

Pizza Sales Insights

SQL Project

PRESENTED BY: PRANAV BHAWANE



Objective

Basic Analysis:

To gain a foundational understanding of sales data, focusing on order count, revenue, pricing, and popular items.

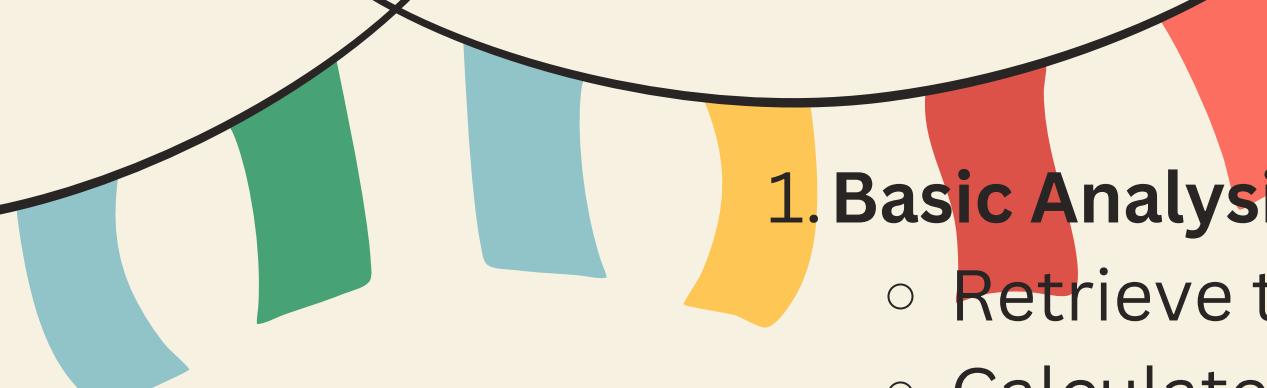
Intermediate Analysis:

To perform more detailed analytics, such as category-wise distribution and time-based trends.

Advanced Analysis:

To derive deeper insights into revenue contribution and detailed performance metrics for each pizza type and category.





Actions



1. Basic Analysis:

- Retrieve the total number of orders placed.
- Calculate the total revenue generated from pizza sales.
- Identify the highest-priced pizza.
- Identify the most common pizza size ordered.
- List the top 5 most ordered pizza types along with their quantities.

