

# PIZZA SALES REPORT

*Standard Query Assignment*

By Mazmур Chrisadrianto Telaumbanua





# STEP 1

## CHOOSE DATASETS

# CREATE DATABASE USING “JOIN” FUNCTION

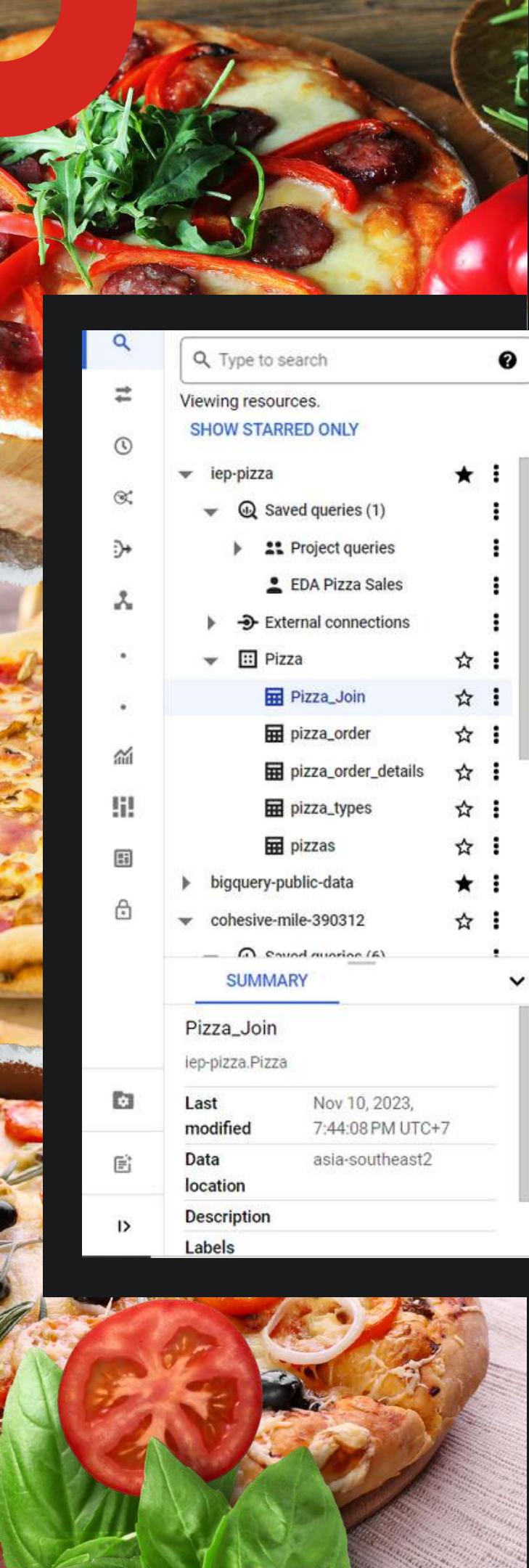


```
1 SELECT
2     o.order_id,
3     o.date,
4     o.time,
5     od.order_details_id,
6     od.pizza_id,
7     Od.quantity,
8     p.price,
9     p.price * od.quantity as total_price,
10    p.pizza_type_id,
11    CASE
12        WHEN p.size = "L" THEN "Large"
13        WHEN p.size = "S" THEN "Small"
14        WHEN p.size = "M" THEN "Medium"
15        WHEN p.size = "XL" THEN "Extra Large"
16        WHEN p.size = "XXL" THEN "Double Extra Large"
17    END
18    AS size,
19    pt.string_field_2 as pizza_category,
20    pt.string_field_3 as pizza_ingredients,
21    pt.string_field_1 as pizza_name
22    FROM
23        `iep-pizza.Pizza.pizza_orders` o
24    JOIN
25        `iep-pizza.Pizza.pizza_order_details` od
26    USING
27        (order_id)
28    JOIN
29        `iep-pizza.Pizza.pizzas` p
30    ON
31        od.pizza_id=p.pizza_id
32    JOIN
33        `iep-pizza.Pizza.pizza_types` pt
34    ON
35        p.pizza_type_id=pt.string_field_0 ;
36
```

This query uses the **join function** to combine several tables, namely the **pizza\_order table** with fields order id, date, time, **then the\_order\_details table** with fields order details id, pizza id, quantity, then the **pizzas table** with fields price, pizza type id, size, and finally the **pizza\_types table** with fields pizza category, pizza ingredients, pizza name and provides a case when function to manipulate Size data.

