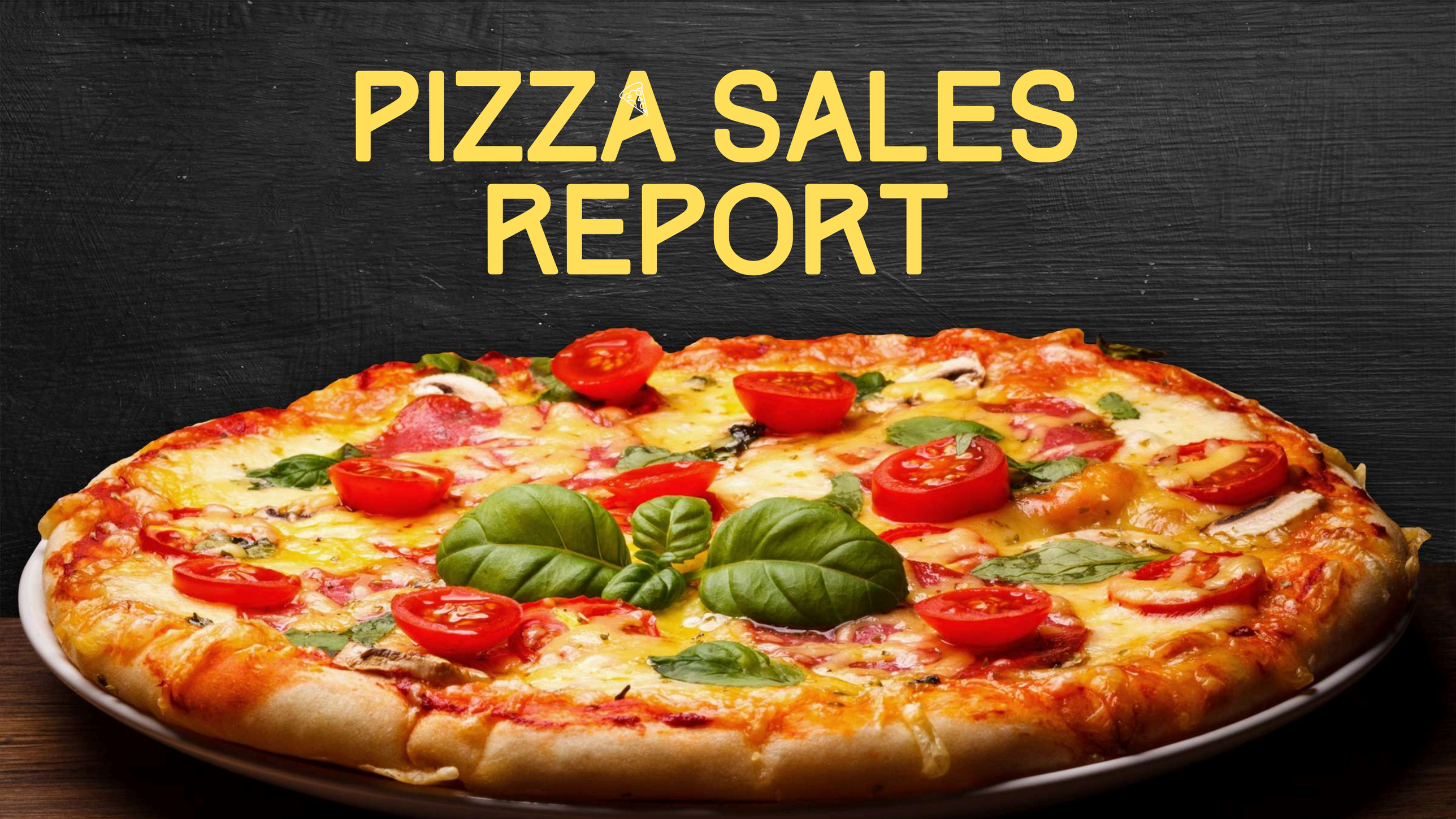
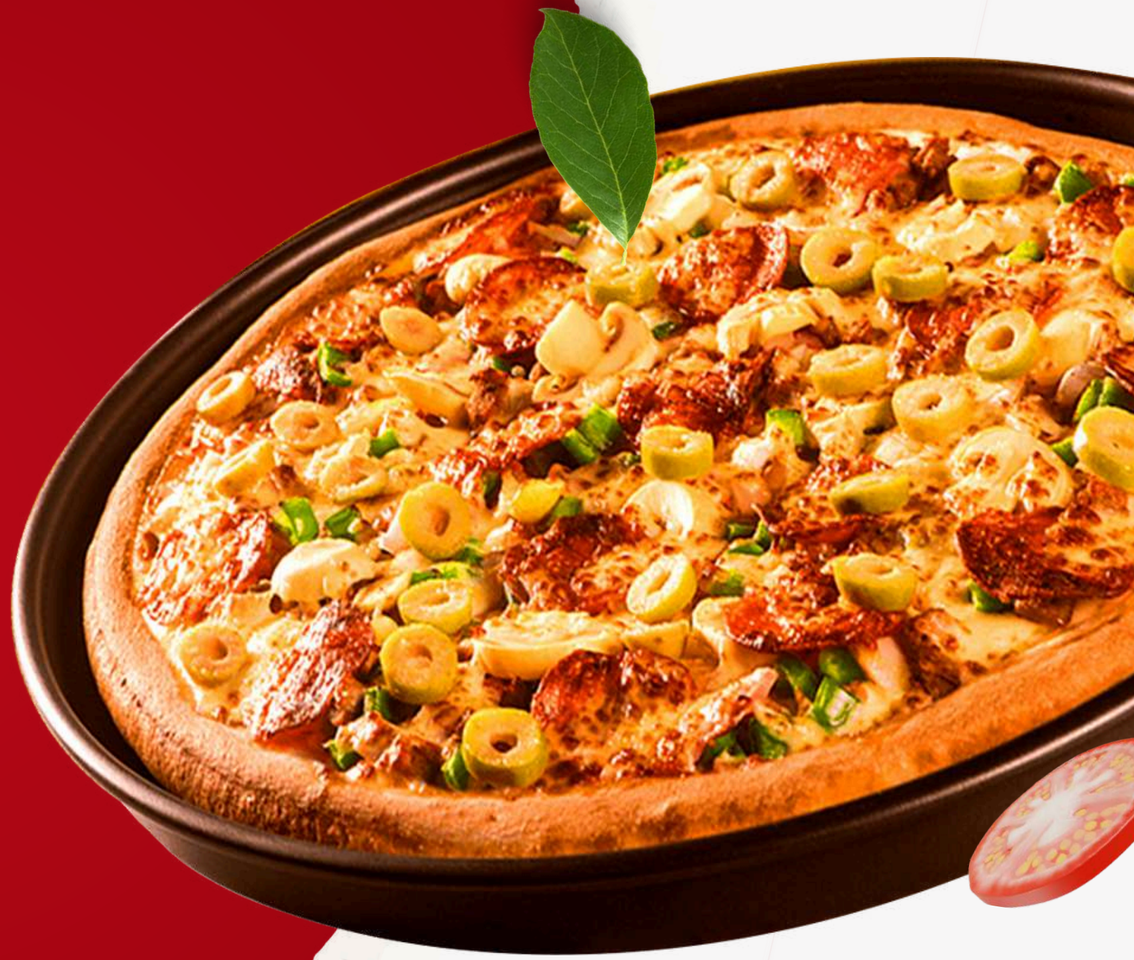
PIZZA SALES REPORT