2. Intermediate Analysis:

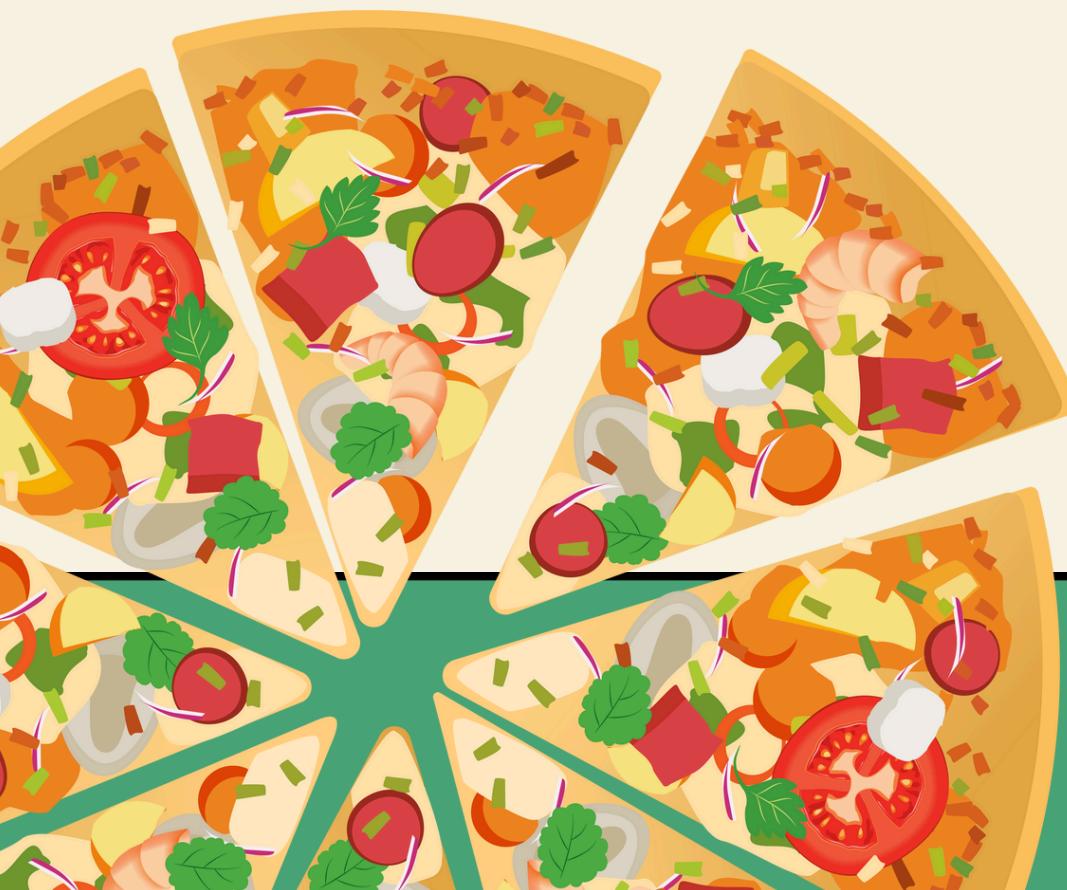
- Join the necessary tables to find the total quantity of each pizza category ordered.
- Determine the distribution of orders by hour of the day.
- Join relevant tables to find the category-wise distribution of pizzas.
- Group the orders by date and calculate the average number of pizzas ordered per day.
- Determine the top 3 most ordered pizza types based on revenue.

3. Advanced Analysis:

- Calculate the percentage contribution of each pizza type to total revenue.
- Analyze the cumulative revenue generated over time.
- Determine the top 3 most ordered pizza types based on revenue for each pizza category.

Basic Analysis:

Total number of orders



--- Retrieve the total number of orders placed

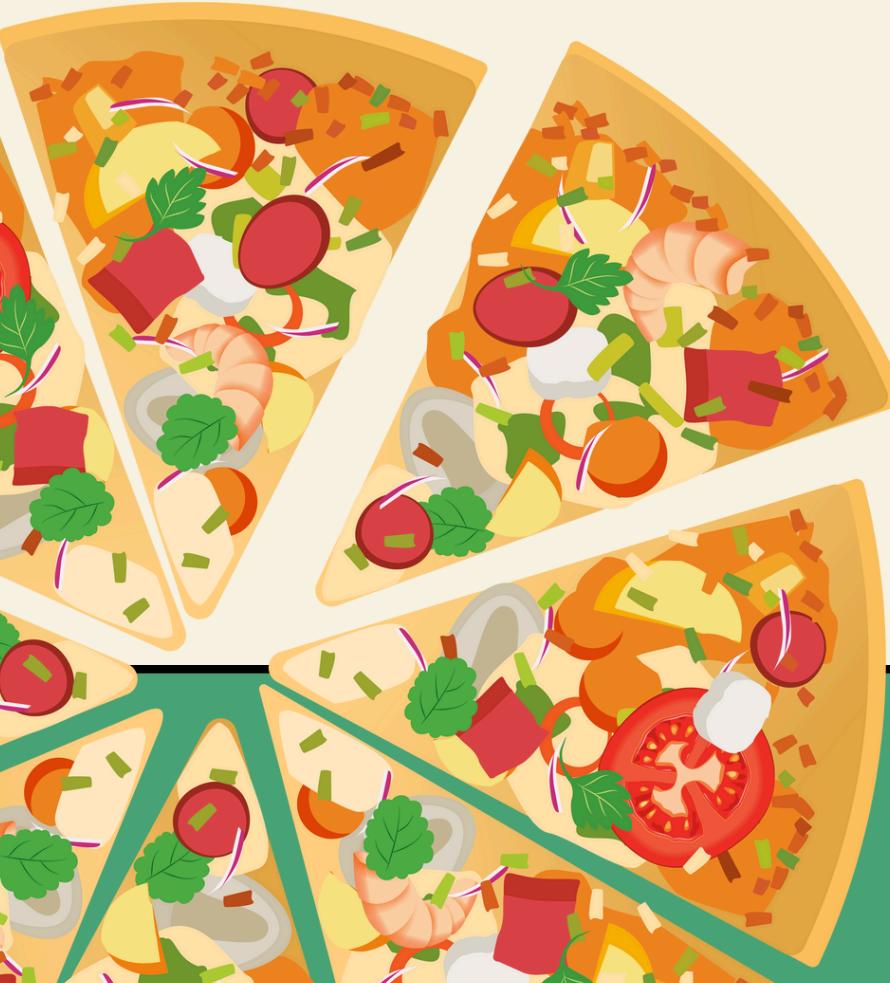
```
1 --- Retrieve the total number of orders placed
2 • SELECT
3     COUNT(order_id) AS Total_Orders
4 FROM
5     orders
```