---

# CREATE NEW PIZZA SALES TABLE



Pizza\_Join

QUERY SHARE COPY SNAPSHOT DELETE EXPORT

SCHEMA DETAILS PREVIEW LINEAGE DATA PROFILE DATA QUALITY

Filter Enter property name or value

Field name	Type	Mode	Key	Collation	Default Value	Policy Tags	Description
order_id	INTEGER	NULLABLE					
date	DATE	NULLABLE					
time	TIME	NULLABLE					
order_details_id	INTEGER	NULLABLE					
pizza_id	STRING	NULLABLE					
quantity	INTEGER	NULLABLE					
price	FLOAT	NULLABLE					
total_price	FLOAT	NULLABLE					
pizza_type_id	STRING	NULLABLE					
size	STRING	NULLABLE					
pizza_category	STRING	NULLABLE					
pizza_ingredients	STRING	NULLABLE					
pizza_name	STRING	NULLABLE					

EDIT SCHEMA VIEW ROW ACCESS POLICIES

PERSONAL HISTORY PROJECT HISTORY

For efficiency, we save **the result of the latest query** used as **a database** for the required analysis.

The purpose of this step is to make an easier and more efficient analysis.



# STEP 2

## UNDERSTANDING THE PROBLEM AND DEFINING PROJECT OBJECTIVES

---

# BUSINESS BACKGROUND

We are hired as BI Consultant and have been hired to help the restaurant use data to improve operations.

transactional data for the past year, but haven't been able to put it to good use. I'm hoping you can look over the data and put together a report to help us find opportunities to drive more sales and work more efficiently.

Here are some questions that we'd like to be able to answer:

1. How many pizzas are we making during peak periods?
2. What are our best and worst-selling pizzas?
3. What's our average order value?
4. What dates could be potentially good sales?





## BUSINES PROBLEM

The business problem is using their data to its full potential. By analyzing their transactional data and answering specific questions about customer demand, product popularity, and sales performance we can identify opportunities to improve their business and drive more sales





## PROBLEM STATEMENT

How to increase sales by 10% in  
2016 compared to 2015?



## OBJECTIVE

To increase sales by 10% in 2016  
compared to 2015, by increasing sales





# STEP 3

## ANALYZE THE DATA



# 26 NOVEMBER 2016 IS THE DATE WE TEND TO BE BUSIEST

Date when we have the highest order

```
SELECT
    SUM (quantity)as total_pizza,
    date as date_time
FROM `iep-pizza.Pizza.Pizza_Join`
GROUP BY 2
ORDER BY 1 DESC
LIMIT 10
```

Row	total_pizza	date_time
1	266	2015-11-26
2	264	2015-11-27
3	262	2015-10-15
4	234	2015-07-04

The **highest total order** in a day dropped on **26 November 2015** with 266 total pizzas sold followed by the next date **27 November 2015** with 264 total pizzas sold.

# FRIDAY IS THE BUSIEST DAY IN THIS YEAR

```
82 SELECT
83 SUM (quantity)as total_pizza,
84 format_date ('%A', date) as day
85 FROM `iep-pizza.Pizza.Pizza_Join`
86 GROUP BY 2
87 ORDER BY 1 DESC
88
```

Row	total_pizza	day
1	8242	Friday
2	7493	Saturday
3	7478	Thursday
4	6946	Wednesday
5	6895	Tuesday
6	6485	Monday
7	6035	Sunday



Friday is the **busiest day** of the week In 2015 a total of **8242 pizzas** were made. Followed by **Saturday** with **7493 pizzas** that were made. For raising the order on other days we could use **a promo** for other days we could use a promo for **increase customer enthusiasm**.



