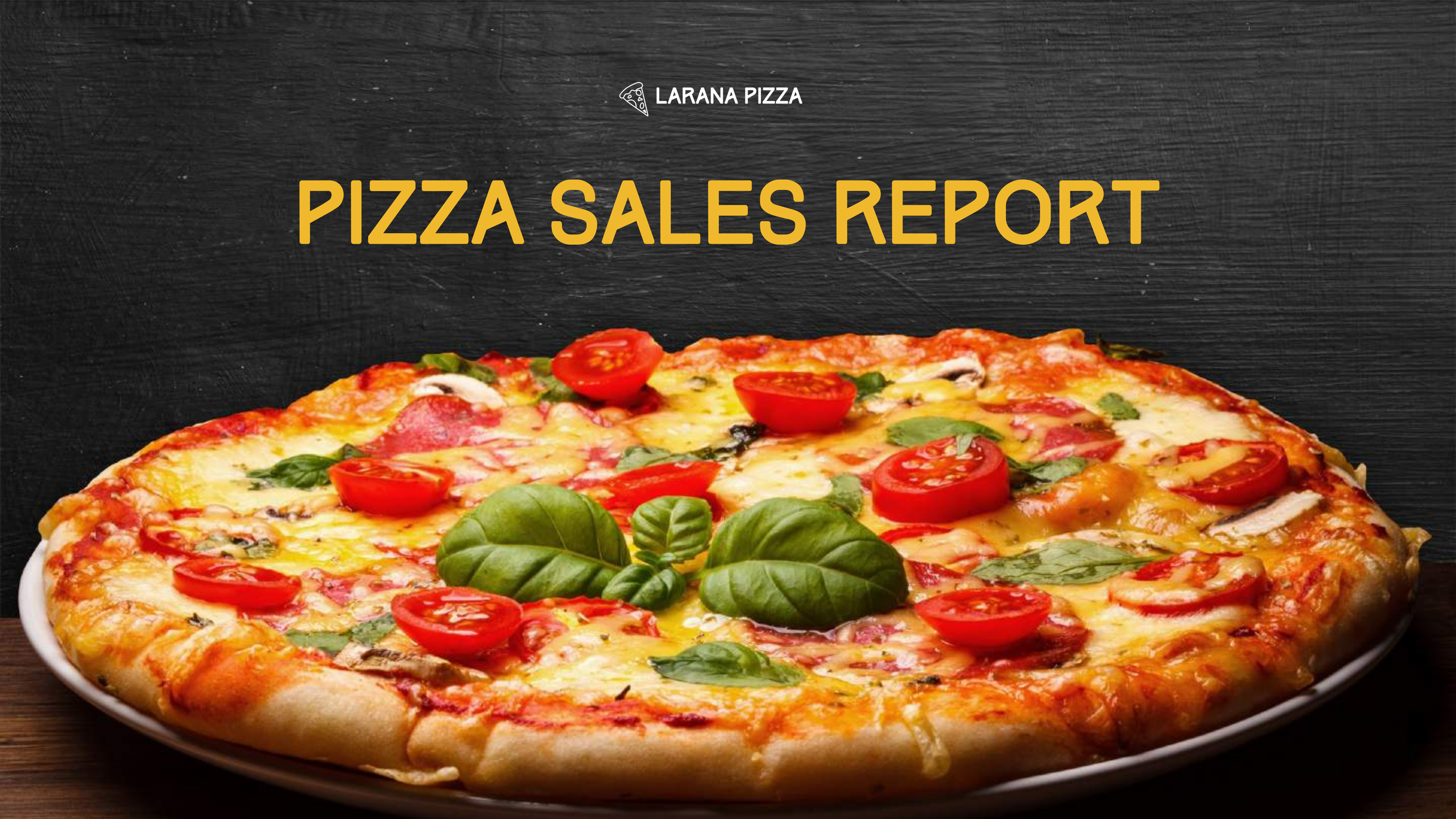
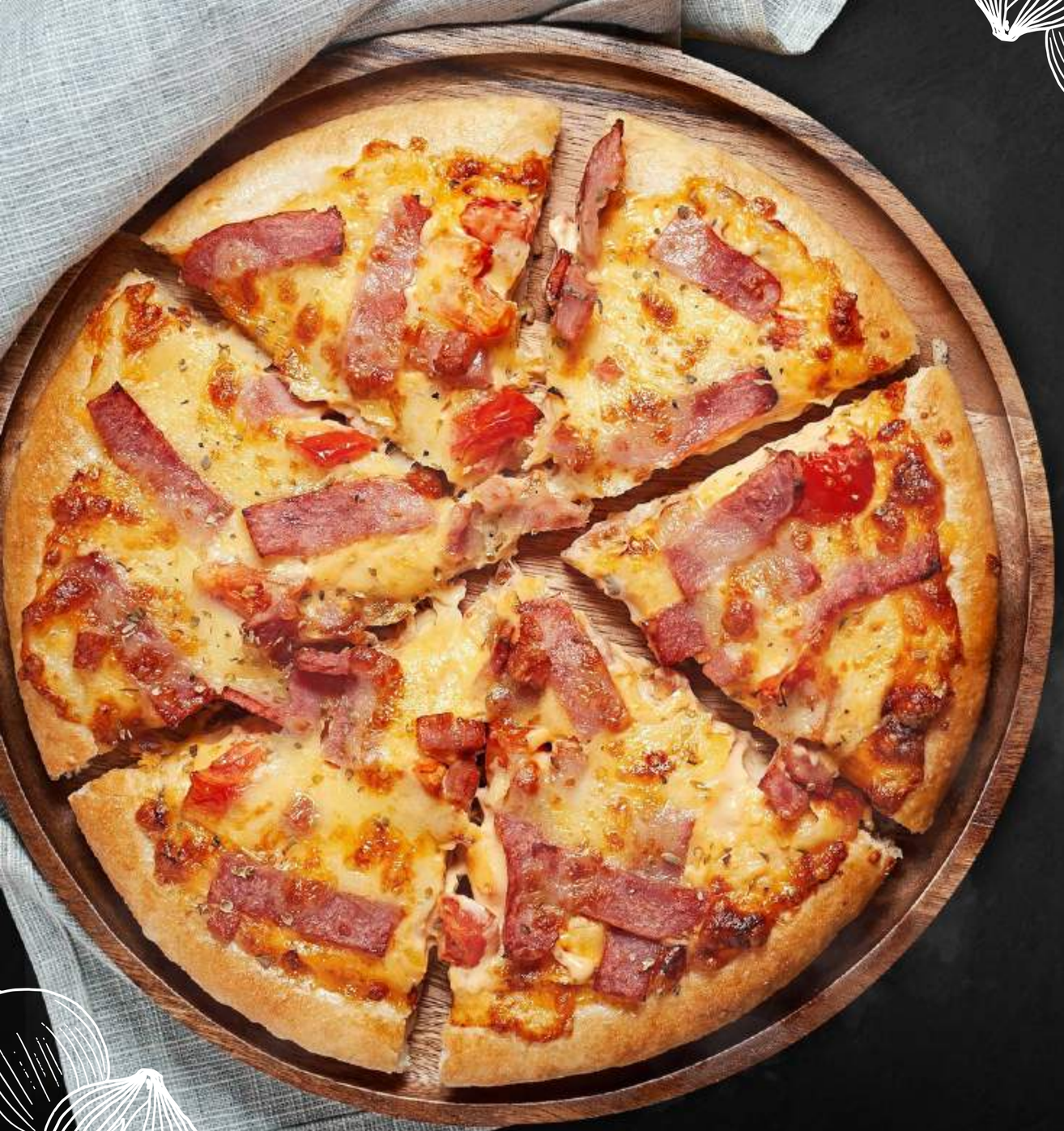
LARANA PIZZA

