

SQL CASE STUDY

DATA IN MOTION TINY SHOP SALES



DATA IN MOTION

Created by: Dr.Neha Chaudhry



Data in Motion

Presentation Outline



Introduction



Tools used



About Dataset



Analysis using
MySQL



Insights and
Implications

Introduction

This case study challenge, conducted by Data in Motion, LLC, focuses on analyzing the dataset from the tiny shop.

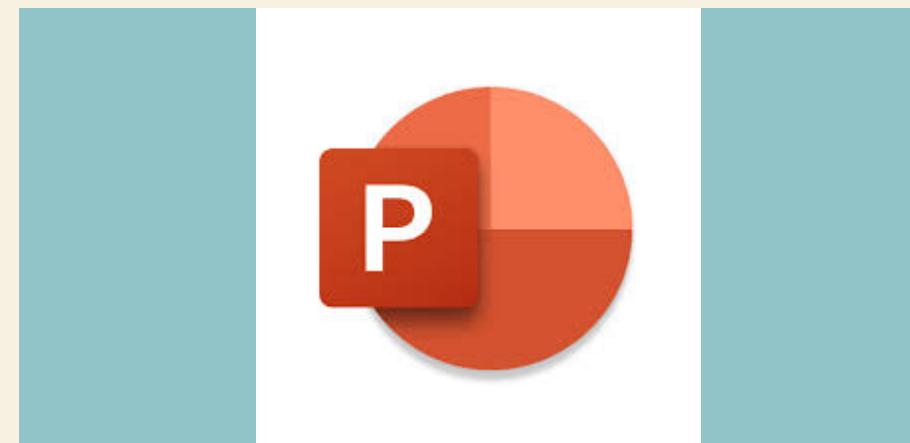
The dataset from the tiny shop offers stakeholders an extensive overview of the shop's operations, encompassing crucial information **on products, customers, orders, and revenue**. With this dataset, stakeholders can delve into key aspects such as identifying the highest-priced product and discerning the customers who contribute the most through frequent orders. They can also gain insights into revenue distribution across different products and pinpoint the day that generates the highest revenue.

Furthermore, the dataset allows stakeholders to examine specific details about individual customers, conduct in-depth product quantity analysis for inventory optimization, and understand overall order totals. **By leveraging these comprehensive insights, stakeholders can make data-driven decisions to optimize pricing strategies, enhance customer experiences, and drive remarkable business growth.**





