



All
About
Pizza!

PIZZAHUT SALES PROJECT

Using MySQL DataBase

JYOTI MALL

ABOUT THIS PROJECT

PizzaHut is a pizza sales database project developed using MySQL to efficiently store, manage, and analyze pizza sales data. The system focuses on organizing essential information such as pizza types, categories, ingredients, orders, and sales records in a structured relational database.

This project is designed to demonstrate strong database concepts including table relationships, normalization, SQL queries, and data analysis. PizzaHum helps in tracking sales performance, identifying popular pizzas, and generating meaningful insights from stored data.

The main objective of this project is to showcase practical usage of MySQL for real-world business scenarios while ensuring data accuracy, consistency, and easy retrieval.



Retrieve the total number of order places

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

Result Grid	
	total_orders
▶	21350

Calulate the total revenue generated from pizza sales

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

Result Grid	
	total_orders
→	21350

Identify the highest pizza sales

```
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

Identify the most common pizza size ordered

```
SELECT
    pizzas.size,
    COUNT(orders_details.order_details_id) AS order_count
FROM
    pizzas
        JOIN
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC;
```

Result Grid | Filter

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

list the top 5 most ordered pizza types along with their quantity

```
SELECT
    pizza_types.name, 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.name
ORDER BY quantity DESC
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 table to find the total quantity of each pizza category order

```
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 Row

	category	quantity
▶	Classic	14888
▶	Supreme	11987
▶	Veggie	11649
▶	Chicken	11050

Determine the distribution of order 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	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336
	18	2399
	19	2009
	20	1642
	21	1198
	22	663
	23	28
	10	8
	9	1

Join relevant group the order by date and calculate the average number of pizza ordered per dayt table to find the category wise distribution of pizzas

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

	Result Grid			Filter Rows:
	avg_pizza_order_perday			
▶	138			

Join relevant table 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

Determine the top 3 most order pizza types based on revenue

- `SELECT`

```
    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.name  
ORDER BY revenue DESC  
LIMIT 3;
```

Result Grid | Filter Rows:

	name	revenue
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5

Calculate the percentage contributes of each pizza type to total revenue

```
SELECT
    pizza_types.category,
    ROUND(SUM(orders_details.quantity * pizzas.price) / (SELECT
        ROUND(SUM(orders_details.quantity * pizzas.price),
        2) AS total_sales
    )
    FROM
        orders_details
        JOIN
            pizzas ON pizzas.pizza_id = orders_details.pizza_id) * 100,
    2) 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
ORDER BY revenue DESC;
```

Result Grid | Filter Row

	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 cumulative_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;
```

Result Grid | Filter Rows:

	order_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
	2015-01-06	14358.5
	2015-01-07	16560.7
	2015-01-08	19399.05

Determine the top 3 most order 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 rank_no 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 rank_no <=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



JYOTI MALL