PIZZA SALES REPORT





REPORT BY ASHUTOSH PATHAK

This report provides a comprehensive analysis of pizza sales using SQL, highlighting key insights into order trends and revenue.

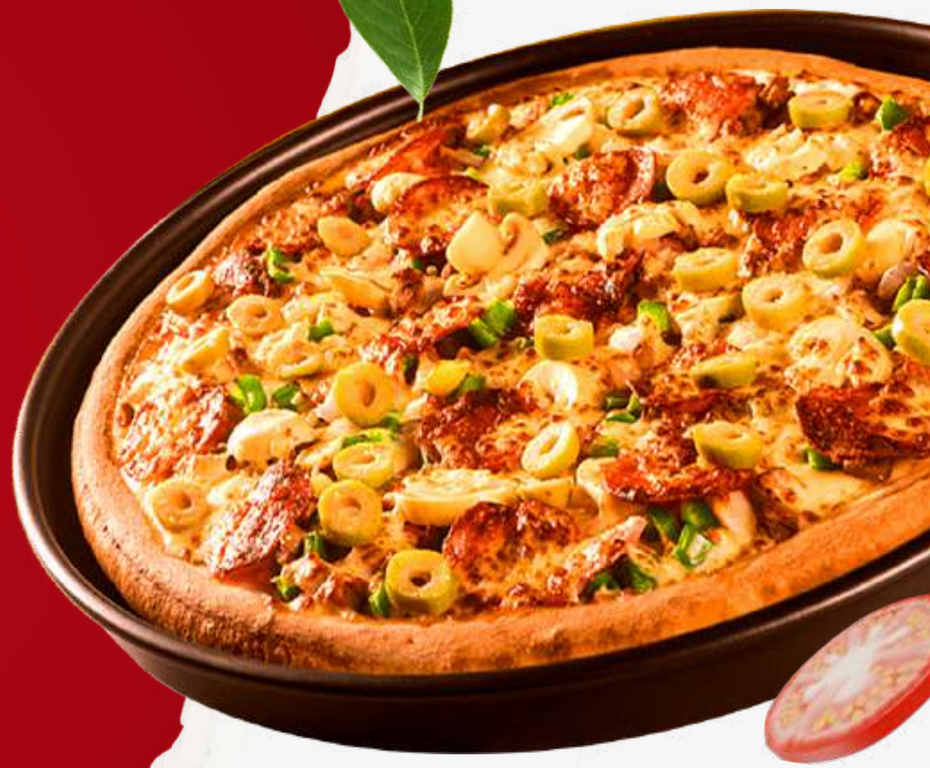
Key Insights:

1. Total Orders & Sales: Overview of the total number of orders and overall sales.
2. Highest Price Pizza: Identifying the most expensive pizza and its sales contribution.
3. Most Common Pizza Size: The most frequently ordered pizza size.
4. Top 5 Most Ordered Pizzas: Ranking the top 5 pizzas by order quantity.
5. Category-wise Pizza Distribution: Total quantity of pizzas ordered per category.
6. Order Distribution by Hour: Analyzing orders based on the time of day.
7. Average Pizzas Ordered Per Day: Average pizza orders per day.
8. Top 3 Pizza Types by Revenue: Most ordered pizzas based on revenue.
9. Revenue Contribution by Pizza Type: Percentage of total revenue per pizza type.
10. Cumulative Revenue Over Time: Tracking revenue growth over time.

RETRIVE THE TOTAL NUMBER OF ORDER PLACE.

```sql
SELECT
    COUNT(order_id) AS total_orders
FROM
    orders;
```