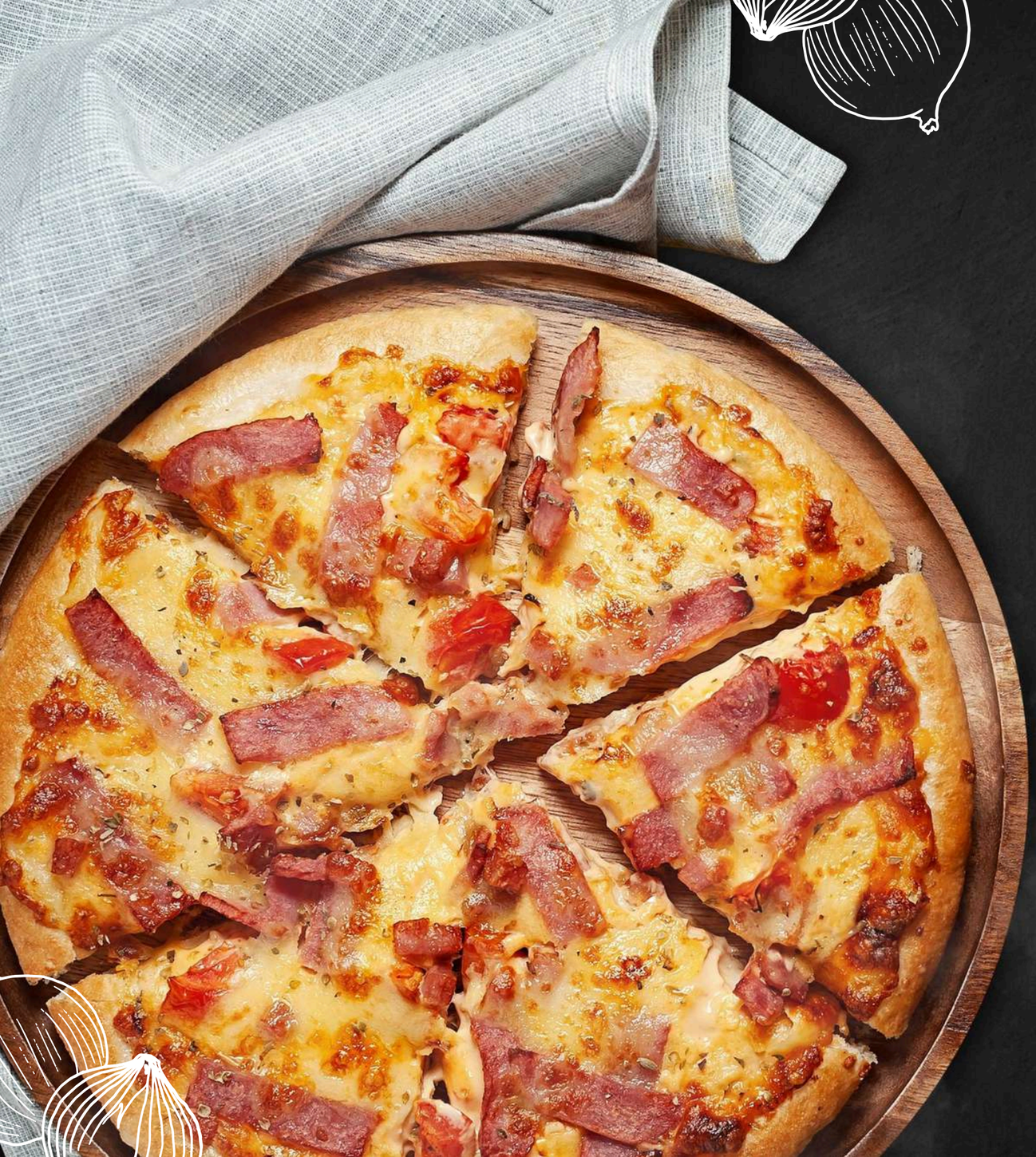
WELCOME

Hi, My name is Hina Verma.
In this project, I have utilized the pizza sales data to analyse the pizza sales using various SQL queries.



DATASETS USED IN ANALYSIS

DAILY ORDER SALES	ORDER AS PER DATE AND TIME	PIZZA TYPES WITH PRICE	PIZZA NAME WITH CATEGORY AND INGREDIENTS
ORDER DETAILS	ORDERS	PIZZAS	PIZZA TYPES



TABLES OVERVIEW

PIZZA_TYPES

pizza_type_id	name	category	ingredients
bbq_ckn	The Barbecue Chicken Pizza	Chicken	Barbecued Chicken, Red Peppers, Green Peppe...
cali_ckn	The California Chicken Pizza	Chicken	Chicken, Artichoke, Spinach, Garlic, Jalapeno P...

PIZZAS

pizza_id	pizza_type_id	size	price
bbq_ckn_s	bbq_ckn	S	12.75
bbq_ckn_m	bbq_ckn	M	16.75

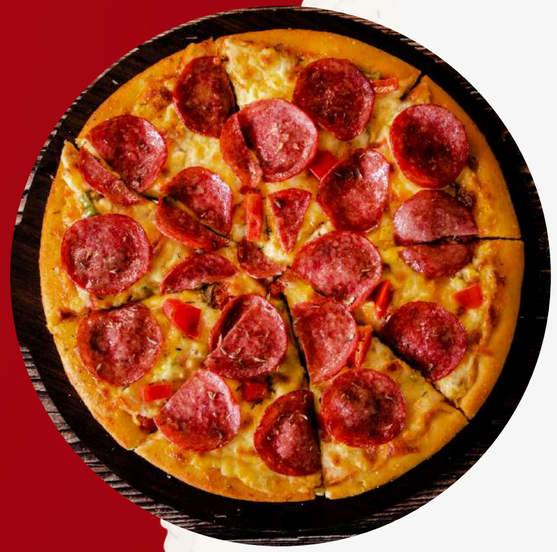
ORDERS

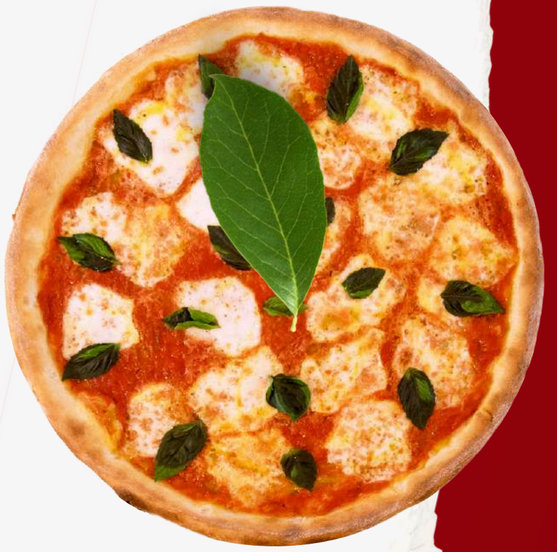
order_id	order_date	order_time
1	2015-01-01	11:38:36
2	2015-01-01	11:57:40

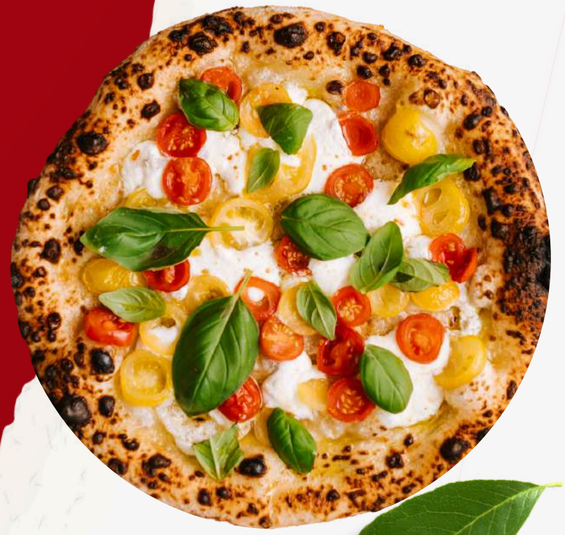
ORDER_DETAILS

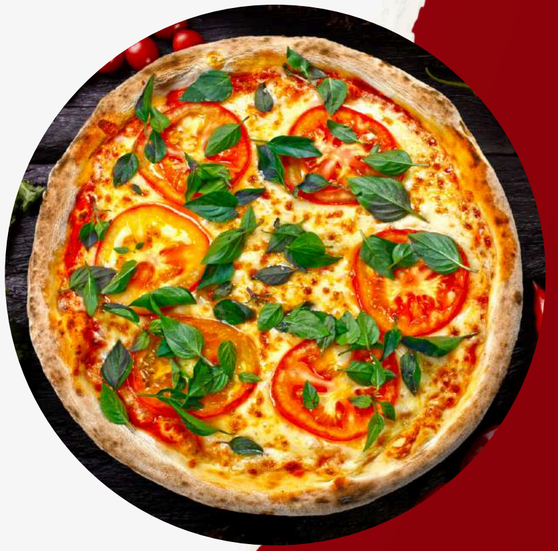
order_id	order_details_id	pizza_id	quantity
4381	2010-01-01	spicy_ital_l	1
4382	2010-01-02	ckn_alfredo_m	1

