

Presented by Shiksha Lahre

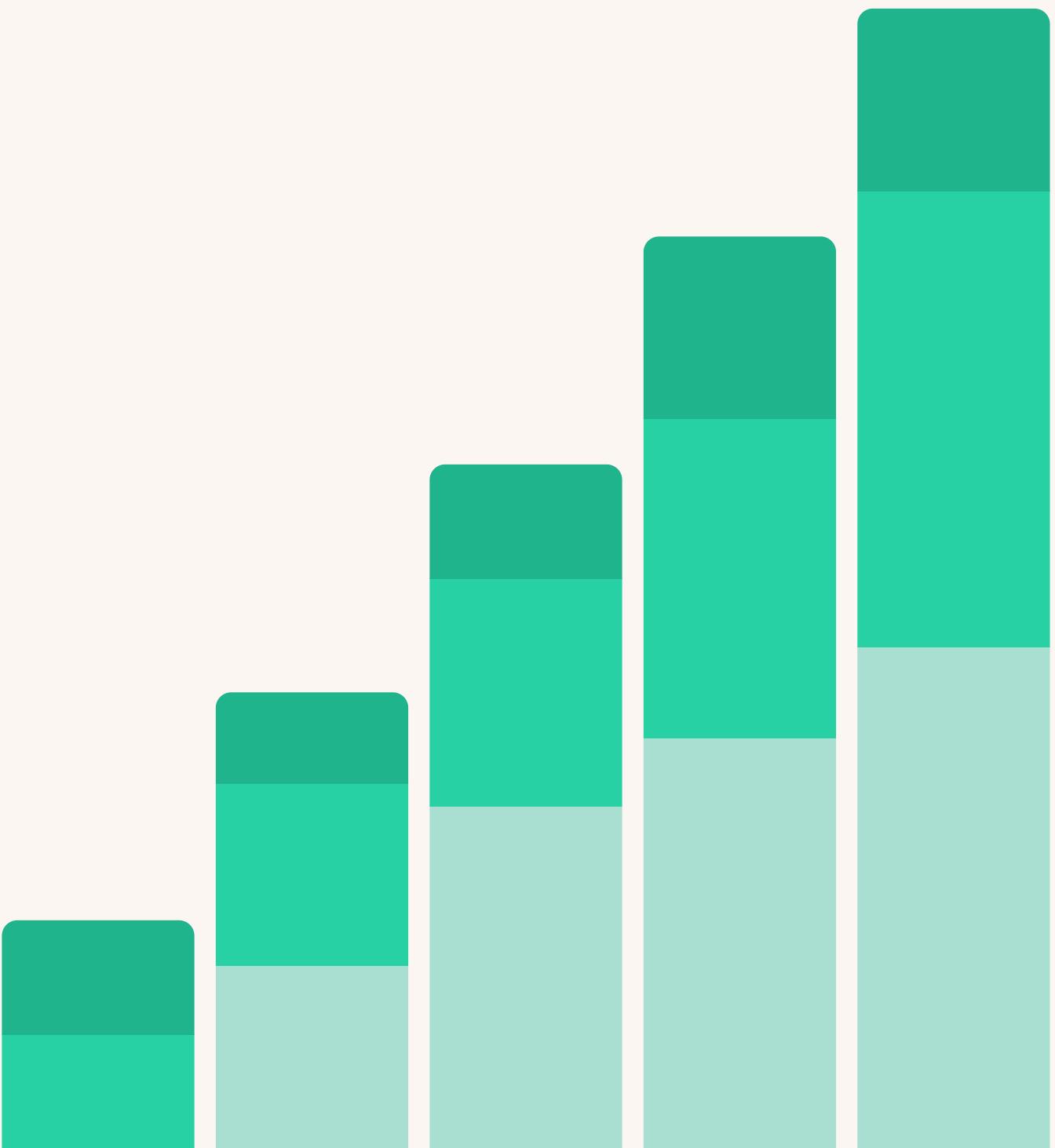
AnalyticsSQL

An SQL Analysis on sales of
electrical appliances

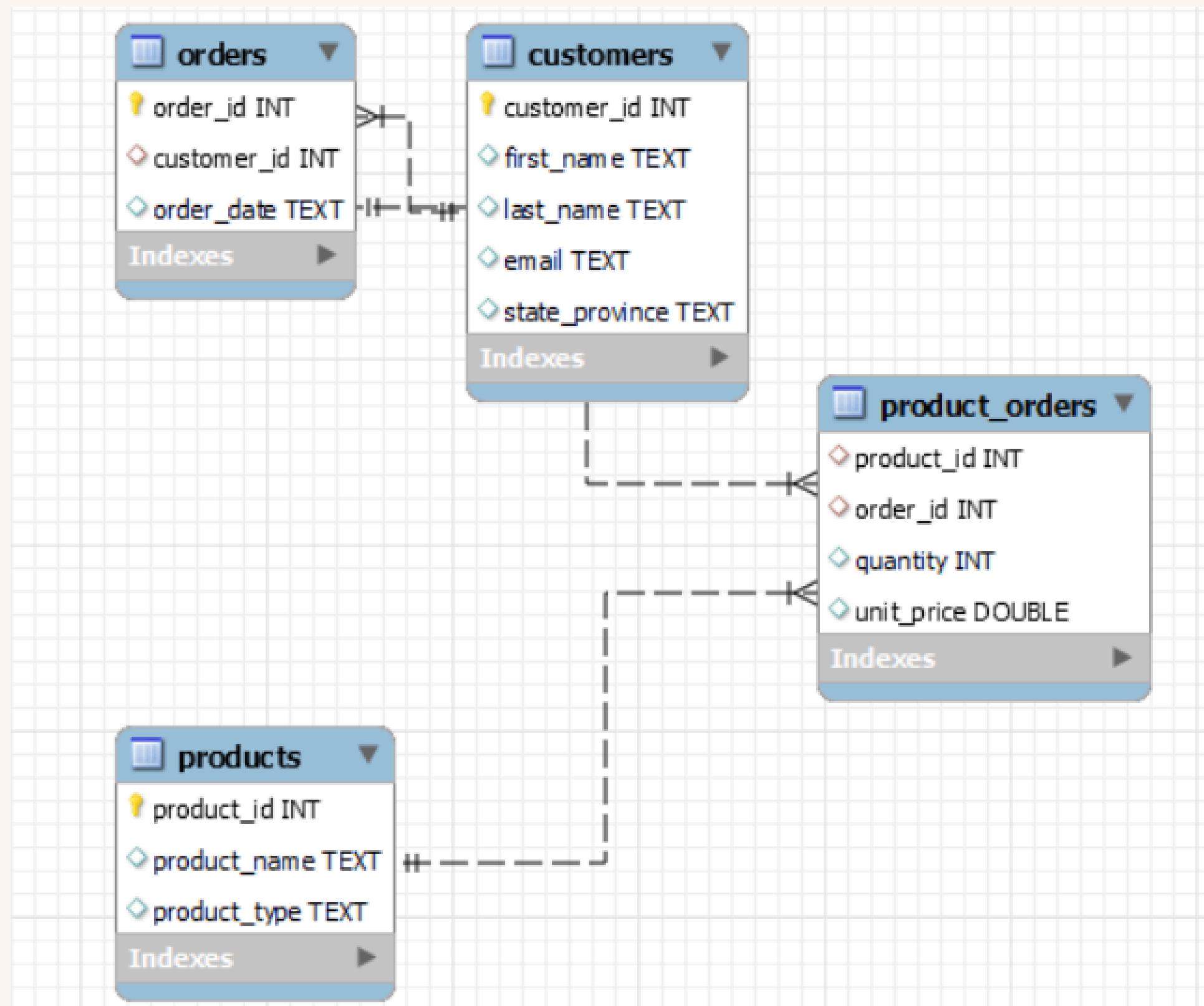


Project Overview

The project aims to analyze sales and revenue data to assess business performance trends over various time periods—daily, monthly, and yearly. Using tables that capture customer details, product information, order transactions, and product orders, the project will calculate revenue based on quantity and price, and provide insights into sales performance by product type. The analysis will focus on identifying trends to determine whether the business is growing or declining over time, helping stakeholders make informed decisions to optimize sales strategies and improve overall business outcomes.



