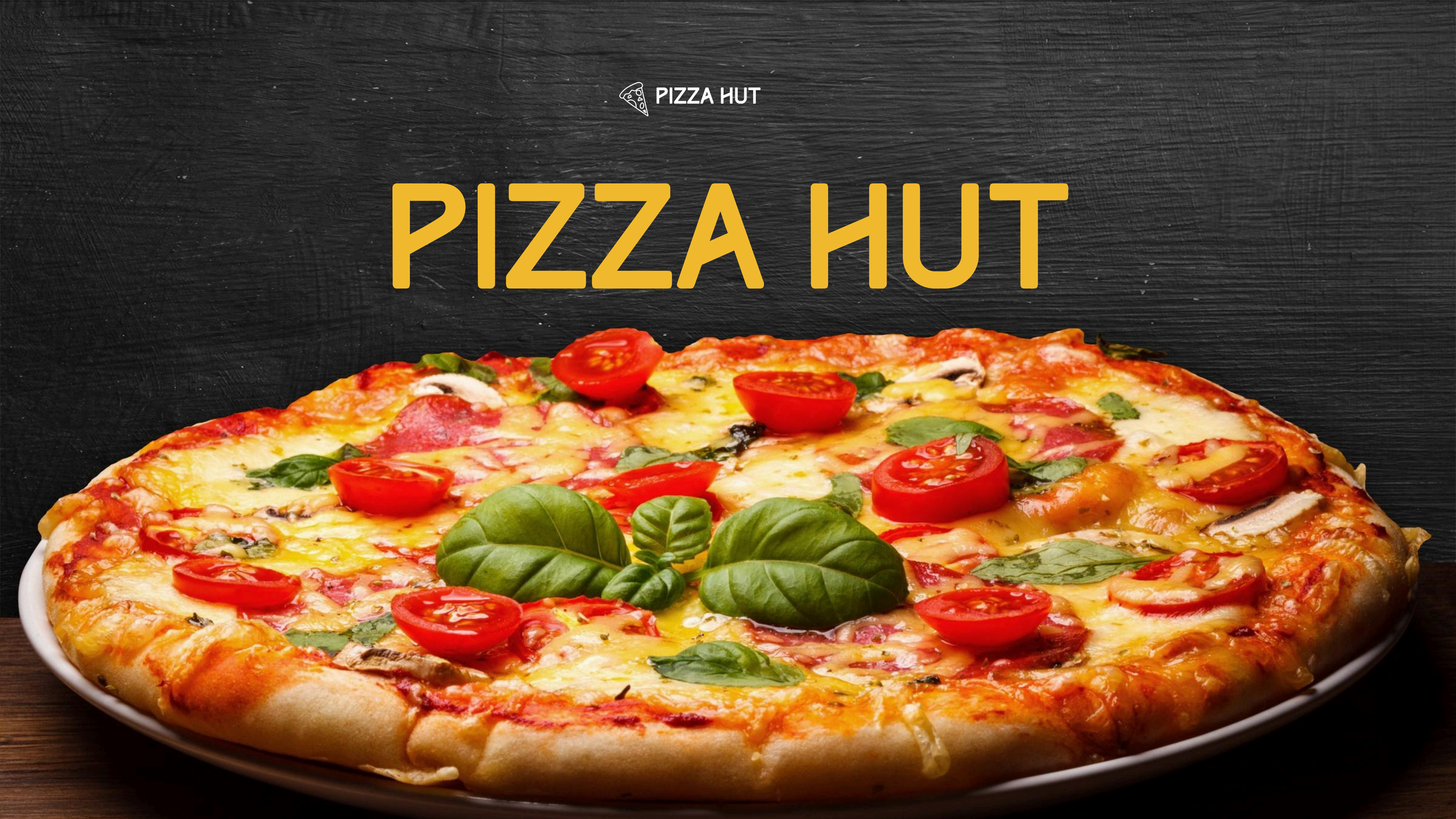
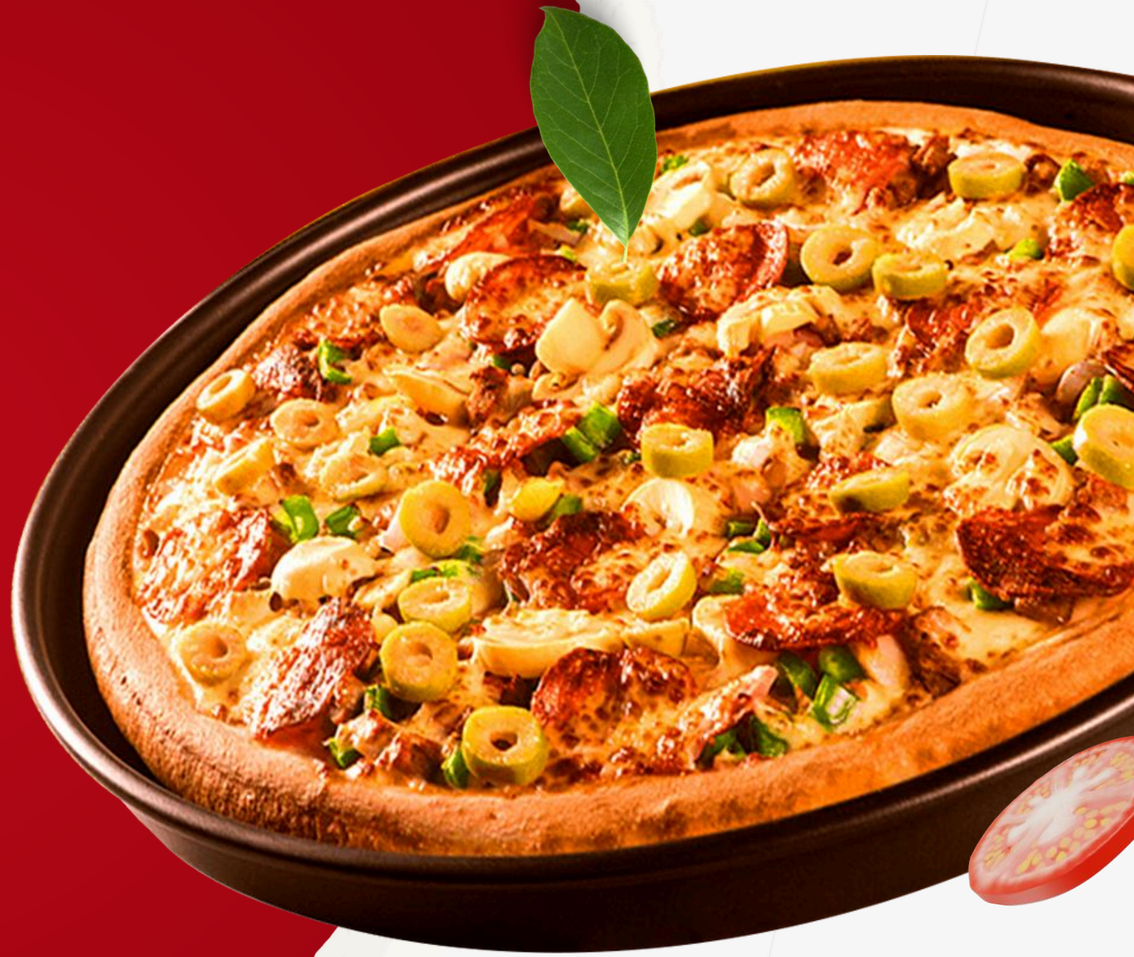
PIZZA HUT

