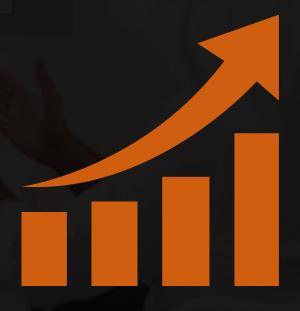# SALES REPORT

2024

presented by
VISHESH VERMA

$\longrightarrow$

INTRODUCTION

Welcome to our sales report presentation. In this report, we will analyze and review our company's sales performance over a specific period. By examining key metrics and trends, we aim to provide solution of the asked questions that inform decision-making and drive business growth.

presented by
VISHESH VERMA

ABOUT PROJECT

This project aims to explore the " PIZZA SALES DATA " to understand the top perform item and category of pizza , trends and performance of the product. The main aim is to study how sale strategies can be improved and optimized.

BUSINESS QUESTIONS TO ANSWER

The questions are divided in three categories :
1. the beginner level
2. the intermediate level
3. the advance level

BEGINNER QUESTIONS

1. Retrieve the total number of orders placed.

2. Calculate the total revenue generated from pizza sales.

3. Identify the highest-priced pizza.

4. Identify the most common pizza size ordered.

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

INTERMEDIATE QUESTIONS

1. Join the necessary tables to find the total quantity of each pizza category ordered.

2. Determine the distribution of orders by hour of the day.

3. Join relevant tables to find the category-wise distribution of pizzas.

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

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

ADVANCE QUESTIONS

1. Calculate the percentage contribution of each pizza type to total revenue.

2. Analyze the cumulative revenue generated over time.

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

INSIGHTS AFTER ANALYSIS

- **BEGINNER QUESTIONS**

1.

```
1    -- Retrieve the total number of orders placed.
2 •  select count(order_id) as total_orders from orders;
```

total_orders
▶ 21350

2.

```sql
1      -- Calculate the total revenue generated from pizza sales.
2
3  •   SELECT
4          ROUND(SUM(order_details.quantity * pizzas.price),
5              2) AS total_revenue
6      FROM
7          order_details
8              JOIN
9          pizzas ON pizzas.pizza_id = order_details.pizz_id
```
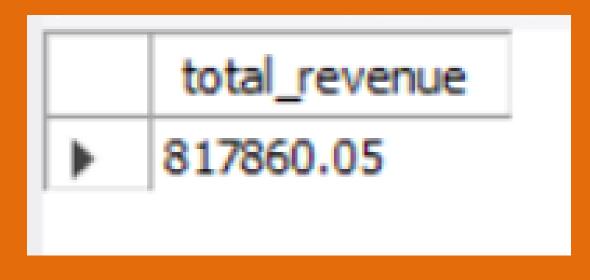
total_revenue
817860.05

3.

```sql
1    -- Identify the highest-priced pizza
2
3 •  SELECT
4        pizza_types.name, pizzas.price
5    FROM
6        pizza_types
7            JOIN
8        pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
9    ORDER BY pizzas.price DESC
10   LIMIT 1;
```

name	price
▶ The Greek Pizza	35.95

4.

```sql
1    -- Identify the most common pizza size ordered.
2
3    SELECT
4        pizzas.size,
5        COUNT(order_details.order_details_id) AS order_count
6    FROM
7        pizzas
8            JOIN
9        order_details ON pizzas.pizza_id = order_details.pizz_id
10   GROUP BY pizzas.size
11   ORDER BY order_count DESC
12   LIMIT 1;
```

	size	order_count
▶	L	18526

5.

```sql
1    -- List the top 5 most ordered pizza types
2    -- along with their quantities.
3
4 •  SELECT
5        pizza_types.name,
6        SUM(order_details.quantity) AS quantity_ordered
7    FROM
8        pizza_types
9            JOIN
10       pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
11           JOIN
12       order_details ON order_details.pizz_id = pizzas.pizza_id
13   GROUP BY pizza_types.name
14   ORDER BY quantity_ordered DESC
15   LIMIT 5;
```

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

- **INTERMEDIATE QUESTIONS**

1.

```
1    -- Join the necessary tables to find the
2    -- total quantity of each pizza category ordered.
3
4 •  SELECT
5        pizza_types.category,
6        SUM(order_details.quantity) AS Quantity
7    FROM
8        pizza_types
9            JOIN
10       pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
11           JOIN
12       order_details ON order_details.pizz_id = pizzas.pizza_id
13   GROUP BY category
14   ORDER BY Quantity DESC;
```

category	Quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

2.

```
1    -- Determine the distribution of orders by hour of the day.
2
3 •  select hour(order_time) , count(orders.order_id)
4    from orders
5    group by hour(order_time);
```

hour(order_time)	count(orders.order_id)
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