Enhanced Entity-Relationship



Dataset

Table: Customers

customer_id: Unique identifier of a customer

first_name: first name of customer

last-name: last name of customer

email: email of customer

state_province: unique code of customer's province

Table: Products

product_id: Unique identifier of a product

product_name: Unique name of the product

product_type: category of product

Dataset

Table: Orders

order_id: Unique identifier of each order

customer_id: corresponding to customer_id in customers table

order_date: date of product ordered

Table: Product_orders

product_id: corresponding to product_id in products table

order_id: corresponding to order_id in orders table

quantity: number of product purchased

unit_price: price of each product (in dollars)

Constraints in Tables

Raw Dataset of tables has no relationship, hence following constraints has been applied

1) Primary Keys:

- i) customer_id from customers table
- ii) order_id from orders table
- iii) product_id from products table

2) Foreign Key

- i) customer_id from orders references customers(customer_id)
- ii) order_id from product_orders references orders(order_id)
- iii) product_id from product_orders references products(product_id)

Data Cleaning

1) We found mismatched product_id values between product_orders and products, making a foreign key connection impossible, so we isolated, cleaned, and corrected the discrepancies.

- i) Finding product_id in product_orders and not in products table
- ii) Deleting product_id in product_orders where products(product_id) is NULL.
- iii) Making product_id as foreign key references on products(product_id).

2) Abnormality in order_date in orders

- i) Raw table contain date in mm/dd/yyyy format with datatype text, making it difficult to analyse.
- ii) making a new date column where string is converted to date format with STR_TO_DATE
- iii) dropping of old_date column

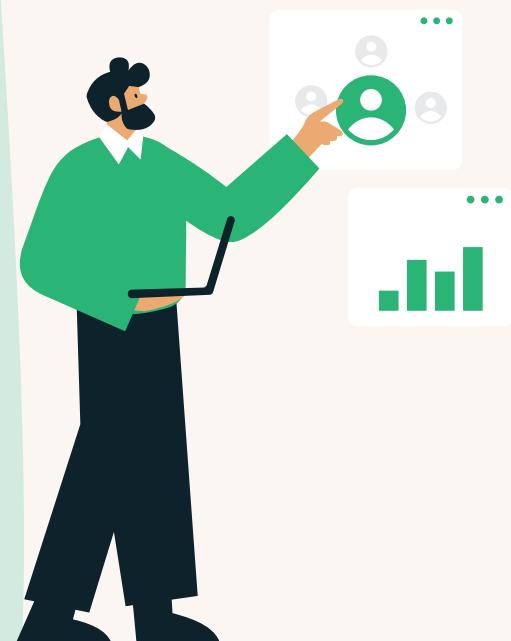
Queries & Output

Query

```
#Total number of orders  
SELECT COUNT(order_id)  
FROM orders;
```

```
#Total number of customers  
SELECT COUNT(customer_id) AS Total_customers  
FROM customers;
```

```
#Total number of product type  
SELECT COUNT(DISTINCT product_type) AS product_types  
FROM products;
```



Output

COUNT(order_id)	300
	▶

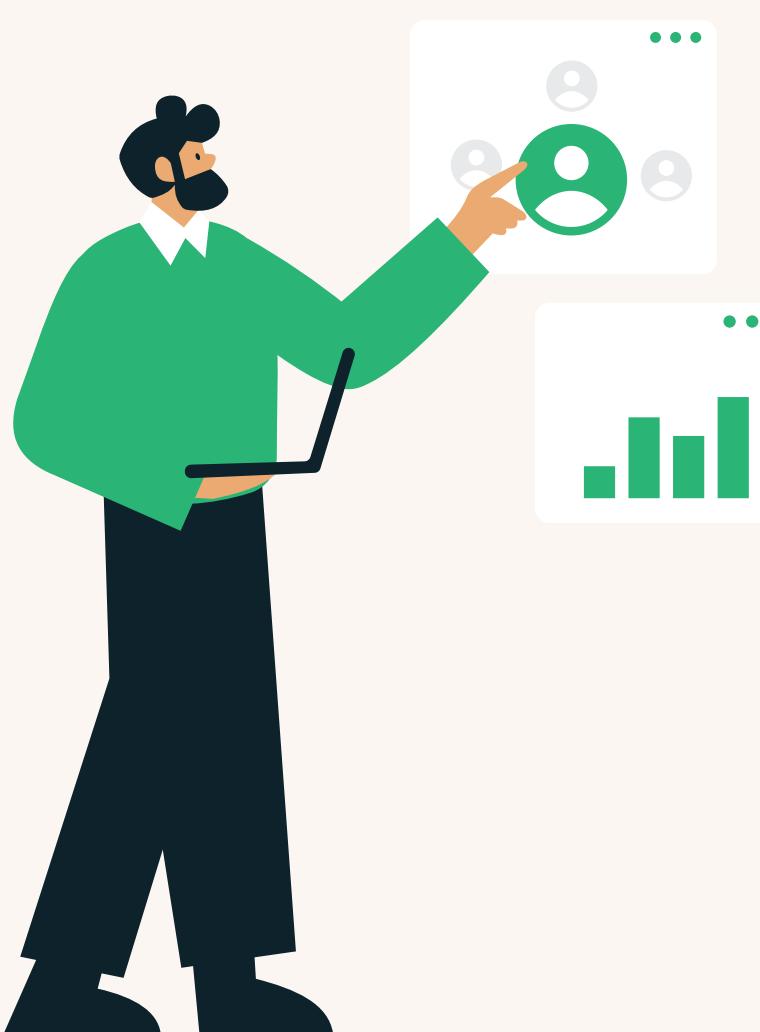
Total_customers	1000
	▶

COUNT(DISTINCT product_type)	12
	▶



Query

```
#Compute Month wise total sales of products  
SELECT DATE_FORMAT(o.order_date, '%Y-%m') AS month,  
       ROUND(SUM(po.quantity * po.unit_price)) AS total_sales  
FROM orders o  
JOIN product_orders po ON o.order_id = po.order_id  
GROUP BY month  
ORDER BY month;
```



Output

	month	total_sales
▶	2019-02	20315
	2019-03	23243
	2019-04	26142
	2019-05	39215
	2019-06	29162
	2019-07	34373
	2019-08	32487
	2019-09	33502
	2019-10	31463
	2019-11	16119
	2019-12	33184
	2020-01	12507
	2020-02	8481