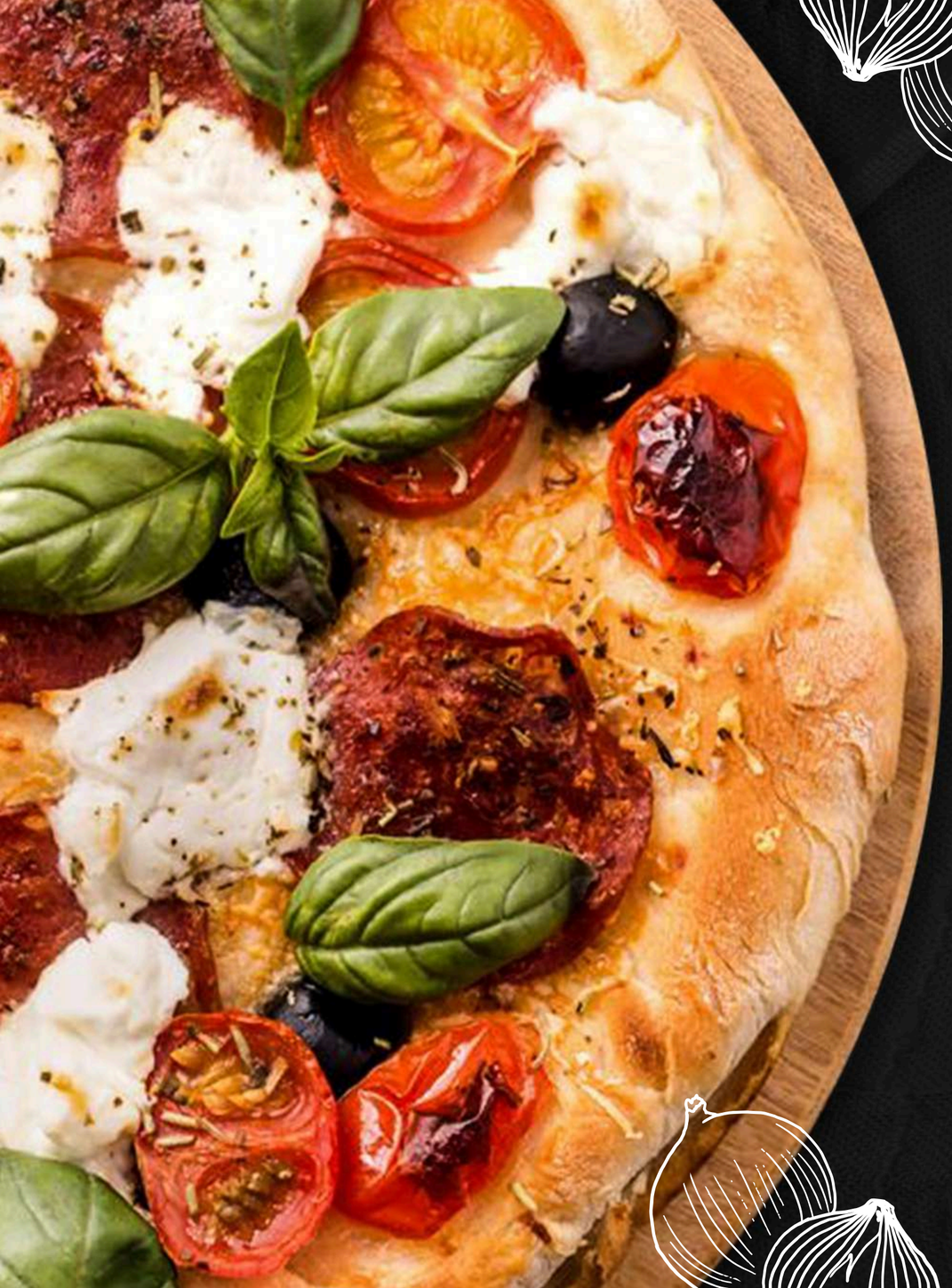
LET'S START ANALYSING









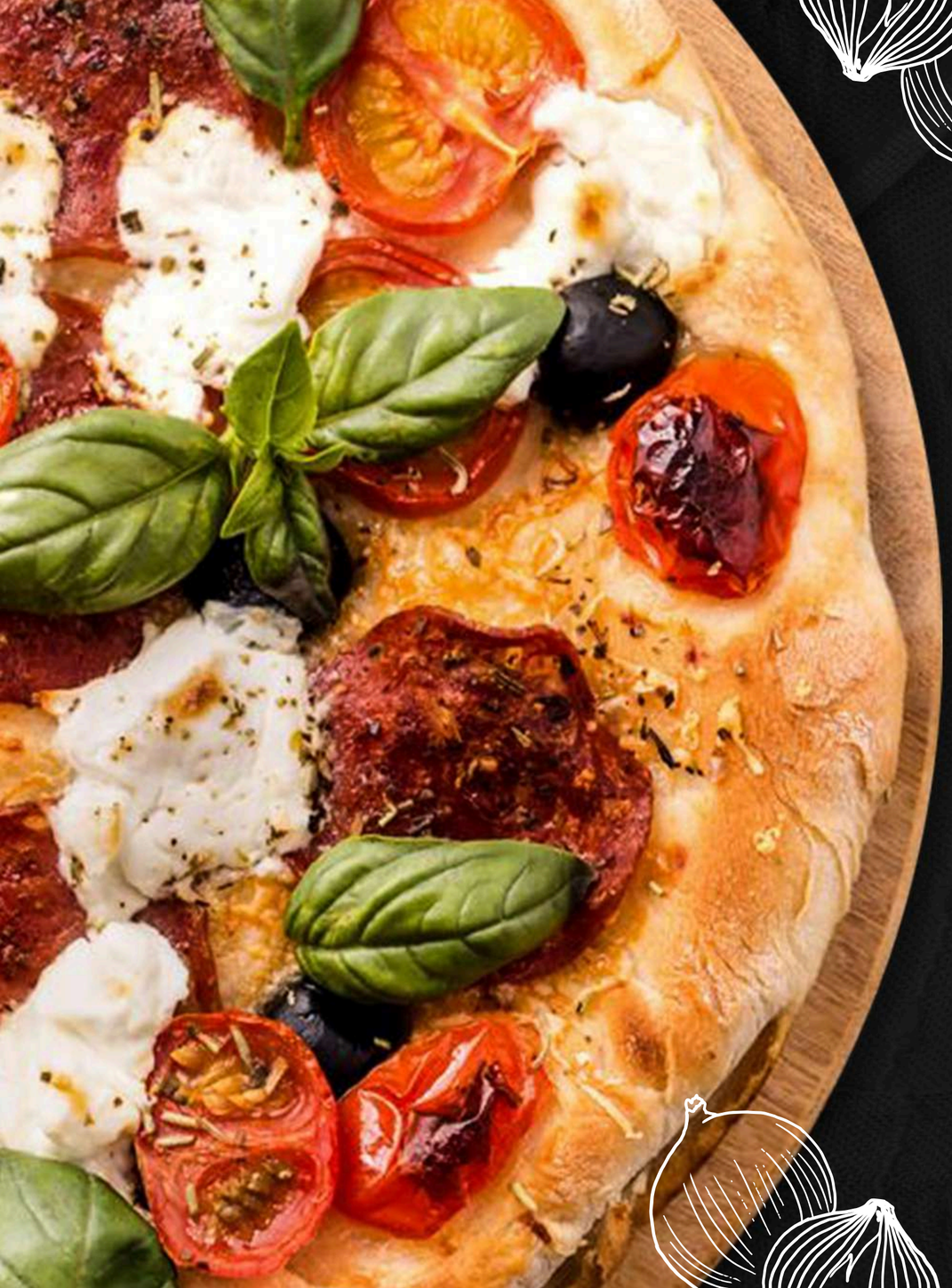


CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```
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