3.

```
1    -- Join relevant tables to find the category-wise distribution of pizzas.
2
3 •  select category , count(name) from pizza_types
4    group by category;
```

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

4.

```sql
1     -- Group the orders by date and calculate the average number of pizzas ordered per day.
2   • SELECT
3         AVG(Quantity)
4     FROM
5         (SELECT
6             order_date, SUM(order_details.quantity) AS Quantity
7         FROM
8             orders
9         JOIN order_details ON orders.order_id = order_details.order_id
10        GROUP BY orders.order_date) AS order_quantity;
11
```

order_date	Quantity
2015-01-01	162
2015-01-02	165
2015-01-03	158
2015-01-04	106
2015-01-05	125
2015-01-06	147
2015-01-07	138
2015-01-08	173
2015-01-09	127
2015-01-10	146
2015-01-11	116
2015-01-12	119
2015-01-13	120
2015-01-14	150

5.

```sql
1    -- Determine the top 3 most ordered pizza types based on revenue.
2
3 •  SELECT
4        pizza_types.name,
5        SUM(order_details.quantity * pizzas.price) AS revenue
6    FROM
7        pizza_types
8            JOIN
9        pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
10           JOIN
11       order_details ON order_details.pizz_id = pizzas.pizza_id
12   GROUP BY pizza_types.name
13   ORDER BY revenue DESC
14   LIMIT 3;
```

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

- ## ADVANCE QUESTIONS

1.

```
1      -- Calculate the percentage contribution of each pizza type to total revenue.
2
3 ●    SELECT
4          pizza_types.category,
5          (SUM(order_details.quantity * pizzas.price) / (SELECT
6                  ROUND(SUM(order_details.quantity * pizzas.price),
7                          2) AS total_sales
8             FROM
9                 order_details
10                    JOIN
11                pizzas ON pizzas.pizza_id = order_details.pizz_id)) * 100 AS revenue
12     FROM
13        pizza_types
14            JOIN
15        pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
16            JOIN
17        order_details ON order_details.pizz_id = pizzas.pizza_id
18     GROUP BY pizza_types.category
19     ORDER BY revenue DESC;
```

category	revenue
Classic	26.90596025566967
Supreme	25.45631126009862
Chicken	23.955137556847287
Veggie	23.682590927384577

2.

```
1       -- Analyze the cumulative revenue generated over time.
2
3  ●    select order_date ,
4       sum(revenue) over (order by order_date) as cum_revenue
5       from
6  ⊖    (select orders.order_date , round(sum(order_details.quantity * pizzas.price) , 2) as revenue
7       from orders join order_details
8       on orders.order_id = order_details.order_id
9       join pizzas
10      on pizzas.pizza_id = order_details.pizz_id
11      group by orders.order_date) as sales;
```

order_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
2015-01-09	21526.399999999998
2015-01-10	23990.35
2015-01-11	25862.649999999998
2015-01-12	27781.699999999997
2015-01-13	29831.299999999996
2015-01-14	32358.699999999997
2015-01-15	34343.5
2015-01-16	36937.65
2015-01-17	39001.75
2015-01-18	40978.6

3.

```sql
1    -- Determine the top 3 most ordered pizza types based on revenue for each pizza category.
2
3 ●  select category , name , revenue from
4    (select category , name , revenue ,
5    rank() over(partition by category order by revenue desc) as rn
6    from
7    (select pizza_types.category , pizza_types.name ,
8    round(sum(order_details.quantity * pizzas.price) , 2) as revenue
9    from pizza_types join pizzas
10   on pizza_types.pizza_type_id = pizzas.pizza_type_id
11   join order_details
12   on order_details.pizz_id = pizzas.pizza_id
13   group by pizza_types.category , pizza_types.name)as a)as b
14   where rn <= 3;
```

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	32265.7
Veggie	The Mexicana Pizza	26780.75
Veggie	The Five Cheese Pizza	26066.5

THANK YOU!

Thank you for your attention to our sales report presentation. If you have any questions or would like to discuss the findings in more detail, please don't hesitate to reach out to our sales team. We appreciate your continued support and partnership.