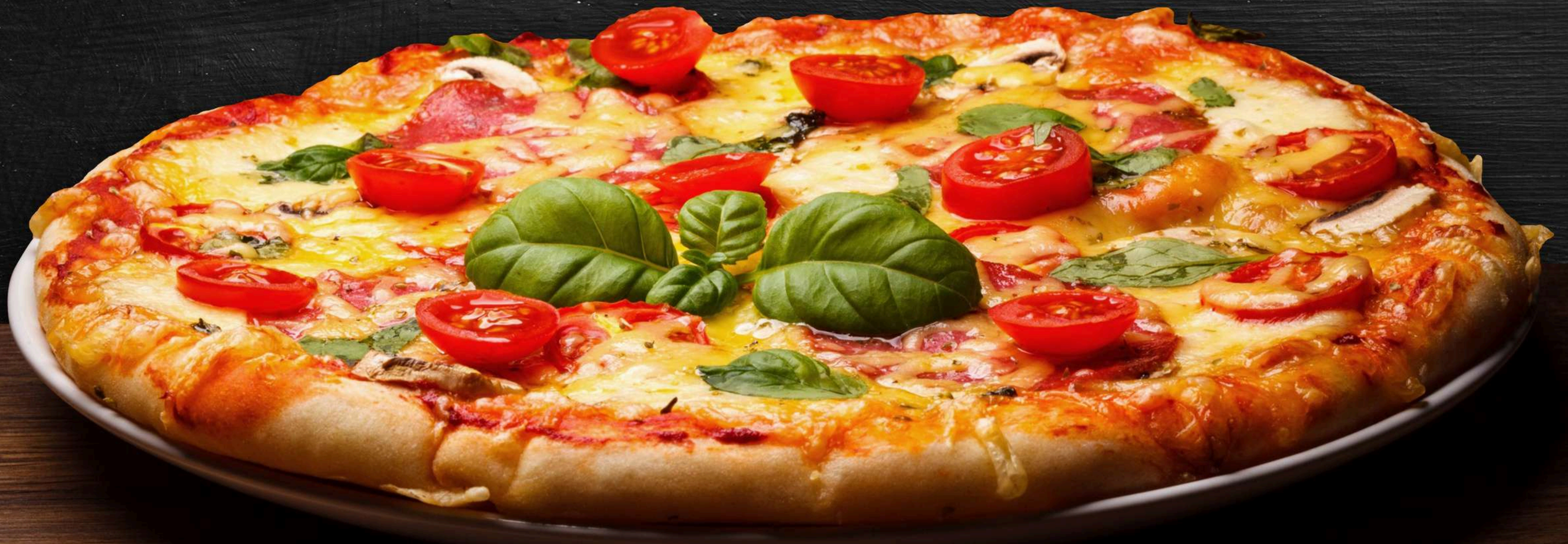
PIZZA SALES DATA ANALYSIS USING SQL

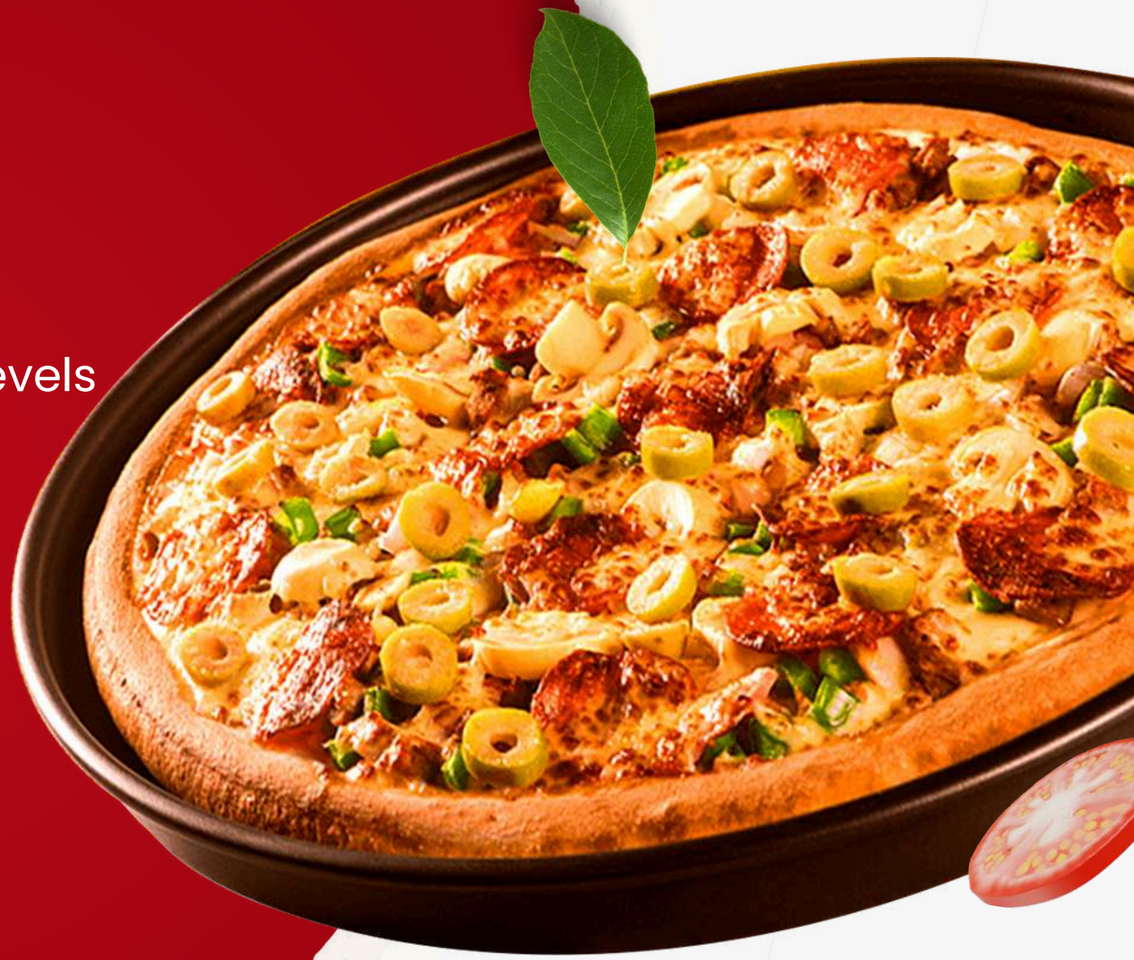
ANALYZING SALES PERFORMANCE AND TRENDS THROUGH SQL QUERIES

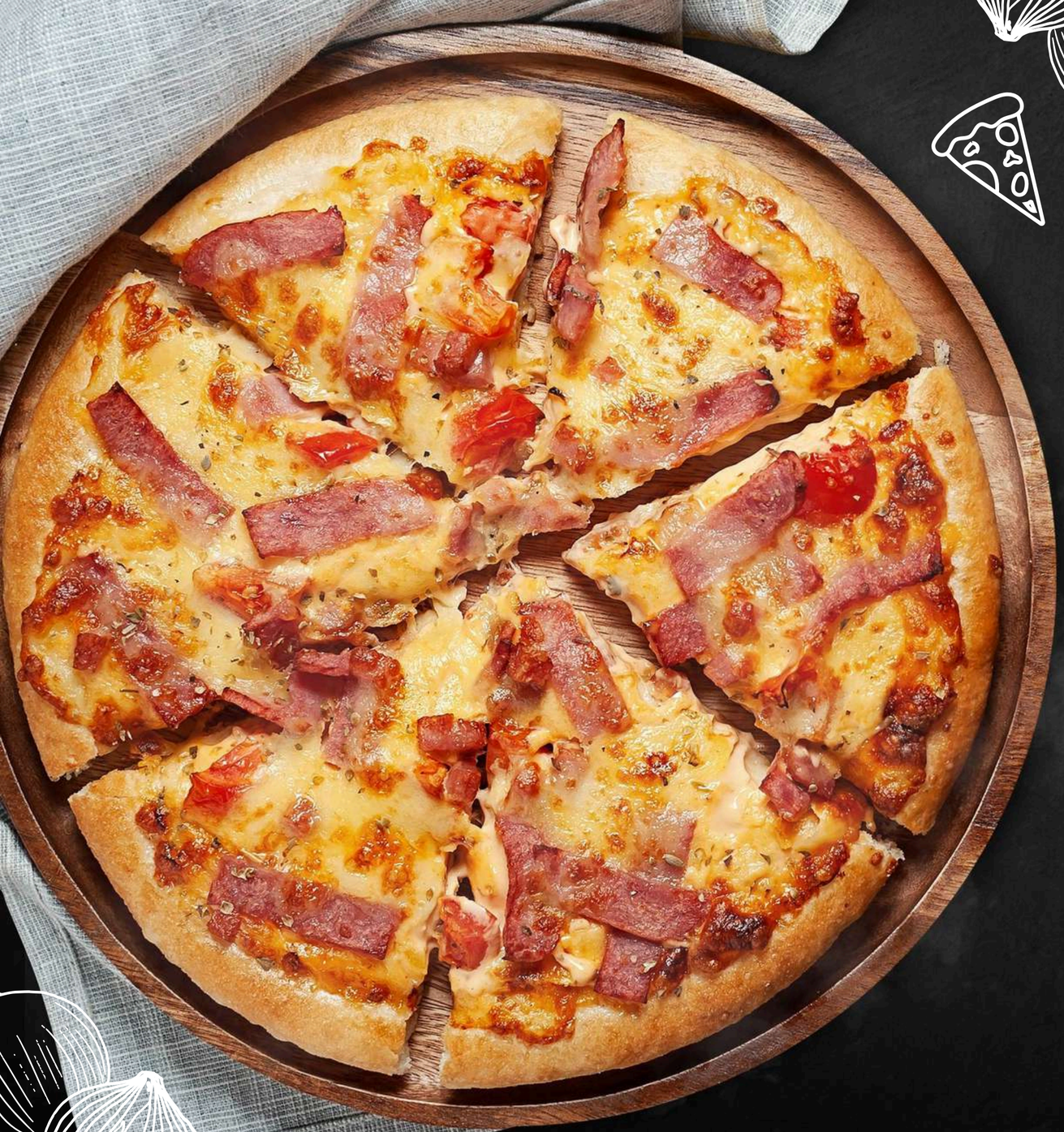
PRESENTED BY: ASHEESH FAUJDAR



INTRODUCTION

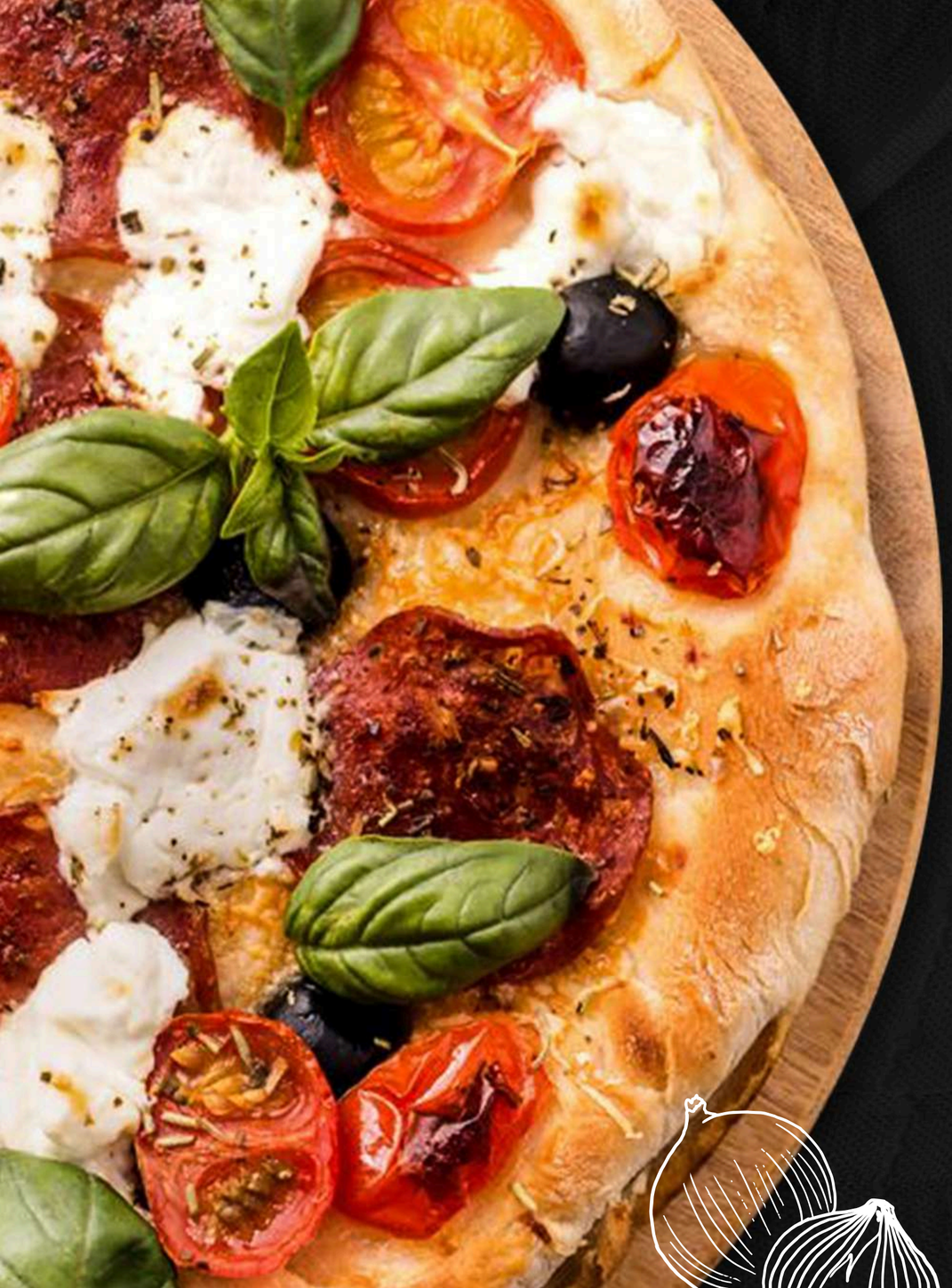
- Project based on analyzing Pizza Sales Dataset using SQL
- Queries divided into Basic, Intermediate, and Advanced levels
- Aims to answer important business questions such as:
    - Total orders and revenue generated
    - Popular pizzas and customer preferences
    - Category-wise distribution and sales trends over time





PROJECT OBJECTIVE

- Extract key insights from raw pizza sales data
- Analyze orders, revenue trends, pizza categories, and customer preferences
- Demonstrate SQL as a powerful tool for business analytics
- Provide practical, data-driven recommendations to improve sales strategy



DATASET INFORMATION

Dataset is provided in 4 CSV files:

1. orders.csv → order_id, order_date, order_time
2. order_details.csv → order_details_id, order_id, pizza_id, quantity
3. pizzas.csv → pizza_id, pizza_type_id, size, price
4. pizza_types.csv → pizza_type_id, name, category, ingredients

