



PIZZA SALES  
**SQL PROJECT**  
DATA ANALYTICS

BY  
KABIR SHUKLA

# ABOUT THE PROJECT

# PIZZA SALE ANALYSIS

## **Objective:**

Perform data analysis on pizza place sales to answer specific business-related questions.

## **Tools Used:**

SQL for querying and data extraction.

**Key Areas of Focus:** Sales trends, pizza popularity, revenue breakdown, and order distribution.

## **Challenges Solved:**

Identified top-selling pizzas.

Analyzed revenue contribution by pizza types.

Investigated ordering patterns based on time and category.

BY

KABIR SHUKLA

# ABOUT THE PROJECT DATASET & SCHEMA

## **Data Structure:**

The dataset is split into 4 CSV files representing different aspects of sales and inventory.

## **Orders Table:**

Contains order IDs, date/time of orders, and customer information.

## **Pizzas Table:**

Includes pizza types, sizes, and their prices.

## **Order Details Table:**

Connects orders with specific pizzas (order ID, pizza type, quantity).

## **Sales Table:**

Tracks revenue, payment methods, and total order amounts.

BY

KABIR SHUKLA

## RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

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

Result Grid	
	total_orders
▶	21350

BY

KABIR SHUKLA

# 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

BY

KABIR SHUKLA

# 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

BY

KABIR SHUKLA

# 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

BY

KABIR SHUKLA

# 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		
	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

BY

KABIR SHUKLA

# 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

BY

KABIR SHUKLA

# DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

```
SELECT  
    HOUR(order_time), COUNT(order_id)  
FROM  
    orders  
GROUP BY HOUR(order_time);|
```

	hour(order_time)	count(order_id)
▶	11	1231
	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

BY

KABIR SHUKLA

# JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.

**SELECT**

```
category, COUNT(name)  
FROM  
pizza_types  
GROUP BY category;
```

Result Grid | Filter Row

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

BY

KABIR SHUKLA

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

```
SELECT
    ROUND(AVG(quantity), 0) AS average_daily_orders
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 orders_per_day
```

	Result Grid	Filter
	average_daily_orders	
▶	138	

BY

KABIR SHUKLA

# 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

BY

KABIR SHUKLA

# 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_per_pizza
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_per_pizza DESC;
```

BY

KABIR SHUKLA

# ANALYZE THE 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	order_date	cumulative_revenue	order_date	cumulative_revenue
2015-01-18	40978.600000000006	2015-02-04	80159.80000000002	2015-02-21	119009.70000000001
2015-01-19	43365.750000000001	2015-02-05	82375.60000000002	2015-02-22	120589.65000000001
2015-01-20	45763.650000000001	2015-02-06	84885.55000000002	2015-02-23	122758.20000000001
2015-01-21	47804.200000000001	2015-02-07	87123.20000000001	2015-02-24	124952.75000000001
2015-01-22	50300.900000000001	2015-02-08	89158.20000000001	2015-02-25	127294.05000000002
2015-01-23	52724.600000000006	2015-02-09	91353.55000000002	2015-02-26	129555.35000000002
2015-01-24	55013.850000000006	2015-02-10	93410.05000000002	2015-02-27	132413.30000000002
2015-01-25	56631.400000000001	2015-02-11	95870.05000000002	2015-02-28	134952.90000000002
2015-01-26	58515.800000000001	2015-02-12	98028.85000000002	2015-03-01	136551.45
2015-01-27	61043.850000000001	2015-02-13	100783.35000000002	2015-03-02	138930.5
2015-01-28	63059.850000000001	2015-02-14	103102.50000000001	2015-03-03	141218.4
2015-01-29	65105.150000000016	2015-02-15	105243.75000000001	2015-03-04	143662.69999999998
2015-01-30	67375.450000000001	2015-02-16	107212.55000000002	2015-03-05	146013.34999999998
2015-01-31	69793.300000000002	2015-02-17	109334.45000000001	2015-03-06	148527.3
2015-02-01	72982.500000000001	2015-02-18	111977.30000000002	2015-03-07	150927.75
2015-02-02	75311.100000000002	2015-02-19	114007.55000000002	2015-03-08	153115.9
2015-02-03	77925.900000000002	2015-02-20	116898.70000000001	2015-03-09	155450.44999999998

*and so on...*

BY

KABIR SHUKLA

# DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.

```
select category, 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;
```

	category	name	revenue
▶	Chicken	The Thai Chicken Pizza	43434.25
	Chicken	The Barbecue Chicken Pizza	42768
	Chicken	The California Chicken Pizza	41409.5
	Classic	The Classic Deluxe Pizza	38180.5
	Classic	The Hawaiian Pizza	32273.25
	Classic	The Pepperoni Pizza	30161.75
	Supreme	The Spicy Italian Pizza	34831.25
	Supreme	The Italian Supreme Pizza	33476.75
	Supreme	The Sicilian Pizza	30940.5
	Veggie	The Four Cheese Pizza	32265.70000000065
	Veggie	The Mexicana Pizza	26780.75
	Veggie	The Five Cheese Pizza	26066.5

BY

KABIR SHUKLA



BY

KABIR SHUKLA



THANK  
YOU

BY  
KABIR SHUKLA