Where Every Slice is a Taste of Perfection

# PIZZA SALES ANALYSIS

ORDER NOW

Start Your Slide

# ABOUT THE PROJECT

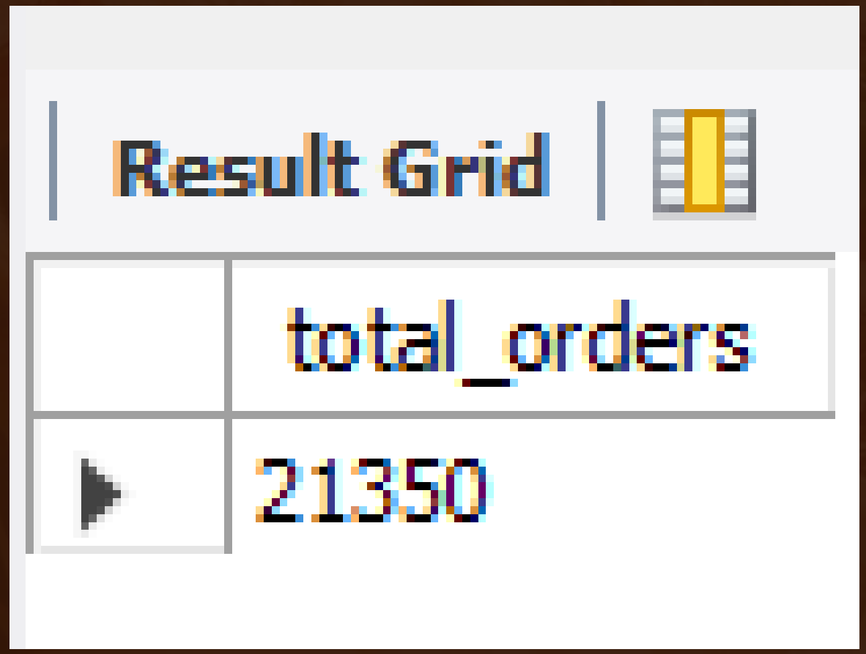
## Our Passion for Pizza

The goal of this Pizza Sales Analysis is to uncover key insights into sales performance, customer preferences, and revenue distribution.

The analysis involves calculating total orders, revenue, and identifying the most popular pizza types and sizes. as well as examining order patterns by time, category-wise distribution, and daily sales trends.

# GET THE TOTAL NUMBER OF ORDERS PLACED

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



# CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES

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

Result Grid

| | total_revenue |
|---|---|
| ▶ | 817860.05 |

# IDENTIFY THE HIGHEST-PRICED PIZZA

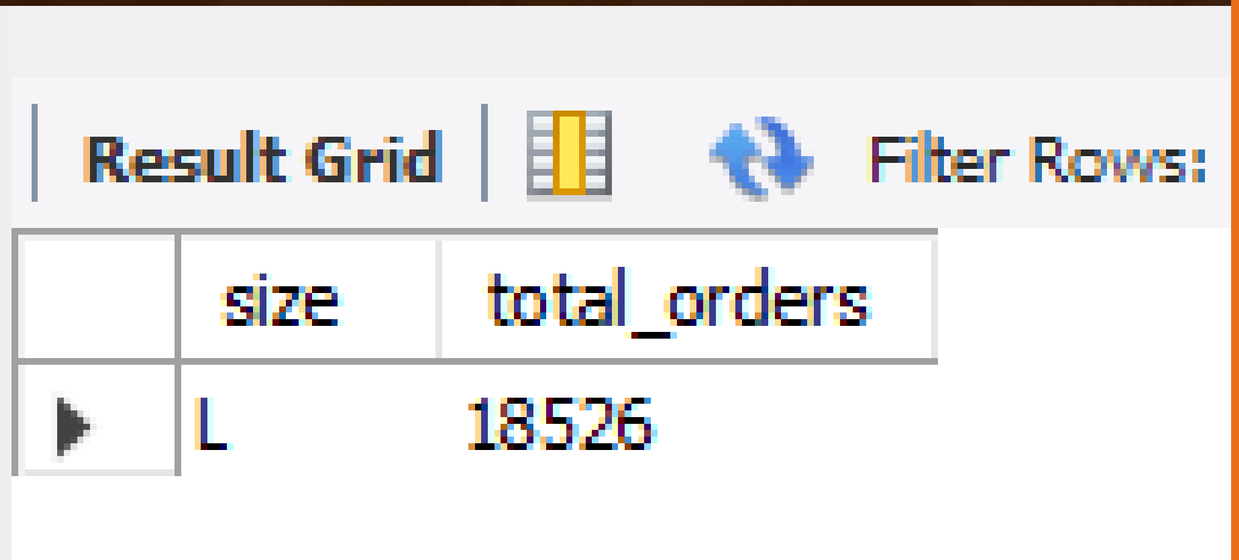
```sql
SELECT pt.name, p.price
FROM pizzas p
JOIN pizza_types pt
ON p.pizza_type_id = pt.pizza_type_id
ORDER BY 2 DESC
LIMIT 1;
```