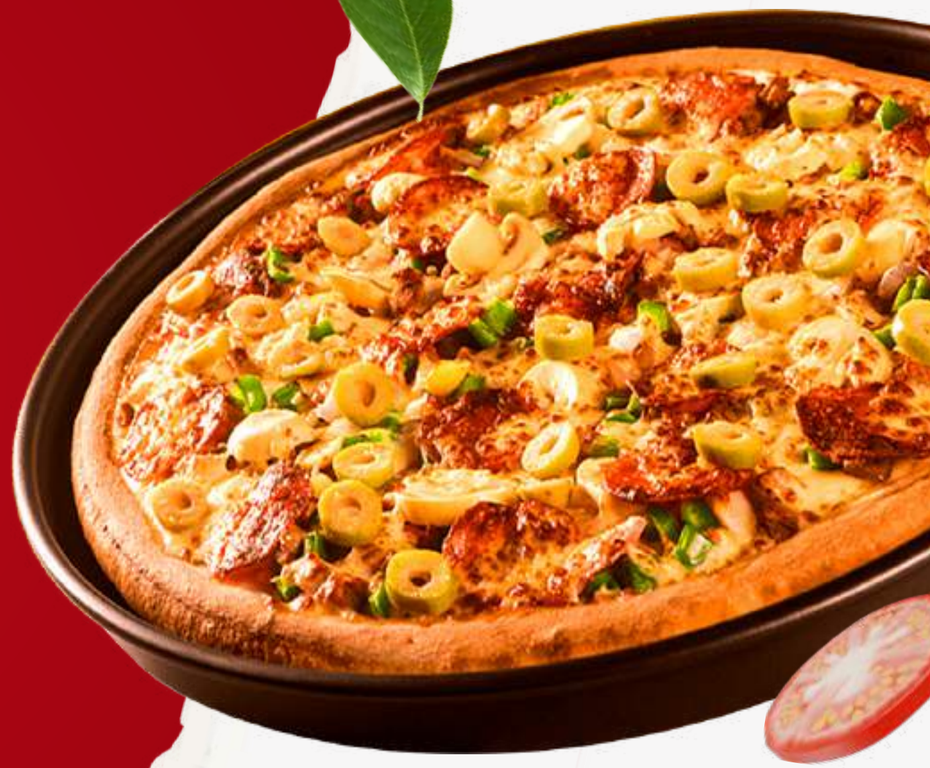
	total_orders
▶	21350



-- CALCULATE THE TOTAL REVENUE GENERATED FROM THE PIZZA SALES ?

```
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS total_sales
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id;
```

	total_sales
▶	817860.05



IDENTIFY THE HIGHEST PRICE PIZZAS.

```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;
```

	name	price
▶	The Greek Pizza	35.95

-- IDENTIFY THE MOST COMMON PIZZA SIZE ORDERD

```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
LIMIT 1;
```

	size	order_count
▶	L	18526

-- IDENTIFY THE MOST COMMON PIZZA SIZE ORDERD

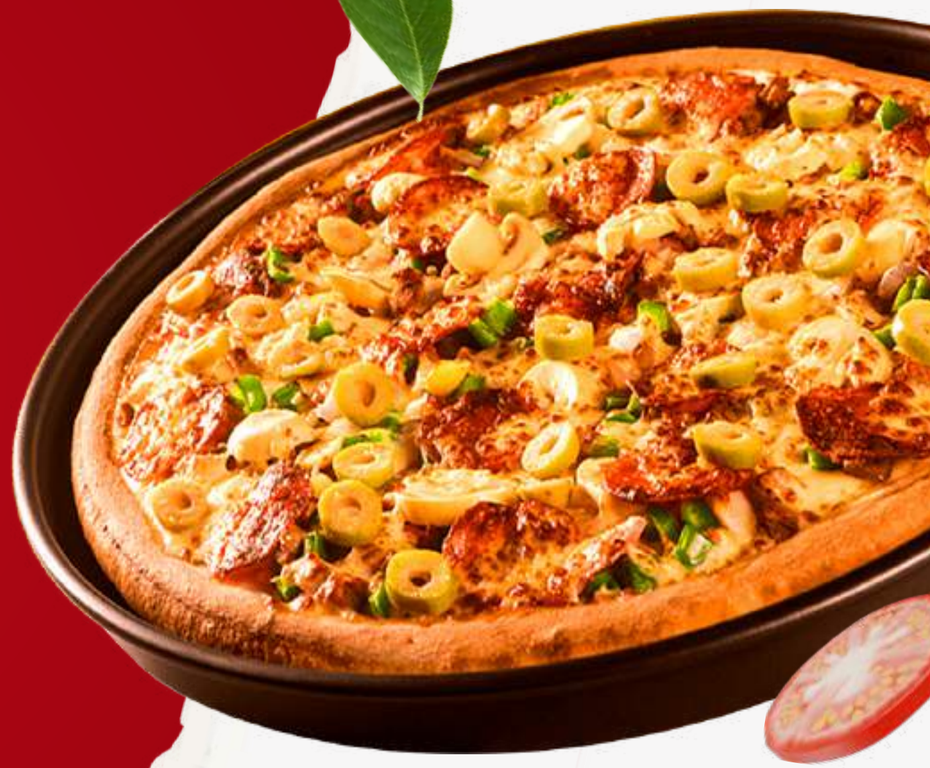
```sql
SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) AS order_count
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC
LIMIT 1;
```

	size	order_count
▶	L	18526

-- LIST THE TOP 5 MOST ORDERD PIZZA TYPE ALONG WITH THEIR QUANTITITES

```sql
SELECT
    pizza_types.name, SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT 5;
```

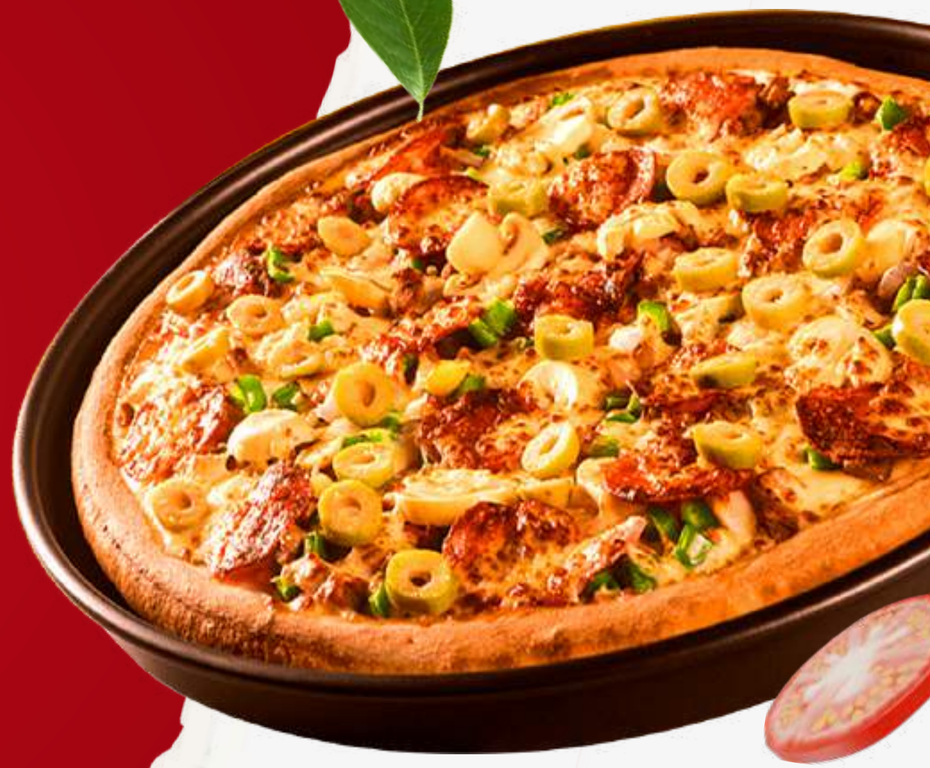
	name	quantity
▶	The Classic Deluxe Pizza	2453
	The Barbecue Chicken Pizza	2432
	The Hawaiian Pizza	2422
	The Pepperoni Pizza	2418
	The Thai Chicken Pizza	2371