	total_sales
▶	34313.75

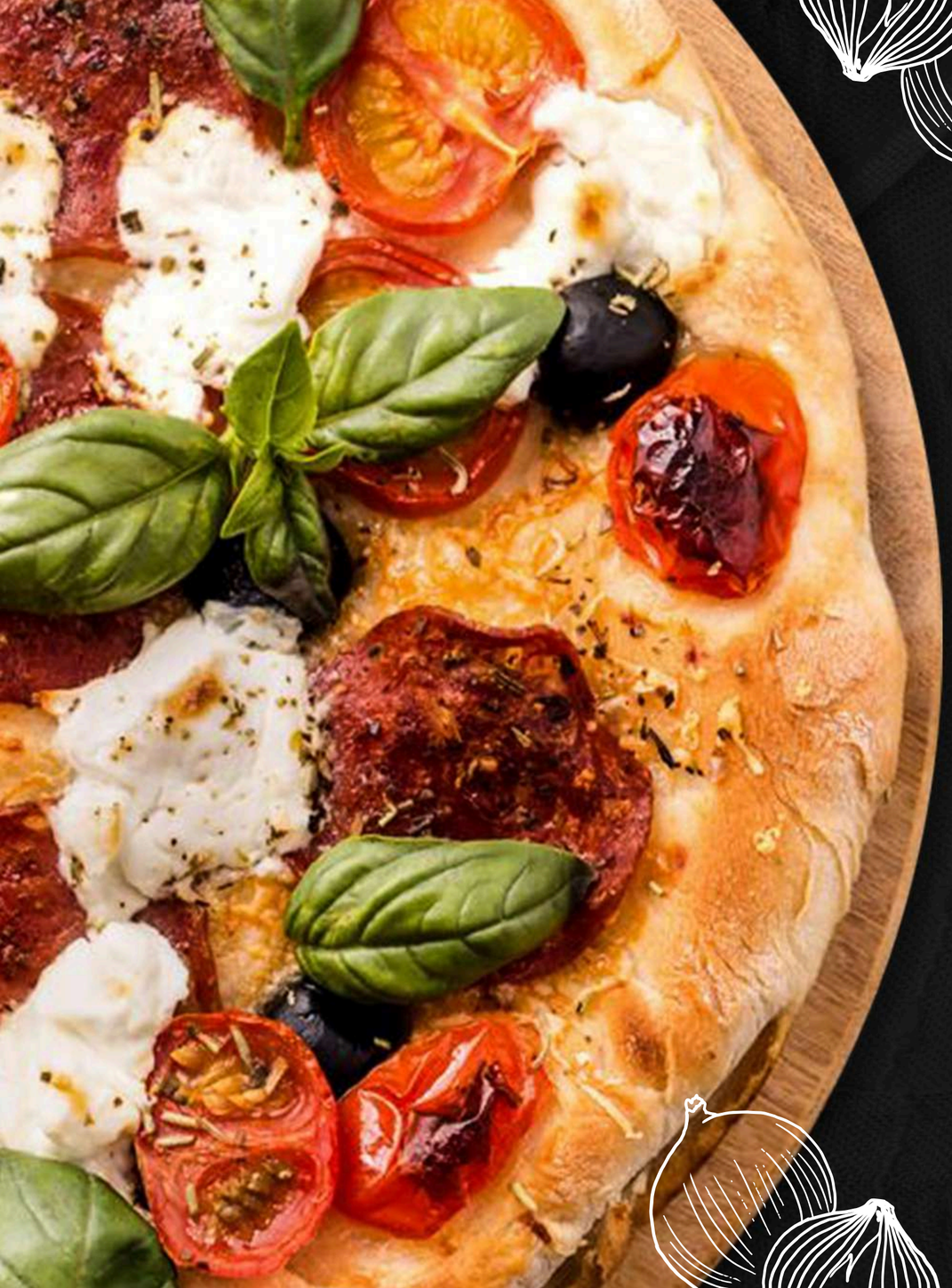


IDENTIFY THE HIGHEST-PRICED PIZZA.

```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

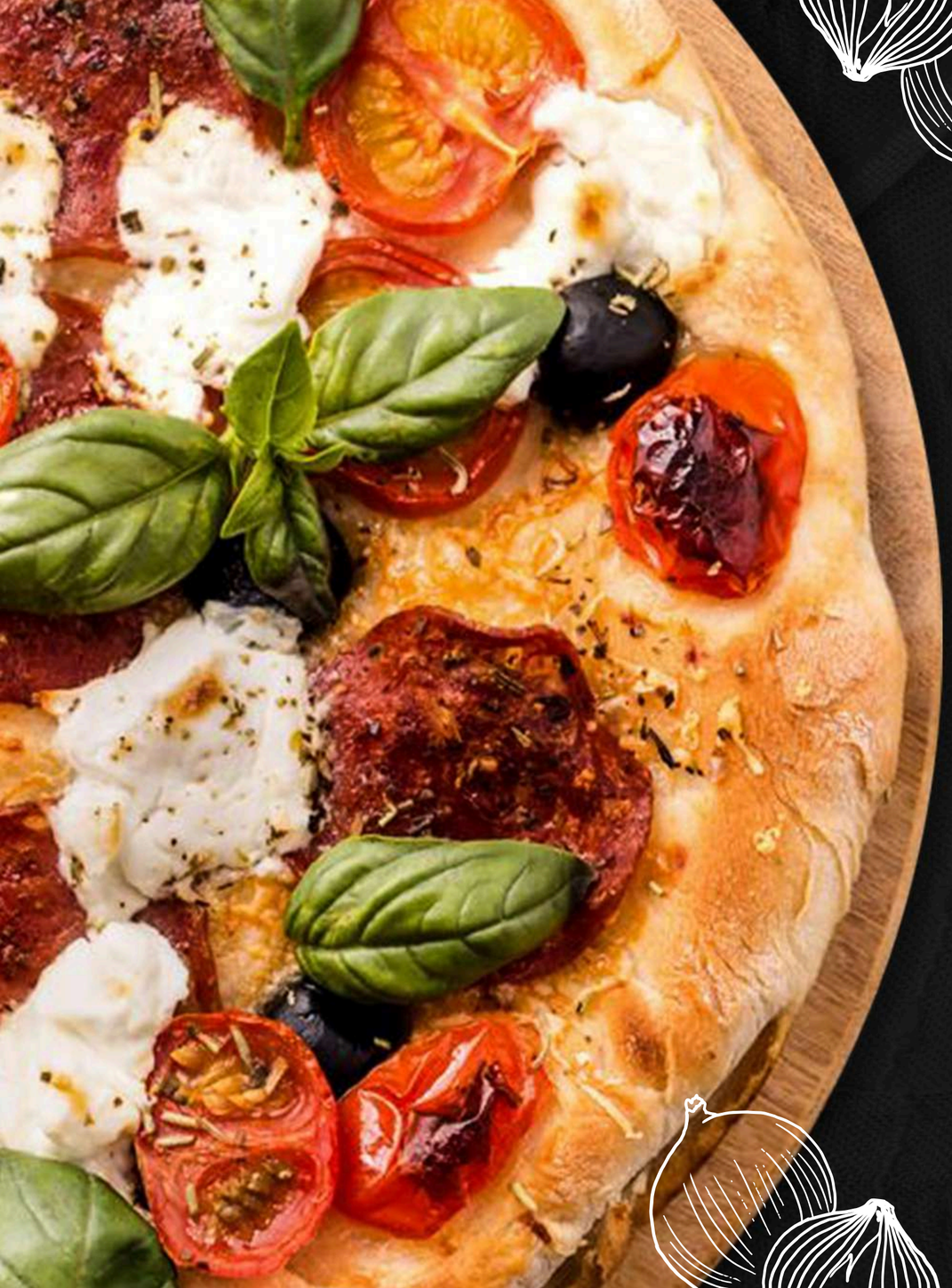


IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```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
LIMIT 1;
```

