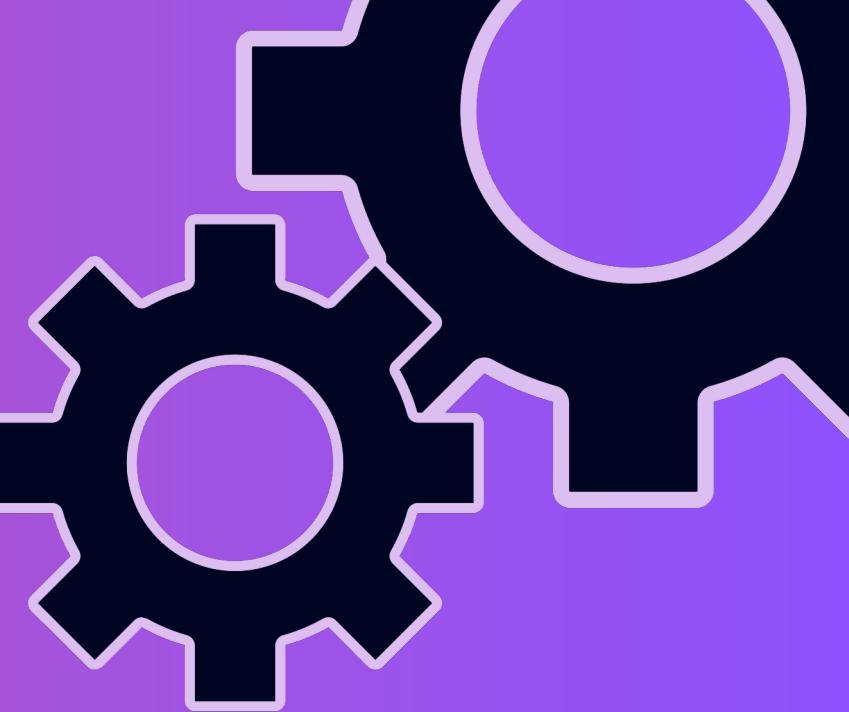


Pizza Sales Analysis using SQL



Project Overview



The main aim of this project is to analyze pizza sales to gain insights into customer behaviour, popular pizza types, sales trends and overall performance.

The analysis will be performed using four tables: orders_details, orders, pizza_types and pizzas.

Dataset:

■ orders_details

- orders_details_id: Unique identifier for the order detail.
- order_id: Identifier linking to the orders table. pizza_id:
- Identifier linking to the pizza table. quantity: Number of pizzas ordered.

■ orders

- order_id: Unique identifier for the order.
- date: Date the order was placed. time:
- Time the order was placed.



