MENTOR NESS

PIZZA SALES ANALYSIS USING SQL

By Pranali Mane

Objectives of the analysis

- Identify Top-Selling Pizzas
- Analyze Sales Trends Over Time
- Customer Purchase Patterns
- Evaluate Marketing Promotions
- Revenue and Profit Analysis

Dataset

The analysis will be performed using four tables: order_details, orders, pizza_type, and pizzas.

1. order_details
- order_details_id: Unique identifier for the order detail.
- order_id: Identifier linking to the orders table.
- pizza_id: Identifier linking to the pizza table.
- quantity: Number of pizzas ordered.

2. orders
- order_id: Unique identifier for the order.
- date: Date the order was placed.
- time: Time the order was placed.

3. pizza_type
- pizza_type_id: Unique identifier for the pizza type.
- name: Name of the pizza.
- category: Category of the pizza (e.g., vegetarian, meat, etc.).
- ingredients: List of ingredients used in the pizza.

4. pizzas
- pizza_id: Unique identifier for the pizza.
- pizza_type_id: Identifier linking to the pizza_type table.
- size: Size of the pizza (e.g., small, medium, large).
- price: Price of the pizza.

Problem Statements

Q1: The total number of order place

Q2: The total revenue generated from pizza sales

Q3: The highest priced pizza.

Q4: The most common pizza size ordered.

Q5: The top 5 most ordered pizza types along their quantities.

Q6: The quantity of each pizza categories ordered.

Q7: The distribution of orders by hours of the day.

Problem Statements

Q8: The category-wise distribution of pizzas.

Q9: The average number of pizzas ordered per day.

Q10: Top 3 most ordered pizza type base on revenue.

Q11: The percentage contribution of each pizza type to revenue.

Q12: The cumulative revenue generated over time.

Q13: The top 3 most ordered pizza type based on revenue for each pizza category.

Solutions

Q1. The total number of order place

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

Result Grid

	total_order
▶	21350

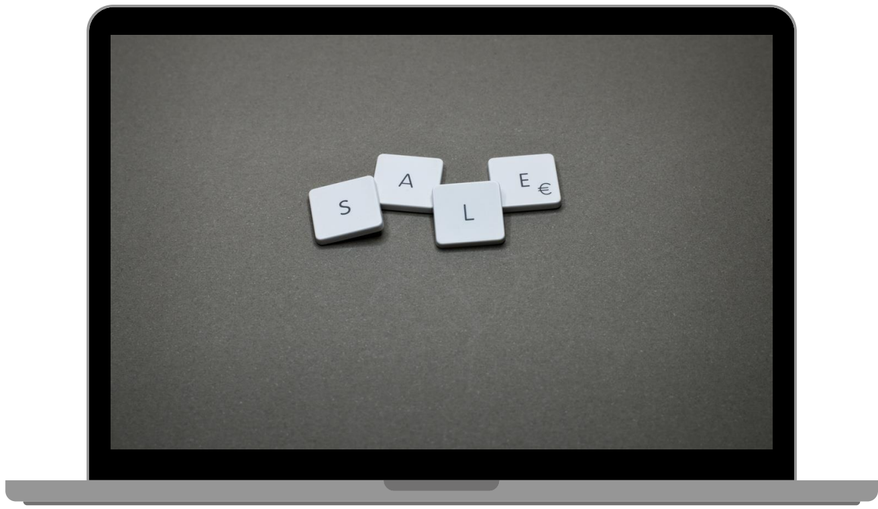
Solutions

Q2: The total revenue generated from pizza sales

```
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) as total_revenue
FROM
    order_details
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id;
```

Result Grid

total_revenue
817860.05



Solutions

Q3: The highest priced pizza.

```sql
-- Q3: The 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 1;
```

