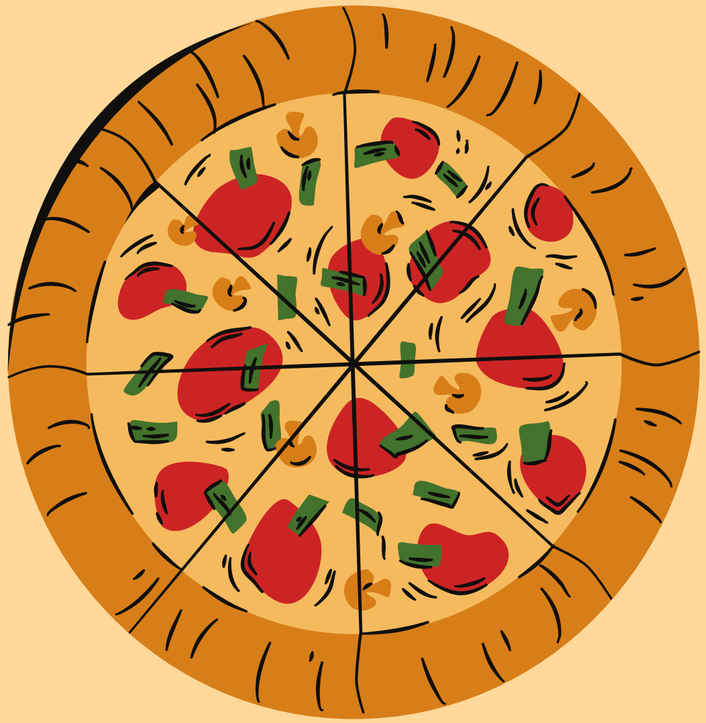


Delicious Pizza for Everyone!

PIZZA ANALYSIS

@rajnish_coder

Hello!

Title: Slicing Through Success: A Delectable Analysis of Pizza Sales

Subtitle: Unveiling Insights, Crust by Crust

Introduction:

Welcome to the world of pizza analysis, where every slice tells a story! In this presentation, we embark on a flavorful journey through the data, uncovering trends, patterns, and opportunities in pizza sales. From classic margheritas to gourmet specialties, join us as we explore the doughy landscape of success. Get ready to devour insights that will leave you craving for more!

Our Dataset

1 - ORDER DETAILS DATASET

This crucial dataset contains the following columns:

Order Details ID: A unique identifier for each order detail entry.

Order ID:An identifier linking each order detail to its corresponding order.

Pizza ID:An identifier specifying the type of pizza ordered.

Quantity:The quantity of pizzas ordered for each specific type.

2 - ORDERS DATASET

This dataset encompasses the following essential columns:

Order ID: A unique identifier assigned to each order.

Order Date:The date when the order was placed.

Order Time:The timestamp indicating the time of order placement.

Our Dataset

3 - PIZZA TYPES DATASET

- This dataset comprises the following key Columns.

Pizza Type ID:A unique identifier for each pizza type.

Pizza Name:The name of the pizza, representing its identity and flavor profile.

Pizza Category:Categorization of pizzas based on characteristics such as size, crust type, or specialty.

Ingredients:A detailed list of ingredients used to craft each pizza, providing insight into flavor profiles and dietary considerations.

2 - PIZZAS DATASET

- This dataset comprises the following key columns:

Pizza ID:A unique identifier assigned to each pizza variant.

Pizza Type ID:An identifier linking each pizza to its corresponding type.

Size:The size of the pizza, indicating its diameter or portion.

Price:The cost of each pizza variant.

Retrieve the total number of orders placed.

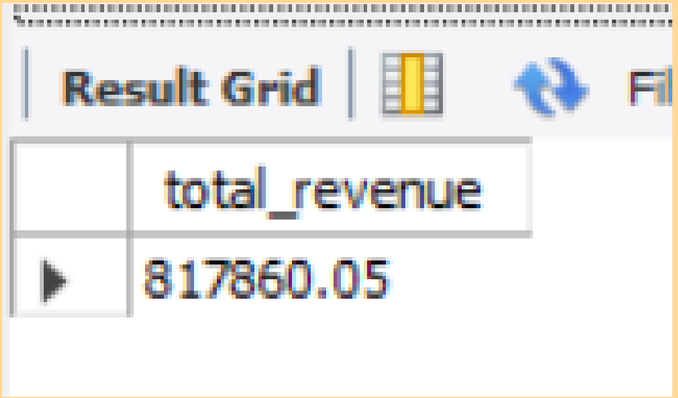
```
2
3 •  SELECT COUNT(order_id) AS 'No_of_order_placed'
4    FROM pizza_sales.orders;
```