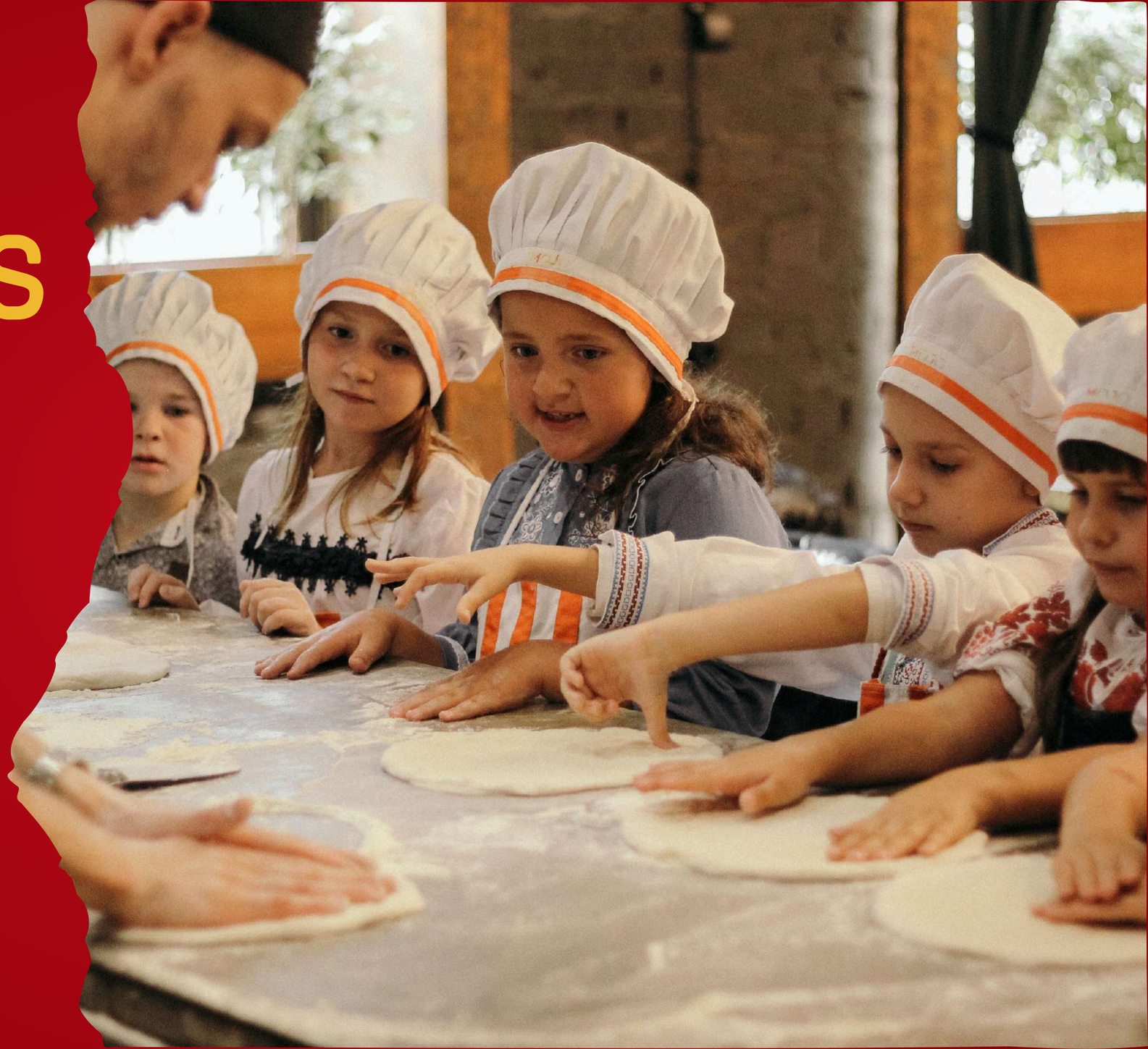
TOOLS & TECHNIQUES

TOOLS

- MySQL
- CSV Files
- SQL Queries
- Canva / Word

TECHNIQUES

- Joins (INNER, CROSS)
- Aggregations (SUM, COUNT, AVG)
- Grouping & Sorting (GROUP BY, ORDER BY)
- Business Metrics (Revenue & % Contribution)



CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```
SELECT
    ROUND(SUM(ORDER_DETAILS.QUANTITY * PIZZAS.PRICE),
            2) AS Total_Revenue
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id
```

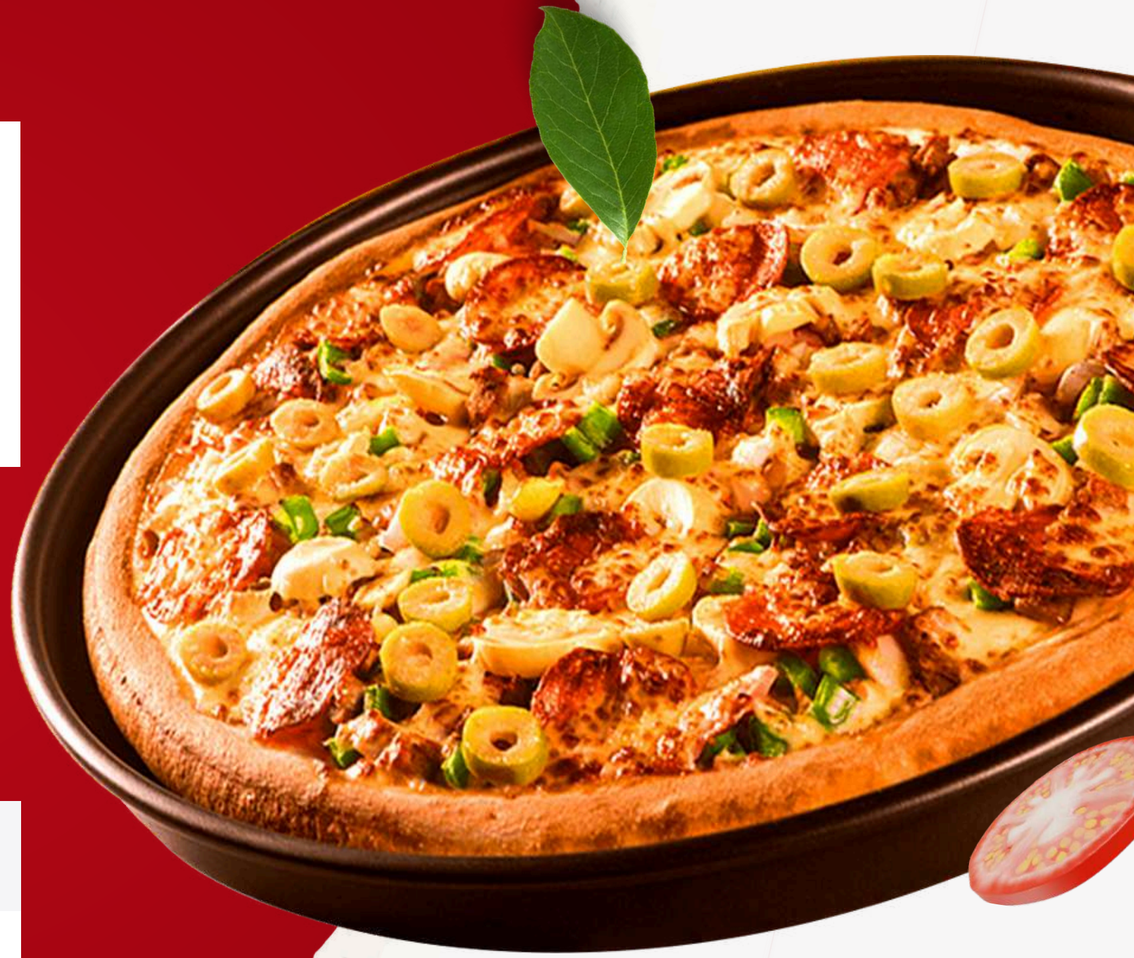
OUTPUT

Result Grid

	Total_Revenue
▶	817860.05

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