PIZZA HUT



PIZZA HUT

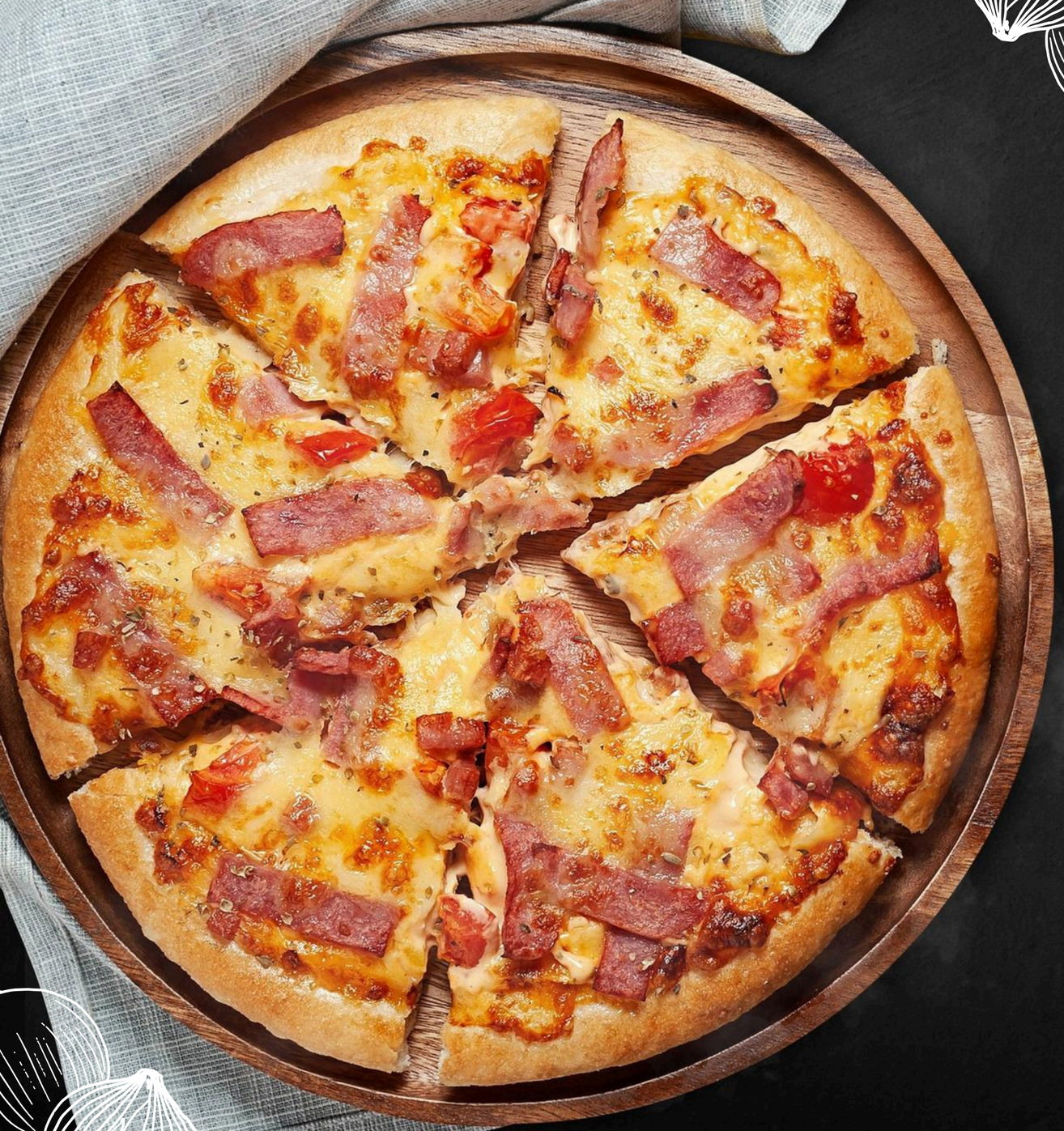
WELCOME TO
PIZZA HUT



Hello,

Im amit chavhan, In this project, we will analyze Pizza Hut's sales data using MySQL, using various SQL queries.

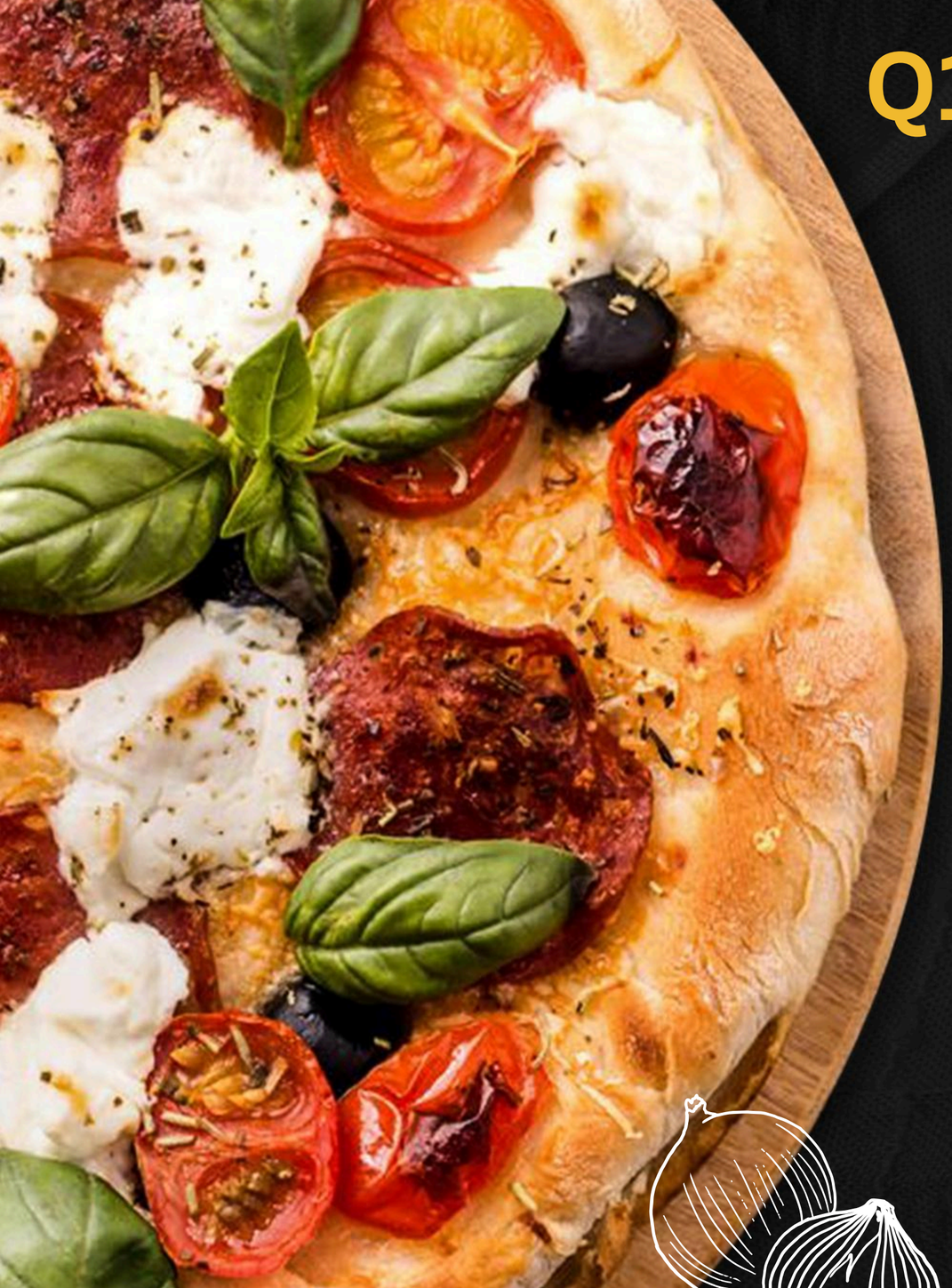
In this project, I address various SQL questions to analyze Pizza Hut's sales data, focusing on discovering valuable insights that can drive better business decisions. The project leverages MySQL to solve real-world problems related to Pizza Hut's operations, helping to understand customer preferences, sales performance, and trends over time.



VISSION

VISSION

The primary goal of the project is to explore the pizza sales data, understand customer preferences, and generate actionable insights by querying and analyzing the dataset