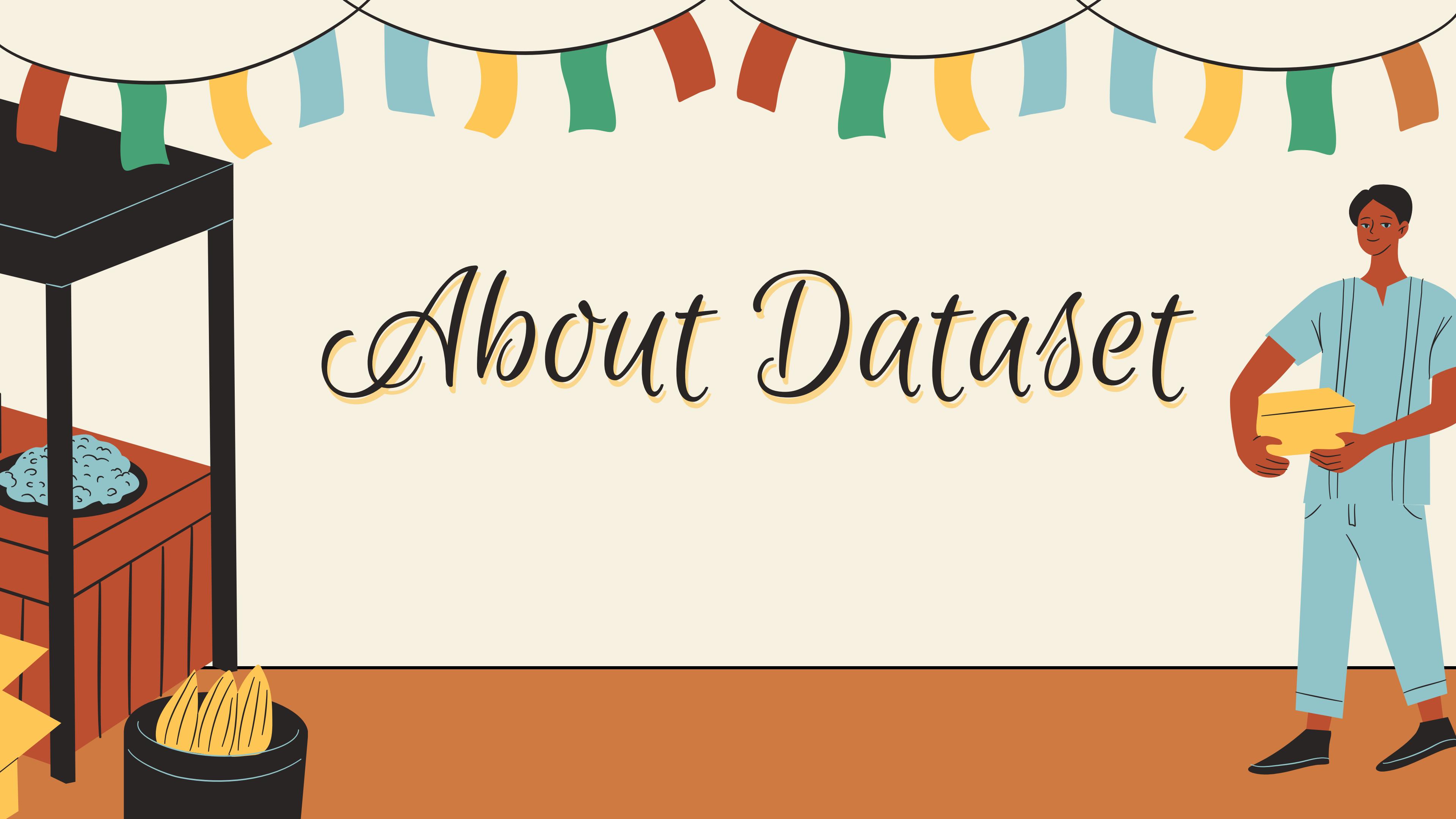
Tools Used



Essential SQL topics for effectively addressing all the questions

- Basic aggregations
- CASE WHEN statements
- Window Functions
- Joins
- Date time functions
- CTEs