JOIN THE NECESSARY TABLE TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERD.

```sql
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity DESC
LIMIT 5;
```

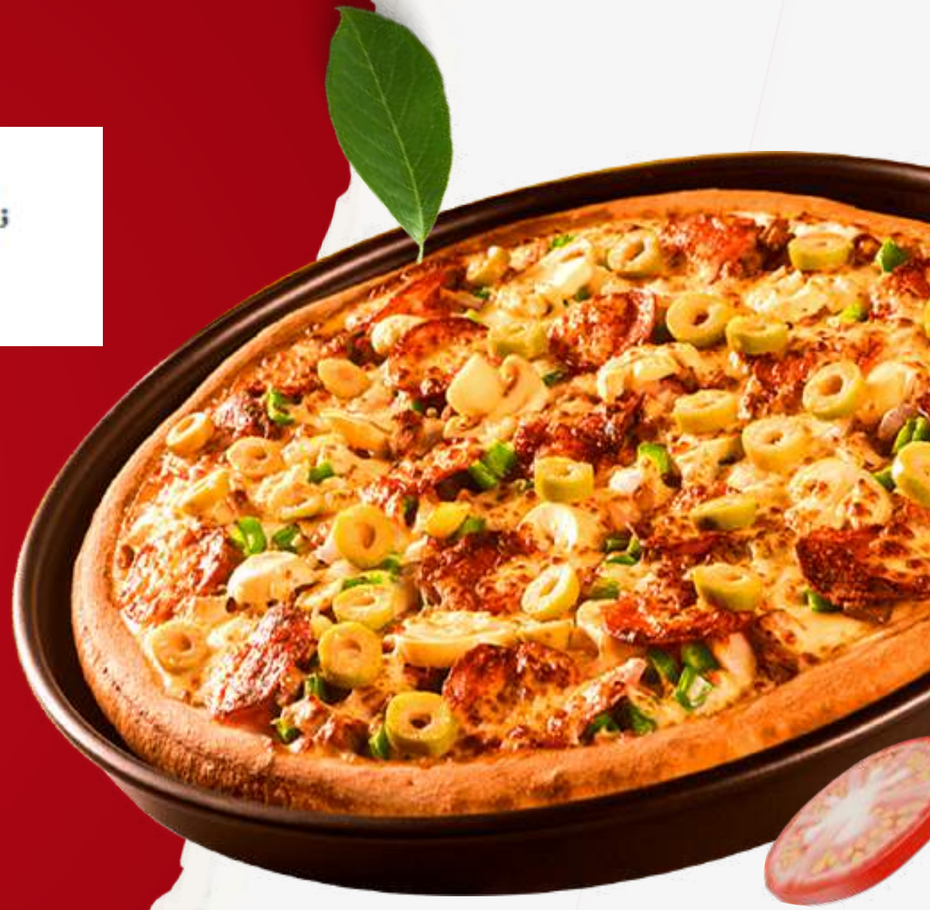
	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050



-- DETERMINE THE DISTRUBITION OF ORDER BY HOUR OF THE DAY

```sql
Select hour(time) as hour, count(order_id) as order_count from orders group by hour(time);
```

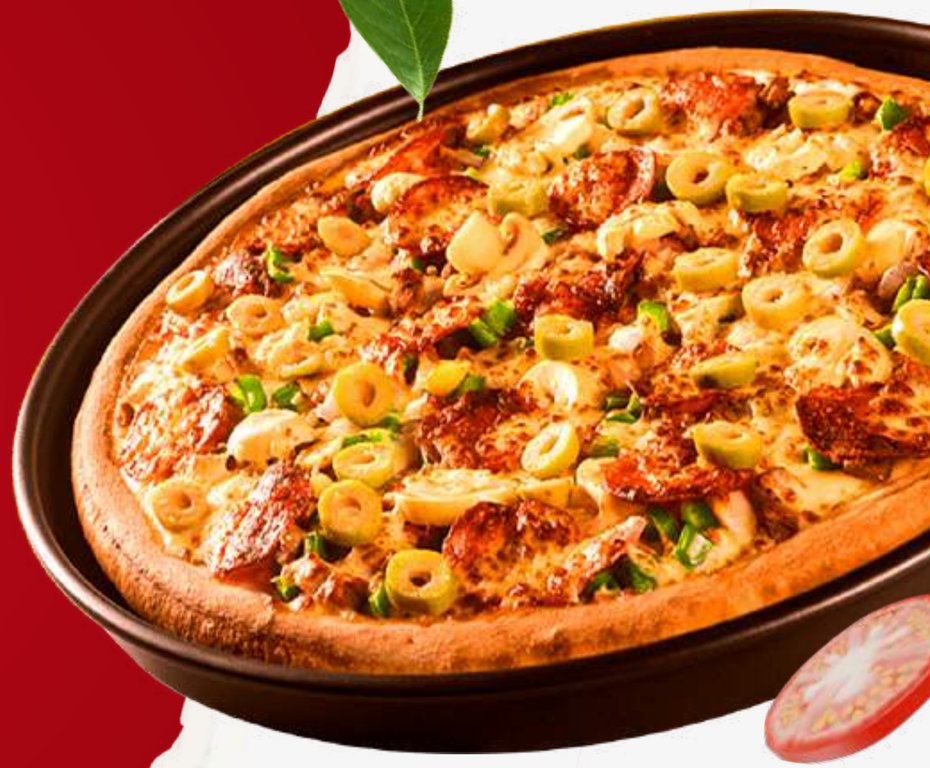
hour	order_count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399