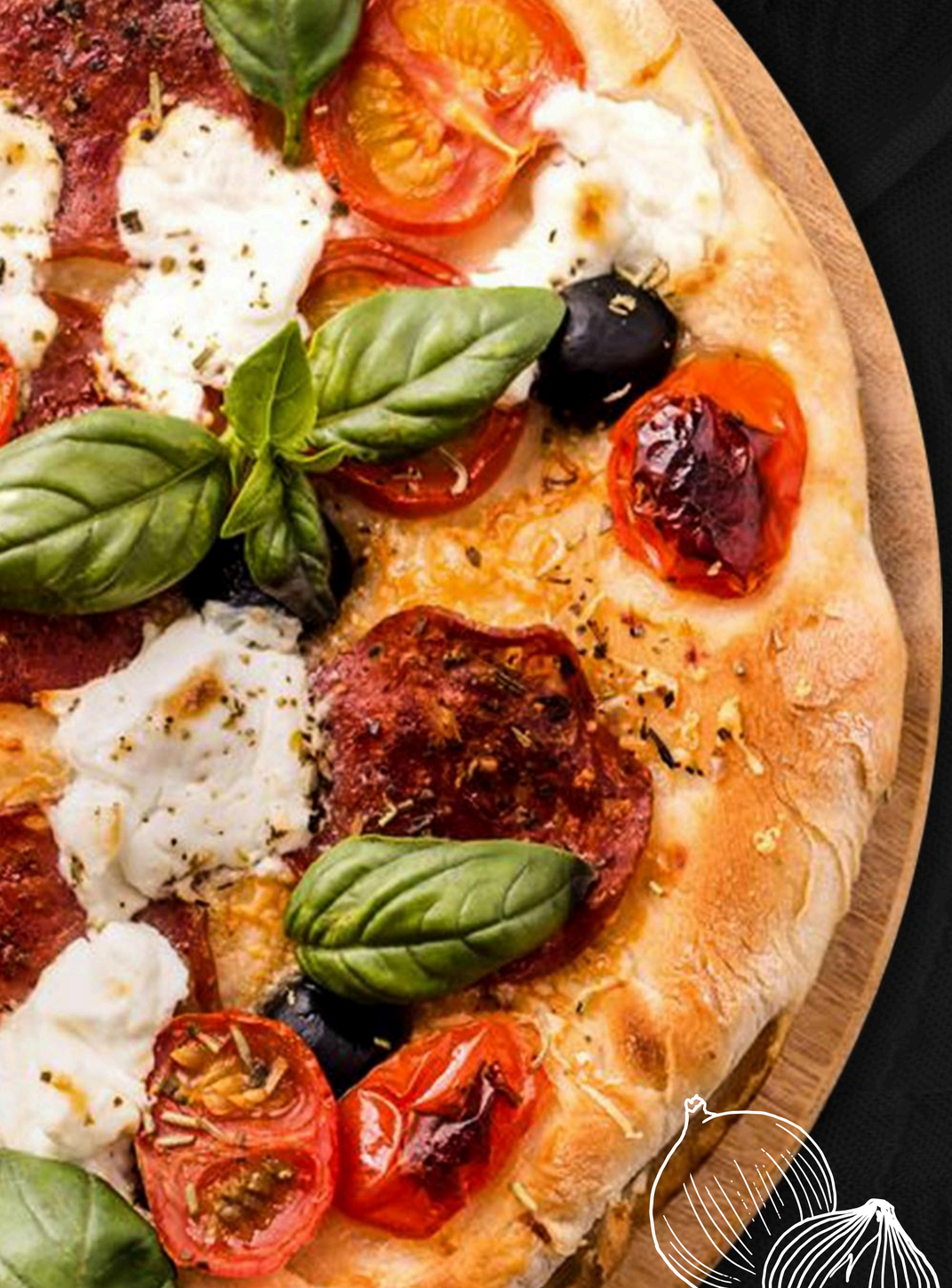


Q1.Retrieve the total number of orders placed.

```sql
SELECT
    COUNT(*) AS 'Total-Order'
FROM
    orders;
```

Result Grid | Filter Rows:

	Total-Order
▶	21350

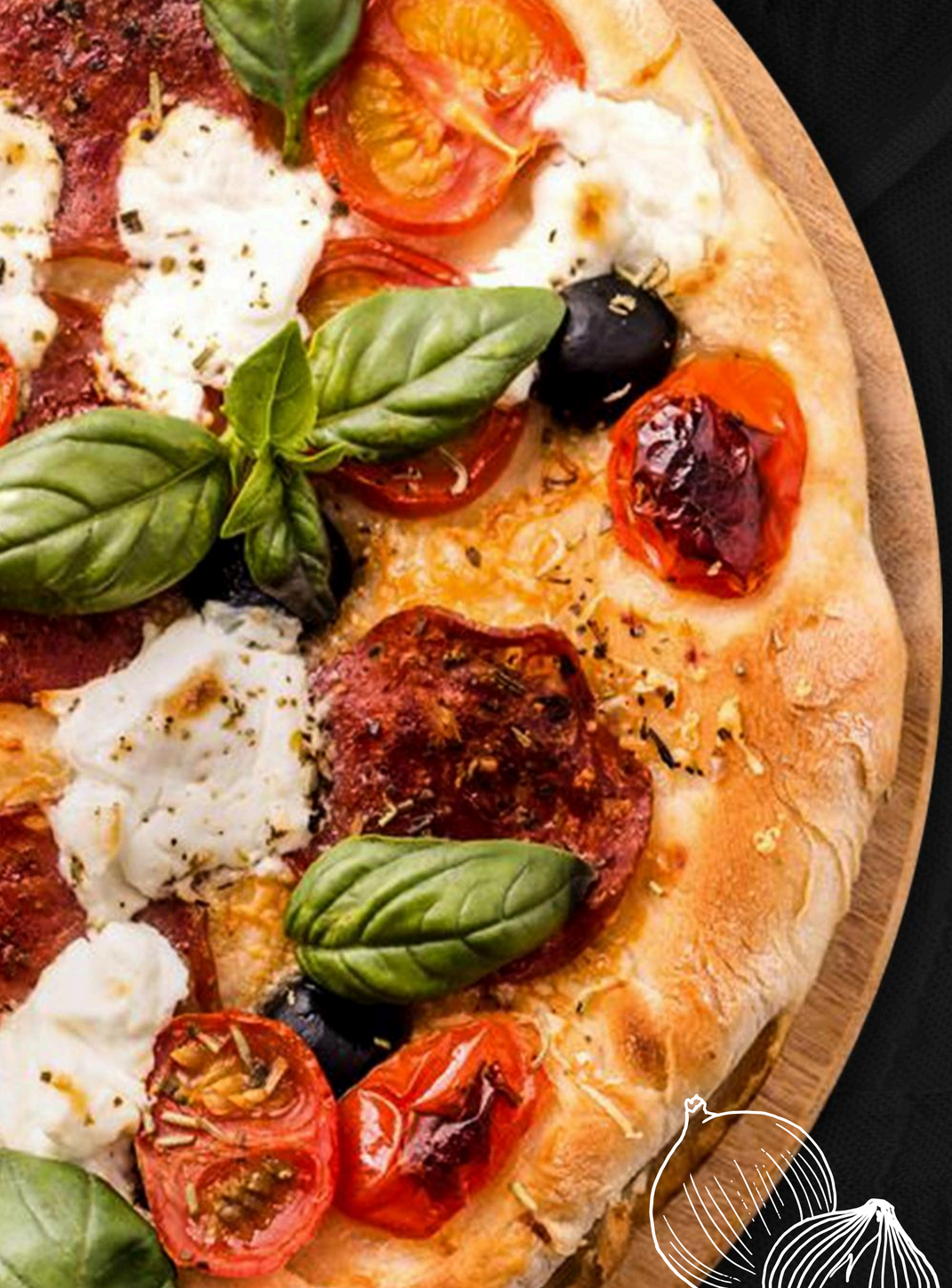


Q2. Calculate the total revenue generated from pizza sales.

```
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
```

Result Grid | Filter Rows:

	Total_Sales
▶	817860.05



Q3.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;
;
```

Result Grid | Filter Rows:

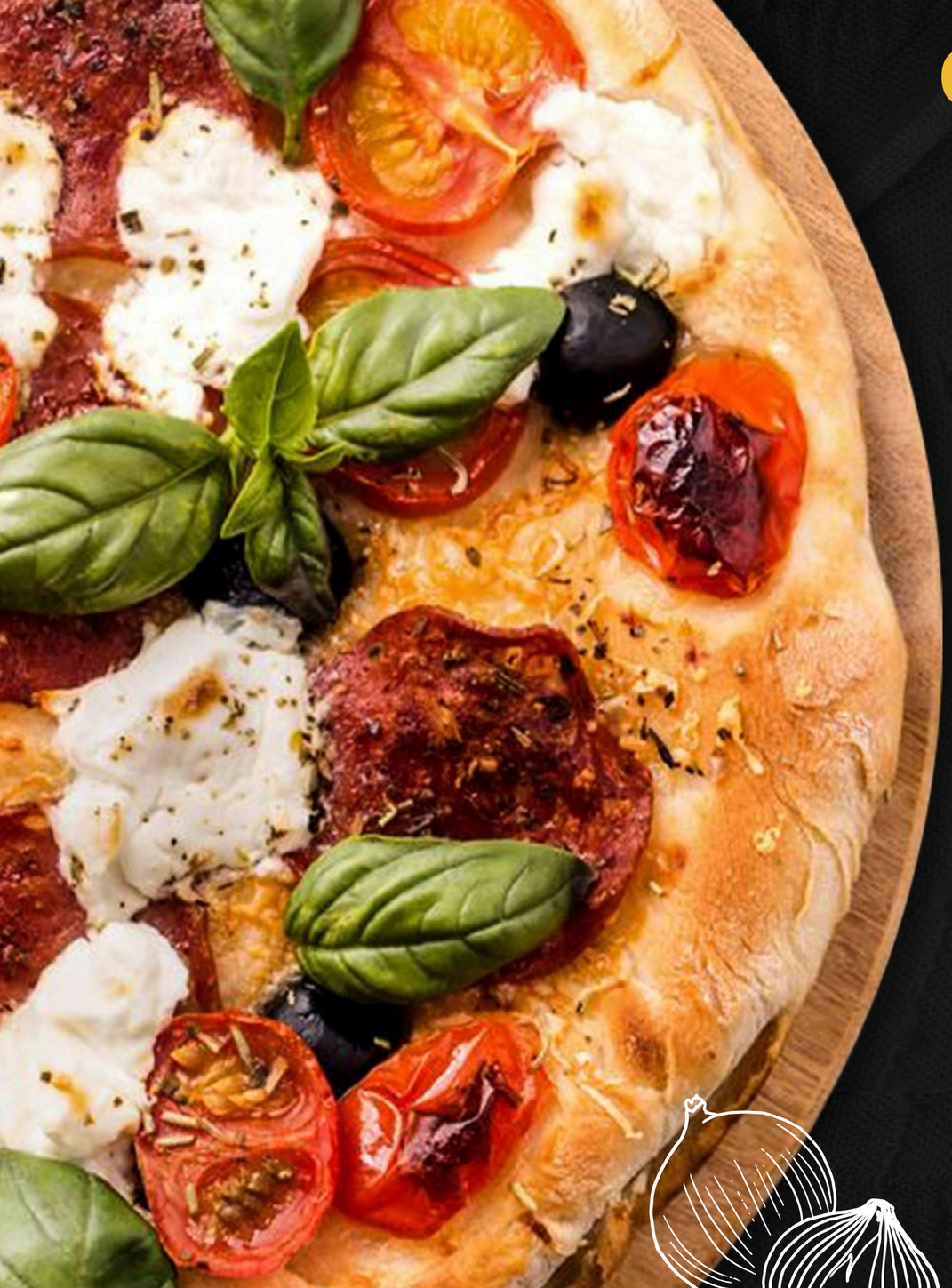
	name	price
▶	The Greek Pizza	35.95

Q4.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;
```