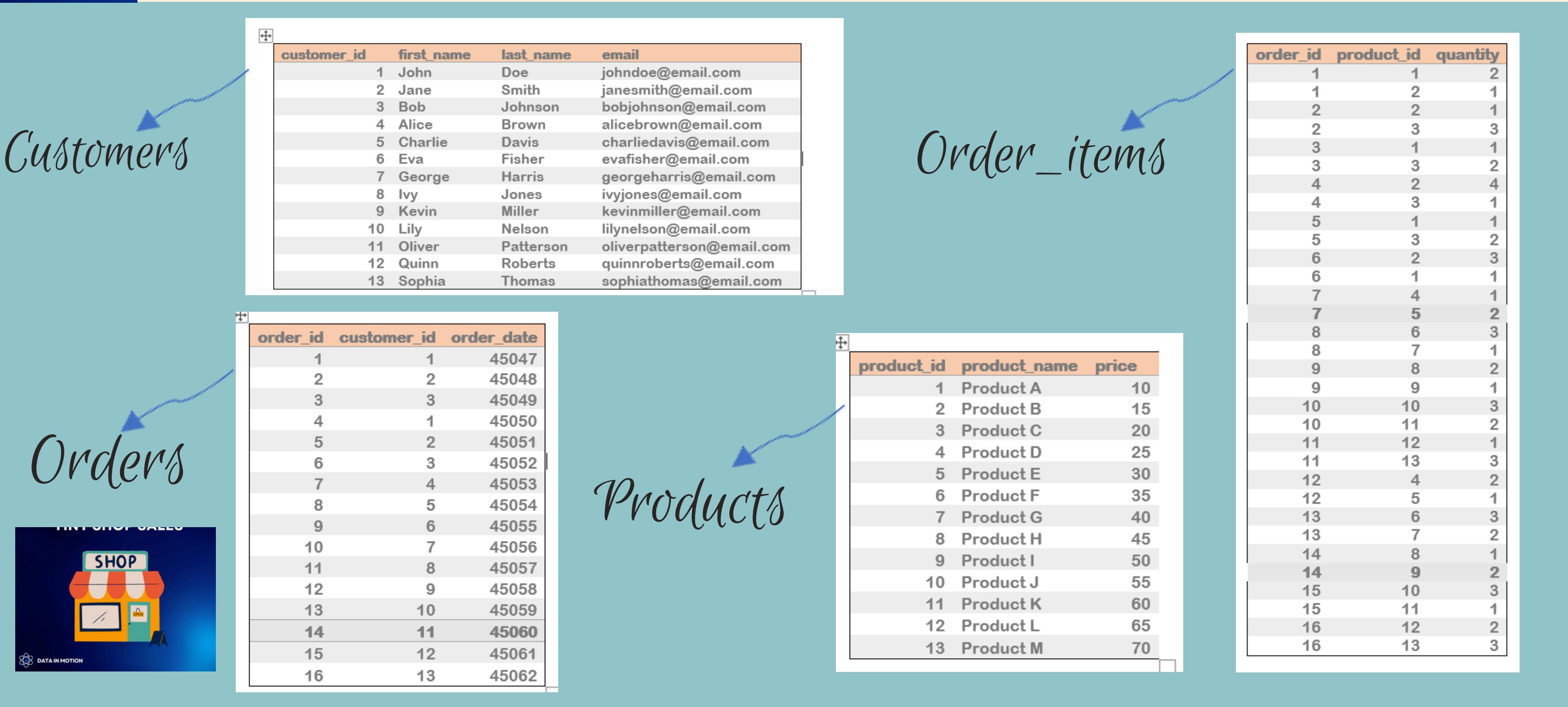


About Dataset





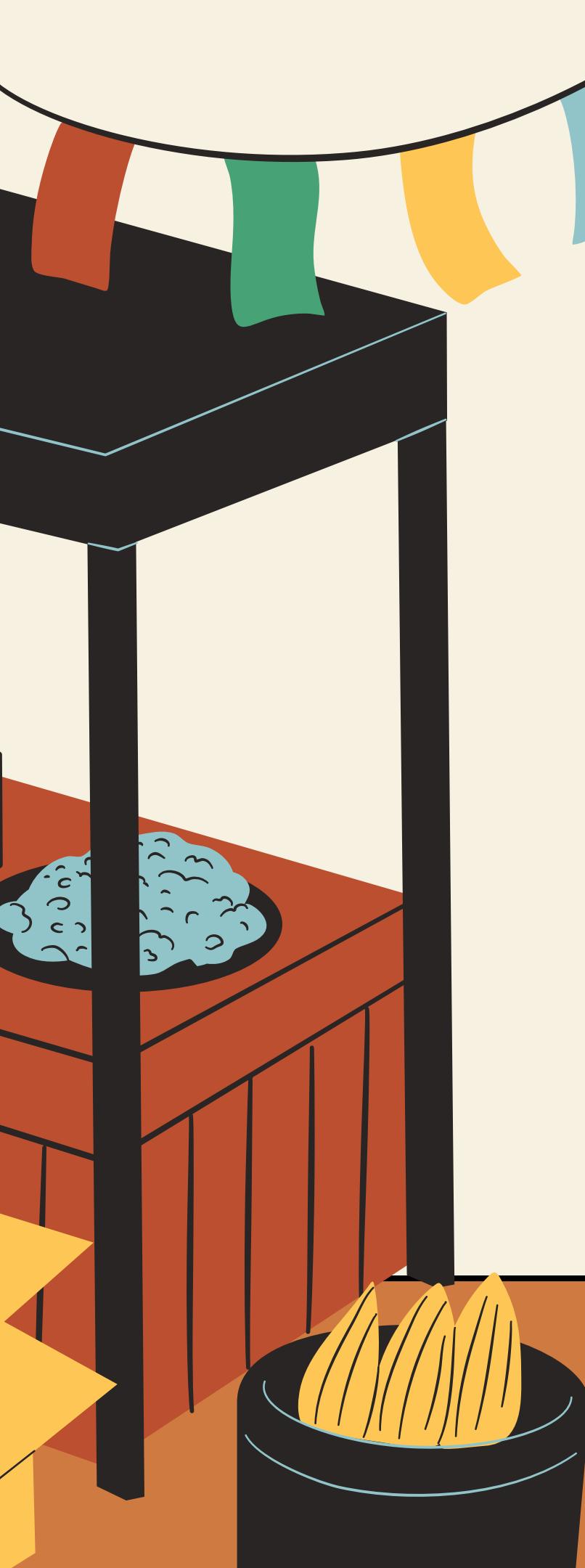
DATASET



Questions



1. Which product has the highest price? Only return a single row.
2. Which customer has made the most orders?
3. What's the total revenue per product?
4. Find the day with the highest revenue.
5. Find the first order (by date) for each customer.
6. Find the top 3 customers who have ordered the most distinct products
7. Which product has been bought the least in terms of quantity?
8. What is the median order total?
9. For each order, determine if it was 'Expensive' (total over 300), 'Affordable' (total over 100), or 'Cheap'.
10. Find customers who have ordered the product with the highest price.

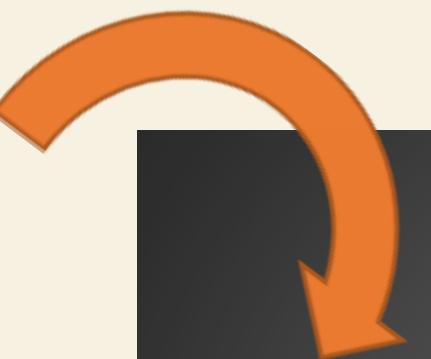


Insights using mysql



1. Which product has the highest price? Only return a single row.

```
SELECT product_name,price  
FROM products  
ORDER BY price DESC  
LIMIT 1;
```



Result Grid | Filter Rows:

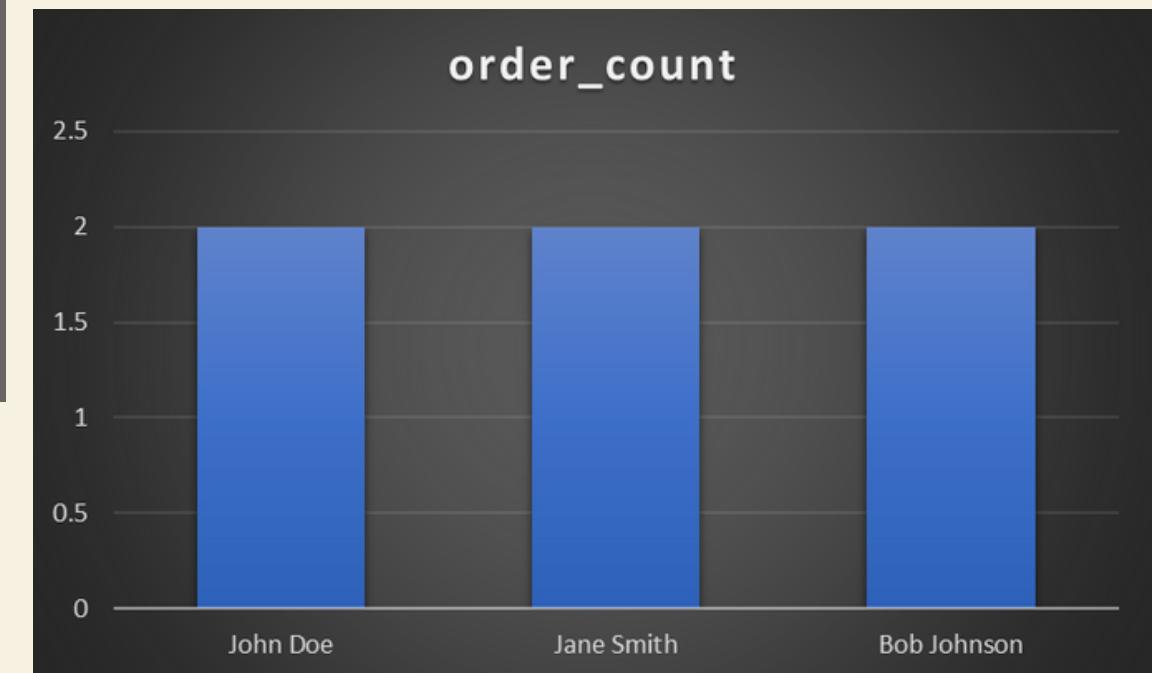
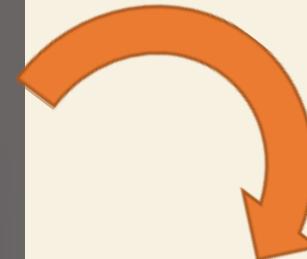
	product_name	price
▶	Product M	70



2. Which customer has made the most orders?

...

```
SELECT c.customer_id,  
       CONCAT(c.first_name, ' ', c.last_name) AS full_name,  
       COUNT(*) AS order_count  
  FROM customers c  
 JOIN orders o ON c.customer_id = o.customer_id  
 GROUP BY c.customer_id, full_name  
 ORDER BY order_count DESC  
 LIMIT 3;
```



	customer_id	full_name	order_count
▶	1	John Doe	2
	2	Jane Smith	2
	3	Bob Johnson	2

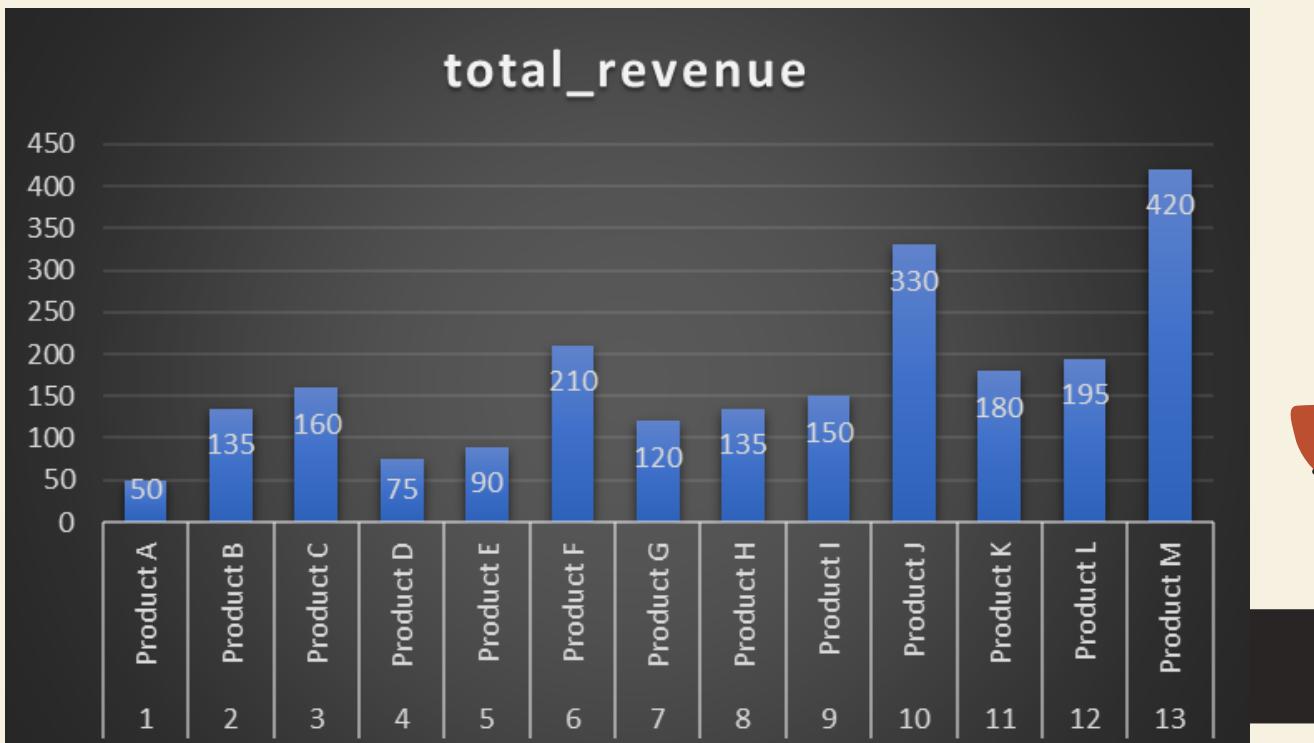
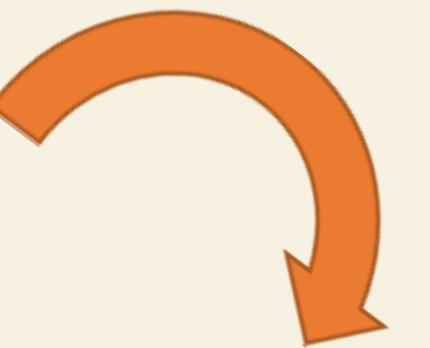