```
SELECT
    COUNT(ORDER_ID) AS TOTAL_NUMBER
FROM
    orders;
```



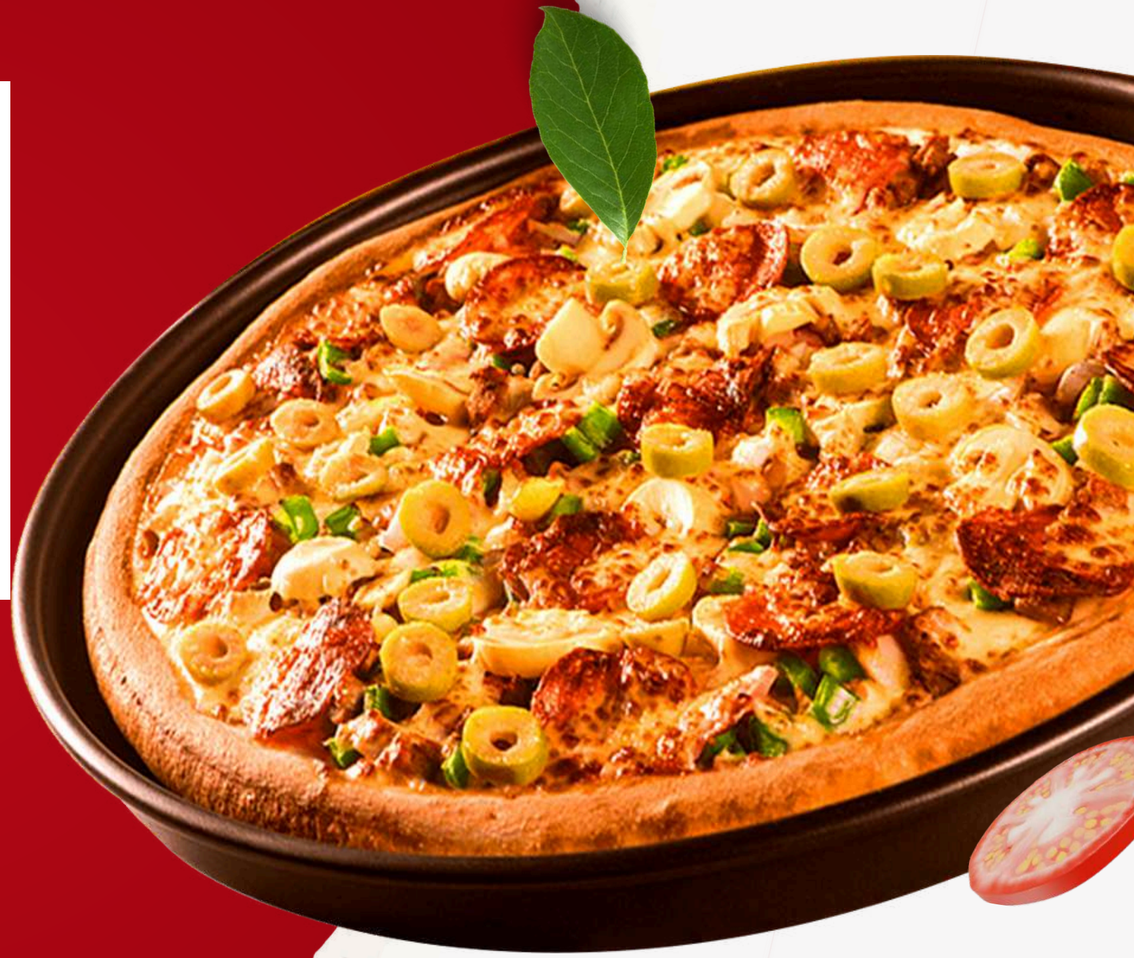
OUTPUT

Result Grid

	TOTAL_NUMBER
▶	21350

IDENTIFY THE HIGHEST-PRICED PIZZA.

```sql
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```



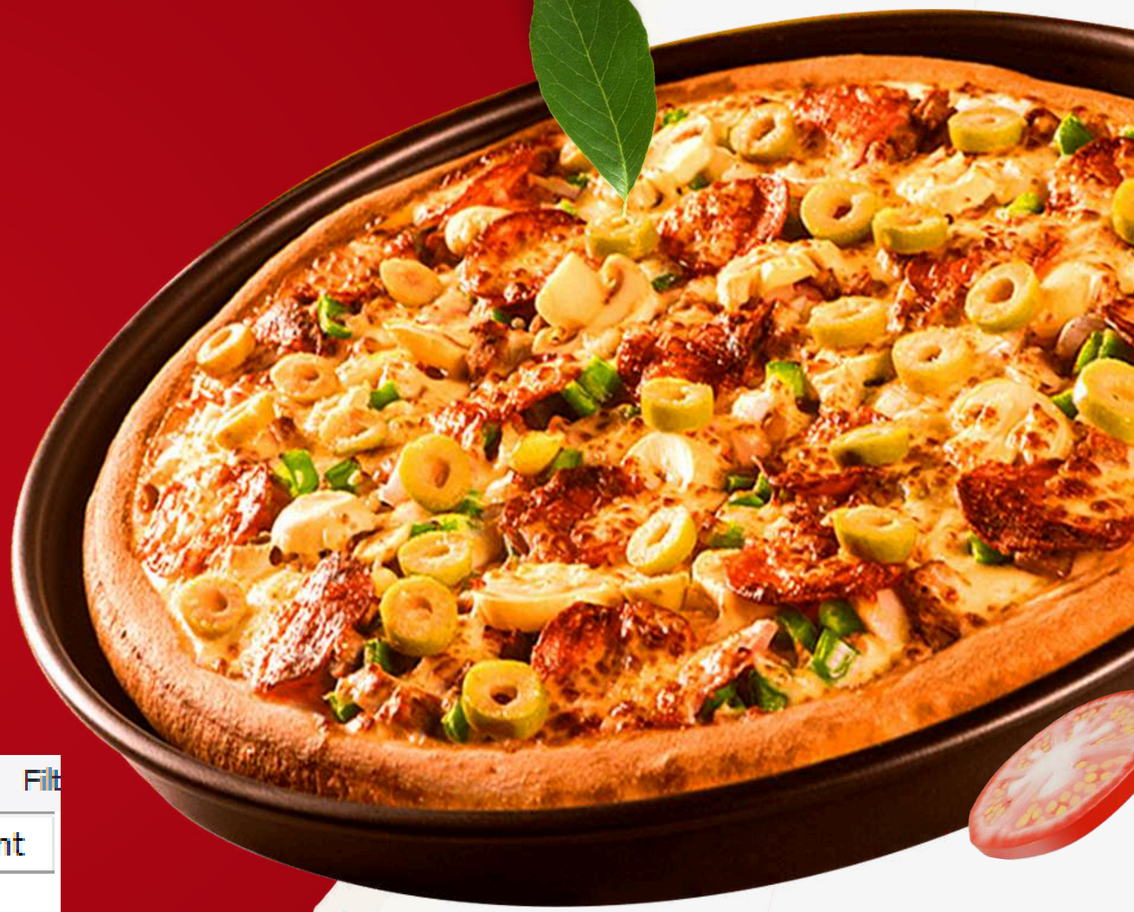
OUTPUT

Result Grid | Filter Ro

	name	price
▶	The Greek Pizza	35.95

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```sql
SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) AS order_count
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC;
```



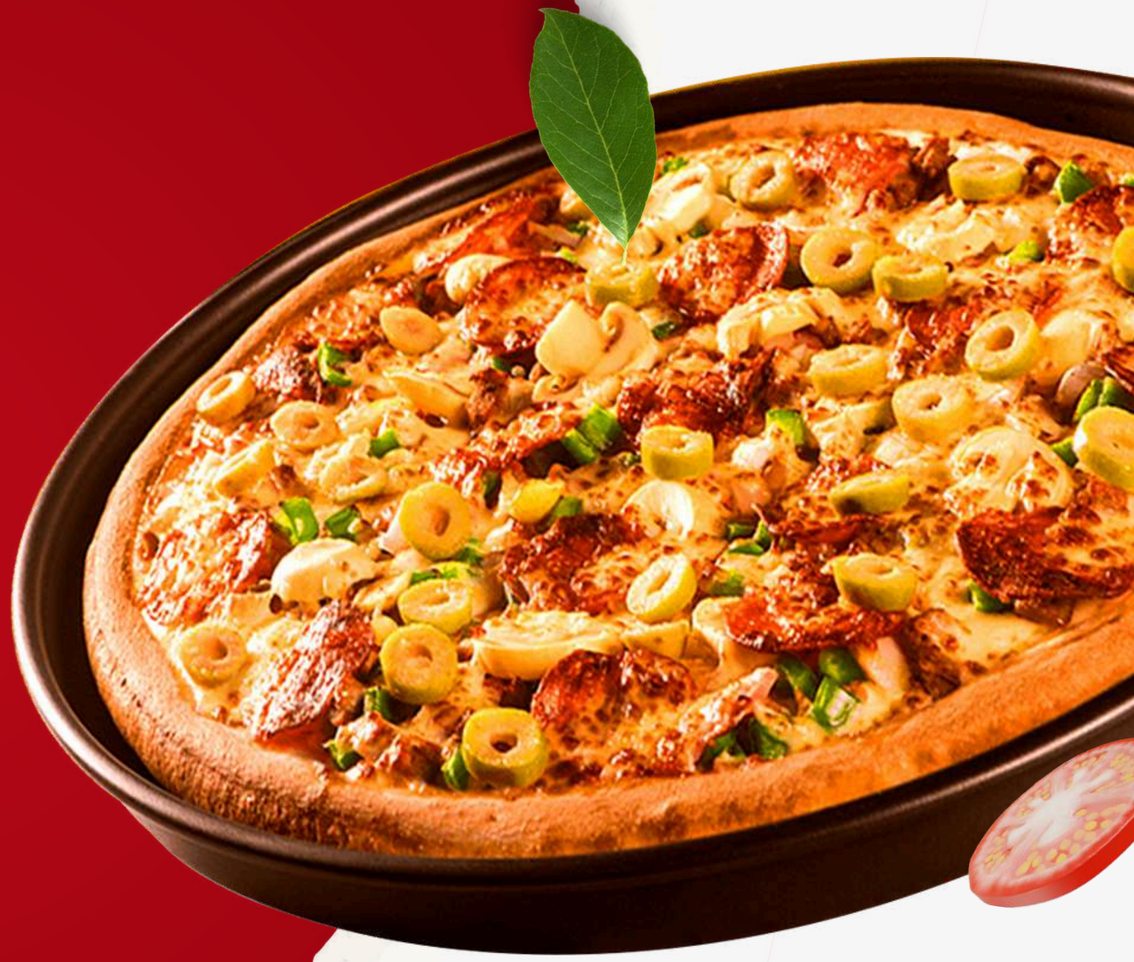
OUTPUT

Result Grid | Filt

size	order_count
L	18526
M	15385
S	14137
XL	544
XXL	28

LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