Result Grid | Filter Rows:

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

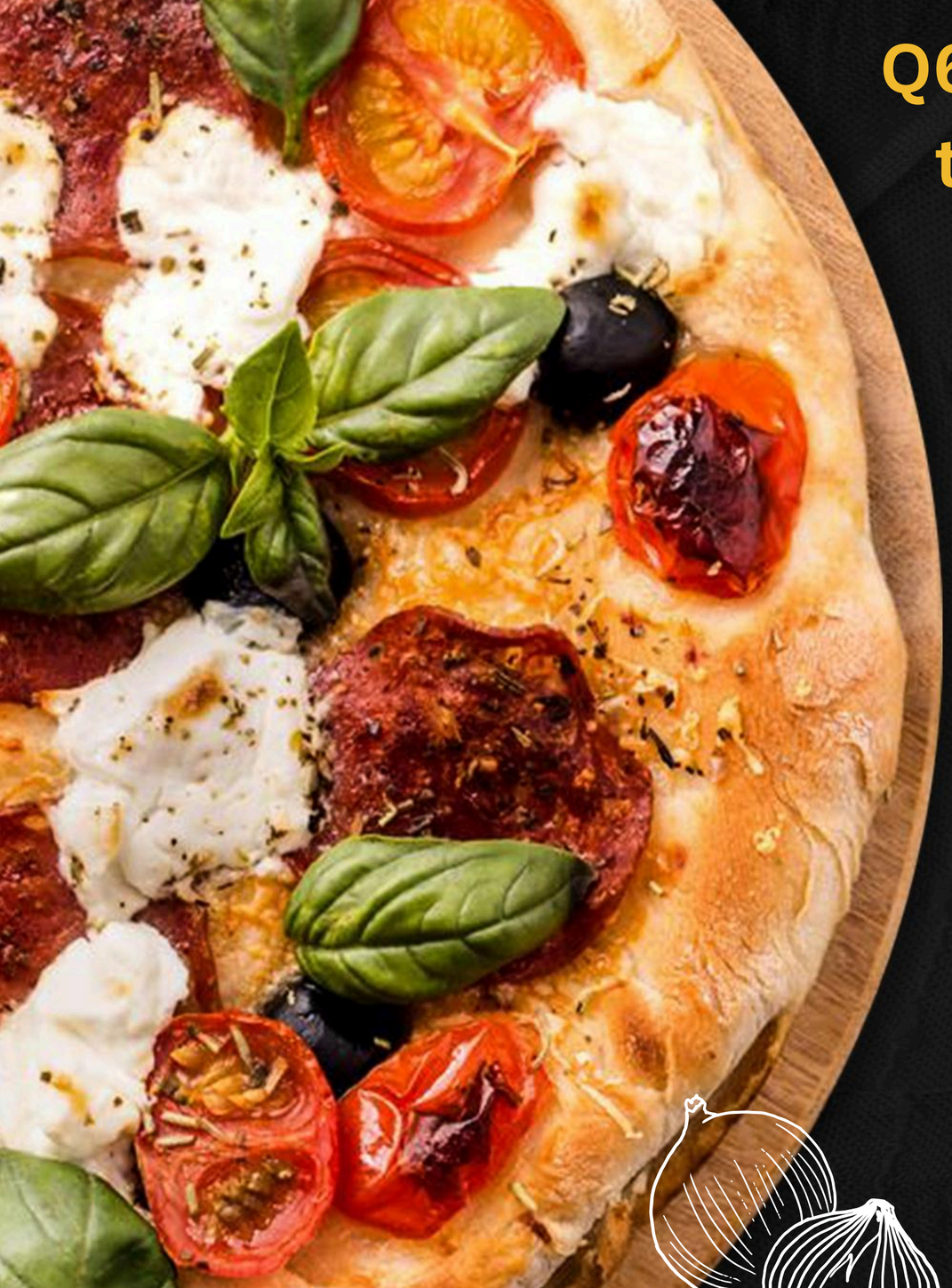


Q5.List the top 5 most ordered pizza types along with their quantities.

```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;
```

Result Grid | Filter Rows:

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

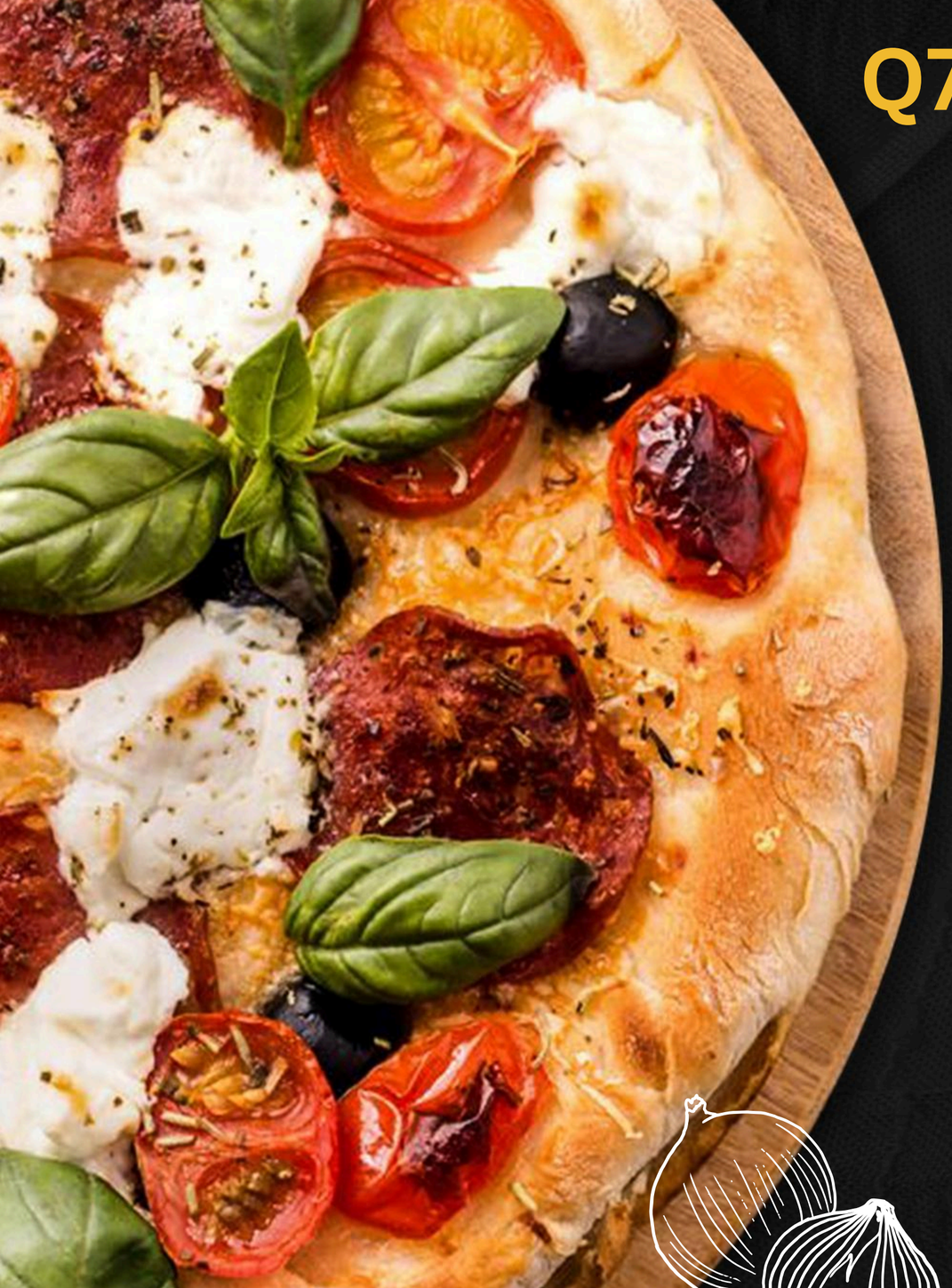


Q6.Join the necessary tables to find the total quantity of each pizza category ordered.

```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;
```