Insights:- The customer with the most orders is 'John Doe' with a total of 2 orders.
The other two customers, 'Jane Smith' and 'Bob Johnson', also have 2 orders each.

3. What's the total revenue per product?

```
WITH revenue_per_product AS (
  SELECT ot.product_id, p.product_name,
  ot.quantity, p.price,
  (ot.quantity * p.price) AS revenue
  FROM products p
  INNER JOIN order_items ot USING (product_id)
)
SELECT product_id, product_name, SUM(revenue) AS total_revenue
FROM revenue_per_product
GROUP BY product_id, product_name;
```

product_id	product_name	total_revenue
1	Product A	50
2	Product B	135
3	Product C	160
4	Product D	75
5	Product E	90
6	Product F	210
7	Product G	120
8	Product H	135
9	Product I	150
10	Product J	330
11	Product K	180
12	Product L	195
13	Product M	420



Insights:-The highest revenue-generating product is Product M with \$420, indicating its popularity and potential profitability. Product J follows closely with \$330 in revenue, while Product D has the lowest revenue of \$75.

4. Find the day with the highest revenue.

```
WITH high_rev_day AS(SELECT *,(ot.quantity*p.price)AS total  
FROM order_items ot  
INNER JOIN orders o USING(order_id)  
INNER JOIN products p USING(product_id))  
SELECT order_date,dayname(order_date) AS name_of_day,  
SUM(total) AS day_total  
FROM high_rev_day  
GROUP BY order_date,name_of_day  
ORDER BY 3 DESC  
LIMIT 1;
```



A	B	C
order_date	name_of_day	day_total
16-05-2023	Tuesday	340

Insights:- The day with the highest revenue is Tuesday, May 16, 2023, with a total revenue of \$340.



5. Find the first order (by date) for each customer.

```
WITH first_order AS(
    SELECT customer_id,order_date,order_id,RANK()
    OVER(PARTITION BY customer_id ORDER BY order_date ASC) AS rnk
    FROM orders
)
SELECT customer_id,order_date
FROM first_order
WHERE rnk = 1;
```



customer_id	order_date
1	01-05-2023
2	02-05-2023
3	03-05-2023
4	07-05-2023
5	08-05-2023
6	09-05-2023
7	10-05-2023
8	11-05-2023
9	12-05-2023
10	13-05-2023
11	14-05-2023
12	15-05-2023
13	16-05-2023

Insights:- The first orders for each customer were placed on various dates in May 2023, indicating individual customer engagement at different times.



6. Find the top 3 customers who have ordered the most distinct products

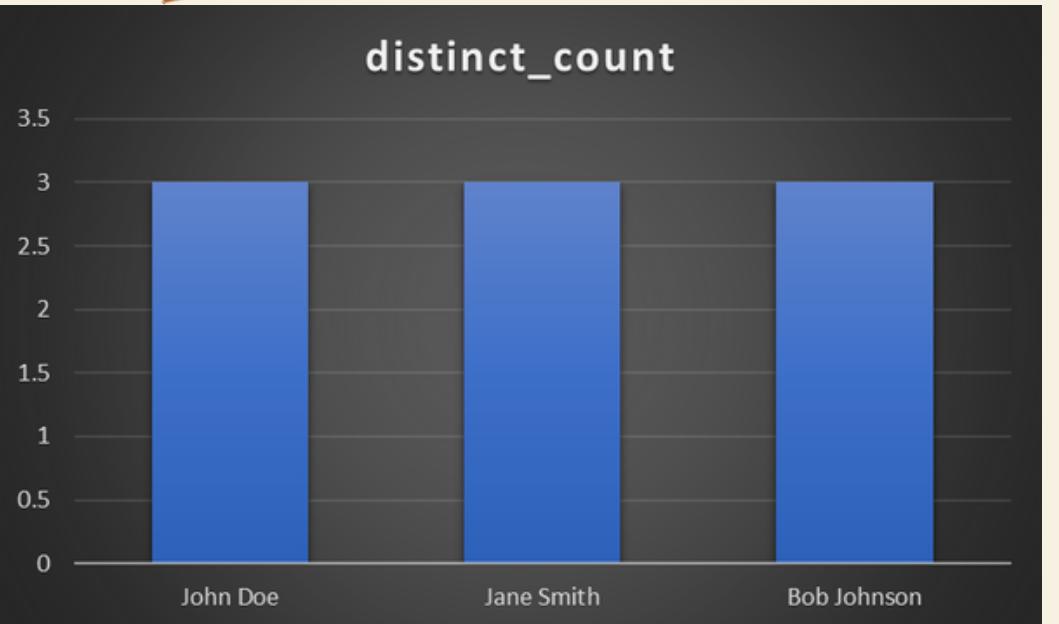
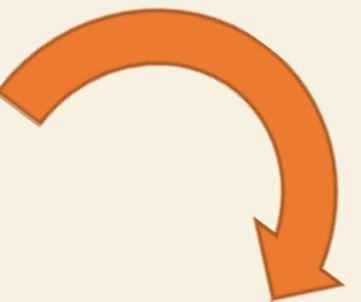
```

WITH table1 AS (
  SELECT customer_id,COUNT(DISTINCT product_id) AS distinct_count
  FROM order_items ot
  INNER JOIN orders o USING(order_id)
  GROUP BY customer_id)

SELECT t1.customer_id,CONCAT(c.first_name, " ",c.last_name)
AS full_name,t1.distinct_count
FROM table1 t1
INNER JOIN customers c USING(customer_id)
WHERE distinct_count = (SELECT max(distinct_count) FROM table1);
  