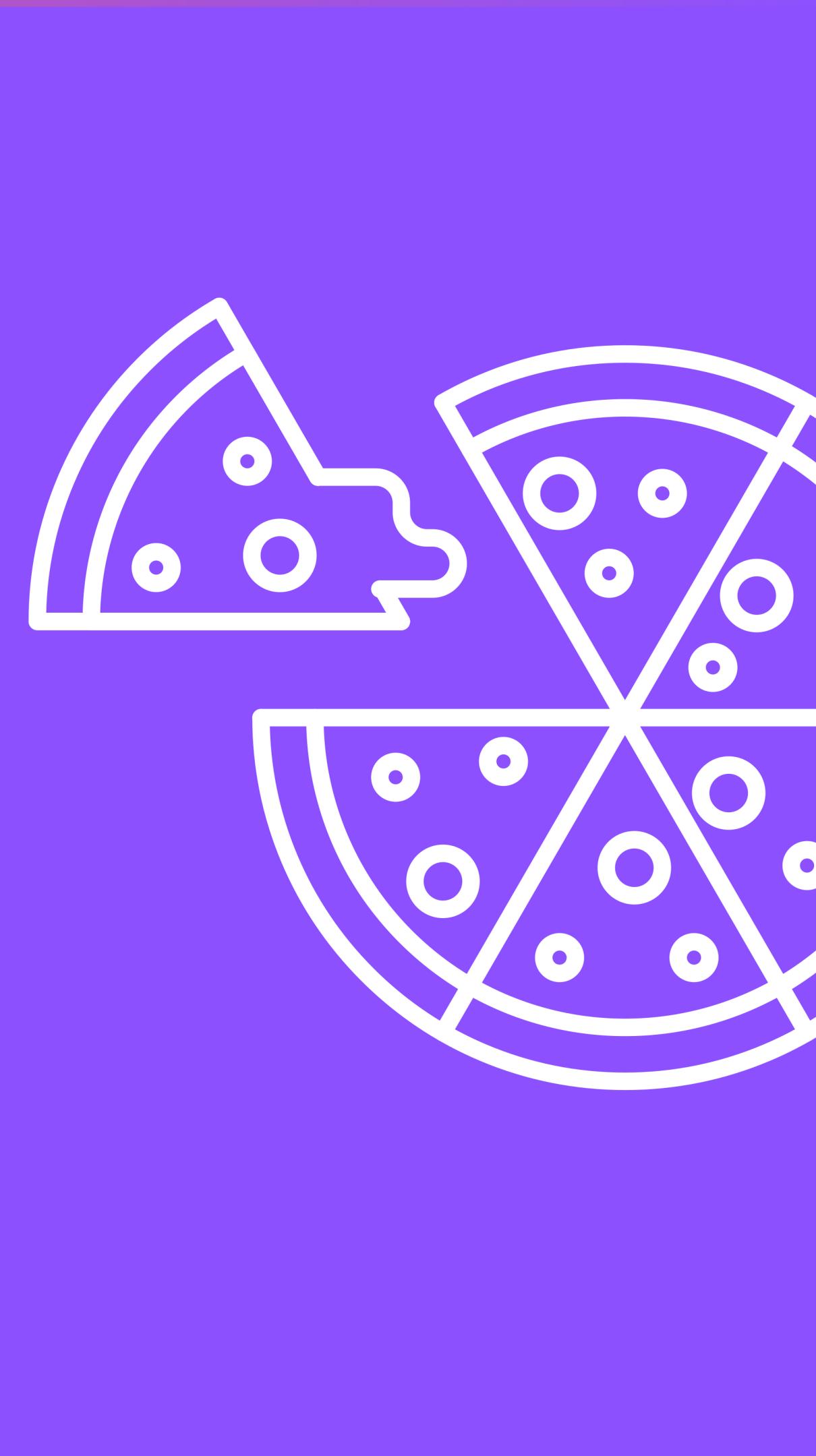
Dataset:

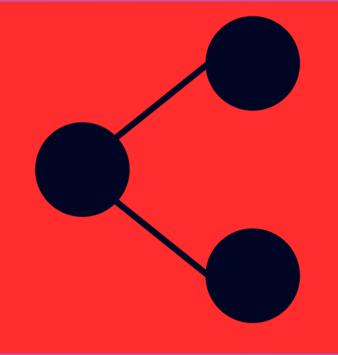
■ pizza_type

- pizza_type_id: Unique identifier for the pizza type.
- name: Name of the pizza.
- category: Category of the pizza(vegetarian, meat, etc.)
- ingredients: List of ingredients used in the pizza.

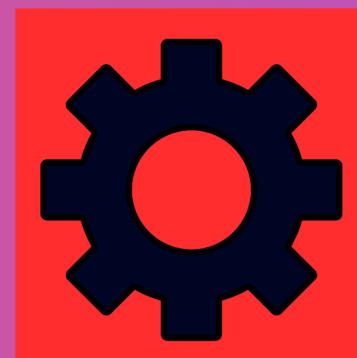
■ pizzas

- pizza_id: Unique identifier for the pizza.
- pizza_type_id: Identifier linking to the pizza_typetable.
- size: Size of the pizza (e.g., small, medium, large).
- price: Price of the pizza.





QUERIES



Question 1 The total number of orders placed.