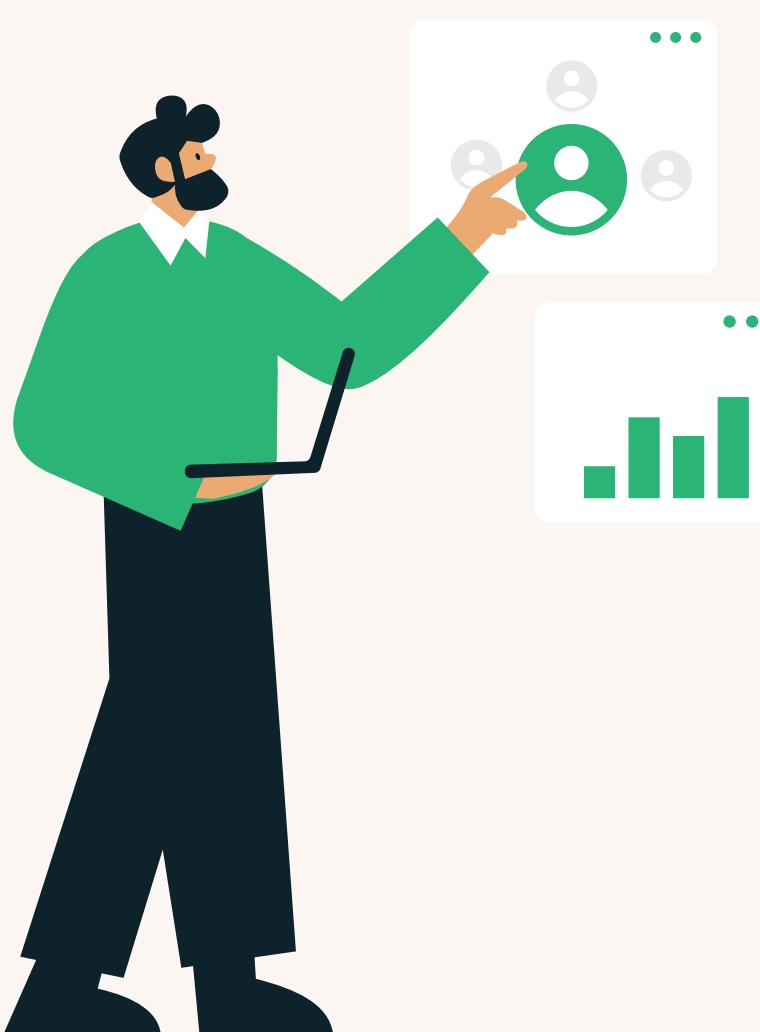


Query

```
#Sales trend over time  
  
SELECT o.order_date,  
       ROUND(SUM(po.quantity * po.unit_price)) AS daily_sales  
  
FROM orders o  
JOIN product_orders po ON o.order_id = po.order_id  
GROUP BY o.order_date  
ORDER BY o.order_date;
```

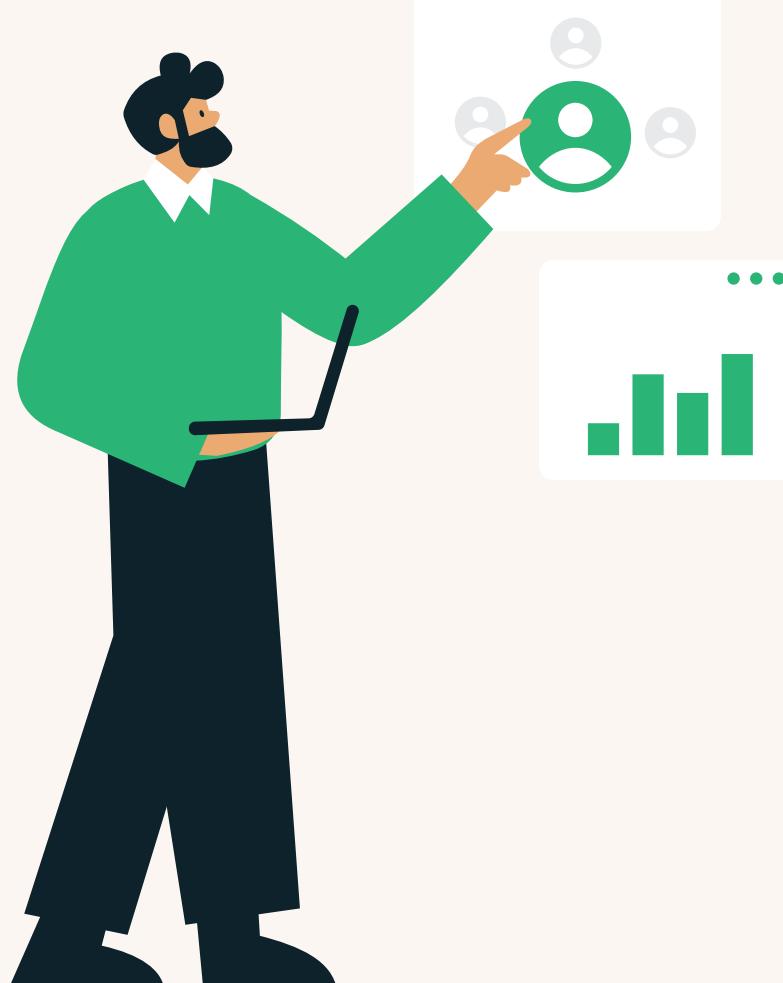
Output

	order_date	daily_sales
1	2019-04-30	3544
2	2019-05-01	2206
3	2019-05-02	2946
4	2019-05-04	3142
5	2019-05-05	3657
6	2019-05-06	3069
7	2019-05-07	517
8	2019-05-08	1403
9	2019-05-11	6334
10	2019-05-12	1292
11	2019-05-13	1372
12	2019-05-18	210
13	2019-05-19	548
14	2019-05-21	3424
15	2019-05-22	2762
16	2019-05-23	2838
17	2019-05-24	1036
18	2019-05-25	777
19	2019-05-26	292
20	2019-05-27	760
21	2019-05-28	631
22	2019-06-02	389
23	2019-06-03	944



Query

```
#Sales each quarter  
  
SELECT YEAR(o.order_date) AS year,  
       QUARTER(o.order_date) AS quarter,  
       ROUND(SUM(po.quantity * po.unit_price)) AS total_sales  
  
FROM orders o  
  
JOIN product_orders po ON o.order_id = po.order_id  
  
GROUP BY year,quarter  
  
ORDER BY year,quarter;
```



Output

	year	quarter	total_sales
▶	2019	1	43558
	2019	2	94519
	2019	3	100362
	2019	4	80766
	2020	1	20988



Query

```
#Sales per year  
  
SELECT YEAR(o.order_date) AS year,  
       COUNT(o.order_id) AS total_orders  
  
FROM orders o  
  
GROUP BY year  
  
ORDER BY year;
```