```

Result Grid

customer_id	full_name	distinct_count
1	John Doe	3
2	Jane Smith	3
3	Bob Johnson	3



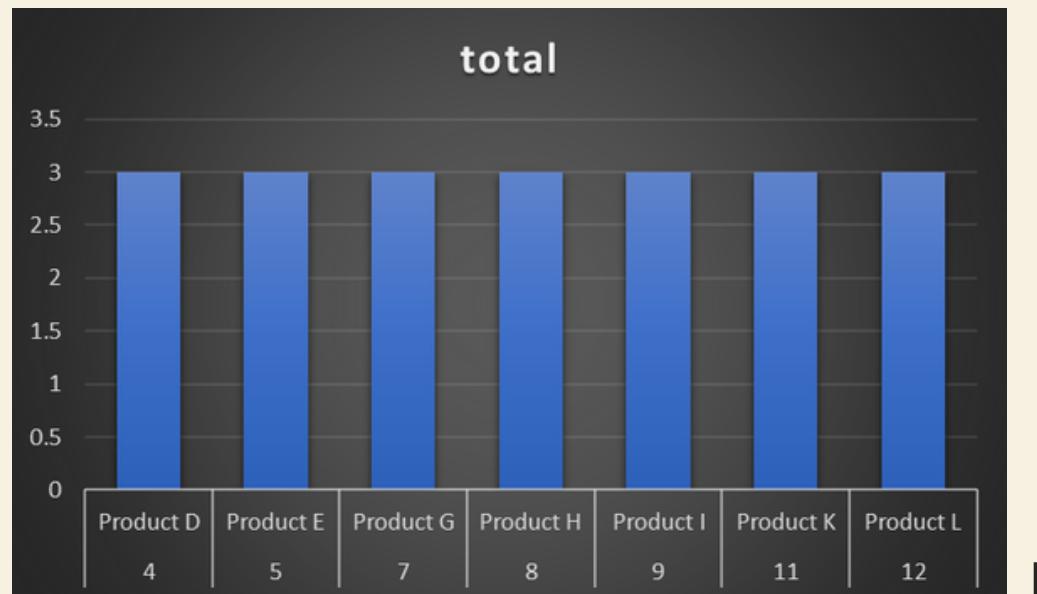
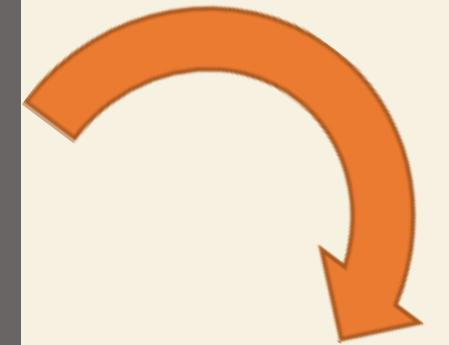
Insights:- The top 3 customers who have ordered the most distinct products are John Doe, Jane Smith, and Bob Johnson, with each of them ordering 3 different products.

7. Which product has been bought the least in terms of quantity?

```

WITH lowest AS (
  SELECT product_id, SUM(quantity) AS total,
  DENSE_RANK() OVER (ORDER BY SUM(quantity) ASC) AS rnk
  FROM order_items
  GROUP BY product_id
)
SELECT lowest.product_id, p.product_name, lowest.total
FROM lowest
INNER JOIN products p USING (product_id)
WHERE rnk = 1;
  
```

product_id	product_name	total
4	Product D	3
5	Product E	3
7	Product G	3
8	Product H	3
9	Product I	3
11	Product K	3
12	Product L	3



Insights:-Products D, E, G, H, I, K, and L have been bought the least in terms of quantity, with a quantity of 3 each.



