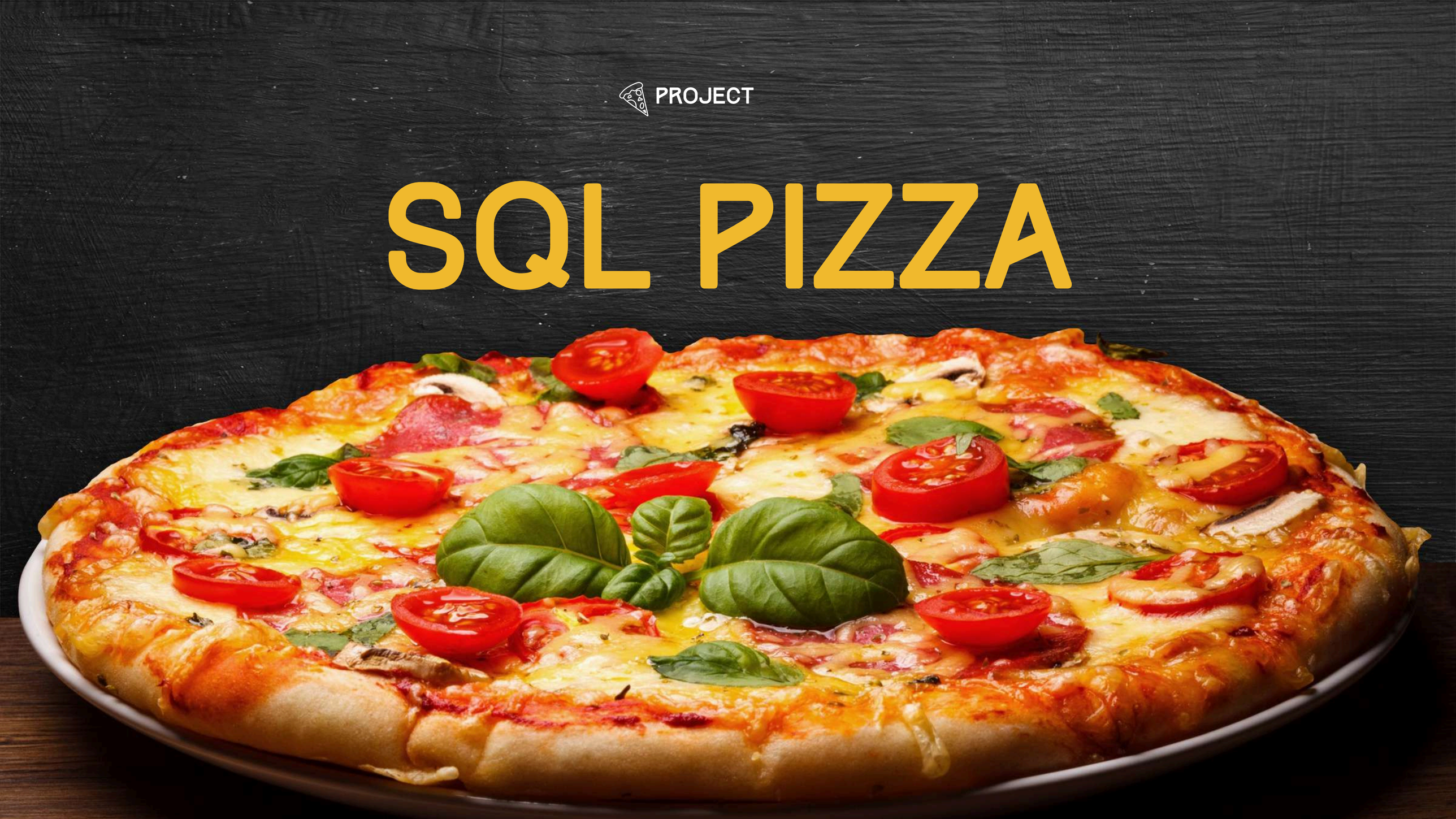
PROJECT

SQL PIZZA

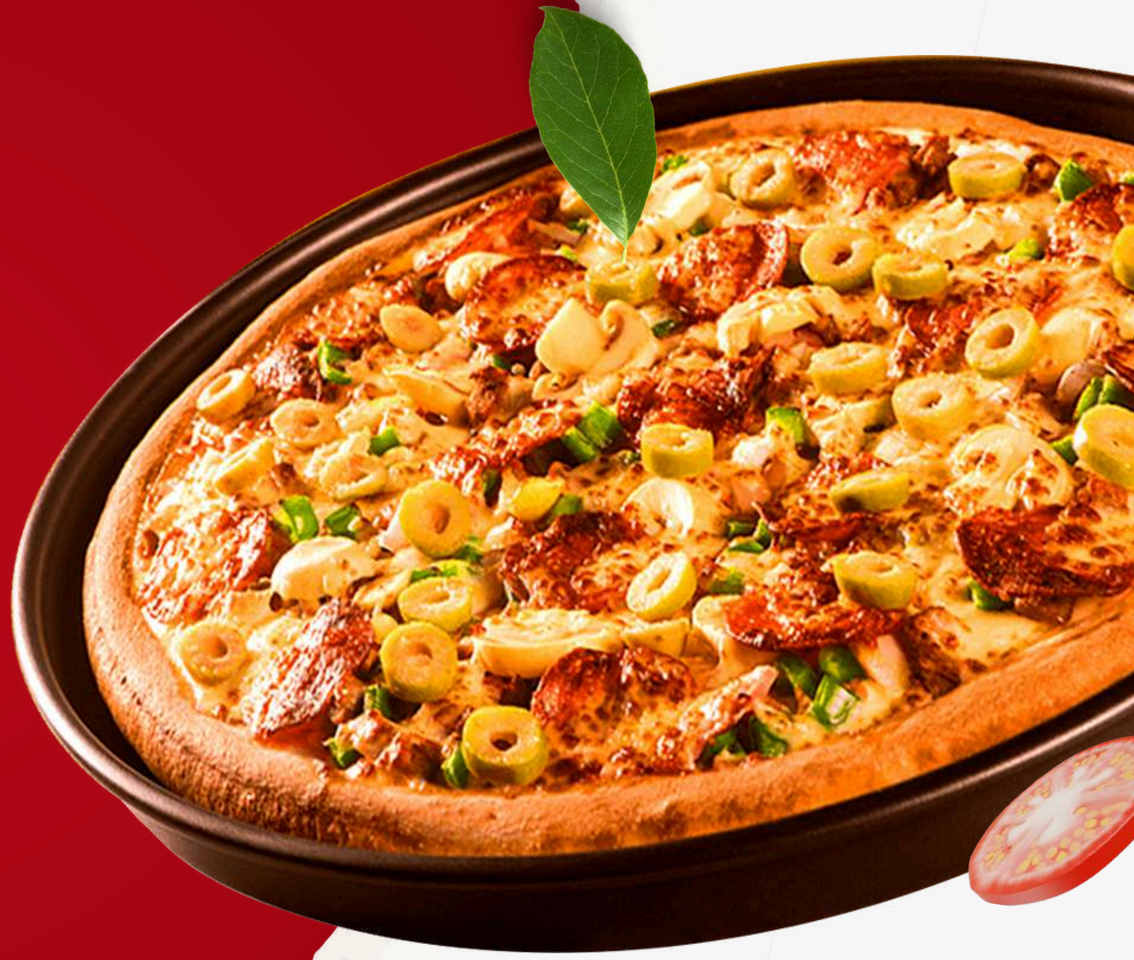


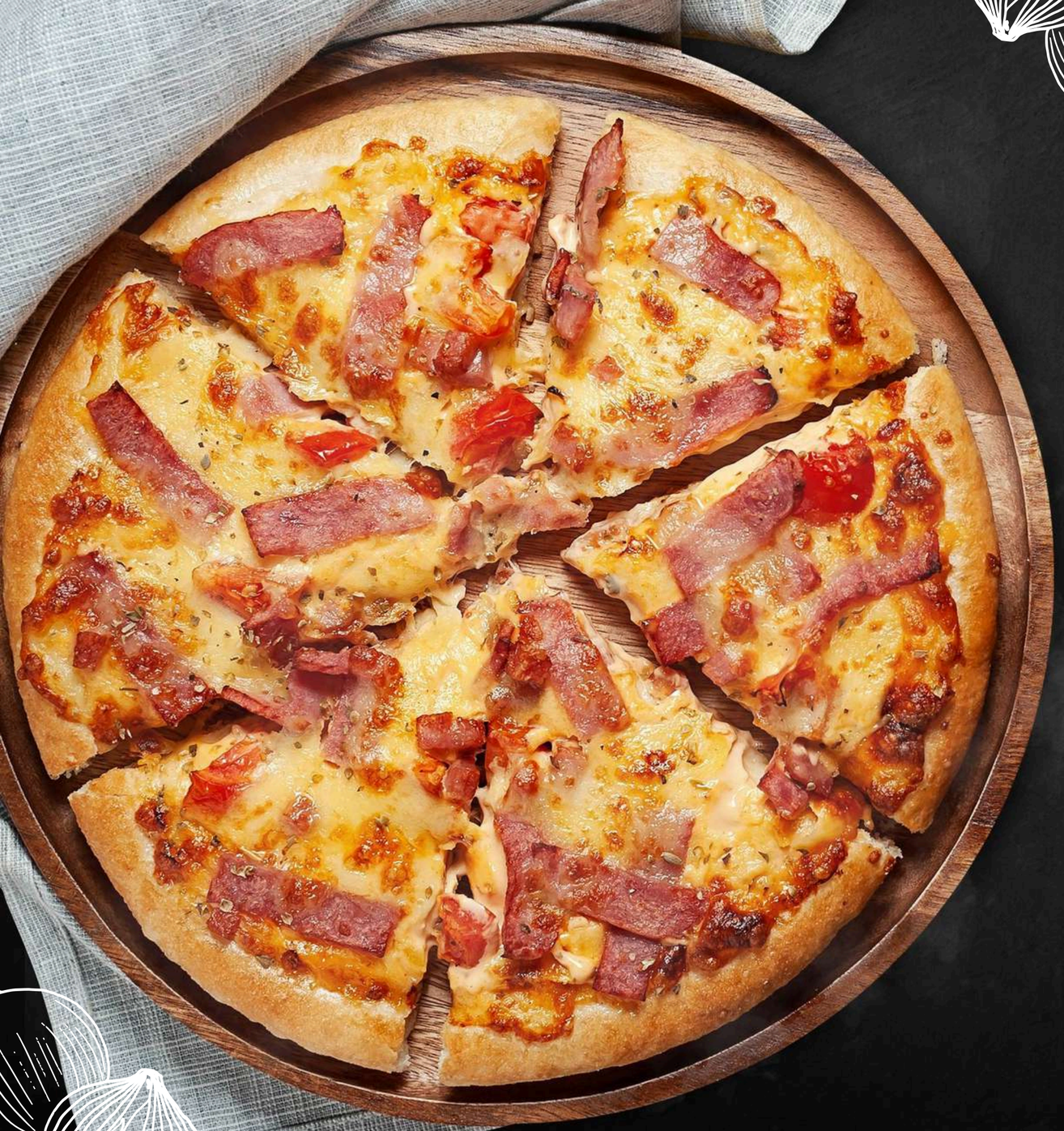
PROJECT

WELCOME TO

SQL PIZZA

The SQL Pizza Project is a data analysis project that aims to explore and analyze pizza sales data using SQL. This project provides insights into sales performance, customer preferences, and operational efficiency for a pizza restaurant. The project involves querying and analyzing sales data to understand customer preferences, sales trends and business performance metrics.





PROJECT

DATASET& GOALS

DATASET

The project is based on four key datasets:

1. Orders
2. Order Details
3. Pizzas
4. Pizza Types

GOALS

- Improve SQL querying skills with real-world business data.
- Gain experience in data exploration, aggregation, and trend analysis.
- Provide actionable insights to optimize menu offerings and increase restaurant profitability.

