

PIZZA TIME

WELCOME TO THE PIZZA SALES REPORT PROJECT, WHERE WE LEVERAGE SQL FOR DATA ANALYSIS AND POWER BI FOR CREATING INTERACTIVE VISUALIZATIONS.

THIS PROJECT PROVIDES IN-DEPTH INSIGHTS INTO PIZZA SALES DATA, INCLUDING DAILY AND MONTHLY TRENDS, SALES BY PIZZA CATEGORY AND SIZE, AND IDENTIFYING TOP-SELLING PIZZAS BASED ON REVENUE, QUANTITY, AND TOTAL ORDERS.

ORDER NOW

FREE DELIVERY :

+123-456-7890



PIZZA SALES REPORT PROJECT, WHERE WE LEVERAGE SQL FOR DATA ANALYSIS AND POWER BI FOR CREATING INTERACTIVE VISUALIZATIONS. THIS PROJECT PROVIDES IN-DEPTH INSIGHTS INTO PIZZA SALES DATA, INCLUDING DAILY AND MONTHLY TRENDS, SALES BY PIZZA CATEGORY AND SIZE, AND IDENTIFYING TOP-SELLING PIZZAS BASED ON REVENUE, QUANTITY, AND TOTAL ORDERS.

YOUR PARAGRAPH TEXT

1-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 revenue generated from pizza sales.

```
select sum(quantity*price)as revenue  
from pizzas  
join order_details on  
pizzas.pizza_id=order_details.pizza_id
```

Result Grid	
	revenue
▶	817860.049999993

3-Identify the highest-priced pizza.

```
select * from pizzas
join pizza_types on
pizzas.pizza_type_id=pizza_types.pizza_type
_id
order by price desc
limit 1
```

pizza_type_id	size	price	pizza_type_id	name	category	in
the_greek	XXL	35.95	the_greek	The Greek Pizza	Classic	Ka

4-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		
	size	order_count
▶	L	18526
	M	15385
	S	14137
	XL	544
	XXL	28

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

```
select sum(order_details.quantity)as  
order_quantity, pizza_types.name from  
pizza_types  
join pizzas on  
pizza_types.pizza_type_id=pizzas.pizza_type  
_id  
join order_details on  
pizzas.pizza_id=order_details.pizza_id  
group by pizza_types.name  
order by order_quantity desc  
limit 5
```

	order_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

6-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;
```

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

7-Determine the distribution of orders by hour of the day.

```
SELECT HOUR(time),  
COUNT(orders.order_id)  
FROM orders  
GROUP BY HOUR(time)
```

	HOUR(time)	COUNT(orders.order_id)
	11	1231
	12	2520
	13	2455
	14	1472
	15	1468

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

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

	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.

```
select avg(quantity) from  
(SELECT orders.date,  
SUM(order_details.quantity) as quantity  
FROM orders JOIN  
order_details ON orders.order_id =  
order_details.order_id  
GROUP BY orders.date) as order_quantity
```

Result Grid	
	avg(quantity)
▶	138.4749

10-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 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

11-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;
```

	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

12-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 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.date) as sales;
```

	date	cumulative_revenue
▶	2015-01-01	2713.8500000000004
	2015-01-02	5445.75
	2015-01-03	8108.15

13-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  
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 a  
) as b  
where rn<=3
```