Result Grid | Filter Rows:

| name | price |
|---|---|
| The Greek Pizza | 35.95 |

# IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED

```sql
SELECT p.size, COUNT(od.order_details_id) AS total_orders FROM order_details od
JOIN pizzas p ON od.pizza_id = p.pizza_id
GROUP BY p.size
ORDER BY 2 DESC
LIMIT 1;
```




| size | total_orders |
|---|---|
| L | 18526 |

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

```
SELECT pt.name, SUM(od.quantity) AS total_quantity  , COUNT(od.order_details_id) AS times_ordered FROM order_details od
JOIN pizzas p ON od.pizza_id = p.pizza_id
JOIN pizza_types pt ON p.pizza_type_id = pt.pizza_type_id
GROUP BY pt.name
ORDER BY 2 DESC
LIMIT 5;
```

Result Grid | Filter Rows: | Export:

| name | total_quantity | times_ordered |
|---|---|---|
| The Classic Deluxe Pizza | 2453 | 2416 |
| The Barbecue Chicken Pizza | 2432 | 2372 |
| The Hawaiian Pizza | 2422 | 2370 |
| The Pepperoni Pizza | 2418 | 2369 |
| The Thai Chicken Pizza | 2371 | 2315 |

# FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED

```sql
SELECT pt.category, SUM(od.quantity) AS total_quantity FROM order_details od
JOIN pizzas p ON od.pizza_id = p.pizza_id
JOIN pizza_types pt ON p.pizza_type_id = pt.pizza_type_id
GROUP BY 1
ORDER BY 2 DESC;
```

Result Grid   Filter Rows:

| category | total_quantity |
|---|---|
| Classic | 14888 |
| Supreme | 11987 |
| Veggie | 11649 |
| Chicken | 11050 |

# FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED

```sql
SELECT pt.category, SUM(od.quantity) AS total_quantity FROM order_details od
JOIN pizzas p ON od.pizza_id = p.pizza_id
JOIN pizza_types pt ON p.pizza_type_id = pt.pizza_type_id
GROUP BY 1
ORDER BY 2 DESC;
```

Result Grid   Filter Rows:

| category | total_quantity |
|---|---|
| Classic | 14888 |
| Supreme | 11987 |
| Veggie | 11649 |
| Chicken | 11050 |

# DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY (TOP 5)

```sql
SELECT HOUR(time), COUNT(order_id) AS number_of_orders
FROM orders
GROUP BY 1
ORDER BY 2 DESC
LIMIT 5;
```

Result Grid | Filter Rows:

| HOUR(time) | number_of_orders |
|---|---|
| 12 | 2520 |
| 13 | 2455 |
| 18 | 2399 |
| 17 | 2336 |
| 19 | 2009 |

# FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS

```sql
SELECT category, COUNT(name) AS total_number
FROM pizza_types
GROUP BY category
ORDER BY 2 DESC;
```

Result Grid | Filter Rows:

| category | total_number |
|---|---|
| Supreme | 9 |
| Veggie | 9 |
| Classic | 8 |
| Chicken | 6 |

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

```sql
SELECT ROUND(AVG(avg_pizza_ordered),0) AS Avg_orders FROM
(
SELECT o.date, SUM(od.quantity) AS avg_pizza_ordered FROM orders o
JOIN order_details od ON o.order_id = od.order_id
GROUP BY 1
ORDER BY 2 DESC) AS total_order_per_day
;
```

Result Grid

| | Avg_orders |
|---|---|
| ▶ | 138 |

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

```sql
SELECT pt.name, SUM(od.quantity*p.price) AS revenue FROM pizza_types pt
JOIN pizzas p ON pt.pizza_type_id = p.pizza_type_id
JOIN order_details od ON p.pizza_id = od.pizza_id
GROUP BY pt.name
ORDER BY 2 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 |

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

```sql
SELECT pt.category, ROUND(SUM(od.quantity*p.price) / (SELECT ROUND(SUM(order_details.quantity * pizzas.price),2) AS total_sales
FROM order_details
JOIN pizzas ON order_details.pizza_id = pizzas.pizza_id) * 100,2)  AS revenue FROM pizza_types pt
JOIN pizzas p ON pt.pizza_type_id = p.pizza_type_id
JOIN order_details od ON p.pizza_id = od.pizza_id
GROUP BY pt.category
ORDER BY 2 DESC;
```

Result Grid | Filter Rows:

| category | revenue |
|---|---|
| Classic | 26.91 |
| Supreme | 25.46 |
| Chicken | 23.96 |
| Veggie | 23.68 |

# ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME

```sql
SELECT revenue.date,ROUND(SUM(total_price) OVER(order by revenue.date),2) AS cummulative_revenue
FROM
(SELECT o.date, SUM(od.quantity * p.price) AS total_price
FROM order_details od
JOIN pizzas p ON p.pizza_id = od.pizza_id
JOIN orders o ON o.order_id = od.order_id
GROUP BY o.date) AS revenue;
```

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

```sql
SELECT category, name, revenue, ranking
FROM (
    SELECT
        pt.category,
        pt.name,
        SUM(od.quantity * p.price) AS revenue,
        RANK() OVER (PARTITION BY pt.category ORDER BY SUM(od.quantity * p.price) DESC) AS ranking
    FROM pizza_types pt
    JOIN pizzas p ON pt.pizza_type_id = p.pizza_type_id
    JOIN order_details od ON od.pizza_id = p.pizza_id
    GROUP BY pt.category, pt.name
) AS ranked_pizzas
WHERE ranking <= 3;
```

Nusan Dubey

# THANK YOU FOR ATTENTION