PROJECT QNS 1

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

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

Result Grid

	total_orders
▶	21350

PROJECT QNS 2

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 order_details.pizza_id = pizzas.pizza_id;
```

Result Grid

	total_revenue
▶	817860.05

PROJECT QNS 3

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 Rov

	name	price
▶	The Greek Pizza	35.95

PROJECT QNS 4

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

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

Result Grid | Filte

	size	no_of_pizza
▶	L	18526

PROJECT QNS 5

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

```sql
SELECT
    pizza_types.name,
    SUM(order_details.quantity) AS most_ordered
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 most_ordered DESC
LIMIT 5;
```

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

PROJECT QNS 6

JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.

```sql
SELECT
    pizza_types.category,
    SUM(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.category
ORDER BY total_quantity DESC;
```

	category	total_quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

PROJECT QNS 7

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

```
SELECT
    HOUR(order_time) AS hour, COUNT(order_id) AS orders
FROM
    orders
GROUP BY hour;
```

hour	orders
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

PROJECT QNS 8

JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.

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

	category	count
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

PROJECT QNS 9

GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.

```sql
SELECT
    ROUND(AVG(quantity), 0) as Avg_pizza_ordered
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_by_date;
```

	Avg_pizza_ordered
▶	138

PROJECT QNS 10

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

```sql
SELECT
    pizza_types.name,
    SUM(order_details.quantity * pizzas.price) AS total_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.name
ORDER BY total_revenue DESC
LIMIT 3;
```

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

PROJECT QNS 11

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

```sql
SELECT
    pizza_types.category,
    ROUND((SUM(pizzas.price * order_details.quantity) / (SELECT
                    SUM(order_details.quantity * pizzas.price) AS total_revenue
                FROM
                    order_details
                        JOIN
                    pizzas ON order_details.pizza_id = pizzas.pizza_id)) * 100,
            2) AS Percentage_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 Percentage_revenue DESC;
```

category	Percentage_revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

PROJECT QNS 12

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```
select
    order_date, sum(revenue) over(order by order_date) as cum_rev
from
    (select
        orders.order_date, sum(order_details.quantity * pizzas.price) as revenue
     from
        orders
            join
        order_details on orders.order_id=order_details.order_id
            join
        pizzas on order_details.pizza_id=pizzas.pizza_id
     group by orders.order_date) as sales;
```