# JULY IS THE MONTH COULD BE POTENTIALLY GOOD SALES

The most possible date could have the highest sales

The date could be based on the highest total order in a day the highest sales **26 November 2015** with **266 total pizzas sold** followed by the next date **27 November 2015** with **264 total pizzas sold**

However if we take **monthly sales**, July will become the most potential for good sales month because of **the highest total pizza orders** with **4392 pizzas** sold in July

Based on this information we have the answer of how many pizzas were made during peak periods. July is the potential in month and Friday is the most potential in day period

Row	total_pizza	month
1	4392	July
2	4328	May
3	4266	November
4	4261	March
5	4232	January
6	4168	August
7	4151	April
8	4107	June
9	3961	February
10	3935	December
11	3890	September
12	3883	October

```
38 SELECT
39 SUM (quantity)as total_pizza,
40 FORMAT_DATE ('%B', date) as month
41 FROM `iep-pizza.Pizza.Pizza_Join`
42 GROUP BY 2
43 ORDER BY 1 DESC
44 ;
```

# THE THAI CHICKEN PIZZA IS THE MOST CONTRIBUTED PIZZA TO REVENUE

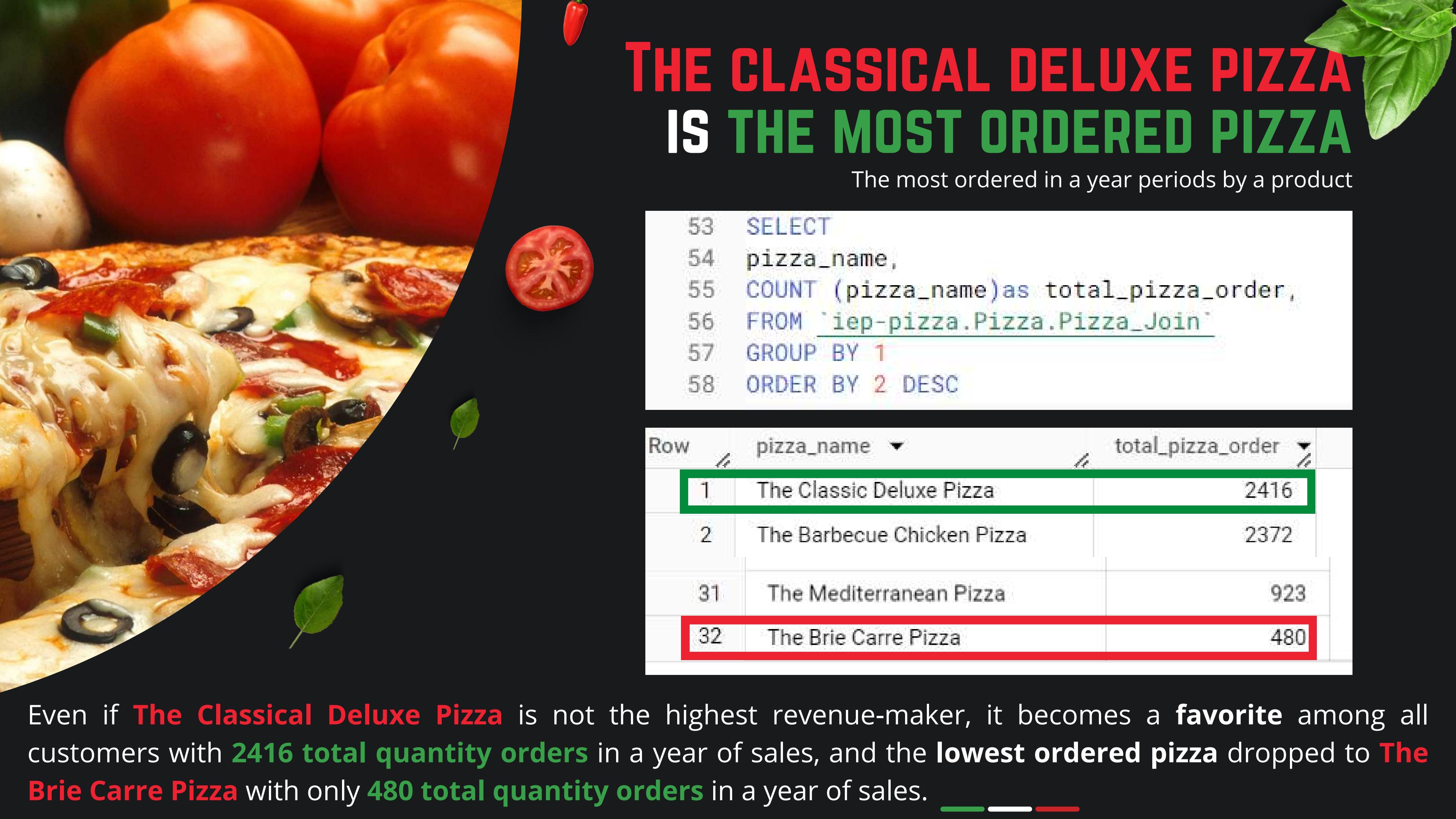
Highest revenue made in a year periods by a product

```
46 SELECT
47 pizza_name,
48 ROUND (SUM(total_price),2) as total_revenue_pizza,
49 FROM `iep-pizza.Pizza.Pizza_Join`
50 GROUP BY 1
51 ORDER BY 2 DESC
52 ;
```

