

Pizza Sales

Analysis Using SQL

Introduction

This project involved a comprehensive data analysis using SQL to extract, manipulate, and uncover insights from diverse datasets. By addressing key questions, I aimed to identify patterns, trends, and relationships within the data. The analysis leveraged multiple data sources, each providing unique value to the overall understanding.

List the top 5 most ordered pizza types along with their quantities.

```
SELECT
    pizza_types.name, 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 orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT 5;
```

Result Grid | Filter Rows: Export: Wrap Cell Content: Fetch 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

Calculate the total revenue generated from pizza sales.

```
SELECT  
    ROUND(SUM(orders_details.quantity * pizzas.price),  
        2) AS total_sales  
FROM  
    orders_details  
        JOIN  
    pizzas ON pizzas.pizza_id = orders_details.pizza_id
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
total_sales				
817860.05				

Identify the 5 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 5;
```

Result Grid		
	name	price
▶	The Greek Pizza	35.95
	The Greek Pizza	25.5
	The Brie Carre Pizza	23.65
	The Italian Vegetables Pizza	21
	The Barbecue Chicken Pizza	20.75

Identify the most common pizza size ordered

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

Result Grid | Filter Rows: Export: Wrap Cell Content:

size	order_count
L	18526
M	15385
S	14137
XL	544
XXL	28

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 orders_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

Group the orders by date and calculate the average number of pizzas ordered per day.

```
SELECT  
    ROUND(AVG(quantity), 0) as avg_pizza_ordered_per_day  
FROM  
    (SELECT  
        orders.order_date, SUM(orders_details.quantity) AS quantity  
    FROM  
        orders  
    JOIN orders_details ON orders.order_id = orders_details.order_id  
    GROUP BY orders.order_date) AS order_quantity;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	avg_pizza_ordered_per_day			
▶	138			

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
	16	1920
	17	2336
	18	2399
	19	2009
	20	1642

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

```
select pizza_types.name,  
sum(orders_details.quantity * pizzas.price) as revenue  
from pizza_types join pizzas  
on pizzas.pizza_type_id = pizza_types.pizza_type_id  
join orders_details  
on orders_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(orders_details.quantity * pizzas.price) / (SELECT  
                           ROUND(SUM(orders_details.quantity * pizzas.price),  
                                 2) AS total_sales  
FROM  
       orders_details  
      JOIN  
       pizzas ON pizzas.pizza_id = orders_details.pizza_id) *100,2) as revenue  
from pizza_types join pizzas  
on pizza_types.pizza_type_id = pizzas.pizza_type_id  
join orders_details  
on orders_details.pizza_id = pizzas.pizza_id  
group by pizza_types.category order by revenue desc;
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

	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 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 sales;
```

order_date	cum_revenue
2015-01-01	2713.850000000004
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	23990.35000000002
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.3000000003
2015-01-14	32358.7000000004
2015-01-15	34343.5000000001
2015-01-16	36937.6500000001
2015-01-17	39001.7500000001
2015-01-18	40978.6000000006
2015-01-19	43365.7500000001
2015-01-20	45763.6500000001
2015-01-21	47804.2000000001
2015-01-22	50300.9000000001
2015-01-23	52724.6000000006
2015-01-24	55013.8500000006

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

```
select 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((orders_details.quantity) * pizzas.price) as revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join orders_details
on orders_details.pizza_id = pizzas.pizza_id
group by pizza_types.category, pizza_types.name) as a) as b
where rn <=3;
```

Result Grid		Filter Rows:
	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	32265.7000000065
	The Mexicana Pizza	26780.75
	The Five Cheese Pizza	26066.5

Conclusion

This analysis shows us how our pizza business is doing. We found out how much money we made, what sizes and types of pizza are most popular, and how many pizzas we sell each day. We also learned which pizzas make us the most money. This information can help us decide what pizzas to sell more of and how to price them.