Output

	year	total_orders
▶	2019	281
	2020	19

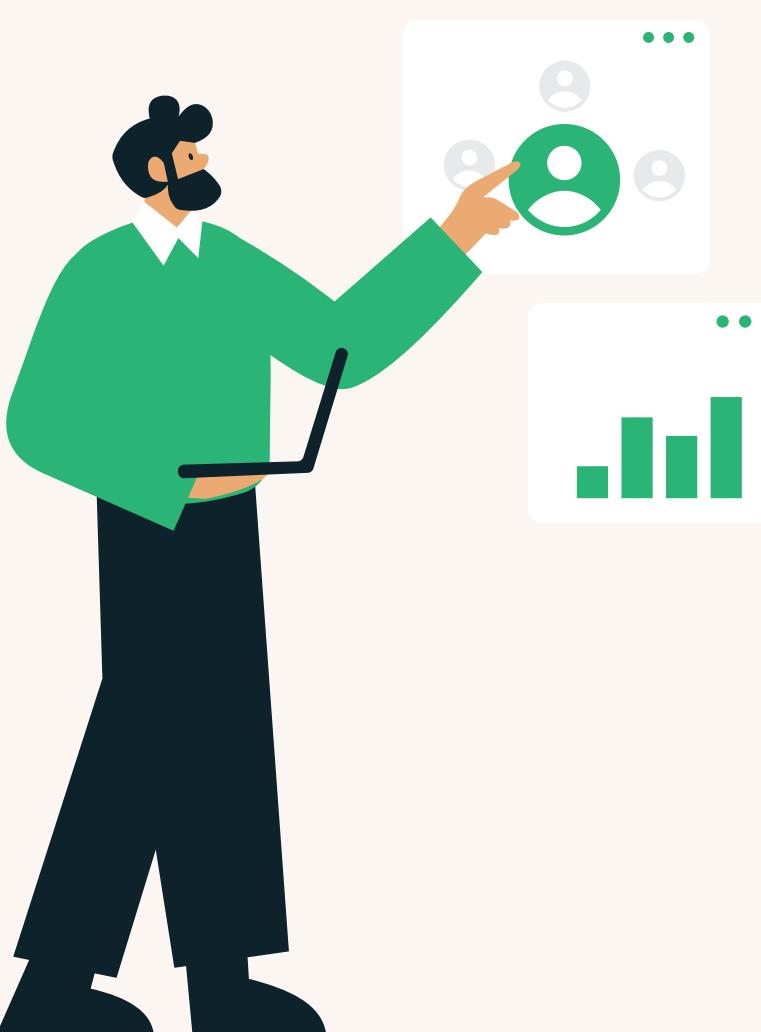


Query

#Monthly average sales

SELECT

```
    DATE_FORMAT(o.order_date, '%Y-%m') AS month,  
    ROUND(SUM(po.quantity * po.unit_price), 2) AS total_monthly_sales  
FROM orders o  
JOIN product_orders po ON o.order_id = po.order_id  
GROUP BY month  
ORDER BY month;
```



Output

month	total_monthly_sales
2019-02	20315.12
2019-03	23242.72
2019-04	26141.98
2019-05	39215.17
2019-06	29161.88
2019-07	34373.28
2019-08	32486.58
2019-09	33501.65
2019-10	31463.28
2019-11	16119.46
2019-12	33183.62
2020-01	12507.33
2020-02	8480.89



Query

#Comparing this month sales to previous month sales

```
SELECT
    month,
    total_sales,
    LAG(total_sales, 1, 0) OVER (ORDER BY month) AS previous_month_sales,
    total_sales - LAG(total_sales, 1, 0) OVER (ORDER BY month) AS month_on_month_change
FROM (SELECT DATE_FORMAT(o.order_date, '%Y-%m') AS month,
            ROUND(SUM(po.quantity * po.unit_price)) AS total_sales
      FROM orders o
      JOIN product_orders po ON o.order_id = po.order_id
     GROUP BY month
   ) AS sales_data
ORDER BY month;
```



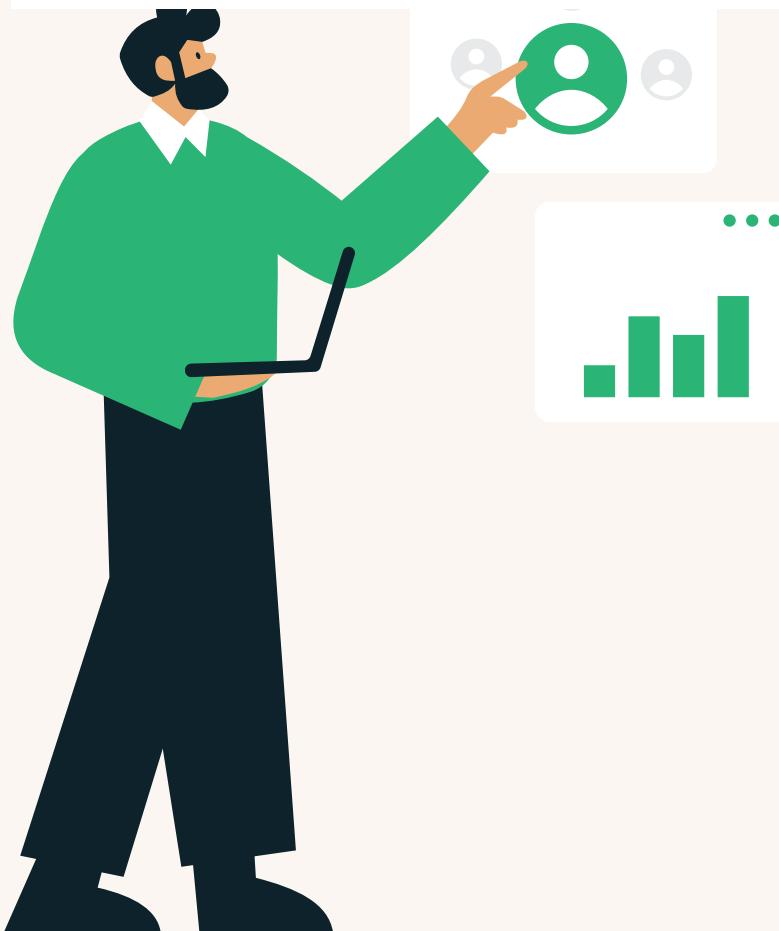
Output

	month	total_sales	previous_month_sales	month_on_month_change
▶	2019-02	20315	0	20315
	2019-03	23243	20315	2928
	2019-04	26142	23243	2899
	2019-05	39215	26142	13073
	2019-06	29162	39215	-10053
	2019-07	34373	29162	5211
	2019-08	32487	34373	-1886
	2019-09	33502	32487	1015
	2019-10	31463	33502	-2039
	2019-11	16119	31463	-15344
	2019-12	33184	16119	17065
	2020-01	12507	33184	-20677
	2020-02	8481	12507	-4026



Query

```
#SHOW PRODUCTS AND THEIR COUNT  
  
SELECT product_type,  
       COUNT(product_type)  
  
FROM products  
  
GROUP BY product_type;
```



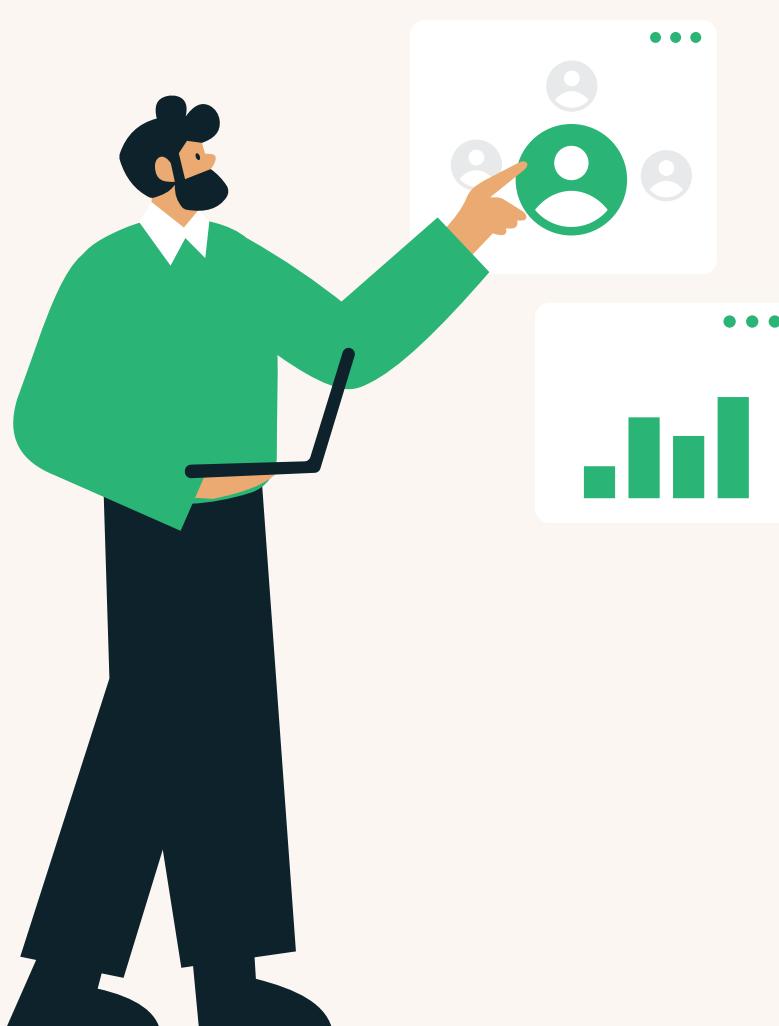
Output

	product_type	COUNT(product_type)
▶	oven	20
	fryer	5
	cooker	14
	steamer	2
	specialty	13
	heater	2
	grinder	1
	roaster	1
	cooker	1
	specialy	1
	grill	2
	coffee	2