Row	pizza_name	total_revenue_pizza
1	The Thai Chicken Pizza	43434.25
2	The Barbecue Chicken Pizza	42768.0
31	The Green Garden Pizza	13955.75
32	The Brie Carre Pizza	11588.5



**The Thai Chicken Pizza** contributed **\$43.434,25** in a year. This is the highest revenue we got from single pizza. And made the **Brie Carre Pizza** is the less contributed In revenue with only earn **\$11.588,5** in a year of sales.



# THE CLASSICAL DELUXE PIZZA IS THE MOST ORDERED PIZZA

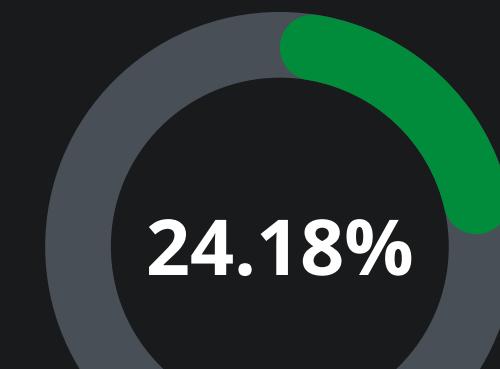
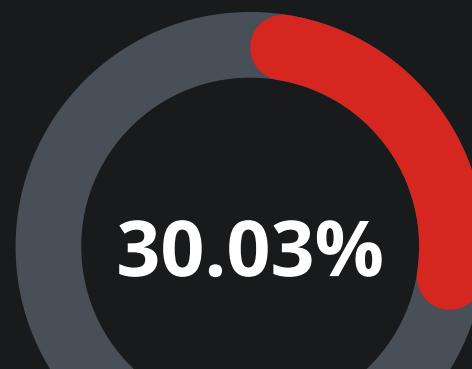
The most ordered in a year periods by a product

```
53 SELECT
54 pizza_name,
55 COUNT (pizza_name)as total_pizza_order,
56 FROM `iep-pizza.Pizza.Pizza_Join`
57 GROUP BY 1
58 ORDER BY 2 DESC
```

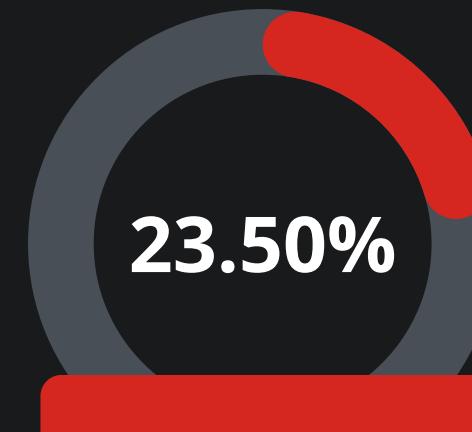
Row	pizza_name	total_pizza_order
1	The Classic Deluxe Pizza	2416
2	The Barbecue Chicken Pizza	2372
31	The Mediterranean Pizza	923
32	The Brie Carre Pizza	480

Even if **The Classical Deluxe Pizza** is not the highest revenue-maker, it becomes a **favorite** among all customers with **2416 total quantity orders** in a year of sales, and the **lowest ordered pizza** dropped to **The Brie Carre Pizza** with only **480 total quantity orders** in a year of sales.