Result Grid | Filter Rows:

size	order_count
L	769

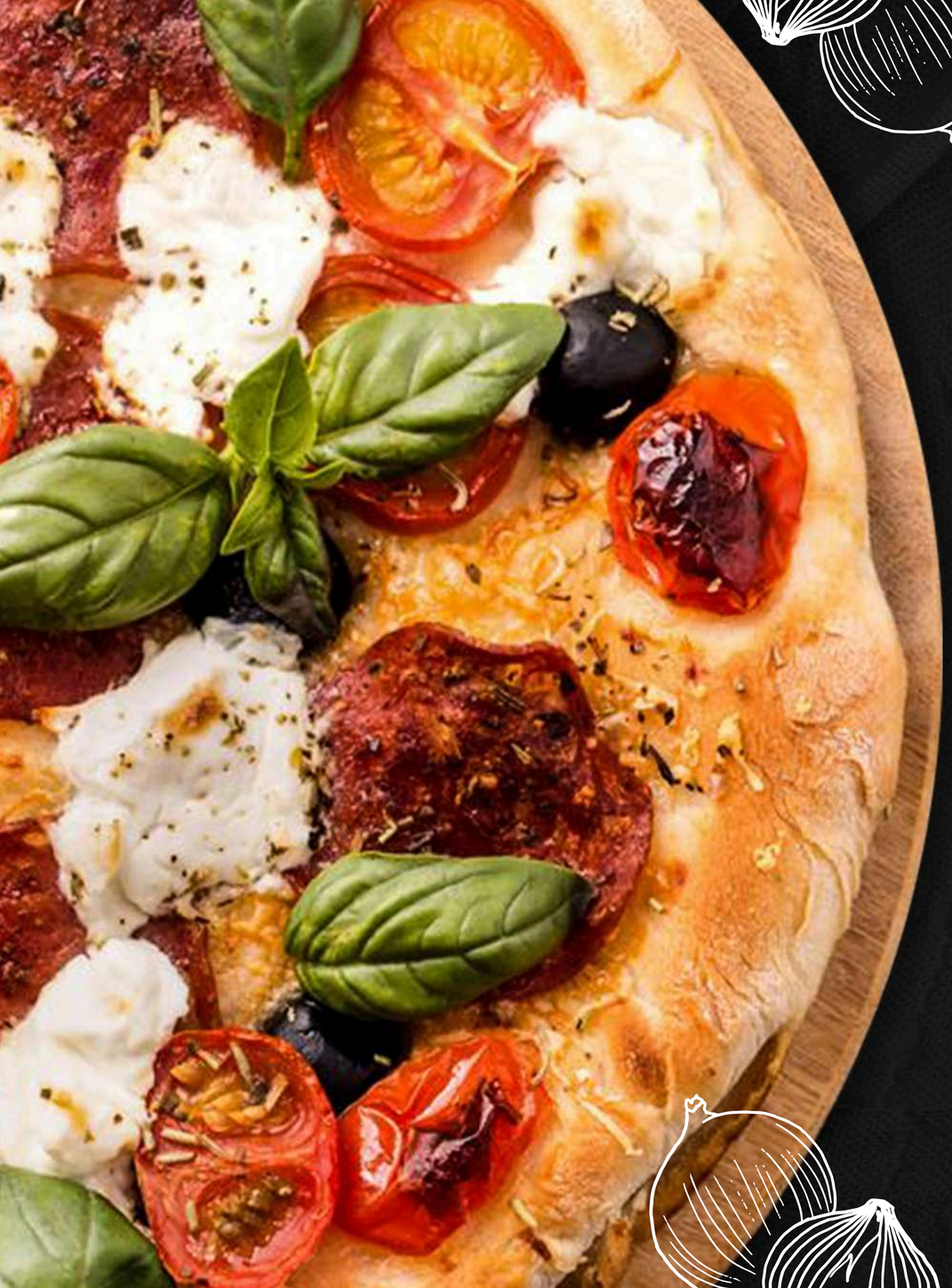


TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

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

Result Grid | Filter Rows:

name	Total_Quantity
The Barbecue Chicken Pizza	185
The California Chicken Pizza	146
The Classic Deluxe Pizza	145
The Big Meat Pizza	129
The Hawaiian Pizza	107

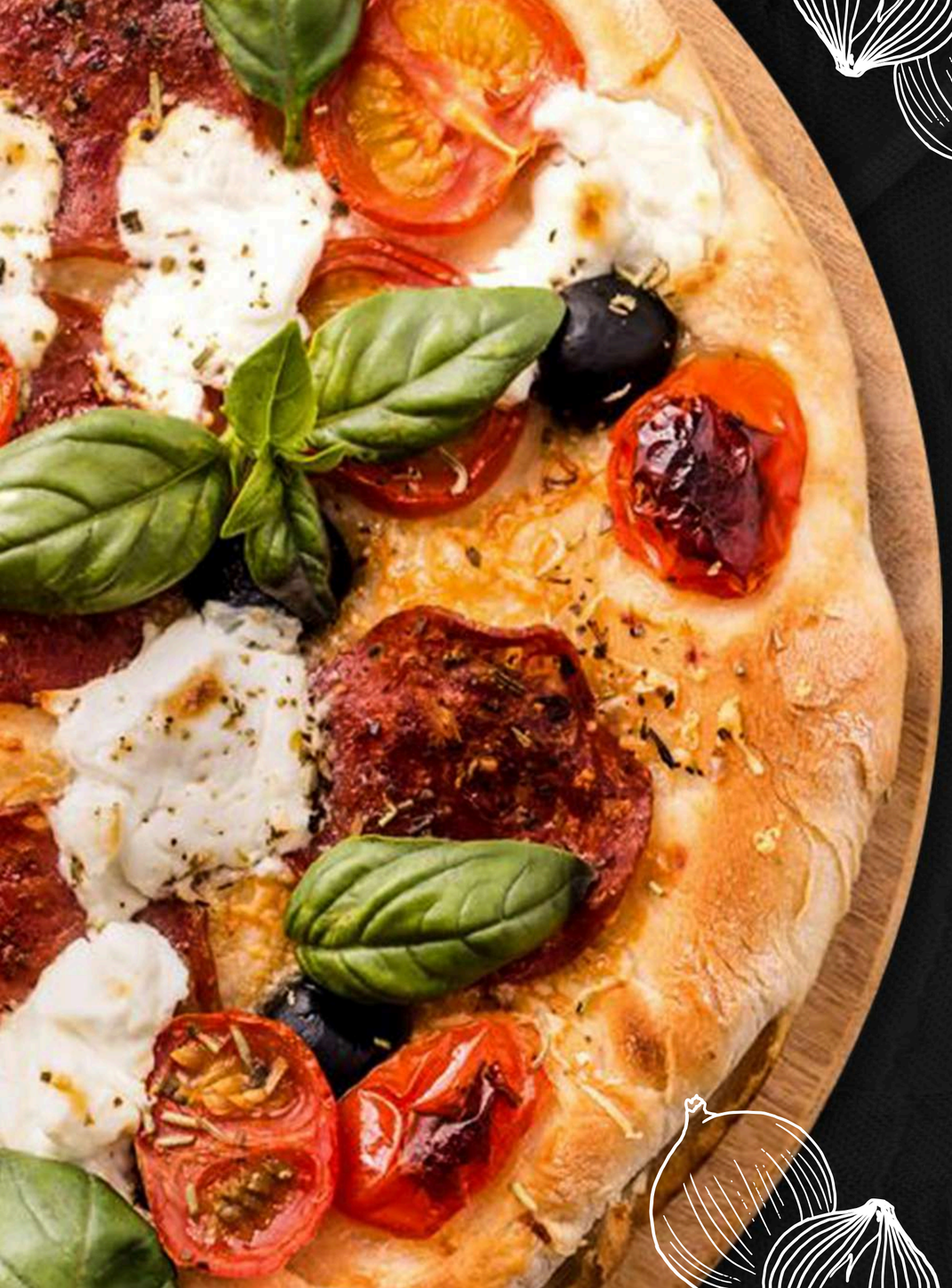


TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.