order_date	cum_rev
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

PROJECT QNS 13

DETERMINE THE 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(pizzas.price * order_details.quantity) 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.name, pizza_types.category) as a) as b
where RN<=3;
```

category	name	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	32273.25

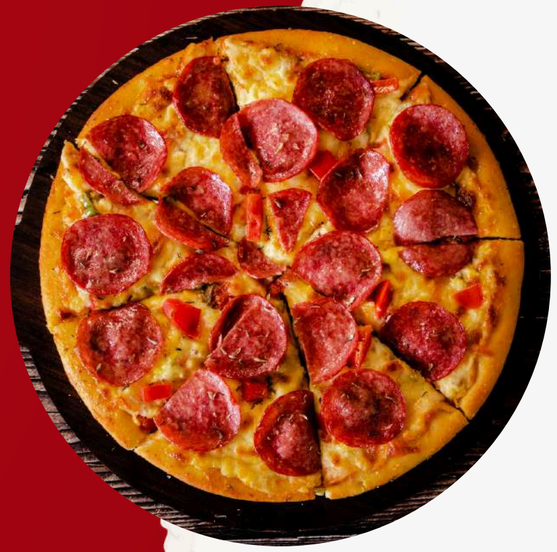
PROJECT

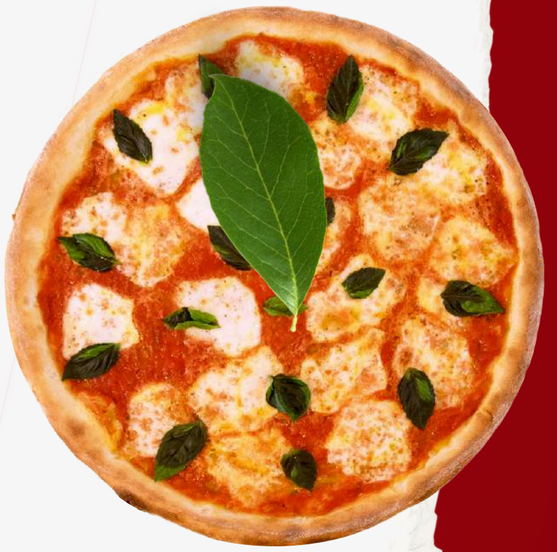
INSIGHT 1: TOTAL REVENUE

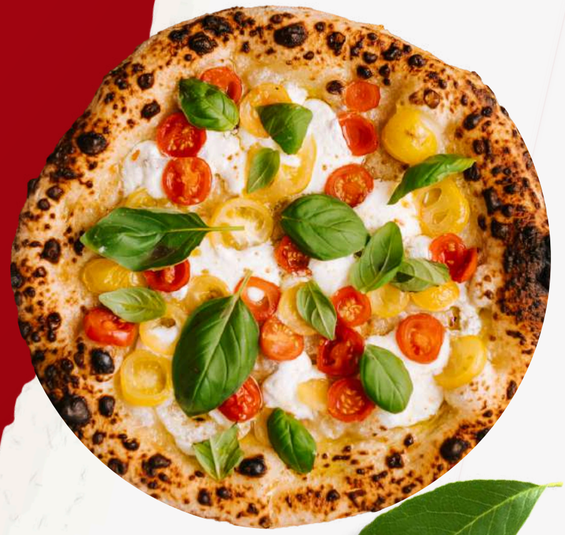
The total revenue generated from pizza sales is $817,860.05. This represents the overall earnings from all orders placed in the dataset.