-- JOIN RELEVANT TABLE TO FIND THE CATEGORY WISE DISTRIBUTION OF PIZZAS

```
SELECT
    category, COUNT(name)
FROM
    pizza_types
GROUP BY category;
```

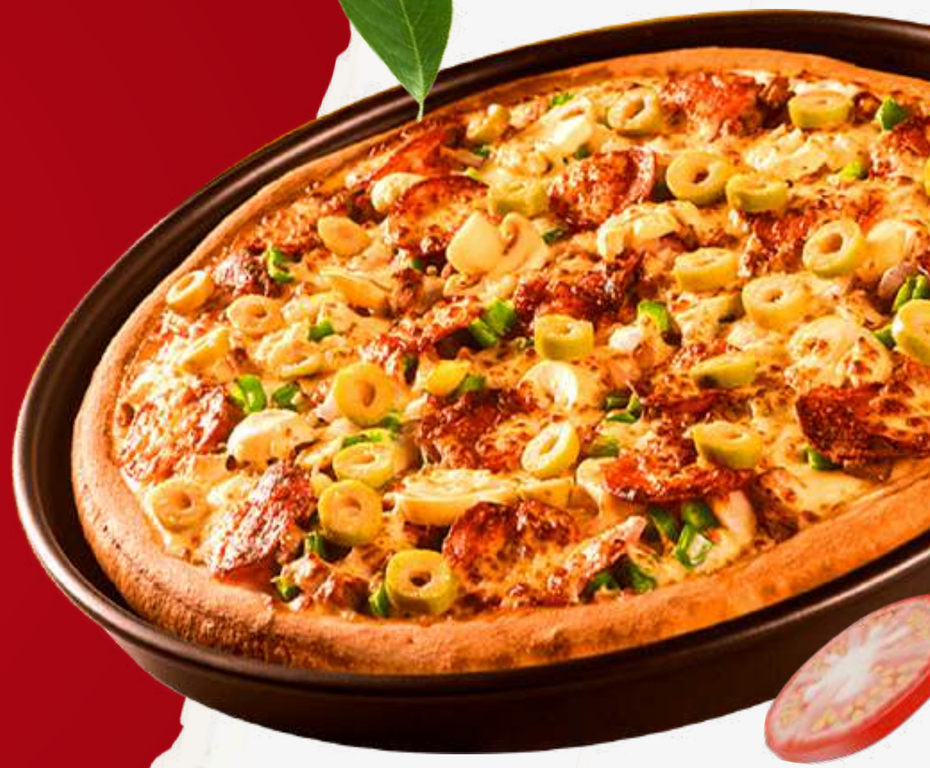
category	COUNT(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9



-- JOIN RELEVANT TABLE TO FIND THE CATEGORY WISE DISTRIBUTION OF PIZZAS

```
SELECT
    category, COUNT(name)
FROM
    pizza_types
GROUP BY category;
```

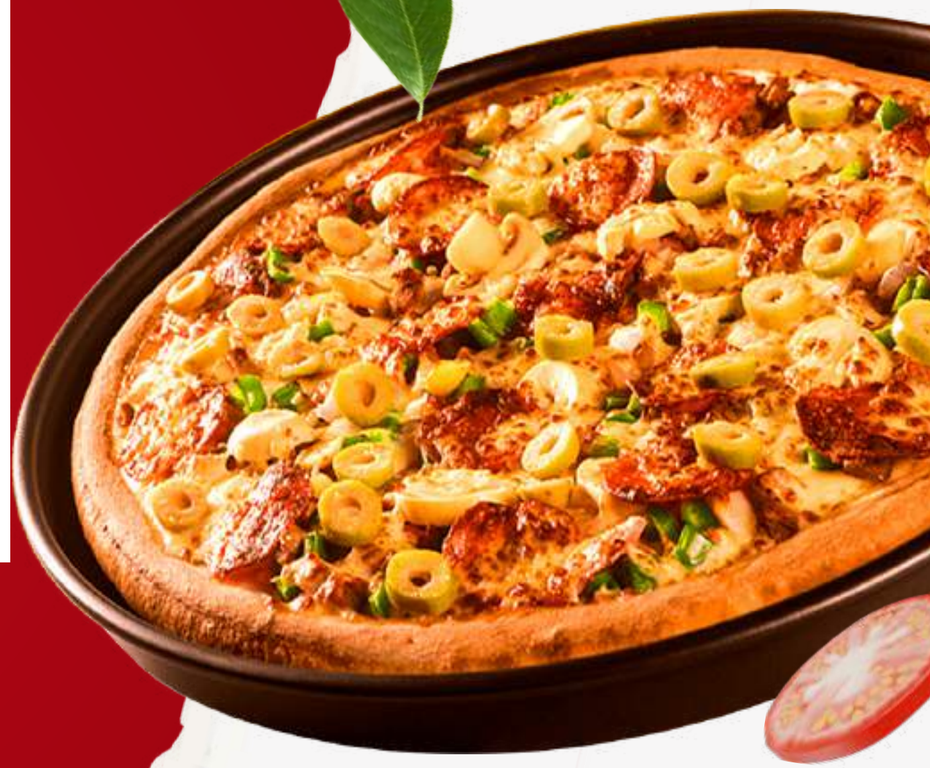
category	COUNT(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9



-- GROUP THE ORDER BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDER PER DAY.

```sql
SELECT
    ROUND(AVG(quantity), 0)
FROM
    (SELECT
        orders.date, SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.date) AS order_quantity;
```