```sql
SELECT
    pt.name, SUM(od.Quantity) AS Total_Ordered
FROM
    order_details od
        JOIN
    pizzas p ON od.Pizza_id = p.pizza_id
        JOIN
    pizza_types pt ON p.pizza_type_id = pt.pizza_type_id
GROUP BY pt.name
ORDER BY Total_Ordered DESC
LIMIT 5;
```



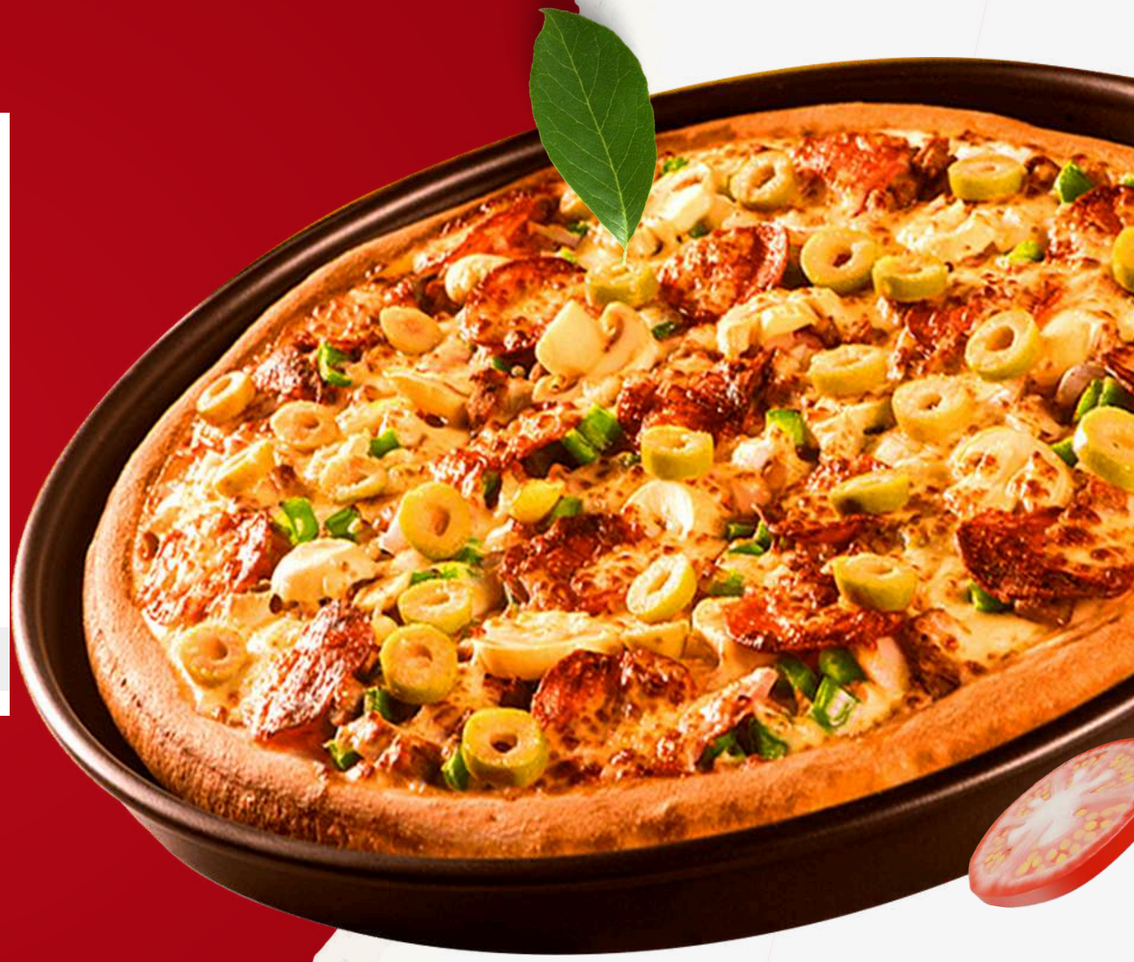
OUTPUT

Result Grid | Filter Rows:

name	Total_Ordered
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED

```sql
SELECT
    pt.category, SUM(od.Quantity) AS Total_Quantity
FROM
    order_details od
        JOIN
    pizzas p ON od.Pizza_id = p.pizza_id
        JOIN
    pizza_types pt ON p.pizza_type_id = pt.pizza_type_id
GROUP BY pt.category;
```



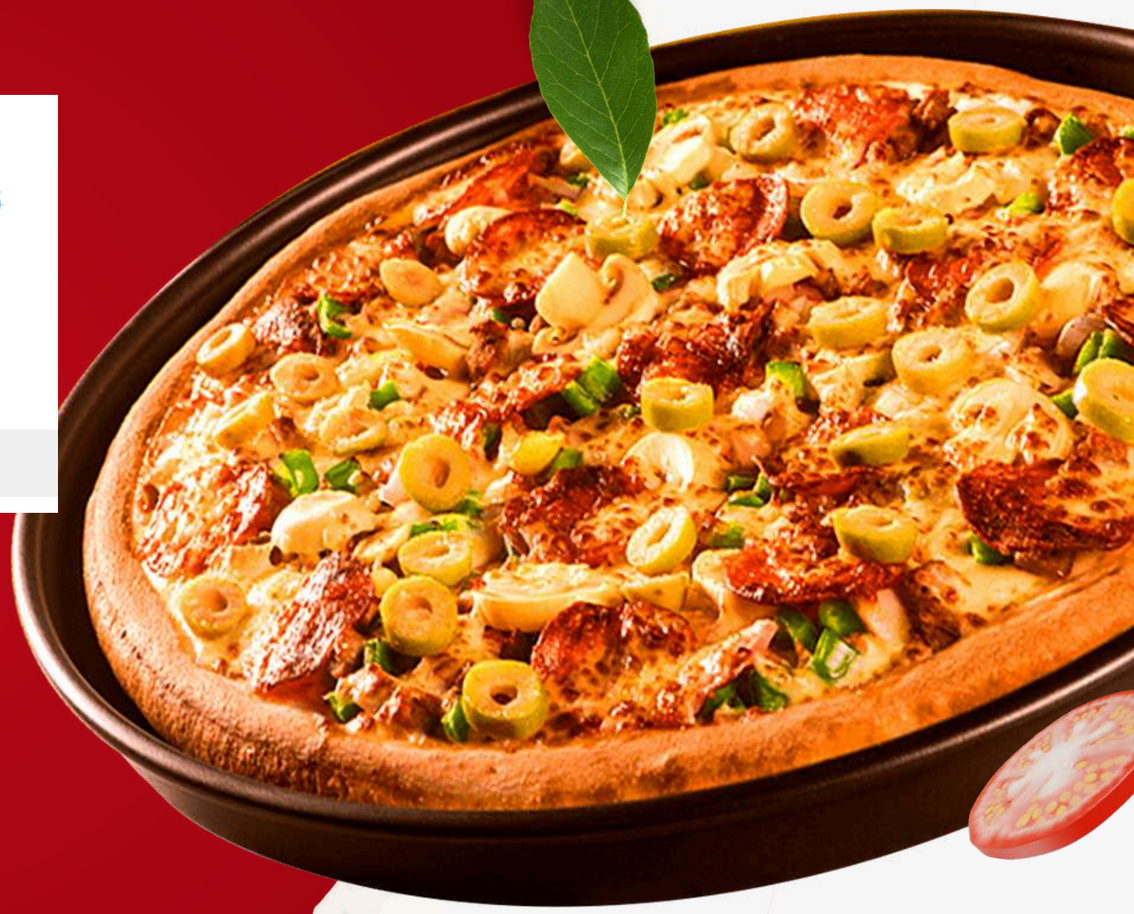
OUTPUT

Result Grid | Filter Row

	category	Total_Quantity
▶	Classic	14888
	Veggie	11649
	Supreme	11987
	Chicken	11050

DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