Result Grid | Filter Rc

	No_of_order_placed
▶	21350

Calculate the total revenue generated from pizza sales.

```sql
USE pizza_sales;
SELECT ROUND(SUM((quantity * price)),2) AS 'total_revenue'
FROM order_details t1
JOIN pizzas t2
ON t1.pizza_id = t2.pizza_id
```



Identify the highest-priced pizza.

```
3 •  SELECT name,price
4    FROM pizzas t1
5    JOIN pizza_types t2
6    ON t1.pizza_type_id = t2.pizza_type_id
7    ORDER BY price DESC LIMIT 1
```

Result Grid | Filter Rows:

	name	price
▶	The Greek Pizza	35.95

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

```
5  SELECT name,COUNT(*) AS 'order_quantity'
6  FROM order_details t1
7  JOIN pizzas t2
8  ON t1.pizza_id = t2.pizza_id
9  JOIN pizza_types t3
10 ON t2.pizza_type_id = t3.pizza_type_id
11 GROUP BY name
12 ORDER BY order_quantity DESC LIMIT 5
```

Result Grid | Filter Rows:

name	order_quantity
The Classic Deluxe Pizza	2416
The Barbecue Chicken Pizza	2372
The Hawaiian Pizza	2370
The Pepperoni Pizza	2369
The Thai Chicken Pizza	2315

Identify the most common pizza size ordered.

```
3  SELECT size, SUM(quantity) AS 'most_common_size_pizza'
4  FROM order_details t1
5  JOIN pizzas t2
6  ON t1.pizza_id = t2.pizza_id
7  GROUP BY size
8  ORDER BY most_common_size_pizza DESC
```

Result Grid | Filter Rows:

size	most_common_size_pizza
L	18956
M	15635
S	14403
XL	552
XXL	28

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

```
4  SELECT category,SUM(quantity) AS 'ordered_quantity'
5  FROM order_details t1
6  JOIN pizzas t2
7  ON t1.pizza_id = t2.pizza_id
8  JOIN pizza_types t3
9  ON t2.pizza_type_id = t3.pizza_type_id
10 GROUP BY category
11 ORDER BY ordered_quantity DESC
```

Result Grid | Filter Rows:

category	ordered_quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Determine the distribution of orders by hour of the day.

```
3 •  SELECT HOUR(order_time) AS 'hour',
4    COUNT(*) AS 'order_count'
5    FROM orders
6    GROUP BY hour
7    ORDER BY order_count DESC;
```

Result Grid | Filter

hour	order_count
12	2520
13	2455
18	2399
17	2336
19	2009
16	1920
20	1642
14	1472
15	1468

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

```
2
3 •  SELECT category,COUNT(*) AS 'no_of_count'
4    FROM pizza_types
5    GROUP BY category
```

Result Grid | Filter Row

	category	no_of_count
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

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

```
4 •  SELECT ROUND(AVG(no_of_order),0) AS 'no_of_order_per_day'
5    FROM
6    (SELECT DATE(order_date),COUNT(*) AS 'no_of_order'
7    FROM order_details t1
8    JOIN orders t2
9    ON t1.order_id = t2.order_id
10   GROUP BY DATE(order_date)) t
```

Result Grid | Filter R

	no_of_order_per_day
▶	136

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

```sql
SELECT t3.name,
ROUND(SUM(t1.quantity*t2.price),2) AS 'revenue'
FROM order_details t1
JOIN pizzas t2
ON t1.pizza_id = t2.pizza_id
JOIN pizza_types t3
ON t2.pizza_type_id = t3.pizza_type_id
GROUP BY t3.name
ORDER BY revenue DESC LIMIT 3
```

Result Grid | Filter Rows:

name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768.00
The California Chicken Pizza	41409.50