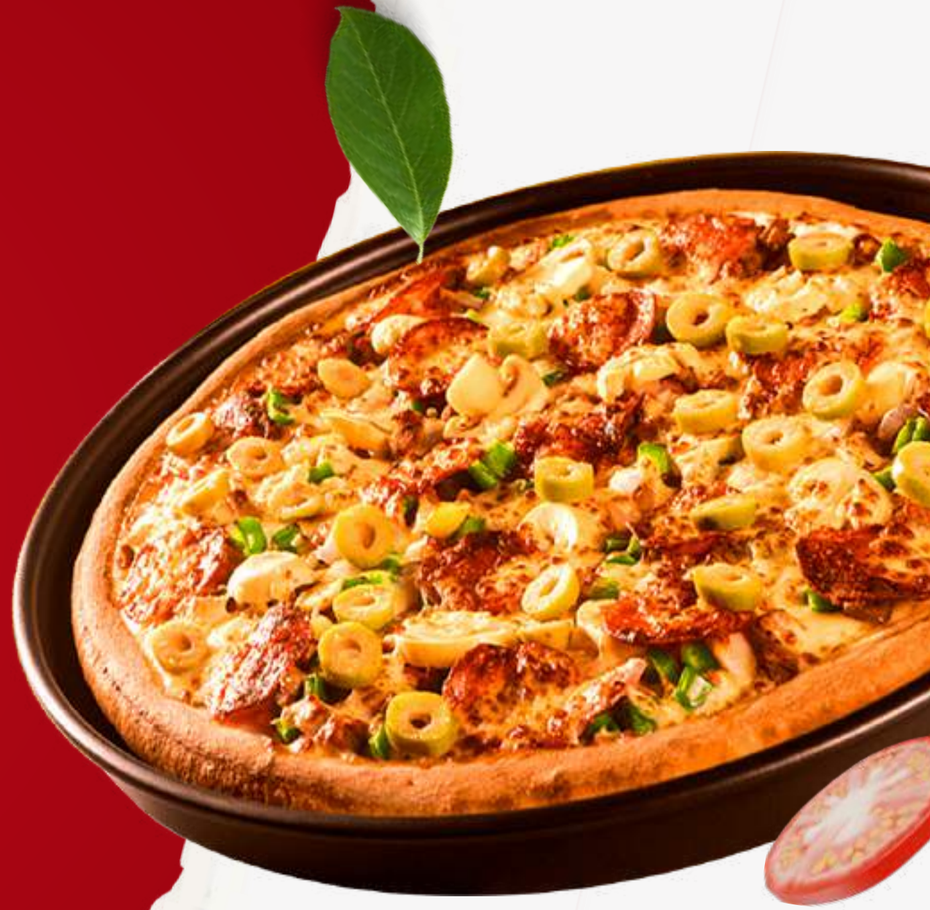
	ROUND(AVG(quantity), 0)
▶	138



-- DETERMINE THE TOP 3 MOST ORDERD PIZZA TYPE BASED ON REVENUE.?

```sql
SELECT
    pizza_types.name,
    SUM(order_details.quantity * pizzas.price) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue DESC
LIMIT 3;
```

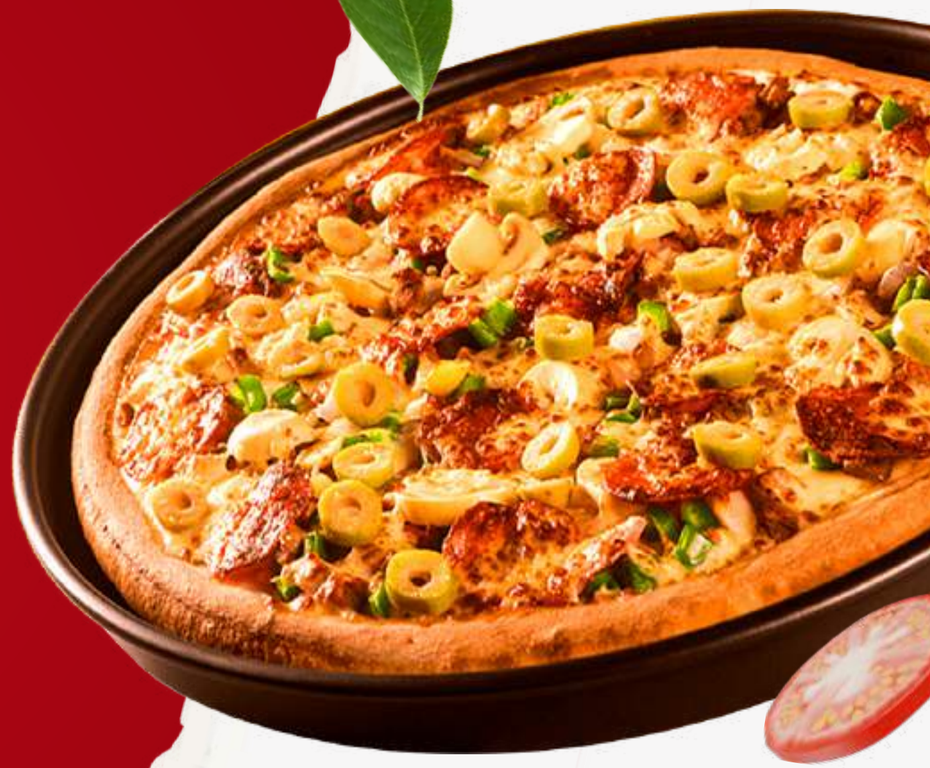
name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5



-- CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.

```sql
SELECT
    pizza_types.category,
    ROUND(
        (SUM(order_details.quantity * pizzas.price) /
        (SELECT SUM(order_details.quantity * pizzas.price)
         FROM order_details
         JOIN pizzas ON pizzas.pizza_id = order_details.pizza_id)) * 100, 2
    ) AS revenue_percentage
FROM pizza_types
JOIN pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
JOIN order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY revenue_percentage DESC;
```