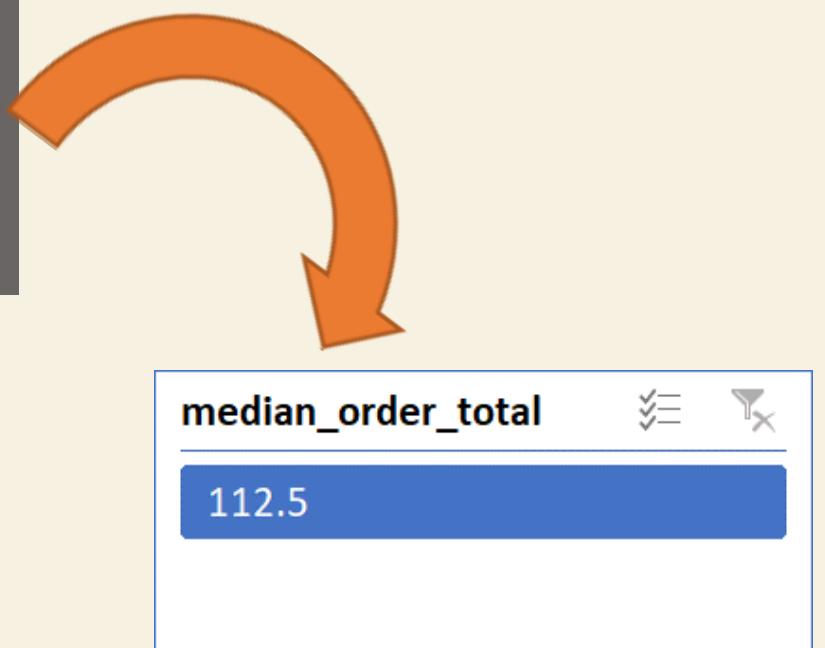
Data in Motion

8. What is the median order total?

```
SELECT ROUND(avg(order_total), 2) AS median_total_order
FROM (
    WITH cte2 AS (
        WITH cte1 AS (
            SELECT order_id, (p.price * ot.quantity) AS total
            FROM order_items ot
            INNER JOIN products p USING (product_id)
        )
        SELECT order_id, SUM(total) AS order_total, COUNT(*) / 2.0 AS total_rows
        FROM cte1
        GROUP BY order_id
    )
    SELECT *, (ROW_NUMBER() OVER (ORDER BY order_total) * 1.0) AS row_num
    FROM cte2
) AS modified_table
WHERE row_num BETWEEN total_rows / 2.0 AND total_rows / 2.0 + 1;
```



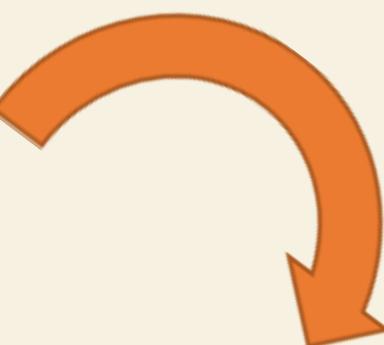
Result Grid	
median_order_total	
▶	112.50
Filter Rows:	



Insights:- the median order total is 112.50.

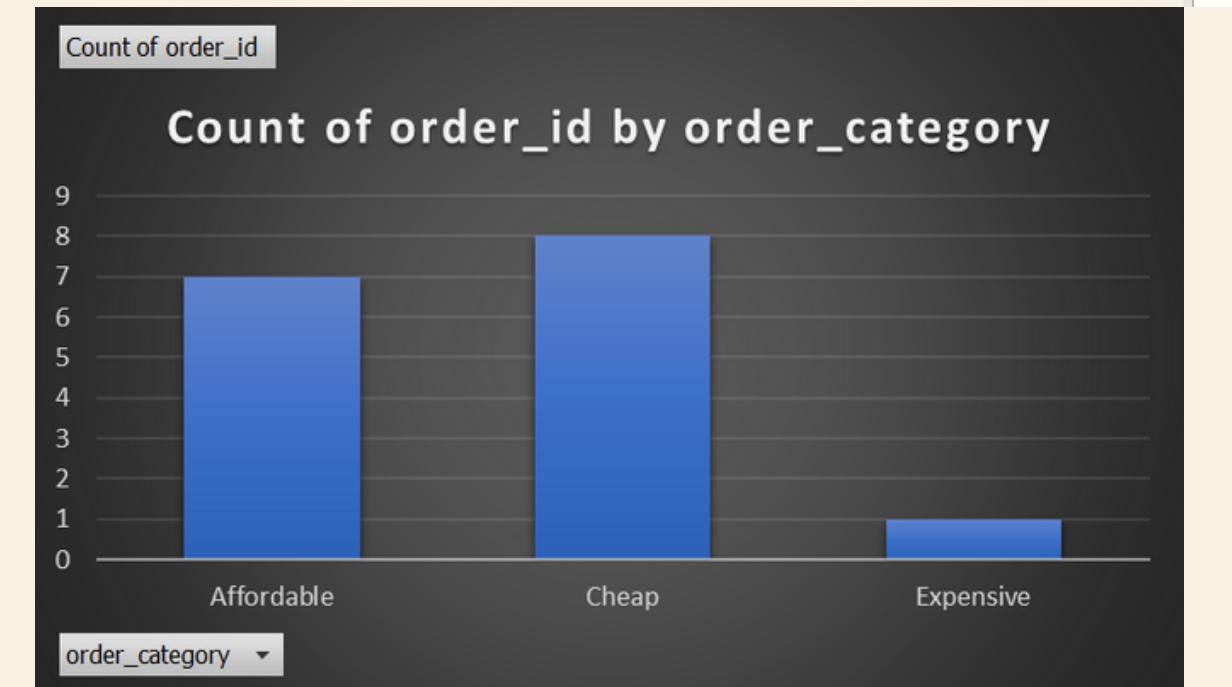
9. For each order, determine if it was 'Expensive' (total over 300), 'Affordable' (total over 100), or 'Cheap'.

```
SELECT order_id,
CASE
    WHEN SUM(price * quantity) > 300 THEN 'Expensive'
    WHEN SUM(price * quantity) > 100 THEN 'Affordable'
    ELSE 'Cheap'
END AS order_category
FROM order_items
JOIN products ON order_items.product_id = products.product_id
GROUP BY order_id;
```