Result Grid | Filter Rows:

name	price
The Greek Pizza	35.95

Q4: The most common pizza size ordered.

```
select pizzas.size, count(order_details.order_details_id) as order_count
FROM
    pizzas
        JOIN
    order_details
on
    order_details.pizza_id = pizzas.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

Q5: The top 5 most ordered pizza types along their quantities.

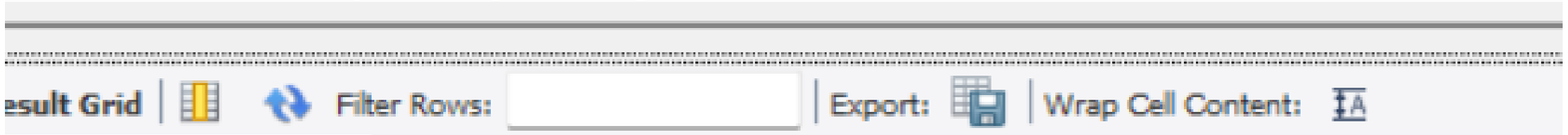
```sql
-- Q5: The top 5 most ordered pizza types along their quantities.
SELECT
    pizza_types.name, SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT 5;
```

Result Grid | Filter Rows:

name	quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

Q6: The quantity of each pizza categories ordered.

```sql
-- Q6: The quantity of each pizza categories ordered.
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS quantity
FROM
    order_details
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id
        JOIN
    pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
GROUP BY pizza_types.category
ORDER BY quantity DESC ;
```



category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Q7: The distribution of orders by hours of the day.

```
 SELECT
    HOUR(time) AS hours, COUNT(order_id) AS order_count
FROM
    orders
GROUP BY HOUR(orders.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

Q8: The category-wise distribution of pizzas.

```sql
-- Q8: The category-wise distribution of pizzas.
SELECT
    category, COUNT(name)
FROM
    pizza_types
GROUP BY category;
```

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

category	COUNT(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9

Q9: The average number of pizzas ordered per day.

```
3 •  SELECT
4        ROUND(AVG(quantity), 0) AS pizas_ordered_per_day
5    FROM
6        (SELECT
7            orders.date, SUM(order_details.quantity) AS quantity
8        FROM
9            orders
10       JOIN order_details ON orders.order_id = order_details.order_id
11       GROUP BY orders.date) AS order_quantity;
```

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

pizas_ordered_per_day
138

Q10: Top 3 most ordered pizza type base on revenue.

```
2 •   SELECT pizza_types.name,
3         SUM(order_details.quantity * pizzas.price) AS revenue
4     FROM
5         pizzas
6             JOIN
7         pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
8             JOIN
9         order_details ON order_details.pizza_id = pizzas.pizza_id
10    GROUP BY pizza_types.name
11    ORDER BY revenue DESC
12    LIMIT 3;
```

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

name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

Q11: The percentage contribution of each pizza type to revenue.

```sql
SELECT pizza_types.category,
    ROUND(SUM(order_details.quantity * pizzas.price) / (SELECT
                    SUM(order_details.quantity * pizzas.price) AS total_revenue
                FROM
                    order_details
                        JOIN
                    pizzas ON order_details.pizza_id = pizzas.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;
```

Result Grid | Filter Rows:

category	revenue_percentage
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

Q12: The cumulative revenue generated over time.

```
2 •  select date, sum(revenue) over(order by date) as cum_revenue from
3    (SELECT
4        orders.date,
5        ROUND(SUM(order_details.quantity * pizzas.price),
6                2) AS revenue
7    FROM
8        order_details
9            JOIN
10       pizzas ON order_details.pizza_id = pizzas.pizza_id
11           JOIN
12       orders ON orders.order_id = order_details.order_id
13   GROUP BY orders.date) as sales;
```

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

date	cum_revenue
2015-01-01	2713.85
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

Q13: The top 3 most ordered pizza type based on revenue for each pizza category.

```sql
-- Q13: The top 3 most ordered pizza type based on revenue for each pizza category.
select category,name, revenue from
(select category,name,
revenue,rank() over(partition by category order by revenue desc) as rank1 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 rank1 <= 3;
```

Result Grid | Filter Rows: | Export:

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
Veggie	The Four Cheese Pizza	32265.70000000065
Veggie	The Mexicana Pizza	26780.75
Veggie	The Five Cheese Pizza	26066.5

Conclusion

The pizza sales analysis project using SQL has provided a comprehensive view of various aspects of the business, offering actionable insights that can drive strategic decision making. Through detailed analysis of sales data, customer behavior, and operational metrics the project has achieved several key objectives:

- ❑ Enhanced understanding of sales performance
- ❑ Insights into product popularity
- ❑ Customer behavior analysis
- ❑ Sales by category
- ❑ Effective reporting


thank you