

Pizza Sales Report

Delicious Pizza for Everyone!



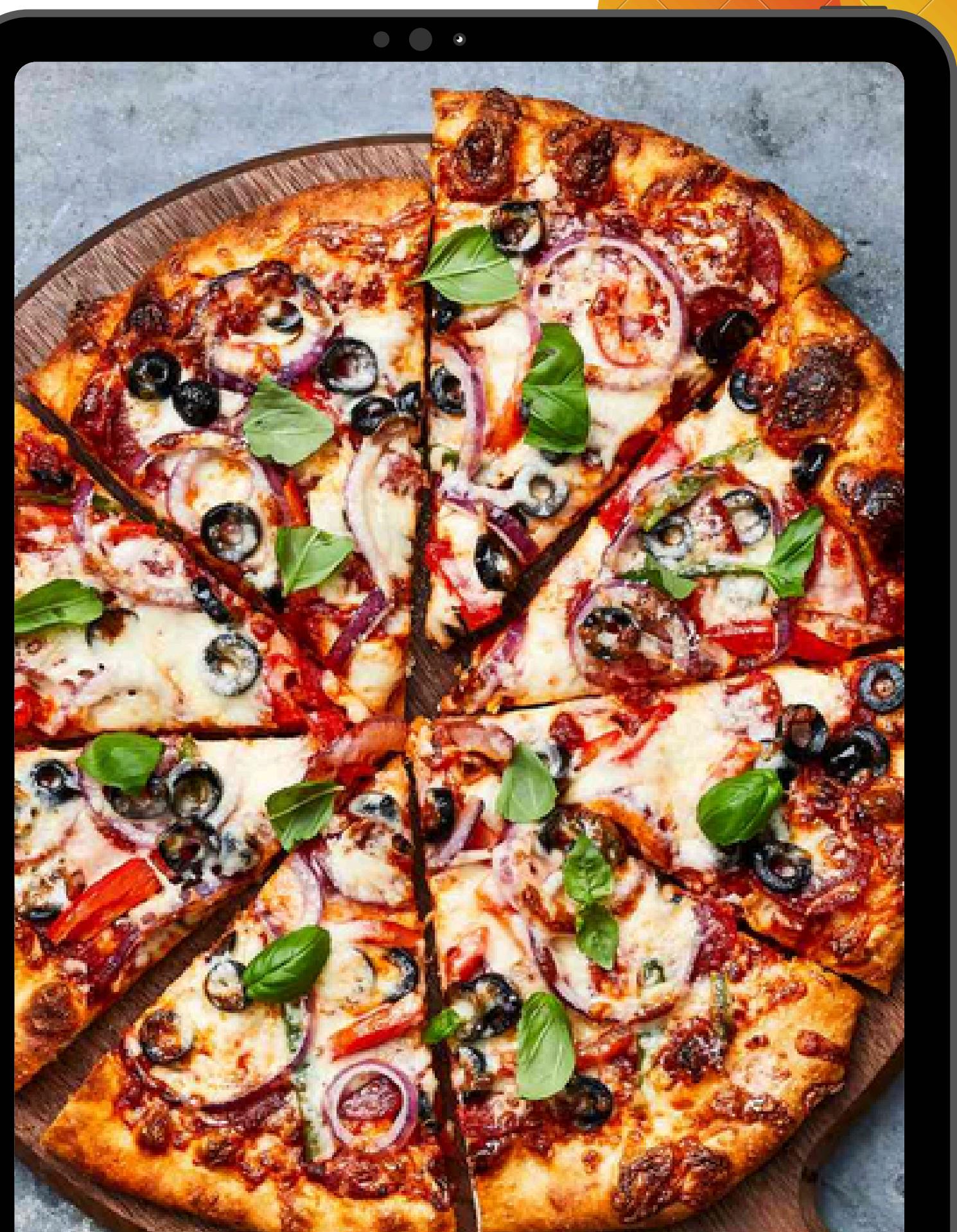
HELLO!

We are PizzaHut serving delicious pizza via the food truck way. We are ready to go around to deliver and serve pizza for you lovers!