```sql
SELECT
    HOUR(ORDER_TIME) AS Order_Hour, COUNT(*) AS Total_Orders
FROM
    orders
GROUP BY Order_Hour
ORDER BY Order_Hour;
```



OUTPUT

Result Grid | Filter Rows:

Order_Hour	Total_Orders
9	1
10	8
11	1231
12	2520
13	2455
14	1472
15	1468

CATEGORY-WISE DISTRIBUTION OF PIZZAS.

```sql
SELECT
    pt.category, COUNT(*) AS Number_of_Pizzas
FROM
    pizzas p
        JOIN
    pizza_types pt ON p.pizza_type_id = pt.pizza_type_id
GROUP BY pt.category;
```

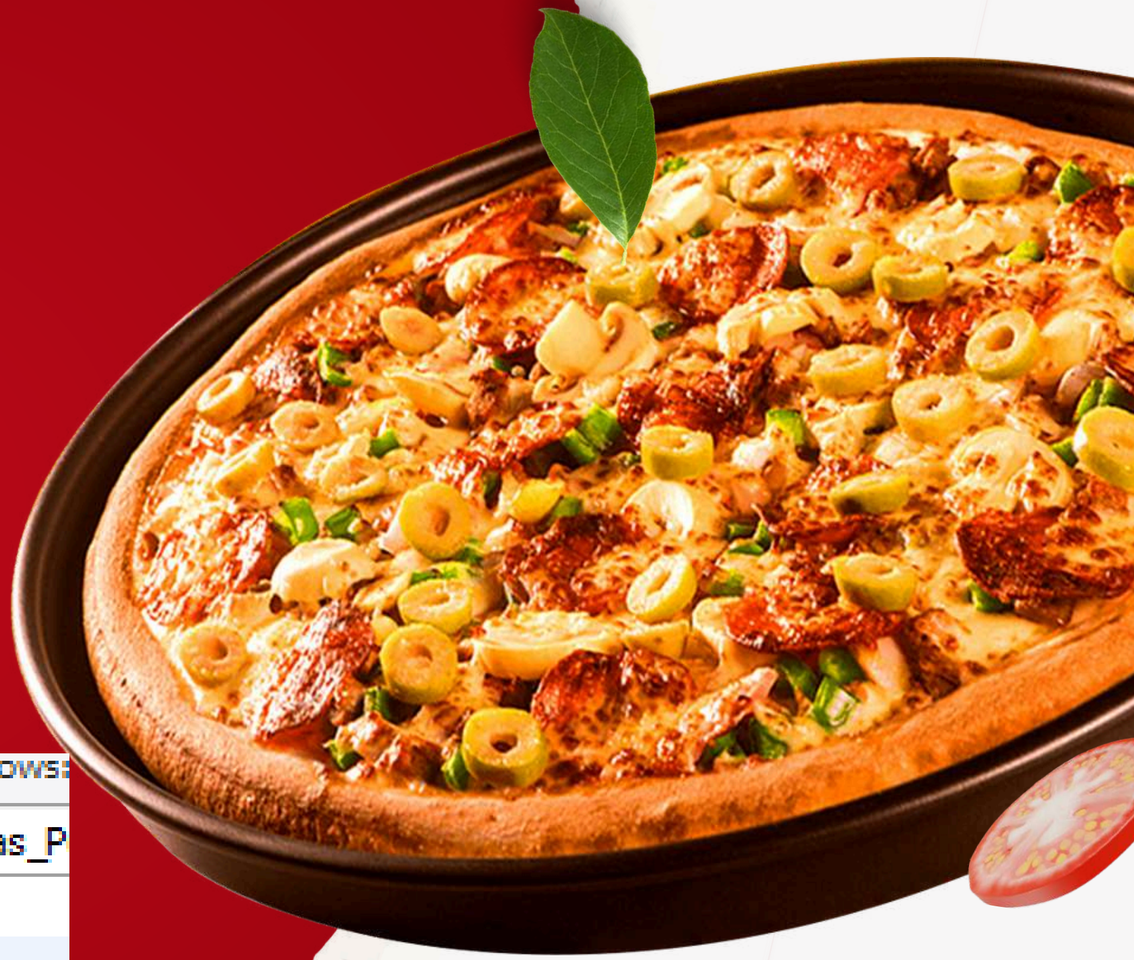
OUTPUT

Result Grid | Filter Rows:

	category	Number_of_Pizzas
▶	Chicken	18
	Classic	26
	Supreme	25
	Veggie	27

AVERAGE NUMBER OF PIZZAS ORDERED PER DAY

```sql
SELECT
    ORDER_DATE, AVG(od.Quantity) AS Avg_Pizzas_Per_Order
FROM
    orders o
        JOIN
    order_details od ON o.ORDER_ID = od.Order_id
GROUP BY ORDER_DATE
ORDER BY ORDER_DATE;
```



OUTPUT

Result Grid | Filter Rows:

ORDER_DATE	Avg_Pizzas_P
2015-01-01	1.0062
2015-01-02	1.0313
2015-01-03	1.0260
2015-01-04	1.0000
2015-01-05	1.0331

TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE

```sql
SELECT
    pt.name, SUM(od.Quantity * p.price) AS Revenue
FROM
    order_details od
        JOIN
    pizzas p ON od.Pizza_id = p.pizza_id
        JOIN
    pizza_types pt ON p.pizza_type_id = pt.pizza_type_id
GROUP BY pt.name
ORDER BY Revenue DESC
LIMIT 3;
```

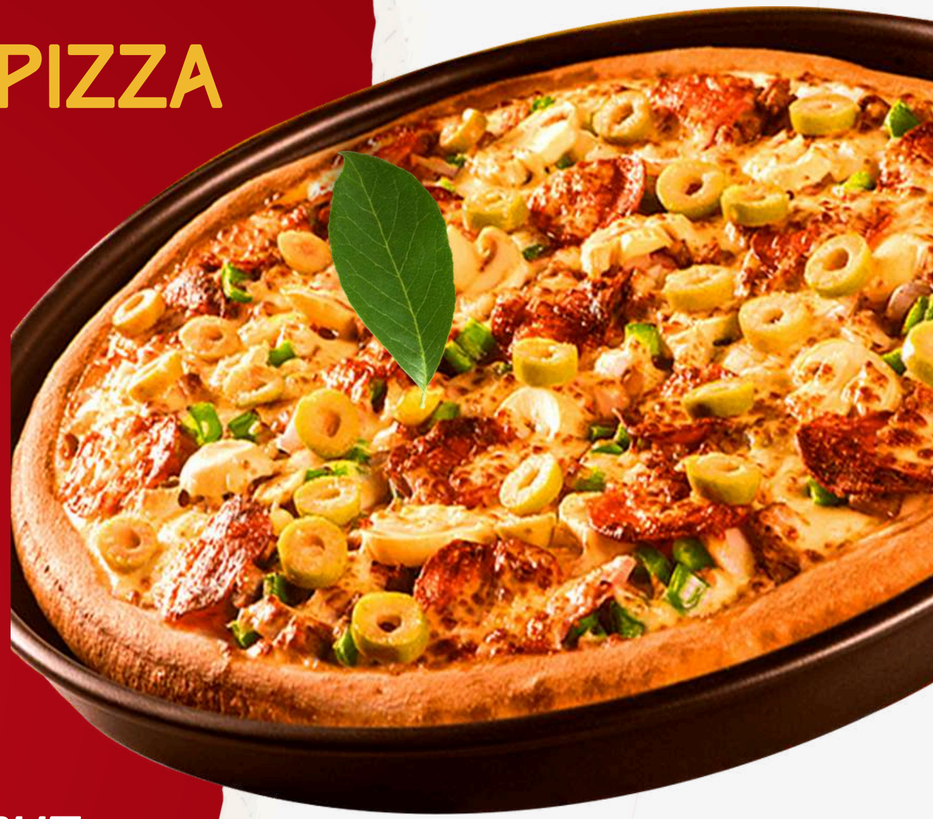
OUTPUT

Result Grid | Filter Rows:

name	Revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE