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

Result Grid | Filter Rows:

name	Total_Quantity
The Barbecue Chicken Pizza	185
The California Chicken Pizza	146
The Classic Deluxe Pizza	145
The Big Meat Pizza	129
The Hawaiian Pizza	107
The Four Cheese Pizza	102
The Chicken Alfredo Pizza	78

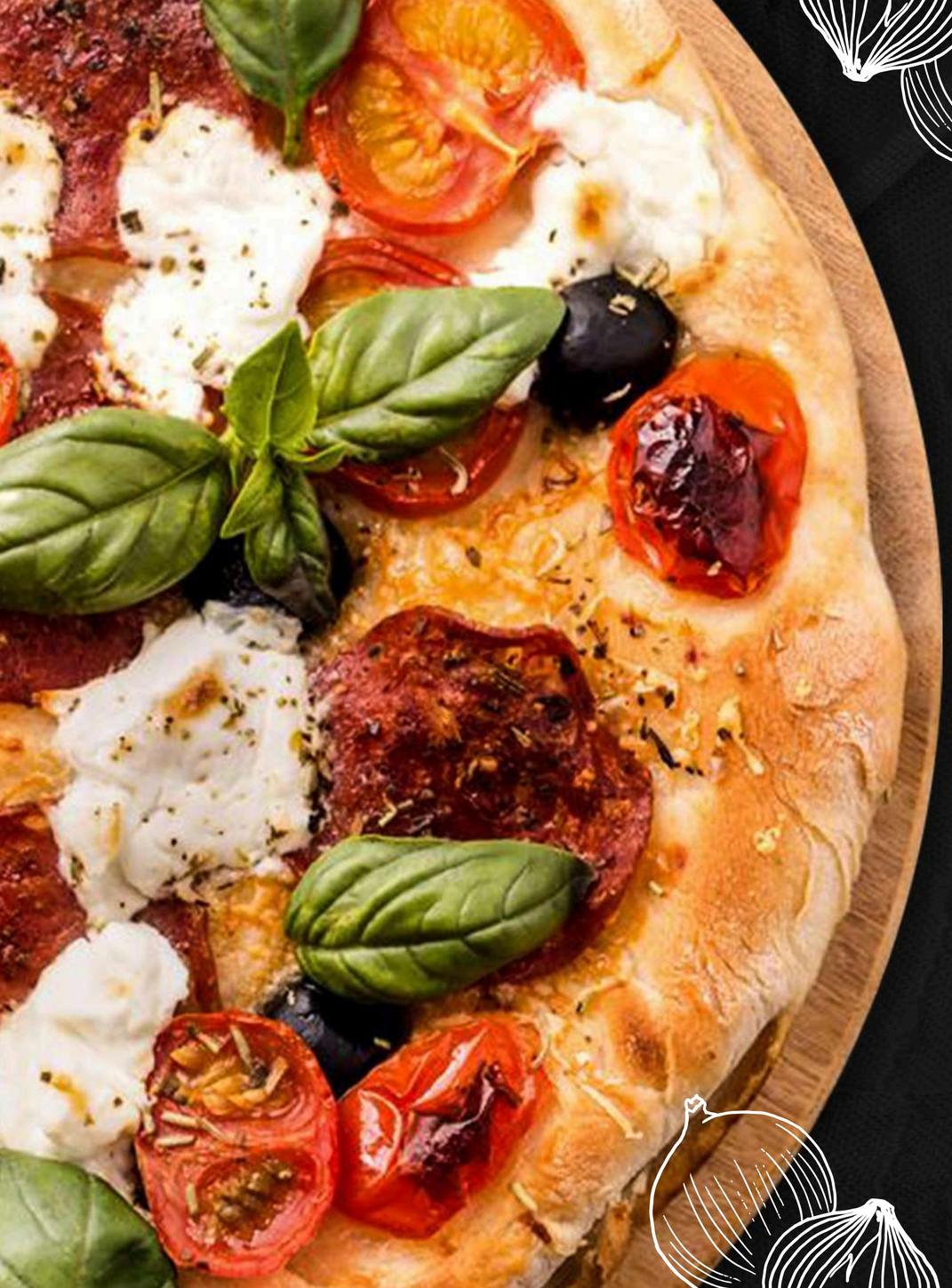


DISTRIBUTION OF ORDERS BY HOUR OF THE DAY

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

Result Grid | Filter Rows:

order_by_hour	order_count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336

