



**HELLO, MY NAME IS ABHAY KUMAR. IN THIS PROJECT, I HAVE DEVELOPED SQL QUERIES TO ADDRESS VARIOUS ASPECTS OF PIZZA SALES BUSINESS ANALYTICS...**



## ANALYSIS QUESTION.

1. RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.
2. CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.
3. IDENTIFY THE HIGHEST-PRICED PIZZA.
4. IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.
5. LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.
6. JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.
7. GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.
8. DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.



## 1.RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

```
1  -- --Retrieve the total number of orders placed.  
2  
3 • select COUNT(ORDER_ID) AS TOTAL_ORDERS from orders;  
4
```

	TOTAL_ORDERS
▶	21350



## 2.CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```
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
```

	TOTAL_SALES
▶	817860.05



### 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 price DESC
LIMIT 1
```

	name	price
▶	The Greek Pizza	35.95



## 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;
```

size	order_count
L	18526
M	15385
S	14137



## 5. 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;
```

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 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



## 7. 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;
```

	avg(quantity)
▶	138.4749



## 8.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



## CONCLUSION:

THROUGH THIS PROJECT, COMPREHENSIVE SQL QUERIES WERE CONSTRUCTED TO EXTRACT VALUABLE INSIGHTS FROM THE PIZZA SALES DATA. THE ANALYSIS PROVIDED A CLEAR UNDERSTANDING OF SALES PERFORMANCE, CUSTOMER PREFERENCES, AND REVENUE GENERATION. THESE INSIGHTS ARE INSTRUMENTAL FOR STRATEGIC DECISION-MAKING, HELPING TO OPTIMIZE INVENTORY, ENHANCE MARKETING STRATEGIES, AND ULTIMATELY DRIVE BUSINESS GROWTH.

**THANKYOU.....**