```sql
SELECT
    pt.name,
    SUM(od.Quantity * p.price) AS Revenue,
    ROUND((SUM(od.Quantity * p.price) / total.Total_Revenue) * 100,
            2) AS Percentage_Contribution
FROM
    order_details od
        JOIN
    pizzas p ON od.Pizza_id = p.pizza_id
        JOIN
    pizza_types pt ON p.pizza_type_id = pt.pizza_type_id
        JOIN
    (SELECT
        SUM(od.Quantity * p.price) AS Total_Revenue
    FROM
        order_details od
    JOIN pizzas p ON od.Pizza_id = p.pizza_id) total ON 1 = 1
GROUP BY pt.name , total.Total_Revenue
ORDER BY Revenue DESC;
```



OUTPUT

Result Grid | Filter Rows: | Export: | Wrap Cell Conte

name	Revenue	Percentage_Contribution
The Thai Chicken Pizza	43434.25	5.31
The Barbecue Chicken Pizza	42768	5.23
The California Chicken Pizza	41409.5	5.06
The Classic Deluxe Pizza	38180.5	4.67
The Spicy Italian Pizza	34831.25	4.26

CUMULATIVE REVENUE OVER TIME (BY ORDER DATE)

```sql
SELECT
    o.ORDER_DATE,
    SUM(od.Quantity * p.price) AS Daily_Revenue,
    @cum_revenue:=@cum_revenue + SUM(od.Quantity * p.price) AS Cumulative_Revenue
FROM
    orders o
        JOIN
    order_details od ON o.ORDER_ID = od.Order_ID
        JOIN
    pizzas p ON od.Pizza_id = p.pizza_id
        JOIN
    pizza_types pt ON p.pizza_type_id = pt.pizza_type_id
        CROSS JOIN
    (SELECT @cum_revenue:=0) AS vars
GROUP BY o.ORDER_DATE
ORDER BY o.ORDER_DATE;
```

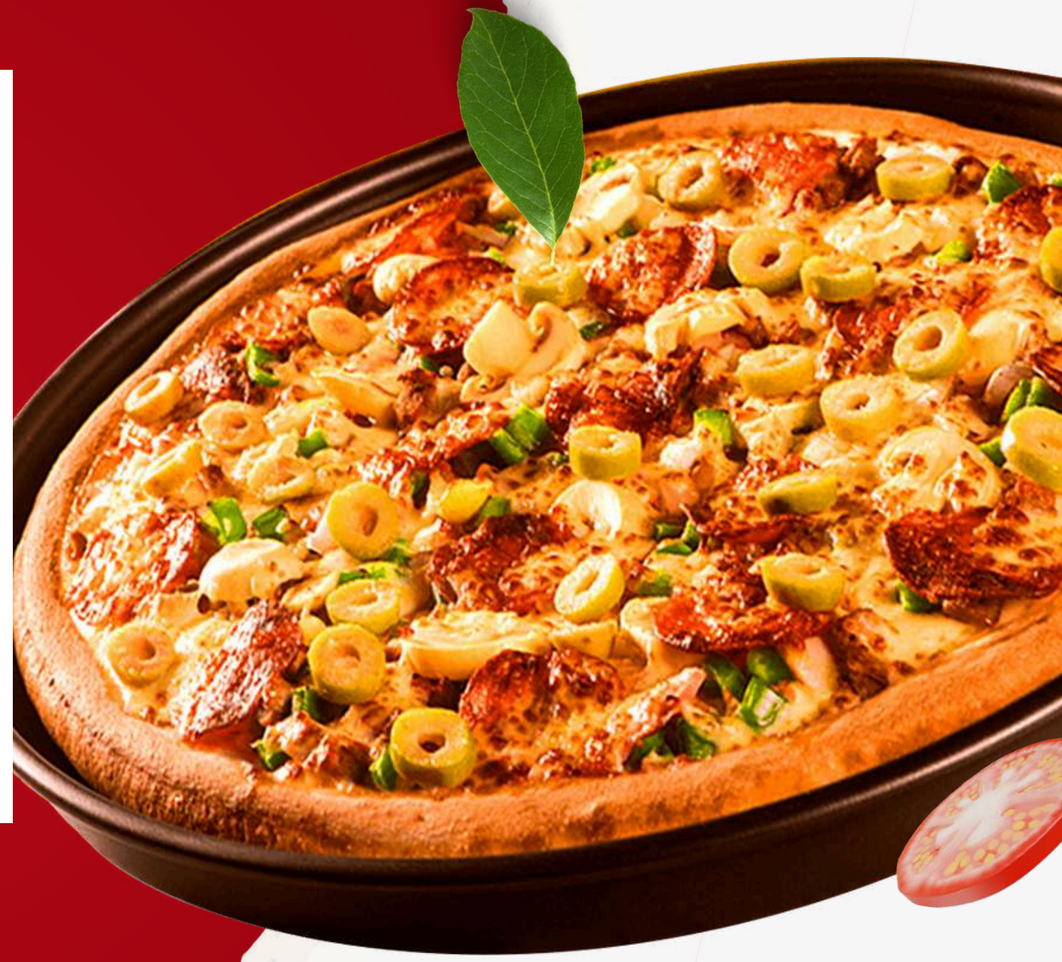
OUTPUT

Result Grid | Filter Rows: | Export:

ORDER_DATE	Daily_Revenue	Cumulative_Revenue
2015-01-01	2713.8500000000004	2713.8500000000004
2015-01-02	2731.8999999999996	2731.8999999999996
2015-01-03	2662.4	2662.4
2015-01-04	1755.4500000000003	1755.4500000000003
2015-01-05	2065.95	2065.95

TOP 3 MOST ORDERED PIZZA TYPES BY REVENUE PER CATEGORY.