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

```
3 •  SELECT t3.category,
4    (ROUND(SUM(t1.quantity*t2.price),2)/
5    (SELECT ROUND(SUM((quantity * price)),2) AS 'total_revenue'
6    FROM order_details t1
7    JOIN pizzas t2
8    ON t1.pizza_id = t2.pizza_id))*100 AS 'percentage'
9    FROM order_details t1
10   JOIN pizzas t2
11   ON t1.pizza_id = t2.pizza_id
12   JOIN pizza_types t3
13   ON t2.pizza_type_id = t3.pizza_type_id
14   GROUP BY t3.category
15   ORDER BY percentage DESC
```

Result Grid | Filter Rows:

category	percentage
Classic	26.905960
Supreme	25.456311
Chicken	23.955138
Veggie	23.682591

Analyze the cumulative revenue generated over time.

```
3  SELECT order_date,revenue,
4  SUM(revenue) OVER(ORDER BY revenue) AS 'cumulative revenue'
5  FROM(SELECT order_date,
6  ROUND(SUM(quantity*price),2) AS 'revenue'
7  FROM orders t1
8  JOIN order_details t2
9  ON t1.order_id = t2.order_id
10 JOIN pizzas t3
11 ON t2.pizza_id = t3.pizza_id
12 GROUP BY order_date) t
```

Result Grid | Filter Rows:

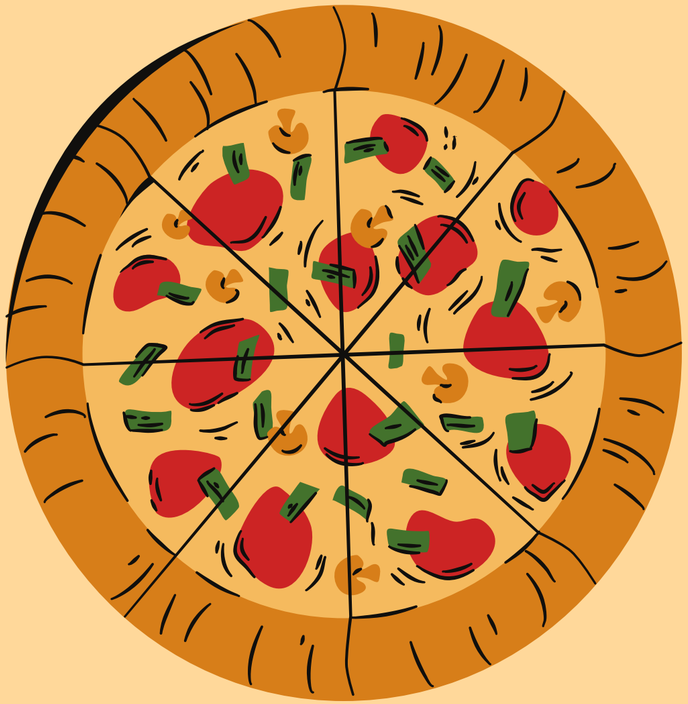
order_date	revenue	cumulative revenue
2015-03-22	1259.25	1259.25
2015-12-30	1337.80	2597.05
2015-12-29	1353.25	3950.30
2015-11-22	1368.70	5319.00
2015-12-27	1419.00	6738.00
2015-09-06	1491.65	8229.65

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

```sql
SELECT * FROM(SELECT name,category,revenue,
RANK() OVER(PARTITION BY category ORDER BY revenue DESC) AS 'rank'
FROM
(SELECT name,category,
ROUND(SUM(quantity*price),2) AS 'revenue'
FROM order_details t1
JOIN pizzas t2
ON t1.pizza_id = t2.pizza_id
JOIN pizza_types t3
ON t2.pizza_type_id = t3.pizza_type_id
GROUP BY name,category) t) tt
WHERE tt.rank < 4
```

Result Grid | Filter Rows: | Export:

name	category	revenue	rank
The Thai Chicken Pizza	Chicken	43434.25	1
The Barbecue Chicken Pizza	Chicken	42768.00	2
The California Chicken Pizza	Chicken	41409.50	3
The Classic Deluxe Pizza	Classic	38180.50	1
The Hawaiian Pizza	Classic	32273.25	2
The Pepperoni Pizza	Classic	30161.75	3
The Spicy Italian Pizza	Supreme	34831.25	1
The Italian Supreme Pizza	Supreme	33476.75	2
The Sicilian Pizza	Supreme	30940.50	3
The Four Cheese Pizza	Veggie	32265.70	1
The Mexicana Pizza	Veggie	26780.75	2
The Five Cheese Pizza	Veggie	26066.50	3



Pizza Boxcar Present

THANK YOU

@rajnish_coder