

SQL



@YASH_PATIL

PIZZA SALES



PIZZA SALES BUSSINESS PROBLEM

In this project, we dive deep into the sales data of a pizza business to uncover meaningful insights and help the business make informed decisions. By leveraging SQL, I have tackled several important business-related questions that shed light on the overall performance and trends in pizza sales.

The store needs to boost revenue and efficiency by identifying top-selling pizzas, peak sales times, and optimizing inventory to avoid missed opportunities and potential revenue loss.





@YASHPATIL

VISSION & MISSION

To Analyse pizza sales data,
focusing on revenue, order trends
and make business decisions.





1. RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

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

Result Grid	
	total_orders
▶	885



2. CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```
SELECT  
    SUM(order_details.quantity * pizzas.price) AS total_sale  
FROM  
    order_details  
    JOIN  
    pizzas ON pizzas.pizza_id = order_details.pizza_id
```

Result Grid	
	total_sale
▶	151731.44999999996



3. 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;
```

★★★★★
THE GREEK PIZZA

Result Grid		Filter Rows:
	name	price
▶	The Greek Pizza	35.95





4. IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```
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

	size	order_count
▶	L	3499
	M	2774
	S	2659
	XL	97
	XXL	5



@YASH_PATIL

5. 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	486
	The Barbecue Chicken Pizza	459
	The California Chicken Pizza	459
	The Hawaiian Pizza	446
	The Classic Deluxe Pizza	413



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

@YASH_PATIL 



```
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

	category	quantity
▶	Classic	2733
	Supreme	2242
	Veggie	2203
	Chicken	2025

< BACK



7. DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.



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

Result Grid | Filter Rows

	HOUR(order_time)	orders
▶	11	44
	12	113
	13	100
	14	81
	15	68
	16	80
	17	95

< BACK

NEXT >



8. 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



< BACK

NEXT >

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

```
SELECT  
    ROUND(AVG(quantity), 0) as avg_pizza_ordered_per_day  
FROM  
(SELECT  
    orders.order_data, SUM(order_details.quantity) AS quantity  
FROM  
    orders  
JOIN order_details ON orders.order_id = order_details.order_id  
GROUP BY orders.order_data) AS order_quantity
```

Result Grid	
	avg_pizza_ordered_per_day
▶	136



NEXT >

10. 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;
```

	name	revenue
▶	The Barbecue Chicken Pizza	8144.25
	The California Chicken Pizza	7980.25
	The Thai Chicken Pizza	7446.75



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

```
• SELECT
    pizza_types.category,
    ROUND(
        (SUM(order_details.quantity * pizzas.price) /
        (SELECT SUM(order_details.quantity * pizzas.price) FROM order_details
        JOIN pizzas ON pizzas.pizza_id = order_details.pizza_id)
        ) * 100, 2
    ) AS revenue_percentage
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_percentage DESC;
```

	category	revenue_percentage
▶	Classic	26.53
	Supreme	25.58
	Veggie	24.21
	Chicken	23.68



12. 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, -- Renaming from 'order_data' to '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
▶	2015-01-01	2713.8500000000004
	2015-01-02	5445.75
	2015-01-03	8108.15
	2015-01-04	9863.6
	2015-01-05	11929.55



13. DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.



```
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;
```



	name	revenue
▶	The Barbecue Chicken Pizza	8144.25
	The California Chicken Pizza	7980.25
	The Thai Chicken Pizza	7446.75
	The Classic Deluxe Pizza	6403
	The Pepperoni Pizza	6088.75



YASH_PATIL



Pune , Maharashtra



yash.data6302@gmailcom



+91 9112104063

THANK YOU!