Query:

```
select count(order_id) from orders;
```

Output:

	Total_orders
▶	21350

Question 2

The total revenue generated from pizza sales

Query:

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

Output:

Total_revenue
817860

Question 3 The highest priced pizza.

Query:

```
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:

	NAME	PRICE
▶	The Greek Pizza	35.95

Question 4

The most common pizza size ordered.

Query:

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

Output:

	SIZE	NO. OF ORDERS
▶	L	18526
	S	14137
	M	15385
	XL	544
	XXL	28

Question 5

The top 5 most ordered pizza types along their quantities.

Query:

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

Output:

	NAME	TOTAL QUANTITY
▶	The Hawaiian Pizza	2422
	The Classic Deluxe Pizza	2453
	The Five Cheese Pizza	1409
	The Italian Supreme Pizza	1884
	The Mexicana Pizza	1484

Question 6

The quantity of each pizza categories

Ordered.

Query:

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

Output:

	CATEGORY	TOTAL QUANTITY
▶	Classic	14888
	Veggie	11649
	Supreme	11987
	Chicken	11050

Question 7 The distribution of orders by hours of the day.

Query:

```
select hour(order_time), count(order_id) as order_count from orders  
group by hour(order_time);
```

Output:

HOUR OF THE DAY	NO. OF 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

Question 8

The category-wise distribution of pizzas.

Query:

```
select category, count(name) from pizza_types  
group by category;
```

Output:

CATEGORY	NO. OF PIZZA
Chicken	6
Classic	8
Supreme	9
Veggie	9

Question 9

The average number of pizzas ordered per day.

Query:

```
SELECT
    ROUND(AVG(quantity), 0) AS avg_pizza_ordered_per_day
FROM
    (SELECT
        orders.order_date, SUM(orders_details.quantity) AS quantity
    FROM
        orders
    JOIN orders_details ON orders.order_id = orders_details.order_id
    GROUP BY orders.order_date) AS order_quantity;
```

Output:

AVG ORDERS PER DAY	
▶	138

Question 10 Top 3 most ordered pizza type based on revenue.

Query:

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

Output:

NAME	TOTAL REVENUE
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

Question 11

The percentage contribution of each pizza type to revenue.

Query:

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

Output:

	CATEGORY	REV
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

Question 12

The cumulative revenue generated over time.

Query:

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

Output:

	DATE	DAILY_REVENUE	CUMULATIVE_REVENUE
▶	2015-01-01	2713.85	2713.85
	2015-01-02	2731.9	5445.75
	2015-01-03	2662.4	8108.15
	2015-01-04	1755.45	9863.6
	2015-01-05	2065.95	11929.55
	2015-01-06	2428.95	14358.5
	2015-01-07	2202.2	16560.7
	2015-01-08	2838.35	19399.05
	2015-01-09	2127.35	21526.4
	2015-01-10	2463.95	23990.35
	2015-01-11	1872.3	25862.65
	2015-01-12	1919.05	27781.7
	2015-01-13	2049.6	29831.3
	2015-01-14	2527.4	32358.7
	2015-01-15	1984.8	34343.5
	2015-01-16	2594.15	36937.65
	2015-01-17	2064.1	39001.75
	2015-01-18	1976.85	40978.6
	2015-01-19	2387.15	43365.75
	2015-01-20	2397.9	45763.65
	2015-01-21	2040.55	47804.2

Question 13

The top 3 most ordered pizza type based on revenue for each pizza category.

Query:

```
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(orders_details.quantity) * pizzas.price) as revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join orders_details
on orders_details.pizza_id = pizzas.pizza_id
group by pizza_types.category , pizza_types.name) as a ) as b
where rn <= 3;
```

Output:

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
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	32265.7
Veggie	The Mexicana Pizza	26780.75
Veggie	The Five Cheese Pizza	26066.5

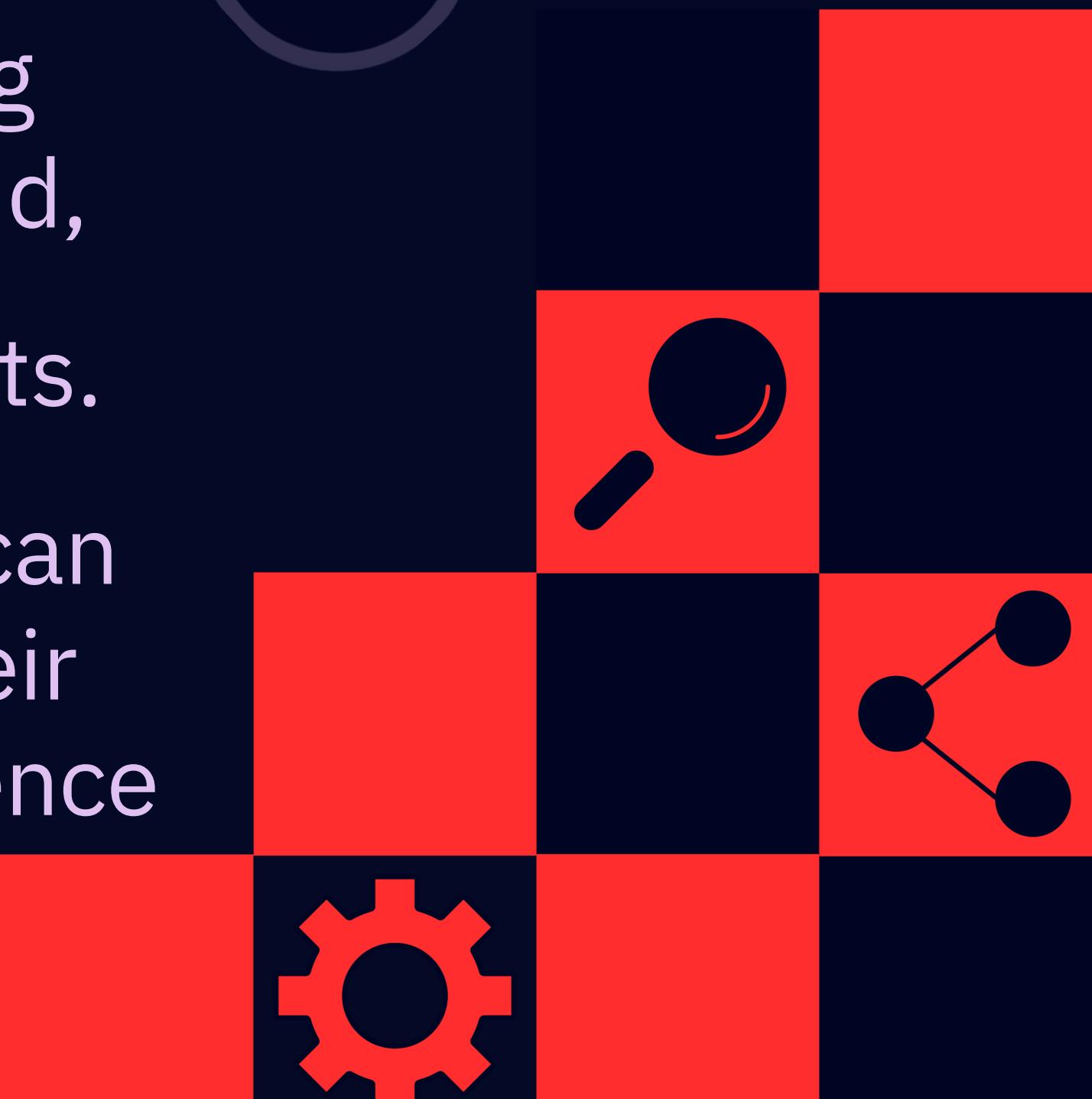
Conclusion

Through the sales analysis we have explored

various aspects of pizza orders, including customer preferences, overall sales trend,

product and cumulative revenue insights.

Based on these insights the pizza store can make informed decisions to enhance their customer experience, drive sales and hence achieve sustained growth.



Thank you!

SOM TIWARI