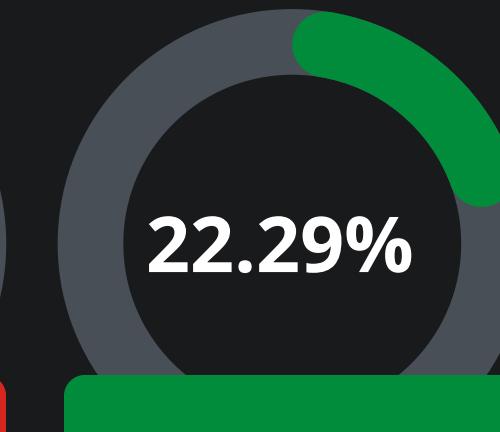
# THE CLASSIC PIZZA IS THE FAVORITE CATEGORY OF PIZZA



CLASSIC



VEGGIE



CHICKEN

```
73  SELECT
74    pizza_category,
75    SUM (quantity)as total_pizza_category,
76    FORMAT('%s%%',CAST(ROUND((SUM(quantity)/49574)*100,2) AS STRING)) as Percentage
77  FROM iep-pizza.Pizza.Pizza_Join
78  GROUP BY 1
79  ORDER BY [2] DESC
```

Row	pizza_category	total_pizza_category	Percentage
1	Classic	14888	30.03%
2	Supreme	11987	24.18%
3	Veggie	11649	23.5%
4	Chicken	11050	22.29%

**Classic category** has the **highest demand** with **14.888 total quantity orders** in a year of sales, and the **lowest ordered pizza** dropped to **chicken category** with **11050 total quantity orders** in a year of sales.



\$ 38,307  
AVERAGE ORDER OF VALUE

```
29 SELECT
30 sum(total_price)/count (distinct order_id) as aov
31 FROM "iep-pizza.Pizza.Pizza_Join"
32 ;
```

Row	aov
1	38.30726229507...

# FRIDAY IS THE BUSIEST DAY IN THIS YEAR

Row	total_pizza	day
1	8242	Friday
2	7493	Saturday
3	7478	Thursday
4	6946	Wednesday
5	6895	Tuesday
6	6485	Monday
7	6035	Sunday



**Friday** is the **busiest day** of the week In 2015 a total of **8242 pizzas** were made. Followed by **Saturday** with **7493 pizzas** that were made. For raising the order on other days we could use **a promo** for other days we could use a promo for **increase customer enthusiasm**. surprisingly on **Sunday** there are the least orders, where this should be on holidays such as Sundays can be maximized more to get more orders and get higher revenue again.

