

*Hello everyone, my name is
RISHABH DIXIT and i'm working
on this project to solve queries in
SQL for pizza sales analysis.*

PIZZA SALES



INTRODUCTION

PIZZA IS ONE OF THE MOST POPULAR AND WIDELY CONSUMED FOODS AROUND THE WORLD.

THIS REPORT AIMS TO EXPLORE THE CURRENT TRENDS IN PIZZA SALES, ANALYZE THE KEY DRIVERS BEHIND ITS MARKET GROWTH.

BY UNDERSTANDING THESE DYNAMICS, BUSINESSES CAN BETTER NAVIGATE THE COMPETITIVE LANDSCAPE AND CATER TO THE EVER-CHANGING DEMANDS OF PIZZA LOVERS.



RETRIEVE THE TOTAL NUMBER OF
ORDERS PLACED.

SELECT

COUNT(order_id) **AS** Total_orders

FROM

orders;

	Total_orders
▶	21350



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 order_details.pizza_id = pizzas.pizza_id;
```

	Total_Revenue
▶	817860.05



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;
```

	name	price
▶	The Greek Pizza	35.95



IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```
SELECT
    SUM(order_details.quantity), pizzas.size
FROM
    order_details
    JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizzas.size;
```

	sum(order_details.quantity)	size
►	15635	M
	18956	L
	14403	S
	552	XL
	28	XXL



LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

```
SELECT
    pizza_types.name, SUM(order_details.quantity) as Total_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 Total_Quantity desc
LIMIT 5;
```

	name	Total_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(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



DETERMINE THE DISTRIBUTION OF ORDERS 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



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



GROUP THE ORDERS BY DATE AND CALCULATE THE
AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.

SELECT

ROUND(AVG(quantity), 0)

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;

round(avg(quantity),0)



138



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



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 order_details.pizza_id = pizzas.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



ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```
select order_date, round(sum(revenue) over(order by order_date),2) as cum_sales
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	cum_sales
►	2015-01-01	2713.85
	2015-01-02	5445.75
	2015-01-03	8108.15
	2015-01-04	9863.6
	2015-01-05	11929.55



THANK YOU