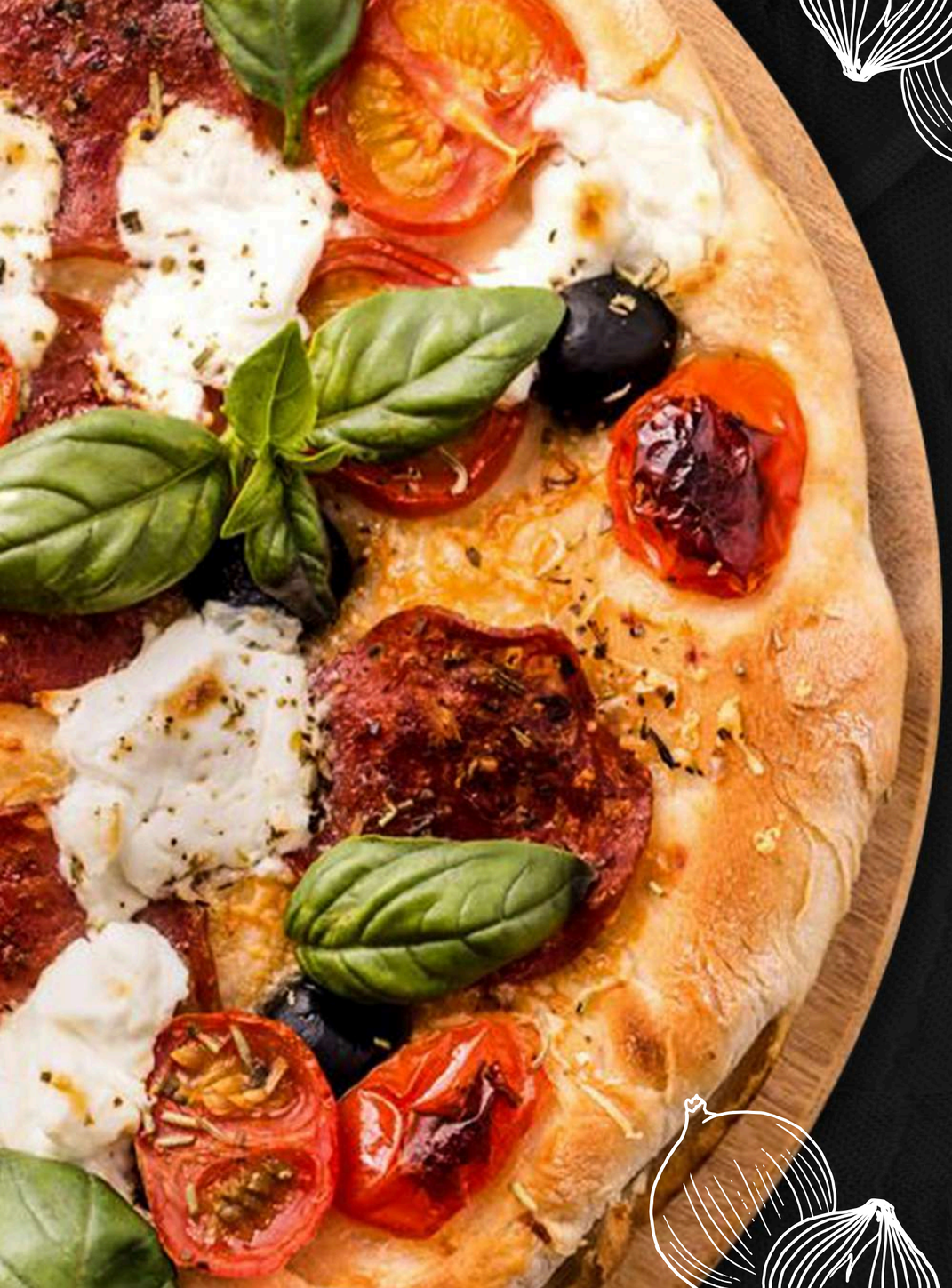


CATEGORY-WISE DISTRIBUTION OF PIZZAS

```
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

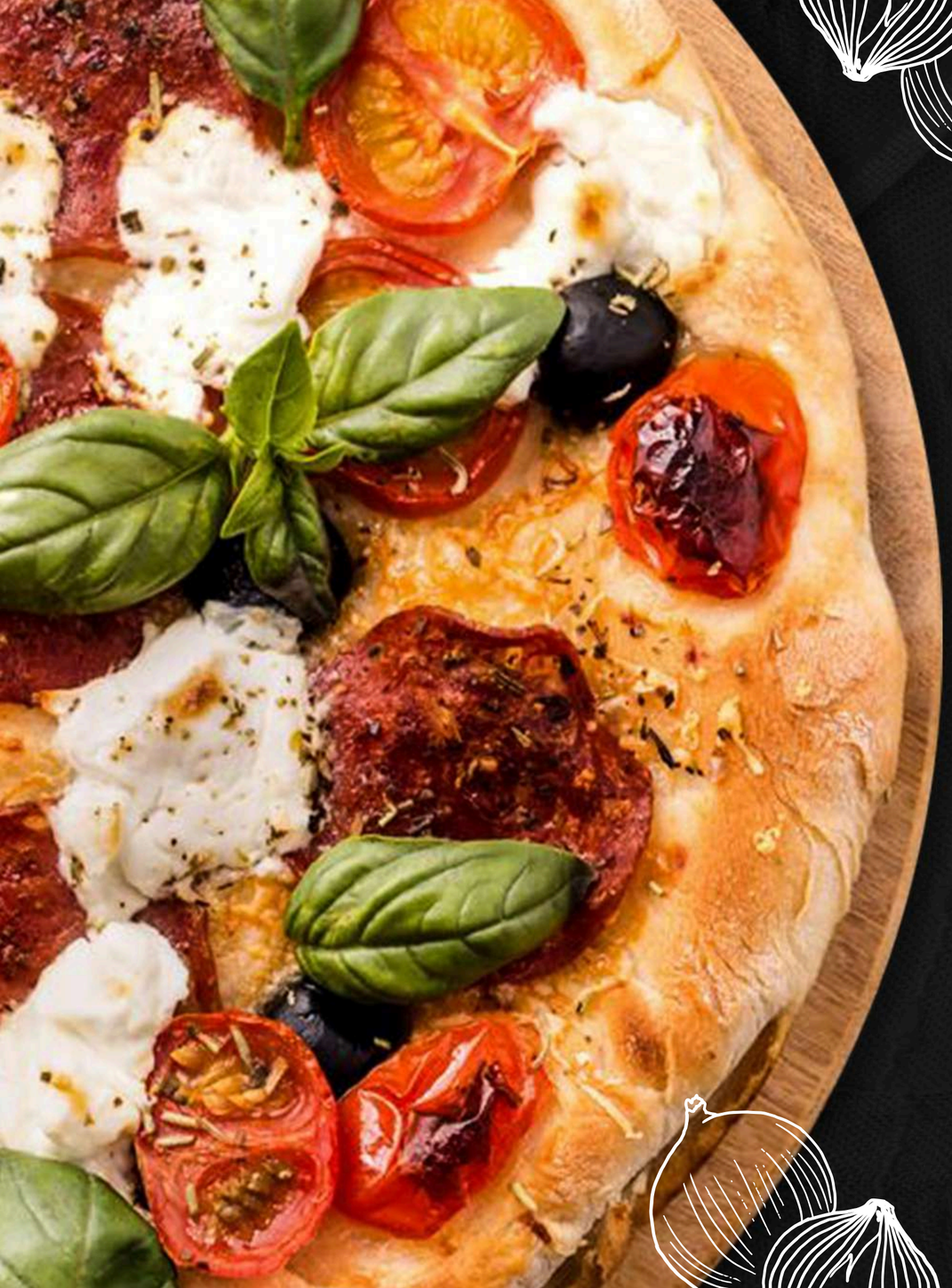


AVERAGE NUMBER OF PIZZAS ORDERED PER DAY (ORDERS BY DATE)

```
SELECT
    ROUND(AVG(order_placed),0) as Avg_pizza_per_day
FROM
    (SELECT
        orders.order_date,
            SUM(order_details.quantity) AS order_placed
    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:

Avg_pizza_per_day
26

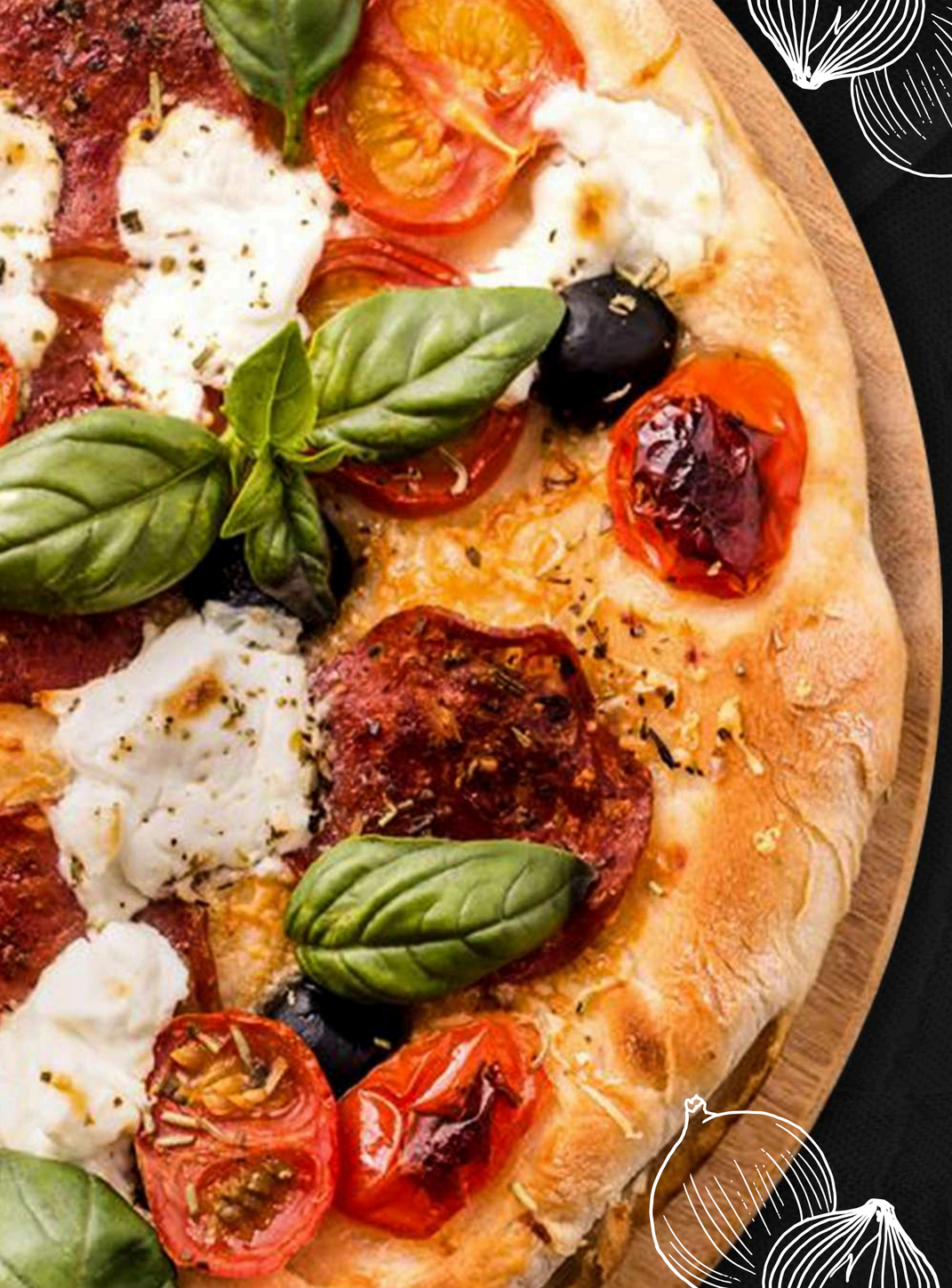


TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE

```sql
SELECT
    pizza_types.name AS Top_pizza_ordered,
    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 Top_pizza_ordered
ORDER BY revenue DESC
LIMIT 3
```

Result Grid | Filter Rows:

Top_pizza_ordered	Revenue
The Barbecue Chicken Pizza	3429.75
The California Chicken Pizza	2645.25
The Classic Deluxe Pizza	2309