Result Grid | Filter R

category	Quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

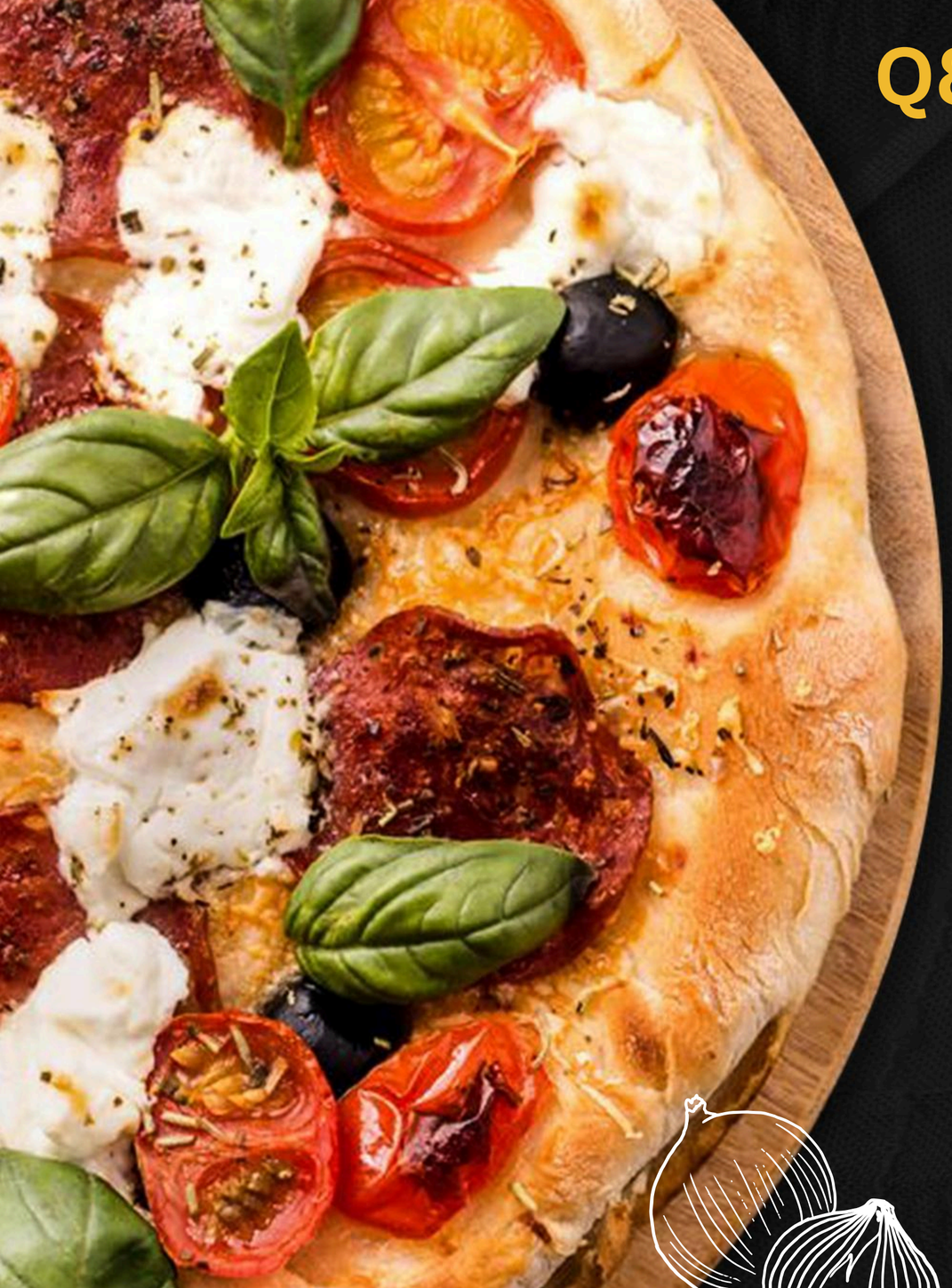


Q7.Determine the distribution of orders by hour of the day.

```sql
SELECT
    HOUR(Order_time) AS hour, COUNT(order_id) AS order_count
FROM
    orders
GROUP BY HOUR(order_time)
```

Result Grid | Filter Rows

hour	order_count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663
23	28
10	8
9	1

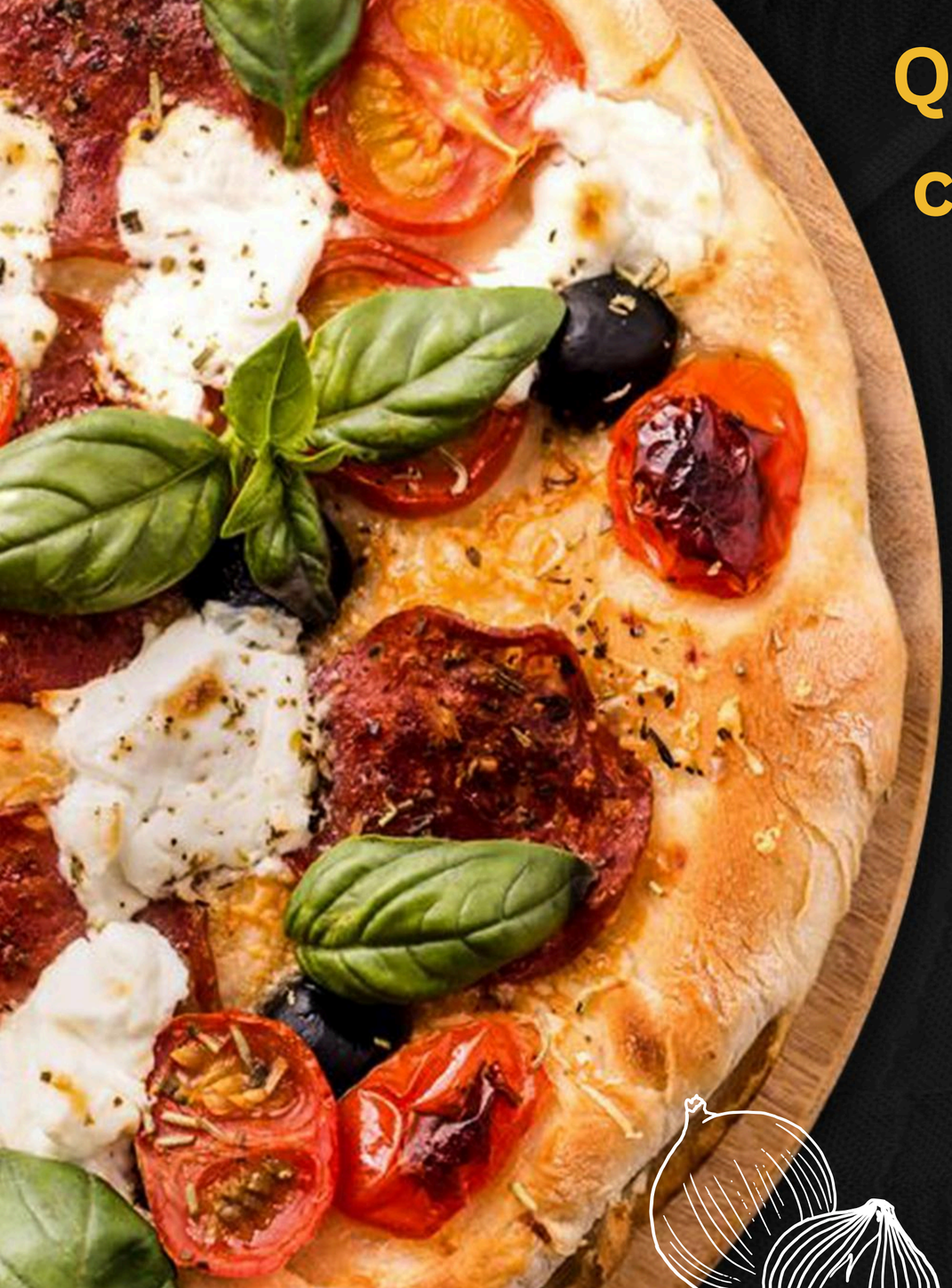


Q8.Join relevant tables to find the category-wise distribution of pizzas.

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

Result Grid | Filter Rows:

	category	count(name)
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

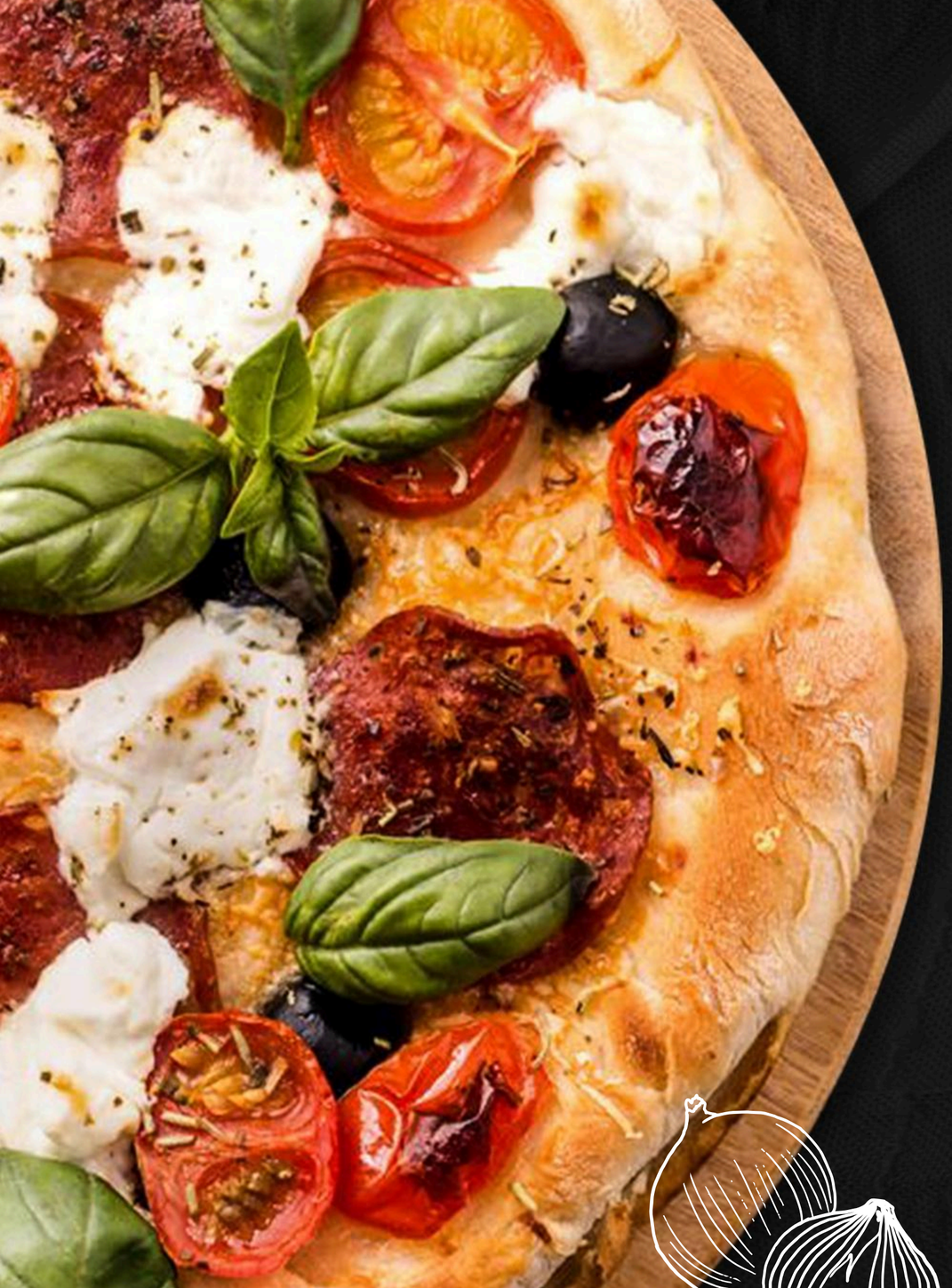


Q9. Group the orders by date and calculate the average number of pizzas ordered per day.

```sql
SELECT
    ROUND(AVG(quantity), 0) as 'Average-pizza-per-day'
FROM
    (SELECT
        orders.Order_date, SUM(order_details.Quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.Order_ID = order_details.Order_id
    GROUP BY orders.Order_date) AS order_quantity;
```

Result Grid | Filter Rows:

	Average-pizza-per-day
▶	138

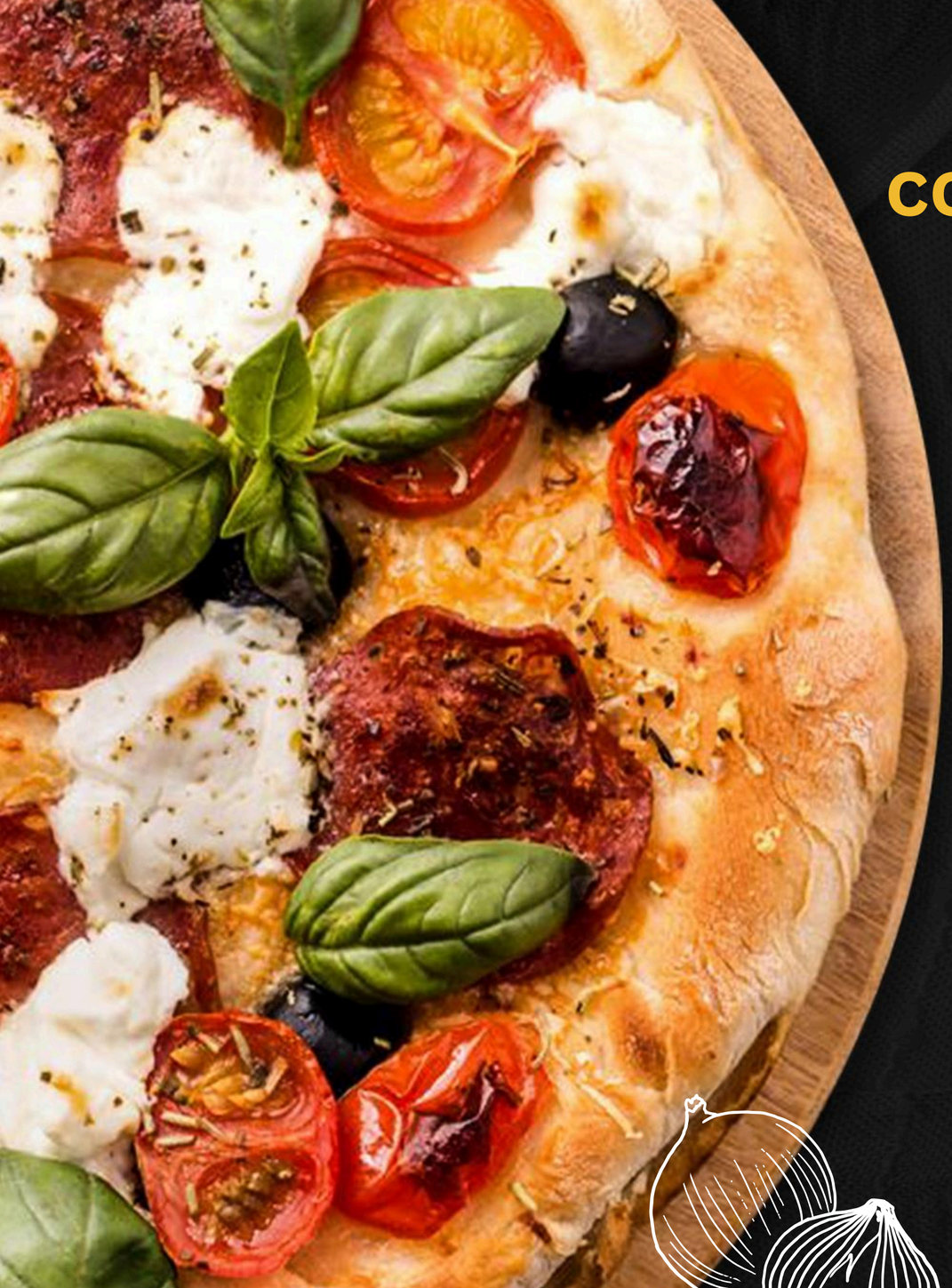


Q10.Determine the top 3 most ordered pizza types based on revenue.

```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
```

Result Grid | Filter Rows:

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

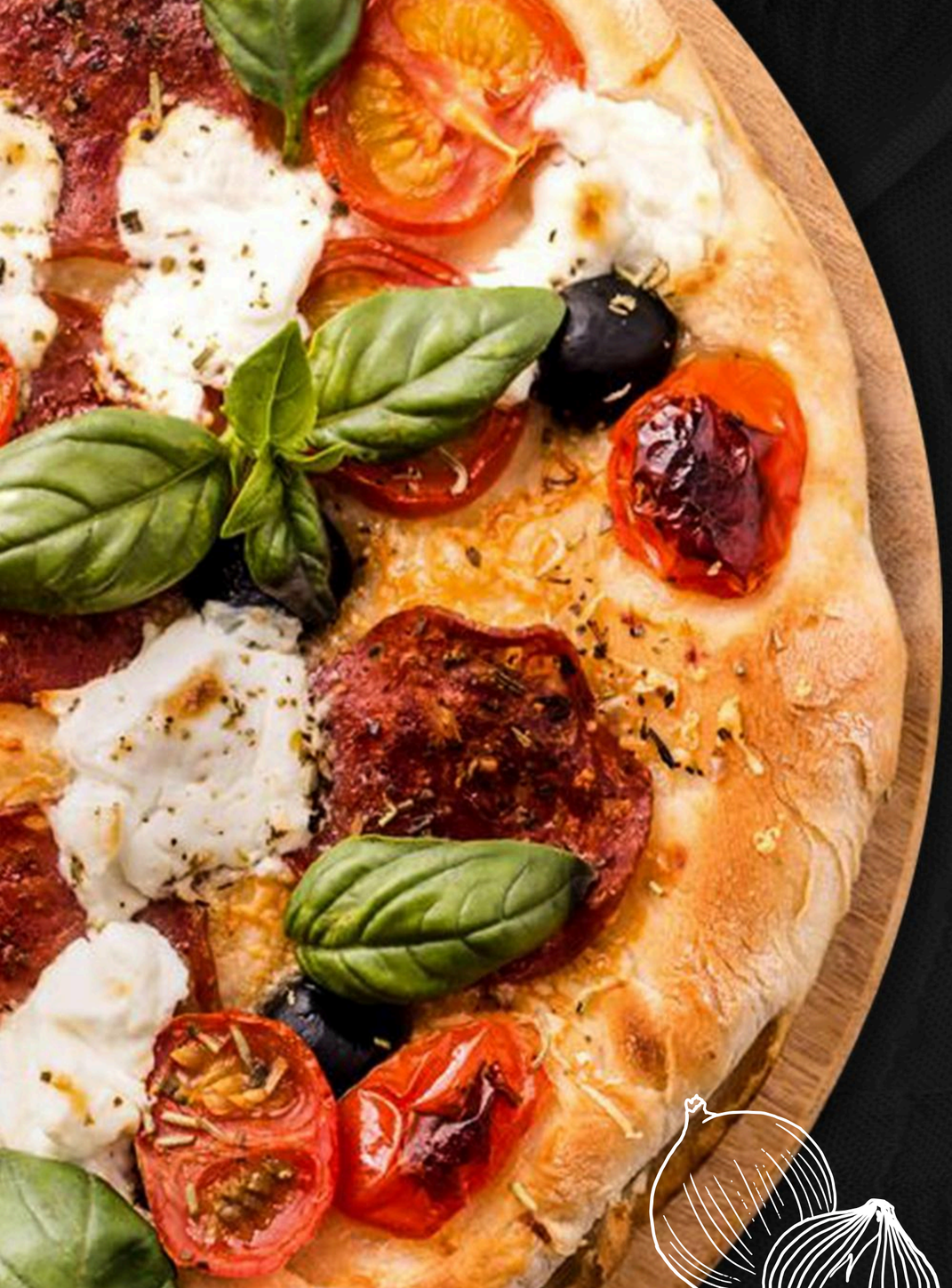


Q11.Calculate the percentage contribution of each pizza type to total revenue.

```sql
SELECT
    pizza_types.category,
    ROUND(SUM(order_details.Quantity * pizzas.price) / (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) * 100,
            2) 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.category
ORDER BY revenue DESC;
```