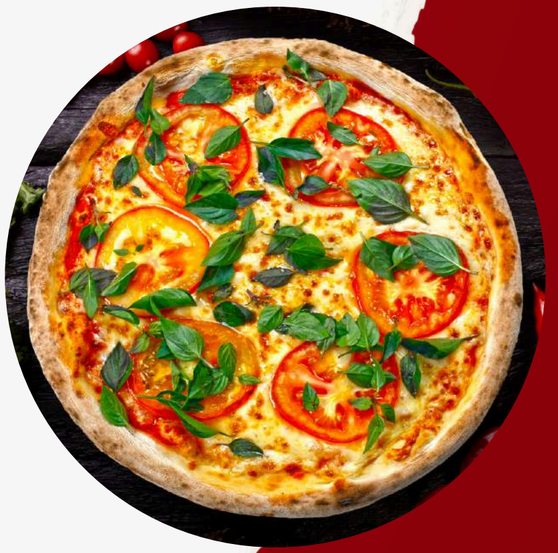
PROJECT

INSIGHT 2: MOST & LEAST POPULAR PIZZAS









- The most popular pizza is "big_meat_s" with 1,914 orders. This suggests that customers prefer this specific pizza the most.
- The least popular pizza is "the_greek_xxl", with only 28 orders, indicating low customer demand for this variant.

PROJECT

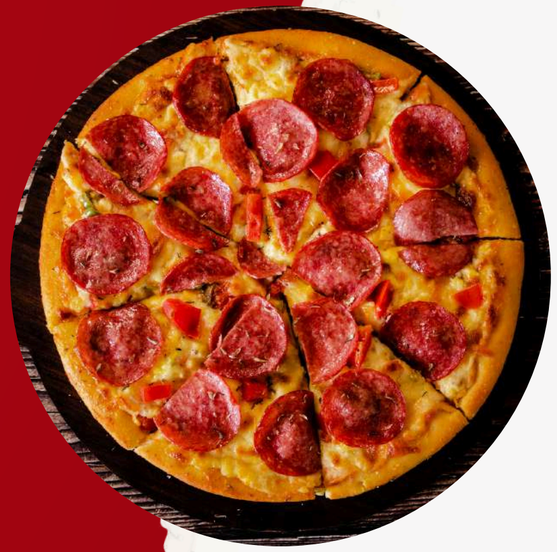
INSIGHT 3: PEAK ORDER TIMES

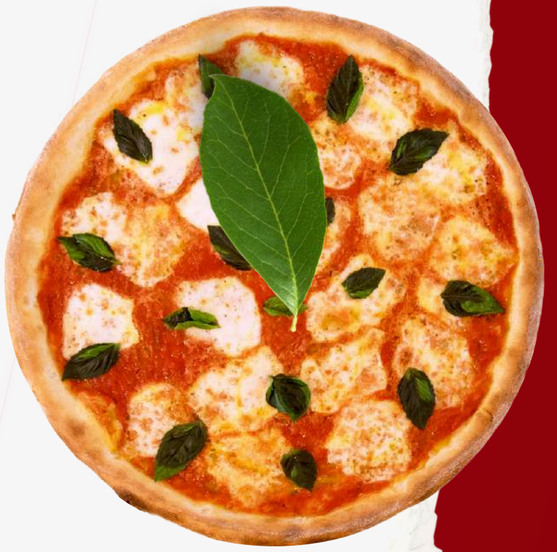
- The busiest hour for pizza orders is 12 PM (Noon) with 2,520 orders, followed closely by 1 PM (2,455 orders) and 6 PM (2,399 orders).
- Sales gradually decrease after 9 PM, with very few orders after 11 PM.