Project Objective

The primary objective of this project was to analyze pizza sales data to identify trends, evaluate sales performance, and generate actionable insights for improving business strategies. The analysis aimed to help the pizza store understand customer preferences, peak sales times, and the performance of different pizza types



Retrieve the total number of orders placed.

```
-- Retrieve the total number of orders placed.
```

```
SELECT count(order_id) as total_orders FROM orders;
```

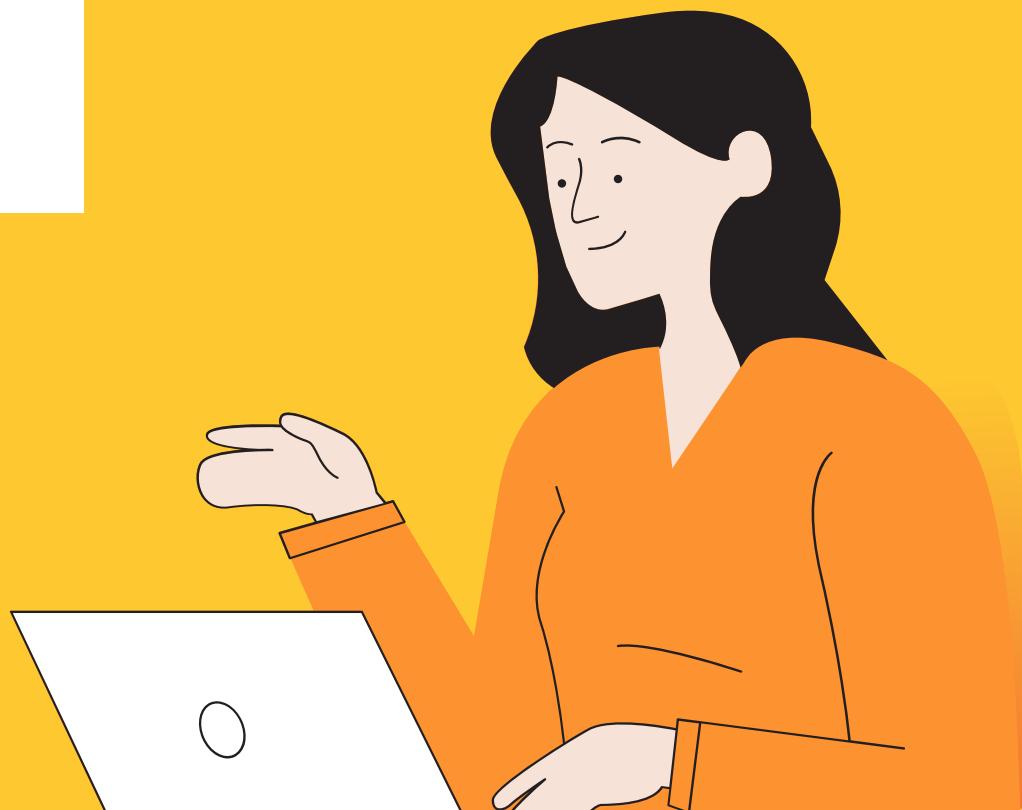
Result Grid	
	total_orders
▶	16821

Total revenue generated from pizza sales.

```
-- 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
▶	158005.8



The highest-priced pizza.

```
-- Identify the highest-priced pizza.

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

The most common pizza size ordered.

```
-- Identify the most common pizza size ordered.
```

```
SELECT  
    pizzas.size, COUNT(order_details.order_details_id)  
FROM  
    pizzas  
    JOIN  
        order_details ON pizzas.pizza_id = order_details.pizza_id  
GROUP BY pizzas.size;
```

	size	count(order_details.order_details_id)
▶	M	2879
	L	3659
	S	2765
	XL	97
	XXL	5

Top 5 most ordered pizza types along with their quantities.