order_category	Count of order_id
Affordable	7
Cheap	8
Expensive	1

A	B
order_id	order_category
1	Cheap
2	Cheap
3	Cheap
4	Cheap
5	Cheap
6	Cheap
7	Cheap
8	Affordable
9	Affordable
10	Affordable
11	Affordable
12	Cheap
13	Affordable
14	Affordable
15	Affordable
16	Expensive



Insights:-The finding suggests that the majority of the orders can be categorized as 'Cheap' or 'Affordable', with only a few orders falling under the 'Expensive' category.

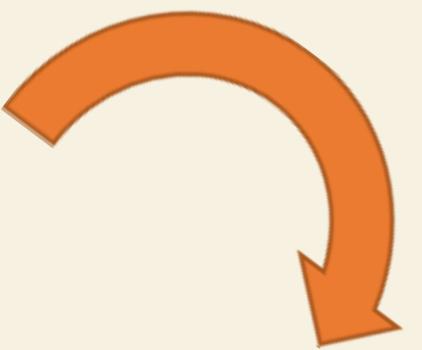
10. Find customers who have ordered the product with the highest price.

```
WITH highest_priced_product AS (
  SELECT product_id
  FROM products
  WHERE price = (SELECT MAX(price) FROM products)
)
SELECT c.customer_id, CONCAT(c.first_name, ' ', c.last_name) AS full_name
FROM customers c
JOIN orders o ON c.customer_id = o.customer_id
JOIN order_items oi ON o.order_id = oi.order_id
WHERE oi.product_id IN (SELECT product_id FROM highest_priced_product);
```



Result Grid | Filter Rows:

	customer_id	full_name
▶	8	Ivy Jones
	13	Sophia Thomas



A B

customer_id	full_name
8	Ivy Jones
13	Sophia Thomas



Insights:- The finding indicates that customers Ivy Jones and Sophia Thomas have ordered the product with the highest price.



Summary and Implications



Summary

The analysis of the dataset from the tiny shop provided valuable insights into various aspects of the business. Here is a summary of the findings

- The highest-priced product, Product M, indicates its potential profitability.
- Customer John Doe stands out as the most frequent orderer, highlighting his loyalty.
- The total revenue per product provides insights into popularity and profitability, with 'Product J' generating the highest revenue.
- The day with the highest revenue, Tuesday, May 16th, 2023-05-16', suggests a potential peak in sales.
- Examining the first order for each customer offers insights into their initial preferences.
- The top 3 customers who ordered the most distinct products are John Doe, Jane Smith, and Bob Johnson.
- Products D, E, G, H, I, K, and L have the least quantity sold.
- The median order total is \$112.50, and customers who ordered the highest-priced product can be identified.
- Orders are categorized as 'Expensive', 'Affordable', or 'Cheap' based on their total value, enabling a better understanding of customer spending patterns.
- Notably, Ivy Jones and Sophia Thomas emerge as customers with the most orders.

These findings provide valuable information to drive strategic decisions and optimize the shop's operations.



Implications

Based on the summary findings of the case study, the following implications can be drawn:

1. **Pricing Optimization:** The highest-priced product indicates potential for premium pricing strategies to target niche market segments and maximize profitability.
2. **Customer Relationship Management:** Implementing personalized offers and loyalty programs can enhance customer retention and foster long-term relationships.
3. **Product Portfolio Management:** Evaluating the performance of products in terms of revenue and quantity can guide decisions on product improvement, promotion, or discontinuation.
4. **Revenue Maximization:** Identifying peak sales periods and aligning resources accordingly can help maximize revenue generation.
5. **Market Expansion:** Diversifying the product offering based on customer preferences can attract a wider customer base and drive growth.
6. **Inventory Optimization:** Analyzing low-selling products can inform inventory management strategies, such as promotions or cross-selling, to improve sales and reduce holding costs.
7. **Pricing Segmentation:** Implementing pricing tiers based on the median order total can attract customers across different spending ranges and increase market reach.
8. **Targeted Marketing Campaigns:** Segmenting customers based on spending patterns allows for personalized marketing strategies and more effective promotional campaigns.
9. **High-Value Customer Engagement:** Identifying customers who purchase the highest-priced product enables targeted marketing efforts to engage and retain these valuable customers.
10. **Data-Driven Decision Making:** Continuous analysis of sales data and market trends provides valuable insights for making informed decisions and driving business growth.





Thank you

