

MASTERING SQL : ANALYZING 100,000+ PIZZA ORDERS FOR BUSINESS INSIGHTS

Overview :

SQL-Driven Analysis of Large-Scale Pizza Orders

Uncovering Business Insights from 100,000+ Records

SQL-powered pizza sales analysis.

Covering simple to advanced data queries.

Strategic insights for business enhancement.

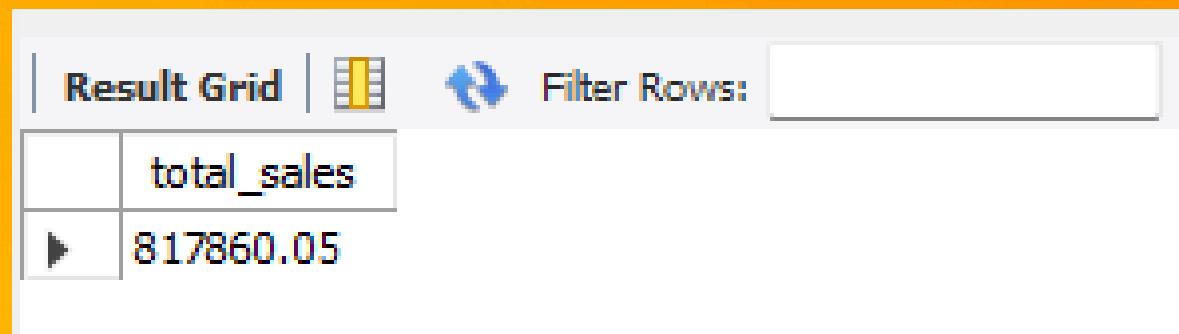




Calculate the total revenue generated from pizza sales.



```
SELECT  
    ROUND(SUM(pz.price * od.quantity), 2) AS total_sales  
FROM  
    pizzas AS pz  
        LEFT JOIN  
    order_details AS od ON pz.pizza_id = od.pizza_id;
```



	total_sales
▶	817860.05

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 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 quantity DESC  
LIMIT 5;
```

Result Grid		Filter Rows:
	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

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
	XL	544
	XXL	28

Identify the highest-priced pizza.

```
SELECT  
    pizza_types.name, pizzas.price  
FROM  
    pizza_types  
        JOIN  
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id  
ORDER BY pizzas.price DESC  
LIMIT 1;
```

	name	price
	The Greek Pizza	25.5
	The Brie Carre Pizza	23.65
	The Italian Vegetables Pizza	21
	The Barbecue Chicken Pizza	20.75
	The Spinach Supreme Pizza	20.75
	The Italian Supreme Pizza	20.75
	The California Chicken Pizza	20.75

Join the necessary tables to find the total quantity of each pizza category ordered.

```
SELECT  
    pizza_types.category,  
    SUM(order_details.quantity) AS total_quantity  
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  
ORDER BY total_quantity DESC;
```

	category	total_quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050



Retrieve the total number of orders placed.



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

Result Grid				Filter Rows:
COUNT(order_id)				
▶ 21350				

Determine the distribution of orders by hour of the day.

SELECT

HOUR(orders.order_time) AS hours,

COUNT(order_id) AS order_count

FROM

orders

GROUP BY HOUR(order_time);

hours	order_count
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

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

```
SELECT
```

```
    pizza_types.category, COUNT(name)
```

```
FROM
```

```
    pizza_types
```

```
GROUP BY pizza_types.category;
```

Result Grid | Filter Rows:

	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) AS Avg_Order_PerDay  
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 order_date) AS order_quantity;
```

	Avg_Order_PerDay
▶	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 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;
```

	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(pz.price * od.quantity), 2) AS total_revenue
    FROM
        pizzas AS pz
        LEFT JOIN
        order_details AS od ON pz.pizza_id = od.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,  
       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 orders ON order_details.order_id = orders.order_id  
     JOIN pizzas ON pizzas.pizza_id = order_details.pizza_id  
    GROUP BY orders.order_date  
  ) AS revenue_data;
```

	order_date	cumulative_revenue
2015-01-01	2713.8500000000004	
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	23990.350000000002	
2015-01-11	25862.65	
2015-01-12	27781.7	
2015-01-13	29831.300000000003	

Determine the top 3 most ordered pizza types based on revenue for each pizza category.

```
SELECT category, name, revenue, rev
FROM (
    SELECT pizza_types.category, pizza_types.name,
           SUM(order_details.quantity * pizzas.price) AS revenue,
           RANK() OVER (PARTITION BY pizza_types.category ORDER BY
                         SUM(order_details.quantity * pizzas.price) DESC) AS rev
    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 ranked_pizzas
where rev<=3
```

category	name	revenue	rev
Chicken	The Thai Chicken Pizza	43434.25	1
Chicken	The Barbecue Chicken Pizza	42768	2
Chicken	The California Chicken Pizza	41409.5	3
Classic	The Classic Deluxe Pizza	38180.5	1
Classic	The Hawaiian Pizza	32273.25	2
Classic	The Pepperoni Pizza	30161.75	3
Supreme	The Spicy Italian Pizza	34831.25	1
Supreme	The Italian Supreme Pizza	33476.75	2
Supreme	The Sicilian Pizza	30940.5	3
Veggie	The Four Cheese Pizza	32265.7000000065	1
Veggie	The Mexicana Pizza	26780.75	2
Veggie	The Five Cheese Pizza	26066.5	3



CONCLUSION

The SQL-driven analysis of over 100,000 pizza orders has provided me with actionable insights into PizzaCo's sales dynamics. By leveraging SQL queries, I uncovered key trends and patterns that offer a clear understanding of customer preferences, peak ordering times, and revenue distribution etc.

Key Takeaways:

- Enhanced Profitability: Identified high-performing pizza types and optimal ordering times to refine product offerings and marketing strategies.
- Data-Driven Decisions: Gained insights that can inform inventory management and operational improvements.
- Strategic Recommendations: Developed actionable strategies for boosting sales and increasing overall profitability.

