# SQL DRIVEN PIZZA ANALYTICS

Pizza Sales Report Presentation

# INTRODUCTION


This project focuses on analyzing pizza sales data to uncover valuable insights into customer behavior and sales trends. The SQL scripts developed address key business questions by analyzing data on orders placed, revenue generated, order distribution by hour of the day, category-wise pizza preferences, average pizzas ordered per day, cumulative revenue trends, and top-selling categories. By exploring when sales peak, which pizzas are most popular, and what drives customer choices, this analysis aims to inform decision-making and optimize sales strategies.

**Data Sources:**
The analysis utilizes datasets related to pizza sales, including Order_details, Orders, Pizzas, and Pizza_types, providing detailed information on pizza varieties and customer orders.

**Project Overview:**
The project entails data analysis for pizza sales by crafting SQL queries to address business needs and draw actionable insights, ultimately enhancing sales and strategy formulation.

# RETRIEVE THE TOTAL NUMBER OF ORDERS PLACE

```sql
SELECT
    COUNT(order_id) AS 'total orders'
FROM
    orders;
```

| total orders |
| --- |
| 21350 |

# CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES

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

| total revenue |
| --- |
| 817860.05 |

# IDENTIFY THE HIGHEST-PRICED PIZZA

```sql
SELECT
    p.pizza_type_id, pt.name, p.price
FROM
    pizzas p
        LEFT JOIN
    pizza_types pt ON p.pizza_type_id = pt.pizza_type_id
WHERE
    p.price = (SELECT
            MAX(price)
        FROM
            pizzas);
```

| | pizza_type_id | name | price |
|---|---|---|---|
| ▶ | the_greek | The Greek Pizza | 35.95 |

# IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED

```sql
SELECT
    p.pizza_type_id, pt.name, p.price
FROM
    pizzas p
        LEFT JOIN
    pizza_types pt ON p.pizza_type_id = pt.pizza_type_id
WHERE
    p.price = (SELECT
            MAX(price)
        FROM
            pizzas);
```

| | size | Size_count |
|---|---|---|
| ▶ | L | 18526 |

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

```sql
SELECT
    pt.pizza_type_id,
    pt.name,
    SUM(od.quantity) AS 'Total_Quantity'
FROM
    pizzas p
        JOIN
    order_details od ON p.pizza_id = od.pizza_id
        JOIN
    pizza_types pt ON pt.pizza_type_id = p.pizza_type_id
GROUP BY pt.pizza_type_id , pt.name
ORDER BY Total_Quantity DESC
LIMIT 5;
```

| pizza_type_id | name | Total_Quantity |
|---|---|---|
| classic_dlx | The Classic Deluxe Pizza | 2453 |
| bbq_ckn | The Barbecue Chicken Pizza | 2432 |
| hawaiian | The Hawaiian Pizza | 2422 |
| pepperoni | The Pepperoni Pizza | 2418 |
| thai_ckn | The Thai Chicken Pizza | 2371 |

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

```sql
SELECT
    pt.category, SUM(od.quantity) AS 'Quantity_ordered'
FROM
    pizzas p
        JOIN
    order_details od ON p.pizza_id = od.pizza_id
        JOIN
    pizza_types pt ON pt.pizza_type_id = p.pizza_type_id
GROUP BY pt.category;
```

| | category | Quantity_ordered |
|---|---|---|
| ▶ | Classic | 14888 |
| | Veggie | 11649 |
| | Supreme | 11987 |
| | Chicken | 11050 |

# DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY

```sql
SELECT
    HOUR(order_time) as 'Hour', COUNT(order_id) as 'Order_count'
FROM
    orders
GROUP BY HOUR(order_time);
```

| Hour | Order_count |
|---|---|
| 12 | 2520 |
| 13 | 2455 |
| 18 | 2399 |
| 17 | 2336 |
| 19 | 2009 |
| 16 | 1920 |
| 20 | 1642 |
| 14 | 1472 |
| 15 | 1468 |
| 11 | 1231 |
| 21 | 1198 |
| 22 | 663 |
| 23 | 28 |

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

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

| | 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

```sql
SELECT
    ROUND(AVG(Order_qty)) AS 'AVG_ORDER_QUANTITY'
FROM
    (SELECT
        o.order_date, ROUND(SUM(od.quantity)) AS 'Order_qty'
    FROM
        orders o
    JOIN order_details od ON o.order_id = od.order_id
    GROUP BY o.order_date) AS A;
```

| | AVG_ORDER_QUANTITY |
|---|---|
| ▶ | 138 |

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

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

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

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

# ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME

```sql
select order_date, sum(revenue) over (order by order_date) as 'cumulative_reveue'
from (
select o.order_date, sum(p.price*od.quantity) as 'revenue'
from
 pizzas p
        JOIN
    order_details od ON p.pizza_id = od.pizza_id
        JOIN
    orders o ON od.order_id = o.order_id
    group by o.order_date
    ) as A
    ORDER BY
    order_date;
```

| order_date | cumulative_reveue |
|---|---|
| 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 |

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

```sql
With A as
(
select pt.category,pt.name, sum(p.price*od.quantity) as 'revenue',
rank() over (partition by pt.category order by sum(p.price*od.quantity) desc) as 'rnk'
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,pt.category
)
select category, name, round(revenue,2) as 'Revenue'
from A
where rnk<=3
order by category, revenue desc;
```

| category | name | Revenue |
|---|---|---|
| Chicken | The Thai Chicken Pizza | 43434.25 |
| Chicken | The Barbecue Chicken Pizza | 42768 |
| Chicken | The California Chicken Pizza | 41409.5 |
| Classic | The Classic Deluxe Pizza | 38180.5 |
| Classic | The Hawaiian Pizza | 32273.25 |
| Classic | The Pepperoni Pizza | 30161.75 |
| Supreme | The Spicy Italian Pizza | 34831.25 |
| Supreme | The Italian Supreme Pizza | 33476.75 |
| Supreme | The Sicilian Pizza | 30940.5 |

# CONCLUSION

The Pizza Sales Analysis project highlights key sales trends and customer preferences, uncovering peak demand periods, top-selling items, and revenue patterns. These insights help enhance operational efficiency, align offerings with customer needs, and support strategic decision-making.

This project demonstrates my SQL expertise in analyzing real-life datasets, identifying trends, and evaluating product performance. It has also strengthened my skills in relational databases, efficient query writing, and deriving actionable insights to solve business challenges.