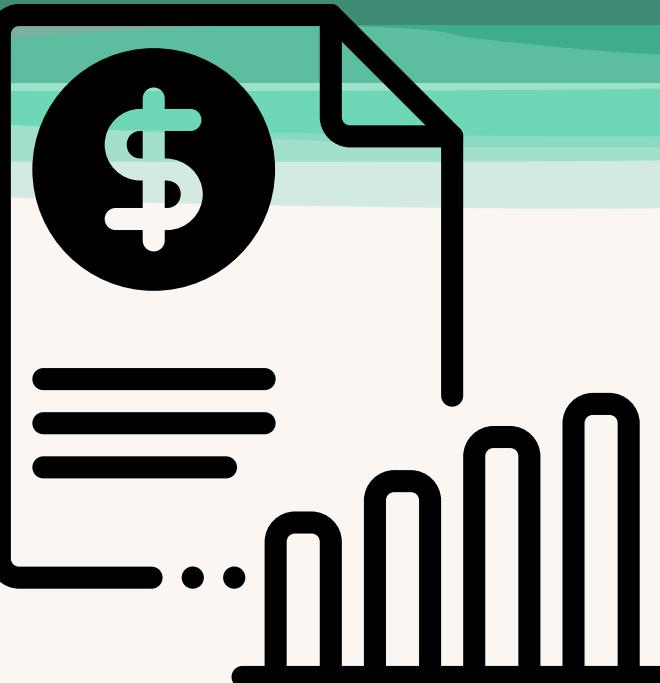
Query

```
#Listing of different product names of each product type and their sales
SELECT YEAR(o.order_date) AS year,
       p.product_type,
       GROUP_CONCAT(DISTINCT p.product_name ORDER BY p.product_name SEPARATOR ', ') AS product_names,
       ROUND(SUM(po.quantity * po.unit_price)) AS total_sales
FROM orders o
JOIN product_orders po ON o.order_id = po.order_id
JOIN products p ON po.product_id = p.product_id
GROUP BY YEAR(o.order_date), p.product_type
ORDER BY YEAR(o.order_date), p.product_type;
```



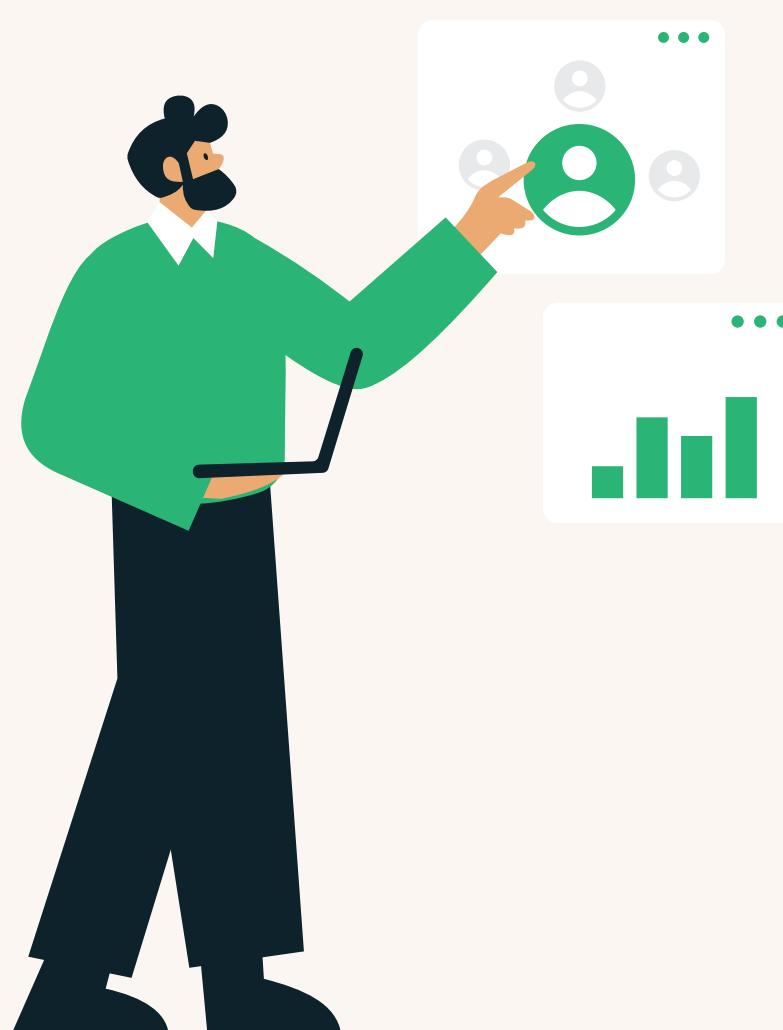
Output

	year	product_type	product_names	total_sales
	2019	fryer	Air fryer, Deep fryer, Pressure fryer, Turkey fr...	22738
	2019	grill	Barbecue grill, Panini sandwich grill	5514
	2019	grinder	Wet grinder	3888
	2019	heater	Brasero (heater), Rocket mass heater	9537
	2019	oven	Beehive oven, Chorkor oven, Clome oven, Com...	110141
	2019	roaster	Corn roaster	6650
	2019	specialty	Bread machine, Chapati maker, Cheesemelter, ...	67421
	2019	specialy	Fufu Machine	4373
	2019	steamer	Combi steamer, Food steamer	7274
	2020	cooker	Fire pot	257
	2020	coffee	Coffee pot	1870
	2020	cooker	Butane torch, Electric cooker, Hot plate, Instant...	3576
	2020	fryer	Air fryer, Pressure fryer	1166
	2020	heater	Rocket mass heater	453
	2020	oven	Beehive oven, Chorkor oven, Clome oven, Com...	9076
	2020	specialty	Bread machine, Cheesemelter, Chocolatera, Pa...	3286
	2020	steamer	Combi steamer	1304



Query

```
#Total sales by product type
SELECT p.product_type,
       ROUND(SUM(po.quantity * po.unit_price)) AS total_sales
FROM product_orders po
JOIN products p ON po.product_id = p.product_id
GROUP BY p.product_type
ORDER BY total_sales DESC;
```



