

PIZZA SALES ANALYSIS



PROJECT OVERVIEW

Objective:

- Conducted an Exploratory Data Analysis (EDA) on pizza sales data to uncover key business insights and trends.
- Analyzed a dataset containing information about orders, order details, pizzas, and pizza types to generate valuable metrics for decision-making.

Key Goals:

- To analyze sales trends, identify popular pizzas and sizes, and calculate revenue.
- To evaluate peak order times and provide business intelligence for better decision-making.



KEY INSIGHTS & ANALYSIS

1. Sales and Revenue Insights:

- Total Orders: Calculated the total number of pizza orders in the dataset.
- Total Revenue: Determined the overall revenue generated from pizza sales.
- Top 3 Pizza Types by Revenue: Identified the highest revenue-generating pizza types and their quantities.

2. Popular Pizzas & Categories:

- Most Ordered Pizzas: Listed the top 5 most ordered pizzas along with their quantities.
- Pizza Size Analysis: Identified the most common pizza size ordered across all sales.
- Pizza Categories: Analyzed category-wise pizza distribution and total quantities ordered.



KEY INSIGHTS & ANALYSIS

3. Time-based Analysis:

- Orders by Hour: Analyzed the distribution of orders by the hour of the day, helping identify peak times for pizza orders.
- Average Orders per Day: Calculated the average number of pizzas ordered each day.

4. Revenue Distribution & Cumulative Analysis:

- Revenue Contribution: Calculated the percentage contribution of each pizza type to total revenue.
- Cumulative Revenue: Analyzed the cumulative revenue generated over time to track sales performance.



LARANA PIZZA

```
14    -- RETIEVE THE TOTAL NUMBER ORDERS  
15  
16 •   SELECT COUNT(ORDER_ID) AS TOTAL_ORDERS FROM ORDERS;  
17
```

Result Grid				Filter Rows:	<input type="text"/>	Export:		Wrap Cell Content:		
<table border="1"><thead><tr><th>TOTAL_ORDERS</th></tr></thead><tbody><tr><td>21350</td></tr></tbody></table>									TOTAL_ORDERS	21350
TOTAL_ORDERS										
21350										



LARANA PIZZA

```
18      -- CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES  
19  
20 •   SELECT  
21     ROUND(SUM(ORDER_DETAILS.QUANTITY * PIZZAS.PRICE),  
22           2) AS TOTAL_REVENUE  
23   FROM  
24     ORDER_DETAILS  
25     INNER JOIN  
26     PIZZAS ON ORDER_DETAILS.PIZZA_ID = PIZZAS.PIZZA_ID;  
27
```

Result Grid			Filter Rows:	<input type="text"/>	Export:		Wrap Cell Content:	
TOTAL_REVENUE								
817860.05								



LARANA PIZZA

```
28    -- IDENTIFY THE HIGHEST PRICED PIZZA
29
30 •  SELECT
31      PIZZA_TYPES.NAME, PIZZAS.PRICE
32  FROM
33      PIZZA_TYPES
34          INNER JOIN
35      PIZZAS ON PIZZA_TYPES.PIZZA_TYPE_ID = PIZZAS.PIZZA_TYPE_ID
36 WHERE
37      PIZZAS.PRICE = (SELECT
38          MAX(PRICE)
39      FROM
40          PIZZAS);
41
```

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

	NAME	PRICE
▶	The Greek Pizza	35.95



LARANA PIZZA

```
42    -- IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED
43
44 •   SELECT
45      PIZZAS.SIZE, COUNT(ORDER_DETAILS.ORDER_DETAIL_ID) AS ORDER_COUNT
46  FROM
47      ORDER_DETAILS
48      INNER JOIN
49          PIZZAS ON ORDER_DETAILS.PIZZA_ID = PIZZAS.PIZZA_ID
50  GROUP BY PIZZAS.SIZE
51  ORDER BY ORDER_COUNT DESC
52  LIMIT 1;
53
```