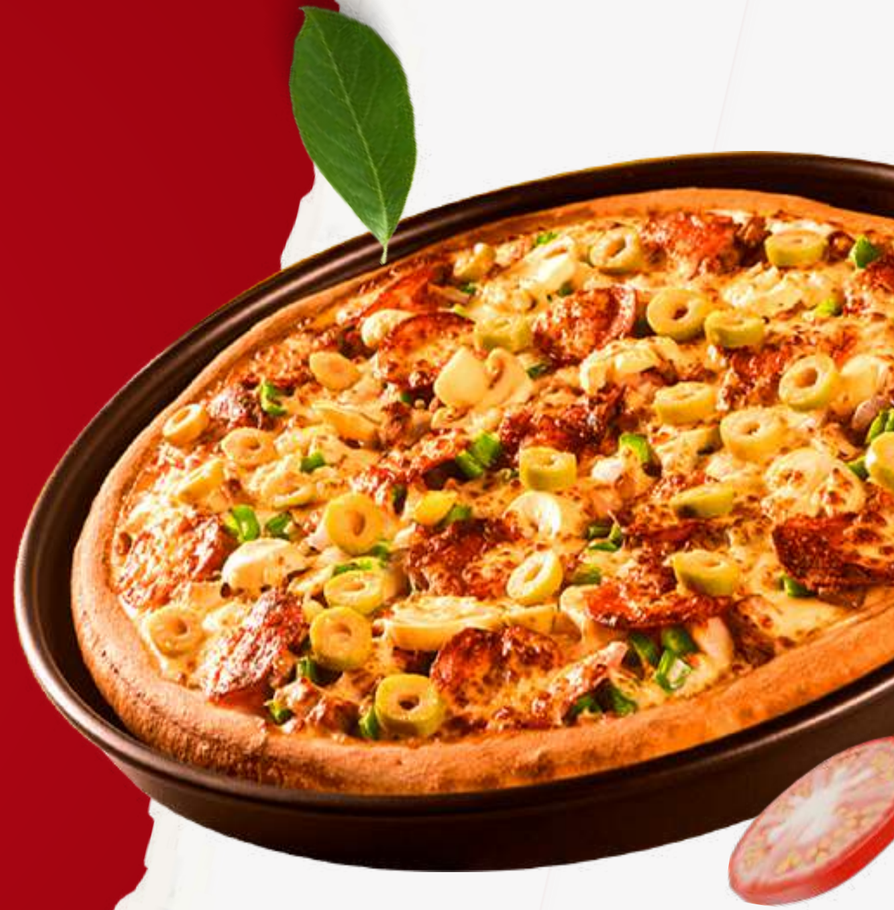
category	revenue_percentage
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68



-- ANALYZE THE CUMULATIVE REVENUE GENERATED OVERTIME.

```sql
SELECT date,
       SUM(revenue) OVER (ORDER BY date) AS cum_revenue
FROM (
    SELECT orders.date,
           SUM(order_details.quantity * pizzas.price) AS revenue
    FROM order_details
    JOIN pizzas ON order_details.pizza_id = pizzas.pizza_id
    JOIN orders ON orders.order_id = order_details.order_id
    GROUP BY orders.date
) AS sales;
```

date	cum_revenue
2015-01-01 00:00:00	2713.8500000000004
2015-01-02 00:00:00	5445.75
2015-01-03 00:00:00	8108.15
2015-01-04 00:00:00	9863.6
2015-01-05 00:00:00	11929.55



-- DETERMINE THE TOP 3 MOST ORDERD PIZZA TYPE BASED ON REVENUE FOR EACH PIZZA CATEGORY

```sql
Select name, revenue from
(select category, name, revenue,
rank() over(partition by category order by revenue desc) as rn from
(select pizza_types.category, pizza_types.name, sum((order_details.quantity)*pizzas.price) as revenue
from pizza_types join pizzas on pizza_types.pizza_type_id=pizzas.pizza_type_id
join order_details
on order_details.pizza_id=pizzas.pizza_id
group by pizza_types.category, pizza_types.name)as a) as b
where rn<=3;
```

name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5
The Classic Deluxe Pizza	38180.5
The Hawaiian Pizza	32273.25

38180.5

TOTAL PIZZA ORDERS: 21,350

TOTAL SALES: 817860.05

HIGHEST PRICR PIZZA: THE GREEK PIZZA(35.95)

COMMON PIZZA SIZE ORDERD: LARGE

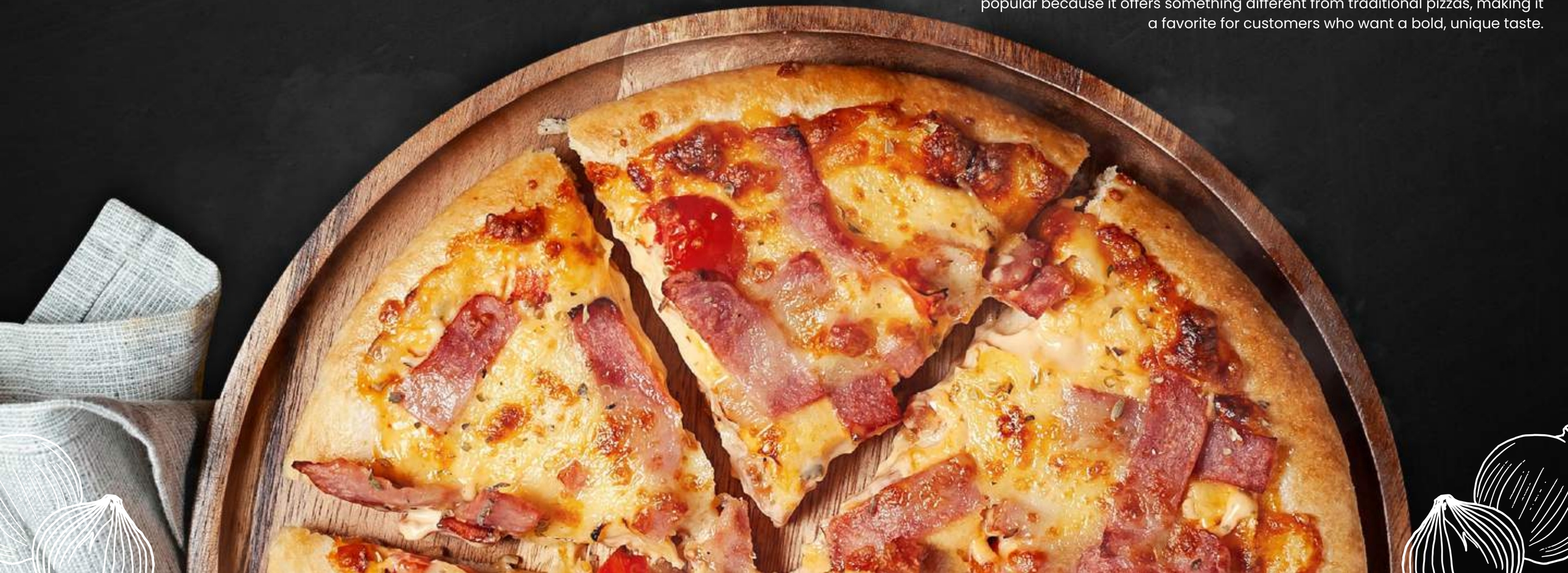
AVERAGE NUMBER OF PIZZA ORDERD PER DAY: 138

BEST SELLER

★★★★★

THE THAI CHICEK PIZZA

The Thai Chicken Pizza is the best-seller because it combines delicious Thai flavors with pizza. It's topped with grilled chicken, fresh vegetables, and a sweet and spicy Thai sauce. The pizza is known for its balance of sweet and spicy, with toppings like peanuts and cilantro that add extra flavor and texture. This pizza is popular because it offers something different from traditional pizzas, making it a favorite for customers who want a bold, unique taste.