Result Grid | Filter Rows

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

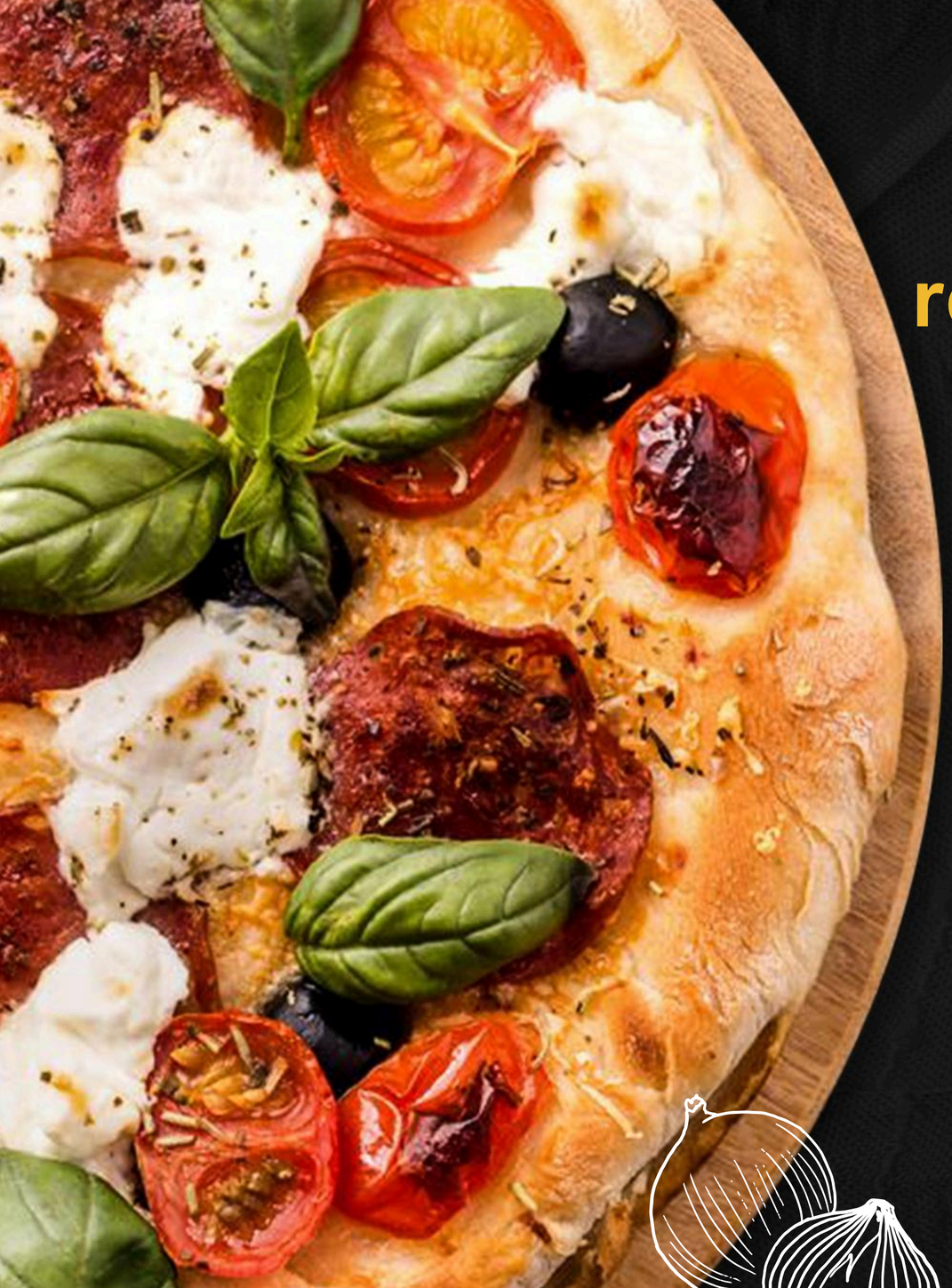


Q12. Analyze the cumulative revenue generated over time.

```sql
select Order_date,
sum(revenue) over(order by order_date) as cum_revenue
from
(select orders.Order_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.Order_date) as sales;
```

Result Grid | Filter Rows:

Order_date	cum_revenue
2015-01-01	2713.8500000000004
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55
2015-01-06	14358.5
2015-01-07	16560.7
2015-01-08	19399.05
2015-01-09	21526.4
2015-01-10	23990.350000000002
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.300000000003



