



SQL PROJECT : PIZZA SALES ANALYSIS

By analysing the pizza sales data using SQL, We uncover valuable insights that help optimize business decisions.

~ By Shreya Kirti
Course : BCA



ABOUT THE PROJECT :

Objective:

To analyze pizza sales data using SQL and derive meaningful business insights to improve operations and customer satisfaction.

Tools Used:

MySQL for querying the database
SQL (Joins, Aggregations, Window Functions)
Canva / PowerPoint for presentation design

04

Dataset Tables:

- **orders:** Information about pizza orders
- **orders_details:** Quantity and pizza reference per order
- **pizzas:** Price, size, and type reference
- **pizza_types:** Name and category of pizza types



DATASET SCHEMA:

Understanding the data structure is essential for writing effective queries. The database contains interconnected tables:

orders(order_id, order_date)

orders_details(order_details_id, order_id, pizza_id, quantity)

pizzas(pizza_id, pizza_type_id, size, price)

pizza_types(pizza_type_id, name, category)

MySQL Relationships:

orders -> orders_details -> pizzas -> pizza_types



1. Retrieve the total number of orders placed.

About

Contact

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

A screenshot of a MySQL Workbench interface showing the results of the SQL query. The results are displayed in a grid with one row and two columns. The first column is labeled 'total_orders' and contains the value '21350'.

	total_orders
▶	21350



2. Calculate the total revenue generated from pizza sales.

Home

Contact

```
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		Filter Rows:	Export:	Wrap Cell Content:	
total_sales		817860.05			



3. Identify the highest-priced pizza.

[About](#)[Contact](#)

```
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: Export: Wrap Cell Co

	name	price
▶	The Greek Pizza	35.95

4. Identify the most common pizza size ordered.

[Home](#)[About](#)[Contact](#)

```
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: Export:  Wrap Cell Content: 

	size	order_count
▶	L	18526

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

[Home](#)[Contact](#)

```
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;
```

Result Grid |  Filter Rows: Export:  Wrap Cell Content:

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



6. 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;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Con
	category	quantity		
▶	Classic	14888		
	Supreme	11987		
	Veggie	11649		
	Chicken	11050		





7. 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);
```



Result Grid | Filter Rows: _____ | Export: | Wrap Cell Content: |

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

Result 1 ×



8. Join relevant tables to find the category wise distribution of pizzas.

[Home](#)[About](#)[Contact](#)

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



Result Grid |  Filter Rows: _____ | Export:  Wrap Cell Content: 

	category	COUNT(name)
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9





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

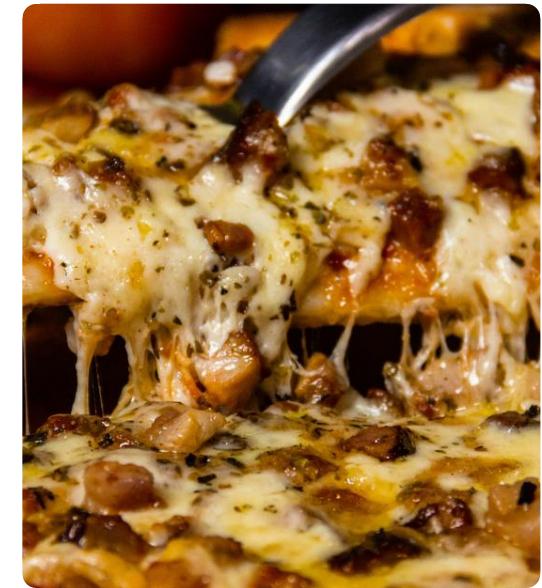
```
SELECT  
    ROUND(AVG(quantity), 0) AS avg_pizza_ordered_per_day  
FROM  
    (SELECT  
        orders.order_date, SUM(order_details.quantity) AS quantity  
    FROM  
        orders  
    JOIN order_details ON orders.order_id = order_details.order_id  
    GROUP BY orders.order_date) AS order_quantity;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



Result Grid		Filter Rows:	Export:	Wrap Cell Content:		
<table border="1"><thead><tr><th>avg_pizza_ordered_per_day</th></tr></thead><tbody><tr><td>138</td></tr></tbody></table>				avg_pizza_ordered_per_day	138	
avg_pizza_ordered_per_day						
138						



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

[Home](#)[About](#)

```
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 Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5



12. 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
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;
```



Result Grid		
	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68





13. 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 BY order_date;
```



	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
	2015-01-06	14358.5
	2015-01-07	16560.7
	2015-01-08	19399.05
	2015-01-09	21526.4
	2015-01-10	23990.350000000002
	2015-01-11	25862.65
	2015-01-12	27781.7
	2015-01-13	29831.300000000003



14. 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 Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5
	The Classic Deluxe Pizza	38180.5
	The Hawaiian Pizza	32273.25
	The Pepperoni Pizza	30161.75
	The Spicy Italian Pizza	34831.25
	The Italian Supreme Pizza	33476.75
	The Sicilian Pizza	30940.5
	The Four Cheese Pizza	32265.70000000065
	The Mexicana Pizza	26780.75
	The Five Cheese Pizza	26066.5





THANK YOU

