

Sql Project On Pizza Sales





INTRODUCTION

Hy My Name is Tarun Here are a couple of options for a pizza sales introduction, depending on your audience and goals:

For a general audience:

Let's talk pizza! It's a global phenomenon, a delicious comfort food, and a billion-dollar industry. Today, we're diving into the world of pizza sales, exploring trends, customer preferences, and strategies to make your slice of the pie a success.

For a business audience:

In today's competitive pizza market, understanding what drives sales is crucial. We'll be analyzing data on peak periods, popular toppings and sizes, and effective marketing tactics to help you optimize your pizza business and boost your bottom line.

Some Pizza Sales Project Based Questions



Basic:-

1. Reterive The Total Number Of Order Placed
2. Calculate The Revenue Generated Of Pizza Sales
3. Identify the Higest -pizza Price
4. Identify the most Common pizza Sized Ordered
5. List The Top 5 most ordered pizza Types along their with Quantity

Intermediate:

6. Join the necessary tables to find the total quantity of each pizza category ordered.
7. Determine the distribution of orders by hour of the day.
8. Join relevant tables to find the category-wise distribution of pizzas.
9. Group the orders by date and calculate the average number of pizzas ordered per day.
10. Determine the top 3 most ordered pizza types based on revenue

Advanced:

11. Calculate the percentage contribution of each pizza type to total revenue.
12. Analyze the cumulative revenue generated over time.
13. Determine the top 3 most ordered pizza types based on revenue for each pizza category.



1. Reterive The Total Number Of Order Placed .

```
-- Retrieve the total number of orders placed.
```

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

Result Grid

	Total_orders
▶	21350

2. Calculate The Total Number Of Revenue Genrated From Pizza Sales

```
-- Calculate the total revenue generated from pizza sales.  
  
SELECT  
    ROUND(SUM(order_details.quantity * pizzas.price),  
          2) AS Total_Revenue  
FROM  
    order_details  
    JOIN  
    pizzas ON pizzas.pizza_id = order_details.pizza_id
```

Result Grid	
	Total_Revenue
▶	817860.05

3. Identify The Highest-priced pizza

3

```
-- Identify the 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;
```

Result Grid			Filter Rows:	
	name	price		
▶	The Greek Pizza	35.95		

4 Identify The Most Common Pizza Sized Ordered

```
-- 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 pizzas.pizza_id = order_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	



5. List The top 5 Most Ordered pizza types Along their Quantities

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

```
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:
	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 the total
Quantity of the each pizza Category
Ordered

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

7. Determine the Distribution of orders
by hour of the day

```
SELECT
    *
FROM
    orders;

SELECT
    HOUR(order_time) AS Hour, COUNT(order_id) order_count
FROM
    orders
GROUP BY HOUR(order_time);
```

Result Grid			Filter Rows:
	Hour	order_count	
	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	



8. Join the Revelent Table to find the category- wise distribution of the pizzas

```
#Join relevant tables 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	



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

-- Group the orders 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

avg_pizza_ordered_per_day
138



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

```
-- 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 ;
```

Result Grid			Filter Rows:
	name	revenue	
▶	The Thai Chicken Pizza	43434.25	
	The Barbecue Chicken Pizza	42768	
	The California Chicken Pizza	41409.5	

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



```
-- 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(order_details.quantity * pizzas.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 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			Filter Rows
	category	revenue	
▶	Classic	26.91	
	Supreme	25.46	
	Chicken	23.96	
	Veggie	23.68	



12. Analyze the cumulative revenue generated over time.

```
-- 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;
```

Result Grid  Filter Rows: 		
	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
	2015-01-14	32358.700000000004
	2015-01-15	34343.500000000001

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

```
-- 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;
```

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 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 PLZ LIKE
AND SHARE**