

# PIZZA HUT



# PROJECT PIZZA HUT

"My name is Abhishek, "I have analyzed the Pizza Hut sales dataset, utilizing **MySQL** to query and extract meaningful insights from the data."

In this project, I explored and analyzed the Pizza Hut sales dataset by performing various SQL queries to uncover trends, patterns, and key performance metrics. The analysis aimed to provide valuable insights into sales performance, customer behavior, and product preferences, offering a deeper understanding of the business's operations.



```
--- ----- # Retrieve total number of orders placed?  
• SELECT  
    COUNT(order_id) AS 'totalorders'  
FROM  
    orders;
```

Result Grid	
	<b>totalorders</b>
▶	21350

```
8      --- # Calculate total revenue generated from pizza sales
9
0 • SELECT
1   ROUND(SUM(orders_details.quantity * pizzas.price),
2           2) AS totalrevenue
3
4 FROM
5   orders_details
6   JOIN
7     Pizzas ON pizzas.pizza_id = orders_details.pizza_id;
```

Result Grid	
	totalrevenue
▶	817860.05

```
18      --- # Identify the Highest priced Pizza
19
20 •   SELECT |
21     pizza_types.name, pizzas.price
22   FROM
23     pizza_types
24   JOIN
25     pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
26 ORDER BY pizzas.price DESC
27 LIMIT 1;
```

Result Grid | Filter Rows

	name	price
▶	The Greek Pizza	35.95

```
9      ---- # Identify the most common pizza size ordered
0
1 •   SELECT
2     pizzas.size,
3     COUNT(orders_details.order_details_id) AS Order_count
4   FROM
5     pizzas
6       JOIN
7     orders_details ON orders_details.pizza_id = pizzas.pizza_id
8   GROUP BY pizzas.size
9   ORDER BY order_count desc;
```

Result Grid | Filter

	size	Order_count
▶	L	18526
	M	15385
	S	14137
	XL	544
	XXL	28

```
41  -- --- # List the top 5 most ordered pizza types along with their quantities
42
43 • Select
44      pizza_types.name, Sum(orders_details.quantity) as quantity
45  FROM
46      pizza_types
47      JOIN
48      pizzas ON Pizza_types.pizza_type_id = pizzas.pizza_type_id
49      JOIN
50      orders_details ON orders_details.pizza_id = pizzas.pizza_id
51  GROUP BY pizza_types.name
52  ORDER BY quantity DESC
53  LIMIT 5;
```

Result Grid | Filter Rows:

	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

```
-- --- # Join the necessary tables to find the total quantity of each pizza category ordered
• 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;
```

Result Grid | Filter

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

```
68  -- ---- Determine the distribution of orders by hour of the day.  
69  
70 • SELECT  
71      HOUR(order_time) AS Hour, COUNT(order_id) AS order_count  
72  FROM  
73      orders  
74  GROUP BY HOUR(order_time);  
75
```

Result Grid | Filter

	Hour	order_count
11	1231	
12	2520	
13	2455	
14	1472	
15	1468	
16	1070	

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

- **SELECT**

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

Result Grid

	category	count
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

```
34 -- Group the orders by date and calculate the average number of Pizzas per day.  
35  
36 • SELECT  
37     ROUND(AVG(quantity), 0) as Avg_pizzaperday  
38 FROM  
39     (SELECT  
40         orders.order_date, Sum(orders_details.quantity) AS quantity  
41     FROM  
42         orders  
43     JOIN orders_details ON orders.order_id = orders_details.order_id  
44     GROUP BY orders.order_date) AS order_quantity;
```

Result Grid |

	Avg_pizzaperday
▶	138

```
96  -- Determine the top 3 most ordered pizza types based on revenue.  
97  
98 • SELECT  
99     pizza_types.name,  
100    SUM(orders_details.quantity * pizzas.price) AS revenue  
101   FROM  
102     pizza_types  
103       JOIN  
104     pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id  
105       JOIN  
106     orders_details ON orders_details.pizza_id = pizzas.pizza_id  
107   GROUP BY pizza_types.name  
108   ORDER BY revenue DESC  
109   LIMIT 3;
```

Result Grid | Filter Rows:

	name	revenue
1	The Thai Chicken Pizza	43434.25
2	The Barbecue Chicken Pizza	42768
3	The California Chicken Pizza	41409.5

```
11      -- Calculate the percentage Contribution of pizza type to total revenue
12
13 •   SELECT
14     pizza_types.category,
15     ROUND(SUM(orders_details.quantity * pizzas.price) / (SELECT
16                     ROUND(SUM(orders_details.quantity * pizzas.price),
17                           2) AS total_sales
18
19             FROM
20
21             orders_details
22             JOIN
23             pizzas ON pizzas.pizza_id = orders_details.pizza_id) * 100,
24                           2) AS revenue
25
26             FROM
27             pizza_types
28             JOIN
29             pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
30             JOIN
31             orders_details ON orders_details.pizza_id = pizzas.pizza_id
32             GROUP BY pizza_types.category
33             ORDER BY revenue DESC;
```

Result Grid | Filter

	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

```
11    -- Calculate the percentage Contribution of pizza type to total revenue
12
13 • SELECT
14     pizza_types.category,
15     ROUND(SUM(orders_details.quantity * pizzas.price) / (SELECT
16         ROUND(SUM(orders_details.quantity * pizzas.price),
17             2) AS total_sales
18     FROM
19         orders_details
20         JOIN
21             pizzas ON pizzas.pizza_id = orders_details.pizza_id) * 100,
22         2) AS revenue
23     FROM
24     pizza_types
25         JOIN
26         pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
27         JOIN
28             orders_details ON orders_details.pizza_id = pizzas.pizza_id
29     GROUP BY pizza_types.category
30     ORDER BY revenue DESC;
```

Result Grid | Filter

	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

-- -- 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(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;
```

|   | 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 | 14259.5            |

-- Determine the top 3 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((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;
```

Result Grid | Filter Rows:

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 Donnaroni Pizza	30161.75

# THANK YOU!

