

# PIZZA SALES ANALYSIS

WHERE EVERY SLICE TELLS A STORY



Data Analyst





# USING SQL TO ANALYZE SALES DATA

# DATA USED

About

Contact

I analyzed pizza sales using SQL queries on the following tables:

- 📝 orders → Contains all orders placed
- 🍕 pizzas → All prizes and size
- 📦 order\_details → Tracks quantity
- 📄 pizza\_types → Categorizes pizzas

# RETRIVE THE TOTAL NUMBER OF ORDERS PLACED

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

	Total_orders
21350	

# CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES

[About](#)[Contact](#)

**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

# IDENTIFY THE HIGHEST PRICED PIZZA

[About](#)[Contact](#)

```
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

# IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED

[About](#)[Contact](#)

```
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

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

[About](#)[Contact](#)

```
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 | Filter Rows:

	name	Quantity
▶	The Classic Delu...	2453
▶	The Barbecue C...	2432
▶	The Hawaiian Pizza	2422
▶	The Pepperoni Pi...	2418
▶	The Thai Chicke...	2371

# JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED

Contact

```
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	14388
	Supreme	11987
	Veggie	11649
	Chicken	11050



# DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY

[About](#)[Contact](#)

```
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
	21	1198
	22	663
	23	28
	10	8
	9	1



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

About  
Contact

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

Result Grid | Filter R

	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_pizza_ordered_per_day  
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 Order_quantity;
```

	AVG_pizza_ordered_per_day
▶	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	43434.25
▶	The Barbecue Chicken	42768
▶	The California Chicken	41409.5

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

Contact



```
SELECT
    pizza_types.category,
    ROUND(SUM(order_details.quantity * pizzas.price) / (SELECT
        ROUND(SUM(order_details.quantity * pizzas.price),
        2))
    )
FROM
    order_details
    JOIN
        pizzas ON pizzas.pizza_id = order_details.pizza_id) * 100,
    2) AS revenue_percentage
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_percentage DESC;
```

	category	revenue_percentage
	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68



# ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME

[About](#)[Contact](#)

```
SELECT order_date,  
       sum(revenue) over(order by order_date) as cum_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	cum_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

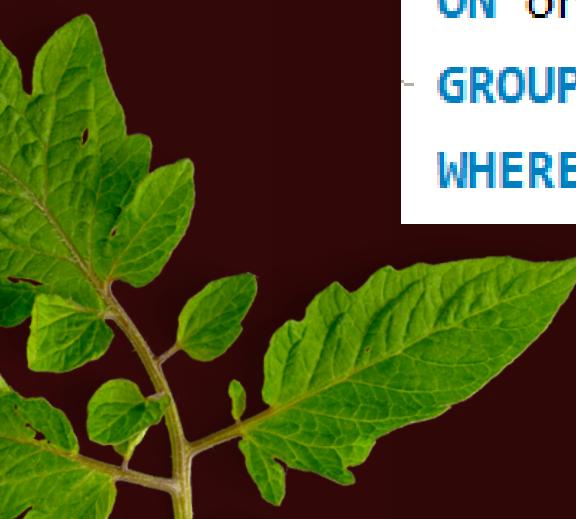


# DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY

Contact

```
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((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;
```

	name	revenue
>	The Thai Chicke...	43434.25
	The Barbecue C...	42768
	The California C...	41409.5
	The Classic Delu...	38180.5
	The Hawaiian Pizza	32273.25
	The Pepperoni Pi...	30161.75
	The Spicy Italian...	34831.25
	The Italian Supr...	33476.75
	The Sicilian Pizza	30940.5
	The Four Chees...	32265.70000000065
	The Mexicana Pi...	26780.75
	The Five Cheese...	26066.5



# CONCLUSION & KEY INSIGHTS

About

Contact



**Best-Selling Pizza: The Classic Deluxe Pizza**



**Most Popular Pizza Category: Classic**



**Lowest-Selling Pizzas: The Brie Carre Pizza**



**Total Revenue Generated: 817860.05**



**Category with the Lowest Sales: Veggie**

# BUSINESS RECOMMENDATIONS

Home

About



## 🔧 Increase Promotions for Low-Selling Pizzas

- ✓ Offer discounts & combo deals to boost sales
- ✓ Highlight veg pizzas in marketing campaigns

## 🔍 Optimize Menu Pricing

- ✓ Adjust prices for low-demand pizzas to increase sales

## 📅 Peak Sales Timing Analysis

- ✓ Analyze when customers order most & offer targeted discounts



Home

# THANK YOU

CONNECT WITH ME FOR  
MORE DATA INSIGHTS!

🔗 Connect with me on LinkedIn