Result Grid | Filter Rows: Export: Wrap Cell Co

Total_Orders
21350

Basic Analysis:

Total revenue generated:



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

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

Revenue
817860

Basic Analysis:

Highest-priced pizza:



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

The Result Grid shows the following data:

	name	price
▶	The Greek Pizza	35.95

Basic Analysis:

Most common pizza size ordered:



sqlproject* SQL File 4* x SQL File 5* SQL File 6* SQL File 7* SQL File 9*

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

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

	size	order_count
▶	L	18526

Basic Analysis:

Top 5 most ordered pizza types
and quantities



```
sqlproject* SQL File 4* SQL File 5* X SQL File 6* SQL File 7* SQL File 9* SQL File 10* S
1      -- List the top 5 most ordered pizza types
2      -- along with their quantities.--
3
4 •   SELECT
5         pizza_types.name, SUM(order_details.quantity) AS quantity
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.pizza_id = pizzas.pizza_id
12        GROUP BY pizza_types.name
13        ORDER BY quantity DESC
14        LIMIT 5;
```

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

Intermediate Analysis:

- Total quantity of each pizza category ordered



sqlproject* SQL File 4* SQL File 5* SQL File 6* × SQL File 7* SQL File 9*

2 -- quantity of each pizza category ordered.

3 • SELECT

4 pizza_types.category,

5 SUM(order_details.quantity) AS quantity

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.pizza_id = pizzas.pizza_id

12 GROUP BY pizza_types.category

13 ORDER BY quantity DESC

14

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

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

Intermediate Analysis:

- Distribution of orders by hour



sqlproject* SQL File 4* SQL File 5* SQL File 6* SQL File 7* X SQL File 9*

1 -- Determine the distribution of orders by hour of the day.
2 • SELECT
3 HOUR(time) AS hour, COUNT(order_id) AS order_count
4 FROM
5 orders
6 GROUP BY HOUR(time);
7

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

	hour	order_count
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468

Intermediate Analysis:

Category-wise distribution of pizzas



sqlproject* SQL File 4* SQL File 5* SQL File 6* SQL File 7* SQL File 9* X SQL File 10