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

```
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 Pepperoni Pizza	499
	The Barbecue Chicken Pizza	485
	The California Chicken Pizza	471
	The Hawaiian Pizza	468
	The Classic Deluxe Pizza	435

The total quantity of each ~~pizza~~ category ordered.

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

```
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;
```

	category	quantity
▶	Classic	2842
	Supreme	2318
	Veggie	2293
	Chicken	2126

The distribution of orders by hour of the day.

```
-- Determine the distribution of orders by hour of the day.  
SELECT  
    HOUR(order_time) AS hour, COUNT(order_id) AS order_count  
FROM  
    orders  
GROUP BY HOUR(order_time);
```

	hour	order_count
▶	11	964
	12	1967
	13	1920
	14	1208
	15	1151
	16	1505
	17	1879
	18	1892
	19	1561
	20	1272
	21	956
	22	516
	23	23
	10	7

The category-wise distribution of pizzas.

```
-- Join relevant tables to find the 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

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

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

```
SELECT  
    ROUND(AVG(quantity), 0)  
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	
	ROUND(AVG(quantity), 0)
▶	137

The top 3 most ordered pizza types based on revenue.

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

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 Barbecue Chicken Pizza	8587.75
	The California Chicken Pizza	8205.25
	The Thai Chicken Pizza	7884.75

The percentage contribution of each pizza type to total revenue.

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

```
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 order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY revenue DESC;
```

	category	revenue
▶	Classic	26.48
	Supreme	25.45
	Veggie	24.2
	Chicken	23.87

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

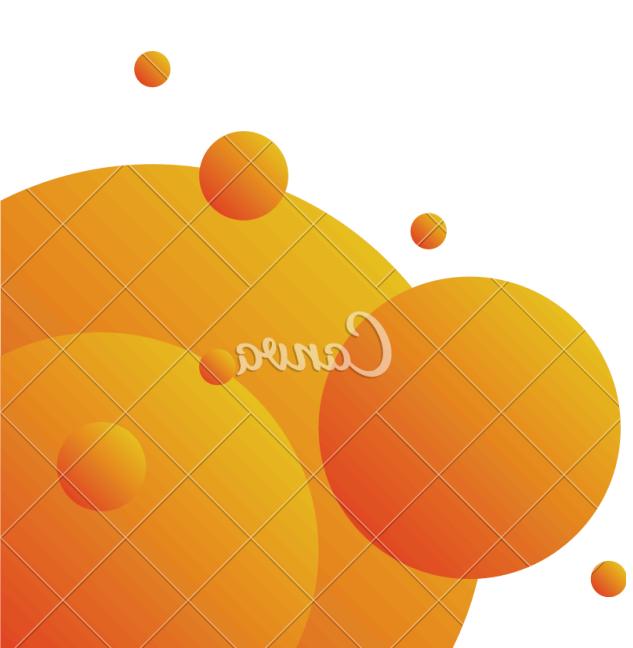
-- Analyze the cumulative revenue generated over time.

```
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;
```

order_date	cum_revenue
2015-01-24	55013.850000000006
2015-01-25	56631.400000000001
2015-01-26	58515.800000000001
2015-01-27	61043.850000000001
2015-01-28	63059.850000000001
2015-01-29	65105.150000000016
2015-01-30	67375.450000000001
2015-01-31	69793.300000000002
2015-02-01	72982.500000000001
2015-02-02	75311.100000000002
2015-02-03	77925.900000000002
2015-02-04	80159.800000000002
2015-02-05	82375.600000000002
2015-02-06	84885.550000000002
2015-02-07	87123.200000000001
2015-02-08	89158.200000000001
2015-02-09	91353.550000000002
2015-02-10	93410.050000000002
2015-02-11	95870.050000000002
2015-02-12	98028.850000000002
2015-02-13	100783.350000000002
2015-02-14	103102.500000000001
2015-02-15	105243.750000000001

CONCLUSION

The project provided valuable insights into pizza sales patterns and customer preferences. By understanding peak sales times and popular pizza choices, the pizza store can optimize its inventory management, improve marketing strategies and enhance overall customer satisfaction. The findings support data-driven decision-making to boost sales and operational efficiency.



Thank You