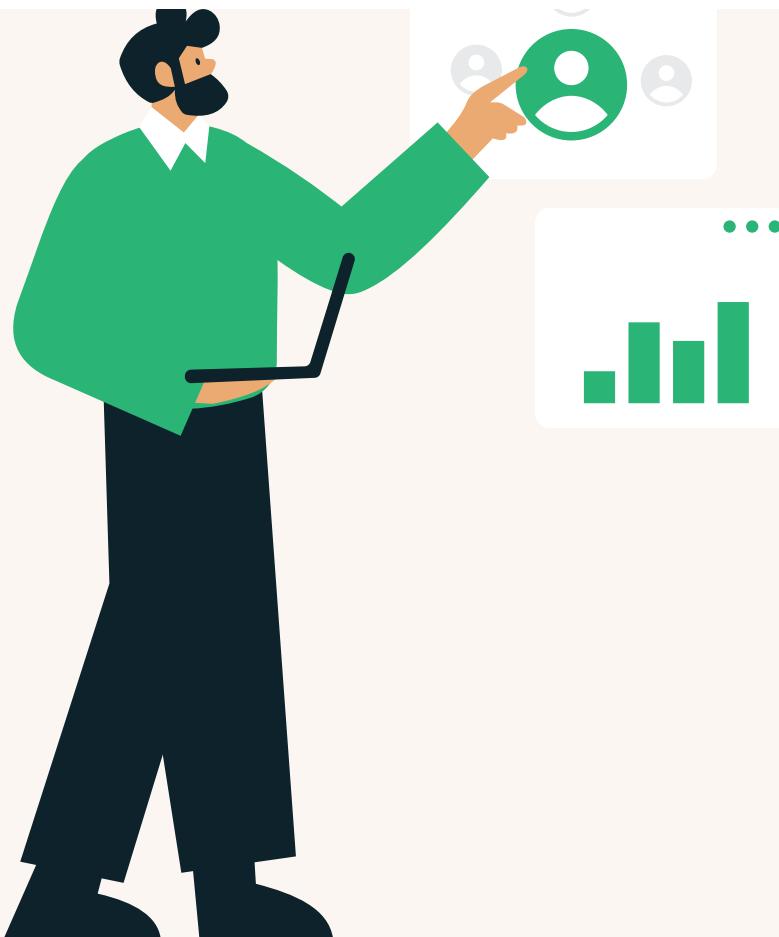
Output

	product_type	total_sales
▶	oven	119217
	specialty	70708
	cooker	69293
	fryer	23905
	heater	9990
	coffee	9284
	cooker	8795
	steamer	8578
	roaster	6650
	grill	5514
	specialy	4373
	grinder	3888



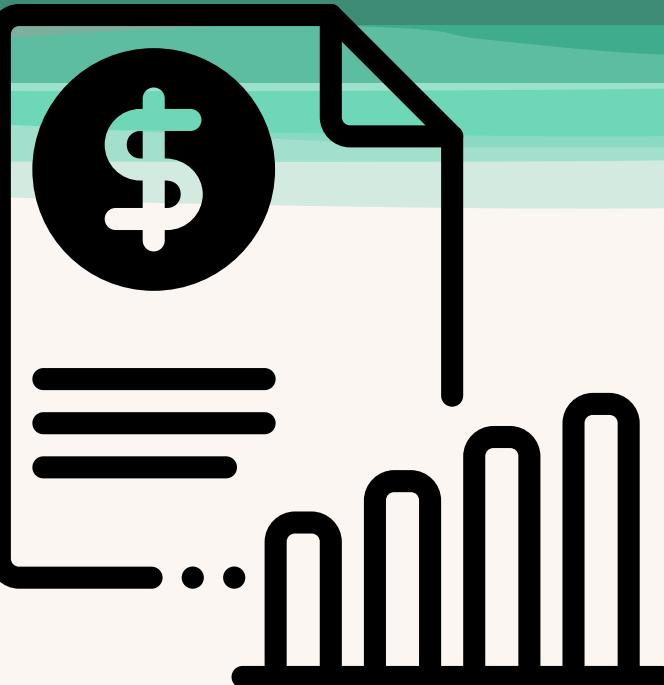
Query

```
#Numbers of order per product type  
SELECT p.product_type,  
       COUNT(DISTINCT o.order_id) AS number_of_orders  
FROM product_orders po  
JOIN products p ON po.product_id = p.product_id  
JOIN orders o ON po.order_id = o.order_id  
GROUP BY p.product_type  
ORDER BY number_of_orders DESC;
```



Output

	product_type	number_of_orders
▶	oven	179
	cooker	139
	specialty	113
	fryer	48
	heater	21
	coffee	18
	steamer	18
	grill	17
	cooker	16
	grinder	14
	roaster	14
	specialy	8



Query

```
#Quaterly sales by product type
SELECT YEAR(o.order_date) AS year,
       QUARTER(o.order_date) AS quarter,
       p.product_type,
       ROUND(SUM(po.quantity * po.unit_price)) AS total_sales
FROM orders o
JOIN product_orders po ON o.order_id = po.order_id
JOIN products p ON po.product_id = p.product_id
GROUP BY year,quarter,p.product_type
ORDER BY year,quarter,p.product_type;
```



Output

	product_type	number_of_orders
▶	oven	179
	cooker	139
	specialty	113
	fryer	48
	heater	21
	coffee	18
	steamer	18
	grill	17
	cooker	16
	grinder	14
	roaster	14
	specialy	8



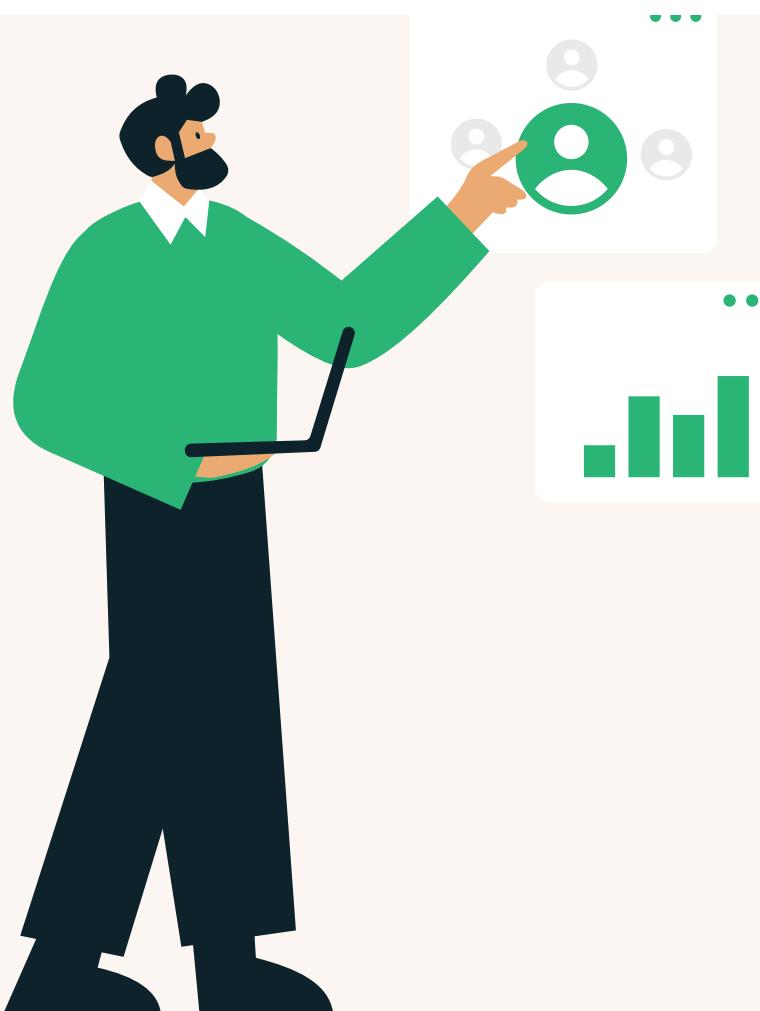
Query

#Monthly sales trend by product type

```
SELECT DATE_FORMAT(o.order_date, '%Y-%m') AS month,  
       p.product_type,  
       ROUND(SUM(po.quantity * po.unit_price)) AS total_sales  
FROM orders o  
JOIN product_orders po ON o.order_id = po.order_id  
JOIN products p ON po.product_id = p.product_id  
GROUP BY month,p.product_type  
ORDER BY month,p.product_type;
```