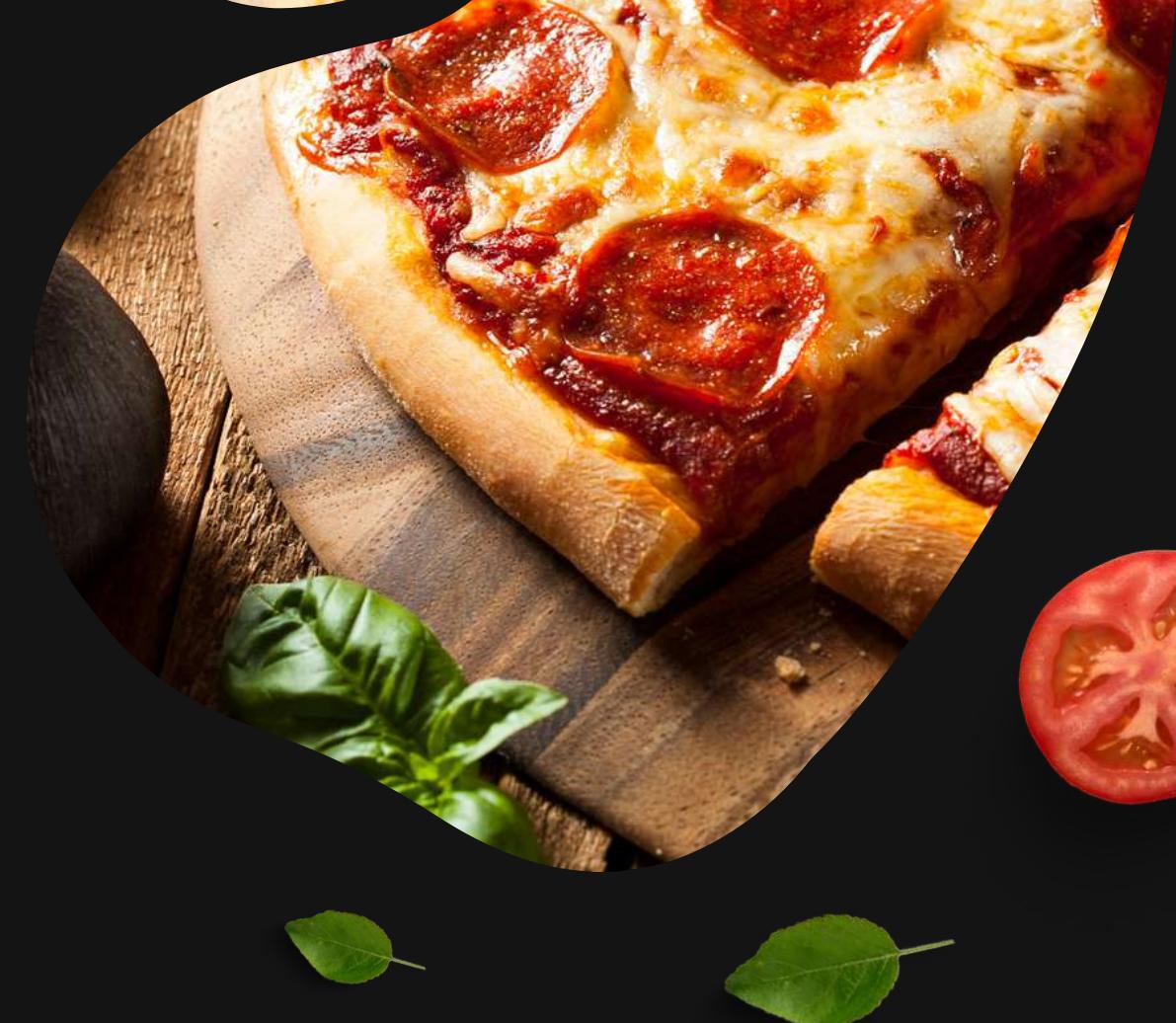
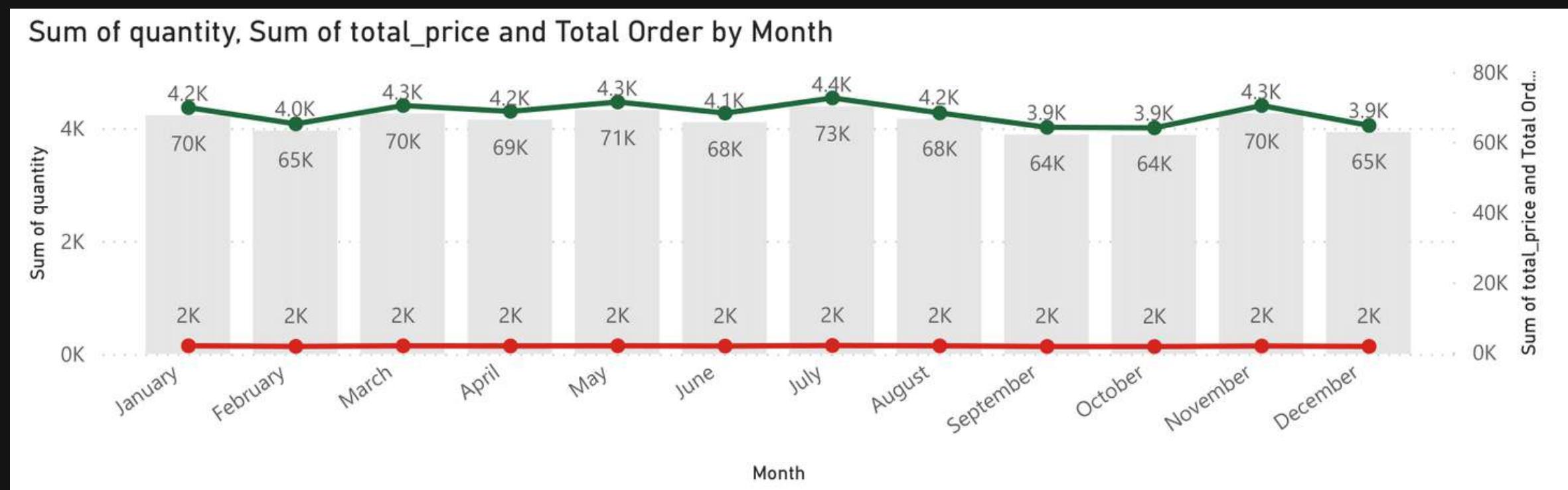
therefore on **Sunday** should be given attractive discounts such as special days, and given other offer offers so that sales on **Sunday** increase, make sure if you provide the best service to customers, so they will come back again the following Sunday.

