

PIZZAS SALES (SQL)

-SHRUNKHALA SISODIA



INTRODUCTION

I am Shrunkhala Sisodia A final year student of MSC-IT specialization in Business Intelligence and Analytics. I have created a project on Pizzas Sales using SQL.

This project explores pizza sales data using SQL, analyzing orders, order details, pizza types, and pizzas. By leveraging structured queries, it provides insights into sales trends, customer preferences, and business performance.

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

```
select count(order_id) as total_orders from orders;
```

Result Grid		Filter
	total_orders	
▶	21350	

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```
• select  
  sum(orders_details.quantity * pizzas.price) as total_revenue  
  from orders_details join pizzas  
  on orders_details.pizza_id = pizzas.pizza_id;
```

Result Grid	
	total_revenue
▶	817860.049999993

IDENTIFY THE HIGHEST-PRICED PIZZA.



```
select max(price) from pizzas;
```

Result Grid | Filter Rows:

	max(price)
▶	35.95

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```
• SELECT  
    pizzas.size,  
    COUNT(orders_details.order_details_id) AS total_orders  
FROM  
    pizzas  
    JOIN  
    orders_details ON pizzas.pizza_id = orders_details.pizza_id  
GROUP BY pizzas.size  
ORDER BY total_orders DESC  
limit 1;
```

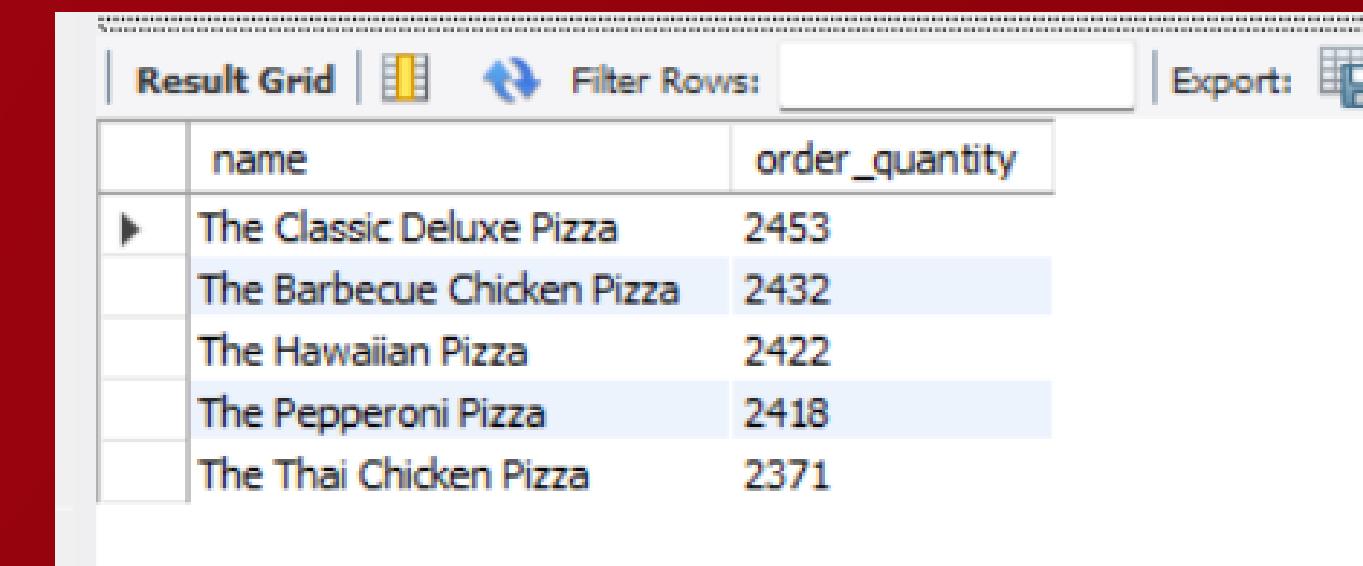
Result Grid | Filter Rows:

	size	total_orders
▶	L	18526

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

- **SELECT**

```
  pizza_types.name,  
  sum(orders_details.quantity) AS order_quantity  
FROM  
  pizza_types  
  JOIN  
  pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
  JOIN  
  orders_details ON pizzas.pizza_id = orders_details.pizza_id  
GROUP BY pizza_types.name  
ORDER BY order_quantity DESC  
LIMIT 5;
```



The screenshot shows a MySQL Workbench result grid with the following data:

	name	order_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(orders_details.quantity) AS quantity  
FROM  
  pizza_types  
  JOIN  
  pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
  JOIN  
  orders_details ON pizzas.pizza_id = orders_details.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(orders.order_time),  
COUNT(orders.order_id) AS order_count  
FROM  
orders  
GROUP BY HOUR(orders.order_time);
```

-- 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

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

```
88 •  SELECT
89      pizza_types.name,
90      SUM(pizzas.price * orders_details.quantity) AS revenue
91  FROM
92      pizza_types
93      JOIN
94          pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
95      JOIN
96          orders_details ON pizzas.pizza_id = orders_details.pizza_id
97  GROUP BY pizza_types.name
98  ORDER BY revenue DESC
99  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

CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.

```
SELECT
    pizza_types.category,
    (SUM(pizzas.price * orders_details.quantity) / (SELECT
        SUM(orders_details.quantity * pizzas.price) AS Total_scales
    FROM
        orders_details
        JOIN
        pizzas ON pizzas.pizza_id = orders_details.pizza_id)) * 100 as revenue
FROM
    pizza_types
    JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
    JOIN
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY pizza_types.category
ORDER BY revenue DESC;
```

Result Grid | Filter Rows:

	category	revenue
▶	Classic	26.905960255669903
	Supreme	25.45631126009884
	Chicken	23.955137556847493
	Veggie	23.682590927384783

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

```
• SELECT
  AVG(avg_pizza_order)
  FROM
  (SELECT
    DATE(orders.order_date),
    SUM(orders_details.quantity) AS avg_pizza_order
  FROM
    orders
  JOIN orders_details ON orders.order_id = orders_details.order_id
  GROUP BY DATE(orders.order_date)) AS order_quatity;
```

	AVG(avg_pizza_order)
▶	138.4749

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```
select order_date ,  
round(sum(revenue) over (order by order_date),2 )as Cum_revenue  
from  
(select orders.order_date,  
sum(orders_details.quantity * pizzas.price) as revenue  
from orders_details join pizzas  
on orders_details.pizza_id = pizzas.pizza_id  
join orders  
on orders.order_id = orders_details.order_id  
group by orders.order_date) as total_sales;
```

	order_date	Cum_revenue
▶	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
	2015-01-06	14358.5
	2015-01-07	16560.7
	2015-01-08	19399.05
	2015-01-09	21526.4
	2015-01-10	23990.35
	2015-01-11	25862.65
	2015-01-12	27781.7
	2015-01-13	29831.3
	2015-01-14	32358.7
	2015-01-15	34343.5
	2015-01-16	36937.65
	2015-01-17	39001.75
	2015-01-18	40978.6
	2015-01-19	43365.75
	2015-01-20	45763.65

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

```
6 • select name, revenue from
7   (select category, name, revenue,
8    rank() over (partition by category order by revenue desc) as rn
9    from
10   (select pizza_types.category, pizza_types.name,
11    sum(orders_details.quantity * pizzas.price) as revenue
12    from pizza_types join pizzas
13    on pizza_types.pizza_type_id = pizzas.pizza_type_id
14    join orders_details on orders_details.pizza_id = pizzas.pizza_id
15    group by pizza_types.category, pizza_types.name ) as a ) as b
16   where rn<=3
```

	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!

