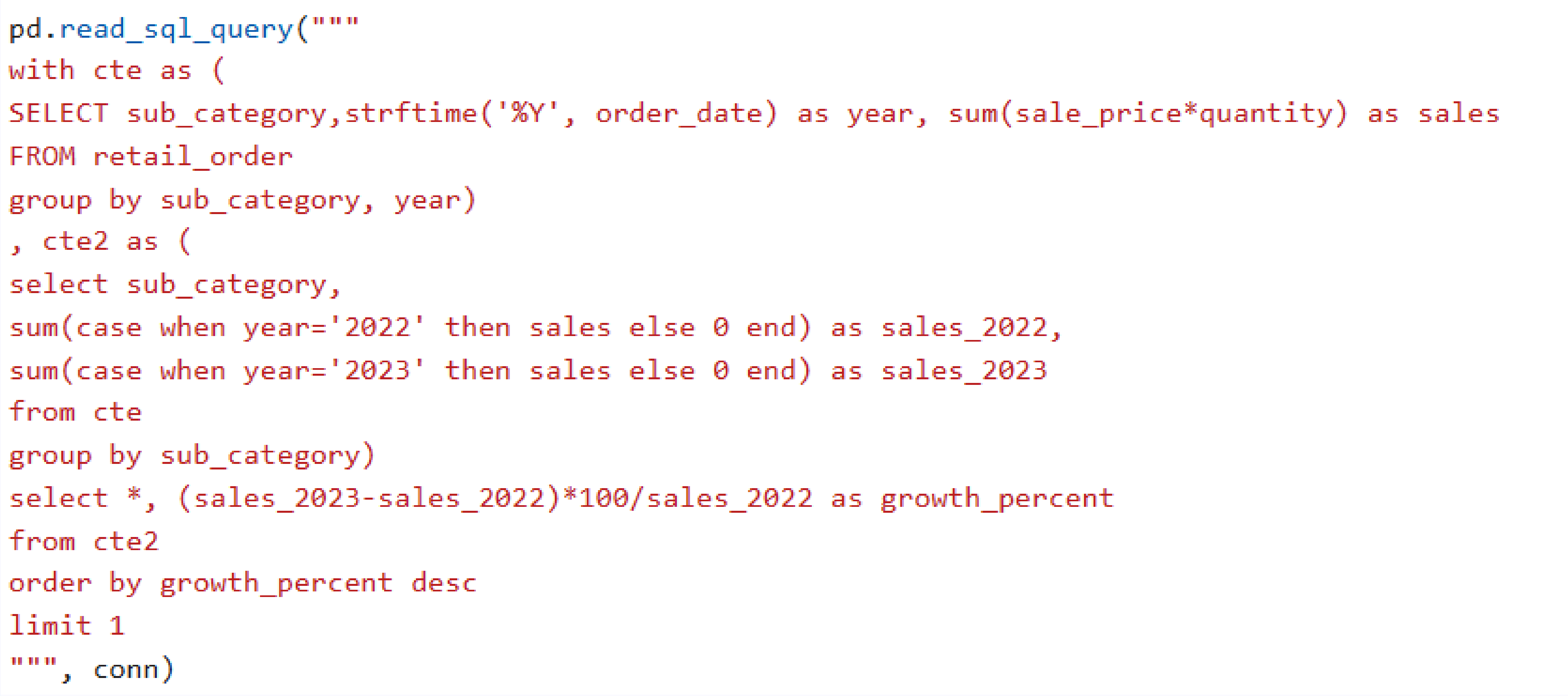


```
pd.read_sql_query("""
with cte as (
select category, strftime('%Y', order_date) as year,strftime('%m', order_date) as month, sum(sale_price*quantity) as sales
from retail_order
group by category,year, month)
select * from (
select *,
row_number() over(partition by category order by sales desc) as rn
from cte) as A
where rn=1
""", conn)
```

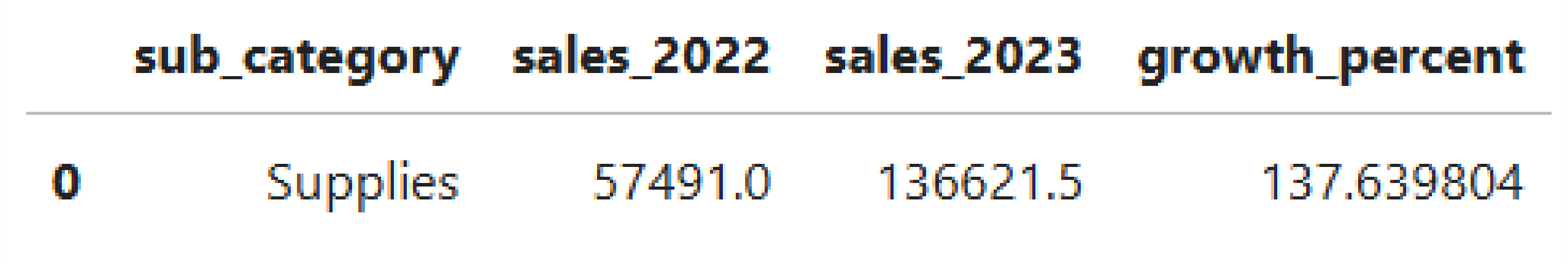


	category	year	month	sales	rn
0	Furniture	2023	08	230523.5	1
1	Office Supplies	2023	02	287244.6	1
2	Technology	2023	10	295586.5	1

WHICH SUB-CATEGORY HAD THE HIGHEST GROWTH BY PROFIT IN 2023 AS COMPARED TO 2022



```
pd.read_sql_query("""
with cte as (
SELECT sub_category,strftime('%Y', order_date) as year, sum(sale_price*quantity) as sales
FROM retail_order
group by sub_category, year)
, cte2 as (
select sub_category,
sum(case when year='2022' then sales else 0 end) as sales_2022,
sum(case when 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 growth_percent
from cte2
order by growth_percent desc
limit 1
""", conn)
```



	sub_category	sales_2022	sales_2023	growth_percent
0	Supplies	57491.0	136621.5	137.639804

THANK YOU