- This suggests that the restaurant experiences peak demand during lunch (12-1 PM) and dinner (5-7 PM).

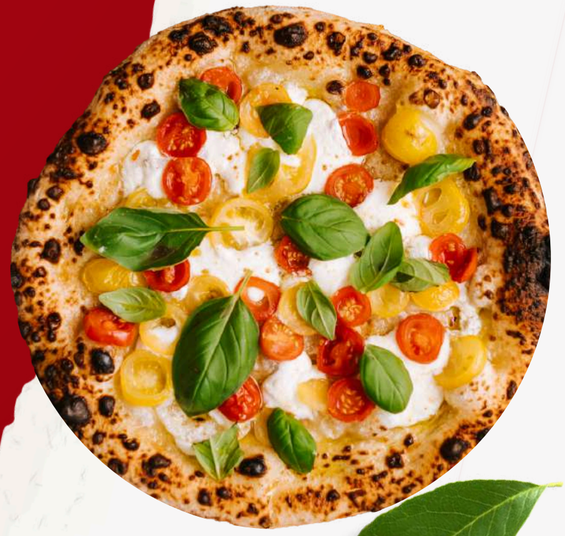
PROJECT

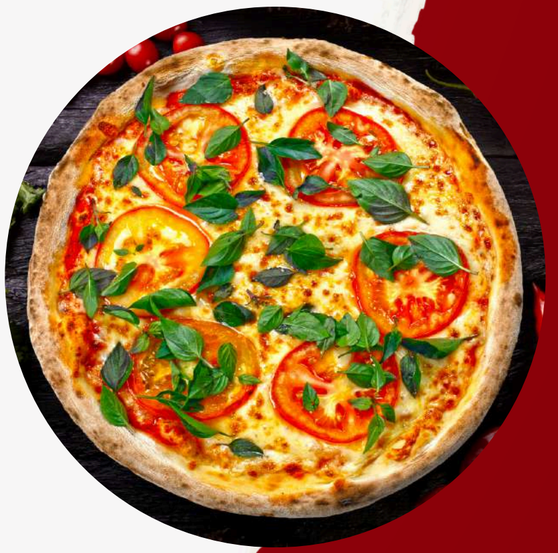
INSIGHT 4: SALES DISTRIBUTION BY PIZZA CATEGORY

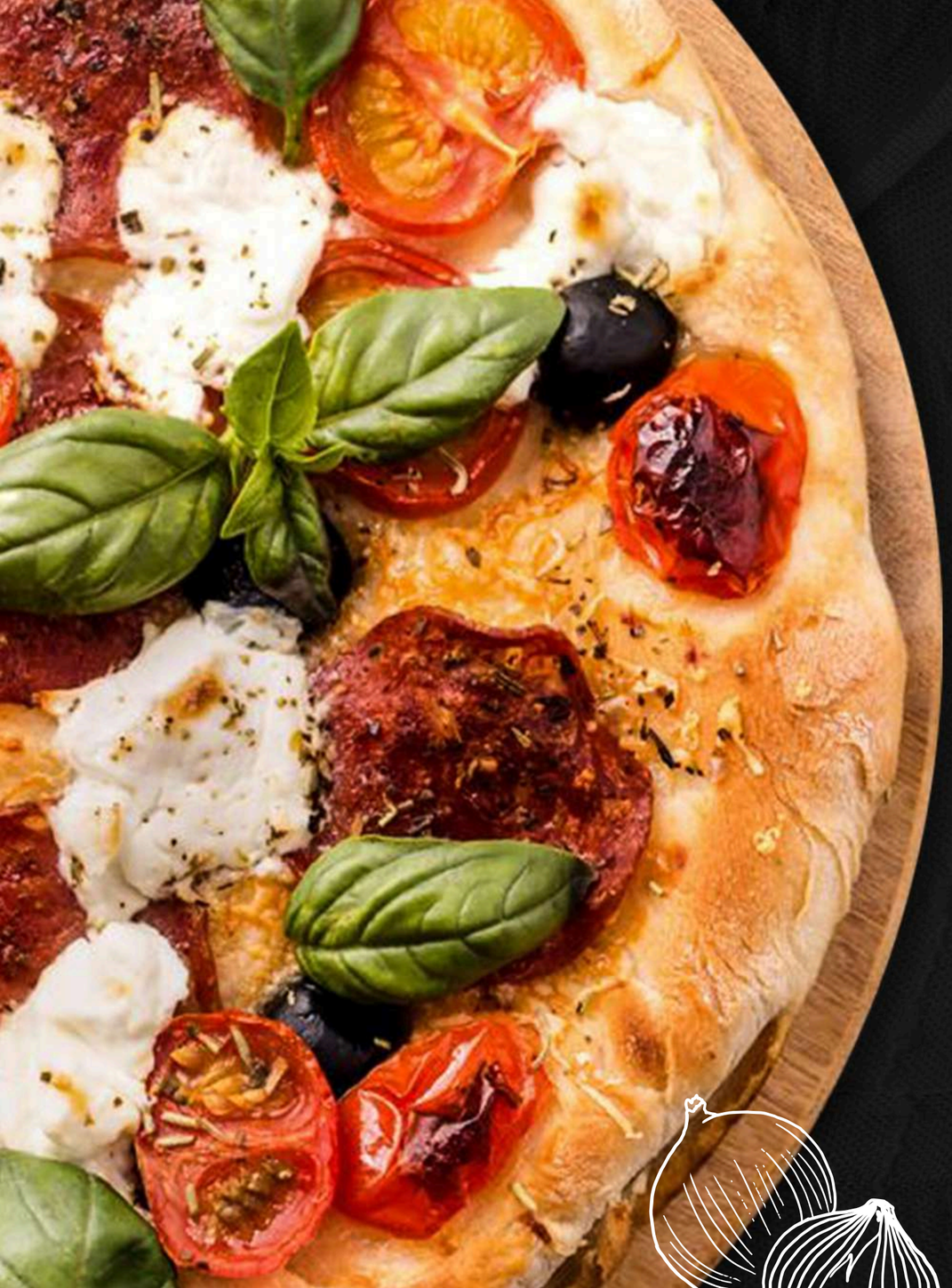
- Classic pizzas are the most popular, with 14,888 orders.
- Supreme and Veggie pizzas follow closely, with 11,987 and 11,649 orders, respectively.
- Chicken-based pizzas are the least ordered, with 11,050 orders.











PROJECT

MY CONTACT

+91-7736960562

muhammadadiltt10@gmail.com

Kozhikode, Kerala

https://www.linkedin.com/in/muhd-adil

PROJECT

THANK YOU!