



DOMINO'S PIZZA





Domino's Pizza Sales Data Analysis

Hello !

My name is Manoj, I am passionate about leveraging data to solve real-world problems and enhance business performance. In my free time, I enjoy exploring new technologies and working on personal projects related to data analytics.

I have a knack for finding insights in complex data sets and translating them into practical solutions that drive business success.

My expertise lies in data analysis, SQL, and database management, which I've applied in this project to generate valuable insights.



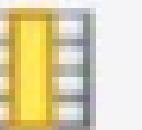


Project Objective :

- The objective of a Domino's Pizza sales data analysis project using MySQL typically involves extracting valuable insights from sales data to enhance decision-making processes and optimize sales strategies.
- Calculate key metrics such as total revenue, average order value, total pizzas sold, and average pizzas per order.
- Identify trends and patterns in sales data, such as daily and hourly trends, seasonal variations, and growth trends,
- Assess the performance of different pizza categories and individual pizzas to determine top sellers and revenue contributors.
- Provide actionable insights and recommendations to optimize business strategies, such as menu adjustments, promotional offers, and marketing campaigns.

Retrieve the total number of orders placed.

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

| Result Grid |  

	total_orders
▶	21350

Calculate the total revenue generated from pizza sales.

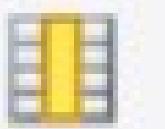
```
SELECT  
    ROUND(SUM(quantity * price), 2) AS total_sales  
FROM  
    order_details  
    JOIN  
    pizzas ON pizzas.pizza_id = order_details.pizza_id;
```

| Result Grid |  

	total_sales
▶	817860.05

Identify the highest-priced pizza.

```
SELECT  
    pizza_types.name, pizzas.price  
FROM  
    pizza_types  
        JOIN  
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id  
ORDER BY pizzas.price DESC  
LIMIT 1;
```

Result Grid |  Filter Rows: 

	name	price
▶	The Greek Pizza	35.95

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 order_details.pizza_id = pizzas.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC
LIMIT 1;
```

Result Grid |   Filter Rows:

	size	order_count
▶	L	18526

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

• **SELECT**

```
    pizza_types.name,  
    SUM(order_details.quantity) AS total_quantity  
  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 total_quantity DESC  
  LIMIT 5;
```

Result Grid | Filter Rows:

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

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

- **SELECT**

```
pizza_types.category,  
SUM(order_details.quantity) AS total_quantity  
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.category  
ORDER BY total_quantity DESC;
```

Result Grid | Filter Rows:

	category	total_quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

Determine the distribution of orders by hour of the day.

```
• SELECT  
    HOUR(orders.time) AS hours, COUNT(order_id) AS order_count  
FROM  
    orders  
GROUP BY hours;
```

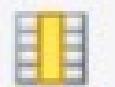
Result Grid | Filter Rows:

	hours	order_count
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468

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

• **SELECT**

```
category, COUNT(name) AS total_types  
FROM  
pizza_types  
GROUP BY category  
ORDER BY total_types DESC;
```

Result Grid |  Filter Rows:

	category	total_types
▶	Supreme	9
	Veggie	9
	Classic	8
	Chicken	6

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

• **SELECT**

```
    ROUND(AVG(total_quantity), 0) AS avg_pizza_orderd_per_day  
FROM  
(SELECT  
    orders.date, SUM(order_details.quantity) AS total_quantity  
FROM  
    orders  
JOIN order_details ON order_details.order_id = orders.order_id  
GROUP BY orders.date) AS quantity_data;
```

Result Grid | Filter Rows:

	avg_pizza_orderd_per_day
▶	138

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

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(quantity * 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 pizzas.pizza_type_id = pizza_types.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	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

Analyze the cumulative revenue generated over time.

```
• select date,  
    sum(revenue) over(order by date) as cumulative_revenue  
from  
    (select orders.date,  
        sum(order_details.quantity * pizzas.price)as revenue  
    from orders  
    join order_details  
    on order_details.order_id = orders.order_id  
    join pizzas  
    on pizzas.pizza_id = order_details.pizza_id  
    group by orders.date) as total_revenue;
```

Result Grid |  Filter Rows:

	date	cumulative_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

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

```
• select category, name, revenue
  from
  (select category, name, revenue,
  rank() over(partition by category order by revenue desc) as ranking
  from
  (select pizza_types.category, 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.category, pizza_types.name)as table_a)as table_b
  where ranking <=3;
```

Result Grid | Filter Rows: Export:

	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



DOMINO'S PIZZA

THANK YOU!

