

PIZZA SALES ANALYSIS USING SQL



WELCOME TO

PIZZA SALES ANALYSIS PROJECT

I am Prafulla Nalawade. In this project, I performed a pizza sales analysis using MySQL queries. I imported the pizza sales and orders data into MySQL Workbench and answered multiple business questions using SQL.

@prafulla nalawade



PROBLEM STATEMENTS :

TOTAL ORDERS PLACED :

To retrieve the total number of orders placed

TOTAL REVENUE :

To calculate the total revenue generated from pizza sales.

HIGHEST PRICED PIZZA :

To identify the highest priced pizza.

Most common size pizza ordered



PROBLEM STATEMENTS :

To find the total quantity of each pizza category ordered.

Determine the distribution of orders by hour of the day.

To find the category-wise distribution of pizzas.

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

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



PROBLEM STATEMENTS :

To Calculate the percentage contribution of each pizza type to total revenue.

To Analyze the cumulative revenue generated over time.

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



TOTAL ORDERS :

```
select sum(order_id) as total_orders from orders;
```

total_orders
227921925

TOTAL REVENUE :

```
SELECT  
    round(sum((order_details.quantity * pizzas.price)),2) as total_revenue  
FROM  
    order_details  
    JOIN  
    pizzas ON order_details.pizza_id = pizzas.pizza_id ;
```

total_revenue
817860.05



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;

name	price
The Greek Pizza	35.95



MOST COMMONLY ORDERED PIZZA SIZE :

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;
```

size	order_count
L	18526



MOST ORDERED PIZZA :

SELECT

 SUM(order_details.quantity) AS quantity, pizza_types.name

FROM

 pizzas

 JOIN

 pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_i

 JOIN

 order_details ON order_details.pizza_id = pizzas.pizza_id

GROUP BY pizza_types.name

ORDER BY quantity DESC

LIMIT 5;

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



TOTAL QUANTITY OF PIZZA CATEGORY :

```
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;
```

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050



ORDERS BY HOUR OF 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



CATEGORY WISE DISTRIBUTION :

```
select category , count(name) from pizza_types  
group by category;
```

category	count(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9



AVG. PIZZA 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 ;
```

avg_pizza_ordered_per_day
130



MOST ORDERED PIZZA BASED ON REVENUE :

```
select 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.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



PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE :

```
SELECT pt.category,ROUND(SUM(od.quantity * p.price) * 100.0 /  
    (SELECT SUM(od2.quantity * p2.price)  
     FROM order_details od2  
     JOIN pizzas p2 ON p2.pizza_id = od2.pizza_id),2) AS revenue_percentage  
FROM pizza_types pt  
JOIN pizzas p ON pt.pizza_type_id = p.pizza_type_id  
JOIN order_details od ON od.pizza_id = p.pizza_id  
GROUP BY pt.category  
ORDER BY revenue_percentage DESC;
```

category	revenue_percentage
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68



CUMULATIVE REVENUE GENERATED OVER TIME :

```
select order_date,  
sum(revenue) over (order by order_date) as cumulative_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_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
2015-01-09	21526.4
2015-01-10	23899.350000000002



CONTACT DETAILS:

📞 8530689531

✉️ prafullnalawade1121@gmail.com

</> <https://github.com/prafullanlawade>

@prafulla nalawade



THANK YOU!