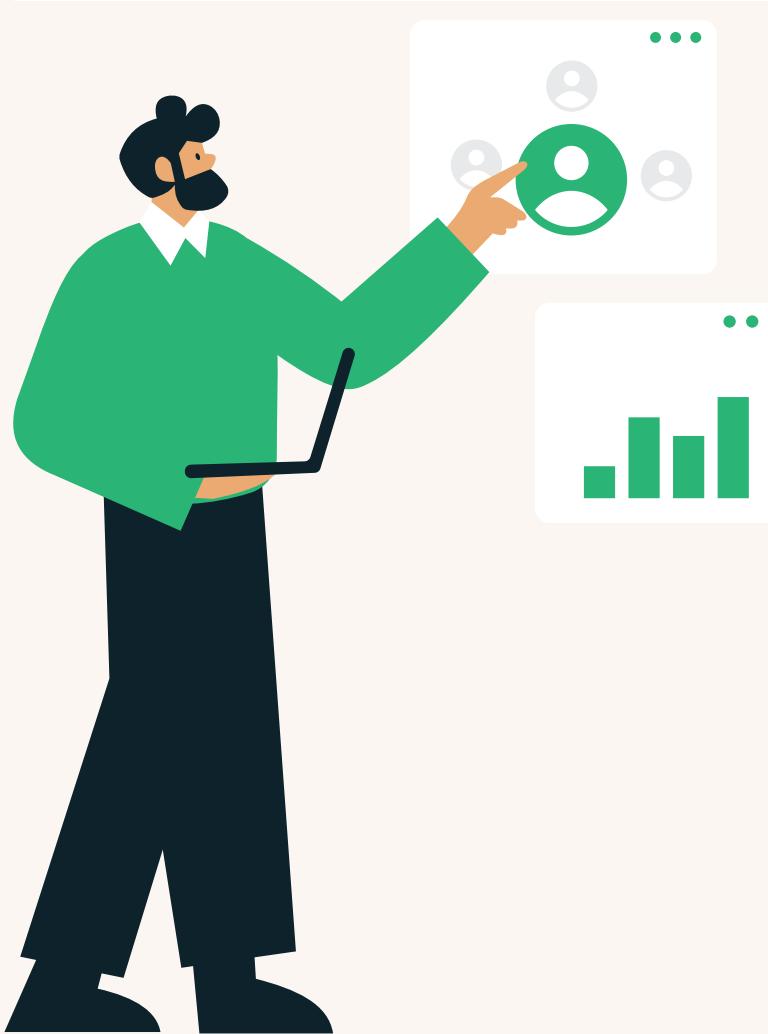
Output

	month	product_type	total_sales
2019-02	heater	1333	
2019-02	oven	6730	
2019-02	specialty	7514	
2019-02	steamer	275	
2019-03	cooker	1173	
2019-03	cooker	3829	
2019-03	fryer	1077	
2019-03	grill	264	
2019-03	grinder	444	
2019-03	heater	821	
2019-03	oven	7214	
2019-03	specialty	7360	
2019-03	specialy	391	
2019-03	steamer	671	
2019-04	cooker	1727	
2019-04	coffee	656	
2019-04	cooker	5484	
2019-04	fryer	1706	
2019-04	grill	1333	
2019-04	oven	8882	
2019-04	specialty	5819	
2019-04	specialy	535	
2019-05	coffee	571	
2019-05	cooker	6832	
2019-05	fryer	5185	



Query

```
#Percentage distribution of Sales by prout type
SELECT p.product_type,
       ROUND(SUM(po.quantity * po.unit_price)) AS total_sales,
       ROUND(SUM(po.quantity * po.unit_price) / (SELECT SUM(po2.quantity * po2.unit_price)
                                                    FROM product_orders po2) * 100,2) AS percentage_of_total
FROM product_orders po
JOIN products p ON po.product_id = p.product_id
GROUP BY p.product_type
ORDER BY total_sales DESC;
```



Output

	product_type	total_sales	percentage_of_total
▶	oven	119217	35.04
	specialty	70708	20.78
	cooker	69293	20.37
	fryer	23905	7.03
	heater	9990	2.94
	coffee	9284	2.73
	cooker	8795	2.59
	steamer	8578	2.52
	roaster	6650	1.95
	grill	5514	1.62
	specialy	4373	1.29
	grinder	3888	1.14



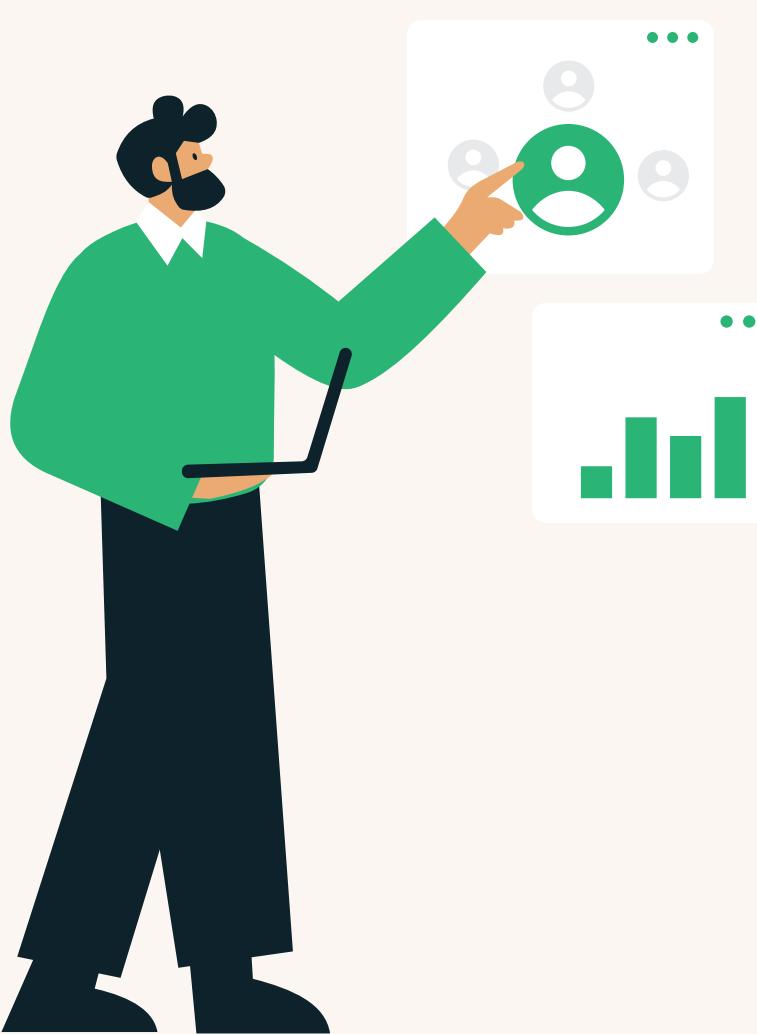
Query

```
#Year over year sales growth by product type
SELECT current.year, current.product_type,ROUND(current.total_sales) AS current_year_sales,
       ROUND(COALESCE(previous.total_sales, 0)) AS previous_year_sales,
       ROUND((current.total_sales - COALESCE(previous.total_sales, 0)) /
COALESCE(previous.total_sales, 1) * 100,2) AS growth_rate
FROM
  (SELECT YEAR(o.order_date) AS year, p.product_type, SUM(po.quantity * po.unit_price) AS total_sales
   FROM orders o
   JOIN product_orders po ON o.order_id = po.order_id
   JOIN products p ON po.product_id = p.product_id
   GROUP BY YEAR(o.order_date),p.product_type
  ) AS current
LEFT JOIN
  (SELECT YEAR(o.order_date) AS year, p.product_type, SUM(po.quantity * po.unit_price) AS total_sales
   FROM orders o
   JOIN product_orders po ON o.order_id = po.order_id
   JOIN products p ON po.product_id = p.product_id
   GROUP BY YEAR(o.order_date),p.product_type
  ) AS previous
  ON current.product_type = previous.product_type
  AND current.year = previous.year + 1
ORDER BY current.year,current.product_type;
```