# JULY IS THE MONTH COULD BE POTENTIALLY GOOD SALES

However if we take **monthly sales**, July will become the most potential for good sales month because of **the highest total pizza orders** with **4392 pizzas** sold in July

But in **September and October**, there was a decrease in total orders that affected the amount of revenue, so we are looking for **obstacles** in that month, it should be in the fall in September and October will **increase sales**.

In **September and October** the revenue was the least throughout the year with **\$64K**, we can provide some attractive promo bundles, to increase sales each month with a target of **10% growth** from the previous year.





# STRATEGY TO INCREASING INCOME BY 10% NEXT YEAR?

**Total Revenue** is currently **\$817.860** if we give a **10% revenue increase target** then we should get an increase of **\$81.786 per year**, If we divide the target into monthly sales, it would be **\$6,816 per month**.

This amount will be allocated according to the target distribution, on **weekends (Saturday and Sunday)** it will be increased again so that it can get an **increase of \$570 per week**, if multiplied by 4 weeks it will reach **\$2,280**, for **Monday to Friday** is given a target of **\$980 per week**, so if 4 weeks then there is an **increase of \$3,920**.

Target Weekday Promo (Monday to Friday)			Target Weekend Promo (Saturday and Sunday)		
By Day	By Week	By 4 Week	By Day	By Week	By 4 Week
\$200	\$1000	\$4000	\$360	\$720	\$2840

# VARIOUS TYPES OF PROMOTIONS

## Weekday Promo

- 10% discount for pizza purchase above \$10
- Free shipping for purchases above \$15
- 20% discount for office pizza purchase

## Weekend Promo

- 30% off all pizzas
- Buy 3 get 1 free for size small pizzas
- Free additional toppings for pizza purchase above \$15
- 50% discount for party pizza purchase

With various promos on each type of promotion, we can increase pizza sales.

1. Weekday promos may help attract customers who are busy on weekdays
2. Weekend promos may help attract customers who are looking for delicious and diverse food on holidays

By implementing effective promotions, can increase pizza sales and achieve business goals.