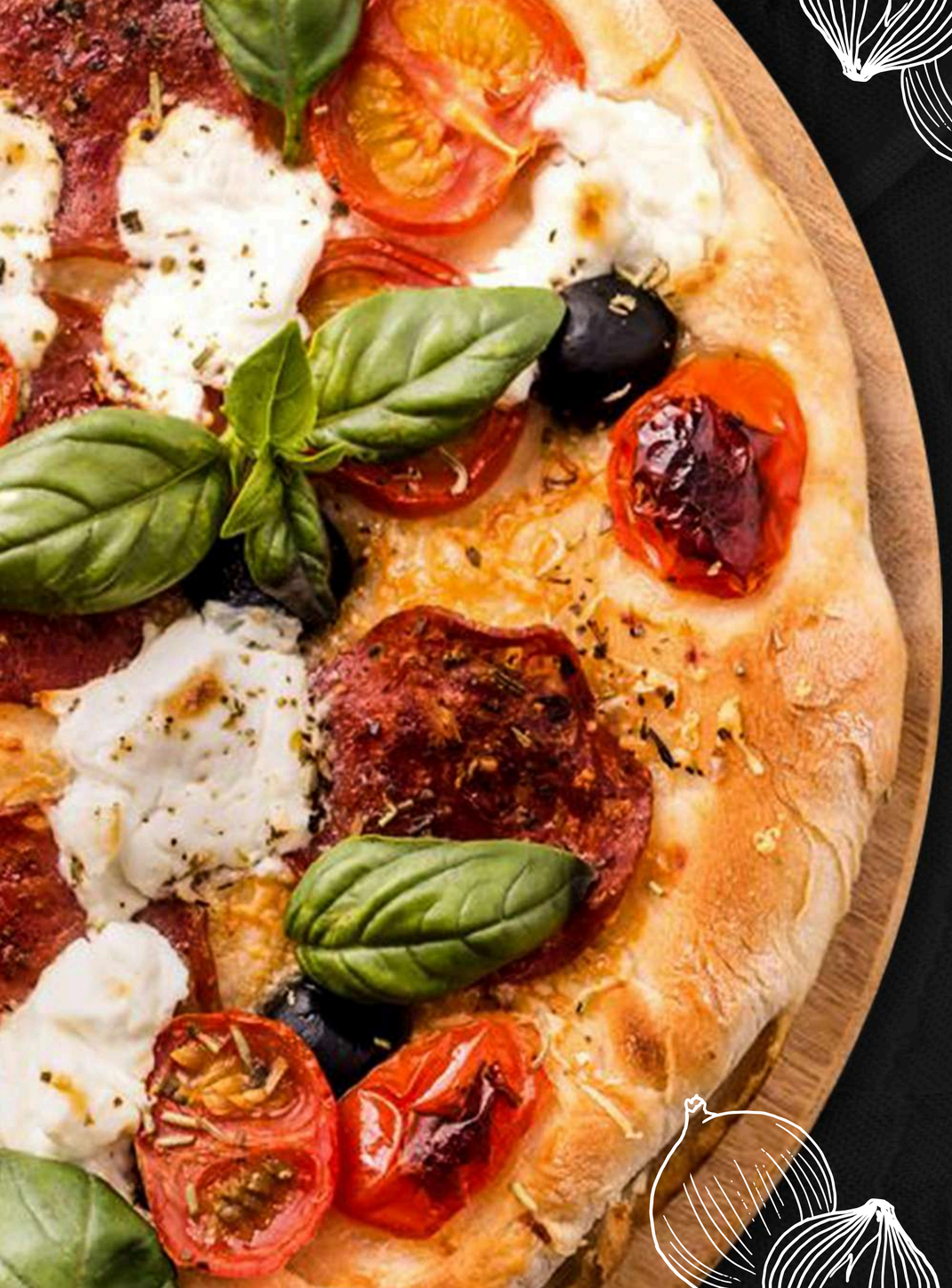


PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE

```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 pizza_types.pizza_type_id = pizzas.pizza_type_id
join order_details
on pizzas.pizza_id = order_details.pizza_id
group by pizza_types.category  order by revenue desc
```

Result Grid | Filter Rows:

category	revenue
Chicken	29.37
Classic	27.4
Veggie	21.73
Supreme	21.5

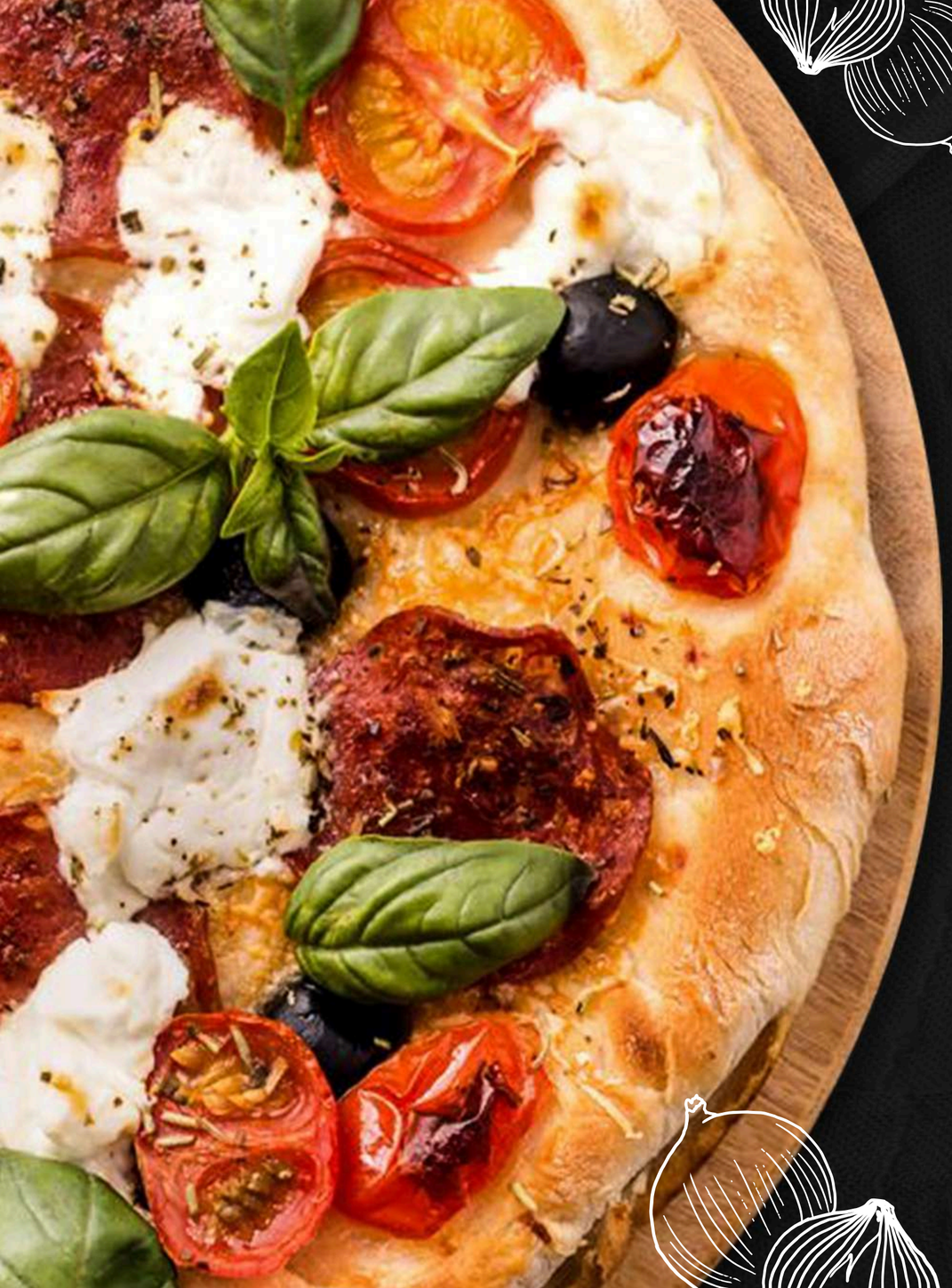


CUMULATIVE REVENUE GENERATED OVER TIME

```sql
select order_date,
sum(revenue) over(order by order_date) as cum_revenue
from
(select orders.order_date,
round(sum(order_details.quantity* pizzas.price),2) 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
2015-03-15	361.1
2015-03-16	844.8
2015-03-23	1782.35
2015-03-30	2281.6
2015-03-31	2609.35
2015-04-06	3148
2015-04-07	3333.25
2015-04-13	3514.75



TOP 3 MOST ORDERED PIZZA TYPES -- BASED ON REVENUE FOR EACH PIZZA CATEGORY

```sql
SELECT category, 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: | Export:

category	name	revenue
Chicken	The Barbecue Chicken Pizza	3429.75
Chicken	The California Chicken Pizza	2645.25
Chicken	The Chicken Alfredo Pizza	1318.5
Classic	The Classic Deluxe Pizza	2309
Classic	The Big Meat Pizza	1632
Classic	The Hawaiian Pizza	1479
Supreme	The Italian Supreme Pizza	1371
Supreme	The Calabrese Pizza	1100.25

THANK YOU!