Output

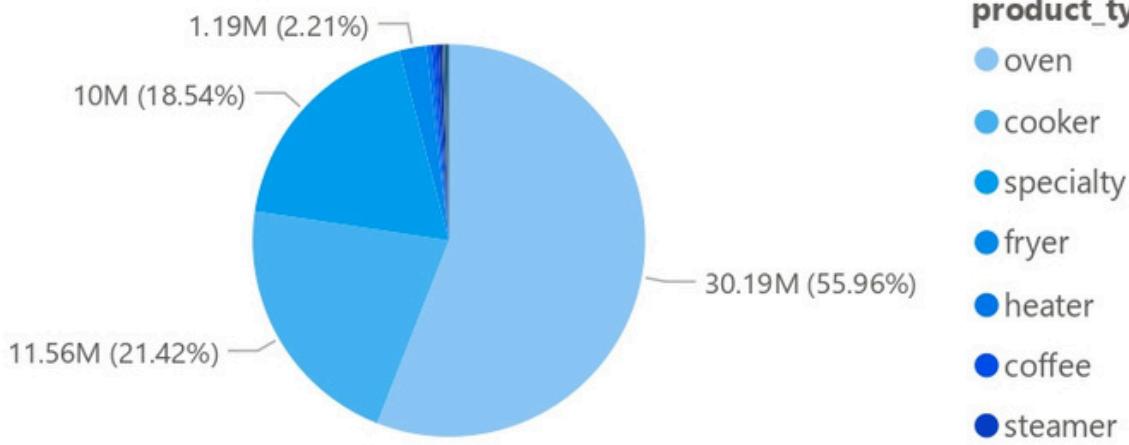


	year	product_type	current_year_sales	previous_year_sales	growth_rate
▶	2019	cooker	8538	0	853796
	2019	coffee	7414	0	741432
	2019	cooker	65718	0	6571762
	2019	fryer	22738	0	2273833
	2019	grill	5514	0	551385
	2019	grinder	3888	0	388798
	2019	heater	9537	0	953670
	2019	oven	110141	0	11014071
	2019	roaster	6650	0	664950
	2019	specialty	67421	0	6742147
	2019	specialy	4373	0	437274
	2019	steamer	7274	0	727356
	2020	cooker	257	8538	-96.99
	2020	coffee	1870	7414	-74.78
	2020	cooker	3576	65718	-94.56
	2020	fryer	1166	22738	-94.87
	2020	heater	453	9537	-95.25
	2020	oven	9076	110141	-91.76
	2020	specialty	3286	67421	-95.13
	2020	steamer	1304	7274	-82.07

Data Visualization

Electronics Sales Insights

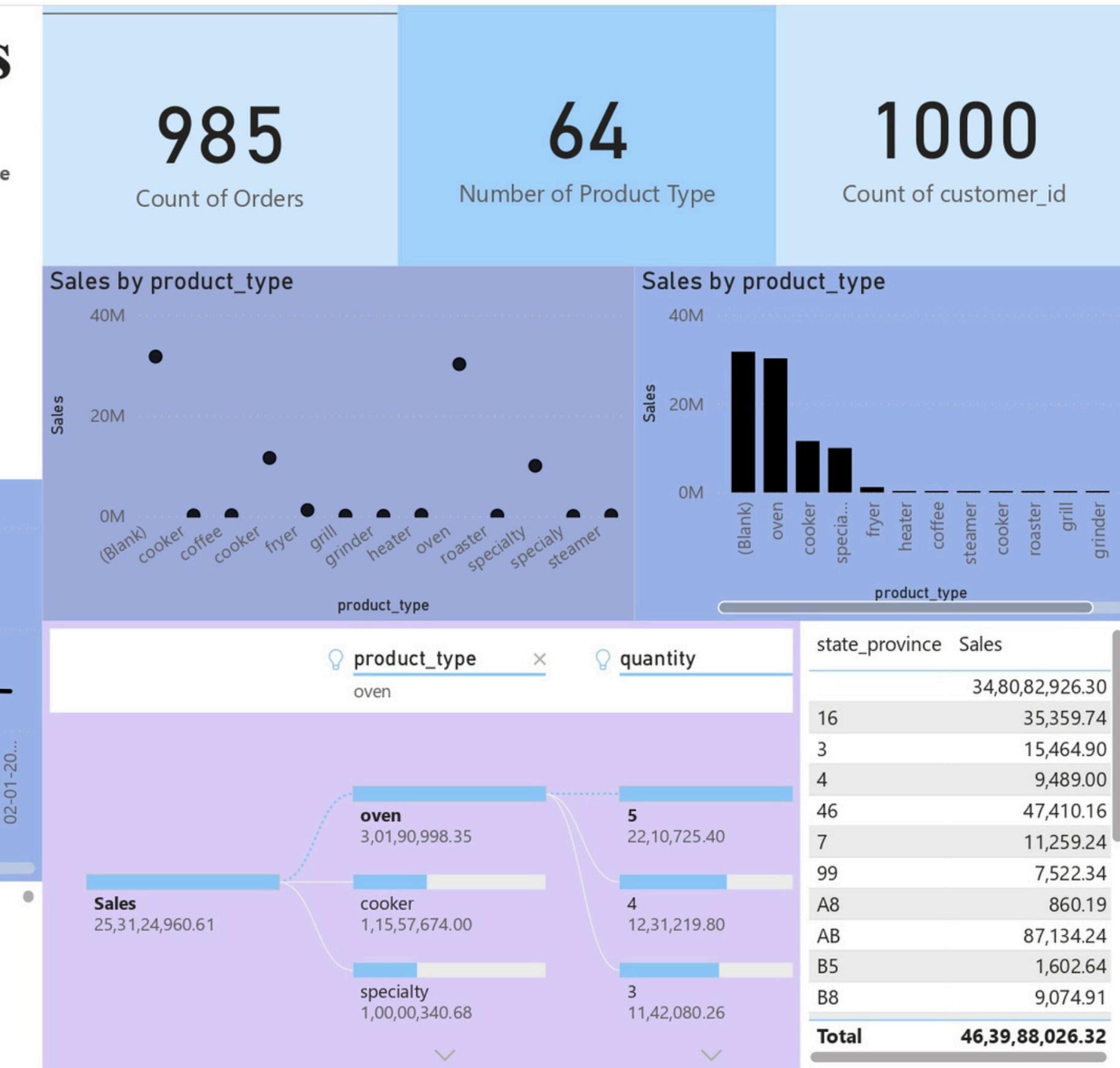
Sales by product_type



Sales by order_date



customer_id	Sum of quantity	product_type	Sales
117	1	grill	20.18
24	1	specialty	21.15
52	1	cooker	21.63
Total	2892		46,39,88,026.32



Conclusion

Based on the analysis, it was found that the maximum sales were driven by a variety of ovens, with the highest-priced item also being an oven. Despite this, the sales during the analyzed time period have shown a significant decline. This suggests that while certain high-value products like ovens are performing well, the overall business is experiencing a downward trend in sales, highlighting the need for strategic adjustments to reverse this decline and improve overall revenue performance.

OR IN SIMPLE WORDS:

Despite turning up the heat with top-selling ovens, our sales still took a shocking downturn—looks like we need to plug into a new strategy to keep the business current!

Laughtronical



Why did the electrician become a successful salesman?

Because he knew how to make a shocking deal!

Thankyou!