PIZZA DATABASE

Analyzed a pizza sales dataset to uncover actionable insights using SQL. Key highlights include total orders, revenue trends, top-performing pizza categories, and time-based order patterns.

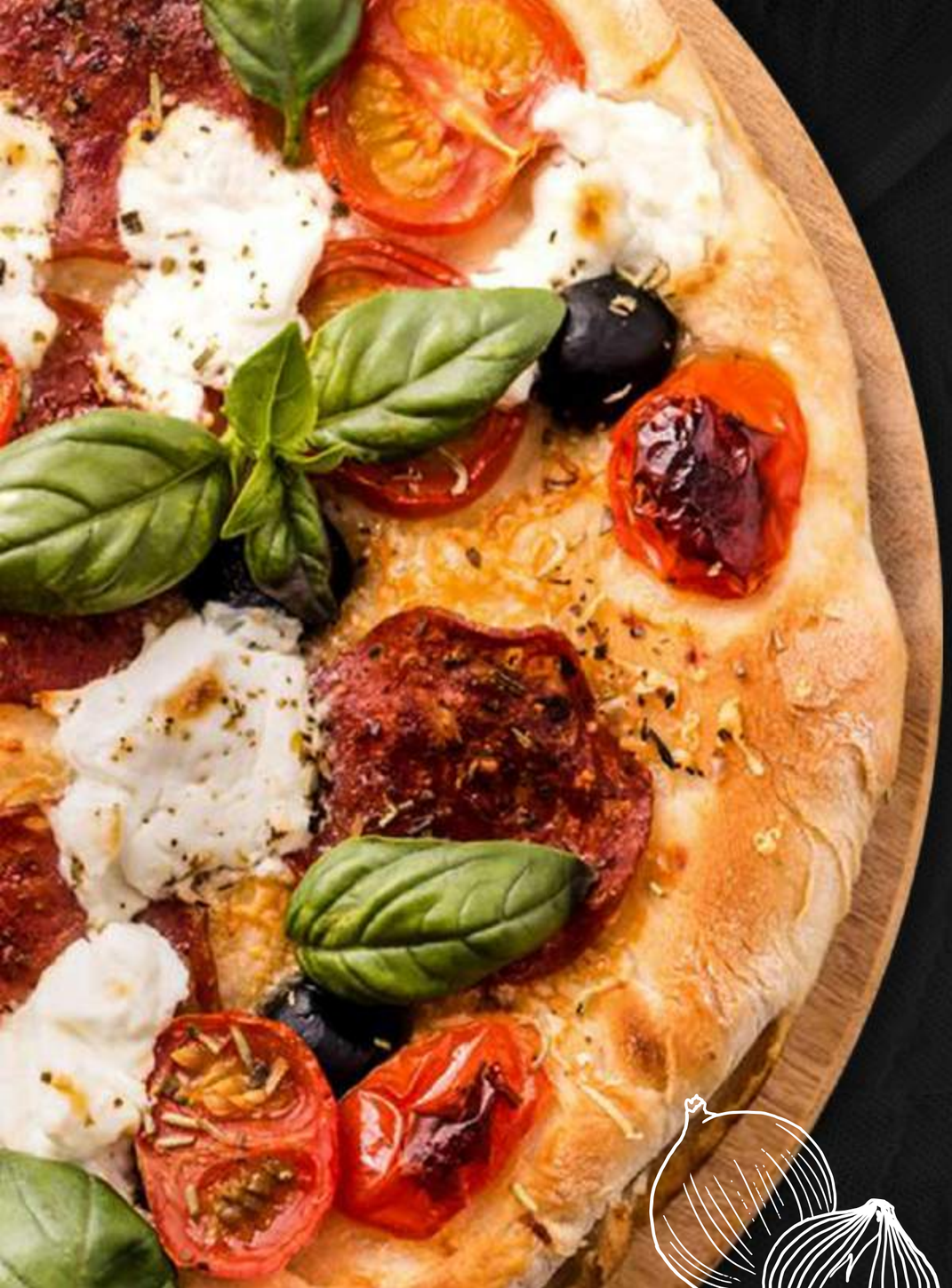
The Pizza Database is a comprehensive dataset that captures the details of pizza sales and orders. It consists of four interconnected tables that provide insights into orders, pizza types, and their variations. The first table, order_details, records the specifics of each order, such as the unique identifier for the order details, the associated order ID, the pizza ID, and the quantity of pizzas ordered. This table serves as the link between individual pizzas and the orders they belong to.

The second table, orders, contains the overarching information about each customer order, including a unique order ID, the date the order was placed, and the time of the transaction. This table is essential for understanding the temporal trends in pizza sales, such as peak ordering hours or days.

The third table, pizza_types, provides information about the different types of pizzas available. Each pizza type is assigned a unique ID and includes attributes such as the pizza's name, its category (e.g., Veggie, Chicken, Supreme), and a list of its ingredients. This table offers a detailed breakdown of the variety of pizzas offered and their characteristics.

Lastly, the pizzas table connects pizza types to their specific variations. Each entry in this table has a unique pizza ID and links back to its corresponding pizza type. Additionally, it specifies the size of the pizza (e.g., Small, Medium, Large) and its price.

Together, these tables provide a robust framework for analyzing pizza sales, including customer preferences, revenue trends, and product performance. This database is a valuable resource for exploring data-driven strategies to enhance business operations.



LARANA PIZZA

CONTACT

6392579889

6392579889ashu@gmail.com

D45/140 varanasi

THANK YOU!