# DASHBOARD

## PIZZA SALES REPORT

Total Order by size



817.86K

49574

21350

38.31

2.32

Total Revenue

49574

Total Pizza Sold

21350

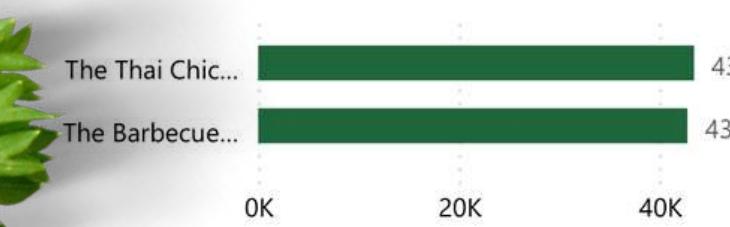
Total Order

38.31

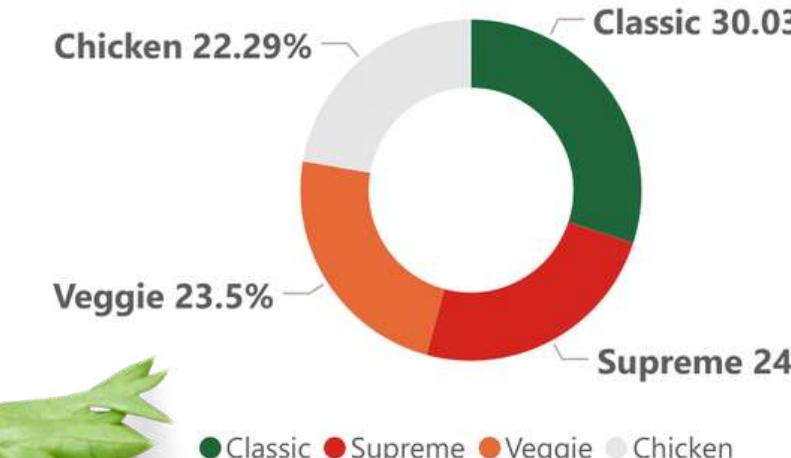
Average Order Value

2.32

Total Revenue by pizza\_name



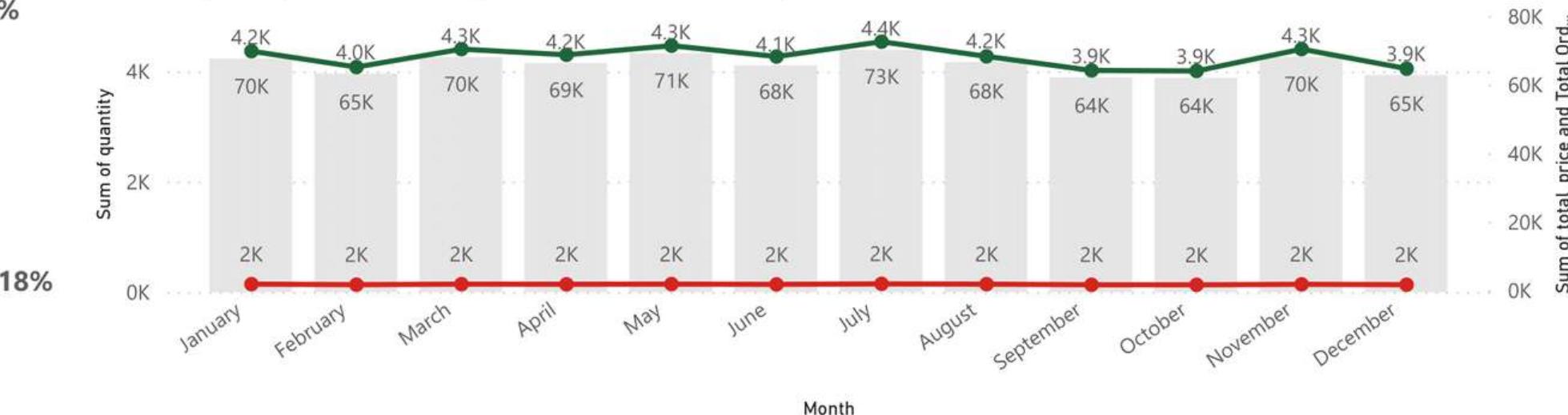
Sum of quantity by pizza\_category



Total Order by Month



Sum of quantity, Sum of total\_price and Total Order by Month



# Thank You