```sql
SELECT
    category,
    name AS Pizza_Type,
    Revenue
FROM (
    SELECT
        pt.category,
        pt.name,
        SUM(od.Quantity * p.price) AS Revenue,
        ROW_NUMBER() OVER (PARTITION BY pt.category ORDER BY SUM(od.Quantity * p.price) DESC) AS rn
    FROM order_details od
    JOIN pizzas p ON od.Pizza_id = p.pizza_id
    JOIN pizza_types pt ON p.pizza_type_id = pt.pizza_type_id
    GROUP BY pt.category, pt.name
) AS ranked
WHERE rn <= 3
ORDER BY category, Revenue DESC;
```



OUTPUT

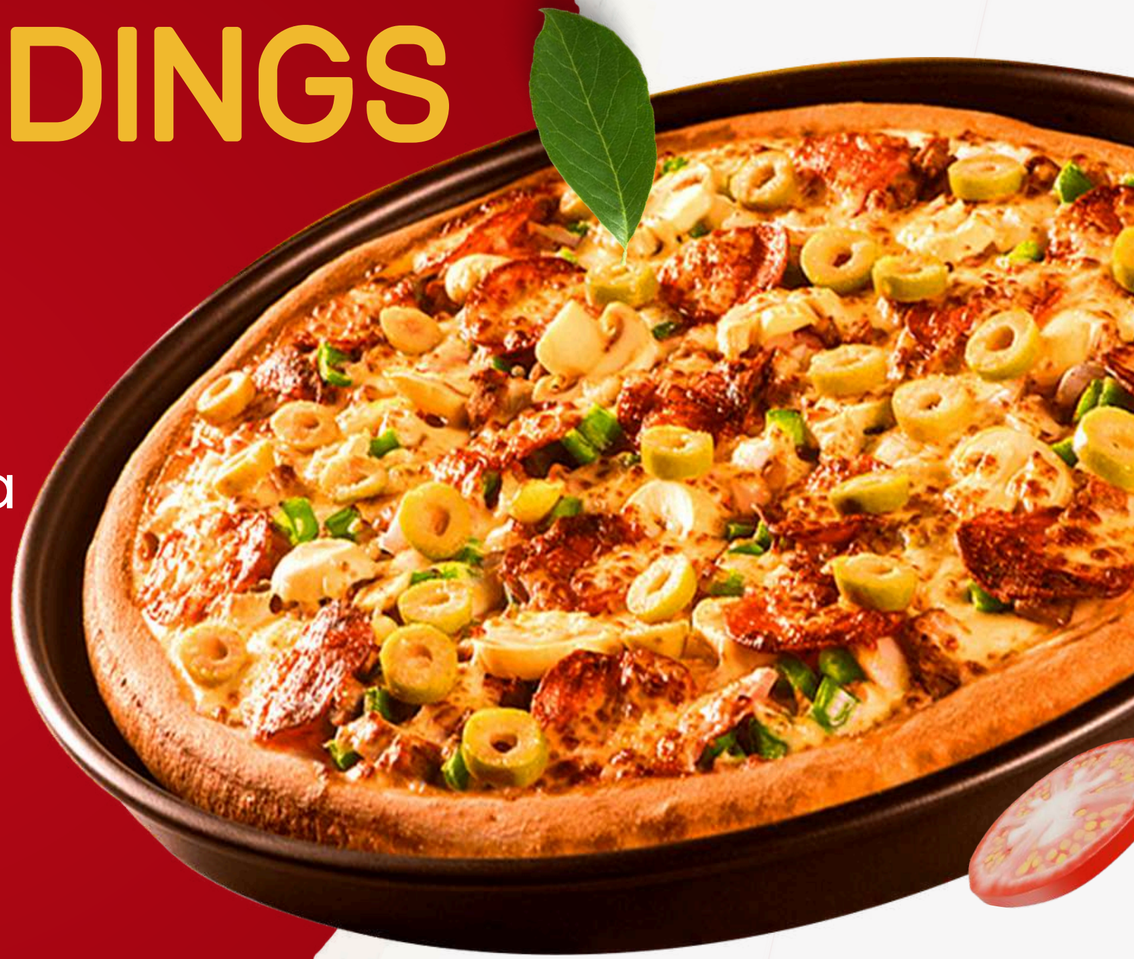
Result Grid | Filter Rows: | Exp

category	Pizza_Type	Revenue
Chicken	The Thai Chicken Pizza	43434.25
Chicken	The Barbecue Chicken Pizza	42768
Chicken	The California Chicken Pizza	41409.5
Classic	The Classic Deluxe Pizza	38180.5
Classic	The Hawaiian Pizza	
Classic	The Pepperoni Pizza	30161.75
Supreme	The Spicy Italian Pizza	34831.25
Supreme	The Italian Supreme Pizza	33476.75
Supreme	The Sicilian Pizza	30940.5
Veggie	The Four Cheese Pizza	32265.7000000

The Classic Deluxe

KEY INSIGHTS / FINDINGS

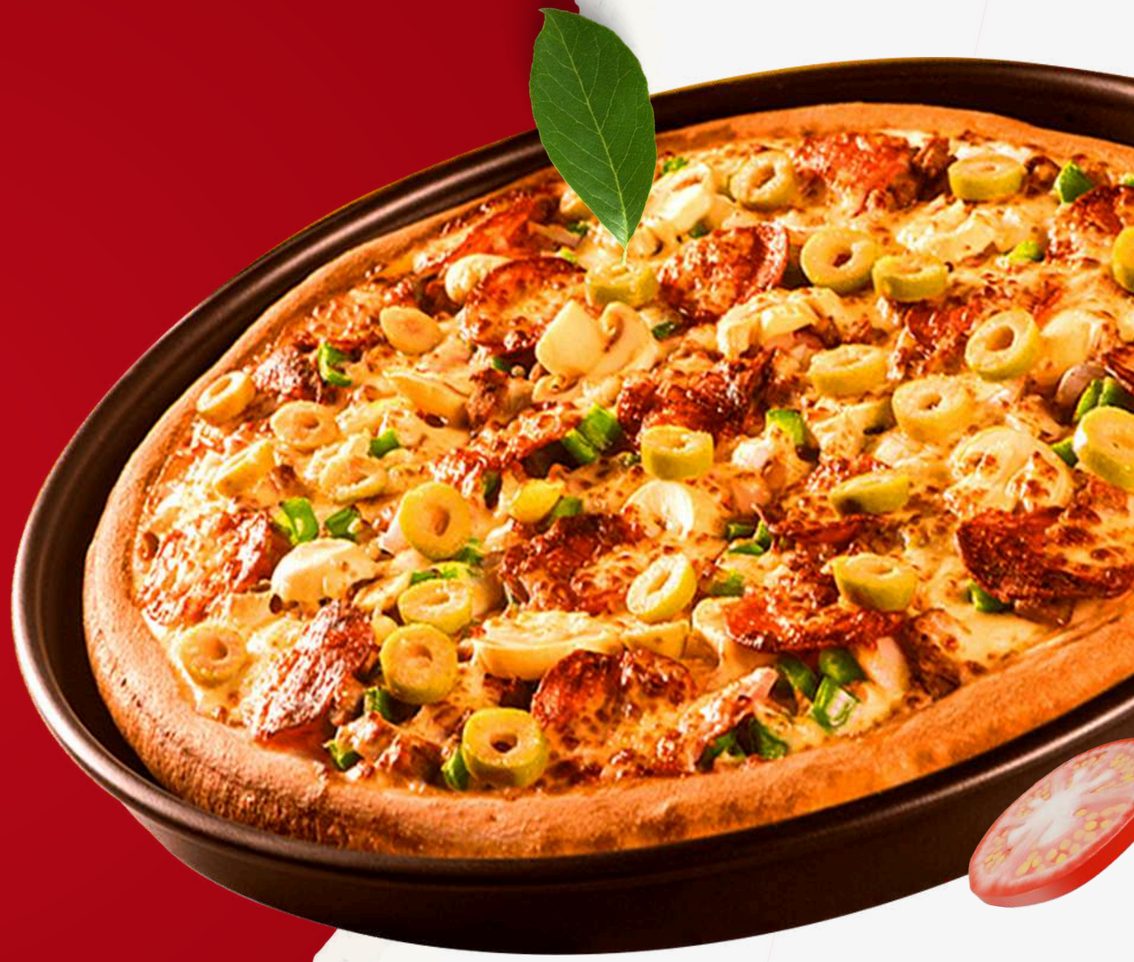
- Total Orders: 21,350
- Total Revenue: $817,860.05
- Most Popular Pizza: The Classic Deluxe Pizza
- Most Common Size: Medium (M)
- Peak Ordering Time: Around 12 PM
- Top Category by Revenue: Classic



CONCLUSION

- IDENTIFY TOP-SELLING PRODUCTS
- OPTIMIZE INVENTORY AND PRICING
- UNDERSTAND CUSTOMER BEHAVIOR

"DATA-DRIVEN DECISIONS BOOST SATISFACTION AND REVENUE."



THANK YOU!

Asheesh Faujdar