Q13.Determine the top 3 most ordered pizza types 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;
```

Result Grid | Filter Rows: | Expo

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
The Pepperoni Pizza	30161.75
The Spicy Italian Pizza	34831.25
The Italian Supreme Pizza	33476.75
The Sicilian Pizza	30940.5
The Four Cheese Pizza	32265.70000000065
The Mexicana Pizza	26780.75
The Five Cheese Pizza	26066.5

PIZZA HUT

MEET OUR CHEF

NEIL TRAN

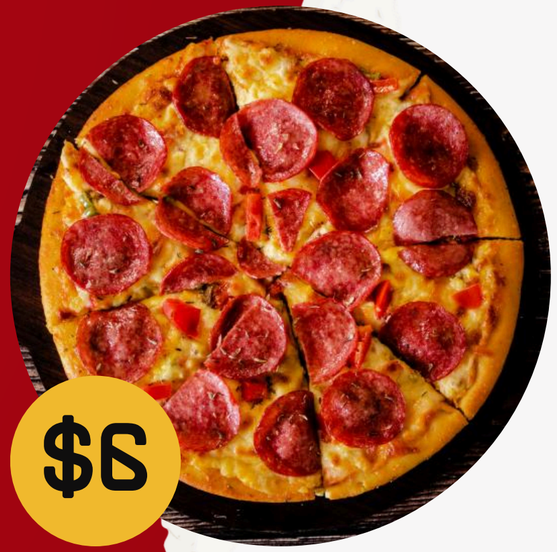
CHIDI EZE

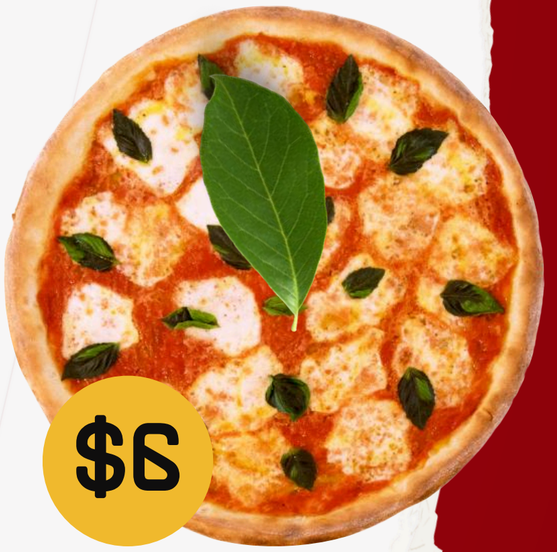
TEDDY YU

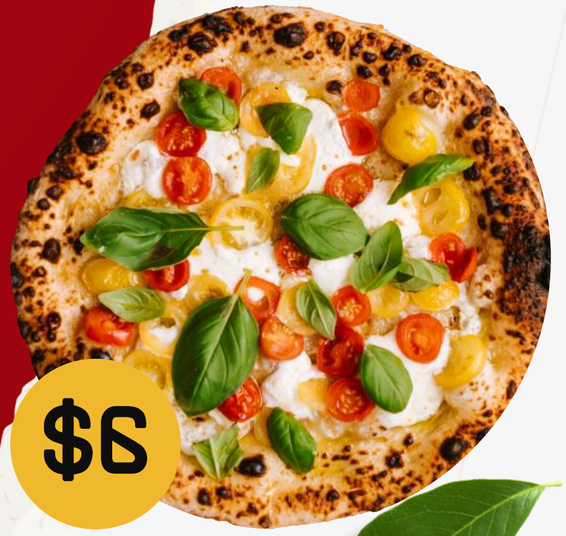
PIZZA HUT

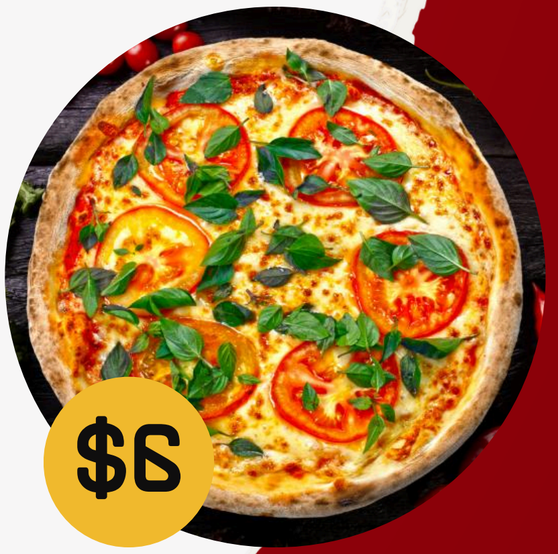
OUR MENU

Under various varities
1.classic
2.Supreme
3.veggies
4.chicken



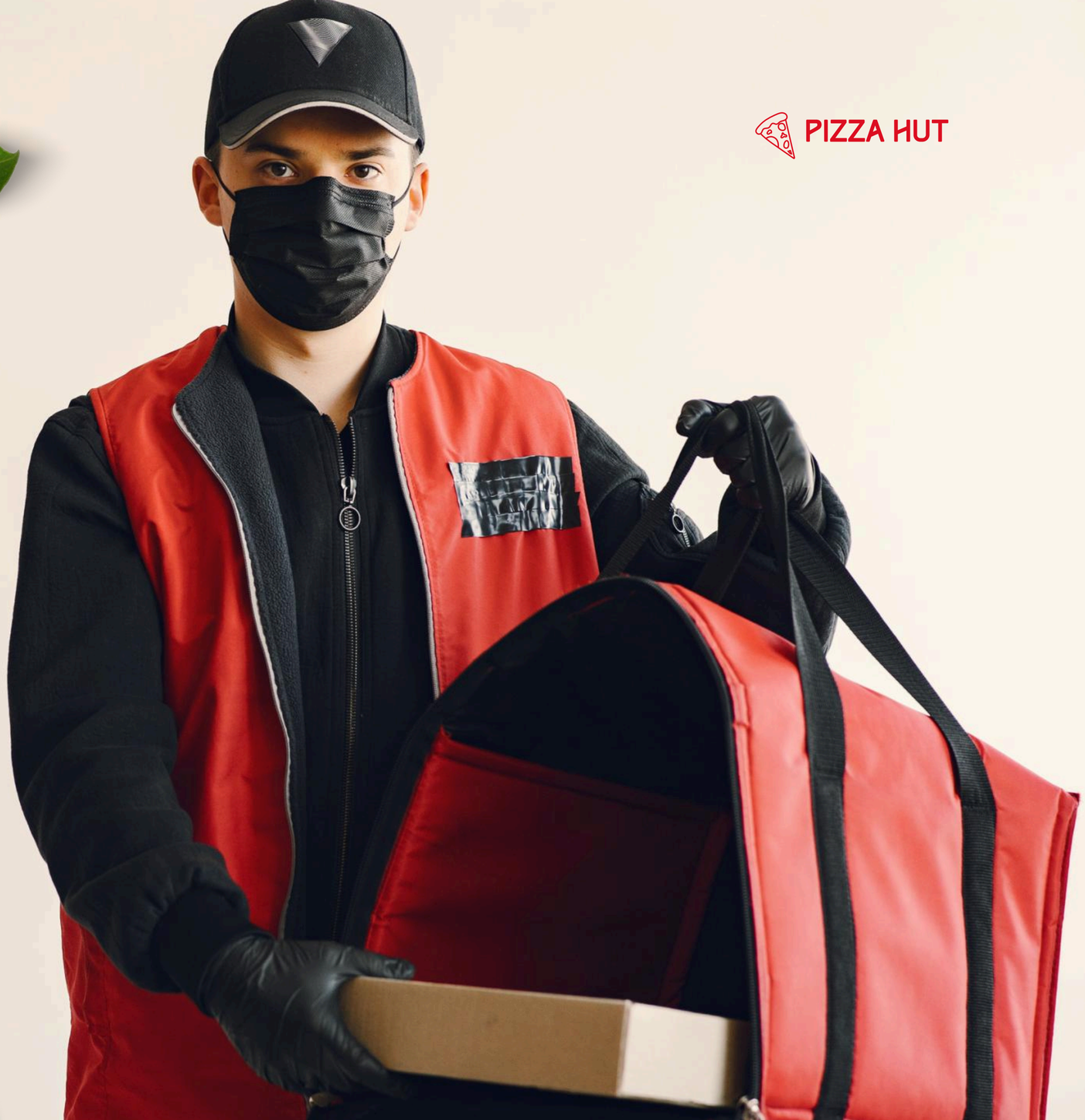


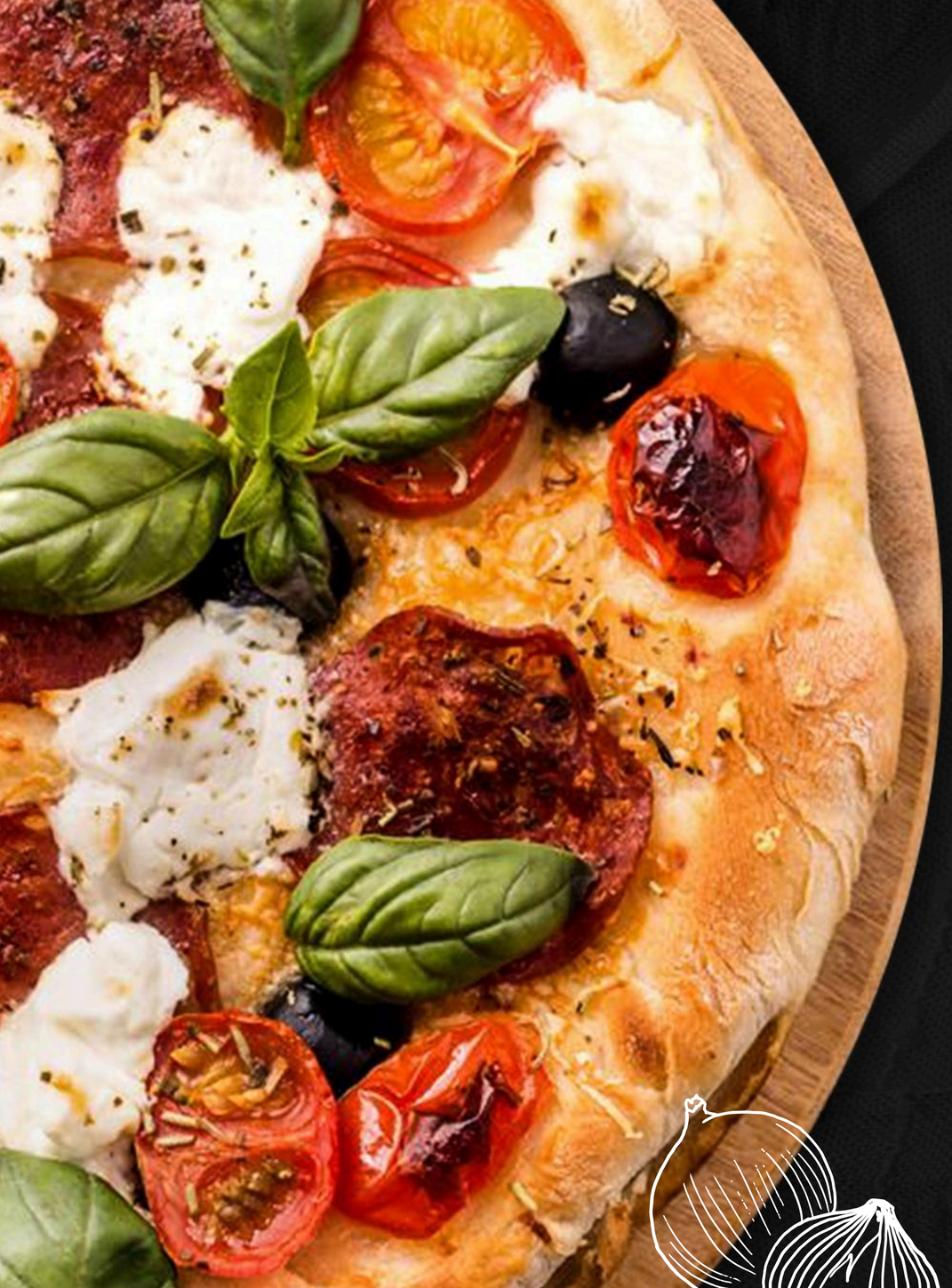




PIZZA HUT

FREE DELIVERY SERVICE





PIZZA HUT

OUR CONTACT

91-8459774885

amit.chavhan319@gmail.com

Maharashtra-India

PIZZA HUT

THANK YOU!