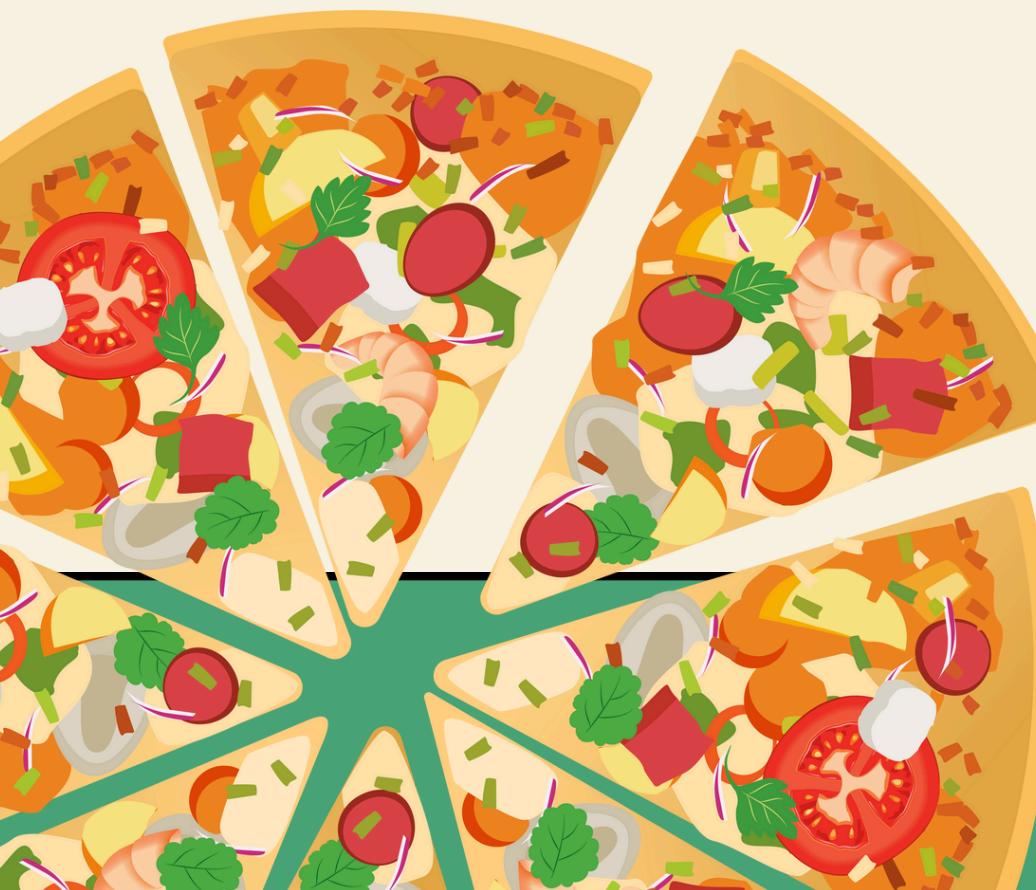
1 -- Join relevant tables to find the category-wise distribution of pizzas.
2 • **SI** Execute the selected portion of the script or everything, if there is no selection
3 category, COUNT(name)
4 FROM
5 pizza_types
6 GROUP BY category;

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

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

Intermediate Analysis:

Average number of pizzas ordered per day



```
sqlproject* SQL File 4* SQL File 5* SQL File 6* SQL File 7* SQL File 9* SQL
1      -- Group the orders by date and calculate
2      -- the average number of pizzas ordered per day.
3 •  SELECT
4          ROUND(AVG(quantity), 0) as Orders_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: |

Orders_per_Day
138

Intermediate Analysis:

Top 3 most ordered pizza types based on revenue



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

Advanced Analysis

Percentage contribution of each pizza type to total revenue:



sqlproject* SQL File 4* SQL File 5* SQL File 6* SQL File 7* SQL File 9* SQL File 10*

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

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

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

Advanced Analysis

Top 3 most ordered pizza types based on revenue for each pizza category



sqlproject* SQL File 4* SQL File 5* SQL File 6* SQL File 7* SQL File 9* SQL File 10* S

1 -- List the top 5 most ordered pizza types
2 -- along with their quantities...
3
4 • SELECT
5 pizza_types.name, SUM(order_details.quantity) AS quantity
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.pizza_id = pizzas.pizza_id
12 GROUP BY pizza_types.name
13 ORDER BY quantity DESC
14 LIMIT 5;

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