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

SIZE	ORDER_COUNT
L	18526



LARANA PIZZA

```
54      -- LIST THE TOP 5 MOST ORDERED PIZZAS ALONG WITH THEIR QUANTITIES
55
56 •   SELECT
57         A.PIZZA_ID, SUM(B.QUANTITY) AS QUANTITY
58     FROM
59         PIZZAS AS A
60             INNER JOIN
61                 ORDER_DETAILS AS B ON A.PIZZA_ID = B.PIZZA_ID
62             GROUP BY A.PIZZA_ID
63             ORDER BY QUANTITY DESC
64             LIMIT 5;
65
```

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

	PIZZA_ID	QUANTITY
▶	big_meat_s	1914
	thai_dkn_l	1410
	five_cheese_l	1409
	four_cheese_l	1316
	classic_dlx_m	1181



LARANA PIZZA

```
66    -- LIST THE TOP 5 MOST ORDERED PIZZAS TYPES ALONG WITH THEIR QUANTITIES
67
68 •  SELECT
69      PIZZA_TYPES.NAME, SUM(ORDER_DETAILS.QUANTITY) AS QUANTITY
70  FROM
71      PIZZA_TYPES
72      JOIN
73          PIZZAS ON PIZZA_TYPES.PIZZA_TYPE_ID = PIZZAS.PIZZA_TYPE_ID
74      JOIN
75          ORDER_DETAILS ON PIZZAS.PIZZA_ID = ORDER_DETAILS.PIZZA_ID
76  GROUP BY PIZZA_TYPES.NAME
77  ORDER BY QUANTITY DESC
78  LIMIT 5;
79
```

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



LARANA PIZZA

```
80      -- JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED
81
82 •  SELECT
83      PIZZA_TYPES.CATEGORY, SUM(ORDER_DETAILS.QUANTITY) AS QUANTITY
84  FROM
85      PIZZA_TYPES
86      JOIN
87      PIZZAS ON PIZZA_TYPES.PIZZA_TYPE_ID = PIZZAS.PIZZA_TYPE_ID
88      JOIN
89      ORDER_DETAILS ON PIZZAS.PIZZA_ID = ORDER_DETAILS.PIZZA_ID
90  GROUP BY PIZZA_TYPES.CATEGORY
91  ORDER BY QUANTITY DESC;
92
```

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

	CATEGORY	QUANTITY
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050



LARANA PIZZA

```
93      -- DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY  
94  
95 •  SELECT  
96      HOUR(ORDER_TIME) AS HOUR, COUNT(ORDER_ID) AS ORDERS  
97  FROM  
98  ORDERS  
99  GROUP BY HOUR(ORDER_TIME);  
100
```

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

	HOUR	ORDERS
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336



LARANA PIZZA

```
101    -- JOIN RELEVANT TABLES TO FIND THE CATEGORY WISE DISTRIBUTION OF PIZZAS
102
103 • SELECT
104     CATEGORY, COUNT(NAME) AS TOTAL_PIZZAS
105   FROM
106     PIZZA_TYPES
107 GROUP BY CATEGORY;
108
```

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

	CATEGORY	TOTAL_PIZZAS
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9



LARANA PIZZA

```
109      -- GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY  
110  
111 •   SELECT  
112         ROUND(AVG(TOTAL_QUANTITY),0) AS AVERAGE  
113     FROM  
114     (SELECT  
115         ORDERS.ORDER_DATE,  
116             SUM(ORDER_DETAILS.QUANTITY) AS TOTAL_QUANTITY  
117     FROM  
118         ORDERS  
119     JOIN ORDER_DETAILS ON ORDERS.ORDER_ID = ORDER_DETAILS.ORDER_ID  
120     GROUP BY ORDERS.ORDER_DATE) AS A;  
121
```

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

	AVERAGE
▶	138



THANK YOU

